Utilização de Redes Bayesianas como Agrupador de Classificadores Locais e Global

Leonardo Nogueira Matos

Tese de Doutorado submetida ao Programa de Pós-Gradução em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

João Marques de Carvalho, PhD.
Orientador

Campina Grande, Paraíba, Brasil

M433u Matos, Leonardo Nogueira
Utilizacao de redes bayesianas como agrupador de classificadores locais e global / Leonardo Nogueira Matos. - Campina Grande, 2004.
129 f. : il.

Tese (Doutorado em Engenharia Eletrica) - Universidade Federal de Campina Grande, Centro de Ciencias e Tecnologia.

1. Computacao - 2. Reconhecimento Optico - 3. Tese I. Carvalho, Joao Marques de, Dr. II. Universidade Federal de Campina Grande - Campina Grande (PB) III. Título

CDU 004.855(043)

UTILIZAÇÃO DE REDES BAYESIANAS COMO AGRUPADOR DE CLASSIFICADORES LOCAIS E GLOBAIS

LEONARDO NOGUEIRA DE MATOS

Tese Aprovada em 25.10.2004

JOÃO MARQUES DE CARVALHO, Ph.D., UFCG
Orientador

FLÁVIO BORTOLOZZI, Dr., PUC-PR
Componente da Banca

NELSON DELFINO D'ÁVILA MASCARENHAS, Dr., UFSCAR
Componente da Banca

RONEI MARCOS DE MORAES, Dr., UFPB
Componente da Banca

FRANCISCO MARCOS DE ASSIS, Dr., UFCG
Componente da Banca (Ausência Justificada)

HERMAN MARTINS GOMES, Dr., UFCG
Componente da Banca

CAMPINA GRANDE – PB
OUTUBRO - 2004

Dedicatória

Às pessoas a quem estive mais próximo durante o desenvolvimento do trabalho:

Alexsandra e Carolina,
Érico, Aliane, Adriano, Davi e Eduardo

Agradecimentos

Primeiramente agradeço ao profressor João Marques por sua orientação e todo auxílio que proporcionou para o desenvolvimento desta tese.

Aos membros da banca pelas minuciosas revisões realizadas no texto, pela boa-vontade com que me receberam e me orientaram, pelas suas sugestões e material bibliográfico fornecido.

Aos colegas do Laboratório de Processamento de Imagens e Sinais que, sendo muitos, não cito os nomes para não ser enfadonho, bem como aos professores da PUC-PR ligados ao programa de cooperação acadêmica com o grupo de Análise e Classificação de Sinais e Imagens da UFCG, pela convivência e troca de experiências.

Ao professor Luis Brunelli e demais colegas do Departamento de Ciência da Computação e Estatística da Universidade Federal de Sergipe pelo estímulo e apoio prestado durante meu afastamento para cursar o doutorado.

Ao Governo Federal do Brasil, no papel da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo suporte financeiro e pela manutenção do Portal de Periódicos.

À funcionária Ângela pelo excelente trabalho na secretaria da COPELE, especialmente durante os períodos de greve.

Às pessoas a quem faço a dedicatória. À minha esposa e filha, por terem me acompanhado à cidade de Campina Grande. A Meu irmão Adriano pelas palavras de incentivo e interesse em acompanhar a evolução do trabalho. Aos meus pais e às pessoas que me apoiaram no retorno à cidade de Aracaju: meu tio Eduardo e meus sogros.

Resumo

O problema de classificação em reconhecimento de padrões pode ser interpretado como um problema de estimação de uma distribuição de probabilidade alvo. Trabalhos recentes apontam para sua modelagem como uma soma ponderada de distribuições, tratando-se portanto de uma abordagem paramétrica, já que pesos e parâmetros necessitam ser estimados. Neste trabalho a distribuição alvo é aproximada sem realizar estimação de parâmetros de uma distribuição modelo. Admitindo-se que a saída dos classificadores possam ser tratados como distribuições de probabilidades, utiliza-se uma rede Bayesiana como instrumento para realizar a combinação de classificadores locais e global. Em linhas gerais o objetivo do trabalho é apresentar uma metodologia que estabelece como realizar o particionamento do espaço de atributos originando um conjunto de classificadores e como agrupá-los em uma estrutura que combina suas saídas.

Um estudo de caso foi desenvolvido para avaliar o desempenho do sistema proposto no reconhecimento de imagens de dígitos manuscritos, tendo sido obtido resultados competitivos com os mais recentes mencionados na literatura.

Abstract

The classification problem in pattern recognition can be viewed as a probability distribution estimation task. Recent developments try to model it as a weight sum of distributions which is a parametric approach, since weights and parameters should be estimated. In this work the target distribution is reached without the need to estimate parameters from a model distribution. Considering that the output of classifiers are probability measurements, a Bayesian network is used to combine local and global classifiers. Briefly, the main objective of this work is to present a methodology that establishes how to partition the feature space in order to generate a set of classifiers and group them in a framework that combines their outputs.

A case study was developed for a handwritten digit recognition application. The results reveal that the proposed system is competitive with the best classifiers pointed in the literature.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

Um dos grandes desafios da ciência no início do século XXI é desenvolver máquinas que executem com habilidade tarefas que os seres humanos realizam corriqueiramente, tais como interpretar informações visuais e auditivas. As máquinas têm sido utilizadas com eficiência para processar grandes volumes de informação, como os computadores com sistemas de banco de dados, ou para processar em larga escala produtos manufaturados, como os robôs em aplicações industriais, mas o sucesso de sua aplicação em reconhecimento de padrões é ainda bastante incipiente se comparado com os seres humanos. Uma criança em idade de alfabetização é capaz de identificar letras e dígitos isolados em diferentes texturas e variações de formato e escala, como em rótulos comerciais e em livros infantis, melhor do que os mais sofisticados sistemas de reconhecimento de caracteres óticos disponíveis. Uma criança ainda muito cedo é capaz de atender a uma ligação telefônica, o que só é realizado automaticamente quando o ser-humano fornece para a máquina códigos digitados no teclado do aparelho. A eficiência dos computadores em aplicações tradicionais deve-se ao fato de que as taxas de processamento são muito elevadas e as condições de operação bastante uniformes, a localização de um registro em um banco de dados, por exemplo, pode ser feita rapidamente porque a freqüência dos processadores é elevada e porque a representação binária da chave de busca é bastante uniforme, de tal modo que uma variação no estado de um bit resulta em uma chave distinta. Em princípio, para o computador recuperar uma imagem em uma base de dados com a mesma rapidez e precisão seria necessário que as imagens comparadas fossem capturadas com as mesmas condições de luminosidade e posicionamento da câmera. Ao contrário dos computadores, os sistemas naturais de reconhecimento de padrões não são baseados na rígida arquitetura binária nem executam operações de forma puramente seqüencial, por esta razão, apesar das taxas de transferência de dados na mente humana serem muito menores, os seres humanos são capazes de reconhecer padrões com muito mais eficiência. Desenvolver sistemas computacionais que imitem a capacidade dos seres humanos de reconhecer padrões é portanto um grande desafio. Este é o objeto de estudo de segmentos específicos nos domínios da Inteligência Artificial (IA), Processamento de Imagens e Estatística e constitui uma

das principais motivações para realização desta tese.

Ao longo deste caítulo será apresentado em maiores detalhes o objeto de estudo da tese, apresentado na Seção 1.1, as contribuições do trabalho, Seção 1.2, e a organização do texto como um todo, Seção 1.3.

1.1 Objeto de estudo da tese

Em reconhecimento de padrões, o problema de classificação está relacionado com a construção de um modelo probabilístico que relaciona um conjunto de atributos, $\mathbb{R}^p \subset \mathbb{R}^n$ (o espaço de atributos), e um conjunto discreto de m possíveis classes, $\Omega = \{\omega_i\}_{i=1}^m$. Se forem conhecidas informações estatísticas completas sobre as distribuições de $\boldsymbol{x} \in \mathbb{R}^p$ para cada classe i, $i = 1 \ldots m$, isto é, $P(\boldsymbol{x}|\omega_i)$, então, a um padrão de teste $\boldsymbol{x}$ de classe desconhecida pode ser atribuída a categoria que maximiza a distribuição $P(\omega_i|\boldsymbol{x})$, obtida pela fórmula de Bayes, isto é:

$$\omega^* = \underset{\omega_i}{\operatorname{argmax}}\{P(\omega_i|\boldsymbol{x})\} \quad \boldsymbol{x} \in \mathbb{R}^p \tag{1.1}$$

com

$$P(\omega_i|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\omega_i)P(\omega_i)}{\sum_j P(\boldsymbol{x}|\omega_j)P(\omega_j)} \tag{1.2}$$

O valor de ω^* obtido pelas Equações (1.1) e (1.2) minimiza o erro de classificação, quando o custo associado a uma classificação errada é igual para todas as classes. Portanto ω^* é considerado um valor ótimo (Webb [115]). Na prática, entretanto, como as probabilidades $P(\omega_i)$ e $P(\boldsymbol{x}|\omega_i)$ não são conhecidas, o cálculo de ω^* não pode ser realizado exatamente. Uma grande variedade de soluções para realização do cálculo aproximado de (1.1) foram propostas desde finais dos anos 1950. Algumas referências realizam uma cobertura ampla, resumindo as diversas linhas nesta área, como em Jain *et al.* [60], Webb [115], Schalkoff [100] e Duda e Hart [33]. De um modo geral, a distribuição alvo é aproximada através de um processo de treinamento que consiste no aprendizado estatístico de $P(\cdot)$ a partir de um conjunto amostral.

O processo de treinamento realiza um particionamento do espaço de atributos em segmentos chamados regiões de decisão. Uma maneira de estabelecer estas regiões pode ser através do emprego de funções discriminantes, denotadas por $f(\boldsymbol{x};\boldsymbol{\theta})$ em que $\boldsymbol{\theta}$ corresponde a um conjunto de parâmetros aprendidos durante o treinamento. Este aprendizado consiste de fato em um processo de otimização em $\boldsymbol{\theta}$ de uma função objetivo que ajusta $f(\cdot)$ aos dados de treinamento. Algumas funções objetivo mencionadas na literatura são a minimização do erro médio quadrático, a maximização da função logarítmica de verossimilhança e a minimização da entropia cruzada (Webb [115]). Quanto ao tipo da função discriminante, pode-se ter funções lineares e não-lineares. Sendo $f(\cdot)$ linear, pode-se expressá-la como

$$f(x;\theta) = \sum x_i\theta_i + \theta_0 \tag{1.3}$$

ou resumidamente como $f(x;\theta) = x'^T\theta'^T$, sendo x' e θ' os vetores aumentados $x'^T =< x^T, 1 >^T$ e $\theta'^T =< \theta^T, \theta_0 >$. Alguns métodos para obtenção de θ são popularmente conhecidos, dentre eles pode-se citar: redes neurais Perceptron (Haykin [55]), função discriminante de Fischer (Cover e Hart [26]) e máquinas de vetores de suporte (Vapnik [112]). A fim de discretizar o resultado do produto interno, adaptando-o ao problema de classificação é comum aplicar sobre ele a função logística (*logistic function*) ou sigmóide, que tem o efeito de mapear a entrada para valores muito próximos de uma saída binária, conservando propriedades importantes como continuidade e diferenciabilidade. Particularmente, em se tratando de um problema de classificação binária, admitindo que x seja normalmente distribuído para as classes ω_1 e ω_2 e supondo que suas matrizes de covariância sejam iguais, a distribuição de ω_i dado x corresponde à aplicação da função logística sobre uma combinação linear de x para um conjunto de pesos ótimos $\hat{\theta}_i$, para cada classe ω_i (Jordan [64]). Embora esta seja uma propriedade de um caso particular, de um modo geral a função logística tem sido usada heuristicamente como ferramenta de discretização em sistemas de classificação. A contra-parte da função logística para o problema de classificação multinomial é a função *softmax* (Bridle [13], Bishop [7]), que assim como a função logística preserva propriedades de continuidade e diferenciabilidade aplicando-se a um problema de classificação com $m > 2$ classes. A expressão da função *softmax* é dada pela equação:

$$f_i(x;\theta) = f(x;\theta_i) = \frac{\exp(x^T\theta_i)}{\sum_k \exp(x^T\theta_i)} \tag{1.4}$$

Funções discriminantes não linear, obtidas pela introdução de um termo não linear em (1.3), podem ser expressadas pela forma geral

$$f(x;\theta,\mu) = \sum_i \theta_{ji}\phi(x;\mu_i) + \theta_{j0} \quad j = 1,\ldots,C \tag{1.5}$$

em que θ e μ são parâmetros de $f(\cdot)$ e $\phi(\cdot)$ é uma função não-linear de x. Existem diferentes tipos de funções discriminantes não-linear, associadas a diferentes métodos citados na literatura, tais como: redes neurais multicamada (*multilayer perceptron*) funções de bases radiais, máquinas de vetores de suporte não-lineares, dentre outros (Haykin [55], Webb [115]). De um modo geral estes métodos diferem entre si pelo tipo de função não-linear e pelo procedimento de busca empregados para obtenção dos parâmetros.

Uma outra categoria de métodos de reconhecimento de padrões procura estabelecer as regiões de decisão realizando um processo de divisão hierárquica do espaço de atributos. Árvores de decisão (*Classification and Regression Trees — CART*) (Breiman *et al.* [11]) e MARS (*Multivariate Adaptive Regression Spline*) (Friedman [39]) são exemplos destes métodos. Árvores de decisão

realizam uma divisão recursiva do espaço $\mathbb{R}^P$ em regiões disjuntas, que são associadas às classes dos padrões. O método MARS também realiza um particionamento recursivo de $\mathbb{R}^P$, mas ao invés de estabelecer fronteiras rígidas, a introdução de uma função de interpolação permite que ocorra um entrelaçamento entre estas regiões, suavizando as fronteiras de separação.

Uma abordagem natural para resolver o problema de classificação consiste em realizar a estimação da probabilidade a posteriori $P(\boldsymbol{x}|\omega)$ a partir de um conjunto de observações, admitindo-se por hipótese que $P(\cdot)$ seja regida por uma distribuição modelo, tipicamente assumida como sendo a distribuição Gaussiana ou Normal. Esta abordagem está localizada em uma categoria de métodos estatísticos conhecidos como métodos paramétricos pois, com base em uma amostra de $P(\cdot)$, realiza-se um treinamento estatístico objetivando-se estimar os parâmetros da distribuição modelo. A adoção desta linha, entretanto, baseia-se em uma hipótese bastante restritiva. Em uma aplicação de reconhecimento de imagens de caracteres óticos, por exemplo, a distribuição dos padrões associados à classe do dígito sete pode ter diferentes concentrações no espaço de atributos, relativas a grupos em que ocorre a escrita em estilo anglicano (imagens sem traço transversal) ou relativas a grupos que utilizam a escrita em estilo latino (imagens com traço transversal). Neste exemplo, $P(\boldsymbol{x}|\omega)$ possui mais de uma moda sendo, portanto, pouco apropriado que seja aproximada por uma distribuição Gaussiana.

A fim de minimizar as limitações impostas pela hipótese de normalidade, pode-se admitir que a distribuição alvo seja resultante de um modelo de mistura gaussiana (Jordan e Jacobs [65]) ou que a regra de decisão apresentada na Equação (1.1) possa ser realizada através de estimações não-paramétricas de $P(\omega|\boldsymbol{x})$. O método dos k vizinhos mais próximos (k-NN) é um exemplo conhecido de método não-paramétrico. Uma rede neural com função de ativação *softmax* (Bridle [13]) é também um método não-paramétrico, ambos realizam a estimação de $P(\omega|\boldsymbol{x})$ sem requerer a hipótese de que esta se ajuste a uma distribuição modelo.

Este trabalho apresenta uma forma de implementar a regra de decisão expressa na Equação (1.1) com base em um modelo de mistura, isto é, procura-se compor a distribuição alvo como uma soma ponderada de um conjunto de outras distribuições. Estas distribuições são, por sua vez, estimação não-paramétrica de $P(\omega|\boldsymbol{x})$ em regiões localmente definidas no espaço de atributos. Uma vez que $P(\omega|\boldsymbol{x})$ é de fato a saída de um classificador, o modelo de mistura proposto compara-se a um método de combinação de classificadores (Kittler *et al.* [69]).

A literatura aponta duas abordagens para o modo como classificadores podem ser combinados: abordagem estática e abordagem baseada em treinamento. Na abordagem estática as regras de combinação são fixas e portanto conhecidas a priori. As regras da soma, produto e voto majoritário (Kittler *et al.* [69]) são tipicamente regras estáticas pois independentemente dos valores originados pelos classificadores elas se aplicam da mesma forma. Neste caso o combinador não necessita ser treinado para aprender a regra de combinação. Na segunda abordagem o combinador é treinado com base nas saídas dos classificadores para encontrar uma regra de

combinação ótima. Se o espaço de saída dos classificadores fornece medições da probabilidade a posteriori de ω dado x então a construção de um combinador que implementa a regra de decisão apresentada na Equação (1.1), realiza efetivamente a composição de uma distribuição de probabilidade complexa e fatorada.

A proposta deste trabalho é apresentar um método de combinação de classificadores baseado em treinamento em que as saidas dos classificadores, tratadas como probabilidades, são sucessivamente refinadas e combinadas, culminando em uma unidade que resume as saídas produzidas em todas etapas anteriores, Figura 1.1. A regra de combinação é implementada em uma rede Bayesiana (Jensen [61]). A motivação em usar uma rede Bayesiana ocorreu em razão desta ser um método não-paramétrico que realiza eficientemente a estimação de uma distribuição de probabilidade complexa, um problema em geral de complexidade não-polinomial. Seu uso como método de classificação foi investigado em diversos trabalhos como em Friedman *et al.* [41], Sing e Provan [104], Ezawa e Schermann [35] e Frey [37]. Entretanto seu uso é mais destacado em aplicações nas quais relações de causa e efeito são bem caracterizadas, como aplicações médicas (Olesen *et al.* [66], Hamilton *et al.* [52]) e análise financeira (Abramson [1]). Do ponto-de-vista da comunidade de IA, uma rede Bayesiana é vista como um sistema especialista baseado em regras em que os valores verdade, quantificados numa escala contínua entre 0 e 1, são associados a probabilidades. Sob esta perspectiva o combinador proposto pode ser entendido como um sistema especialista, uma visão que simplifica o entendimento do método.

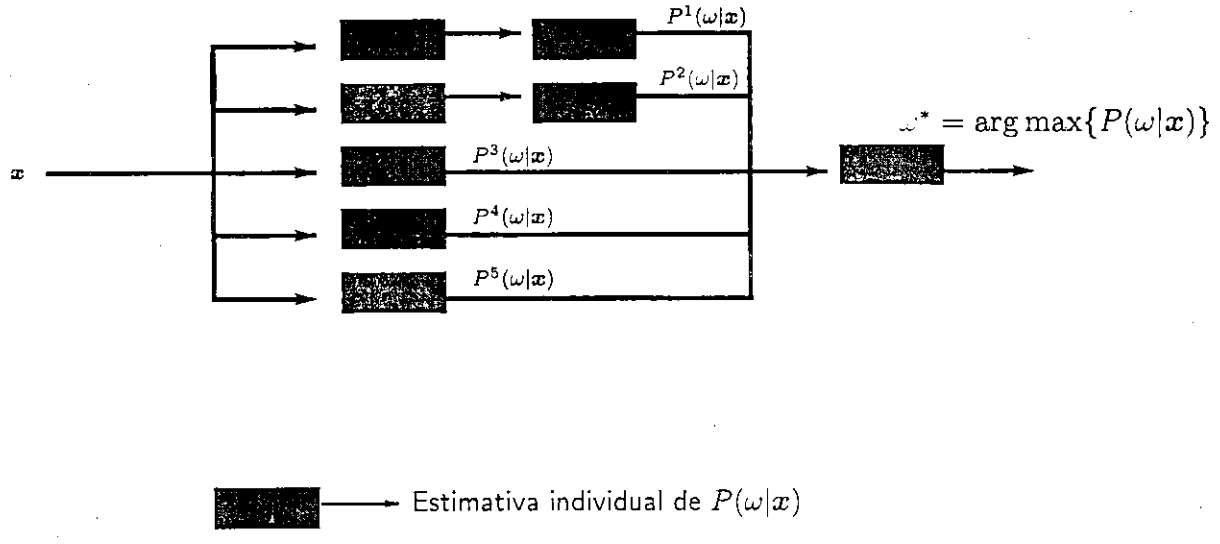


Figura 1.1: Diagrama de blocos do método proposto

1.2 Contribuições deste trabalho

- **Um método de combinação de classificadores construído de modo a destacar as superioridades individuais dos seus componentes** Redes neurais do tipo *Perceptron* multi-camada são aproximadores de funções que, quando concebidas com um número suficientemente grande de neurônios na camada escondida, podem aproximar ao grau de precisão desejado qualquer função (Li [78], Leshno *et al.* [77]). Quando treinadas com função de ativação *softmax* ou com um conjunto de treinamento suficientemente grande podem aproximar satisfatoriamente uma distribuição de probabilidade (Bridle [13], Gish [49], Ruck *et al.* [97]). Formam, portanto, uma solução bastante apropriada para o problema de classificação de padrões. Le Cun [75] aponta redes neurais como sendo o método de aprendizado baseado em gradiente mais bem sucedido para área de reconhecimento de padrões em espaços de grande dimensão, em particular para área de reconhecimento de manuscritos. Entretanto, estas redes também apresentam algumas limitações. Por um lado, o processo de treinamento é fortemente influenciado pela ocorrência de regiões planas e mínimos locais na superfície de erro. Além disto, a escolha do número de neurônios da camada escondida é crítica, se muito pequeno resulta em baixa capacidade de predição, se muito grande torna a rede instável. Métodos de aprendizado local (Friedman [40], Peng and Bhanu [89]) procuram minimizar estas limitações pelo aprendizado da função de predição em subconjuntos do espaço de atributos. A aproximação da função de predição em uma pequena vizinhança pode ser realizada por um polinômio de baixa ordem, pois os termos de mais alta ordem podem ser truncados em sua expansão por uma série de Taylor. Como conseqüência, em uma pequena vizinhança o erro de aproximação é reduzido. Exemplos de métodos de aprendizado local são o método dos k vizinhos mais próximos e outros baseados em instância (Aha *et al.* [3]) e regressão localmente ponderada (*locally weight regression*) (Atkeeson *et al.* [6]). Métodos de aprendizado local, por sua vez, apresentam desvantagens por requererem a realização de uma busca por k instâncias mais próximas de um padrão de teste. Este procedimento tanto demanda um grande esforço computacional, uma vez que em $\mathbb{R}^{n\geq 2}$ não existe nenhum procedimento de ordenação conveniente e rápido, quanto demanda grande espaço de armazenamento, pois é necessário manter uma memória de instâncias usadas no treinamento.

  Nesta tese é proposta uma estratégia original de geração e combinação de classificadores global e locais que reforça as superioridades individuais de cada um e minimiza suas limitações. A geração dos classificadores locais é orientada por um processo de particionamento do espaço de atributos que restringe o espaço em que os mesmos são definidos a uma vizinhança em $\mathbb{R}^P$, contribuindo desta forma para minimizar o esforço computacional envolvido com o procedimento de busca. As saídas dos classificadores, por sua vez, são combinadas

atendendo um critério de otimalidade, o que declina a contribuição de predições com erro e reforça a de predições acertadas.

- **Uso de redes Bayesianas em uma aplicação de combinação de classificadores** Apesar do emprego de redes Bayesianas, para classificação de padrões ter sido investigado em diversos trabalhos, seu uso como agrupador de classificadores é uma linha pouco explorada na literatura (Webb [115], pp 289 – 290). Esta tese explora a utilização de redes Bayesianas como agrupador de classificadores e propõe procedimentos de aprendizado da rede adaptados para o problema em foco. A utilização do sistema em um problema de reconhecimento de imagens de dígitos manuscritos produziu bons resultados numéricos, demonstrando a viabilidade de construir um sistema complexo a partir de classificadores simples, adequado para treinamento com grandes bases de dados.

- **Proposição de um sistema conexionista cuja arquitetura é ajustável em função do problema** O procedimento de particionamento do espaço de atributos define regiões que são associadas a classificadores locais. Para realizar uma predição única os classificadores conectam-se entre si formando a rede Bayesiana, cuja arquitetura depende da complexidade do problema, quando maior a dificuldade em aprender os dados de treinamento mais complexa torna-se sua arquitetura. Esta abordagem limita a interferência de um especialista humano em configurar parâmetros livres associados aos classificadores, tais como a especificação do número de neurônios e a quantidade de camadas escondidas em uma rede MLP, possibilitando a criação de um sistema complexo a partir de classificadores simples como redes *Perceptron* e classificadores locais k-NN.

- **Apresentação de um procedimento para extração de características de imagens de dígitos manuscritos isolados** No estudo de caso realizado no Capítulo 6 foi apresentado um algoritmo de extração de características para processamento de imagens de dígitos manuscritos inspirado no modelo biológico. O vetor de atributos é construído levando-se em conta a imagem do contorno, admitindo-se que estes pontos possuam toda informação necessária para a tarefa de classificação. A motivação de se trabalhar com a imagem do contorno deve-se ao fato de que o sistema visual humano é seletivo às componentes espaciais de alta freqüência (pontos de borda em uma imagem monocromática), que são fundamentais para reconhecimento de formas e do sentido de profundidade. Os resultados obtidos nos experimentos foram competitivos quando comparados aos publicados na literatura especializada, o que revela a boa capacidade do extrator de características em melhorar a separabilidade entre as classes.

1.3 Organização geral do trabalho

Para abordar os assuntos tratados, organizou-se o texto da seguinte forma: os Capítulos 2 e 3 realizam uma revisão de literatura versando, respectivamente, sobre redes Bayesianas e combinação de classificadores; o Capítulo 4 apresenta o método proposto em um nível de abstração mais elevado, abordando aspectos teóricos; o Capítulo 5 realiza uma abordagem prática, faz-se experimentos com diversas bases de padrões, compara-se os resultados obtidos com outros métodos e procura-se identificar pontos positivos e negativos da proposta; o Capítulo 6 realiza um estudo de caso em reconhecimento de imagens de dígitos manuscritos e o Capítulo 7 contém considerações gerais sobre a tese e apresenta linhas para desenvolvimentos futuros.

Capítulo 2

Redes Bayesianas

Este capítulo apresenta uma revisão teórica sobre redes Bayesianas. São discutidos os principais aspectos relacionados com o aprendizado e uso destas redes e apresenta-se o algoritmo de inferência usado posteriormente no método de combinação de classificadores proposto. O capítulo está organizado da seguinte forma: a Seção 2.1 apresenta definições e alguns conceitos básicos sobre teoria de probabilidades e grafos empregados ao longo das outras seções; a Seção 2.1.2 apresenta o algoritmo de inferência usado no método proposto no Capítulo 4 e discute os principais aspectos relacionados com o aprendizado destas redes; a Seção 2.4 discute a utilização de redes Bayesianas como classificadores revisando alguns dos principais trabalhos na área; a Seção 2.5 encerra o capítulo com conclusões gerais sobre o conteúdo apresentado.

2.1 Fundamentos

O formalismo da teoria de probabilidades tal como se conhece atualmente se deve à fundamentação axiomática de Kolmogorov (Kolmogorov [70]), baseada em teoria dos conjuntos. O trabalho de Kolmogorov, publicado originalmente em alemão no início do século XX, estendeu a noção de probabilidade baseada numa interpretação frequencista, permitindo associá-la a um número que mede incerteza. Na abordagem freqüencista, o conceito de probabilidade é interpretado como o caso limite de uma freqüência relativa associada a um número de observações infinitamente longo de um experimento aleatório. Na abordagem axiomática este conceito pode também ser relacionado com uma conjectura ou avaliação subjetiva. Neste caso a probabilidade expressa uma noção de chance que pode não ter nenhuma associação com um evento do qual se possa extrair uma freqüência relativa, por exemplo, a probabilidade de que Isaac Newton soubesse jogar xadrez. Esta segunda abordagem é estudada em Inteligência Artificial em áreas que procuram reproduzir o modo como os seres-humanos realizam raciocínio baseado em conhecimento incerto. Redes Bayesianas são um tipo de sistema especialista em que utiliza-se probabilidades para quantificar a certeza ou valores verdade das assertivas. O sistema manipula fatos novos e

o conhecimento armazenado em sua base apoiado na teoria de probabilidades. Rede Bayesiana é um tipo de sistema especialista que procura imitar o modo como os seres-humanos combinam conhecimentos novos e adquiridos para fazer novas conjecturas. A informação, expressa como probabilidade, é utilizada para atualizar o conhecimento existente através do Teorema de Bayes, apresentado abaixo.

Teorema 1 (Bayes) *Dados dois eventos E e F tais que $P(E) \neq 0$ e $P(F) \neq 0$, tem-se que:*

$$P(E|F) = \frac{P(F|E)P(E)}{P(F)} \tag{2.1}$$

O Teorema de Bayes é visto como um instrumento de atualização de conhecimento quando novos fatos são apresentados ao sistema. Por relacionar probabilidades *a priori*, $P(E)$, com probabilidades *a posteriori*, $P(E|F)$, este teorema pode ser usado para atualizar o conhecimento sobre um determinado domínio, representado pelo evento E, quando um novo conhecimento é aprendido, representado pelo evento F. O modo como a base de conhecimento é armazenada e como um conhecimento novo é usado para atualizar a base existente também se apoia em alguns conceitos de teoria dos grafos. Estes conceitos estão definidos na Seção 2.1.1, que realiza uma rápida cobertura de teoria dos grafos voltada especificamente para o tratamento de redes Bayesianas.

2.1.1 Definições

Esta seção apresenta algumas definições que serão usadas posteriormente ao longo do capítulo. Considere inicialmente a terminologia empregada para descrição de grafos. Um grafo não-direcionado $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ é formado por um conjunto finito e não vazio de vértices ou nós $\mathcal{V}$, representando variáveis de um domínio, e um conjunto de pares não-ordenados de vértices $\mathcal{E}$, chamados arcos. Um grafo direcionado é também formado por um conjunto de vértices e arcos, entretanto, os arcos constituem pares ordenados de vértices. Na representação gráfica de um grafo ordenado os vértices são representados como círculos e os arcos como setas ligando os círculos. A existência de um arco (A, B) exprime o sentido de adjacência entre os nós A e B, no diagrama deste grafo haverá uma seta com origem no nó A e terminação no nó B. O nó na origem do arco é chamado pai e aquele na terminação é chamado filho. A extensão destas relações originam outras mais abrangentes que definem os conceitos de ancestral e descendente. Dados dois nós distintos A e B, se A é pai de B ou se, recursivamente, A é pai do pai de B então A é dito ser ancestral de B e B descendente de A. Chama-se caminho entre dois nós A e B um sub conjunto de vértices adjacentes com origem em A e terminação em B. Um cíclo é

Definição 1 (DAG — Grafo Direcionado e Acíclico) *Um grafo direcionado e acíclico, chamado DAG (do inglês* Directed Acyclic Graph*), é um grafo direcionado que não contém ciclos.*

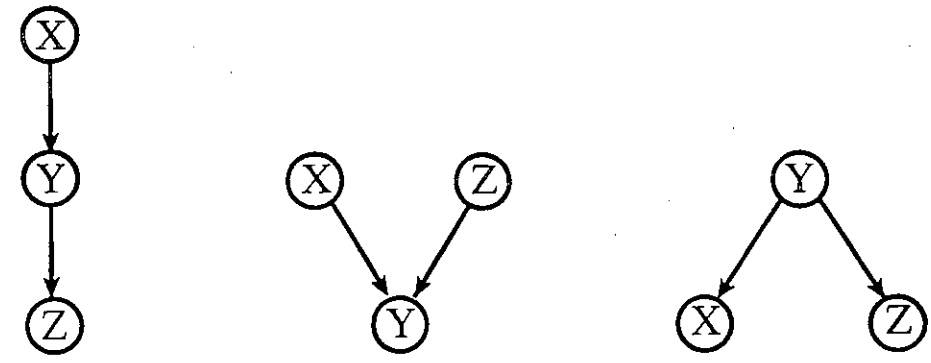
Definição 2 (Poliárvore (polytree)) *Chama-se poliárvore (do inglês* polytree*) um DAG unicamente conectado, isto é, um DAG que, desconsiderando o sentido dos arcos, não contém ciclos.*

Definição 3 (Condição de Markov) *Dado um grafo $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ no qual $\mathcal{V}$ associa-se a variáveis aleatórias de um dado domínio. $\mathcal{G}$ é dito satisfazer a condição de Markov se, para um nó X_i, chamando $\boldsymbol{ND_i}$ o conjunto de nós não-descendentes de X_i, tem-se que X_i é condicionalmente independente de $\boldsymbol{ND_i}$ dado o estado de seus pais, denotado por $\boldsymbol{Pa_i}$. Isto é*

$$P(X_i|\boldsymbol{Pa_i}, \boldsymbol{ND_i}) = P(X_i|\boldsymbol{Pa_i}) \tag{2.2}$$

A condição de Markov é utilizada para caracterizar o modelo gráfico de uma distribuição de probabilidade conjunta. Em Buntine [16] é realizada uma cobertura ampla sobre modelos gráficos dentre os quais se inclui redes Bayesianas e cadeias de Markov. Em se tratando de redes Bayesianas, relações de independência condicional podem ser estabelecidas considerando o conceito de d-separção. Este conceito, dentre outros, serão abordados a seguir. Considere deste ponto em diante que os nós dos grafos tratados sejam associados a variáveis aleatórias discretas.

Definição 4 (Evidência) *Dado um grafo $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, chama-se evidência o conjunto de nós instanciados, isto é, nós em que se conhece o estado das variáveis. Uma evidência é denotada por $\boldsymbol{E} \subset \mathcal{V}$, com $\boldsymbol{E} = \{X_i | X_i = x_{ik}\}$.*



(a) Linear (b) Convergente (c) Divergente

Figura 2.1: Caminhos possíveis entre nós X e Z

Definição 5 (caminho d-conectado) *Dada uma evidência, E, o caminho entre dois nós, X e Z, é dito ser d-conectado (conexão de dependência) em relação a E se uma das condições abaixo se verificarem*

1. *O caminho entre X e Z é linear ou divergente (Figuras 2.1(a) e 2.1(c)) e não possui nós em E.*

2. *O caminho entre X e Z é convergente (Figura 2.1(b)) e os nós no interior do caminho ou um de seus descendentes está contido em E.*

Definição 6 (nós d-separados) *Dada uma evidência, dois nós são ditos ser d-separados (separação de dependência) se não existir nenhum caminho d-conectado que os una.*

O conceito de separação permite identificar a existência de independência condicional entre variáveis aleatórias numa rede Bayesiana. Uma forma de independência se verifica quando num caminho linear um dos nós em seu interior está no conjunto de evidência. Esta situação, ilustrada na Figura 2.2, indica que independentemente do estado da variável X, dado que seja conhecido o estado do ascendente imediato de Y, a probabilidade de Y não é mais inflenciada por X e depende unicamente do valor de seu ascendente, assim X e Y são independentes dado E. Posto desta forma, pode-se apresentar a definição de independência em redes Bayesianas como apresentado em Pearl [88].

Definição 7 (Independência entre nós) *Dado uma evidência, E, diz-se que dois nós são independentes em relação a E se forem* d-separados.

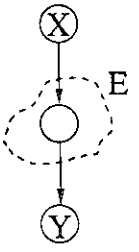


Figura 2.2: Indepência entre nós num caminho linear

2.1.2 Redes Bayesianas

Uma rede Bayesiana, denotada como B, é um modelo gráfico (Buntine [16]) para representação da probabilidade conjunta de um grupo de n variáveis aleatórias $U = \{X_1, \ldots, X_n\}$. Embora estas variáveis possam ser contínuas, este estudo aborda apenas o caso em que U contém somente variáveis aleatórias discretas, tendo em vista que no método proposto no Capítulo 4 a

rede utilizada contém somente nós discretos. O modelo é composto por duas partes: uma estrutura, representada por B_S, e um conjunto de probabilidades, denotado como B_P. A estrutura de uma rede Bayesiana consiste em um DAG no qual os nós representam variáveis aleatórias e os arcos relações de dependência condicional entre eles. O sentido dos arcos denotam relações de causalidade. Assim, se um arco possui origem em X_1 e terminação em X_2, admite-se que X_1 é causa de X_2 consequentemente X_2 é efeito de X_1. A outra parte que forma uma rede Byesiana, isto é, o conjunto de probabilidades, contém o conjunto de probabilidades *a priori* dos nós raízes e o conjunto de probabilidades condicionais dos eventos associados às ligações entre nós adjacentes.

Além das matrizes que formam o conjunto B_P, que são probabilidades que não são alteradas se o estado da rede muda, cada nó possui um vetor com as probabilidades *a posteriori*, que refletem o grau de certeza atribuído a cada estado, face os estados das demais variáveis. Este vetor é denominado vetor de crença, do inglês *belief*, definido como a seguir:

Definição 8 (Crença) *Chama-se crença, denotada por $BEL(\cdot)$, a probabilidade de um nó assumir um valor, dados os valores de todos os demais nós instanciados.*

$$BEL(x_i) = p(x_i|\boldsymbol{E}), \tag{2.3}$$

em que $\boldsymbol{E}$ é uma evidência.

Quando associada a um nó, e não a uma instância, a notação $BEL(\cdot)$ referencia o vetor de crenças do nó, isto é, $BEL(X_i) = (BEL(x_{i1}), BEL(x_{i2}), \cdots, BEL(x_{in}))$, em que $x_{i1}, x_{i2}, \cdots, x_{in}$ são os possíveis valores que o nó X_i pode assumir.

A ocorrência de uma evidência gera na rede um desequilibrio, que pode ser interpretado como a existência de inconsistências nos vetores de crenças de alguns nós. Para reintroduzir a rede em um novo estado de equilíbrio executa-se um algoritmo denominado algoritmo de inferência (Lauritzen *et al.* [74], Jensen [61], Pearl [87]). O mecanismo de funcionamento das redes Bayesianas pode, portanto, ser entendido como a execução do algoritmo de inferência em resposta à ocorrência de evidências. O problema de inferência em redes Bayesianas é em geral da classe *NP-HARD* (Cooper [22]), entretanto, quando o grafo que a compõe é uma poliárvore existem algoritmos de complexidade polinomial.

Os dois principais procedimentos envolvidos com redes Bayesianas são a realização de inferência e a obtenção da rede a partir de um conjunto de observações, procedimento denominado aprendizado. O aprendizado de uma rede Bayesiana é um problema largamente estudado na literatura (Buntine [15]). De um modo geral, este problema é colocado como sendo a obtenção de B_S e B_P a partir de um conjunto de observações ou casos, sendo que a maior dificuldade reside, de fato, em obter a estrutura da rede – B_S. Nas seções seguintes serão estudadas separadamente

estas duas partes. Na Seção 2.2 será abordado um algoritmo de inferência proposto por Pearl [87], na Seção 2.3 será discutido o problema de aprendizado em redes Bayesianas, isto é, como obter a rede a partir de um conjunto de casos, e na Seção 2.4 será comentada a aplicação de redes Bayesianas ao problema de classificação.

2.2 Inferência em redes Bayesianas

Esta seção descreve o algoritmo de inferência proposto por Pearl [87] e [88], que é utilizado no método proposto apresentado no Capítulo 4. Este algoritmo possui complexidade polinomial e se aplica a redes do tipo poliárvore. A idéia do método baseia-se em trocas de mensagens entre os nós, que ocorrem quando a rede recebe uma evidência. Como mencionado anteriormente, o propósito do algoritmo é reorganizar a rede em um novo estado no qual os vetores de crenças não contém inconsistência.

No método proposto em Pearl, o cômputo de $p(x_i|e)$ para um nó X, e uma evidência $\boldsymbol{E}$, deve ser expresso como uma função envolvendo separadamente as probabilidades de X assumir um valor x_i, dado o estado de seus ascendentes e a probabilidade de seus descendentes terem assumido os valores apresentados em $\boldsymbol{E}$ dado x_i. Este cálculo baseia-se numa partição do conjunto de evidências em relação ao nó X_i ilustrado na Figura 2.3. $\boldsymbol{E} = \boldsymbol{E}_{X_i} = \boldsymbol{E}^-_{X_i} \cup \boldsymbol{E}^+_{X_i}$, em que $\boldsymbol{E}^-_{X_i}$ compreende o conjunto de nós formado pela interseção entre $\boldsymbol{E}$ e o conjunto formado por X_i e todos os seus descendentes diretos e indiretos, e $\boldsymbol{E}^+_X$ a interseção entre $\boldsymbol{E}$ e o conjunto dos ascendentes de X_i. Obviamente, como $\boldsymbol{E}^-_{X_i}$ e $\boldsymbol{E}^+_{X_i}$ são mutuamente exclusivos tem-se que

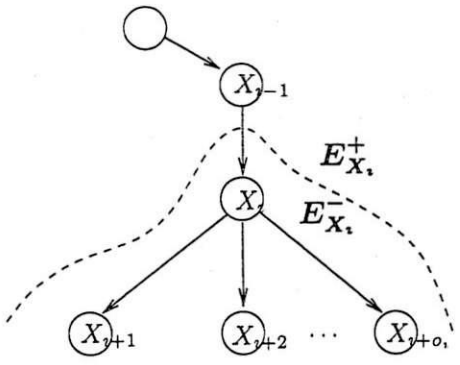


Figura 2.3: Partição de $\boldsymbol{E}$ em relação a X

$$BEL(x_i) = p(x_i|e_{X_i}) = p(x_i|e^+_{X_i}e^-_{X_i}) \tag{2.4}$$

Pearl mostra que (2.4) pode ser decomposto como

$$\begin{aligned} BEL(x_i) &= \alpha p(e^-_{X_i}|x_i)p(x_i|e^+_{X_i}) \\ &= \alpha\lambda(x_i)\pi(x_i) \end{aligned}$$

sendo α uma constante de normalização e

$$\lambda(x_i) = p(e^-_{X_i}|x_i) \tag{2.5}$$

$$\pi(x_i) = p(x_i|e^+_{X_i}) \tag{2.6}$$

O vetor de crença de X pode ser escrito com relação a λ e π como

$$BEL(X_i) = \alpha\lambda(X_i) \odot \pi(X_i)$$

Com a operação produto, denotado por $\odot$, representado a multiplicação componente a componente dos vetores λ e π definidos abaixo

$$\lambda(X_i) = (\lambda(x_{i1}), \lambda(x_{i2}), \cdots, \lambda(x_{ir_i}))$$
$$\pi(X_i) = (\pi(x_{i1}), \pi(x_{i2}), \cdots, \pi(x_{ir_i}))$$

sendo r_i o número de instâncias de X_i.

Cada nó, portanto, precisa manter um par de vetores auxiliares λ e π a fim de realizar a atualização de seu vetor de crenças. A atualização destes vetores auxiliares pode ser realizada através de mensagens provenientes de nós adjacente. Estas mensagens recebem denominações muito próximas aos nomes dos vetores auxiliares, o que pode trazer dificuldades para a compreensão do método. Vetores auxiliares e mensagens diferem na notação apenas pela presença de um índice subscrito, conforme comentado a seguir.

Considere um DAG, G, um nó X_i que possui o_i descendentes e p_i ascendentes, ou pais, e uma partição de G em relação a X_i. Sejam $G^{i+}_{X_{i-1}}, \ldots, G^{i+}_{X_{i-p_i}}$ partições de G formada por ascendentes de X_i, sejam $G^{i-}_{X_{i+1}}, \ldots, G^{i-}_{X_{i+o_i}}$ partições formadas por seus descendentes, como ilustrado na Figura 2.4, sejam $\boldsymbol{E}^{i+}_{\boldsymbol{X}_{i-1}}, \ldots, \boldsymbol{E}^{i+}_{\boldsymbol{X}_{i-p_i}}$ e $\boldsymbol{E}^{i-}_{\boldsymbol{X}_{i+1}}, \ldots, \boldsymbol{E}^{i-}_{\boldsymbol{X}_{i+o_i}}$ a parte da evidência, $\boldsymbol{E}_{\boldsymbol{X}_i}$ contida nos respectivos sub-grafos, os vetores auxiliares $\lambda(x_i)$ e $\pi(x_i)$ usados na atualização de $BEL(x_i)$ e as mensagens λ e π enviadas de e para X_i e um nó adjacente X_j, como definidos abaixo.

Vetor λ — $\lambda(x_i) = p(e^-_{\boldsymbol{X}_i}|x_i)$

Vetor π — $\pi(x_i) = p(x_i|e^+_{\boldsymbol{X}_i})$

Mensagem λ — $\lambda_i(x_j) = p(e^{i-}_{X_j}|x_i)$

Mensagem π — $\pi_i(x_j) = p(x_i|e^{i+}_{X_j})$

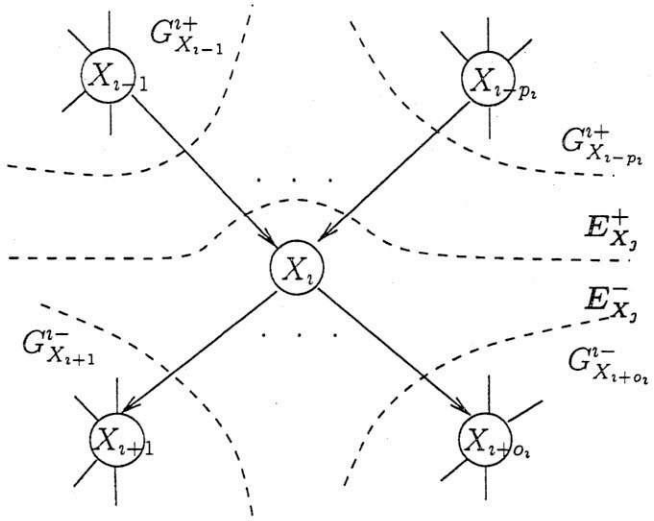


Figura 2.4: Particionamento de um DAG

O algoritmo proposto por Pearl é formado por duas etapas denominadas fusão e propagação. A fusão descreve o procedimento desempenhado por um nó para atualizar seu vetor de crenças ao receber mensagens provenientes de nós adjacentes. O procedimento de propagação descreve como um nó compõe uma mensagem a ser enviada para seus vizinhos após ter sido realizada uma revisão de probabilidades referente a uma evidência ocorrida na rede. Os procedimentos de fusão e propagação serão apresentados em detalhes a seguir.

Fusão Da Definição 8 tem-se que

$$BEL(x_i) = p(x_i|e)$$

Inicialmente separa-se $BEL(\cdot)$ em duas partes contendo as probabilidades envolvendo X_i e seus ascendentes e X_i e seus descendentes. Assim, tem-se que

$$\begin{aligned} BEL(x_i) &= p(x_i|e^{+}_{X_i} e^{-}_{X_i}) \\ &= p(x_i|e^{+}_{X_i} e^{i-}_{X_{i+1}} \ldots e^{i-}_{X_{i+o_i}}) \end{aligned}$$

Como $G^{+}_{X_{i-1}} \cup \ldots \cup G^{+}_{X_{i-p_i}}$ é d-separado de $G^{-}_{X_{i+1}} \cup \ldots \cup G^{-}_{X_{i+o_i}}$ dado X_i, então é válido que

$$
\begin{aligned}
BEL(x_i) &= \alpha \underbrace{p(e^-_{\mathbf{X}_{i+1}}|x_i)\ldots p(e^-_{\mathbf{X}_{i+o_i}}|x_i)}_{\lambda(x_i)} \underbrace{p(x_i|e^+_{\mathbf{X}_i})}_{\pi(x_i)} \\
&= \alpha\lambda(x_i)\pi(x_i) \qquad (2.7)
\end{aligned}
$$

Como $\lambda_j(x_i) = p(e^-_{\mathbf{X}_j}|x_i)$, $j \in \{i+1, i+2, \ldots, i+o_i\}$, corresponde à mensagem λ enviada por X_j a X_i, tem-se que

$$
\lambda(x_i) = \prod_{j=1}^{o_i} \lambda_j(x_i) \qquad (2.8)
$$

Expandindo o termo $p(x_i|e^+_{\mathbf{X}_i})$ tem-se que

$$
\begin{aligned}
p(x_i|e^+_{\mathbf{X}_i}) = \sum_{h_1=1}^{r_{i-1}} \cdots \sum_{h_{p_i}=1}^{r_{i-p_i}} & p(x_i|x_{i-1h_1}\ldots x_{i-p_ih_{p_i}}) \\
& p(x_{i-1h_1}\ldots x_{i-p_ih_{p_i}}|e^+_{\mathbf{X}_i})
\end{aligned}
$$

Como $x_{i-1}\ldots x_{i-p_i}$ são marginalmente independentes em relação a $e^+_{\mathbf{X}_i}$, além disso, como $p(x_{i-l}|e^+_{\mathbf{X}_i}) = p(x_{i-l}|e^{i+}_{\mathbf{X}_i})$ então

$$
\begin{aligned}
p(x_i|e^+_{\mathbf{X}_i}) = \sum_{h_1=1}^{r_{i-1}} \cdots \sum_{h_{p_i}=1}^{r_{i-p_i}} & p(x_i|x_{i-1h_1}\ldots x_{i-p_ih_{p_i}}) \\
& p(x_{i-1h_1}|e^{i+}_{\mathbf{X}_{i-1}})\ldots p(x_{i-p_ih_{p_i}}|e^{i+}_{\mathbf{X}_{i-p_ih_{p_i}}})
\end{aligned} \qquad (2.9)
$$

Uma vez que $p(x_{jk}|e^{i+}_{\mathbf{X}_j}) = \pi_i(x_{jk})$ é a mensagem π que o ascendente X_j envia para X_i, então, substituindo $\pi_i(x_{jk})$ em (2.9) tem-se que

$$
\pi(x_i) = \sum_{j_1=1}^{r_{i-1}} \cdots \sum_{j_{p_i}=1}^{r_{i-p_i}} p(x_i|x_{i-1j_1}\ldots x_{i-p_ij_{p_i}})\pi_i(x_{i-1j_1})\ldots\pi_i(x_{i-p_ij_{p_i}}) \qquad (2.10)
$$

$$
\begin{aligned}
&\Rightarrow BEL(x_i) = \\
&\alpha\left(\prod_{j=1}^{o_i}\lambda_j(x_i)\right)\left(\sum_{j_1=1}^{r_{i-1}}\cdots\sum_{j_{p_i}=1}^{r_{i-p_i}} p(x_i|x_{i-1j_1}\ldots x_{i-p_ij_{p_i}})\pi_i(x_{i-1j_1})\ldots\pi_i(x_{i-p_ij_{p_i}})\right)
\end{aligned} \qquad (2.11)
$$

Propagação As equações de propagação descrevem como um nó, X_i, elabora as mensagens λ e π enviadas aos seus ascendentes e descendentes, respectivamente, após ter realizado a atualização em seu vetor de crenças.

A mensagem $\lambda_i(x_{jk})$ enviada por X_i a um ascendente X_j referente ao seu k-ésimo estado é uma revisão da probabilidade de toda parte de $\boldsymbol{E}_{\boldsymbol{X}_i}$ que envolve X_i, o que inclui seus descendentes e ascendentes exceto, logicamente, X_j, dado x_{jk}. De acordo com a notação utilizada $\lambda_i(x_{jk})$ é dado por

$$\lambda_i(x_{jk}) = p(e^{j-}_{\boldsymbol{X}_j}|x_{jk})$$

Reescrevendo $\lambda_i(x_{jk})$ com base na partição de G em relação a X_i, tem-se que

$$\lambda_i(x_{jk}) = p((e^{+}_{\boldsymbol{X}_i} \backslash e^{i+}_{\boldsymbol{X}_j}) e^{-}_{\boldsymbol{X}_{i+1}} \dots e^{-}_{\boldsymbol{X}_{i-l+o_i}} x_i | x_{jk}) \tag{2.12}$$

em que $e^{+}_{\boldsymbol{X}_i} \backslash e^{i+}_{\boldsymbol{X}_j}$ corresponde à toda parte de $\boldsymbol{E}^{+}_{\boldsymbol{X}_i}$ excluindo X_j e seus ascendentes e descendentes exceto X_i.

Como nos desenvolvimentos anteriores, procura-se neste ponto tirar proveito das relações de independência que podem ser descobertas a partir do modelo gráfico. Sabe-se que X_i origina uma separação-d nas sub-redes formadas pelos seus descendentes o que origina a seguinte relação

$$p(e^{i-}_{\boldsymbol{X}_{i+1}} \dots e^{i-}_{\boldsymbol{X}_{i+o_i}} | x_i) = \prod_{l=1}^{o_i} p(e^{i-}_{\boldsymbol{X}_{i+l}} | x_i) \tag{2.13}$$

Por sua vez, cada um dos ascendentes diretos de X_i também origina separações-d, expressas como

$$p(e^{i+}_{\boldsymbol{X}_{i-1}} \dots e^{i+}_{\boldsymbol{X}_{i-p_i}} | x_{i-1} \dots x_{i-p_i}) = \prod_{l=1}^{p_i} p(e^{i+}_{\boldsymbol{X}_{i-l}} | x_{i-l}) \tag{2.14}$$

Então reescreve-se (2.12) como

$$\begin{aligned} \lambda_i(x_{jk}) = & \sum_{h_1=1}^{r_{i-1}} \cdots \sum_{h_{p_i}=1}^{r_{i-p_i}} \sum_{h=1}^{r_i} \\ & p(e^{+}_{\boldsymbol{X}_{i-1}} \dots e^{+}_{\boldsymbol{X}_{i-p_i}} e^{-}_{\boldsymbol{X}_{i+1}} \dots e^{-}_{\boldsymbol{X}_{i+o_i}} | x_{jk} x_{ih} x_{i-1h_1} \dots x_{i-1h_{p_i}}) \\ & p(x_{i-1h_1} \dots x_{i-p_i h_{p_i}} x_{ih} | x_{jk}) \quad \text{para } j \neq i-l \end{aligned} \tag{2.15}$$

Substituindo as relações expressas em (2.13) e (2.14) em (2.15) obtém-se

$$\begin{aligned} \lambda_i(x_{jk}) = & \sum_{h_1=1}^{r_{i-1}} \cdots \sum_{h_{p_i}=1}^{r_{i-p_i}} \sum_{h=1}^{r_i} \\ & p(e^{i-}_{\boldsymbol{X}_{i+1}} | x_{ih}) \dots p(e^{i-}_{\boldsymbol{X}_{i+o_i}} | x_{ih}) \\ & p(e^{i+}_{\boldsymbol{X}_{i-1}} | x_{i-1h_1}) \dots p(e^{i+}_{\boldsymbol{X}_{i-p_i}} | x_{i-p_i h_{p_i}}) \\ & p(x_{i-1h_1} \dots x_{i-p_i h_{p_i}} x_{ih} | x_{jk}) \quad \text{para } j \neq i-l \end{aligned} \tag{2.16}$$

Ocorre que $p(e^{i-}_{X_{i+l}}|x_{ih}) = \lambda_l(x_{ih})$ é a mensagem λ recebida por X_i de seu l-ésimo descendente. Além disto, pode-se reescrever $p(x_{i-1h_1} \dots x_{i-p_ih_{p_i}} x_{ih}|x_{jk})$ como

$$p(x_{ih}|x_{jk}x_{i-1h_1} \dots x_{i-p_ih_{p_i}})p(x_{i-1h_1} \dots x_{i-p_ih_{p_i}}|x_{jk}) \tag{2.17}$$

Mais uma vez, como as variáveis $X_j, X_{i-1}, \dots, X_{i-p_i}$ são marginalmente independentes, então

$$p(x_{i-1h_1} \dots x_{i-p_ih_{p_i}}|x_{jk}) = p(x_{i-1h_1}) \dots p(x_{i-p_ih_{p_i}}) \tag{2.18}$$

Substituindo (2.17) e (2.18) em (2.16) e reagrupando os termos tem-se que

$$\begin{aligned}\lambda(x_{jk}) = \sum_{h_1=1}^{r_{i-1}} \cdots \sum_{h_{p_i}=1}^{r_{i-p_i}} \sum_{h=1}^{r_i} & \\ & \lambda_{i+1}(x_{ih}) \dots \lambda_{i+o_i}(x_{ih}) \\ & p(e^{i+}_{X_{i-1}}|x_{i-1h_1})p(x_{i-1h_1}) \dots p(e^{i+}_{X_{i-p_i}}|x_{i-p_ih_{p_i}})p(x_{i-p_ih_{p_i}}) \\ & p(x_{ih}|x_{jk}x_{i-1h_1} \dots x_{i-p_ih_{p_i}}) \quad \text{para } j \neq i-l\end{aligned}$$

Usando o Teorema de Bayes, cada termo $p(e^{+}_{X_{i-l}}|x_{i-lh_l})p(x_{i-lh_l})$ pode ser rearranjado como

$$\alpha_l p(x_{i-lh_l}|e^{+}_{X_{i-l}})$$

em que α_l é uma constante. Finalmente, como $p(x_{i-lh_l}|e^{i+}_{X_{i-l}}) = \pi_i(x_{i-lh_l})$, tem-se que

$$p(e^{i+}_{X_{i-l}}|x_{i-lh_l})p(x_{i-lh_l}) = \alpha_l \pi_i(x_{i-lh_l})$$

$$\begin{aligned}\Rightarrow \lambda(x_{jk}) = \alpha \sum_{h_1=1}^{r_{i-1}} \cdots \sum_{h_{p_i}=1}^{r_{i-p_i}} \sum_{h=1}^{r_i} & \\ & \lambda_{i+1}(x_{ih}) \dots \lambda_{i+o_i}(x_{ih}) \\ & \pi_i(x_{i-1h_1}) \dots \pi_i(x_{i-p_ih_{p_i}}) \\ & p(x_{ih}|x_{jk}x_{i-1h_1} \dots x_{i-p_ih_{p_i}}) \quad \text{para } j \neq i-l\end{aligned} \tag{2.19}$$

em que α é uma constante de normalização e $p(x_{ih}|x_{jk}x_{i-1h_1} \dots x_{i-p_ih_{p_i}})$ é parte de B_P, portanto, conhecido.

É importante destacar que no cômputo de (2.15) a (2.19), obviamente, não está incluída a contribuição proveniente de X_j, assim o somatório se realiza sobre todos os ascendentes de X_i exceto aquele para o qual ele envia a mensagem λ.

A mensagem π que X_i envia a um de seus descendentes, X_j, dada por

$$\pi_j(x_i) = p(x_i|e^+_{\mathbf{X}_j})$$

equivale ao cálculo de $BEL(x_i)$ considerando que na evidência seja excluída a parte relativa a X_j e seus descendentes e ascendentes, exceto X_i. Logo, a partir de (2.11) tem-se que

$$\begin{aligned}\pi_j(x_i) &= p(x_i|e^+_{\mathbf{X}_j}) \\ &= \alpha\Big(\prod_{l\neq j}\lambda_l(x_i)\Big)\Big(\sum_{j_1=1}^{r_1-1}\cdots\sum_{j_{p_i}=1}^{r_i-p_i} p(x_i|x_{i-1j_1}\ldots x_{i-p_ij_{p_i}})\pi_i(x_{i-1j_1})\ldots\pi_i(x_{i-p_ij_{p_i}})\Big)\end{aligned} \tag{2.20}$$

2.3 Aprendizado de redes Bayesianas

Ao longo dos últimos 10 anos diversos autores propuseram e aperfeiçoaram métodos para resolver o problema de obter uma rede Bayesiana a partir de um conjunto de observações. Referências importantes nesta área podem ser obtidas em Buntine [15], que realiza uma completa revisão de literatura sobre o assunto até o ano de 1996, Heckerman [56], que descreve os fundamentos de vários métodos e assuntos correlatos e Krause [71] que realiza uma revisão abrangente e acessível procurando localizar-se entre as duas publicações anteriores, sendo mais profundo que Buntine [15] e menos técnico que Heckerman [56].

O maior problema ao obter uma rede Bayesiana a partir de dados é determinar sua estrutura, B_S, devido à enorme quantidade de redes que pode ser originada a partir de um pequeno conjunto de dados. Bouckaert [10] cita que a quantidade de redes, denotado como $G(\cdot)$, que pode ser formada por um conjunto de n nós é dada pela expressão

$$\begin{cases} G(0) &= 1, & \text{se } n = 0 \\ G(n) &= \sum_{i=1}^{n}(-1)^{i+1}\binom{n}{i}G(n-i), & \text{caso contrário} \end{cases}$$

Assim, para um conjunto com 10 nós existem aproximadamente $4,2\times10^{18}$ diferentes estruturas possíveis. A maioria das soluções propostas são baseadas em heurísticas que utilizam uma medida de adequação da estrutura aos dados (métrica) e em uma estratégia de busca (*search procedure*). Estes métodos também são denominados métodos de busca e pontuação (Carneiro [17]). Além destes, há também algoritmos que procuram obter a estrutura da rede através de testes de independência condicional que sucessivamente vão acrescentando arcos a uma rede inicialmente formada por nós não interligados. Estes são denominados métodos baseados em análise de dependência (Carneiro [17]).

Ao longo desta seção serão abordados os problemas relacionados ao aprendizado de B_P e B_S. Inicialmente será tratado o problema de aprendizado de B_P considerando os casos em que se dispõe de bases completa e incompleta. Em seguida será tratado o problema de aprendizado de B_S considerando a abordagem baseada busca e pontuação.

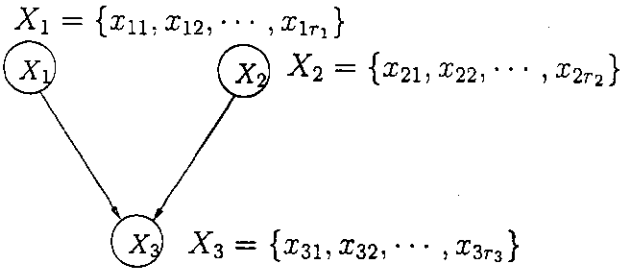


Figura 2.5: Configuração de um conjunto de nós usado para modelar aquisição de probabilidades

2.3.1 Aprendizado de probabilidades condicionais

O problema de aprendizado de probabilidades consiste em determinar para todas as instâncias de um nó, $X_i = \{x_{i1}, \cdots, x_{ir_i}\}$, a probabilidade condicional da ocorrência de x_{ir} dada a ocorrência da j-ésima instância dos pais de X_i, denotado por $\boldsymbol{Pa_i}$.

Considere inicialmente o desenvolvimento de uma solução baseado nos dados da Figura 2.5, que apresenta uma rede Bayesiana bastante simples formada por apenas três nós. Como X_1 e X_2 são nós raízes então o conjunto de pais de X_1 e X_2 são vazios. O problema de obter probabilidades condicionais se restringe portanto ao nó X_3, então, com base nos dados desta figura, o problema de aquisição de probabilidades pode ser colocado como encontrar os valores de $p(x_{31}|\boldsymbol{pa_{3j}}), p(x_{32}|\boldsymbol{pa_{3j}}), \ldots, p(x_{3r_3}|\boldsymbol{pa_{3j}})$, em que $\boldsymbol{pa_{3j}}$ pode ser qualquer uma das $r_1 r_2$ possíveis combinações de instâncias dos nós X_1 e X_2, isto é $\boldsymbol{pa_{3j}} \in \{x_{11}x_{21}, x_{11}x_{22}, \ldots, x_{1r_1}x_{2r_2}\}$.

Considere que a base de dados de observações das variáveis X_1, X_2 e X_3 seja formada por um conjunto de tuplas com os valores instanciados de cada uma destas três variáveis, isto é, $D = \{(x^1_{1k_{11}}, x^1_{2k_{12}}, x^1_{3k_{13}}), \ldots, (x^N_{hk_{11}}, x^N_{hk_{12}}, x^N_{hk_{13}})\}$. Cada uma destas tuplas é chamada caso e a base de dados é dita ser completa se em cada caso existirem observações sobre cada uma das n variáveis que compõem a rede.

Considere agora o caso geral em que o conjunto D é obtido a partir de uma rede com n variáveis. Uma vez que B_P é o elemento desconhecido e que se conhece B_S, suponha que a base de dados possa ser reagrupada conforme o arranjo seguinte:

$$
\begin{aligned}
D = \{\ & N_{111}(x_{11}, \boldsymbol{pa_{11}}), && N_{112}(x_{12}, \boldsymbol{pa_{11}}), && \ldots, && N_{11r_1}(x_{1r_1}, \boldsymbol{pa_{11}}), \\
& N_{121}(x_{11}, \boldsymbol{pa_{12}}), && N_{122}(x_{12}, \boldsymbol{pa_{12}}), && \ldots, && N_{11r_1}(x_{1r_1}, \boldsymbol{pa_{12}}), \\
& && \vdots && && \\
& N_{1q_11}(x_{11}, \boldsymbol{pa_{1q_1}}), && N_{1q_12}(x_{12}, \boldsymbol{pa_{1q_1}}), && \ldots, && N_{1q_1r_1}(x_{1r_1}, \boldsymbol{pa_{1q_1}}), \\
& && \vdots && && \\
& N_{n11}(x_{n1}, \boldsymbol{pa_{n1}}), && N_{n12}(x_{n2}, \boldsymbol{pa_{n1}}), && \ldots, && N_{n1r_n}(x_{nr_n}, \boldsymbol{pa_{n1}}), \\
& && \vdots && && \\
& N_{nq_n1}(x_{n1}, \boldsymbol{pa_{nq_n}}), && N_{nq_n2}(x_{n2}, \boldsymbol{pa_{nq_n}}), && \ldots, && N_{nq_nr_n}(x_{nr_n}, \boldsymbol{pa_{nq_n}})\}
\end{aligned}
\tag{2.21}
$$

com N_{ijk} denotando todas as observações da tupla $(x_{ik}, \boldsymbol{pa}_{ij})$. Utiliza-se o índice r_i para denotar a quantidade máxima de instâncias de um nó X_i. O índice q_i denota a quantidade máxima de instâncias do conjunto $\boldsymbol{Pa}_i$. Ao longo de toda esta seção as letras i j e k serão usadas com o seguinte sentido: a letra i denota o índice do nó, que pode variar de 1 até n, a letra j denota o índice dos pais de um nó e k, o índice de uma instância.

Base de casos completa

Considere as seguintes premissas:

I- A distribuição x_i dado $\boldsymbol{pa}_{ij}$ denotado por $p(x_{ik}|\boldsymbol{pa}_{ij})$ é multinomial com parâmetros $\boldsymbol{\theta}_{ij} = \{\theta_{ij1}, \ldots, \theta_{ijr_i}\}$.

II- A distribuição *a priori* de $\boldsymbol{\theta}_{ij}$ é Dirichlet, com parâmetros $\nu_{ij1}, \ldots, \nu_{ijr_i}$, denotada como $Dir(\theta_{ij1}, \ldots, \theta_{ijr_i}; \nu_{ij1}, \ldots, \nu_{ijr_i})$

III- Os dados amostrados são completos.

Naturalmente, a obtenção de θ_{ijk} consiste na estimação das proporções $N_{ijk}/\sum_{k=1}^{r_i} N_{ijk}$, que por hipótese possui distribuição conjunta *a priori* Dirichlet. Utilizando o método de inferência Bayesiana a estimação de θ_{ijk} é dada pelo valor esperado da distribuição *a posteriori* de θ_{ijk} dado as observações, isto é, $E_{p(\theta_{ij}|D)}[\theta_{ijk}]$. A distribuição *a posteriori* de θ_{ij} é também Dirichlet quando os dados possuem distribuição multinomial, por esta razão as distribuições Dirichlet e multinomial são chamadas complementares (Neapolitan [85]). A literatura (Heckerman [56], Neapolitan [85]) mostra que a distribuição *a posteriori* de θij é dada por

$$p(\boldsymbol{\theta}_{ij}|D) = Dir(\theta_{ij1}, \ldots, \theta_{ijr_i}; \nu_{ij1} + N_{ij1}, \ldots, \nu_{ijr_i} + N_{ijr_i}) \tag{2.22}$$

Segue-se a partir da Equação (2.22) que o valor esperado de θ_{ijk}, usado para estimar $p(x_{ik}|\boldsymbol{pa}_{ij})$ é dado por

$$p(x_{ik}|\boldsymbol{pa}_{ij}) = E_{p(\theta_{ij}|D)}[\theta_{ijk}] = \frac{\nu_{ijk} + N_{ijk}}{\nu_{ij} + N_{ij}} \tag{2.23}$$

em que $\nu_{ij} = \sum_{k=1}^{r_i - 1} \nu_{ijk}$ e $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

A equação (2.23) pode ser empregada satisfatoriamente no cálculo de $E_{p(\boldsymbol{\theta}_{ij}|D)}[\theta_{ijk}]$ se existir um conhecimento *a priori* sobre a distribuição de $\boldsymbol{\theta}_{ij}$. Foi assumido que $p(\boldsymbol{\theta}_{ij})$ é Dirichlet, mas nada foi dito sobre os valores dos parâmetros da distribuição. Sem o conhecimento *a priori* sobre estes parâmetros, Zabell [106] mostra que a equação (2.23) pode ser aproximada por

$$\theta_{ijk} = \frac{N_{ijk} + K}{N_{ij} + Kr_i} \tag{2.24}$$

em que N_{ijk} e N_{ij} são definidos como em (2.23) e K é uma constante, cujo valor sugerido é 1, $\frac{1}{2}$ ou $\frac{1}{r}$ (Herskovits [57]).

Base de casos incompleta

Na prática, muitas vezes é necessário tratar com bases de casos incompletas, isto é, bases em que, em algumas tuplas, há valores não observados de algumas variáveis. O tratamento de bases de dados incompletas tem sido largamente estudado na literatura dada a sua importância prática. Alguns trabalhos que resumem os principais desenvolvimentos nesta área devem-se a Singh [103] e Heckerman [56], que realizam uma abordagem considerando que os valores não preenchidos obedecam o Princípio da Informação Ausente (do inglês *Missing Information Principle*) (Ramoni e Sebastiani [91]). Este princípio estabelece que os dados não preenchidos tenham uma ocorrência aleatória na base de casos e que possam ser preenchidos artificialmente com base na informação presente.

Quando a ocorrência dos valores ausentes obedece ao Princípio da Informação Ausente, as soluções de um modo geral consistem em realizar o preenchimento dos dados omissos. Este preenchimento pode ser feito de diversas formas, como pela atribuição de um novo estado caracterizando um dado não observado, ou pelo emprego de um método estatístico. Devido à grande variedade de métodos estatísticos, há por conseguinte, muitas formas de estimar os dados não observados. As duas formas mais populares são pelo emprego do algoritmo EM (*Expectation and Maximization*) (Depster *et al.* [29]) e pela amostragem de Gibbs (Neal [84], Andrieu *et al.* [4]). Quando os dados ausentes possuem um viés, isto é, ocorrem de uma forma sistemátia, Ramoni e Sebastiani [91] mostraram que pode ser usado um algoritmo determinístico que, no caso médio, converge rapidamente para a solução exata. O algoritmo de Ramoni e Sebastiani, denominado *Bound and Collapse*, ao invés de realizar o preenchimento da base de casos, estabelece limites (*bounds*) para os intervalos em que os parâmetros da rede podem se localizar (no pior caso este intervalo é igual a $[0, 1]$), estes intervalos são iterativamente reduzidos (*collapse*) convergindo assintoticamente para o valor esperado do parâmetro estimado.

Nesta seção a idéia dos algoritmos de preenchimento baseados nos algoritmos EM e amostragem de Gibbs será explicada em maiores detalhes.

Algoritmo Expectation Maximization O algoritmo EM é comumente empregado para maximização da função de verossimilhança, que neste caso é a probabilidade condicional dos dados dado o vetor de parâmetros, θ, que pode ser denotada como

$$\max_{\theta} l(\theta|D) \quad \text{com } l(\theta|D) = p(D|\theta)$$

sendo D a amostra. Num modelo de mistura, uma distribuição $p(\boldsymbol{x})$ é tida como o resultado de uma mistura ou combinação linear de outras distribuições, podendo ser expressa como

$$p(\boldsymbol{x}) = \sum_i \pi_i p_i(\boldsymbol{x}|\theta_i)$$

com $\pi_i > 0$ sendo as proporções ou pesos da mistura, satisfazendo $\sum_i \pi_i = 1$ e $p_i(\boldsymbol{x};\theta)$ as densidades que compõem o modelo. No caso da distribuição $p(D|\boldsymbol{\theta})$, pode-se reescrevê-la como um modelo de mistura considerando uma partição do conjunto $D = \{D^{(o)}, D^{(e)}\}$, em que $D^{(o)}$ refere-se às variáveis observáveis e $D^{(e)}$, às variáveis escondidas ou não observadas. Assim, tem-se que

$$p(D|\boldsymbol{\theta}) = \sum_{D^{(e)}} p(D^{(o)}|D^{(e)}, \boldsymbol{\theta}) p(D^{(e)}|\boldsymbol{\theta}) \tag{2.25}$$

Com base na notação empregada na equação (2.25), a idéia do algoritmo EM pode ser apresentada como a seguir. Inicializa-se o vetor $\boldsymbol{\theta}$ e, com base nestes valores, realiza-se a estimação das variáveis escondidas, $D^{(e)}$, que corresponde à fase E (*expectation*) do algoritmo. A base completa resultante da fase E é então empregada para a maximização de $l(\boldsymbol{\theta}|D)$, fase M (*maximization*) do algoritmo. Os novos valores de $\boldsymbol{\theta}$ resultantes da fase M são por sua vez usados para iniciar uma nova fase E, dando início a um processo iterativo que finaliza quando um dado critério de convergência é atingido.

Na literatura é frequente a apresentação da fase E do algoritmo como uma etapa em que se obtém uma aproximação da verdadeira função logarítmica de verossimilhança, com base no valor corrente de $\boldsymbol{\theta}$. Normalmente emprega-se o símbolo $\boldsymbol{\theta}^{(t)}$ para denotar o valor de $\boldsymbol{\theta}$ na iteração t, e denota-se a aproximação de $l(\boldsymbol{\theta}|D)$ como $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$. Graficamente, o comportamento do algoritmo EM pode ser entendido com base na ilustração da Figura 2.6, em que se apresentam duas iterações que seguem após a atribuição do valor inicial de $\boldsymbol{\theta}$ — $\boldsymbol{\theta}^{(0)}$.

Para uma dada classe de problemas, como o problema de classificação com base em mistura Gaussiana, há estudos realizados que tornam simples a implementação do algoritmo EM (Webb [115], Ghahramani e Jordan [47]). Para o problema de estimação de parâmetros em redes Bayesianas, Lauritzen [73] mostrou que as etapas E e M podem ser realizadas como apresentadas a seguir:

Etapa E Dado que

$$l(\boldsymbol{\theta}|D) = \prod_{l=1}^{m} p(d_l|\boldsymbol{\theta}) \tag{2.26}$$

em que m é a quantidade de casos e d_l, a l-ésima tupla de D. Aplicando o operador $E[\cdot]$ sobre o logaritmo de (2.26), segue-se que

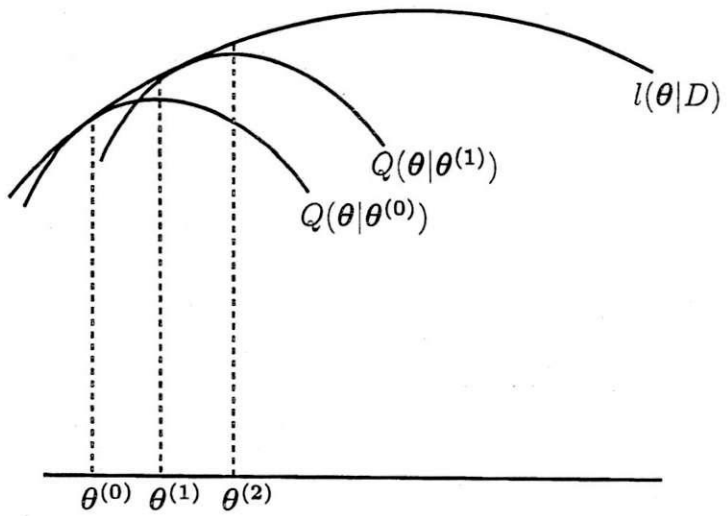


Figura 2.6: Evolução do algoritmo EM

$$\begin{aligned} l(\boldsymbol{\theta}|D) &= \prod_{i=1}^{n}\prod_{j=1}^{q_i}\prod_{k=1}^{r_i}\theta_{ijk}^{N_{ijk}} \\ \Rightarrow Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \sum\sum\sum E[N_{ijk}|D^{(o)},\boldsymbol{\theta}] \end{aligned}$$

Segundo Lauritzen, o valor de N_{ijk} pode ser estimado como

$$E[N_{ijk}|D^{(o)},\boldsymbol{\theta}] = \sum_{l=1}^{m} E_{\boldsymbol{\theta}}[\chi_{ijk}^{l}|d_l]$$

com $E_{\boldsymbol{\theta}}[\chi_{ijk}^{l}|d_l]$ dado por

$$E_{\boldsymbol{\theta}}[\chi_{ijk}^{l}|d_l] = \begin{cases} 1 \text{ se } X_i \text{ e } \boldsymbol{Pa}_i \text{ são observados e } X_i = x_{ik} \text{ e } \boldsymbol{Pa}_i = \boldsymbol{pa}_{ij} \\ 0 \text{ se } X_i \text{ e } \boldsymbol{Pa}_i \text{ são observados e } X_i \neq x_{ik} \text{ ou } \boldsymbol{Pa}_i \neq \boldsymbol{pa}_{ij} \\ p(x_{ijk}\boldsymbol{pa}_{ij}|d_l^{(o)},\boldsymbol{\theta},B_S) \qquad \text{caso contrário} \end{cases}$$

Podendo ser o valor $p(x_{ijk}\boldsymbol{pa}_{ij}|d_l^{(o)},\boldsymbol{\theta},B_S)$ calculado por algoritmo de inferência após a instanciação de $d_l^{(o)}$ e propagação das probabilidades na rede.

Etapa M Na etapa M, utiliza-se os valores de N_{ijk}, calculados na etapa E, fazendo

$$\theta_{ijk} = \frac{E[N_{ijk}|D^{(o)},\boldsymbol{\theta}]}{\sum_k E[N_{ijk}|D^{(o)},\boldsymbol{\theta}]}$$

Portanto o valor de θ_{ijk} é simplesmente aproximado pela proporção que N_{ijk} ocupa em N_{ij}.

O algoritmo EM aplicado ao problema de aprendizado de parâmetros em redes Bayesianas é, portanto, fácil de implementar. Segundo Heckerman [56], o algoritmo possui rápida convergência mas conduz a resultados sub-ótimos. Por outro lado a amostragem de Gibbs, que será detalhada a seguir, pode gerar resultados muito mais precisos, mas requer um número muito maior de iterações.

Amostragem de Gibbs A amostragem de Gibbs é um método de simulação de Monte Carlo baseado em cadeia de Markov (MCMC — do inglês *Markov chain Monte Carlo*). Simulação de Monte Carlo é um procedimento de amostragem para obtenção de elementos de uma determinada densidade alvo. O sentido da palavra amostragem utilizada aqui e empregada nos textos que tratam deste assunto, difere daquele normalmente empregado nos livros de estatística. Entende-se como amostrar um elemento de uma dada distribuição $p(\cdot)$ a realização de um experimento aleatório, simulado em computador, que produz como resultado um elemento $\boldsymbol{x}$ cuja distribuição de probabilidade é regida por $p(\cdot)$. A simulação de Monte Carlo pode ser usada para gerar seqüências de números aleatórios com distribuições bastante complexas que, ao contrário de distribuições conhecidas e bem exploradas como a normal, Cauchy ou uniforme, não podem ser geradas a partir de expressões conhecidas. Através deste procedimento é possível resolver problemas difíceis como o cálculo do volume de um corpo convexo em d dimensões em tempo polinomial (Andrieu *et al.* [4]). Simulação MCMC é também uma ferramenta conhecida e largamente utilizada no cômputo de problemas de física, economia e em inferência Bayesiana (Andrieu *et al.* [4], Webb [115]).

A amostragem de Gibbs, introduzida por Geman e Geman [46], é um método de simulação MCMC que gera amostras $\boldsymbol{x}$ de uma distribuição multivalorada caracterizando-se por gerar cada componente x_i sequencialmente, uma após a outra. Presume-se portanto que seja mais fácil simular a geração de $p(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ do que a geração de $\boldsymbol{x}$ a partir da distribuição conjunta de $p(x_i|x_1, \ldots, x_n)$.

Para usar a amostragem de Gibbs na obtenção dos parâmetros de uma rede Bayesiana baseada em casos incompletos deve-se satisfazer a restrição de que para cada vetor amostrado $\boldsymbol{x}$ tenha-se $p(\boldsymbol{x}) > 0$, denominado condição de irredutibilidade, e que $\boldsymbol{x}$ possa ser amostrado teoricamente em um número infinito de vezes, condição denominada aperiodicidade. Heckerman [56] e Singh [103] apresentam o algoritmo como:

1. Inicializa-se aleatoriamente os casos não observados de modo a ser obtida uma aproximação inicial de D completa — $D^{(0)}$.

2. Para cada valor omisso do conjunto D^e original – d_{il} (i-ésima posição da l-ésima tupla) – reavalia-se seu estado com base na distribuição $p(d_{il}|D \backslash d_{il}, B_S)$ até ser obtida uma nova base completa $D^{(t)}$.

3. Executa-se o passo 2 repetidas vezes e, ao final, realiza-se a estimação de $p(\boldsymbol{\theta}|D^o, B_S)$ pela média das probabilidades $p(d_{il}|D\backslash d_{il}, B_S)$ originadas no passo 2.

Se um número muito grande de iterações for empregado e se forem atendidas as restrições de que para cada instância $\boldsymbol{x}$, tenha-se $p(\boldsymbol{x}) > 0$ e que cada instância possa ser amostrada teoricamente num número infinito de vezes, a amostragem de Gibbs assegura uma estimação acurada do verdadeiro valor de $p(\boldsymbol{\theta}|D^o, B_S)$. Apesar de ser mais preciso do que o algoritmo EM, a amostragem de Gibbs costuma ser preterida em razão de convergir mais lentamente.

2.3.2 Aprendizado da estrutura da rede

Esta seção aborda métodos de aprendizado da estrutura da rede (B_S) baseados em busca e pontuação. São apresentadas as medidas de pontuação, ou métricas, mais comuns e comentados alguns algoritmos de busca.

métrica Bayesiana

O critério referenciado por alguns autores (Bouckaert [10], Tian [109]) como medida Bayesiana é usado para estabelecer a adequação de uma estrutura, B_S, na representação dos dados D. A métrica Bayesiana é derivada da expressão

$$p(B_S|D) = \frac{p(D|B_S)p(B_S)}{p(D)}$$

Assim, se B_S é melhor representativo que $B_{S'}$ deve-se ter que

$$\frac{p(B_S|D)}{p(B_{S'}|D)} = \frac{p(D|B_S)p(B_S)}{p(D|B_{S'})p(B_{S'})} > 1$$

logo, a solução, B_S para o problema de encontrar a estrutura da rede a partir dos dados pode ser expressa como

$$B_S^* = \arg\max_{B_S} = \{p(D|B_S)p(B_S)\}$$

ou

$$B_S^* = \arg\max_{B_S} = \{p(B_S, D)\}$$

Em seu projeto de doutorado Herskovits [57] desenvolveu uma expressão que fornece o valor de $p(B_S D)$ e um algoritmo, denominado $K2$ para maximizá-la. A métrica Bayesiana e o procedimento para maximizá-la serão explorados nos parágrafos seguintes.

Teorema 2 (Herskovits [57]) *Seja U um conjunto de nós $\{X_1, X_2, \ldots, X_n\}$, $n \geq 1$, em que cada X_i pode assumir valores em $\{x_{i1}, \ldots, x_{ir_i}\}$, $r_i \geq 1$, $i = 1, \ldots, n$. Seja D uma base de dados de casos em U e para cada variável X_i, seja $\boldsymbol{Pa_i}$ o conjunto de pais de X_i em B_S. Além*

disso para cada conjunto $\boldsymbol{Pa_i}$, seja $\boldsymbol{pa_i}$ sua j-ésima instância em relação a D, $j = 1, \ldots, q_i$, $q_i \geq 0$. Considere N_{ijk} o número de casos em D no qual a variável X_i possui o valor x_{ik} e $\boldsymbol{Pa_i}$ possui o valor $\boldsymbol{pa_{ij}}$. Assumindo as seguintes hipóteses:

1. *O processo que gera os dados é modelado como uma rede Bayesiana que contém somente variáveis em $\boldsymbol{U}$, que são discretas.*

2. *Os casos ocorrem independentemente, dado um modelo de rede.*

3. *A base de casos é completa.*

4. *As distribuições $f(\theta_{ij1}), \ldots, f(\theta_{ijr_i})$ e $f(\theta_{i'j'1}), \ldots, f(\theta_{i'j'r_{i'}})$, para $1 \leq i' \leq n, 1 \leq j \leq q_i, 1 \leq j' \leq q_{i'}$ são marginalmente independentes.*

5. *As probabilidades de segunda ordem são uniformes.*

Fazendo $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$, então

$$p(B_s, D) = p(B_s) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{2.27}$$

A expressão (2.27) fornece uma medida para comparar a adequação de uma estrutura aos dados. O algoritmo $K2$, proposto em Herskovits [57] descreve um procedimento para obtenção de $B_S^* = \arg\max_{B_S} = \{p(D|B_S)p(B_S)\}$ baseado na métrica enunciada no Teorema 2. A idéia deste algoritmo será apresentada posteriormente.

métrica MDL

Chama-se comprimento do descritor (do inglês *description length*) uma medida (em unidade de informação — bit) usada para descrever um conjunto de dados (Rissanen [94]). Se esta descrição puder ser realizada sem redundâncias obtem-se um comprimento do descritor mínimo (*Minimum Description Length*). Suponha por exemplo que em um grande conjunto de dados observa-se uma lei de formação. Então, existe uma máquina de Turing que modela tal conjunto. Logo, a descrição dos dados envolve a codificação de um algoritmo e a codificação de alguns elementos do conjunto, já que os demais poderiam ser deduzidos a partir da lei de formação. Em um caso extremo, suponha que em um conjunto de dados não teria sido identificado nenhum padrão de regularidade entre seus elementos, isto é, os dados poderiam ter sido gerados de forma completamente aleatória, neste exemplo sua descrição envolveria a de todos os elementos do conjunto. A medida de descrição de todos os elementos do conjunto é geralmente maior que a de uma lei de formação seguida de alguns elementos representativos, sendo assim deve ser evitada.

Seja $\mathcal{H} = \{H_1, H_2, \ldots\}$ um espaço de hipóteses, ou de leis de formação. Como não se conhece a lei de formação que originou os dados, D, a MDL é obtida por uma busca em $\mathcal{H}$. Mais precisamente busca-se obter de $\mathcal{H}$ o elemento mais provável dado o conjunto D, isto é, procura-se maximizar

$$p(H|D)$$

que pelo Teorema de Bayes pode ser expresso como

$$p(H|D) = \frac{p(DH)}{p(D)} = \frac{p(D|H)p(H)}{p(D)}$$

Aplicando o negativo do logaritmo em ambos os lados, tem-se

$$-\log p(H|D) = -\log p(D|H) - \log p(H) + \log(p(D)) \tag{2.28}$$

Como $p(D)$ é constante quando se varia H, então, para maximizar (2.28) deve-se minimizar

$$-\log p(D|H) - \log p(H) \tag{2.29}$$

Em Teoria da Informação, tomando o logaritmo na base 2, $-\log p(x)$ é o número de bits necessário para codificar a realização $X = x$ de uma variável aleatória X. O códibo de Huffman (Lelewer e Hirschberg [76]) aproxima esse valor assintoticamente quando se considera n observações da variável aleatória X. Baseado na equação (2.29), a definição da MDL como descrito em Vitányi e Li [113] é dada por

Definição 9 (Descritor de comprimento mínimo (MDL)) *Dado um conjunto de dados, e uma enumeração efetiva de modelos, a MDL ideal seleciona o modelo que minimiza as seguinte somas:*

1- do tamanho, em bits, da descrição efetiva do modelo; e

2- do tamanho, em bits, da descrição dos dados quando codificados com a ajuda do modelo.

Intuitivamente, se H for muito simples poderia ser descrita ou codificada mais simplesmente, mas não iria explicar satisfatoriamente os dados, o que acarretaria em maior erro na descrição dos mesmos. Sendo, por outro lado, H complexa os dados seriam explicados mais precisamente mas seria necessário maior espaço em sua codificação. A MDL, como será explicado a seguir, é uma solução de compromisso que busca encontrar um balanço entre estas duas partes.

A idéia empregada para a descrição dos dados e modelo é baseada na codificação de Huffman, um código cujo valor esperado do tamanho é mínimo (Cover e Thomas [27]). Ocorre que a quantidade de casos requeridos é uma função exponencial do número de nós e do número de

instâncias de cada nó (Lam e Bachus [72]). Esta limitação faz com que o cálculo ideal da MDL não seja realizável, obrigando a serem adotadas soluções heurísticas. Algumas destas heurísticas foram propostas por Lam e Bachus [72] e Bouckaert [10], sendo a segunda mais utilizada em trabalhos subsequentes, como Suzuki [107], Tiam [109] e Friedman e Getor [42].

A solução heurística proposta por Bouckaert [10] se fundamenta no Teorema 3

Teorema 3 (Boukaert [10]) [1] *Seja U um conjunto de variáveis aleatórias $U = \{X_1, X_2, \ldots, X_n\}$. Seja B_S a estrutura de uma rede Bayesiana e D uma base de dados completa com N casos. Seja $p(B_S, D)$ a medida Bayesiana de B_S e D e seja $L(B_S, D)$ uma medida de descrição de B_S e D. Então*

$$L(B_S, D) = -\log p(B_S, D) + C$$

em que C é uma constante que não depende de N, sendo L dado pela expressão

$$L(B_S, D) = -\log p(B_S) + NH(D, B_S) + \frac{\log N}{2} K \tag{2.30}$$

com

$$H(D, B_S) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} -\frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N_{ij}}$$

$$K = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \Big(\prod_{X_j \in \boldsymbol{Pa}_i} r_j \Big)(r_i - 1)$$

O termo $H(D, B_S)$ corresponde à entropia condicional dos dados em relação à rede e o termo K é uma constante em relação a N. K diz respeito à complexidade da rede, quanto maior for o número de conexões entre os nós, maior será o seu valor. Pelo gráfico da Figura 2.7, pode-se perceber que a MDL estimada por (2.30) procura selecionar uma solução de compromisso entre os termos $NH(B_S, D)$ e $\frac{\log N}{2} K$, considerando $-\log p(B_S)$ constante, ou seja, considerando que todas as redes tenham igual probabilidade de serem geradas. Na realidade procura-se redes mais simples, mas não tão simples que não expliquem satisfatoriamente o comportamento dos dados.

Algoritmos de busca

Como mencionado no início da Seção 2.3, o número de estruturas possíveis aumenta exponencialmente com a quantidade de nós, portanto, a avaliação de cada elemento do conjunto de

[1] No artigo original Boukaert apresenta a medida de descrição, $L(B_S, D)$, como sendo $L(B_S, D) = \log p(B_S, D)$, que é uma medida negativa. Neste texto $L(B_S, D)$ difere daquela apresentada em Boukaert por um sinal, tornando L uma medida positiva

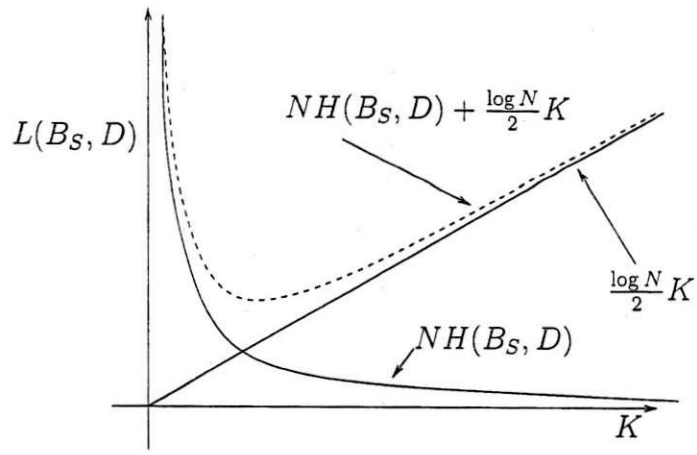


Figura 2.7: *Description Length* em função da complexidade da rede

estruturas individualmente com base em uma métrica estabelecida requer um tempo exponencial. Herskovits [57] apresentou uma heurística gulosa, isto é, um método de busca que emprega um critério de otimização local, que fornece uma solução aproximada em tempo polinomial, como descrito a seguir.

A fim de simplificar a dimensão do espaço de busca, estabelece-se uma ordenação nos nós oriunda de conhecimento de especialistas. Esta ordenação assegura que, dados dois nós X_i e X_j, se $i < j$, então X_i pode ser pai de X_j mas X_j não pode ser pai de X_i. Considere o conjunto de nós $\boldsymbol{U} = \{X_1, X_2, X_3\}$ dispostos segundo esta ordenação. Neste exemplo os possíveis conjuntos $\boldsymbol{Pa_i}$ seriam

$$\begin{array}{lll} \boldsymbol{Pa_3} = \{X_1, X_2\} & \boldsymbol{Pa_2} = \{X_1\} & \boldsymbol{Pa_1} = \{\} \\ \boldsymbol{Pa_3} = \{X_1\} & \boldsymbol{Pa_2} = \{\} & \\ \boldsymbol{Pa_3} = \{X_2\} & & \\ \boldsymbol{Pa_3} = \{\} & & \end{array}$$

Baseado nos dados deste exemplo, pode-se perceber que se fosse possível realizar para cada nó, X_i, a busca do conjunto $\boldsymbol{Pa_i}$ que seja ótimo em relação à métrica estabelecida seria necessário realizar um número exponencial de avaliações de estruturas, de fato, a obtenção de B_S^* requereria $\sum_i \binom{n}{i} = 2^n - 1$ avaliações para um conjunto ordenado de nós.

O algoritmo K2, proposto por Herskovits, obtém B_S^* em tempo polinomial fazendo com que para cada nó, X_i, um dos possíveis predecessores, X_j, é acrescido ao conjunto $\boldsymbol{Pa_i}$ se $\{X_j\} \cup \boldsymbol{Pa_i}$ maximiza métrica Bayesiana.

Em Bouckaert [10] é proposto um algoritmo, denominado K3, em que se emprega a métrica MDL em substituição à medida Bayesiana no algoritmo K2.

Baseado no algoritmo K3, Suzuki [107] desenvolveu um método de busca em largura utilizando (*Branch and Bound*) que assegura a obtenção de uma solução exata para (2.30). Tian [109]

melhorou o algoritmo de Suzuki utilizando limites (*Bounds*) mais precisos. Uma cobertura ampla dos algoritmos de busca foge à proposta deste trabalho, em Carneiro [17] pode-se encontrar uma revisão extensa de diversos algoritmos nesta área.

2.4 Redes Bayesianas como classificadores

A aplicação mais comum de redes Bayesianas está focada em problemas em que relações de causa e efeito são bem definidas, já que o sentido dos arcos no grafo subjacente denotam relações de causalidade. Por esta razão, embora redes Bayesianas sejam uma ferramenta apropriada para estimação da densidade conjunta de um grupo de variáveis aleatórias, a estimação da classe de um padrão dado o estado de seus atributos, tende a produzir resultados insatisfatórios (Friedman *et al.* [41]). Esta seção apresenta soluções a este problema referenciadas na literatura bem como o classificador Bayesiano (Duda e Hart [33]) que, mesmo não sendo propriamente uma rede Bayesiana adaptada ao problema de classificação (já que sua origem antecede ao das redes Bayesianas) é considerado uma rede Bayesiana simplificada (*Naive Bayesian Network*).

2.4.1 Classificador Bayesiano (Naive Bayesian Network) (Duda e Hart [33])

Considere um conjunto de n variáveis aleatórias distintas associadas a medições sobre um determinado domínio e uma variável de decisão relacionada à classe a qual estes atributos se associam. Sejam $X_1, \ldots, X_n$ as variáveis aleatórias associadas aos atributos e Y a variável aleatória associada à classe ou rótulo. Supondo, por hipótese, que as distribuições de X_i dado Y sejam independentes e conhecidas, usando a regra de Bayes pode-se obter uma expressão para fornecer a estimação da distribuição de Y dado $X_1, \ldots, X_n$. Uma vez que se pode mensurar $p(y_i|\boldsymbol{x})$, atribui-se a um padrão de teste o rótulo da classe que maximiza esta distribuição, considerando uma variação sobre os valores de Y. O classificador assim definido é chamado classificador Bayesiano. Naturalmente, há uma associação direta entre este classificador e a rede Bayesiana ilustrada na Figura 2.8, por isto alguns autores o chamam de rede Bayesiana simplificada (*Naive Bayesian Network*).

Na prática, apesar de ter sido proposto há várias décadas e de ser baseado em uma hipótese muito restritiva, pois nem sempre é válido que ocorrências dos atributos sejam independentes dado a classe, o classificador Bayesiano é um método de classificação competitivo comparado aos recentes avanços na área

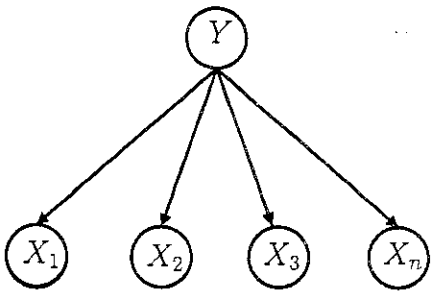


Figura 2.8: Classificador Bayesiano (*Naive Bayesian Network*)

2.4.2 Classificadores Bayesianos seletivos (Singh e Provan [104])

Singh e Provan [104] propõem um algoritmo composto por duas fases. Na primeira realiza-se uma seleção de características, da qual se extrai do conjunto de atributos aqueles mais representativos. Na segunda, utiliza-se uma variação do algoritmo K2, denominada CB (Singh e Valtorta [105]), para criar uma rede Bayesiana com base nos casos observados. O algoritmo CB não requer o fornecimento da ordenação dos nós por um especialista, esta ordenação é realizada com base em testes de independência condicional e, uma vez conhecida tal ordenação, é aplicado o algoritmo K2. No processo de seleção de características os autores propõem uma estratégia gulosa baseada, também, no algoritmo CB. Partindo de uma rede que possui apenas um nó, aquele associado à variável de decisão — Y, adicionam-se sucessivamente novos atributos enquanto a medida de adequação dos dados à rede (métrica Bayesiana) estiver aumentando, algoritmo denominado K2-AS [2], ou não estiver diminuíndo, algoritmo denominado K2-AS<. A rede Bayesiana obtida pelos algoritmos K2-AS e K2-AS< são esperadamente mais complexas que o classificador Bayesiano. Além disto, não impõem restrições quando à independência estatística dos atributos. Em experimentos realizados observou-se que as redes K2-AS e K2-AS< apresentam melhor desempenho que o classificador Bayesiano, especialmente quando os classificadores são treinados com bases de casos mais numerosas.

2.4.3 Classificadores Bayesianos explorados por Friedman et al. [41]

Friedman *et al.* [41] também realizaram uma extensão do classificador Bayesiano, relaxando a hipótese de independência dos atributos dado o valor da classe. O trabalho de Friedman *et al.* [41], assim como o de Singh e Provin [104], procura estabelecer regras de criação de redes Bayesianas adaptadas ao problema de classificação. Naturalmente, redes Bayesianas com estrutura mais complexa que o classificador Bayesiano eliminam, ou minimizam, o problema imposto pela hipótese de independência dos atributos dado a classe. Por outro lado, redes obtidas a partir dos dados que buscam unicamente otimizar a função de escore (métrica Bayesiana

[2] O nome do algoritmo significa K2 com seleção de atributos, do inglês K2 *with Atribute Selection*

ou MDL) tendem a desempenhar de maneira insatisfatória quando aplicada ao problema de classificação. Friedman *et al.* [41] propõe uma modificação da métrica MDL pela incorporação de um termo que penaliza a geração de redes que não atendem à maximização da probabilidade condicional de ω_i dado $\boldsymbol{x}$. A função de escore apresentada em seu trabalho é dada por

$$\sum_{i=1}^{N} \log p(\omega^i|\boldsymbol{x}^i) + \sum_{i=1}^{N} \log p(\boldsymbol{x}^i) \tag{2.31}$$

Friedman *et al.* apresentam um procedimento de complexidade $O(n^2N)$ que minimiza (2.31) construindo um tipo de rede Bayesiana denominada TAN (*Tree Augmented Naive Bayes*). Os autores também utilizaram um procedimento para obtenção dos parâmetros da rede (B_P) diferente daquele apresentado na Seção 2.3, o qual referenciaram como método de suavização dos parâmetros. Experimentalmente eles constataram que redes TAN com suavização de parâmetros de um modo geral têm um desempenho semelhante às redes TAN que não empregam o procedimento de suavização e, em alguns casos, apresentaram desempenho superior. Também foi constatado experimentalmente que o erro obtido com redes TAN, com e sem suavização de parâmetros, na maioria das vezes é menor que aquele obtido com o classificador Bayesiano.

Friedman *et al.* também investigaram outra categoria de redes Bayesianas aplicadas ao problema de classificação. Trata-se de uma combinação de redes Bayesianas denominadas *Bayesian mutinets* (Geiger e Heckerman [44]). Nesta abordagem, a base de casos é dividida em n partições disjuntas, cada uma associada a uma classe ω_i. Para cada um destes subconjuntos constrói-se uma TAN segundo o procedimento proposto e usado no caso anterior. Como cada uma destas redes fornece uma aproximação da distribuição $p(\boldsymbol{x}|\omega_i)$, aproximando $p(\omega_i)$ pela sua freqüência relativa, pode-se obter uma expressão que fornece a distribuição $p(\omega_i|\boldsymbol{x})$, usada efetivamente como critério de decisão para classificar um padrão de teste. Em experimentos realizados os dois métodos (TAN e *Bayesian multinets*) mostraram-se equivalentes.

2.4.4 Classificador Bayesiano explorado por Frey [37]

Frey [37] desenvolve um estudo bastante amplo de aplicações de redes Bayesianas, não se detendo apenas a seu uso como classificador, ele também a explora como ferramenta para compressão de dados e codificação de canais.[3] A linha de pesquisa investigada em seu trabalho difere bastante daquela adotada nos trabalhos revisados nas Seções 2.4.2 e 2.4.3, Frey estuda modelos de redes mais complexos, com muitos laços, e que, portanto, não podem ser utilizados por algoritmos de propagação de probabilidades nem obtidos com base nos algoritmos estudados na Seção 2.3. Os classificadores abordados neste trabalho exploram fortemente a propriedade da rede Bayesiana em modelar uma distribuição conjunta, sendo assim, ao invés de tomar a de-

[3] do inglês, *channel coding*.

cisão com base na avaliação de $p(\omega_i|\boldsymbol{x})$, a tomada de decisão é feita com base na avaliação da distribuição conjunta de $\boldsymbol{x}$ para um grupo de n redes Bayesianas, sendo n o total de classes do problema. Chamando $p_i(\boldsymbol{x})$ a densidade de $\boldsymbol{x}$ calculada pela i-ésima rede, classifica-se um padrão de teste atribuindo-lhe o índice da rede que maximiza $p_i(\boldsymbol{x})$. Neste caso, o tipo de procedimento adotado está mais próximo daquele utilizado pelo classificador modelos escondidos de Markov (Rabiner and Schafer [90]).

Os tipos de redes estudados estão classificados em dois grupos denominados redes autoregressivas e redes de múltiplas-causas, que serão revisados nas seções seguintes.

Redes autoregressivas

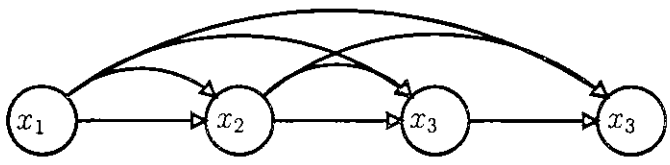


Figura 2.9: Redes Bayesianas autoregressivas

Redes autoregressivas são na realidade uma cadeia de nós totalmente conectados no sentido esquerda/direita, como ilustrado na Figura 2.9. Para estas redes a probabilidade da ocorrência conjunta de $\boldsymbol{x}$ é dada por

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \prod_i p(x_i|x_1, \ldots, x_{i-1}, \boldsymbol{\theta}) \tag{2.32}$$

sendo $\boldsymbol{\theta}$ um grupo de parâmetros da rede.

Se as variáveis, por sua vez, forem binárias, tem-se um modelo de regressão logístico (McCullagh e Nelder [82]), podendo $p(\cdot)$ ser aproximado pela aplicação da função logística sobre uma projeção de $\boldsymbol{x}$, como colocado abaixo

$$p(x_i|x_1, \ldots, x_{i-1}, \boldsymbol{\theta}) = x_i f\left(\sum_{k=0}^{i-1} \theta_{ik} x_k\right) + (1 - x_{i-1})\left(1 - f\left(\sum_{k=0}^{i-1} \theta_{ik} x_k\right)\right) \tag{2.33}$$

com $f(x) = 1/(1 + e^{-x})$ e $x_0 = 1$.

Frey [37] apresentou um procedimento de busca para obter os valores de $\boldsymbol{\theta}$ com base na aplicação do método do gradiente conjugado à maximização da função logarítmica de verossimilhança. Em experimentos realizados com reconhecimento de caracteres manuscritos Frey estabeleceu o sentido de ordenação dos nós na cadeia com o mesmo sentido da varredura empregada na leitura de documentos, isto é, esquerda para direita e cima para baixo. Os resultados destes experimentos foram extremamente promissores quando comparado a outros métodos tradicionalmente

empregados nestas comparações, como o classificador Bayesiano, o método dos k vizinhos mais próximos e árvores de decisão.

Redes de múltiplas causas

Nas redes de múltiplas causas, presume-se que o estado das variáveis observadas — x — seja de fato conseqüência da influência direta de um conjunto de variáveis não-observadas ou escondidas — h — como ilustrado na Figura 2.10. Neste caso, em princípio, o valor de $p(x|\theta)$ é estimado como

$$p(x|\theta) = \sum_{h} p(x|h,\theta)p(h|\theta) \tag{2.34}$$

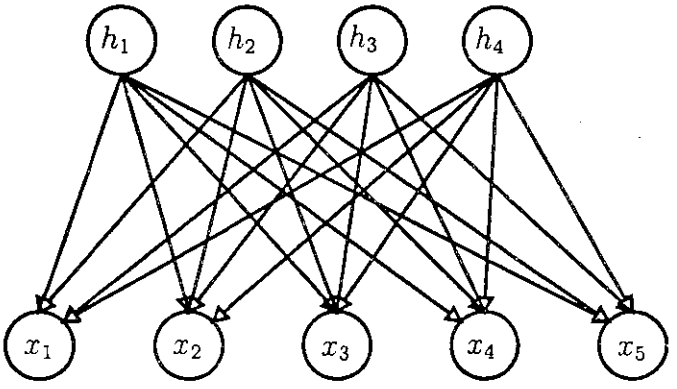


Figura 2.10: Redes Bayesianas de múltiplas causas

Ocorre que a expansão do somatório em (2.34) origina uma explosão combinatorial o que inviabiliza a realização de um cálculo exato. Frey explora duas linhas para realizar uma aproximação do cálculo em (2.34), que são: a amostragem de Gibbs e a estimação de um limite inferior para a função de máxima verossimilhança — MLB (*Maximum likelihood-based estimation*) — usando o algoritmo EM (*Expectation Maximization*). Neste segundo caso, são realizadas experiências com dois métodos empregados no algoritmo EM: estimação da MLB por inferência variacional (Saul *et al.* [99]) e estimação da MLB usando uma máquina de Helmholts estocástica (Dayan *et al.* [28]). Nos estudos experimentais, Frey avaliou diversas variações dos algoritmos apresentados, como a utilização de mais de uma camada escondida e agrupamento de máquinas de Helmholtz. De um modo geral pôde-se constatar que o uso de redes autoregressivas e o agrupamento de máquinas de Helmholtz tiveram os melhores desempenhos em testes com reconhecimento de imagens de caracteres manuscritos, sendo o erro originado por estes métodos muito próximos entre si. Entretanto na fase de treinamento, redes autoregressivas demandaram um tempo substancialmente menor (até dez vezes menor) que o utilizado pelas máquinas de Helmholtz.

2.5 Conclusão

Neste capítulo foi apresentada a teoria básica relacionada com redes Bayesianas que cobre duas linhas principais: como realizar inferência com base em um algoritmo de propagação de probabilidades e como obter a rede a partir de uma base de casos. Foi também realizado um estudo importante destas redes no contexto em que se localiza a proposta de tese, isto é, procurou-se investigar o uso de redes Bayesianas aplicadas ao problema de classificação. Através deste estudo pode-se perceber que, apesar de ser uma ferramenta apropriada para estimação de densidades, a aplicação de redes Bayesianas ao problema de classificação não é uma extensão natural da teoria apresentada nas seções iniciais do capítulo. Conforme pode ser observado nos trabalhos realizados por Friedman *et al.* [41] e Sing e Provan [104], é necessário realizar alguns ajustes de modo a adaptar as redes Bayesianas ao problema de classificação tirando proveito do conhecimento existente na área. No trabalho desenvolvido por Frey [37], os algoritmos e a estrutura das redes estudados se distanciaram ainda mais das proposições que nortearam os trabalhos de Friedman *et al.* e de Singh e Provin. Neste último, as redes Bayeianas concebidas possuiam uma estrutura tão complexa que não pôde ser empregado o mecanismo de inferência estudado na Seção 2.2, o que demandou a investigação de algoritmos de inferência para realização de um cálculo aproximado. Esta é uma linha que parece promissora mas que infelizmente não aproveita desenvolvimentos bem consolidados na área abordados no início do capítulo. O método apresentado nesta proposta, assim como os revisados nas Seções 2.4.3 e 2.4.2, também procura realizar adaptações para aplicar os algoritmos revisados nas Seções 2.2 e 2.3 ao problema de classificação.

O problema de classificação será estudado em maior profundidade no contexto de combinação de classificadores no Capítulo 3 que, assim como redes Bayesianas, fornecem uma base teórica que apóia a proposição do método apresentado no Capítulo 4.

Capítulo 3

Combinação de classificadores

A combinação de classificadores é uma linha de pesquisa cujos trabalhos mais importantes foram propostos a partir do início dos anos 90. Trata-se portanto de uma linha relativamente nova e potencialmente promissora. O objetivo deste capítulo é fornecer uma visão contextual do assunto a fim de que se que se possa compreender as motivações para construção do método proposto. O capítulo está organizado da seguinte forma: na Seção 3.1 apresentam-se as dificuldades inerentes ao aprendizado de classificadores e uma justificativa teórica para utilização de combinação de classificadores; na Seção 3.2 realiza-se uma revisão de literatura sobre os principais trabalhos na área; e a Seção 3.3 encerra o capítulo com conclusões gerais sobre o conteúdo apresentado.

3.1 Dificuldades relacionadas com aprendizado de classificadores

Um classificador, segundo o ponto-de-vista estatístico, é um modelo matemático que generaliza o conhecimento adquirido a partir de um conjunto de treinamento para realizar predições sobre elementos não vistos. De acordo com este enfoque, o termo aprendizado se aplica à criação de um modelo estatístico que, como tal, possui vantagens e limitações. Alguns dos principais problemas relacionados com a construção deste modelo serão discutidos ao longo desta seção.

Considere um problema hipotético. Suponha serem conhecidas as distribuições dos padrões de três classes bem como os valores *a priori* de cada uma delas, isto é, conhecem-se os valores de $p(\boldsymbol{x}|\omega_i), i = 1, 2, 3$, cujas curvas de nível estão ilustradas na Figura 3.1, e $p(\omega_i)$. Uma vez que as distribuições são conhecidas, pode-se classificar um padrão de teste, $\boldsymbol{x}$, atribuindo a ele o valor da classe que maximiza $p(\omega_i|\boldsymbol{x})$ usando o Teorema de Bayes

$$p(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i)p(\omega_i)}{p(\boldsymbol{x})}$$

Sendo conhecidas informações estatísticas completas sobre $p(\boldsymbol{x}|\omega_i)$ e $p(\omega_i)$, o classificador Bayesiano

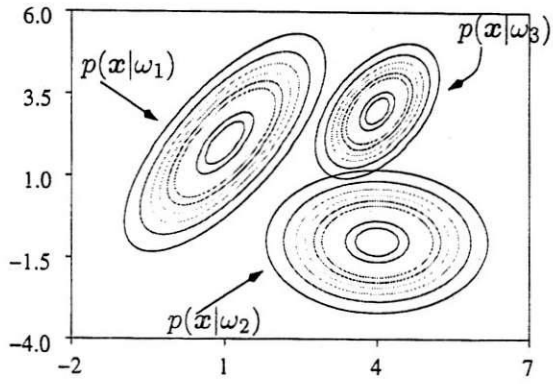


Figura 3.1: Curvas de nível de densidades conhecidas

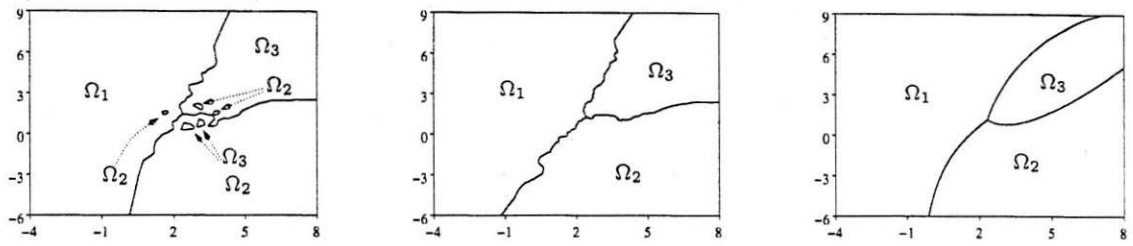
obtido é chamado ideal, denotado por $h^*(\boldsymbol{x})$. Eliminando-se o fator de escala $p(\boldsymbol{x})$, $h^*(\boldsymbol{x})$, pode ser formalmente definido como

$$h(\boldsymbol{x}) : \mathbb{R}^p \longrightarrow C = \{\omega_i\} \subset \mathbb{Z}$$
$$h(\boldsymbol{x}) = \underset{i}{\text{argmax}}\{p(\boldsymbol{x}|\omega_i)p(\omega_i)\}$$

O classificador Bayesiano ideal pode ser adotado como um referencial para indicar o quão eficiente é um classificador, pois o erro médio cometido por $h^*(\boldsymbol{x})$ é mínimo, sendo $h^*(\boldsymbol{x})$ ótimo quando considera-se que o custo associado a uma predição com erro seja igual para todas as classes (Webb [115]). Na prática nem sempre é possível avaliar o quão próximo um classificador, $h(\boldsymbol{x})$, está de $h^*(\boldsymbol{x})$ mas, neste exemplo, pode-se realizar tais comparações usando $h^*(\boldsymbol{x})$ como referencial.

Suponha que sejam extraídas amostras aleatórias das distribuições $p(\boldsymbol{x}|\omega_i)$ para formar um conjunto $\mathcal{T} = \{(\boldsymbol{x}_i, y_i); \boldsymbol{x} \in \mathbb{R}^p, y \in C, i = 1, \ldots, n\}$ usado no treinamento de um classificador 1-NN, isto é, o classificador que atribui a um padrão de teste o rótulo de seu vizinho mais próximo encontrado no conjunto de treinamento, denotado por $h_{1NN}(\boldsymbol{x})$, cujas regiões de decisão estão apresentadas na Figura 3.2(a). O símbolo Ω será usado para denotar regiões de decisão, sendo Ω_i a região em que os padrões são classificados como pertencentes à classe ω_i. Em função de $h_{1NN}(\boldsymbol{x})$ considerar apenas o rótulo do vizinho mais próximo, ele classifica corretamente todos os elementos do conjunto de treinamento, mas alguns elementos expúrios ocorridos em $\mathcal{T}$ originam a formação de fronteiras, tal como ilhas ou agrupamentos (*clusters*), que difirem significativamente daquelas traçadas por $h^*(\boldsymbol{x})$, Figura 3.2(c). Aumentando-se o número de vizinhos visitados, por exemplo, considerando um classificador 5-NN, obtém-se regiões de decisão mais próximas daquelas geradas pelo classificador $h^*(\boldsymbol{x})$ (Figura 3.2(b)). Na realidade, sendo n suficientemente grande, aumentado-se o valor de k (número de vizinhos visitados), o classificador resultante se

aproxima assintoticamente de $h^*(\boldsymbol{x})$ (Cover e Hart [26]), o que justifica o fato de $h_{1NN}(\boldsymbol{x})$ estar mais distante de $h^*(\boldsymbol{x})$ do que $h_{5NN}(\boldsymbol{x})$ neste exemplo. O classificador $h_{5NN}(\boldsymbol{x})$, tanto quanto $h^*(\boldsymbol{x})$, erram em realizar a predição de alguns elementos do conjunto de treinamento mas na média, considerando um grande número de testes, o erro cometido por $h_{5NN}(\boldsymbol{x})$ é inferior àquele cometido por $h_{1NN}(\boldsymbol{x})$. Diz-se neste caso que $h_{1NN}(\boldsymbol{x})$ tornou-se superespecializado[1] em classificar os elementos do conjunto de treinamento ao custo de ter perdido a capacidade de generalização.



(a) $h_{1NN}(\boldsymbol{x})$ (b) $h_{5NN}(\boldsymbol{x})$ (c) $h^*(\boldsymbol{x})$

Figura 3.2: Regiões de decisão

Com classificadores MLP (*multilayer perceptron*), o problema da superespecialização ocorre quando se acrescentam novas camadas ou novos neurônios à camada escondida. Por outro lado, poucos neurônios nesta camada podem redundar na incapacidade de construir um modelo eficiente, o que origina o dilema superespecialização-generalização. Em se tratando de classificadores neurais, algumas técnicas de minimizar tal problema são obtidas pelo emprego da validação cruzada, parada brusca[2] e regularização. A validação cruzada, que é uma das formas mais popular, consiste em realizar um treinamento com base em dois conjuntos, um deles é utilizado para treinamento efetivamente e outro para validação. O processo de treinamento dura enquanto as curvas de erro obtidas nos conjuntos de validação e treinamento estiverem em declínio. Sabe-se que a partir de uma determinada iteração a curva do erro de validação torna-se ascendente enquanto àquela do conjunto de treinamento continua em declínio. Esta iteração sinaliza o término do processo de treinamento pois a partir deste ponto a rede começa a tornar-se superespecializada.

O dilema superespecialização-generalização pode ser apresentado de outra forma, com base no erro empírico, isto é, o erro médio que o classificador comete em um conjunto de teste independente. Para facilitar esta análise considere uma mudança de enfoque quanto ao tipo de problema estudado. Considere um modelo de regressão, ao invés do modelo de classificação,

[1] do inglês *overfitting*
[2] do inglês *early stop*

como está sendo tratado ao longo do texto. O modelo de regressão é um modelo probabilístico que relaciona um conjunto de variáveis independentes ou preditoras e um conjunto de variáveis dependentes que, ao contrário do modelo de classificação, são contínuas. Além disto, para simplificar os desenvolvimentos considere que sejam tratados funções de uma única variável. As conclusões tiradas desta análise podem ser estendidas para o tratamento de funções de mais de uma variável e para o problema de classificação sem perda de generalidade, a mudança de enfoque e simplificação adotadas atendem unicamente ao propósito de facilitar a apresentação do assunto.

Sejam X e Y duas variáveis aleatórias. Sendo X, chamada preditor, e Y, chamada resposta. O modelo de regressão é um modelo estatístico que relaciona X e Y como

$$Y = f(X) + \epsilon \tag{3.1}$$

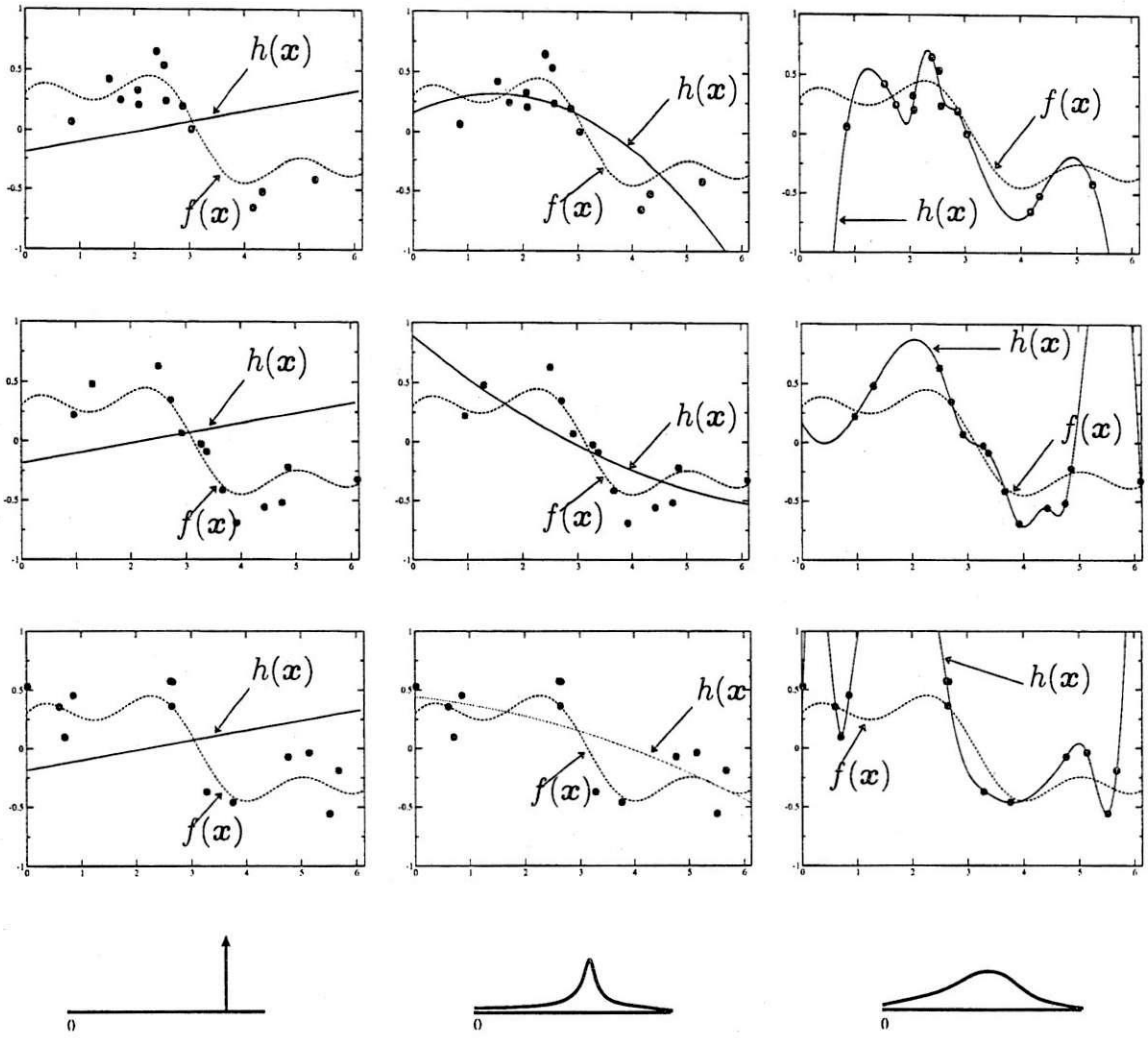
em que $f(\cdot)$ é uma função determinística, possivelmente não-linear, e ϵ um erro aleatório com valor esperado nulo e independente de $f(\cdot)$. Um algoritmo de aprendizado é um procedimento que utiliza um conjunto de treinamento $\mathcal{T} = \{(x_i, y_i); i = 1, \ldots, n\}$ para atualizar os parâmetros w de uma aproximação de $f(\cdot)$, denotada como $h(\cdot; w)$.

Na análise que se deseja fazer, procura-se avaliar a distribuição do erro empírico para diferentes métodos de aprendizado identificando sua tendência, ou valor médio, e dispersão.

Considere inicialmente que a função $h(\cdot, w)$ produzido pelo método de aprendizado seja sempre a mesma independentemente do conjunto de treinamento, Figura 3.3(a). Neste caso, independentemente do conjunto de treinamento empregado, o erro empírico em um dado conjunto de teste será sempre o mesmo, visto que $h(x; w)$ é sempre o mesmo. Como não existe dispersão na distribuição do erro, já que ele se concentra em um único ponto, sua variância é nula. Se um procedimento mais complexo for utilizado, por exemplo, a aproximação por um polinômio de segundo grau, obtém-se uma descrição melhor do conjunto de treinamento que implica no deslocamento do valor de tendência, ou viés, do erro para mais próximo da origem, entretanto aumenta-se um pouco a variância pois nem toda aproximação por um polinômio de grau dois reproduz o mesmo erro em um conjunto de teste independente, Figura 3.3(b). Aumentando-se mais a complexidade do procedimento de aprendizado, por exemplo, utilizando-se uma aproximação por um polinômio de grau dez, obtém-se um modelo bastante hábil em descrever o conjunto de treinamento (superespecialização) que, no entanto, é bastante sensível ao conjunto de treinamento e tenderá a possuir erro elevado quando testado com elementos não-vistos (baixa generalização), caso os dados de treinamento não sejam representativos, Figura 3.3(c).

Considere que o erro empírico seja fornecido pelo erro médio quadrático multiplicado pelo fator de escala $1/2$, denotado por $\mathcal{E}(w)$. Tem-se que

$$\mathcal{E}(w) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - h_{NN}(x; w))^2 \tag{3.2}$$



(a) $h(\boldsymbol{x};\boldsymbol{w}) = linear$ (b) $h(\boldsymbol{x};\boldsymbol{w}) = w_2x^2+w_1x+w_0$ (c) $h(\boldsymbol{x};\boldsymbol{w}) = \sum_{k=0}^{10} w_k x^k$

Figura 3.3: Análise do viés e variância do erro empírico para três métodos de regressão

que abreviadamente pode ser expresso por

$$\mathcal{E}(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{E}_{\mathcal{T}}[(Y - h_{NN}(\boldsymbol{X};\boldsymbol{w}))^2] \tag{3.3}$$

em que o operador $\boldsymbol{E}_{\mathcal{T}}$ representa uma expectância calculada com base nos elementos de $\mathcal{T}$.

Somando-se e subtraíndo-se $f(\cdot)$ em (3.3) segue-se que

$$\begin{aligned}
\mathcal{E}(\boldsymbol{w}) &= \frac{1}{2}\boldsymbol{E}_{\mathcal{T}}[(Y - f(\boldsymbol{X}) + (f(\boldsymbol{X}) - h_{NN}(X;\boldsymbol{w})))^2] && (3.4)\\
&= \frac{1}{2}\boldsymbol{E}_{\mathcal{T}}[(\epsilon + (f(\boldsymbol{X}) - h_{NN}(X;\boldsymbol{w})))^2] && (3.5)\\
&= \frac{1}{2}\boldsymbol{E}_{\mathcal{T}}[\epsilon^2] + \frac{1}{2}\boldsymbol{E}_{\mathcal{T}}[(f(\boldsymbol{X}) - h_{NN}(\boldsymbol{X};\boldsymbol{w}))^2] + \boldsymbol{E}_{\mathcal{T}}[\epsilon(f(\boldsymbol{X}) - h_{NN}(\boldsymbol{X};\boldsymbol{w}))] && (3.6)
\end{aligned}$$

Como ϵ tem valor esperado nulo e é independente de $f(\cdot)$, por hipótese, e independente de $h_{NN}(\cdot\,;\boldsymbol{w})$, por pertencerem a modelos distintos, então o termo mais à direita de (3.6) tende a zero. O erro médio quadrático, portanto, pode ser aproximado por

$$\mathcal{E}(\boldsymbol{w}) \approx \frac{1}{2}\boldsymbol{E}_{\mathcal{T}}[\epsilon^2] + \frac{1}{2}\boldsymbol{E}_{\mathcal{T}}[(f(\boldsymbol{X}) - h_{NN}(\boldsymbol{X};\boldsymbol{w}))^2] \tag{3.7}$$

Sendo que $\boldsymbol{E}_{\mathcal{T}}[\epsilon^2]$ é um termo que não depende da rede, trata-se de um componente de erro intínseco do modelo e que portanto sempre existirá, por melhor que seja a escolha de $\boldsymbol{w}$ adotada. Por conta disto, $\boldsymbol{E}_{\mathcal{T}}[\epsilon^2]$ será ignorado na análise seguinte que procura avaliar o comportamento do erro médio quadrático em função de uma escolha em $\boldsymbol{w}$

Geman *et al.* [45] mostram que o termo à direita em (3.7) pode ser decomposto como

$$\boldsymbol{E}_{\mathcal{T}}[(h_{NN}(\boldsymbol{X};\boldsymbol{w}) - f(\boldsymbol{X}))^2] = B^2(\boldsymbol{w}) + V(\boldsymbol{w}) \tag{3.8}$$

com

$$V(\boldsymbol{w}) = \boldsymbol{E}_{\mathcal{T}}[(h_{NN}(\boldsymbol{X};\boldsymbol{w}) - \boldsymbol{E}_{\mathcal{T}}[h_{NN}(\boldsymbol{X};\boldsymbol{w})])^2] \tag{3.9}$$

e

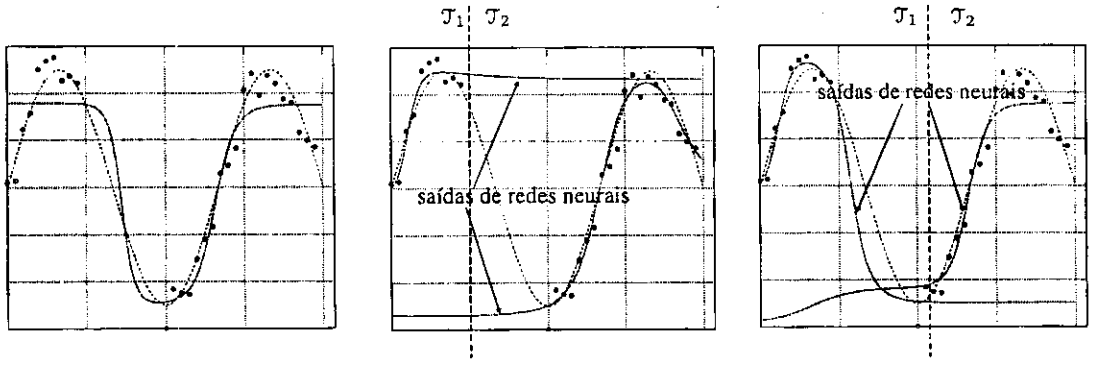
$$B^2(\boldsymbol{w}) = (\boldsymbol{E}_{\mathcal{T}}[h_{NN}(\boldsymbol{X};\boldsymbol{w})] - f(\boldsymbol{X}))^2 \tag{3.10}$$

O termo $V(\boldsymbol{w})$, pode-se observar, é a variância do erro produzido por $h(\boldsymbol{x};\boldsymbol{w})$. Na prática indica um termo influenciado pela sensibilidade em relação aos dados de treinamento. Pode-se portanto alterar a variância alterando-se o conjunto de treinamento. O termo $B(\boldsymbol{w})$, o quadrado do viés, por sua vez, é influenciado pela capacidade de $h(\cdot,\boldsymbol{w})$ em se ajustar à função de predição, então independe do conjunto de treinamento, naturalmente, desde que o conjunto seja representativo. A minimização de $\mathcal{E}(\boldsymbol{w})$ requer a minimização simultânea de $V(\boldsymbol{w})$ e $B^2(\boldsymbol{w})$. Ocorre que ao minimizar $V(\boldsymbol{w})$ aumenta-se o valor de $B(\boldsymbol{w})$, e vice-versa, o que origina o dilema viés-variância. Em outras palavras, ao tornar a saída desejada mais próxima dos dados de treinamento, aumenta-se a variância. Por outro lado, ao procurar minimizar a variância, tão somente, obtém-se uma aproximação que ignora a função de predição.

Uma maneira de reduzir o erro de aproximação, como sugerido na literatura (Haykin [55], Bishop [7]) pode ser realizado através da introdução de um conhecimento *a priori* sobre a função de predição. Isto contribuiria naturalmente para redução do viés e, tomando um conjunto de treinamento suficientemente grande, também poderia-se reduzir a variância. Bishop [7] discute várias abordagens para solucionar este problema, uma delas é pela utilização de um agrupamento de redes neurais. Tumer e Ghosh [111] demonstraram que ao tomar o valor médio de R classificadores, a contribuição da variância é decrescida de $1/R$. A combinação de classificadores,

portanto, pode ser uma estratégia plausível para encontrar uma solução de compromisso para o dilema viés-variância.

O agrupamento de classificadores, entretanto, introduz novos problemas, tais como: como realizar a divisão do conjunto de treinamento e que estratégia usar para combinar as predições individuais. Para se ter uma idéia destes problemas considere o seguinte exemplo: suponha que a variável dependente seja obtida pela adição da função seno a um ruído uniforme de valor esperado nulo, definida no intervalo $[0, 3\pi/2]$, Figura 3.4(a). Suponha que uma rede MLP totalmente conectada, com dois neurônios na camada escondida seja usada para realizar a aproximação de $f(x) = \sin(x)$. Observe que, como a saída da rede é uma aplicação de uma função monotonicamente crescente sobre uma combinação linear duas funções sigmóides, já que existem apenas dois neurônios na camada escondida, deslocada por um escalar, a rede possui uma dificuldade intrínseca de aproximar funções com mais de um ponto crítico. Assim, independentemente de quantas iterações sejam utilizadas no treinamento, o erro empírico terá um viés elevado já que neste intervalo há mais de um ponto-crítico. Entretanto, se o conjunto de treinamento for dividido em duas partes como ilustrado na Figura 3.4, poderia-se obter um modelo seguramente mais representativo, Figura 3.4(c). O que nem sempre é uma tarefa simples, visto que uma má escolha dos conjuntos de treinamento pode redundar em uma total falta de conhecimento de determinadas regiões do espaço de atributos, Figura 3.4(b). Estes problemas serão abordados ao longo da Seção 3.2, na qual se realiza um estudo sobre algumas soluções propostas na literatura.



(a) Aproximação original (b) Um agrupamento de RN (c) Outro agrupamento

Figura 3.4: Exemplo de agrupamento de RN

3.2 Combinação de classificadores

A idéia de combinar múltiplos classificadores não é nova, mas um grande números de contribuições nesta área surgiu apenas a partir dos anos 90 com a renovação das pesquisas em redes

neurais. De um modo geral, a vantagem em usar um combinador é que pode-se melhorar o desempenho do sistema fazendo com que as deficiências de um classificador sejam suprimidas pelo bom desempenho de outros (Kittler *et al.* [69]). Em detalhes, algumas das principais vantagens em empregar uma estratégia de combinação, como citado em Jain *et al.* [60] são:

i Poder lidar com predições de classificadores baseados em diferentes espaços de atributos. Uma pessoa, por exemplo, poderia ser identificada por sua voz, imagem da face, assinatura, etc.

ii Poder lidar com tipos diferentes de classificadores baseados no mesmo espaço de atributos. Diferentes classificadores têm desempenhos locais diferentes, isto é, em dados intervalos do espaço de características, os classificadores possuem desempenhos distintos. Desta forma, o bom desempenho local de um classificador pode compensar as deficiências locais de outros.

iii Poder lidar com classificadores homogêneos baseados no mesmo espaço de características. Alguns classificadores, como redes neurais, mesmo quando treinados com os mesmos dados realizam predições distintas em função da aleatóriedade no processo de inicialização.

Um esquema de combinação não deve ser aplicado quando os classificadores componentes são muito equivalentes. Se não existir divergência a utilização do combinador torna-se sem sentido. Também não se deve usar um combinador quando o problema pode ser satisfatoriamente resolvido por um classificador isoladamente.

Tipicamente, um esquema de classificação é composto por um conjunto de classificadores organizados em uma arquitetura e uma regra de combinação. Quanto à arquitetura estes esquemas são classificados como lineares, paralelos e hierárquicos. Na abordagem linear, os classificadores são invocados em série. A saída de um classificador fornece uma classificação *a priori* que é sucessivamente refinada por classificadores mais especializados em cadeia. Na abordagem em paralelo, as saídas dos diversos classificadores são combinadas simultaneamente para realização de uma predição única pelo combinador. Na abordagem hierárquica, ocorre uma combinação das arquiteturas mencionadas anteriormente, isto é, as predições dos classificadores são combinadas tanto em série como em paralelo.

Quanto à regra de combinação, os combinadores são classificados como baseados em regras fixas ou estáticas e baseados em treinamento (Webb [115]). Uma análise mais detalhada a respeito da avaliação do combinador quanto à regra de combinação será desenvolvida nas Seções 3.2.1 e 3.2.3.

3.2.1 Combinadores baseados em regras fixas

Regras de combinação fixas realizam um procedimento pré-determinado. A forma como as entradas são combinadas para gerar uma saída única é estabelecida *a priori*, isto é, o conhecimento

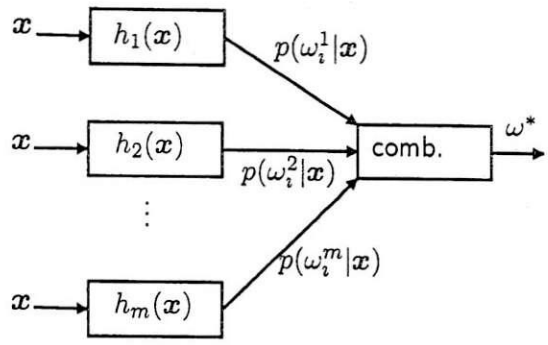


Figura 3.5: Esquema de combinação paralelo

requerido para realizar a combinação não é adquirido através de um processo de treinamento. A fim de entender seu princípio de funcionamento, considere a classificação apresentada por Xu e Krzyżak [116] para categorizar um classificador. Xu e Krzyżak [116] avaliam combinação de classificadores quanto ao que eles referem como níveis de informação. Segundo os autores há três tipos de níveis de informação, a saber:

Tipo 1 Nível abstrato — são classificadores que produzem como saída apenas o rótulo a qual o padrão apresentado fora classificado. Ex: classificador sintático puro.

Tipo 2 Nível de posto — são classificadores que organizam os rótulos de saída em postos. Tal organização possibilita listar em uma dada ordem, da melhor para a pior opção, os rótulos associados a um dado padrão. Ex: redes neurais.

Tipo 3 Nível de medição — são classificadores que fornecem para cada rótulo uma medição da probabilidade *a posteriori* $p(\omega_i|\boldsymbol{x})$. Ex: classificador Bayesiano, HMM (cadeias escondidas de Markov).

A classificação segundo níveis de informação é muito importante para o entendimento das regras de combinação apresentadas a seguir. Nesta seção serão estudadas algumas das regras de combinação mais populares mencionadas na literatura.

Regra da média (Xu e Krzyżak [116])

Esta regra aplica-se a classificadores do tipo 3 organizados em um esquema de combinação paralelo, como ilustrado na Figura 3.5. Xu e Krzyżak [116] consideram que a decisão tomada pelo combinador pode ser realizada com base numa estimação de $p(\omega_i|\boldsymbol{x})$. $p(\omega_i|\boldsymbol{x})$ é aproximado pelo valor médio de $p(\omega_i^j|\boldsymbol{x})$, $j = 1, \ldots, m$, em que m é o número de classificadores. Sendo assim, atribui-se a um padrão desconhecido $\boldsymbol{x}$, o rótulo ω^*, que maximiza o somatório em j de $p(\omega_i^j|\boldsymbol{x})$. Portanto,

$$\omega^* = \underset{i}{\operatorname{argmax}}\left\{ \sum_{j=1}^{m} p(\omega_i^j|\boldsymbol{x}) \right\} \tag{3.11}$$

Regra do produto (Kittler et al. [69])

A regra do produto também é aplicada a classificadores do tipo 3 mas, ao contrário do esquema abordado anteriormente, admite-se neste caso que as entradas possam ser distintas. Tem-se assim m entradas, denotadas por $\boldsymbol{x}_i$, $i = 1, \ldots, m$, em que m representa o número de classificadores que, por hipótese, são independentes, isto é, admite-se que $p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m|\omega_i) = \prod_j p(\boldsymbol{x}_j|\omega_i)$.

Como usual, considera-se como meta introdutória a obtenção de uma expressão que forneça o valor de $p(\omega_i|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$. Assim, usando o Teorema de Bayes tem-se que

$$\begin{aligned} p(\omega_i|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) &= \frac{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m|\omega_k)p(\omega_i)}{\sum_{k=1}^{n} p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m|\omega_k)p(\omega_k)} \\ &= \frac{p(\omega_i)\prod_{j=1}^{m} p(\boldsymbol{x}_j|\omega_i)}{\sum_{i=1}^{n} p(\omega_k)\prod_{j=1}^{m} p(\boldsymbol{x}_j|\omega_k)} \end{aligned} \tag{3.12}$$

Como $p(\boldsymbol{x}_j|\omega_i)$ é desconhecido, pode-se aplicar novamente o Teorema de Bayes para expressar a equação (3.12) em função das distribuições de $p(\omega_i|\boldsymbol{x}_j)$ que são conhecidas, pois o método é aplicado a classificadores do tipo 3. Além disto, como o denominador de (3.12) é constante para todo i, ele pode ser ignorado nos desenvolvimentos subsequentes. Assim, o valor de ω_i que maximiza $p(\omega_i|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$, denotado por ω^*, pode ser expresso como

$$\begin{aligned} \omega^* &= \underset{i}{\operatorname{argmax}}\{p(\omega_i|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)\} \\ &= \underset{i}{\operatorname{argmax}}\left\{ p(\omega_i) \prod_{j=1}^{m} \frac{p(\omega_i|\boldsymbol{x}_j)p(\boldsymbol{x}_j)}{\sum_{j=1}^{m} p(\omega_i|\boldsymbol{x}_j)p(\boldsymbol{x}_j)} \right\} \\ &= \underset{i}{\operatorname{argmax}}\left\{ p(\omega_i)^{-(m-1)} \prod_{j=1}^{m} p(\omega_i|\boldsymbol{x}_j)p(\boldsymbol{x}_j) \right\} \end{aligned}$$

Supondo que as ocorrências de $\boldsymbol{x}_j$ sejam equiprováveis, então segue-se que

$$\omega^* = \underset{i}{\operatorname{argmax}}\left\{ p(\omega_i)^{-(m-1)} \prod_{j=1}^{m} p(\omega_i|\boldsymbol{x}_j) \right\} \tag{3.13}$$

A expressão dada em (3.13) é bastante susceptível a problemas de instabilidade numérica, já que algum dos fatores que a compõe pode ter valor muito próximo de zero. A fim de resolver este problema, Kittler *et al.* [69] propuseram a regra da soma, que consiste numa variação da regra do produto pela introdução de uma nova hipótese.

Regra da soma (Kittler et al. [69])

A regra da soma consiste numa variação da regra do produto, pela introdução da hipótese de que $p(\omega_i|\boldsymbol{x}_j)$ é próximo de $p(\omega_i)$. Admite-se que

$$p(\omega_i|\boldsymbol{x}_j) = p(\omega_i)(1+\delta_{ij}) \tag{3.14}$$

com $\delta_{ij} \ll 1$.

Substituindo (3.14) em (3.13), tem-se que

$$\begin{aligned}\omega^* &= \underset{i}{\operatorname{argmax}}\left\{p(\omega_i)^{-(m-1)}\prod_{j=1}^{m}p(\omega_i)(1+\delta_{ij})\right\} \\ &= \underset{i}{\operatorname{argmax}}\left\{p(\omega_i)\prod_{j}(1+\delta_{ij})\right\}\end{aligned}$$

Como $\delta_{ij} \ll 1$, pode-se aproximar $\prod_j(1+\delta_{ij})$ pela sua expansão de primeira ordem, o que implica em

$$\omega^* = \underset{i}{\operatorname{argmax}}\left\{p(\omega_i)(1+\sum_{j=1}^{m}\delta_{ij})\right\} \tag{3.15}$$

A fim de expressar $\sum_j \delta_{ij}$ em função de termos conhecidos, pode-se aplicar o somatório em j nos dois lados da igualdade em (3.14), obtendo

$$\begin{aligned}\sum_{j=1}^{m}p(\omega_i|\boldsymbol{x}_j) &= \sum_{j=1}^{m}p(\omega_i)(1+\delta_{ij}) \\ \Rightarrow \sum_{j=1}^{m}p(\omega_i|\boldsymbol{x}_j) &= p(\omega_i)(m+\sum_{j=1}^{m}\delta_{ij}) \\ \Rightarrow \sum_{j=1}^{m}\delta_{ij} &= \frac{\sum_{j=1}^{m}p(\omega_i|\boldsymbol{x}_j) - mp(\omega_i)}{p(\omega_i)}\end{aligned} \tag{3.16}$$

Substituindo (3.16) em (3.15), tem-se que

$$\omega^* = \underset{i}{\operatorname{argmax}}\left\{(1-m)p(\omega_i) + \sum_{j=1}^{m}p(\omega_i|\boldsymbol{x}_j)\right\} \tag{3.17}$$

Considerando $p(\omega_i)$ equiprovável, a expressão (3.17) redunda em (3.11).

Empiricamente, bons resultados em experimentos realizados por Kittler *et al.* [69], que avaliaram diversas regras de combinação fixas, foram obtidos quando utilizaram a regra da soma. Nestes experimentos, a combinação de classificadores baseada na regra da soma apresentou taxas de acerto superior àquelas obtidas por classificadores individuais. A regra do produto, por sua vez mostrou um desempenho geral inferior, tendo inclusive taxa de acerto menor que as obtidas por classificadores isolados.

Voto majoritário (Xu e Krzyżak [116])

Utiliza-se a regra do voto majoritário para classificadores do tipo 1, ou quando pode-se admitir as hipóteses de que

$$p(\omega_i|\boldsymbol{x}_j) = \begin{cases} 1 & \text{se } \omega_i = \omega_j \\ 0 & \text{se } \omega_i \neq \omega_j \end{cases}$$

e $p(\omega_i)$ eqüiprovável.

Assim, chamando $freq(\omega_i)$ a contagem ou freqüência absoluta do rótulo ω_i entre os resultados originados pelos classificadores, faz-se

$$\omega^* = \underset{i}{\text{argmax}}\{freq(\omega_i)\}$$

Uma atitude conservadora propõe admitir que ω^* seja considerado desconhecido caso exista empate entre os rótulos vencedores neste processo de votação.

Regras do máximo, mínimo e mediana (Kittler et al. [69])

Estas regras são na realidade extensões das regras do produto (3.13) e da soma (3.17) que executam um procedimento diferente pela adoção de algumas simplificações.

A regra do máximo, originada a partir de (3.17), é dada por

$$y^* = \max_i\{(1-m)p(\omega_i) + m \max_j p(\omega_i|\boldsymbol{x}_j)\}$$

em que se utiliza $m \max p(\omega_i|\boldsymbol{x}_j)$ para limitar superiormente $\sum_j p(\omega_i|\boldsymbol{x}_j)$. Naturalmente, admitindo iguais distribuições *a priori* de ω_i, tem-se

$$y^* = \max_i \max_j p(\omega_i|\boldsymbol{x}_j)$$

A regra do mínimo é uma extensão de (3.13) na qual maximiza-se $\prod_j p(\omega_i|\boldsymbol{x}_j)$ por $\min_j p(\omega_i|\boldsymbol{x}_j)$, obtendo-se

$$y^* = \max_i\{p^{-(m-1)}(\omega_i)\min_j p(\omega_i|\boldsymbol{x}_j)\}$$

em que, admitindo-se iguais valores de $p(\omega_i)$, tem-se

$$y^* = \max_i \min_j p(\omega_i|\boldsymbol{x}_j)$$

A regra da mediana é uma extensão da regra da soma quando se assume iguais distribuições *a priori* de ω_i. Neste caso a regra da soma torna-se a média de $p(\omega_i|\boldsymbol{x}_j)$, fazendo j variar para um

grupo de m classificadores. A regra da mediana aproxima a média de $p(\omega_i|\boldsymbol{x}_j)$ pela sua mediana. Tem-se portanto,

$$y^* = \max_i \text{med}\, p(\omega_i|\boldsymbol{x}_j)$$

Outras regras fixas

Além das regras mencionadas até então, uma variedade de outras está relacionada na literatura (Webb [115], Kittler *et al.* [69] e Xu e Krżyzak [116]). Foge ao propósito deste texto realizar uma cobertura exaustiva sobre este assunto. Sugere-se uma leitura das referências supra citadas para aprofundamento em agrupamento de classificadores baseados em regras de combinação fixas.

3.2.2 Métodos de amostragem do conjunto de treinamento

Um dos principais problemas envolvendo combinação de classificadores é a existência de dependência entre os mesmos. Como comentado na Seção 3.2.1 este é um problema particularmente influente quando se emprega a regra do produto. De fato, se os classificadores estiverem correlacionados apenas se consegue reduzir a variância do erro, enquanto que o viés mantém-se praticamente inalterado. Ocorre que mesmo classificadores distintos podem ser correlacionados se forem treinados com os mesmos dados. Uma maneira de efetivamente fazer os classificadores discordarem é treiná-los com conjuntos de treinamento distintos. Esta é a idéia dos métodos de amostragem do conjunto de treinamento. Dois dos principais métodos nesta linha serão discutidos ao longo desta seção.

Bagging (Breiman [12])

Bagging, contração de *bootstrap aggregating*, é um método de geração de conjuntos de treinamentos para um dado número de classificadores previamente estabelecidos. A idéia consiste em gerar conjuntos de treinamento por um processo de amostragem aleatória com reposição. Os classificadores são então treinados e aplica-se no combinador a regra do voto majoritário. Como a amostragem é feita com reposição, surgirão alguns elementos replicados nos novos conjuntos de treinamento. A probabilidade de um elemento ocorrer pelo menos uma vez em uma amostra de tamanho n, que corresponde à proporção de elementos distintos na amostra, é dada por

$$1 - \left(1 - \frac{1}{n}\right)^n$$

de tal modo que, no limite, quando $n \to \infty$, espera-se encontrar $1 - e^{-1} = 63\%$ elementos distintos.

É importante observar que, se os classificadores que constituem o agrupamento forem instáveis, por exemplo, redes neurais e árvores de decisão, isto é, se respondem de forma bastante diferenciada quando treinados com dados ligeiramente distintos, então o processo de amostragem proposto resultará em um conjunto de classificadores distintos, sendo portanto um procedimento válido. Se por outro lado, os classificadores forem estáveis, ex. k-NN, então a combinação resultante teria pouco efeito, já que os classificadores tenderiam a apresentar a mesma predição.

Boosting (Freund e Schapire [36])

O método *boosting* também é aplicado a classificadores baseados no mesmo espaço de características. Assim como o anterior, também trata-se de um método para geração de conjuntos de treinamento e utiliza-se no combinador a regra do voto majoritário. Mas, diferentemente do método *bagging*, os conjuntos de treinamento não são gerados simultaneamente. Neste caso os conjuntos são gerados em série e para cada um deles atribui-se um peso usado no processo de combinação. A regra do voto majoritário, portanto, é regida com base nas ponderações determinadas no treinamento.

O método foi concebido para combinar classificadores que podem tratar entradas com pesos. Considera-se a princípio que os classificadores envolvidos sejam capazes de levar em consideração além do par ordenado padrão/rótulo – $(\boldsymbol{x}_i, y_i)$ – um peso associado a ele – w_i. A idéia de um modo geral consiste em, dado um conjunto de treinamento $\mathcal{T} = \{(\boldsymbol{x}_i, y_i); i = 1, \ldots, n\}$ e um conjunto de classificadores $h_j(\boldsymbol{x}), j = 1, \ldots, m$, iniciar um processo iterativo em que, a cada iteração, sejam estabelecido pesos w_i para os elementos do conjunto de treinamento, seja treinado um classificador $h_j(\boldsymbol{x})$ e seja estabelecido um peso associado ao classificador, denotado por e_j. Os pesos w_i são estabelecidos de tal modo que seu somatório é sempre igual a 1 em todas as iterações.

O algoritmo proposto por Freund e Schapire [36], denominado *ADABOOST*, para classificação binária, compreende os seguintes a seguir:

- Inicializar w_i com valores iguais — fazer $w_i = \frac{1}{n}$.
- Para $j = 1, \ldots, m$
  - Treinar $h_j(\boldsymbol{x})$ com os dados de $\mathcal{T}$ e pesos w_i.
  - Calcular e_j como

$$e_j = \sum_i w_i \qquad \text{para os padrões classificados erradamente}$$

  - Se $e_j > 0.5$ ou $e_j = 0$ finalizar o laço, senão

* Fazer $w_i = w_i\left(\frac{1-e_j}{e_j}\right)$ para os padrões classificados erradamente.
* Normalizar os pesos de modo que $\sum_i w_i = 1$

- Classificar um padrão desconhecido, $\boldsymbol{x}$, considerando que $h_j(\boldsymbol{x}) = 1$ para os padrões classificados como pertencentes à classe ω_1 e $h_j(\boldsymbol{x}) = -1$, no caso contrário, calcular $\hat{h}$ como

$$\hat{h} = \sum_j \log\left(\frac{1-e_j}{e_j}\right) h_j(\boldsymbol{x})$$

e, sendo $\hat{y}$ a saída do combinado, fazer $\hat{y} = 1$ se $\hat{h} > 0$, ou $\hat{y} = 0$, caso contrário.

Uma extensão do algoritmo *ADABOOST*, denominado *ADABOOST.MH*, para classificação de múltiplas classes fora proposta por Schapire e Singer [101].

No caso geral, quando os classificadores não são capazes de treinar padrões com pesos, pode-se eliminá-los fazendo um processo de amostragem com reposição, em que considera-se como pesos a proporção ou probabilidade de se sortear um padrão a partir do conjunto de treinamento original.

Segundo a literatura (Webb [115], Dietterich [30]) *boosting* é uma técnica que permite combinar classificadores que isoladamente são fracos, mas que em conjunto podem realizar uma boa predição. Dietterich [30] apresenta resultados experimentais mostrando que, de um modo geral, a utilização de uma estratégia de combinação, como *bagging*, resulta na obtenção de melhores taxas de acerto que aquelas produzidas pelos classificadores isoladamente. Nos seus experimentos, esta taxa de acerto é ainda ligeiramente maior nos casos em que fora empregado como estratégia de combinação a técnica *boosting*.

3.2.3 Combinadores baseados treinamento

As regras baseadas em treinamento formam um espectro muito amplo, especialmente pelas contribuições relativas à literatura que trata da combinação de classificadores neurais (Rogova [95], Hansen e Salamon [53], Hashem e Schmeiser [54], Cho e Kim [19] e [20]). Por se tratar de um assunto muito amplo, será dada ênfase apenas às técnicas mais populares. Aos leitores mais interessados, recomenda-se a leitura das referências anteriormente citadas e do Capítulo 8 de Webb [115], bem como das referências nele citadas.

Mistura de especialistas (Jacobs et al. [59] e Jordan e Jacobs [65])

A mistura de especialistas, proposta por Jacobs *et al.* [59] e posteriormente melhorada em Jordan e Jacobs [65], consiste num agrupamento em paralelo de classificadores baseados no mesmo espaço de características. O processo de combinação emprega uma idéia semelhante

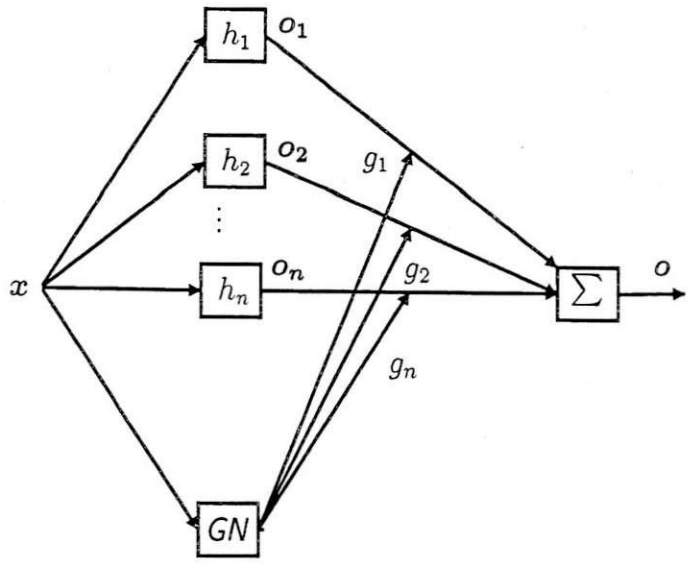


Figura 3.6: Mistura de Especialistas

àquela do voto majoritário. Na realidade, trata-se de uma soma ponderada das saídas de um grupo de redes neurais (especialistas), sendo os pesos obtidos através de um processo de treinamento. Como pode ser observado pela ilustração da Figura 3.6, os pesos, denotados por g_j, são as saídas de uma rede denominada *gatering network* (GN). Observe que se os pesos tiverem a mesma ordem de grandeza, os especialistas agiriam como se cooperassem entre si. Neste caso, a regra de combinação desempenharia um papel semelhante à do voto majoritário. Se, por outro lado, a ordem de grandeza de um único classificador fosse demasiadamente maior que a de todos os demais, estes agiriam como se participassem de um processo de competição. Esta é a idéia do método. Aplica-se um processo de segmentação do espaço de características e para cada uma destas regiões um único especialista responde majoritariamente pela classificação dos padrões nela inseridos. Ao contrário de outros métodos que realizam segmentação do espaço de características, como árvores de classificação (Breiman *et al.* [11]), as fronteiras entre as regiões não são rígidas, são suaves. Isto significa que nas regiões limítrofes, especialistas vizinhos exercem influência no processo de classificação.

O processo de treinamento, que consiste na obtenção dos parâmetros (pesos) dos especialistas e da porta (*gate*), pode ser realizado com base em um enfoque estatístico. Inicialmente considere que os especialistas e a porta sejam redes *perceptron* lineares. A fim de que se estabeleça o processo de competição esperado, poderia ser aplicada sobre a saída da porta a regra vencedor leva tudo[3] fazendo com que o peso associado ao especialista vencedor tivesse valor unitário enquanto que todos os demais tivessem valor nulo. Esta regra, embora atenda ao requisito de estabelecer um processo de competição, falha por não possuir algumas propriedades matemáticas desejáveis, tais como continuidade e diferenciabilidade. Sua suavização, dada pela expressão

[3]do inglês *winner takes all*

$$g_j(\boldsymbol{x}) = g_j(\boldsymbol{x}; \boldsymbol{v}) = \frac{e^{\boldsymbol{v}_j^T \boldsymbol{x}}}{\sum_k e^{\boldsymbol{v}_k^T \boldsymbol{x}}}, \tag{3.18}$$

em que $\boldsymbol{v}$ refere-se aos parâmetros (pesos) do combinador, é mais adequada. Uma rede cuja saída é computada pela expressão dada em (3.18) realiza um procedimento conhecido pela comunidade de redes neurais como método *softmax* que, sob determinadas condições, aproxima a saída da rede às probabilidades *a posteriori* – $p(\omega_i|\boldsymbol{x})$ (Bishop [7], pp 212-222). Para a comunidade de estatística a expressão em (3.18) é vista como um modelo de regressão denominado regressão logística múltipla (Webb [115]), um caso particular de um modelo de regressão mais amplo e largamente estudado pela comunidade de estatística — o modelo linear generalizado (MacCullah e Nelder [82]).

Considerando $g(\boldsymbol{x})$ proporções oriundas de uma distribuição multinomial, Jordan e Jacobs [65] modelaram a mistutra de especialistas como uma mistura de densidades, cujo valor da predição – y – dado um padrão de entrada – $\boldsymbol{x}$ – é dado por

$$p(y|\boldsymbol{x}, \Phi) = \sum_i g(i|\boldsymbol{x}, \boldsymbol{v}) p_i(y|\boldsymbol{x}, \boldsymbol{w}_i) \tag{3.19}$$

em que Φ refere-se a todos os padrões do modelo, incluindo os pesos dos especialistas – $\boldsymbol{w}$ – e da porta (*gate*) – $\boldsymbol{v}$. Jordan e Jacobs estudaram a aplicação do algoritmo EM para maximizar a função logarítmica de verossimilhança obtida a partir de (3.19) estabelecendo, desta forma, um procedimento estatístico para obtenção dos pesos, isto é, para realização do treinamento. Naturalmente, para ajustar o problema ao modelo dado em (3.19) é necessário admitir que as saídas dos especialistas possam ser regidas por distribuições conhecidas. Tipicamente se utiliza a distribuição normal quando se trata de um problema de regressão, utiliza-se a distribuição Bernoulli para problemas de classificação com duas classes e a distribuição multinomial para problemas de classificação com múltiplas classes.

A fim de introduzir não-linearidade ao modelo, uma vez que tanto os especialistas quanto a porta são redes *perceptron* linear, Jordan e Jacobs [65] propuseram um método no qual cada especialista é, em si, uma mistura de especialistas. Este método é denominado mistura hierárquica de especialistas. Embora pudesse ter sido empregada uma estratégia diferente para introduzir a não-linearidade desejada, como o emprego de redes MLP para realizar as tarefas dos especialistas e porta, a vantagem em usar a mistura hierárquica de especialistas é manter uma perspectiva uniforme do sistema de modo a poder ser empregada a mesma estratégia de treinamento, isto é, poder ser realizada a obtenção dos pesos através do algoritmo EM.

Tanto a mistura hierárquica de especialistas quanto as redes MLP podem aproximar ao grau de precisão desejável a distribuição *a posteriori* de ω, desde que se tenha um número de amostras suficientemente grande e um modelo suficientemente complexo (Bishop [7], pg. 214), o que

sugere, na prática, serem métodos com desempenho semelhantes. Uma das vantagens da mistura de especialistas mencionadas por Jordan e Jacobs é sua rápida convergência, uma vez que pode ser empregado na fase M (maximização) um método de otimização de convergência quadrática (método de Newton, por exemplo), comparativamente muito mais rápido que o algoritmo de descida na direção do gradiente empregado no algoritmo *backpropagation*.

Diversas extensões à proposição de Jordan e Jacobs foram propostas subsequentemente. Os principais desenvolvimentos nesta linha foram discutidos por Waterhouse [114] no seu trabalho de doutorado, que realiza uma apresentação minusciosa e clara da mistura de especialistas, sua contextualização em relação a outros métodos e as extensões mais importantes. Dentre estas extensões pode-se destacar a utilização de métodos Bayesianos para obtenção dos parâmetros do modelo, o desenvolvimento de métodos que dinamicamente realizam a expansão ou crescimento da estrutura hierárquica e o de métodos que estabelecem critérios de parada antecipada a fim de aumentar o poder de generalização do sistema.

Regra baseada no formalismo Bayesiano (Xu e Krzyżak [116])

Este método se aplica a classificadores do tipo 1, baseados em um mesmo espaço de características e agrupados em um esquema paralelo. O método emprega valores de $p(\omega_i|y_j^k)$, que correspondem à probabilidade do padrão de teste pertencer à classe ω_i dado que o k-ésimo classificador o tenha classificado como sendo da classe ω_j. Como estas probabilidades são desconhecidas, a fim de estimá-las, Xu e Krzyżak [116] sugerem que estes valores sejam estimados a partir de freqüências relativas calculadas da matriz de confusão dos classificadores. O cálculo de $p(\omega_i|y_j^k)$ é realizado como

$$p(\omega_i|y_j^k) = \frac{\eta_{ij}^k}{\sum_i \eta_{ij}^k} \tag{3.20}$$

em que η_{ij}^k corresponde à entrada (i,j) da matriz de confusão do k-ésimo classificador. Isto é, η_{ij}^k é o número de ocorrências com que h_k classificou um padrão da classe ω_i como sendo da classe ω_j.

O método Bayesiano procura determinar a classe ω_i que maximiza a distribuição de ω_i dado o estado da entrada do combinador, que corresponde ao estado do espaço de saída dos classificadores. Chamando $\hat{y}^k$ a saída do k-ésimo classificador, tem-se que

$$y^* = \underset{i}{\operatorname{argmax}}\{p(\omega_i|\hat{y}^1, \ldots, \hat{y}^m, \xi)\} \tag{3.21}$$

em que ξ representa um conjunto de parâmetros desconhecidos da distribuição. Assumindo a hipótese de independência dos classificadores e aplicando o Teorema de Bayes sobre (3.21) tem-se que

$$
\begin{aligned}
p(\omega_i|\hat{y}^1,\ldots,\hat{y}^m,\xi) &= \frac{p(\hat{y}^1,\ldots,\hat{y}^m|\omega_i,\xi)p(\omega_i|\xi)}{p(\hat{y}^1,\ldots,\hat{y}^m|\xi)} \\
&= \frac{p(\omega_i|\xi)\prod_{k=1}^{m} p(\hat{y}^k|\omega_i,\xi)}{\prod_{k=1}^{m} p(\hat{y}^k|\xi)}
\end{aligned}
$$

Aplicando novamente o Teorema de Bayes a fim de expressar (3.21) em função de $p(\omega_i|\hat{y}^k)$ tem-se que

$$
\begin{aligned}
p(\omega_i|\hat{y}^1,\ldots,\hat{y}^m,\xi) &= \frac{p(\omega_i|\hat{y}^1)p(\hat{y}^1)}{p(\omega_i|\xi)} \cdots \frac{p(\omega_i|\hat{y}^m)p(\hat{y}^m)}{p(\omega_i|\xi)} \frac{p(\omega_i)}{\prod_{k=1}^{m} p(\hat{y}^k)} \\
&= \frac{\prod_{k=1}^{m} p(\omega_i|\hat{y}^k)}{\prod_{k=1}^{m} p(\omega_i)} p(\omega_i) \qquad (3.22)
\end{aligned}
$$

O denominador em (3.22) é um fator de normalização, que assegura que $\sum_{i=1}^{n} p(\omega_i|\hat{y}^k)p(\omega_i) = 1$, logo, pode ser eliminado. Utilizando (3.20) como aproximação de $p(\omega_i|\hat{y}^k)$ e supondo ser $p(\omega_i)$ equiprovável, define-se y^* como

$$
y^* = \underset{i}{\operatorname{argmax}}\{\prod_{k=1}^{m} p(\omega_i|\hat{y}^k)\}
$$

BKS - Behavior-Knowledge Space (Huang e Suen [58])

O método BKS é similar à abordagem proposta por Xu e Krzyżak [116] apresentada na Seção 3.2.3, uma vez que ambos são métodos que realizam combinação de classificadores de tipo 1, organizados em um esquema em paralelo, e obtém uma base de conhecimento originada a partir da aplicação dos classificadores sobre o conjunto de treinamento. O que os autores denominam *behavior-knowledge space* é na realidade um espaço que guarda informações sobre o desempenho dos classificadores. Considere, por exemplo, dois classificadores, $h_1(\cdot)$ e $h_2(\cdot)$, cujas saídas estejam associadas a um conjunto de n categorias. Para este exemplo o BKS associado seria uma matriz de dimensão $n \times n$, em que cada linha estaria associada às predições do classificador h_1 e cada coluna, às predições do classificador h_2. Cada elemento (i,j), denominado unidade focal (*focal unit*), contém as seguintes informações:

$n_{ij}(k)$ — corresponde ao número de ocorrências em que um padrão da classe ω_k foi classificado por h_1 como sendo da classe ω_i e como sendo da classe ω_j pelo classificador h_2.

T_{ij} — corresponde ao número de ocorrências em que os padrões do conjunto de treinamento foram classificados por h_1 como sendo da classe ω_i e por h_2 como sendo da classe ω_j. Isto é, $T_{ij} = \sum_k n_{ij}$.

R_{ij} — corresponde ao elemento mais representativo da unidade focal. É calculado como $R_{ij} = \underset{k}{\text{argmax}}\{n_{ij}(k)\}$

O critério de decisão baseia-se nestas informações. Se, por exemplo, $h_1(\boldsymbol{x}) = i$ e $h_2(\boldsymbol{x}) = j$, consulta-se a unidade focal (i, j) e realiza-se a seguinte decisão

$$y^* = \begin{cases} R_{ij} & \text{se } T_{ij} > 0 \text{ e } \frac{n_{ij}(R_{ij})}{T_{ij}} > \lambda \quad \lambda \in (0,1) \\ \text{desconhecido} & \text{caso contrário} \end{cases}$$

Naturalmente o método pode ser aplicado a conjuntos com mais de dois classificadores sem perda de generalidade. A utilização de dois classificadores neste exemplo foi adotada para facilitar a exposição da idéia. De um modo geral, considerando um conjunto de m classificadores, a regra adotada é

$$y^* = \begin{cases} R_{1,\ldots,m} & \text{se } T_{1,\ldots,m} > 0 \text{ e } \frac{n_{1,\ldots,m}(R_{1,\ldots,m})}{T_{1,\ldots,m}} > \lambda, \quad \lambda \in (0,1) \\ \text{desconhecido} & \text{caso contrário} \end{cases}$$

Ao propor o método, os autores ainda apresentam uma forma de obter o limiar (λ) e realizam uma análise estatística do critério de decisão adotado.

Redes Bayesianas como combinadores de classificadores (Garg et al. [43])

Os métodos desenvolvidos por Xu e Krzyżak [116] e Huang e Suen [58] empregam um método estatístico para estimação da densidade no espaço de saída dos classificadores. Segundo Webb [115], podem-se ser utilizadas novas abordagens para estimar esta densidade, dentre elas o emprego de redes Bayesianas. O trabalho desenvolvido por Garg *et al.* [43] segue esta linha. Os autores desenvolveram uma rede Bayesiana de classificadores com uma topologia bastante simples, trata-se de fato de um classificador Bayesiano puro (Capítulo 2).

Em seu artigo, Garg *et al.* [43] provaram que a probabilidade da rede realizar um erro é sempre menor ou igual ao valor mínimo da probabilidade de algum classificador errar, o que matematicamente pode ser expresso como

$$p(h^*(\boldsymbol{x}_i) \neq y_i) \leq \min_j \{p(h_j(\boldsymbol{x}_i) \neq y_i\}$$

sendo que $\boldsymbol{x}_i, y_i$ correspondem respectivamente à entrada e saída desejada, $h^*(\cdot)$ e $h_j(\cdot)$, à classificação gerada pela rede e pelo j-ésimo classificador, respectivamente.

Garg *et al.* [43] mostraram que o classificador Bayesiano puro possui uma árvore de decisão equivalente, cujo número de nós de decisão, nós folhas, cresce exponencialmente com a profundidade da árvore. Para simlificar a estrutura do combinador, os autores propuseram um processo de poda baseado em uma ordenação dos classificadores quanto ao erro de predição. Este processo

origina a obtenção de árvores de decisão substancialmente mais simples que, embora não sejam ótimas, por não serem equivalentes à rede que as originou, experimentalmente apresentaram desempenho promissor quando aplicadas à classe de problemas investigado pelos autores.

Outros métodos

Além das classes de métodos relacionados nesta seção, outras formas de combinação baseadas em um processo de treinamento do combinador foram propostas ao longo dos últimos dez anos, dentre estas vale salientar os métodos baseados em lógica nebulosa (Cho e Kim [19] e [20]) e métodos basedos no formalismo de Dempster e Shaffer (Xu e Krzyżak [116], Rogova [95]).

3.3 Conclusão

As discussões e revisões abordadas neste capítulo apresentam combinação de classificadores como uma forma de melhorar a predição geral de um sistema de classificação. Os principais problemas relacionados com o aprendizado de classificadores foram apresentados na Seção 3.1. Mostrou-se que o aprendizado de classificadores obedece a um princípio, denominado princípio de Occam (Johannes [62]), o qual diz que não se deve introduzir complexidade extra além do necessário, isto é, deve-se procurar um balanço entre a complexidade do problema e a do algoritmo para solucioná-lo. O algoritmo não deve ser tão simples que não seja capaz de lidar com a complexidade dos dados envolvidos (viés elevado), nem tão complexo a ponto de realizar uma superespecialização dos dados de treinamento (variância elevada), Figura 3.7. A combinação de classificadores contribui para encontrar uma solução de compromisso entre viés e variância distribuindo a complexidade do método de aprendizado entre vários classificadores.

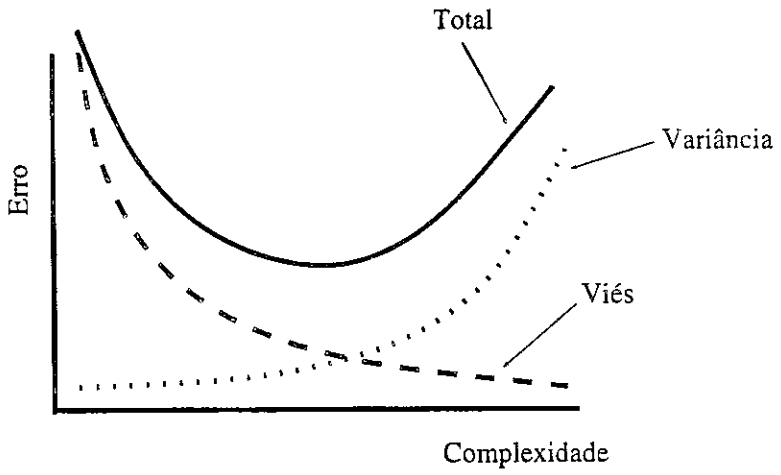


Figura 3.7: Dilema viés-variância (adaptado de Johannes [62])

No Capítulo 4 será discutido em maior profundidade a idéia do método proposto, que con-

siste em obter um agrupamento em que os classificadores são adicionados progressivamente de acordo com a complexidade do problema. Desta forma procura-se encontrar um equilíbrio entre a complexidade do problema e do sistema de classificação. Os classificadores obtidos são então combinados por meio de uma rede Bayesiana.

Capítulo 4

Método proposto – teoria

Este capítulo apresenta o método de combinação de classificadores utilizando redes Bayesianas. Realiza-se uma abordagem de alto nível focada nos aspectos teóricos envolvidos. A abordagem destes assuntos se desenvolve em duas partes principais: na descrição do método de particionamento do espaço de atributos, que consiste na fase de criação de novos classificadores, e na obtenção da rede Bayesiana, criada para combinar suas saídas a fim de gerar uma predição única. Apresenta-se um método de obtenção da rede Bayesiana adaptado para o tipo de problema em foco e discute-se duas diferentes variações dos algoritmos de inferência e aprendizado, o que gera quatro diferentes variações do método de combinação de classificadores proposto.

O capítulo está organizado da seguinte forma: a Seção 4.1 discute o algoritmo de segmentação do espaço de atributos; a Seção 4.2 discorre sobre o sistema de tomada de decisão, que consiste na utilização de uma rede Bayesiana como um combinador baseado em treinamento e na Seção 4.3 encerra-se o capítulo com conclusões e discussões gerais.

4.1 Segmentação do espaço de atributos

Alguns métodos conhecidos como CART (Breiman *et al.* [11]), MARS (Friedman [39]) e mistura de especialistas (Jordan and Jacobs [65]) são baseados em um particionamento recursivo do espaço de atributos. O princípio comum que apoia estes métodos é o de dividir para conquistar. Dividir para conquistar é uma técnica que procura resolver um problema complexo pela sua decomposição em instâncias menores, resolvendo-as sucessiva e independentemente e combinando adequadamente as soluções parciais para obter a solução do problema original.

Este capítulo apresenta um método para particionar recursivamente o espaço de atributos que, tal como na mistura de especialistas, promove um particionamento suave (*soft partitioning*), isto é, as partições criadas possuem um pequeno entrelaçamento de tal modo que um dado padrão pode pertencer a mais de uma partição. O esquema de particionamento pode ser descrito brevemente como a seguir: primeiro um classificador é treinado com todo os elementos do

conjunto de treinamento; se os padrões são aprendidos apropriadamente o processo se encerra; caso contrário, novos classificadore são criados para reconhecer o espaço nas vizinhanças dos padrões não aprendidos. Se alguns dos conjuntos usados para treinar os novos classificadores não são aprendidos apropriadamente, os mesmos serão particionados. Este processo é repetido recursivamente até que nenhum particionamento possa ser realizado. A Figura 4.1 ilustra como este processo pode ser realizado, admitindo-se que o espaço de atributos seja um subconjunto de $\mathbb{R}^2$ e que o classificador global seja linear.

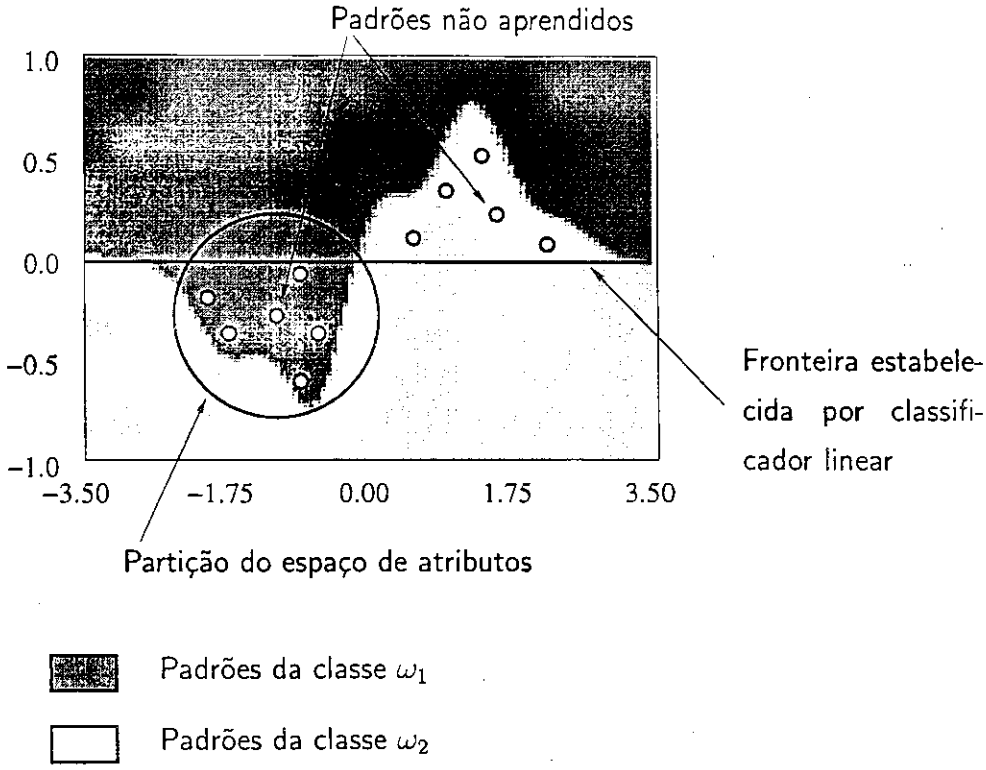


Figura 4.1: Simulação de particionamento em $\mathbb{R}^p \subset \mathbb{R}^2$

O treinamento dos classificadores acima descrito assemelha-se ao realizado em métodos de reamostragem do conjunto de treinamento denominados "*Arcing*" (*Adaptive Resampling and Reweighting*) (Duda e Hart [33]), que procuram fazer com que classificadores baseados no mesmo espaço de atributos difiram entre si treinando-os com subconjuntos diferentes. O método *Boosting* (Freund e Schapire [36]), por exemplo, também estabelece que novos classificadores sejam treinados com padrões não aprendidos pelo classificador anterior. Entretanto, ao contrário deste, o método aqui proposto não é limitado a um número pré-estabelecido de classificadores e, além disto, define efetivamente um conjunto de partições que caracterizam o espaço em que estes classificadores são treinados e testados.

Diferentemente do que ocorre em métodos como CART e MARS, as partições obtidas não cobrem coletivamente todo o espaço de atributos mas apenas os volumes com maior erro de

classificação, já que elas são vizinhanças em torno de padrões não aprendidos.

Por conveniência, a notação empregada neste texto aproveitará parte daquela empregada na literatura que descreve métodos que realizam particionamento de $\mathbb{R}^p$, tal como Friedman [40] e Peng e Bhanu [89]).

Definição 10 (Partição) *Uma partição, denotada por $R_m \subset R^p$ é uma região caracterizada por sua forma $s_m(\alpha)$, expressa pela Equação (4.1), e pelo seu centro u_m.*

$$s_m(\alpha) = ave_{x \in R_m}(|\alpha^t(x - u_m)|), \tag{4.1}$$

sendo α um vetor unitário em R^p e $ave(\cdot)$ a função média.

A expressão (4.1) é uma representação genérica para a forma de uma partição. Sua forma é mais expandida ou mais contraída em uma direção α se, na média, a projeção de $x - u_m$ nesta direção for mais ou menos acentuada.

No método proposto, a forma de R_m deve ser regida pela função de predição $f(x)$, sendo mais larga onde $f(x)$ e sua aproximação $h(x)$, gerada pelo classificador, discordam mais intensamente. A adoção deste princípio implica na criação de novos classificadores especializados em resolver os casos não aprendidos pelo classificador original. Por serem treinados em uma vizinhança do espaço de atributos, o erro entre $h(x)$ e $f(x)$ para cada classificador local deve ser inferior àquele calculado para o classificador original. Nas seções seguintes é apresentada em maiores detalhes a estratégia usada para criar partições pequenas e que ao mesmo tempo favoreçam a criação de um combinador preciso e robusto, isto é, que possua uma elevada taxa de reconhecimento, e seja pouco sensível à variabilidade amostral existente no conjunto de treinamento. As seções 4.1.1 e 4.1.2 explicam respectivamente como obter u_m e a forma de R_m e a Seção 4.1.3 apresenta o algoritmo de particionamento proposto.

4.1.1 Obtenção de u_m

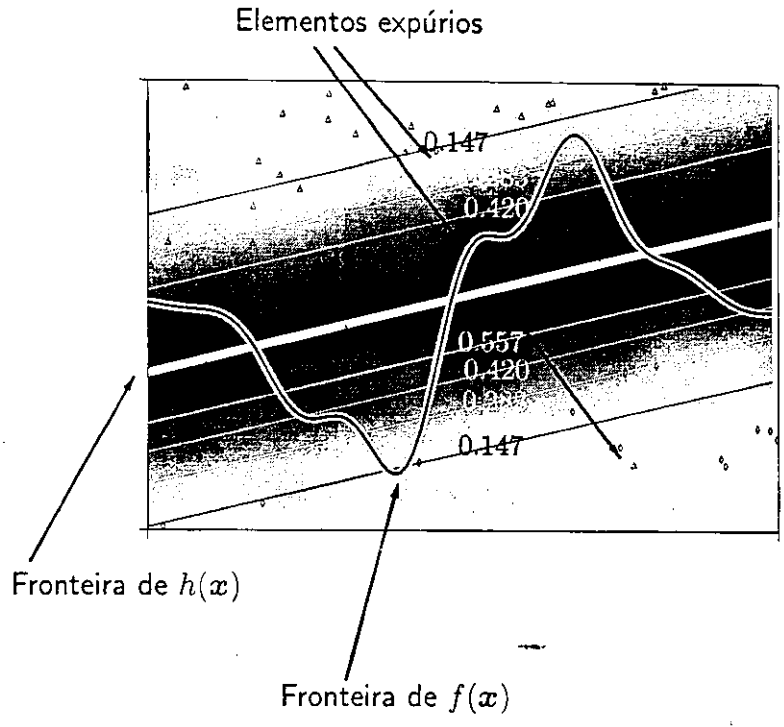
Como as partições são criadas sobre os padrões não aprendidos pelo classificador original, u_m deve escolhido em uma vizinhança com a maior quantidade possível destes padrões. Naturalmente, qualquer procedimento de busca envolvendo uma contagem por padrões em uma dada vizinhança implica em um elevado custo computacional, o que justifica a adoção de uma solução heurística. Admitindo-se que as maiores concentrações de padrões não aprendidos estejam localizadas nas imediações das fronteiras entre as regiões discriminantes de cada classe, a escolha de u_m deve favorecer à escolha de padrões próximos a uma fronteira. Como é requerido que as saídas dos classificadores sejam probabilidades, as regiões de fronteira, que são regiões associadas a uma classificação dúbia, são bem caracterizadas pela entropia no espaço de saída dos classifi-

cadores. Portanto, a obtenção de $\boldsymbol{u_m}$ pode ser estabelecida pela seguinte regra de maximização da entropia:

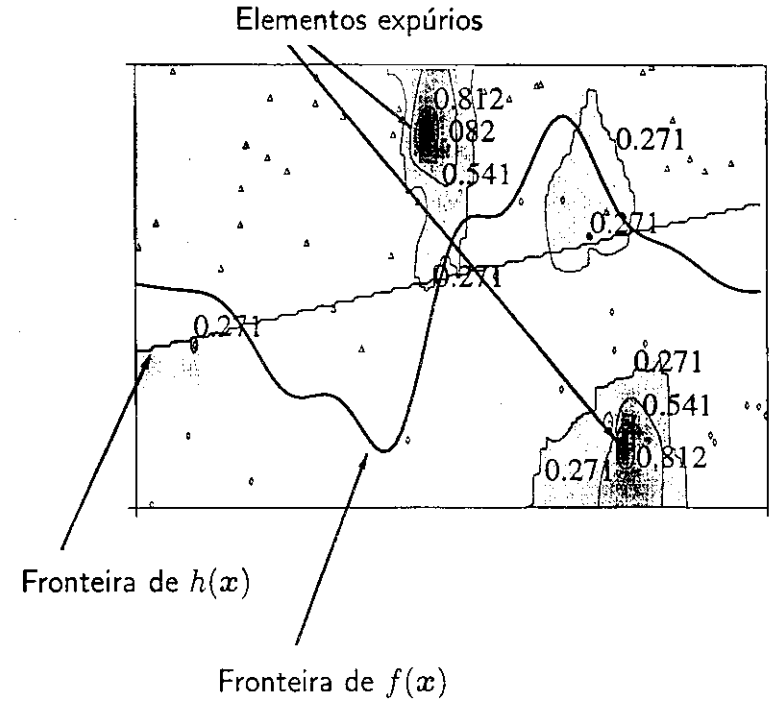
$$\boldsymbol{u_m} = \underset{\boldsymbol{x} \in \mathcal{M}}{\operatorname{argmax}} \{ - \sum_i h_i(\boldsymbol{x}) \ln h_i(\boldsymbol{x}) \} \tag{4.2}$$

sendo $\mathcal{M}$ o conjunto de padrões não aprendidos pelo classificador original.

A escolha do elemento de máxima entropia pode ser executada rapidamente e tende a originar partições com um número elevado de padrões não aprendidos pelo primeiro classificador, como avaliado experimentalmente em testes discutidos na Seção 5.3. Alternativamente, outro critério, como o erro médio quadrático, pode ser usado ao invés da entropia, com a desvantagem de poder conduzir à seleção de elementos expúrios (*outliers*). Um elemento expúrio, caracterizado por ser uma ocorrência isolada e deslocada do centro da classe à qual pertence, possui elevado erro médio quadrático. Entretanto, por estar distante da fronteira, tende a possuir baixa entropia. A entropia, por conseguinte, possui a propriedade desejável de ser pouco suscetível à influência de elementos expúrios. Esta propriedade está ilustrada nas Figuras 4.2(a) e 4.2(b). Neste exemplo, construído com dados simulados, $\mathbb{R}^p \subset \mathbb{R}^2$ e $h(\boldsymbol{x})$ é uma função discriminante linear. Observe que os elementos de mais alta entropia, por estarem localizados nas imediações das fronteiras de separação, tendem a estar próximos a um número maior de padrões não aprendidos, o que não se verifica com a superfície gerada pelo erro médio quadrático, que possui picos nas imediações de elementos expúrios.



(a) Isolinhas de distribuição da entropia



(b) Isolinhas de distribuição do erro médio quadrático

Figura 4.2: Distribuições da entropia e do erro médio quadrático no espaço de atributos

4.1.2 Identificação de $s_m(\alpha)$

A forma de R_m é tanto melhor quanto mais eficiente for em identificar os padrões em torno de u_m que necessitem ser submetidos a um novo ciclo de treinamento. Em um espaço de grande dimensão mesmo que u_m esteja localizado próximo a muitos padrões não aprendidos, ele também pode se localizar próximo a muitos que tenham sido aprendidos, pois o espaço ao seu redor se prolonga em muitas direções. É desejável que $s_m(\alpha)$ se estenda somente nas direções dos padrões não-aprendidos. Para identificar estas direções utilizou-se o erro médio quadrático que, como demonstrado nos Lemas 1 e 2 a seguir, possui a propriedade de identificar se um padrão foi ou não classificado corretamente quando a saída do classificador é uma aproximação de uma distribuição de probabilidades.

Lema 1 *Seja* $h(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ *um classificador que aproxime uma distribuição de probabilidades, seja* $\boldsymbol{y} = h(\boldsymbol{x})$ *a saída produzida, isto é,* $\boldsymbol{y} =< y_1, \ldots, y_m >$ *com* $y_i \geq 0$ *e* $\sum_{i=1}^{m} y_i = 1$ *e seja* $t \in \mathbb{R}^m$ *a saída desejada, então, tem-se que um padrão* $\boldsymbol{x}$ *é corretamente classificado sempre que* $mse(\boldsymbol{x}) < \frac{0,5}{m}$.

Prova:

Por conveniência, assuma que a classe de $\boldsymbol{x}$ seja ω_m então, como se trata de um problema de classificação, o vetor t possui todas as componentes nulas exceto a última em que $t_m = 1$, $t =< 0, \ldots, 0, 1 >$. Observe que $y_m > 0,5$ implica que $\boldsymbol{x}$ foi classificado corretamente pois por hipótese $\sum_{i=1}^{m} y_i = 1$, com $y_i > 0$, $i = 1, \ldots, m$. Em outras palavras, se $y_m < 0,5$ o padrão $\boldsymbol{x}$ pode ter sido classificado corretamente ou não, entretanto, se $y_m > 0,5$ seguramente a classificação terá sido correta.

Considerando que $\boldsymbol{x}$ tenha sido corretamente classificado então tem-se que

$$\begin{aligned} mse(\boldsymbol{x}) &= \frac{1}{m}\sum_{i=1}^{m}(t_i - y_i)^2 \\ &= \frac{1}{m}\sum_{i=1}^{m-1} y_i^2 + (1 - y_m)^2, \text{ com } y_m > 0,5 \end{aligned} \tag{4.3}$$

O vetor y que maximiza (4.3) possui a m-ésima componente igual a $0,5$ pois para qualquer outro valor os dois termos somados em (4.3) são simultaneamente minimizados. Portanto, para obter uma expressão que maximize o erro médio quadrático quando $\boldsymbol{x}$ é classificado sem erro é necessário obter um vetor y tal que o primeiro termo em (4.3) seja maximizado sujeito a restrição $\sum_{i=1}^{m-1} y_i = 0,5$. Isto é, deve-se resolver:

$$\max \sum_{i=1}^{m-1} y_i^2$$

s.a.

$$\sum_{i=1}^{m-1} y_i = 0,5 \tag{4.4}$$

$$\boldsymbol{y} \geq 0$$

Observe que, como $\boldsymbol{y} \geq 0$, então $\sum y_i^2 \leq (\sum y_i)^2$ pois

$$\left(\sum y_i\right)^2 = \sum_i y_i^2 + 2\sum_i \sum_{j>i} y_i y_j \tag{4.5}$$

$$\geq \sum y_i^2 \tag{4.6}$$

Dado que $(\sum_{i=1}^{m-1} y_i)^2 = 0,5^2$ então a igualdade em (4.6) ocorre somente se $\exists y_i = 0,5;\ i = 1 \ldots m-1$, logo, o problema possui mais de uma solução ótima. Entretanto o valor máximo do erro médio quadrático para um padrão corretamente classificado é único e igual a:

$$\max_{\{\boldsymbol{x}|cls(\boldsymbol{x})=\omega_m\}} mse(\boldsymbol{x}) = \frac{1}{m} \left\| \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0,5 \\ \vdots \\ 0 \\ 0,5 \end{bmatrix} \right\|^2 = \frac{2(0,5)^2}{m} = \frac{0,5}{m}, \tag{4.7}$$

em que $cls(\boldsymbol{x})$ denota a classe de um padrão $\boldsymbol{x}$ e $\|\cdot\|$ a norma de um vetor. □

Lema 2 *Seja $h(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ um classificador que aproxime uma distribuição de probabilidades, seja $\boldsymbol{y} = h(\boldsymbol{x})$ a saída produzida, isto é, $\boldsymbol{y} =< y_1, \ldots, y_m >$ com $y_i \geq 0$ e $\sum_{i=1}^m y_i = 1$ e seja $\boldsymbol{t} \in \mathbb{R}^m$ a saída desejada, então, tem-se que um padrão $\boldsymbol{x}$ é classificado com erro sempre que $mse(\boldsymbol{x}) > \frac{1}{m}$.*

Prova: Considerando a notação empregada na prova do Lema 1, pode-se afirmar que $\boldsymbol{x}$ é classificado com erro sempre que $y_m < 1/m$. Então, dado que $\boldsymbol{x}$ tenha sido classificado incorretamente, o erro médio quadrático obtido é dado por

$$mse(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m}(t_i - y_i)^2$$

$$= \frac{1}{m}\sum_{i=1}^{m-1} y_i^2 + (1 - y_m)^2, \text{ com } y_m < \frac{1}{m} \tag{4.8}$$

Observe que (4.8) é minimizado quando $y_m = 1/m$ pois, para qualquer outro valor, os dois termos somados são simultaneamente maximizados. Portanto as componentes do vetor $\boldsymbol{y}$ que minimizam o erro médio quadrático quando $\boldsymbol{x}$ é classificado incorretamente satisfaz

$$\begin{aligned} &\min \sum_{i=1}^{m-1} y_i^2 \\ &\text{s.a.} \\ &\sum_{i=1}^{m-1} y_i = 0,5 \\ &\boldsymbol{y} \geq 0 \end{aligned} \tag{4.9}$$

Considere que $y_i = k + \epsilon_i$, $i = 1, \ldots, m-1$, $0 < k < 1$ constante e $\sum \epsilon_i = 0$. Segue-se que

$$\begin{aligned} \sum y_i^2 &= (k+\epsilon_1)^2 + \ldots (k+\epsilon_{m-1})^2 && (4.10) \\ &= \sum k^2 + \sum 2k\epsilon_i + \sum \epsilon_i^2 && (4.11) \\ &= \sum k^2 + \sum \epsilon_i^2 && (4.12) \\ &\geq \sum k^2 && (4.13) \end{aligned}$$

Portanto a solução que minimiza $\sum y_i^2$ sujeito às restrições do problema (4.9) é única e dada por $y_1 = \ldots = y_{m-1}$. Como as componentes $y_1 \ldots y_{m-1}$ são iguais e $y_m = \frac{1}{m}$ e como por hipótese $\sum y_i = 1$ tem-se que $\boldsymbol{y} =< y_i >$, $y_i = \frac{1}{m}, i = 1 \ldots m$. Para este caso o de $mse(\boldsymbol{x})$ é dado por

$$\min_{\{\boldsymbol{x}|cls(\boldsymbol{x}) \neq \omega_m\}} mse(\boldsymbol{x}) = \frac{1}{m} \left\| \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{m} \\ \vdots \\ \frac{1}{m} \\ \vdots \\ \frac{1}{m} \\ \frac{1}{m} \end{bmatrix} \right\|^2 = \frac{1}{m} \frac{m-1}{m} \tag{4.14}$$

O limiar $\frac{1}{m}$ pode ser adotado como caso limite, isto é, $\lim_{m \to \infty} \min_{\{\boldsymbol{x}|cls\{\boldsymbol{x}\} \neq \omega_m\}} mse(\boldsymbol{x}) = \frac{1}{m}$

□

Com base na propriedade apresentada no Lema 1, a estratégia para delinear a forma de R_m pode ser estabelecida como:

$$R_m = \{\boldsymbol{x} | \quad \|\boldsymbol{x} - \boldsymbol{u_m}\| \leq r \wedge mse(\boldsymbol{x}) > \frac{0.5}{n}\} \tag{4.15}$$

sendo r o raio de uma hiperesfera em $\mathbb{R}^p$ com centro em $\boldsymbol{u_m}$ usada para definir uma vizinhança. Nos experimentos realizados o valor de r foi estabelecido como sendo a distância média de $\boldsymbol{u_m}$ a $\boldsymbol{x} \in \mathcal{M}$, em que $\mathcal{M}$ denota o conjunto de padrões não aprendidos em R. Este procedimento favorece a ocorrência de um entrelaçamento entre fronteiras de partições adjacentes.

O critério do erro médio quadrático origina uma regra simples e eficiente para definir a forma de R_m, contribuindo para uma rápida convergência do processo de treinamento por isolar efetivamente os padrões que necessitam de um novo ciclo de treinamento para serem aprendidos. Entretanto, o erro médio quadrático é bastante suscetível à presença de elementos expúrios, o que favorece a criação de partições com prolongamentos indesejáveis. Uma forma de minimizar este problema consiste em relaxar o critério de inclusão de um padrão em R_m, dado pela Equação (4.15), admitindo-se um número maior de padrões já aprendidos nas novas partições criadas.

4.1.3 Algoritmo de particionamento

O algoritmo de particionamento é apresentado como

Algoritmo 1 (Particionamento do espaço de atributos)

$Particionar(R, h)$
  R : *Região*
  h : *Classificador (Hipótese)*
  $\mathcal{M}$: *Conjunto de exemplos não-aprendidos*
$\mathcal{M} \leftarrow \{\boldsymbol{x} | h(\boldsymbol{x}) \neq cls(\boldsymbol{x})\}$
enquanto $\mathcal{M} \neq \emptyset$
  $\boldsymbol{u_m} = \underset{\boldsymbol{x} \in \mathcal{M}}{\operatorname{argmax}}\{-\sum_i h_i(\boldsymbol{x}) \ln h_i(\boldsymbol{x})\}$
  para todo $\boldsymbol{x} \in R$
    se $(\|\boldsymbol{x} - \boldsymbol{u_m}\| \leq r \wedge mse(\boldsymbol{x}) \geq \tau)$ **então** $\quad R_m \leftarrow R_m \cup \{\boldsymbol{x}\}$
  $\mathcal{M} \leftarrow \mathcal{M} \setminus R_m$

Observe que o particionamento proposto origina uma árvore, de modo similar ao que ocorre com métodos como CART e mistura hierárquica de especialistas. Entretanto, diferentemente daqueles, o número de filhos de cada nó não é uma constante. Na árvore originada deste particionamento, Figura 4.3, chamada diagrama de classificadores, cada nó corresponde a uma região e a um classificador distintos. O classificador global associa-se ao nó raiz e os classificadores locais associam-se aos nós descendentes. Na Seção 4.2 será discutida a construção do sistema de tomada de decisão a partir do diagrama de classificadores.

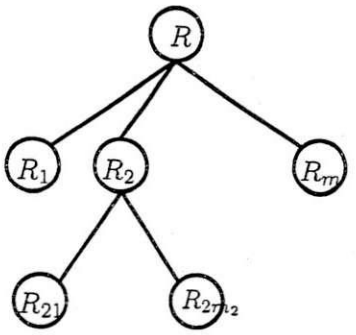


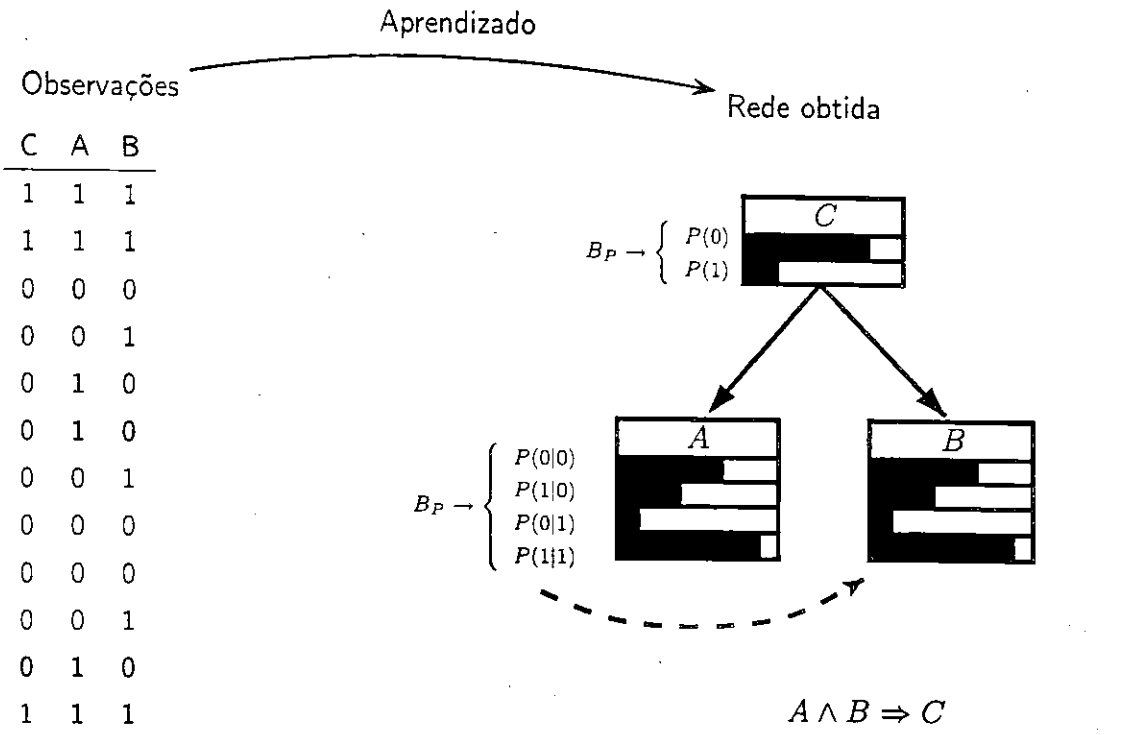
Figura 4.3: Diagrama de classificadores

4.2 O Sistema de tomada de decisão

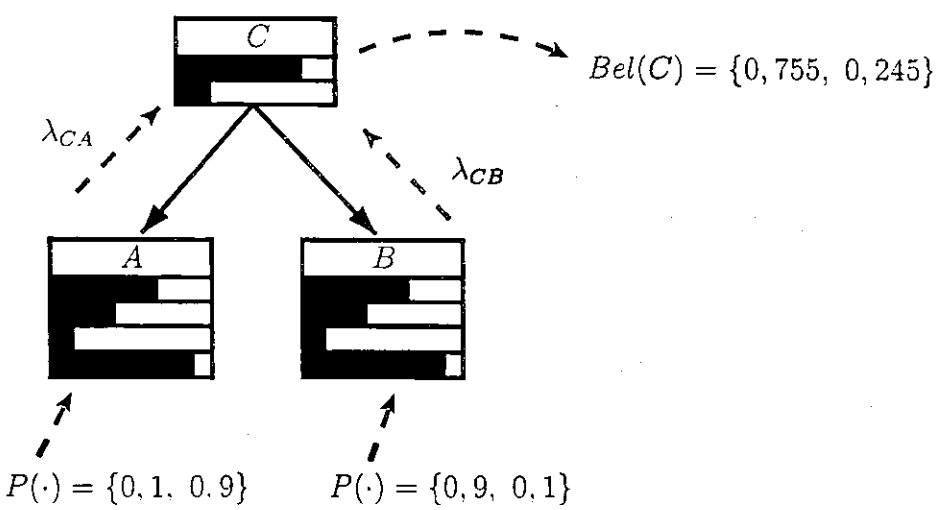
O sistema de tomada de decisão é de fato um combinador de classificadores que funde as predições elaboradas por classificadores locais e global em uma predição única. Este combinador é uma rede Bayesiana obtida a partir do diagrama de classificadores originado pelo processo de particionamento do espaço de atributos. Nas Seções 4.2.1, 4.2.2 e 4.2.3 apresentam-se procedimentos para realizar o aprendizado da estrutura e parâmetros (probabilidades condicionais) e realização de inferência de uma rede Bayesiana orientada em função do problema.

4.2.1 Obtenção da estrutura da rede

Uma rede Bayesiana pode ser vista como um sistema especialista em que as regras da base de conhecimentos são definidas no grafo subjacente. Uma implicação do tipo SE $A \wedge B$ ENTÃO C pode ser associada a um grafo como o ilustrado na Figura 4.4. Neste caso, os valores verdade de A e B, quantificados numa escala contínua entre 0 e 1, contribuem para formação do valor verdade de C. A intensidade com que estas quantidades se somam é ponderada pelas probabilidades condicionais envolvendo os nós pai e filhos. Esta analogia pode ser facilmente transportada para o problema de combinação de classificadores. Um nó tomador de decisão combina as contribuições provenientes de nós folhas, representando classificadores, ponderando-as pelas probabilidades condicionais que são aprendidas com base em um histórico de casos observados.



(a) Aprendizado



(b) Inferência

Figura 4.4: Implementação de uma regra de decisão em uma rede Bayesiana

O diagrama de classificadores gerado na etapa de particionamento do espaço de atributos orienta a aquisição de B_S. Os classificadores, que fornecem uma estimativa individualizada sobre a classificação de um padrão, são associados a nós folha. Os nós de combinação, que sumarizam os resultados de vários classificadores, são associados a nós não-terminais. A fim de preservar a estrutura hierárquica existente no diagrama de classificadores pode-se fazer a introdução de um nó não-terminal para cada nível do diagrama. Este procedimento leva à geração de uma rede Bayesiana que é na realidade uma árvore, formadada pela introdução de nós não-terminais a cada

nível do diagrama de classificadores, Figura 4.5.

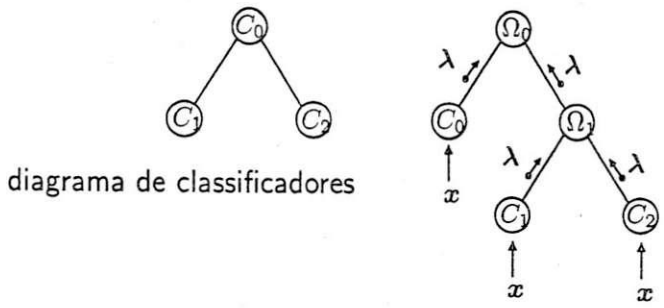


Figura 4.5: Construção da rede Bayesiana

4.2.2 Obtenção das probabilidades condicionais

Uma vez obtida a estrutura da rede, as relações de adjacência passam a ser conhecidas, permitindo a estimação das probabilidades condicionais. Estas probabilidades são estimadas com base em valores observados em uma base de casos contendo ocorrências simultâneas de instâncias de nós filho e pai. Como o sistema é formado por um conjunto de classificadores localmente especializados, para um dado padrão de treinamento apenas algumas observações são realizadas, já que o classificador não avalia um padrão fora da partição em que está definido. A base de observações, portanto, é incompleta. Os métodos estuados no Capítulo 2 para tratar com incompletude são caros computacionalmente e por esta razão não se aplicam ao problema particular tratado neste trabalho, pois a ocorrência de valores não observados é muito freqüênte em todas as tuplas da base de casos. Adotou-se portanto uma solução heurística. Uma vez que os classificadores mais especializados são mais precisos, já que foram criados para resolver os casos não aprendidos pelo seu ancestral, ao realizar a classificação de um padrão de teste, a contribuição destes classificadores deve ser mais influente na tomada de decisão final. Na prática este princípio é implementado fazendo com que as matrizes de probabilidades condicionais dos nós mais especializados possua baixa entropia. A fim de obter B_P com esta característica, o preenchimento das observações também foi feito com os valores dos objetivos. Esta estratégia certamente favorece a obtenção de B_P com baixa entropia nos nós mais especializados visto que a observação associada ao nó ancestral, que é necessariamente um nó de decisão, também é atribuída ao objetivo do padrão de treinamento.

Como apresentado no Capítulo 2, a estimação de $\theta_{ijk} = p(x_{ik}|\boldsymbol{pa}_{ij}, D)$, também chamado parâmetro da rede Bayesiana, isto é, a probabilidade de que $X_i = x_{ik}$ dado que o estado de seu ancestral seja $\boldsymbol{Pa}_i = \boldsymbol{pa}_{ij}$, quando há disponível uma base de casos completa D e sob a hipótese de que Θ seja $Dirichlet(\boldsymbol{\theta})$ com parâmetros $\nu_1, \ldots, \nu_{n-1}$ é dada por

$$\theta_{ijk} = p(x_{ik}|\boldsymbol{pa}_{ij}, D) = \frac{\nu_k + s_k}{N + M} \tag{4.16}$$

sendo $N = \sum_k \nu_k$, e s_k igual ao número de ocorrências de $\boldsymbol{pa}_{ij}$ entre M tuplas observáveis em D.

A Equação (4.16) é usada para calcular os parâmetros de uma rede Bayesiana qualquer. Neste trabalho apresenta-se um procedimento ligeiramente diferente para calcular θ_{ijk} quando x_i é um nó associado a um classificador, isto é um nó folha, visto que neste caso pode-se tratar com valores amostrados diretamente da distribuição de Θ. Esta adaptação do procedimento de aprendizado de B_P está comentada a seguir.

Sejam X_1 e X_2 variáveis aleatórias e $\boldsymbol{Pa_2} = X_1$. Para simplificar a notação, assuma sem perda de generalidade que X_1 e X_2 sejam variáveis binárias. Se X_2 é um nó folha, a base de casos conterá observações sobre instâncias de X_1 e da probabilidade de X_2 dado X_1, já que um nó folha associa-se a um classificador. Considere a representação para esta base de casos ilustrada na Figura 4.6.

Rede Bayesiana

Base de casos

$$D = \begin{bmatrix} x_1^1 & p^1(x_{21}|x_1^1) & p^1(x_{22}|x_1^1) \\ x_1^2 & p^2(x_{21}|x_1^2) & p^2(x_{22}|x_1^2) \\ & \vdots & \\ x_1^M & p^M(x_{21}|x_1^M) & p^M(x_{22}|x_1^M) \end{bmatrix}$$

Figura 4.6: Representação matemática da base de casos de uma rede Bayesiana com dois nós

Assumindo que Θ_2 seja $beta(\theta; \nu_1)$ [1] e que θ_{2jk} seja o limite de uma uma freqüência relativa com uma contagem infinitamente longa, isto é, chamando m_{kj}^c a freqüência em que $X_2 = x_{2k}$ e $X_1 = x_{1j}$ na c-ésima tupla da base de casos, então

$$p^c(x_{2k}|x_{1j}) = \lim_{m \to \infty} \frac{m_{kj}^c}{m} \tag{4.17}$$

com $m = m_{1j}^c + m_{2j}^c$. De acordo com esta notação a base de casos pode ser reescrita como

[1] A distribuição *beta* é um caso particular da distribuição *Dirichlet*, ocorre quando o número de argumentos da distribuição *Dirichlet* é igual a dois

$$D = \begin{bmatrix} x_{11} & m_{11}^1 & m_{12}^1 \\ x_{11} & m_{11}^2 & m_{12}^2 \\ & \vdots & \\ x_{11} & m_{11}^{M_1} & m_{12}^{M_1} \\ x_{12} & m_{21}^{M_1+1} & m_{22}^{M_1+1} \\ & \vdots & \\ x_{12} & m_{21}^{M_1+M_2} & m_{22}^{M_1+M_2} \end{bmatrix} \tag{4.18}$$

sendo M_k o número de tuplas em D em que $X_1 = x_{1k}$.

De (4.16) segue-se que

$$\begin{aligned} p(x_{2k}|x_{1j}) &= \lim_{m\to\infty} \frac{\nu_k + m_{kj}^1 + m_{kj}^2 + \ldots + m_{kj}^{M_k}}{N + m + m + \ldots + m} \\ &= \frac{1}{M_k} \sum p^i(x_{2k}|x_{1j}) \end{aligned} \tag{4.19}$$

A Equação (4.19), usada para calcular B_P nos nós terminais, mostra que o valor estimado de θ_{ijk} para estes nós pode ser aproximado pelo valor médio da freqüência relativa de $x_{ik}|\boldsymbol{pa}_{ij}$ em D.

4.2.3 Cálculo de inferência

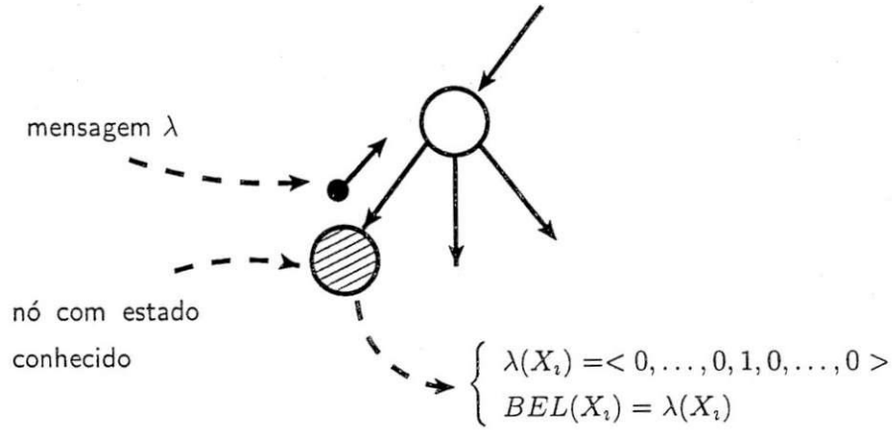
Neste trabalho, a estrutura da rede usada para realizar o agrupamento de classificadores é uma árvore, tratando-se, portanto, de um caso particular de um grafo unicamente conectado, o que torna possível a utilização de um algoritmo de inferência exato. Foi utilizado o algoritmo de propagação de probabilidades apresentado no Capítulo 2.

O cálculo da inferência corresponde à realização de um teste, após a fase de aprendizado ter sido concluída. Quando um padrão de teste é apresentado à rede, os nós associados às partições em que ele se localiza propagam mensagens λ que culminam na atualização do vetor de crença do nó raiz. O resultado de sua classificação é obtida pela avalização deste vetor, visto que o nó raiz é o nó de decisão de mais alto nível.

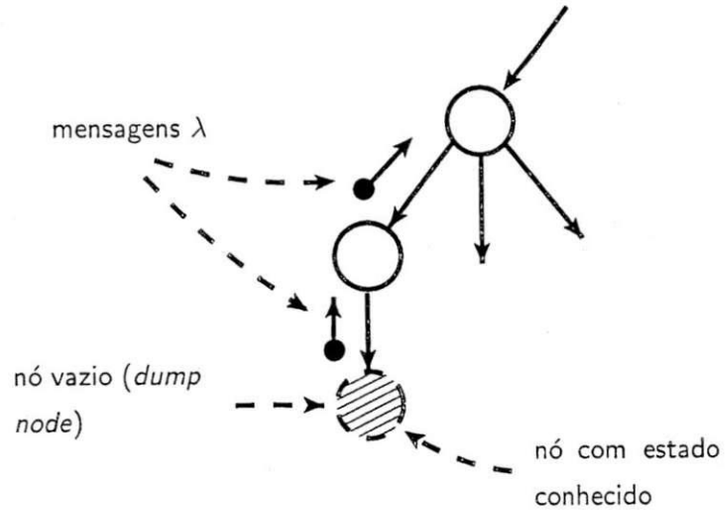
A introdução de informação na rede, que desencadeia o processo de inferência, pode ocorrer de duas formas possíveis: como valores discretos ou como probabilidades. No primeiro caso, a saída de um classificador é usada para modificar o vetor λ do nó terminal associado tornando todas componentes nulas, exceto aquela associada à classe com maior probabilidade. Esta abordagem tem o efeito de tornar o estado do nó conhecido, isto é $X_i = x_{ik}$, Figura 4.7(a), pois pela Equação (2.4) segue-se que

$$\lambda(X_i) =< 0, \ldots, 0, 1, 0, \ldots, 0 > \;\Rightarrow\; BEL(X_i) = \lambda(X_i) \tag{4.20}$$

No segundo caso, a saída do classificador é atribuída ao vetor λ do nó tal como ele é. O estado do nó, então, não passa a ser conhecido, entretanto, seu vetor de crenças é modificado e tende a reproduzir a mesma distribuição do vetor λ, desde que seu vetor π assegure probabilidades aproximadamente iguais para cada possível estado. Esta abordagem é equivalente à introdução de evidência em um nó vazio (*dump node*), Pearl [87]. Um nó vazio é instanciado e propaga para seu único ascendente uma mensagen λ igual à saída de um classificador. Ao receber esta mensagem o nó terminal recalcula seu vetor λ pela Equação (2.8), o que equivale a torná-lo igual à saída do classificador, visto que um nó terminal possui apenas um falso descendente: o nó vazio, Figura 4.7(b).



(a) Modelo discreto



(b) Modelo baseado em probabilidades

Figura 4.7: Modelos de inferência avaliados

A segunda abordagem lida com informação quantitativa do grau de confiança associado a cada um dos estados do nó. Esta informação é dominante quando a distribuição de B_P no nó

raiz é aproximadamente uniforme pois, neste caso, as diferenças entre as componentes do vetor λ contribuem fortemente para a formação do vetor de crença do nó. Esta propriedade é explorada em diversas situações práticas no Capítulo 5.

4.3 Conclusão

Este capítulo apresentou as principais idéias relacionadas com o método proposto, que são: o processo de particionamento do espaço de atributos e a combinação de classificadores por meio de uma rede Bayesiana. Na etapa de particionamento do espaço de atributos procurou-se tirar proveito da bem conhecida premissa de dividir para conquistar, que também apoiou diversos outros métodos na área de classificação de padrões. Um aspecto original da proposta está no fato de que o número de partições criadas não é estabelecido *a priori*, ele varia conforme o grau de dificuldade existente em se traçar as fronteiras de separação entre as classes. Outro aspecto original, pelo que se pôde observar no levantamento bibliográfico realizado, está na forma como se estabeleceu a forma das partições, que se entrelaçam e se extendem na direção dos padrões não-aprendidos. Métodos como *boosting* também são voltados aos padrões não-aprendidos mas, ao contrário deste, o método aqui proposto lida com um conceito de vizinhança de modo que se considera, além do resultado de uma classificação anterior, a proximidade em relação ao centro da partição.

O emprego de rede Bayesiana como agrupador de classificadores da forma como apresentada constitue também contribuição original do trabalho. O emprego do classificador Bayesiano, caso particular de uma rede Bayesiana, já havia sido investigado em trabalhos anteriores, como em Garg *et al.* [43]. Diferentemente do classificador Bayesiano, a rede que forma o método proposto possui uma estrutura mais complexa e não requer, por exemplo, que todos os nós associados a classificadores sejam instanciados, tendo em vista que as evidências apresentadas variam de acordo com o padrão de teste. Padrões localizados próximo às fronteiras entre as classes são passivos de serem classificados por mais de um classificador, logo instanciam mais de um nó. Padrões localizados em regiões distantes das fronteiras geram evidência em apenas um nó da rede.

Devido à estrutura particular da rede Bayesiana e ao tipo de problema abordado, os procedimentos de aprendizado e inferência puderam ser adaptados a fim de tratar as saídas dos classificadores como probabilidades, gerando duas variações do procedimento de aprendizado e duas variações do procedimento de inferência o que resulta em quatro variações do sistema de tomada de decisão como um todo.

Capítulo 5

Método proposto – avaliação

Este capítulo trata de aspectos práticos relacionados com o método proposto. Procura-se avaliar experimentalmente seu comportamento considerando diversos aspectos tais como: a utilização do erro médio quadrático ou da entropia como critério para definição de $s_m(\alpha)$, quais classificadores utilizar e como configurar seus parâmetros e a influência das adaptações apresentadas no Capítulo 4 sobre o procedimento de aprendizado e inferência da rede Bayesiana.

O estudo experimental se apoiou em algumas bases de padrões do repositório de aprendizado de máquina da Universidade da California em Irvine (Blake e Merz [8]). Estas bases são tradicionalmente utilizadas pela comunidade de aprendizado de máquina, apresentam diferentes graus de complexidade e possuem especificidades, como ser ou não uma base completa e guardar uma relação hieráquica implícita entre os atributos (o que a torna apropriada para métodos baseados em segmentação), dentre outras. Particularmente, procurou-se realizar uma avaliação do método proposto não direcionada a uma aplicação em particular. Esta avaliação mais ampla, contemplando um universo maior de aplicações, possibilitou identificar mais precisamente as vantagens e as limitações deste método.

O capítulo está organizado da seguinte forma: a Seção 5.1 descreve as bases de padrões utilizadas, a Seção 5.2 discorre sobre os classificadores que compõem o método e os parâmetros usados em sua construção, a Seção 5.3 avalia experimentalmente o processo de particionamento proposto, a Seção 5.4 realiza uma avaliação sobre o sistema de inferência e a Seção 5.5 encerra o capítulo com discussões e conclusões gerais.

5.1 As bases de padrões utilizadas

O repositório de aprendizado de máquina da Universidade da Califórnia em Irvine possui dezenas de bases de padrões, a maior parte com permissão de acesso pública, coletadas a partir de diversas fontes sendo associadas a muitos problemas distintos, tais como: reconhecimento de caracteres *on-line* e *off-line*, reconhecimento de casos clínicos, análise de crédito, jogos (como

Tabela 5.1: Dimensão das bases avaliadas

Base de padrões	Número de casos ($\times 10^3$)	Número de atributos	Número de classes
Adult	45, 22	14	2
Letter	20	16	26
Musk	6, 59	166	2
Nursery	12, 96	8	5
Pageblocks	5, 4	10	5
Pendigits	10, 99	16	10

xadrez ou jogo-da-velha) etc. Muitas destas bases são bastante pequenas, com apenas algumas centenas de instâncias para treinamento e testes, enquato outras, maiores, são incompletas. Os formatos adotados não são uniformes, há atributos nominais (ex. *proper*, *less_proper*, *improper*, *t*, *f*), atributos contínuos e discretos com intervalos de variação bastante irregulares. Apesar das disparidades existentes, todas contém padrões pré-processados e todos os atributos e classes são bem documentados, o que demanda apenas que seja realizada uma etapa preliminar para condicionar os dados ao formato de entrada do sistema. Portanto, não é necessário realizar extração de características de modo a melhorar a separabilidade entre as classes ou minimizar o número de atributos, visto que neste capítulo objetiva-se realizar exclusivamente uma avaliação do método de classificação proposto sem que exista preocupação com o pré-processamento do vetor de atributos, mas é necessário converter as entradas nominais em números e limitar a amplitude dos atributos numéricos a fim de serem utilizados por uma rede neural sem introduzir problemas de instabilidade numérica.

Ao selecionar as bases deste repositório procurou-se identificar aquelas com número de instâncias disponíveis maiores do que 4000 e menores do que 50000. A Tabela 5.1 apresenta as dimensões das bases utilizadas nos experimentos realizados neste capítulo, discutidas brevemente a seguir.

Adult Esta base contém dados demográficos extraídos do Censo do governo dos Estados Unidos. Há 45222 instâncias sem valores ausentes, das quais 30162 são usadas para treinamento e 15060 usadas para teste. Os conjuntos de treinamento e teste são disponibilizados em separado. O objetivo é classificar a renda anual de um indivíduo em uma das duas categorias: superior ou menor ou igual a US$ 50000, 00. Os atributos mensurados misturam valores nominais tais como sexo, estado civil, país de origem, etc., com valores contínuos como número de horas de trabalho por semana. A taxa de acerto relatada na documentação desta base situa-se em torno de 86%.

Letter Esta base é formada por 20000 vetores de características extraídas de imagens de caracteres manuscritos. Na documentação que acompanha a base sugere-se que sejam usadas as primeiras 16000 amostras para treinamento e as 4000 últimas para teste. As classes dos padrões associam-se às 26 letras do alfabeto inglês e os atributos a 16 características extraídas de imagens de varredura do caracter. Na documentação desta base há uma referência a um artigo publicado em 1991 (Frey e Slate [38]) que obteve uma taxa de acerto um pouco acima de 80%.

Musk Esta base contém 6598 padrões dos quais foram escolhidos aleatoriamente 80% para treinamento e 20% para teste. Os vetores de atributos são formados por 166 características relacionadas a mensurações de distância (em centésimo de Ângstron) usadas para medir moléculas. Os algoritmos de classificação devem classificar as medidas que formam o vetor de atributos em uma de duas classes possíveis, que são: musco e não-musco. Na documentação que acompanha a base, reporta-se uma taxa de acerto de 91%, em Singh [103] reporta-se uma taxa de reconhecimento de $93,56\%$.

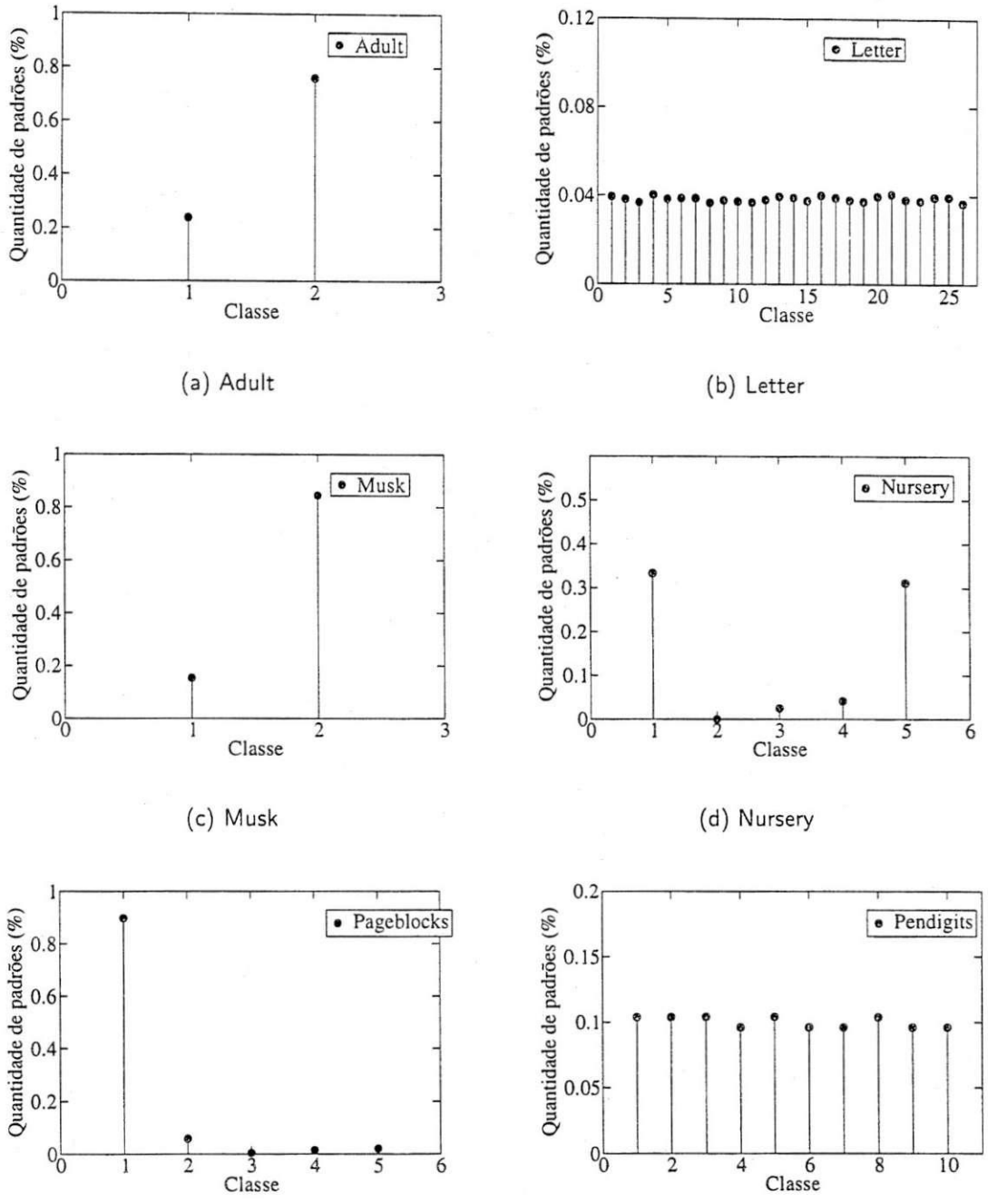
Nursery Esta base foi obtida a partir de questionário de um modelo de decisão hierárquico. Há oito atributos nominais, cinco classes e 12960 instâncias das quais escolheu-se aleatoriamente 10368 para treinamento e 2592 para teste. Em Singh [103] cita-se uma taxa de reconhecimento de $94,88\%$.

Pageblocks Base de padrões designada para testar diferentes métodos de simplificação para árvores de decisão. O problema consiste em identificar o conteúdo de um bloco extraído de uma imagem de documento. O bloco pode ser classificado em uma das cinco categorias: texto, traço vertical, traço horizontal, imagem ou gráfico. O conjunto de atributos envolve dez mensurações tais como altura, largura, percentual de pontos pretos, área, dentre outras. Há 5473 instâncias das quais se utilizou 4378 para treinamento e 1095 para teste, escolhidas aleatoriamente. Em Singh [103], foi obtido taxa de reconhecimento de $96,09\%$ para esta base.

Pendigits Esta base é formada por 10992 padrões distribuídos em dois conjuntos separados: um conjunto de treinamento formado por 7498 elementos e um conjunto de teste com 3498 elementos. O problema consiste em realizar a classificação de uma padrão obtido por mensurações da escrita $on-line$ de dígitos. Há 16 atributos inteiros variando na faixa $0 \ldots 100$ e 10 classes, correspondentes aos dígitos $0 \ldots 9$. Na documentação que acompanha a base reporta-se uma taxa de reconhecimento de $97,74\%$ obtida pela utilização do método $1-NN$ utilizando-se como medida de similaridade a distância euclidiana.

Verificou-se experimentalmente que a distribuição dos padrões por classes exerce alguma influência no desempenho do método proposto, sendo assim, a fim de subsidiar a interpretação dos

resultados que serão apresentados ao longo deste capítulo, as distribuições, em termos percentuais, dos padrões por classes em cada uma das bases avaliadas estão ilustradas na Figura 5.1.



(a) Adult

(b) Letter

(c) Musk

(d) Nursery

(e) Pageblocks

(f) Pendigits

Figura 5.1: Distribuição de padrões por classes (%)

5.2 Definição dos classificadores

Em razão do método proposto usar um combinador e diversos classificadores localmente especializados é razoável que o classificador global seja fraco, isto é, seja capaz de distinguir apenas aproximadamente as fronteiras de separação entre as regiões de decisão, bem como o ajuste fino seja realizado somente pelos classificadores localmente especializados. Sendo assim, a utilização de uma rede *Perceptron* como classificador global torna-se apropriada, tanto pela simplicidade de codificação e baixo custo de treinamento e teste quanto porque a não-linearidade do método será introduzida pelo combinador e pelos classificadores locais.

A definição dos classificadores locais é um problema mais complexo. Há uma grande variedade de abordagens que podem ser adotadas dado que existe uma grande variedade de classificadores, alguns essencialmente locais tais como k-NN e regressão logística local. Igualmente existem classificadores de propósito geral que podem ser usados como métodos de aprendizado local, como redes neurais (Bottou e Vapnik [9]). Além disto alguns classificadores são mais bem sucedidos do que outros para determinados tipos de problema, o que introduz um grau maior de dificuldade nesta tomada de decisão. Adotou-se uma linha que reduz a interferência ou poder de decisão de um especialista, tornando o método mais automático. Assim foram utilizados como classificadores locais redes *perceptron* ou classificadores k-NN, independendentemente do problema a ser resolvido.

Tendo sido estabelecidos os tipos de classificadores global e local a serem utilizados, precisou-se decidir qual classificador local deveria empregar nas partições originadas. Esta decisão foi tomada considerando que uma rede neural deve ser preterida em função do k-NN quando a quantidade de instâncias para treiná-la não é suficientemente grande. É difícil precisar objetivamente um limiar de decisão para o que se considera grande ou pequeno neste contexto. Resolveu-se adotar para este fim o número empírico de instâncias requeridas para treinar um classificador minimizando os efeitos da praga da dimensionalidade. A praga da dimensionalidade (*curse of dimensionality* – Jain *et al.* [60]) é um fenômeno ocasionado pela dimensão do espaço de atributos. A alta dimensão do vetor impõe dificuldades ao sistema de aprendizado tanto porque um número grande de atributos torna o processo computacionalmente mais caro, quanto porque requer grandes conjuntos de treinamento. Em Jain *et al.* [60] sugere-se que o número de instâncias para treinamento seja pelo menos dez vezes superior ao tamanho do vetor de atributos. Adotou-se neste texto uma solução mais conservadora, estabelecendo o limiar como o máximo entre dez vezes a quantidade de atributos e um décimo do tamanho do conjunto de treinamento. Esta atitude favorece a utilização do classificador k-NN em problemas em que o número de atributos é bastante reduzido, sem incorrer nas limitações inerentes ao custo de armazenamento e busca, já que emprega-se no máximo dez por cento das instâncias de treinamento. Esse limiar também estabelece a quantidade mínima de amostras requeridas para iniciar o processo de particionamento

pois, naturalmente, o método proposto se aplica somente quando o conjunto de treinamento for suficientemente grande.

5.2.1 Algoritmo de particionamento do espaço de atributos e treinamento dos classificadores

As regras que orientam o processo de criação dos classificadores estabelecem implicitamente alguns passos do algoritmo de particionamento do espaço de atributos que devem ser colocados mais claramente. Esta seção reapresenta o Algoritmo 1 em maior nível de detalhe, incorporando informações relativas aos classificadores.

Tendo em vista a regra de amarração que relaciona classificadores locais a partições, pode-se definir objetivamente o critério de parada do processo de particionamento. Como uma partição associada a um classificador k-NN não pode ser segmentada em outras que utilizem um classificador diferente, e como o critério de classificação é baseado em uma contagem, torna-se pouco vantajoso realizar o particionamento de uma região associada a este tipo de classificador, já que muitas das instâncias consultadas no processo de votação seriam as mesmas reproduzindo, portanto, os mesmos erros anteriores. Sendo assim, a fixação do k-NN pode ser vista como um caso final para segmentação de uma região do espaço de atributos.

Baseado no que fora mencionado ao longo desta seção, o algoritmo de particionamento pode ser reescrito como

Algoritmo 2 (Particionamento do espaço de atributos)

$Particionar(R, h)$

R : *Região*

h : *Classificador (Hipótese)*

m : *Número de classes*

$\mathcal{M}$: *Conjunto de exemplos não-aprendidos*

$\mathcal{T}$: *Conjunto de treinamento original*

$\mathcal{M} \leftarrow \{\boldsymbol{x} | h(\boldsymbol{x}) \neq cls(\boldsymbol{x})\}$

se $(h = kNN)$ **então retorne;**

enquanto $\mathcal{M} \neq \emptyset$

$\boldsymbol{u}_m = \underset{\boldsymbol{x} \in \mathcal{M}}{\text{argmax}}\{-\sum_i h(\boldsymbol{x}_i) \ln h(\boldsymbol{x}_i)\}$

para todo $\boldsymbol{x} \in R$

se $(\|\boldsymbol{x} - \boldsymbol{u}_m\| \leq r \wedge mse(\boldsymbol{x}) \geq \tau)$ **então** $R_m \leftarrow \{\boldsymbol{x}\} \cup R_m$

$\mathcal{M} \leftarrow \mathcal{M} \setminus R_m$

se $(|R_m| > 10m) \wedge (|R_m| > 0,1|\mathcal{T}|)$ **então**

$h_m \leftarrow Perceptron$

senão

$h_m \leftarrow kNN$

$Treinar(R_m, h_m)$

$Particionar(R_m, h_m)\ m = 1, \ldots, M$

5.2.2 Fixação do parâmetro k (k-NN)

Normalmente o valor do parâmetro k é escolhido por validação cruzada no treinamento de um classificador k-NN (Friedman [40]). Nos testes realizados foram mensuradas taxas de reconhecimento do método quando treinado com classificadores k-NN empregando diferentes valores de k. Foram também implementadas diferentes variações do método empregando procedimentos de aprendizado e inferência baseados em mecanismos para tratamento de evidências e de casos observados como valores discretos ou como probabilidades, como sumarizado na Tabela 5.2

Os resultados destes experimentos estão apresentados nos gráficos das Figuras 5.2-5.7. Pode-se observar que na maioria dos casos a taxa de reconhecimento é mais elevada quando $k = 1$. Este fenômeno pode ser justificado pelo fato de que nas partições criadas há muita ambiguidade, aliado ao fato de que em um espaço de grande dimensão, esparsamente povoado por instâncias do conjunto de treinamento, ao consultar um número maior de vizinhos o resultado de uma classificação acertada fica diluído devido à contribuição de elementos de outras classes. Teoricamente, embora um número maior de vizinhos aproxime melhor a distribuição alvo (Cover e Hart [26]), na prática isto não ocorre por ser limitado o número de instâncias disponíveis no conjunto de

Tabela 5.2: Descrição dos sistemas implementados

Sistema Implementado	Procedimento de treinamento	Procedimento de inferência
Sistema I	Contínuo	Contínuo
Sistema II	Contínuo	Discreto
Sistema III	Discreto	Contínuo
Sistema IV	Discreto	Discreto

treinamento. Além disto, existe o erro associado à construção da rede Bayesiana. Para obter B_P utilizou-se uma estratégia de preenchimento que favorece fortemente as contribuições dos classificadores mais especializados. Portanto, desde que estes classificadores não forneçam uma estimativa acurada da distribuição alvo, o mecanismo de inferência não consegue realizar uma predição precisa.

Este fenômeno é mais acentuado nas saídas produzidas pelos sistemas I e II, em que B_P é construído com base em uma aproximação de uma distribuição de probabilidades. Na prática o k-NN é bem empregado como classificador, mas não revela bons resultados como aproximador de funções, em particular de uma função de distribuição de probabilidades. Assim, à medida que aumenta-se o valor do parâmetro k, o classificador tende a ficar menos específico sem, em contrapartida, tornar-se suficientemente acurado como aproximador de função. Sem fornecer uma estimativa precisa das distribuições subjacentes os classificadores não fornecem suporte à tomada de decisão realizada pelo combinador. Enquanto que, para $k = 1$, embora não seja realizada uma boa aproximação de $p(\omega|\boldsymbol{x})$, a classe atribuída ao padrão de teste, que em geral é acertada, é introduzida na rede Bayesiana com um peso bastante elevado, o que justifica o bom desempenho dos algoritmos para valores de k pequenos.

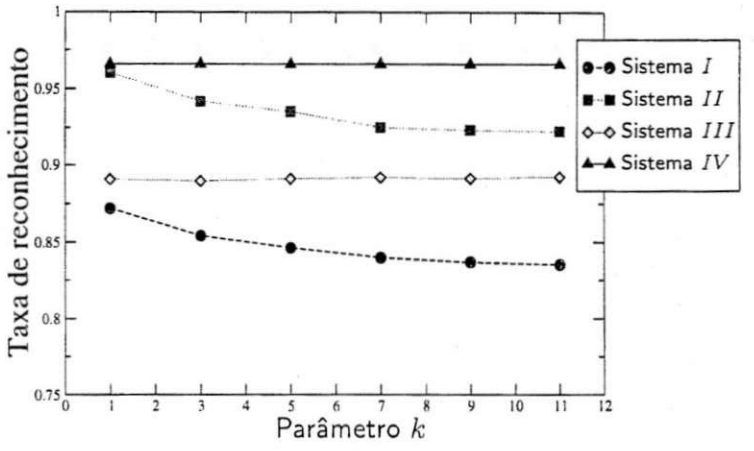


Figura 5.2: Parâmetro k × taxa de reconhecimento (base *adult*)

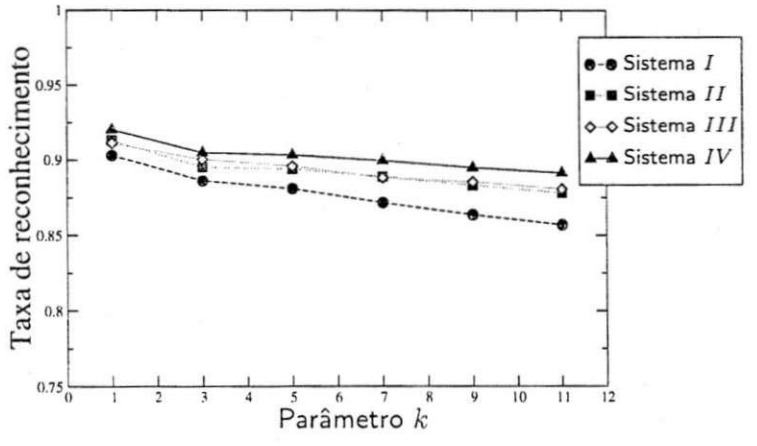


Figura 5.3: Parâmetro k × taxa de reconhecimento (base *letter*)

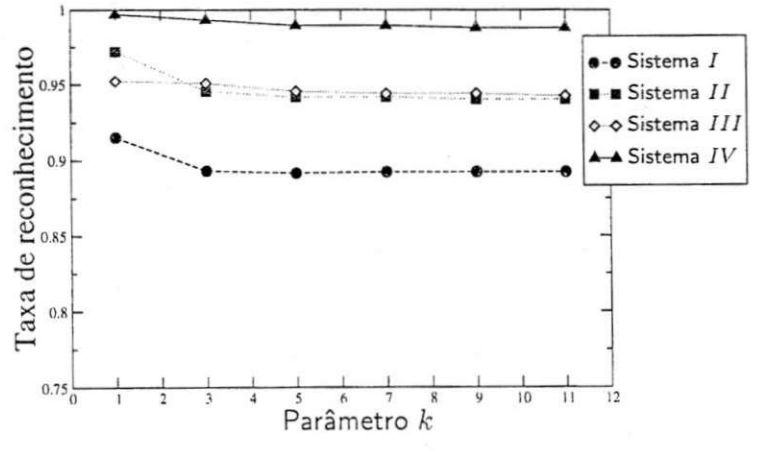


Figura 5.4: Parâmetro k × taxa de reconhecimento (base *musk*)

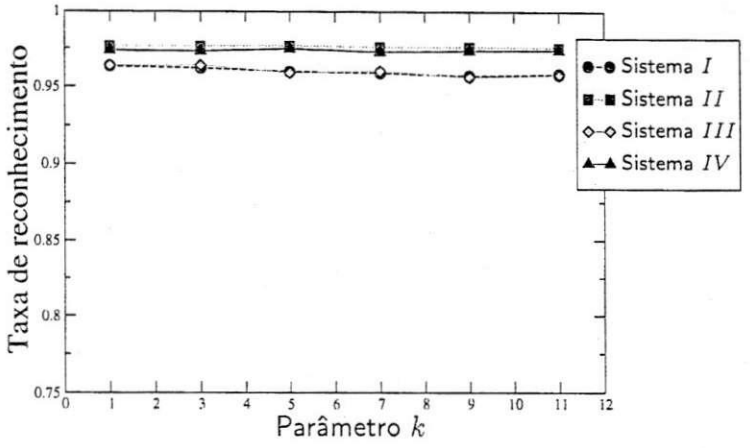


Figura 5.5: Parâmetro k × taxa de reconhecimento (base *nursery*)

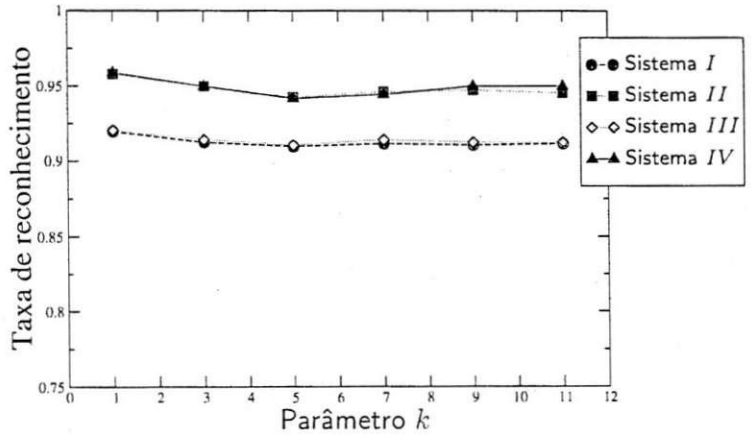


Figura 5.6: Parâmetro k × taxa de reconhecimento (base *pageblocks*)

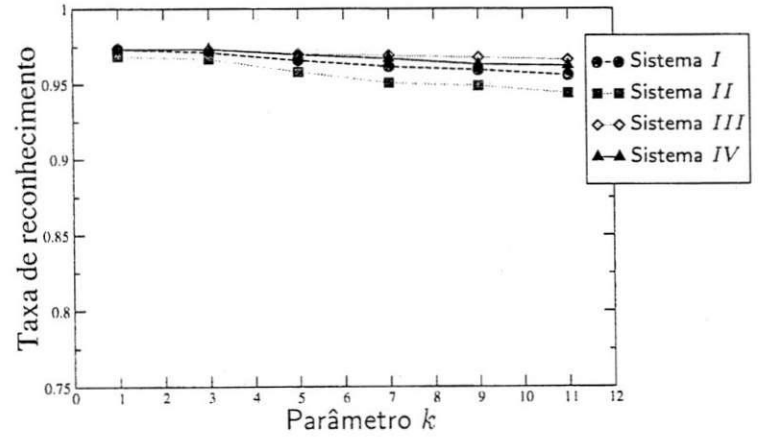


Figura 5.7: Parâmetro k × taxa de reconhecimento (base *pendigits*)

Os resultados também revelaram um comportamento aparentemente paradoxal dos sistemas implementados. Observa-se que o Sistema I, por empregar um tratamento suave na manipulação das saídas dos classificadores incorporando adaptações para construção e utilização da rede Bayesiana, apresenta taxas de reconhecimento muito baixas quando empregado com as bases *adult* e *musk*. Este comportamento se justifica pelo fato destas bases, que tratam de problemas de classificação binária, possuírem distribuição de classes extremamente assimétricas, ilustrados na Tabela 5.3. Esta distribuição assimétrica origina matrizes de probabilidades dos nós imprecisas. Como ilustrado na Figura 5.8, em que são apresentados os dois primeiros nós da rede Bayesiana associada à base *adult*, há forte incerteza na associação dos pares $\{X_0 = 0, X_1 = 0\}$ e $\{X_0 = 0, X_1 = 1\}$, assim, o valor de X_0 não é afetado significativamente quando alguma evidência sobre o valor de X_1 é fornecida. Se a rede admite probabilidades como entrada, então, é necessário que exista forte evidência de que $X_1 = 0$, isto é $P(X_1 = 0) \rightarrow 1$, para tornar o estado $X_0 = 0$ mais provável. Este tipo de imposição não se faz necessária quando a entrada de evidências é tratada como valores discretos.

Tabela 5.3: Distribuição das classes na base *adult*

Classe	Quantidade de instâncias
0	22654 (75,1%)
1	7508 (24,89%)

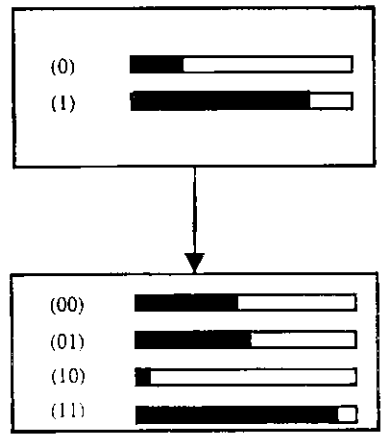


Figura 5.8: Nós da rede Bayesiana associada à base *adult*

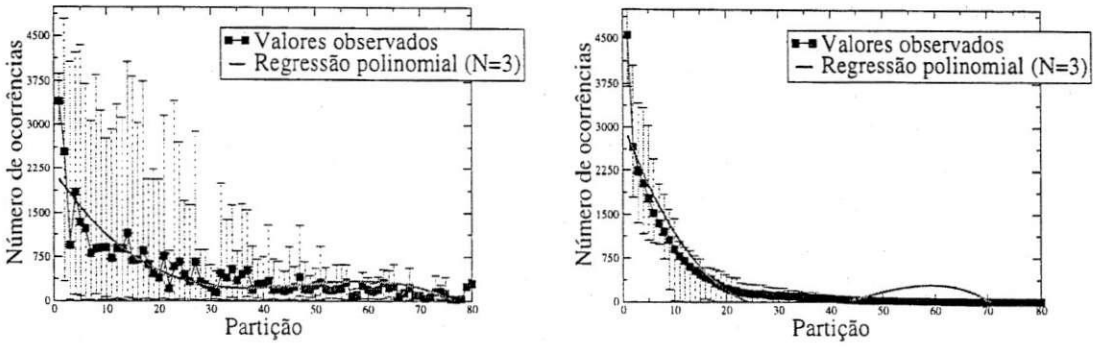
5.3 Avaliação do processo de particionamento

Para avaliar o processo de particionamento foi realizada uma comparação entre os critérios de entropia e erro médio quadrático adotando-se a seguinte metodologia: selecionou-se um sub-

conjunto de treinamento com três quartos das instâncias disponíveis obtidas aleatoriamente, aplicou-se o algoritmo de particionamento e procedeu-se à contagem do número de vetores existentes em cada uma das partições originadas. Este processo foi repetido 30 vezes a fim de se obter uma curva ilustrando o comportamento dos valores médios do número de instâncias por partições, considerando-se apenas o primeiro nível do diagrama de classificadores. Uma extensão desta análise, considerando-se os demais níveis do diagrama de classificadores, torna o experimento mais caro, sem um correspondente ganho na qualidade da informação obtida, já que o experimento se limita a investigar essencialmente as diferenças resultantes da aplicação do critério da entropia ou do erro médio quadrático na definição de $s_m(\alpha)$. Uma vez que o classificador k-NN é considerado um caso limite para aplicação do algoritmo de particionamento, e já que a investigação está restrita ao primeiro classificador, é desnecessário investigar a influência do parâmetro k na análise realizada nesta seção. A rede neural usada como classificador global nestes experimentos foi uma rede *perceptron* treinada com critério de parada igual ao máximo de 100 ciclos ou erro médio quadrático inferior a $0,01$.

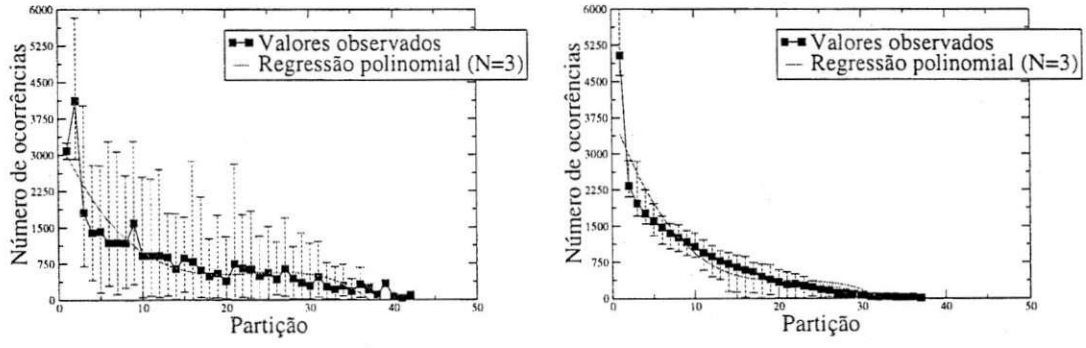
A fim de estabelecer um valor de referência, foram investigadas partições originadas por um processo de busca exaustiva. Para realizar a busca exaustiva definiu-se uma hiperesfera $\mathcal{H}$ em $\mathbb{R}^P$ com os mesmos raios obtidos pelos critérios de entropia e erro médio quadrático. Selecionou-se dentre os elementos de $\mathcal{M}$ aquele que maximiza a quantidade de vizinhos contidos em $\mathcal{M} \cap \mathcal{H}$. Este experimento foi repetido 32 vezes a fim de que se pudesse observar o comportamento médio do número de partições geradas por cada um dos critérios investigados. A curva do número mínimo, máximo e médio de elementos por partição obtidos por cada um dos critérios está ilustrado nas Figuras 5.9 a 5.20. A curva originada pelo procedimento de busca exaustiva pode pode ser considerada um caso ideal, entretanto, em razão de requerer a realização de $O(N^2)$ operações de cálculo de distância em $\mathbb{R}^P$, sua aplicação torna-se proibitiva em casos práticos.

Observa-se que na maioria dos casos o critério da entropia origina um maior número de partições, especialmente para as bases com muitas instâncias (Figuras 5.9 e 5.15, 5.10 e 5.16). Nas bases menores (Figuras 5.11 e 5.17, 5.12 e 5.18, 5.13 e 5.19, 5.14 e 5.20) as curvas obtidas pelos dois critérios são próximas. A razão deste comportamento deve-se ao fato de que o raio da hiperesfera que contém a partição é calculado com base no valor médio das distâncias do seu centro às demais instâncias de $\mathcal{M}$. Como a entropia possui a propriedade de localizar melhor os elementos em uma vizinhança de padões não aprendidos então o raio da hiperesfera é menor, gerando, portanto, um maior número de pequenas partições. O erro médio quadrático, por sua vez, origina poucas partições com muitos elementos. Na prática, é preferível usar o critério da entropia tendo em vista que ele aproxima melhor a curva ideal obtida pela adoção de um procedimento de busca exaustiva. Se for desejável diminuir o número de partições pode-se aumentar o tamanho do raio fazendo $r = (1+\delta) \underset{x_i \in \mathcal{M}}{ave}\{|x_m - x_i|\}$ com $\delta > 0$.



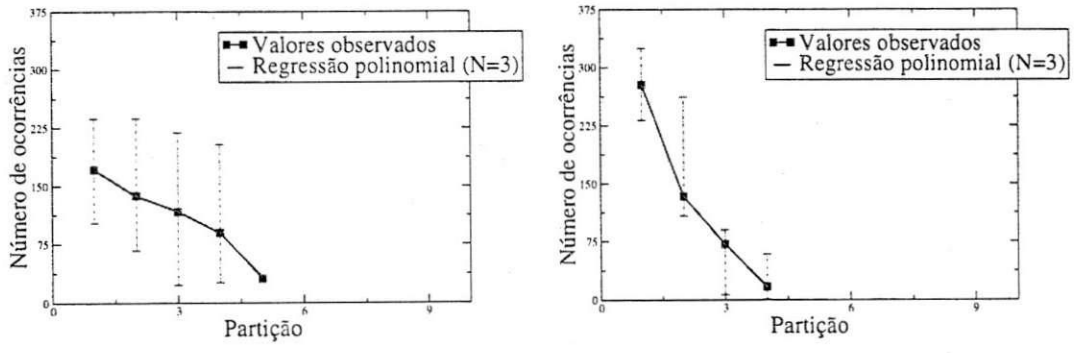
(a) Entropia (b) Busca exaustiva

Figura 5.9: Instâncias × partição × entropia (base *adult*)



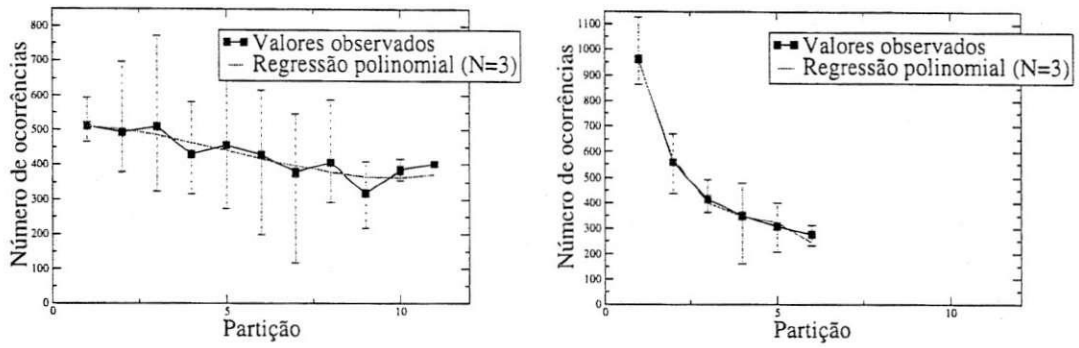
(a) Entropia (b) Busca exaustiva

Figura 5.10: Instâncias × partição × entropia (base *letter*)



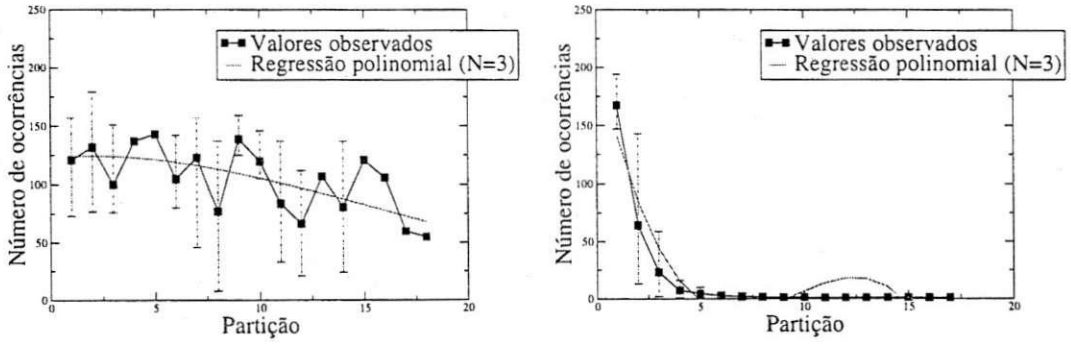
(a) Entropia (b) Busca exaustiva

Figura 5.11: Instâncias × partição × entropia (base *musk*)



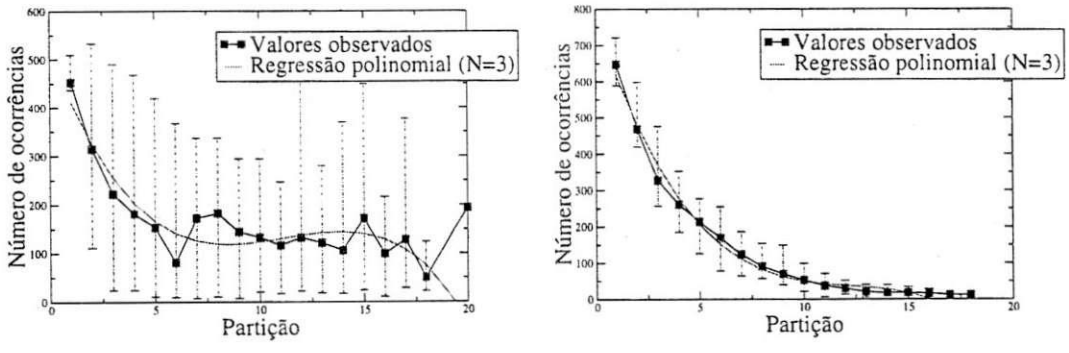
(a) Entropia (b) Busca exaustiva

Figura 5.12: Instâncias × partição × entropia (base *nursery*)



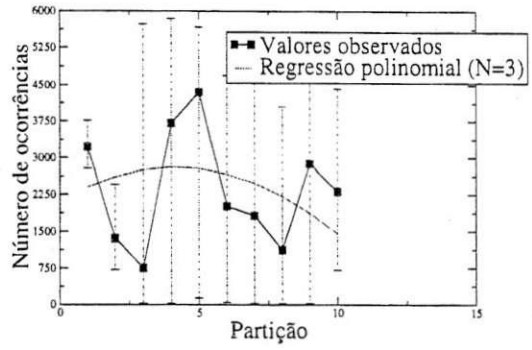
(a) Entropia (b) Busca exaustiva

Figura 5.13: Instâncias × partição × entropia (base *pageblocks*)

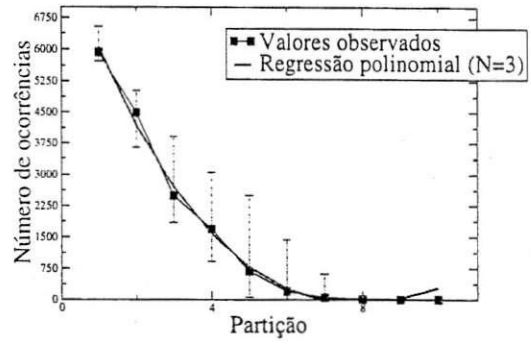


(a) Entropia (b) Busca exaustiva

Figura 5.14: Instâncias × partição × entropia (base *pendigits*)

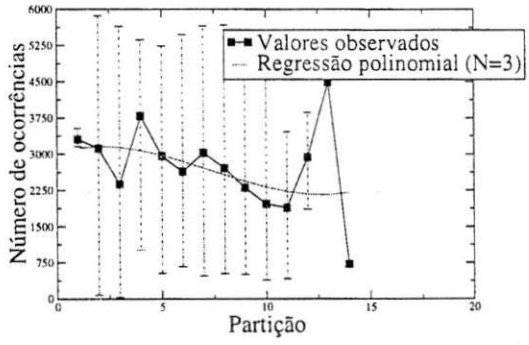


(a) MSE

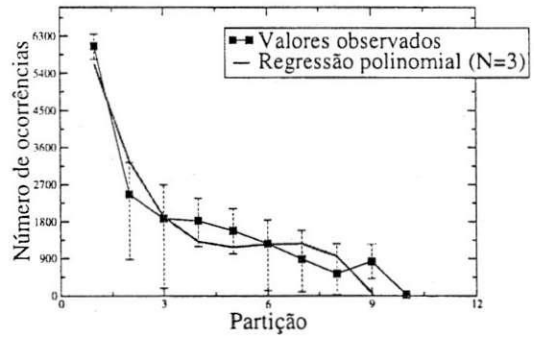


(b) Busca exaustiva

Figura 5.15: Instâncias × partição × MSE (base *adult*)

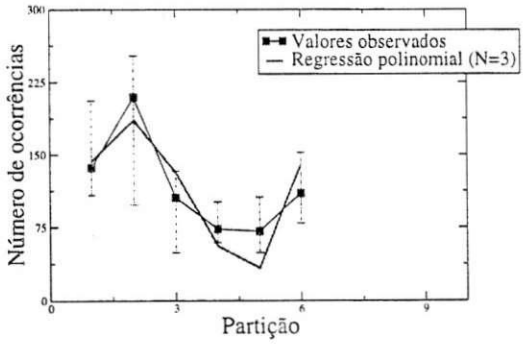


(a) MSE

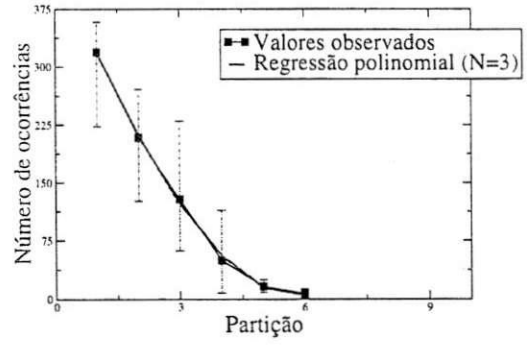


(b) Busca exaustiva

Figura 5.16: Instâncias × partição × MSE (base *letter*)

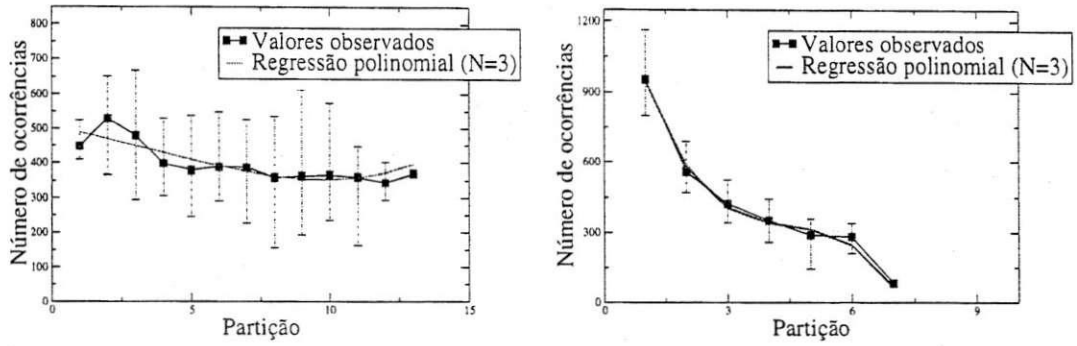


(a) MSE



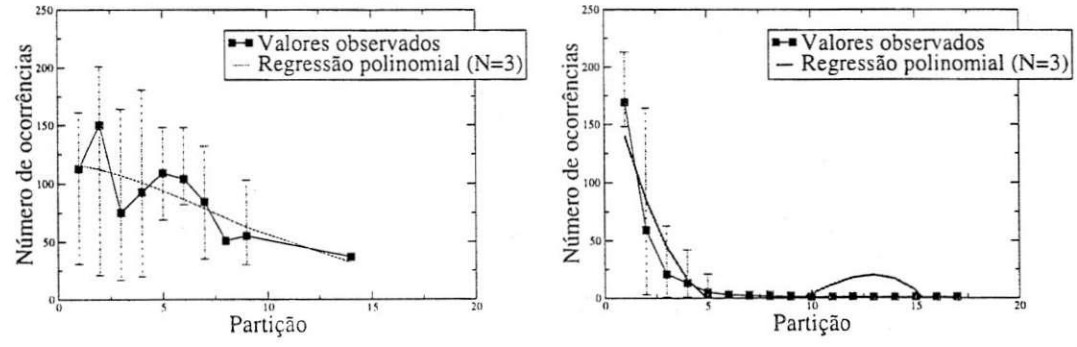
(b) Busca exaustiva

Figura 5.17: Instâncias × partição × MSE (base *musk*)



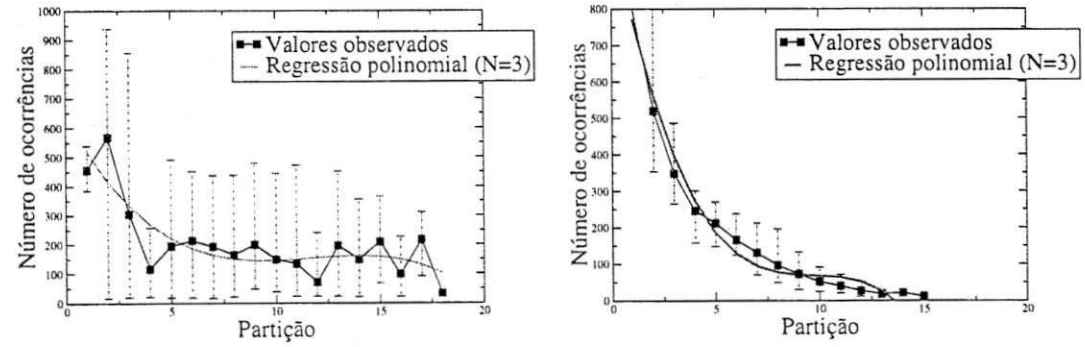
(a) MSE (b) Busca exaustiva

Figura 5.18: Instâncias × partição × MSE (base *nursery*)



(a) MSE (b) Busca exaustiva

Figura 5.19: Instâncias × partição × MSE (base *pageblocks*)



(a) MSE (b) Busca exaustiva

Figura 5.20: Instâncias × partição × MSE (base *pendigits*)

5.4 Avaliação do sistema de inferência

O sistema de inferência foi avaliado comparando-se as taxas de reconhecimento obtidas pela aplicação de cada uma das quatro variações do sistema proposto com aquelas obtidas pela aplicação dos classificadores 1-NN, 3-NN e de redes *Perceptron* (*PCPT*) e MLP isoladamente. Os resultados destes experimentos estão apresentados na Tabela 5.4. Pode-se observar que, na maioria dos casos, a aplicação do algoritmo de inferência eleva a capacidade preditiva do sistema. Para algumas bases, como *adult*, a diferença obtida é ainda mais proeminente. De um modo geral o processo de particionamento realiza uma boa identificação das instâncias que precisam ser empregadas num novo ciclo de treinamento. Nas bases em que o classificador linear tem um desempenho muito baixo o resultado do k-NN isoladamente supera o do combinador, como ocorrido com a base *letter*. Quando as novas partições são treinadas também por classificadores neurais e estes se mostram efetivos, então a predição do combinador supera a dos classificadores componentes isoladamente.

Tabela 5.4: Comparação entre o método proposto e outros algoritmos de aprendizado de máquina

Base de padrões	Taxas de reconhecimento (%)						
	1-NN	*PCPT*	*MLP*	$S-I$	$S-II$	$S-III$	$S-IV$
adult	78,05	82,62	84,36	87,2	96,0	89,21	**96,6**
letter	**95,67**	68,62	83,85	90,32	91,32	91,12	92,02
musk	95,37	94,31	97,87	91,51	97,19	95,22	**99,69**
nursery	80,4	91,89	92,66	96,33	**97,68**	96,41	97,49
pageblocks	93,05	93,42	93,79	91,23	95,79	92,05	**95,89**
pendigits	**97,74**	87,22	92,68	97,34	96,88	97,39	97,34

5.5 Conclusão

A aplicação do método proposto para os problemas abordados neste capítulo revelaram taxas de reconhecimento bastante elevadas, sendo compatíveis com muitos métodos conhecidos na literatura de aprendizado de máquinas. Tais resultados demonstram que a estratégia usada para realizar o particionamento do espaço de atributos conciliada com a combinação dos classificadores originados pode ser empregada em diversas situações práticas. O combinador obtido com uma rede Bayesiana eleva a capacidade preditiva produzindo resultados que superam os obtidos com os classificadores constituintes isoladamente. Esta estratégia permite que possam ser empregados

classificadores fracos[1] na composição do sistema e ainda assim obter taxas de reconhecimento compatíveis com a de métodos mais competitivos.

Os testes também revelaram que a eficiência dos sitemas avaliados, considerando as quatro variações apresentadas na Tabela 5.2, é bastante diferenciada quando a distribuição das classes é desigual ou assimétrica. Os sistemas que empregam probabilidades no tratamento de evidências são bastante afetados quando o número de vizinhos consultados pelos classificadores locais (k-NN) aumenta, passando o sistema a depender mais fortemente da estimativa acurada da distribuição $p(\omega|\boldsymbol{x})$, que vai se tornando menos específica em razão do conjunto de treinamento ser finito, e menos do resultado da classificação. Entretanto a capacidade preditiva do Sistema IV, que trata as evidências e os casos de aprendizado como valores discretos, não é afetada à medida em que k aumenta. Isto demonstra que as idéias discutidas no Capítulo 4, que procuram suavizar o tratamento das entradas empregadas no aprendizado e inferência da rede Bayesiana, não se aplicam às bases em que o número de instâncias por classes não é uniformemente distribuído.

A proposta de criação do sistema de classificação de padrões apresentada procurou atender o requerimento de que o grau de intervenção de um especialista humano na configuração dos classificadores fosse minimizada. A arquitetura do método proposto é modelada dinamicamente. Quanto mais complexa for a tarefa de discernir as fronteiras de decisão do espaço de atributos, mais complexa será a arquitetura do sistema, que cria sob demanda novos nós e conexões a fim de suportar a complexidade do problema apresentado. Esta plasticidade dinâmica entretanto pode não ser alcançada se a natureza do problema for tão complexa que um discriminante linear seja incapaz de realizar uma predição acertada. Se a taxa de reconhecimento de um classificador linear for muito próxima de zero o sistema jamais irá convergir, uma vez que fora estabelecido que o classificador global é uma rede *perceptron*. Sendo assim, as vantagens em se utilizar o método proposto podem ser alcançadas para problemas em que:

1. Um classificador linear, ou um outro classificador fraco, possa ser empregado como classificador global (requisito de convergência). É necessário, portanto, que exista uma boa separabilidade entre as classes. Mesmo não sendo possível separá-las por hiperplanos, a transição entre padrões de classes distintas no espaço de atributos deve ser suave. Se os padrões forem demasiadamente misturados, o classificador global não poderá localizar, mesmo que aproximadamente, as fronteiras de separação.

2. Exista um grande número de amostras de treinamento. Um número elevado de instâncias também implica num elevado número de redundâncias, sendo assim, o custo de treinamento é minimizado já que o balanço entre o número de instâncias usadas no treinamento dos classificadores global e local fica melhor distribuído: muitas instâncias para treinar o classificador global (fraco) e poucas instâncias para treinar os classificadores locais (fortes).

[1] do inglês *weak classifier*

3. Preferencialmente exista uma distribuição de classes aproximadamente uniforme. Desta forma pode-se usar com segurança os Sistemas I e III que fornecem como saída, além da classe do padrão de testes, uma estimativa mais acurada da probabilidade $p(\omega|\boldsymbol{x})$. Esta probabilidade pode ser usada, por exemplo, como critério de rejeição.

Capítulo 6

Estudo de caso – experimento com reconhecimento de dígitos

A fim de validar o método em um caso prático, realizou-se sua comparação com diversos métodos de aprendizado de máquina aplicados a um problema de reconhecimento de dígitos manuscritos isolados. Foi empregada a base de caracteres manúscritos NIST (*US National Institute of Standarts and Technology*) (Grother [51]). Esta base contém imagens de caracteres alfanuméricos agrupados em várias bases menores formadas por imagens de letras maiúsculas, minúsculas, palavras, dígitos isolados e cadeias de dígitos (*digit strings*). A base de dígitos isolados é formada por 341858 imagens binárias, com resolução de 300 dpi, distribuídas em três sub-conjuntos: *hsf_0123* (223123 imagens), *hsf_4* (58646 imagens) e *hsf_7* (60089 imagens). O conjunto *hsf_0123* é normalmente utilizado para treinamento e validação, enquanto que os conjuntos *hsf_4* e *hsf_7* são usados para teste. Do conjunto *hsf_0123* emprega-se as primeiras 19500 imagens de cada classe para treinamento, resultando no particionamento ilustrado na Tabela 6.1. Preferencialmente, utiliza-se mais frequentemente a base de testes *hsf_7*, por ser mais bem comportada, em detrimento da *hsf_4*, que é mais ruidosa. Por conta disto, as taxas de acerto alcançadas são muito elevadas. A Tabela 6.2, ilustra alguns resultados citados na literatura. Naturalmente a comparação direta destes números deve ser evitada. Isto é, deve-se evitar tirar conclusões julgando um método como mais ou menos preciso que outro pois, embora os testes tenham sido realizados com a mesma base, as imagens empregadas em treinamento e teste não foram necessariamente as mesmas, assim como não foram empregadas as mesmas operações de pré-processamento, nem foram usados critérios de rejeição em todos os experimentos. Além disto, as diferenças percentuais existentes precisariam ser comparadas sob a luz de um teste estatístico a fim de que se possa julgar o quão significativa elas são. Pela Tabela 6.2 pode-se perceber que as taxas de reconhecimento são, de fato, bastante elevadas mas não se pode antecipar informações qualitativas que indiquem se um método é mais ou menos preciso que outro.

A fim de tornar as condições de realização dos testes mais homogêneas foram usadas as mes-

Tabela 6.1: Distribuição de amostras na base NIST (adaptado de Correia [25])

Classe	*hsf_0123*			*hsf_4*	*hsf_7*
	Amostras	Treino	Validação	Teste	Teste
0	22971	19500	3471	5560	5893
1	24771	19500	3371	6655	6567
2	22131	19500	2631	5888	5967
3	23172	19500	3672	5819	6036
4	21549	19500	2049	5722	5873
5	19545	19500	45	5539	5684
6	22128	19500	2628	5858	5900
7	23208	19500	3708	6097	6254
8	22029	19500	2529	5695	5889
9	21619	19500	2119	5813	6026
$\sum$	223123	195000	28123	58646	60089

mas imagens para treinamento e teste e o mesmo algoritmo de pré-processamento, apresentado detalhadamente na Seção 6.1. A Seção 6.2 realiza uma avaliação do sistema de extração de características utilizado quando aplicado ao algoritmo de particionamento do espaço de atributos. A Seção 6.3 discorre sobre a metodologia de testes empregada, apresenta e analisa os resultados numéricos obtidos. A Seção 6.4 realiza um fechamento do capítulo traçando considerações finais sobre o conteúdo apresentado.

6.1 Extração de características

Reconhecimento de caracteres óticos é uma linha de pesquisa com um longo histórico na área de reconhecimento de padrões (Arica e Yarman-Vural [5]). A importância das pesquisas nesta área se verifica tanto pelo potencial econômico das aplicações de processamento automático de documentos, considerando um contexto mais amplo onde o reconhecimento de caracteres se localiza, como pela existência de fóruns específicos de abrangência internacional criados para discussão e veiculação destas pesquisas a exemplo das conferências IWFHR (*International Workshop on Frontiers in Handwritten Recognition*) e ICDAR (*International Conference on Document Analysis and Recognition*), além de um grande volume de publicações veiculadas em periódicos científicos internacionalmente reconhecidos tais como: *Pattern Recognition*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* e *Pattern Recognition Letters*.

No resumo elaborado por Arica e Yarman-Vural [5], os autores situam os desenvolvimentos na

Tabela 6.2: Resultados obtidos com classificadores que operaram sobre a base NIST

Referência	Tipo do classificador	Número de amostras	Subconjunto	Reconhecimento (%)	Rejeição (%)	Erro (%)
Britto Jr. [14]	Estatístico	10000	*hsf_7*	98.02	-	1.98
Correia [24]	Neural	58646	*hsf_7*	98.25	-	5.51
Kim *et al.* [67]	Neural	10909	-	98.62	0.42	1.39
Cheung e Yeung [18]	Estatístico	11791	-	94.7	-	5.3
Lin *et al.* [79]	Neural	3000	-	98.24	-	1.76
Teo e Shinghal [108]	Híbrido (neural/estrutural)	20000	-	93.21	3.47	3.26
Zhang *et al.* [117]	Similaridade e correlação	10000	-	95.7	-	4.3
Shi *et al.* [102] *et al.*	Estatístico	58646	*hsf_7*	98.25		1.75
Oliveira [86]	Híbrido	68061	*hsf_7*	99.13		0.87

área de reconhecimento *off-line* de caracteres em três momentos históricos: um tempo anterior aos anos 1980, em que a tecnologia existente para adquirir e processar imagem de caracteres óticos era bastante limitada, a década de $1980 - 1990$ em que os avanços em tecnologia da informação propiciaram um rápido desenvolvimento de antigas metodologias e um tempo posterior aos anos 1990 em que os autores apontam estar ocorrendo o verdadeiro progresso em reconhecimento de caracteres com o desenvolvimento de novas metodologias, tais como redes neurais multi-camada e cadeias escondidas de Markov, aliadas com os contínuos avanços em tecnologia da informação. Um grande desafio no tempo atual é realizar o reconhecimento de imagens de caracteres cursivos, que apresenta uma grande variabilidade de estilos por serem produzidos à mão. Em contraposição à escrita cursiva, o reconhecimento de caracteres óticos impressos é uma linha cujo domínio está consolidade e não requer novos investimentos (Trier *et al.* [110], Arica e Yarman-Vural [5]).

Imagens de caracteres cursivos tipicamente estão associadas a vetores de um espaço de atributos de grande dimensão. É desejável, portanto, que o vetor de atributos seja pré-processado a fim de reduzir sua dimensão. Se este processo é feito preservando a natureza física dos atributos, isto é, mantendo as características mais significativas e eliminando aquelas que são redundantes, ele é denominado seleção de características. Se, por outro lado, o vetor de características é levado a outro de dimensão menor por uma operação de transformação espacial, em que os componentes não preservam os mesmos valores existentes no vetor original, ocorre um processo de extração de características. Um dos métodos mais populares de extração de características é a análise de componentes principais (Johnson e Wichern [63]), largamente explorado em Análise Multivariada, Reconhecimento de Padrões de um modo geral e em Processamento de Imagens.

A literatura contém vasta referência a métodos de extração de características para reconhecimento de caracteres manuscritos, um resumo dos principais desenvolvimentos nesta área pode ser obtido em Trier *et al.* [110], que trata especificamente de extração de características. Os trabalhos de Liu *et al.* [80] e Dong [32] avaliam diversos métodos de extração de características mas realizam uma abordagem mais ampla estendendo sua análise também para métodos de classificação. O trabalho de Dong [32] procura apresentar o estado da arte na área de reconhecimento de caracteres comparando diversos métodos de extração de características e de classificação. Nesse artigo, Dong identificou o método de histogramas direcionais de Shi *et al.* [102] como o mais bem-sucedido em seus experimentos. A descrição detalhada deste método está apresentada na Seção 6.1.1 e uma variação do método de histogramas direcionais, descrito na Seção 6.1.2, foi empregado nos experimentos apresentados neste capítulo.

6.1.1 Método de histogramas direcionais (Shi et al. [102])

Um histograma direcional é um procedimento para realizar cruzamento de características pontuais de uma imagem através de uma distribuição de freqüências. As características pontuais, isto é, pertencentes a um *pixel* isoladamente, exploradas por Shi *et al.* são curvatura e gradiente. Sendo o gradiente um vetor, esta característica é por sua vez desmembrada em duas outras: a magnitude e a fase. Uma vez que a fase tem uma variação limitada ao intervalo 0 a 2π, ela pôde ser usada para gerar intervalos de mesma amplitude empregados como classes para a distribuição de freqüência, por este motivo o método é denominado direcional.

Segundo Shi *et al.* [102] a geração destes histogramas a partir de imagens binárias pode ser obtido através da seqüência de passos listada a seguir:

i Aplica-se um algoritmo de normalização em escala a fim de padronizar a dimensão das imagens para um número de linhas e colunas previamente estabelecido, Figura 6.1.

Figura 6.1: Normalização em escala

ii Aplica-se o filtro da média 2×2 (Dong [32] empregou um filtro de dimensão 3×3) r[1] vezes

[1] os autores não especificam o valor de r utilizado

a fim de obter uma imagem em nível de cinza, Figura 6.2.

Figura 6.2: Conversão de binário para nível de cinza

iii Aplica-se um algoritmo de normalização para limitar a variação dos níveis de cinza ao intervalo $[0, 1]$

iv Aplica-se o filtro de Roberts (Gonzales [50]) para obter a magnitude e fase do vetor gradiente, dado pelas equações abaixo.

$$\Delta u = f(i+1, j+1) - f(i, j) \tag{6.1}$$

$$\Delta v = f(i+1, j) - f(i, j+1) \tag{6.2}$$

$$\theta(i, j) = \arctan\left(\frac{\Delta u}{\Delta v}\right) \tag{6.3}$$

$$s(i, j) = \sqrt{\Delta u^2 + \Delta v^2} \tag{6.4}$$

Em que $\theta(i, j)$ e $s(i, j)$ correspondem respectivamente às matrizes da fase e magnitude (Figura 6.3) da imagem em nível de cinza $f(i, j)$.

Figura 6.3: Imagens da fase e magnitude

Os passos listados acima formam uma etapa inicial que se aplica à construção de qualquer tipo de histograma, seja o originado através do cruzamento da fase com a magnitude do gradiente, seja o originado pelo cruzamento da magnitude com a curvatura. Para realizar o cálculo da curvatura Shi *et al.* [102] apresentam o procedimento descrito a seguir:

i Seja F uma imagem e f_i o valor de F no *pixel* i. Considere a representação abaixo empregada para definir uma vizinhança em F em torno de f_0

f_4	f_3	f_2
f_5	f_0	f_1
f_6	f_7	f_8

Obtém-se as seguintes aproximações das derivadas parciais de $F(\cdot)$ nas direções x (horizontal) e y (vertical):

$$a_{10} = (f_1 - f_5)/2 \approx \frac{\partial F}{\partial x} \tag{6.5}$$

$$a_{20} = (f_1 + f_5 - 2f_0)/4 \approx \frac{\partial^2 F}{\partial x^2} \tag{6.6}$$

$$a_{01} = (f_3 - f_7)/2 \approx \frac{\partial F}{\partial y} \tag{6.7}$$

$$a_{02} = (f_3 + f_7 - 2f_0)/4 \approx \frac{\partial^2 F}{\partial y^2} \tag{6.8}$$

$$a_{11} = ((f_2 - f_8) - (f_4 - f_6))/4 \approx \frac{\partial^2 F}{\partial y \partial x} \tag{6.9}$$

ii O valor da curvatura em f_0 é estimado

$$c = \frac{y''}{\sqrt{(1 + y'^2)}} \tag{6.10}$$

sendo y' e y'' definidos como

$$y' = -(\frac{a_{10}}{a_{01}}) \tag{6.11}$$

$$y'' = -(\frac{2(a_{10}^2 a_{02} - a_{01} a_{10} a_{11} + a_{01}^2 a_{20})}{a_{01}^3}) \tag{6.12}$$

A etapa final consiste em montar os histogramas. Os passos seguintes descrevem como realizar a construção dos histogramas relacionando fase com magnitude do vetor gradiente. O mesmo procedimento pode ser usado para construir histogramas que relacionam intensidade do gradiente com curvatura, as ocorrências em que se aplicam a contagem se mantém constante, isto é, conta-se sempre a magnitude ou intensidade do gradiente, muda-se apenas o conjunto no qual se realiza a estratificação das classes. Para fazer um histograma relacionando fase e magnitude, utiliza-se como classes sub-intervalos da faixa $0 \ldots 2\pi$. Para fazer um histograma relacionando curvatura e magnitude as classes sugeridas pelos autores foram intervalos equispaçados na faixa $(-3, 3)$, em princípio como a curvatura pode ter magnitude infinita a faixa de variação tem que ser aberta nas duas extremidades.

i Realiza-se a quantização do intervalo $[0, 2\pi]$ em 32 classes de amplitude $\pi/16$.

ii Divide-se a imagem normalizada do caracter em 81 blocos (9 linhas por 9 colunas) e acumula-se a intensidade do gradiente em cada uma das 32 classes. Obtendo-se desta forma 81 histogramas direcionais, Figura 6.4.

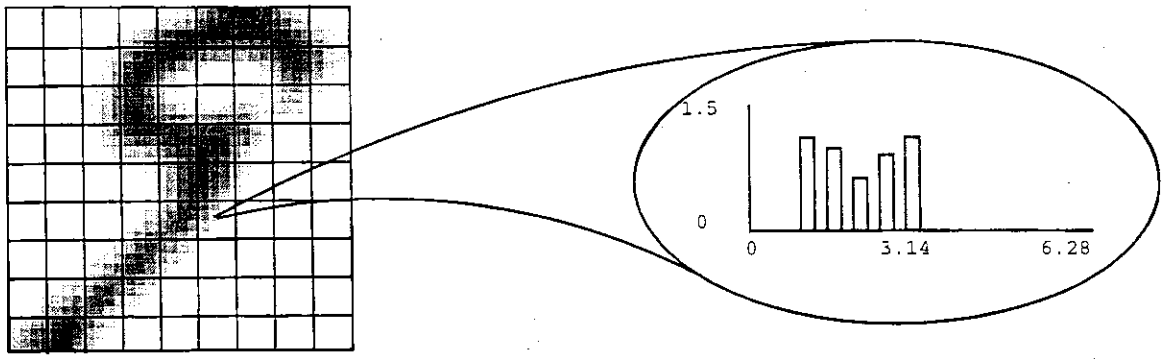


Figura 6.4: Histograma direcional calculado para um bloco da imagem

iii Aplica-se um filtro Gaussiano 5×5 que origina uma nova matriz, cujos histogramas contém valores médios daqueles existentes na matriz orignal. Nesta nova matriz realiza-se uma sub-amostragem a cada duas linhas e duas colunas para reduzir sua dimensão de 9×9 para 5×5, Figura 6.5.

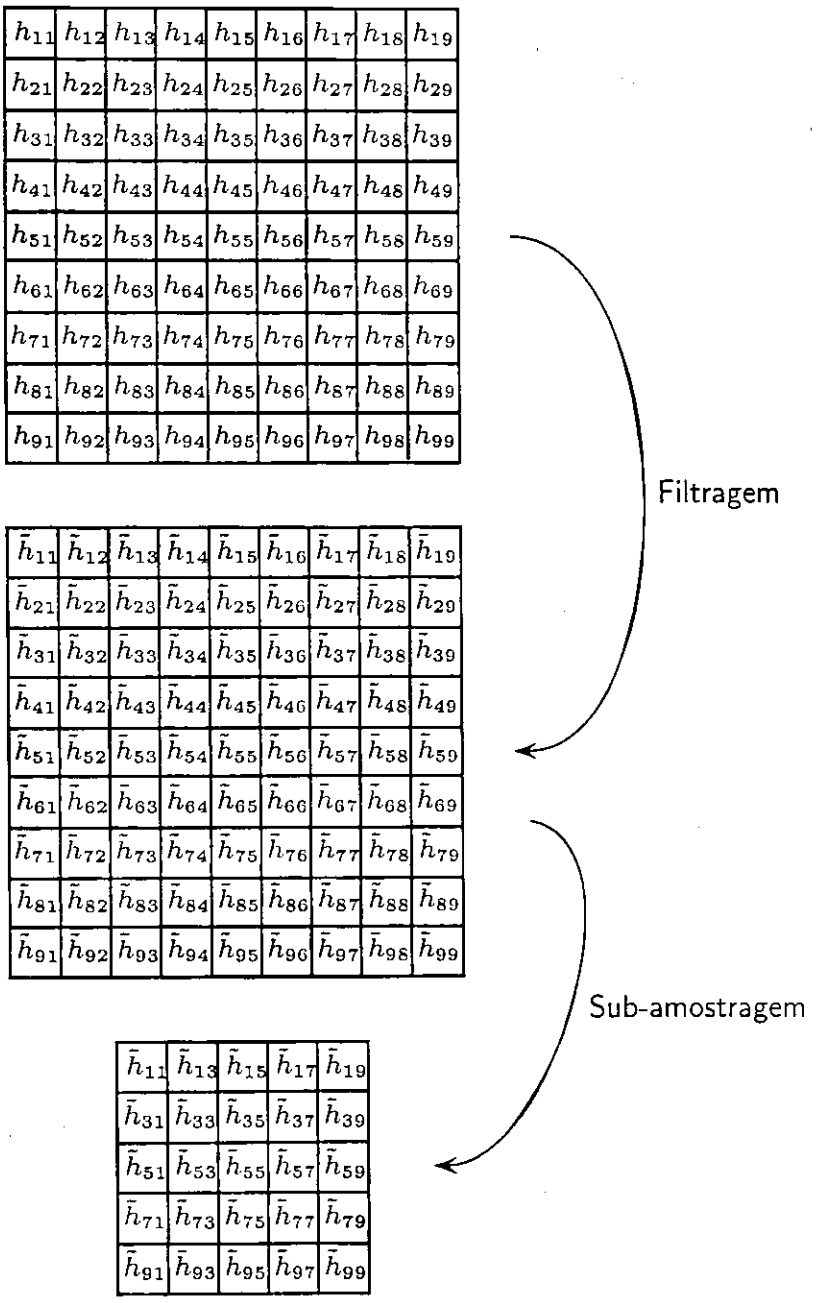


Figura 6.5: Filtragem e sub-amostragem da matriz de histogramas

iv Aplica-se sobre os histogramas resultantes um processo de sub-amostragem similiar ao realizado no passo anterior. Primeiro executa-se uma filtragem linear com uma máscara $w = [1\ 4\ 6\ 4\ 1]^T$, para produzir um histograma de valores médios. Em seguida reduz-se a dimen-

são deste histograma amostrando uma a cada duas classes, produzindo um histograma de 16 classes.

v Compõe-se o vetor de atributos pela concatenação dos 25 histogramas, cada um contendo 16 classes. O vetor de atributos, portanto, possui dimensão 400×1. Aplica-se sobre este vetor a transformação $y = x^{0.4}$ para torná-lo aproximadamente normal.

Foram realizados experimentos com a base NIST avaliando o desempenho de uma função discriminante pseudo-Bayesiana, definido em Kimura *et al.* [68]. O desempenho do classificador fora comparado quando utilizando diversos vetores de atributos obtidos por histogramas de gradiente e curvatura, calculados por diferentes abordagens, e por estratégias combinadas. As taxas de reconhecimento foram bastante elevadas, superiores a 99%, quando utilizando vetores que combinam histogramas de gradiente e curvatura.

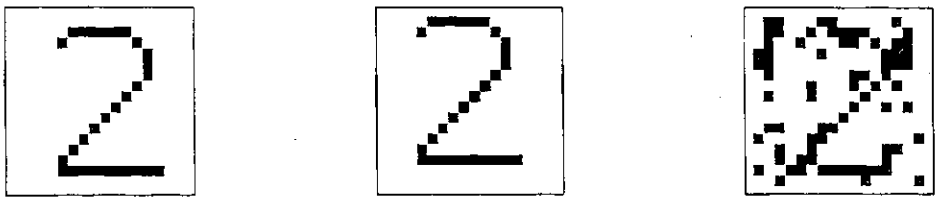
6.1.2 Histogramas direcionais com zoneamento

Estudos na área de visão monstram que a percepção visual humana é fortemente influenciada por características espaciais e temporais de alta freqüência (Cormack [23]). O contorno, onde se manifestam as contribuições das altas freqüências espaciais, desempenha um papel primordial no reconhecimento de formas e interpretação do estímulo visual (Marr [81]), como na identificação do sentido de profundidade, Figura 6.6. Como imagens de caracteres manuscritos são representações visuais de um traçado feito a mão, procurou-se trabalhar com imagens do contorno por se admitir que nele esteja localizado a informação mais relevante para a realização do reconhecimento.

Figura 6.6: Ilustração do sentido de profundidade introduzido pelo contorno

A imagem do contorno, assim como a do esqueleto, está tipicamente relacionada com uma matriz de *pixels* esparsa. Estas matrizes, se consideradas como vetores em um espaço multidimensional, formam um espaço de atributos bastante difícil de ser segmentado em regiões de decisão, dado ao fato de que o espaço de atributos é por sua vez muito esparso. Este fenômeno pode ser observado pela ilustração da Figura 6.7. Neste exemplo uma imagem de esqueleto de um dígito manuscrito é usada como referencial, Figura 6.7(a). A imagem de referência é deslocada no sentido noroeste, Figura 6.7(b), e calcula-se a distância entre as duas. Uma terceira imagem

binária é gerada aleatoriamente com a restrição de que sua distância com relação ao referencial seja igual à da imagem deslocada, Figura 6.7(c). Naturalmente, para um interpretador humano, as imagens das Figuras 6.7(a) e 6.7(b) possuem a mesma interpretação. Entretanto, para um algoritmo de aprendizado de máquina, que baseia-se na segmentação do espaço de atributos em regiões de decisão, estas imagens podem estar tão distantes quanto àquelas das Figuras 6.7(a) e 6.7(c), portanto, podem se inserir em regiões de decisão distintas. Consequentemente, para classificação de imagens de contorno, faz-se necessário proceder a um processo de extração de características que dê suporte à tarefa do classificador, já que a matriz de *pixels* usada como vetor de atributos não assegura boa separabilidade entre as classes.



(a) (b) (c)

Figura 6.7: Imagens equidistantes: $d(a,b) = d(a,c)$

Neste trabalho foi desenvolvido um método que utiliza histograma com algumas simplificações. Uma vez que o método de histograma realiza uma contagem da intensidade do vetor gradiente por classe de variações da fase ou da curvatura, o procedimento de contagem torna-se mais simples já que no contorno, por ser uma região de transição, a intensidade do gradiente é máxima, podendo ser considerada igual em todos os pontos.

Para evitar distorções geométricas introduzidas pela rotina de normalização em escala, optou-se em trabalhar com zonas que são independentes da resolução espacial da imagem. As zonas não possuem dimensão fixa, elas mapeiam uma determinada região, como canto superior direito ou canto inferior esquerdo, etc. Naturalmente, para imagens de dimensões distintas as zonas possuirão dimensões distintas. Entretanto, é esperado que em cada uma delas as proporções de *pixels* com as mesmas intensidades de fase e curvatura sejam aproximadamente iguais desde que elas pertençam a mesma classe. Na imagem do dígito dois, por exemplo, os pontos de curvatura costumam ocorrer nas mesmas posições, as maiores inflexões ocorrem nos cantos superior direito e inferior esquerdo e nas extremidades do traçado, como ilustrado na Figura 6.8. A distribuição dos ângulos do vetor gradiente também possui alguma regularidade, há um abaulamento na parte central e superior, um traçado aproximadamente reto entre o canto superior direito e o canto inferior esquerdo e um traçado reto na base da imagem. Como as zonas não possuem dimensão fixa então a contagem do número de ocorrências de *pixels* por classe de ângulo e curvatura em

cada uma delas é bastante variável: imagens de área grande originam histogramas com frqüências elevadas e imagens de área pequena, histogramas com freqüências menores. Para caracterizar uma imagem sem introduzir variações nos histogramas decorrentes do tamanho, utilizaram-se histogramas de freqüências relativas.

Em função das imagens serem esparsas adotaram-se zonas que cobrem uma região maior, ao contrário daquelas usadas no trabalho de Shi *et al.* , que são bastante refinadas. Os passos para extração de características empregados neste trabalho estão listados a seguir.

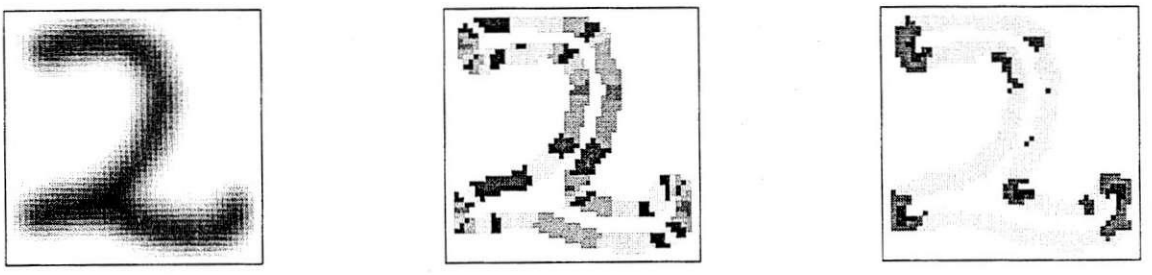
i Aplica-se à imagem binária um algoritmo de extração de bordas a fim de obter as posições dos pontos de contorno. Este passo não requer o uso de um algoritmo sofisticado pois a imagem passará por uma filtragem passa-baixas, de modo que as características extraídas destes pontos levará em conta valores médios calculados em uma vizinhança.

ii Aplica-se o filtro da média 3×3 r vezes sobre a imagem original a fim de se obter uma imagem em nível de cinza (nestes experimentos foi usado $r = 4$).

iii Aplica-se um algoritmo de normalização para limitar a variação dos níveis de cinza ao intervalo $[0, 1]$

iv Aplica-se o filtro de Roberts para obter a magnitude e fase do vetor gradiente.

v Calcula-se a curvatura e fase nos pontos de contorno, Figura 6.8.



(a) Imagem em nível de cinza

(b) Fase

(c) Curvatura

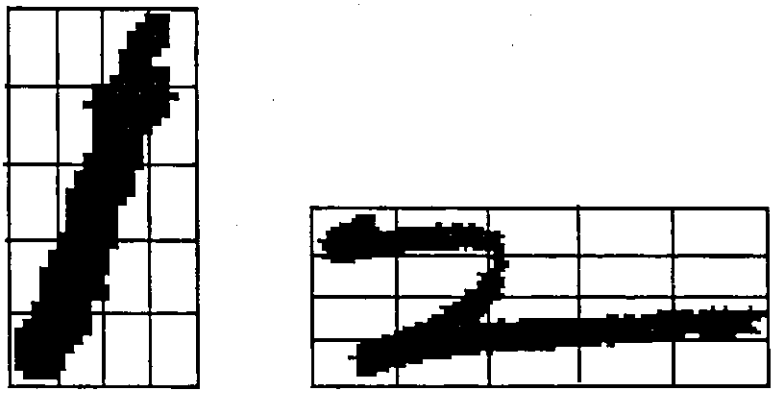
Figura 6.8: Características extraídas do contorno da imagem

A última etapa consiste na construção dos histogramas direcionais e geração do vetor de atributos, que está descrita na seqüência de passos a seguir:

i Realiza-se a extratificação da fase do gradiente em 10 classes de amplitude $\pi/10$. Ângulos separados de π radianos são considerados iguais, logo, considera-se apenas a direção do vetor e ignora-se o sentido.

ii Realiza-se a extratificação dos valores de curvatura em 5 classes.

iii Aplica-se um procedimento de zoneamento dividindo uma imagem $F_{r \times c}$ em 16 blocos (4×4) para imagens aproximadamente quadráticas. Nas imagens retangulares realiza-se uma divisão em 20 blocos, 4 blocos por linha e 5 por coluna caso a imagem tenha uma disposição horizontal $(r > 1.25c)$, ou 5 blocos por linha e 4 por coluna, caso tenha uma disposição vertical $(c > 1.25r)$, Figura 6.9.



(a) (b)

Figura 6.9: Zoneamento aplicado a imagens retangulares

iv Para cada bloco realiza-se a contagem do número de ocorrências de cada uma das classes da fase e curvatura originando histogramas com 15 classes.

v Para cada um dos histogramas, normalizam-se os valores de modo que o número de ocorrências em cada classe fique limitado ao intervalo $[0, 1]$.

vi Realiza-se a concatenação dos histogramas obedecendo a seqüência apresentada abaixo:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	

6.2 Método proposto × características utilizadas

Esta seção mostra o resultado de alguns experimentos que avaliam o comportamento do procedimento de particionamento quando aplicado aos vetores extraídos pela aplicação do método do histograma com zoneamento.

6.2.1 Avaliação do procedimento para obtenção de u_m

A obtenção de u_m pelo critério do elemento de máxima entropia foi avaliada experimentalmente tomando-se um subconjunto de 10000 imagens escolhidas aleatoriamente dentre as 195000 disponíveis para treinamento. O método foi treinado com 16 destes sub-conjuntos ($N = 10000$). Procurou-se avaliar a quantidade de partições originadas quando utilizando os critérios de entropia, erro médio quadrático e busca exaustiva, tal como descrito na Seção 5.3. A Tabela 6.3 mostra a quantidade média e a variância do número de partições obtida quando aplicando cada um destes critérios. Observe que, pelo critério da entropia, a quantidade média de partições originadas é menor que pelo critério do erro médio quadrático, entretanto, a dispersão é maior. Tal como ocorrido com as bases de padrões estudadas no Capítulo 5, aumento da variância se justifica pelo fato de que o raio da hiperesfera $\mathcal{H}$ em que se localiza R_m é função de u_m. Este raio, calculado como a distância média entre u_m e $x \in \mathcal{M}$ tende a ser menor quando empregado o critério da entropia, daí a necessidade de em alguns casos ser originado um número maior de partições para cobrir inteiramente o espaço definido por $\mathcal{M}$.

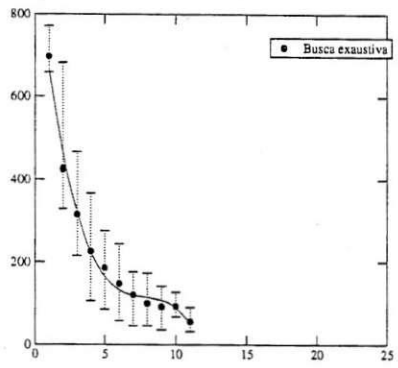
Tabela 6.3: Estatísticas sobre o número de partições originadas por critério de particionamento

Critério	μ	σ_{n-1}
Busca exaustiva	8.31	2.47
Entropia	8.44	6.43
Erro médio quadrático	12.25	1.81

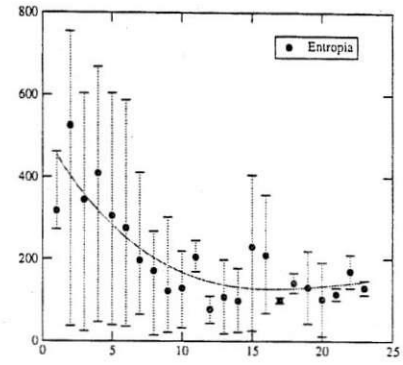
Outra forma de avaliar o critério de definição de u_m é pela contagem do número de padrões inseridos nas partições que são sucessivamente criadas. A contagem do número de ocorrências tende a diminuir à medida que novas partições são criadas. O critério usado na definição de u_m é tanto melhor quanto mais acentuado for este decaimento porque tende a localizar mais precisamente o padrão que contém mais elementos de $\mathcal{M}$ em uma vizinhança. Observe pelo gráfico da Figura 6.10 que o critério da entropia aproxima melhor o gráfico da busca exaustiva, em forma de uma exponencial decrescente, considerado como um modelo de referência.

6.2.2 Avaliação do procedimento para definição de $s_m(\alpha)$

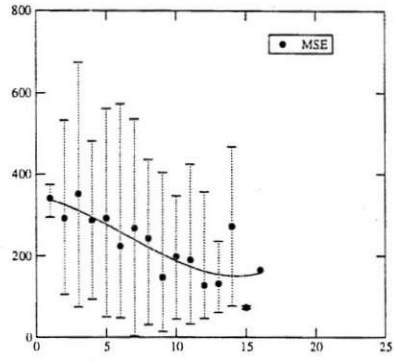
Na avaliação de $s_m(\alpha)$ procurou-se investigar a distribuição dos padrões aprendidos e não aprendidos na hiperesfera $\mathcal{H}$ em que se localiza a partição. A forma de uma partição é tanto melhor quanto maior for a quantidade de instâncias não aprendidas que ela contiver. Num espaço de atributos de grande dimensão, há grande quantidade de instâncias corretamente classificadas próximas de u_m na hiperesfera $\mathcal{H}$, pois há muitas direções em que seu raio se prolonga. A



(a) Busca exaustiva



(b) Entropia



(c) MSE

Figura 6.10: Curva da contagem de padrões por partição gerada

forma da partição deve ser tal que consiga localizar a faixa na qual os padrões não aprendidos se distribuam em $\mathcal{H}$. Para fazer esta avaliação foi feito o seguinte experimento:

1. Selecionou-se um conjunto com 10000 amostras obtidas aleatoriamente dentre as 195000 disponíveis para formar um conjunto de treinamento $\mathcal{T}$

2. Treinou-se um classificador *Perceptron* com cinco ciclos.

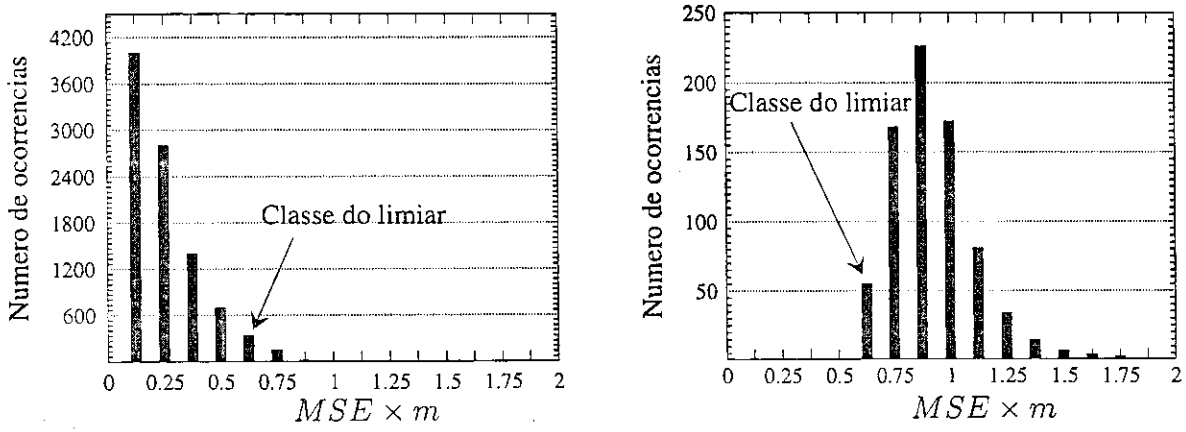
3. Calculou-se u_m pelo critério da entropia.

4. Definiu-se a hiperesfera $\mathcal{H}$ fazendo seu raio igual à distância média entre u_m e os demais padrões em $\mathcal{M}$.

5. Calculou-se uma distribuição de freqüência relacionando a quantidade de amostras por intervalos do erro médio quadrático. O intervalo máximo de variação do erro médio quadrático $[0, 2/n]$ $(n = 10)$ foi dividido em 16 partições disjuntas para formar as classes da distribuição.

6. Repetiu-se os passos 1 a 5 16 vezes a fim de se obter uma média destas distribuições de freqüência.

O gráficos das distribuições médias dos padrões aprendidos e não aprendidos estão ilustrados na Figura 6.11. Como discutido na Seção 4.1.2, os padrões não aprendidos localizam-se acima do valor $0,5/n$, como pode ser visto na Figura 6.11(b). Observe também pelo gráfico da Figura 6.11(a) que a classe que define o limiar de inclusão inclui ainda alguns exemplos corretamente classificados mas ignora uma quantidade substancialmente maior. Esta é uma propriedade bastante atraente porque ignorando um grande número de instâncias que não precisam ser submetidas a um novo ciclo de treinamento reduze-se o custo computacional do método, por outro lado incluindo-se uma pequena quantidade de amostras aprendidas declina-se a influência de elementos expúrios na formação de B_P.



(a) Padrões aprendidos

(b) Padrões não-aprendidos

Figura 6.11: Contagem do número de padrões aprendidos e não-aprendidos em $\mathcal{H}$

Deve-se ressaltar que as distribuições de freqüência apresentadas na Figura 6.11 são específicas da base de padrões e do procedimento de extração de características utilizados. O formato de uma exponencial decrescente da Figura 6.11(a) mostra que a maioria dos padrões podem ser aprendidos satisfatoriamente por um classificador linear. O que se pode considerar um ajuste fino, o aprendizado dos elementos ambíguos localizados próximo às fronteiras entre as classes, depende de uma quantidade de exemplos substancialmente menor. Esta propriedade torna a aplicação do método proposto apropriada pois utilizam-se poucas amostras no ciclo de treinamento seguinte, sem que com isto seja ignorado o conhecimento adquirido na etapa anterior.

6.3 Comparação com outros métodos

O objetivo de muitos estudos na área de aprendizado de máquina é apresentar um novo algoritmo ou variação de um existente que possua um desempenho superior a métodos tradicionais para um domínio específico. Entretanto, a comparação de algoritmos e classificadores realizada através de ensaios experimentais é uma tarefa complexa que, se não for cuidadosamente elaborada, pode conduzir a conclusões estatísticamente incorretas. O surgimento de grande quantidade de artigos apontando a superioridade de um determinado algoritmo sobre outros, especialmente na literatura de redes neurais, fez surgir a preocupação em se analisar criticamente a validade dos procedimentos estatísticos empregados (Salzberg [98]) e a se tentar estabelecer padrões para realização de comparações entre algoritmos e classificadores. Uma das primeiras iniciativas no sentido de fixar uma metodologia de testes a ser empregada em estudos comparativos desta natureza foi o projeto DELVE (*Data for Evaluating Learning in Valid Experiments*) (Rasmussen *et al.* [92]) da Universidade de Toronto. Este projeto consiste de fato em um programa de computador, similar a um *shell* do sistema operacional UNIX, que organiza conjuntos de treinamento, validação e testes e métodos de aprendizado (incluíndo também métodos de regressão) em uma estrutura própria e implementa facilidades para realização de ensaios experimentais. Apesar de sua existência, diversos pesquisadores realizam por conta própria os testes estatísticos necessários. Uma importante fonte de apoio para identificação do tipo de teste a ser empregado é um resumo apresentado por Dietterich [31], em que são analisados cinco testes estatísticos, suas especificidades e situações em que devem ser empregados.

Segundo a análise realizada por Dietterich, deve-se, ao escolher um teste, identificar inicialmente que tipo de objeto procura-se comparar: se um classificador ou um algoritmo de aprendizado. Na literatura em língua portuguesa, o termo algoritmo de aprendizado também é conhecido como indutor (Rezende [93]). Um classificador pode ser interpretado como uma função matemática que dado um vetor de atributos retorna um rótulo ou classe. Um indutor, por sua vez, é um gerador de classificadores que dado um conjunto de treinamento produz uma hipótese ou classificador. A comparação de classificadores leva em consideração essencialmente a capacidade de predição para elementos não-vistos, isto é, procura-se comparar o quanto classificadores já treinados são capazes de acertar em um conjunto com amostras não-vistas. Neste caso, a fonte de variação localiza-se no conjunto de testes. A comparação de indutores procura avaliar a capacidade de um método em produzir um classificador avaliando-se o quanto estes são capazes de acertar em um conjunto com amostras não-vistas. Neste caso a fonte de variação localiza-se também nos conjuntos de treinamento pois procura-se avaliar o quanto os métodos de aprendizado são sensíveis à variabilidade amostral destes dados. O julgamento de dois indutores deve levar em consideração não somente o quanto os classificadores originados podem acertar mas também sua estabilidade em torno deste valor.

Em função do tempo disponível para realização dos experimentos não comportar a avaliação de indutores, pois demandam a realização de várias etapas de treinamento, o estudo comparativo apresentado nesta seção restringe-se a avaliação de classificadores.

6.3.1 Metodologia de testes

Para avaliar quantitativamente os classificadores estudados foi utilizado o coeficiente Kappa (Rossiter [96]), largamente utilizado para comparar o grau de precisão de classificadores em reconhecimento de imagens (Moraes [83], Afonso [2], Giacinto e Roli [48]). Este coeficiente mede o grau de concordância entre observadores. Neste caso, compara-se o grau de concordância entre a saída produzida por um classificador com o rótulo associado ao padrão de entrada. O valor nulo indica que as respostas do classificador não se relacionam com as saídas desejadas, tendo sido geradas ao acaso, enquanto que o valor unitário indica total concordância.

Para calcular o coeficiente Kappa constroe-se uma matriz de erros, cuja notação empregada para representar suas células está apresentada na Tabela 6.4:

Tabela 6.4: Notação empregada para representar matriz de erros

	ω_1	ω_2	$\cdots$	ω_n	Σ
ω_1	n_{11}	n_{12}		n_{1m}	n_{1o}
$\vdots$		$\vdots$		$\vdots$	
ω_n	n_{n1}	n_{n2}		n_{mm}	n_{1o}
Σ	n_{o1}	n_{o2}		n_{om}	N

O coeficiente Kappa, k, é fornecido pela expressão:

$$k = \frac{\theta_1 - \theta_2}{1 - \theta_2} \tag{6.13}$$

com

$$\theta_1 = \frac{\sum_{i=1}^{m} n_{ii}}{N} \tag{6.14}$$

$$\theta_2 = \frac{\sum_{i=1}^{m} n_{io} n_{oi}}{N^2} \tag{6.15}$$

A variância deste coeficiente é dada pela expressão:

$$\sigma^2 = \frac{1}{N}\left(\frac{\theta_1(1-\theta_1)}{(1-\theta_2)^2} + \frac{2(1-\theta_1)(2\theta_1\theta_2 - \theta_3)}{(1-\theta_2)^3} + \frac{(1-\theta_1)^2(\theta_4 - 4\theta_2^2)}{(1-\theta_2)^4}\right) \tag{6.16}$$

com

$$\theta_3 = \frac{\sum_{i=1}^{m} n_{ii}(n_{io} + n_{oi})}{N^2} \quad (6.17)$$

$$\theta_4 = \frac{\sum_{i}^{m} \sum_{j=1}^{m} n_{ij}(n_{io} + n_{oj})^2}{N^3} \quad (6.18)$$

6.3.2 Algoritmos implementados

No estudo comparativo foram testados redes neurais MLP, um classificador 5-NN, um agrupamento usando o método *boosting* e o Sistema I. As redes neurais e o agrupamento usando *boosting* foram implementadas a partir da biblioteca *Torch* (Collobert *et al.* [21]), cuja arquitetura e demais parâmetros livres estão listados na Tabela 6.5. Para o treinamento do sistema usando *boosting* foram usadas 5 redes neurais com estas mesmas configurações.

Tabela 6.5: Parâmetros usados para construção e treinamento das redes neurais

Parâmetro	Valor
Número de camadas	3
Número de neurônios por camada	$375 - 256 - 10$
Camada de saída	*softmax*
Função de transferência	$tanh^{-1}(\cdot)$
Número máximo de ciclos de treinamento	20

O sistema de agrupamento de classificadores com redes Bayesianas empregou classificadores *Perceptron* e 3-NN e utilizou probabilidades como entrada para o sistema de inferência e no tratamento dos casos observados (Sistema I). Para treinamento do classificador global utilizou-se apenas um único ciclo e nas redes neurais originadas a partir do segundo nível do diagrama de classificadores, realizou-se um treinamento com 50 ciclos. A arquitetura do sistema obtido com base neste experimento está ilustrada na Figura 6.12.

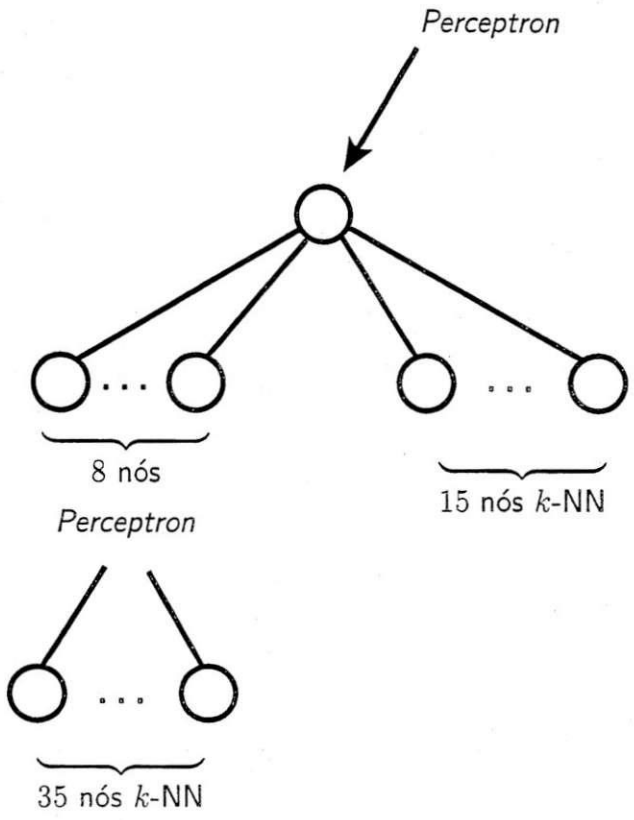


Figura 6.12: Diagrama de classificadores originado do treinamento

Os classificadores k-NN foram implementados usando como medida de dissimilaridade a distância euclidiana. Utilizou-se um algoritmo de busca exaustiva para localização dos k vizinhos mais próximos.

6.3.3 Resultados com a base reduzida

Os classificadores foram treinados com 60000 imagens obtidas aleatoriamente dos conjuntos *hsf_0*, *hsf_1*, *hsf_2* e *hsf_3* e testados com 60089 imagens do conjunto *hsf_7*, isto é, todas imagens disponíveis no conjunto. As taxas de reconhecimento, bem como os valores dos coeficientes e variância da estatística Kappa, estão apresentadas na Tabela 6.6.

Tabela 6.6: Taxas de reconhecimento obtida com a base NIST

Classificador	Taxa de reconhecimento (%)	Kappa (%)	Variância
MLP	97,94	97,7	$3,39 \times 10^{-7}$
Boosting	98,08	97,86	$2,46 \times 10^{-7}$
5-NN	98,57	98,4	$2,9 \times 10^{-7}$
Sistema I	**98,89**	98,72	$2,25 \times 10^{-7}$

A fim de quantificar numericamente as diferenças expressas na Tabela 6.6, as distâncias entre os coeficientes foram tabulados em escores normalizados, calculados em função da quantidade de desvios-padrão associado a cada coeficiente, originando desta forma a matriz apresentada na Tabela 6.7. Esta matriz não é simétrica, pois os desvios-padrão associados a cada classificador são distintos. Cada célula d_{ij} desta matriz é calculada como:

$$d_{ij} = \frac{|k_i - k_j|}{\sigma_i} \tag{6.19}$$

Tabela 6.7: Distâncias normalizadas entre os coeficientes

	Mlp	Boosting	5-NN	Sistema I
Mlp	0	2.43	10.72	**16.29**
Boosting	2.52	0	8.59	**14.37**
5-NN	12.86	9.95	0	**6.69**
Sistema I	**22.27**	**18.95**	**7.62**	0

Os dados da Tabela 6.7 revelam que não há entrelaçamento entre a classe em que se localiza o coeficiente do método proposto com as demais pois, pode-se observar na última linha e coluna da tabela, que as distâncias normalizadas superam pelo menos 6 desvios-padrão. Embora em alguns textos (Evans [34], Afonso [2]) a análise do coeficiente Kappa seja realizada em conjugação com um teste Z, a rigor não se pode desenvolver uma análise à luz de um teste de significância pois a distribuição do coeficiente não é Gaussiana. Apesar desta limitação, ainda assim é possível admitir que há diferenças entre o coeficiente associado ao Sistema I e os demais, face aos resultados apresentados na Tabela 6.7. Isto permite concluir que, no experimento realizado, o método proposto superou os demais métodos comparados. É importante destacar que não se deve interpretar estes resultados como uma realidade estática, o que levaria ao julgamento de que as redes neurais MLP, e consequentemente o agrupamento com *Boosting*, são pouco apropriados para serem utilizados com o método de extração de características desenvolvido. Na criação/treinamento de uma rede neural há diversos parâmetros livres que podem ser configurados de modo a melhorar seu desempenho, tais como: o número de camadas escondidas e a quantidade de neurônios em cada uma delas, a utilização de decaimento de peso (*weight decay*), momentos, condicionamento dos dados de entrada, utilização de arquitetura modular, etc. Certamente outra arquitetura neural conduziria a um resultado mais acurado desde que devidamente configurado para a aplicação alvo. Deve ser destacado que a principal vantagem do método proposto foi ter obtido resultados competitivos utilizando-se um treinamento que exige menor esforço computacional, já que se trabalhou com classificadores lineares ou com k-NN em regiões limitadas do espaço de atributos, e que requer pouca interferência de um especialista humano, pois na configuração destes

classificadores há poucos parâmetros livres que precisem ser ajustados.

6.3.4 Resultados com a base expandida

Realizou-se um experimento em que se testou o Sistema I utilizando-se como classificador global uma rede *perceptron* e como classificadores locais redes *percepton* e 1-NN. Empregou-se no treinamento 180000 imagens dos conjuntos *hsf_012* e *3*, o que representa $92,3\%$ das imagens disponíveis para treinamento. Os testes foram realizados com todas as imagens do conjunto *hsf_7*. Os resultados deste experimento estão colocados na Tabela 6.3.4, em que se apresenta a matriz de erros.

Tabela 6.8: Matriz de erros — Treinamento com 180000 amostras

Saída	Saída produzida										
desejada	0	1	2	3	4	5	6	7	8	9	(%)
0	5802	1	5	0	3	1	11	1	13	7	0.993
1	0	6491	0	2	3	1	4	2	10	2	0.996
2	6	31	5930	7	6	1	3	9	7	0	0.988
3	4	12	5	5963	0	6	0	14	2	2	0.993
4	12	7	5	0	5834	3	4	9	9	10	0.990
5	7	3	0	19	1	5637	4	1	6	1	0.993
6	17	2	2	0	8	3	5870	0	5	0	0.994
7	1	15	7	19	3	3	0	6192	5	17	0.989
8	27	1	10	9	9	10	3	2	5818	11	0.986
9	17	4	3	17	6	19	1	24	14	5976	0.983
μ											99,04

Comparando-se a taxa de reconhecimento obtida neste experimento com outras citadas na literatura, como apresentadas na Tabela 6.2, pode-se observar que o método proposto mostra-se competitivo e revela uma boa adequação para a tarefa de reconhecimento de dígitos manuscritos.

6.4 Conclusão

Neste capítulo foi realizado um estudo de caso da aplicação do método proposto a um problema de reconhecimento de imagens de dígitos manuscritos. Para realizar este estudo desenvolveu-se um método de extração de características baseado na imagem do contorno do caracter, Esta abordagem tem uma inspiração no modelo biológico. A visão dos seres-humanos é seletiva às componentes espaciais de alta freqüência, que são fundamentais para reconhecimento de formas e do sentido de profundidade. Como estes componentes se manifestam nos pontos de contorno

admite-se que estes pontos contenham toda informação necessária para o reconhecimento deste tipo de imagem, que são imagens monocromáticas portanto desprovidas de cor, textura ou outra característica relevante neste contexto. Procurou-se conciliar a informação extraída do contorno com um procedimento de zoneamento, imitando desta forma os campos receptivos do sistema visual humano. Os vetores de atributos obtidos por este processo de extração de características foram utilizados pelo sistema de reconhecimento apresentado no Capítulo 4. Os resultados experimentais demonstraram que para este tipo de aplicação o método proposto é bastante apropriado. Foi obtido uma taxa de reconhecimento competitiva tanto em relação a outros métodos comparados na Seção quanto em relação ao que a literatura especializada apresenta. Estes resultados são conseqüência da boa separação inter-classes que o extrator de características propicia e da boa adequação do sistema de reconhecimento aplicado ao problema em foco.

Capítulo 7

Conclusões e perspectivas futuras

7.1 Conclusões

A proposta deste trabalho foi apresentar um método de classificação de padrões que realizasse o particionamento do espaço de atributos em regiões entrelaçadas associando a cada uma destas um classificador. Para realizar uma predição única, as saídas dos classificadores deveriam ser combinadas atendendo a um critério de otimalidade. Propôs-se usar uma rede Bayesiana como agrupador de classificadores. Apesar de potencialmente promissor como estimador não-paramétrico de uma distribuição de probabilidade, ou como sistema especialista baseado em conhecimento incerto, o uso de redes Bayesianas é ainda insipiente na área de combinação de classificadores, que tem experimentado intensa movimentação desde início dos anos 2000, a exemplo do surgimento de uma conferência internacional (*International Workshop on Multiple Classifier Systems*), em 2000, e de uma revista especializada (*Information Fusion*), também em 2000. Deve-se ressaltar que em alguns artigos há utilização do classificador Bayesiano, que trata-se de um caso particular de uma rede Bayesiana como explicado no Capítulo 2, entretanto, a construção de um sistema com vários níveis de decisão como o proposto neste trabalho não foi encontrada no levantamento bibliografico realizado. Explorou-se a perspectiva de que na área de combinação de classificadores, um combinador baseado em treinamento aprende a regra de combinação atendendo a um critério de otimalidade podendo ser a aproximação de uma distribuição de probabilidade, o que torna apropriado a utilização da rede Bayesiana. O cerne do método proposto está no processo de particionamento do espaço de atributos. É através deste processo que se originam os classificadores locais, arranjados em uma estrutura chamada diagrama de classificadores da qual deriva-se a estrutura da rede Bayesiana. Nos ensaios experimentais realizados nos Capítulos 5 e 6 demonstrou-se que a estratégia de particionamento adotada, conciliada com o combinador construído a partir desta, resulta num método de classificação de padrões que realiza predições em geral mais precisas do que a dos classificadores constituintes. Esta propriedade foi usada para criar um sistema composto por classificadores fracos, que tanto demandavam menor esforço

computacional para serem treinados, quanto menor interferência de um especialista humano na configuração dos parâmetros livres. Para o treinamento de grandes bases de dados, como tratado no Capítulo 6, a adoção do método de combinação proposto revelou vantagens tanto por apresentar resultados competitivos quanto por realizar uma distribuição balanceada no número de amostras necessárias para treinar cada classificador: muitas amostras empregadas no treinamento do classificador global, que requer menor esforço computacional, e poucas amostras empregadas no treinamento dos classificadores locais.

No estudo de caso na área de reconhecimento de imagens de caracteres manuscritos foi apresentado um algoritmo de extração de características inspirado no modelo biológico. A utilização deste algoritmo, bem como do método de agrupamento de classificadores proposto, foi avaliado experimentalmente no reconhecimento de dígitos manuscritos da base NIST. Os resultados destes experimentos revelaram-se competitivos com relação aos referenciados na literatura.

7.2 Perspectivas futuras

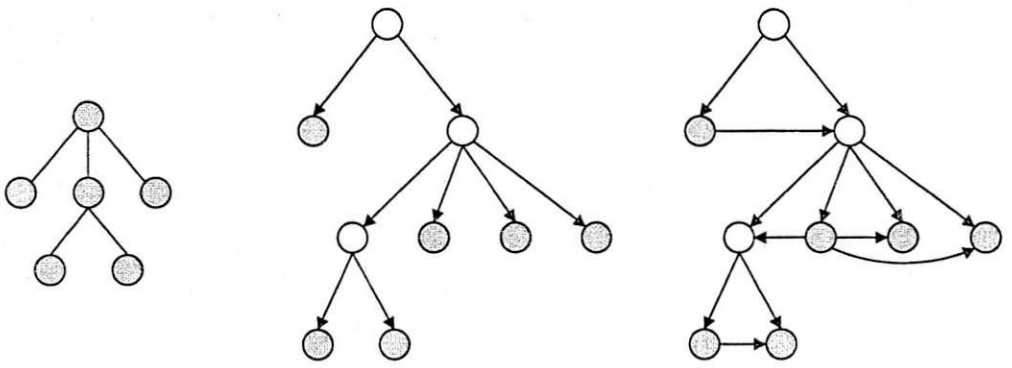
Um trabalho científico nunca é um produto acabado que se encerra em si mesmo. Ao contrário, ao se investigar um tema abrem-se perspectivas para novas explorações e desdobramentos que só se descobrem a partir da investigação inicial. Este trabalho realizou a cobertura de vários aspectos de um problema complexo mas não foi suficiente para cobrir alguns outros que também se mostram potencialmente promissores. Alguns destes estão comentados a seguir:

Aprendizado incremental A arquitetura do sistema proposto como um todo é modelada dinamicamente, isto é, novos nós e conexões são criados no diagrama de classificadores a fim de suportar a complexidade dos dados apresentados. Esta propriedade pode ser usada para realizar um treinamento incremental que viabilize o aprendizado de novos padrões sem perder o conhecimento já adquirido. Em um sistema conexionista o conhecimento, armazenado nas conexões, é perdido sempre que os valores das conexões é alterado, entretanto, em um sistema modular o treinamento incremental pode ser feito localmente em unidades isoladas não comprometendo, portanto, a manutenção do conhecimento armazenado em outros nós. Além disto, novos nós no diagrama de classificadores são criados quando os padrões apresentados diferem significativamente dos modelos associados aos nós já existentes.

Na prática, a aquisição de conhecimento incremental pode ser substituida por uma nova etapa de treinamento com base em um conjunto formado pela união dos conjuntos com antigas e novas instâncias. Entretanto, há situações em que esta estratégia possue um custo proibitivo, por exemplo, em um sistema em produção que precise aprender com os

erros deve-se evitar interromper uma execução em andamento, pois o tempo necessário para treinar todas as amostras pode ser muito longo.

Revogação da hipótese de independência A fim de aproveitar o mecanismo de propagação de mensagens, o procedimento de aquisição de B_S cria, em cada nível da rede, nós que não possuem ligações entre si, são portanto independentes dado o estado do nó ancestral. A hipótese de independência pode ser revogada em prol de um modelo um pouco mais complexo e preciso. Pode-se usar, por exemplo, um agrupamento de redes do tipo *TAN* (*Tree Augmented Bayesian Network*) ([41]), Figura 7.1, que como citado na literatura são modelos mais precisos da distribuição subjacente e cuja complexidade do algoritmo de aquisição possui complexidade polinomial.



(a) Diagrama de classificadores

(b) Rede Bayesiana

(c) Agrupamento de *TAN's*

Figura 7.1: Redes Bayesianas construídas a partir do diagrama de classificadores

Utilização de mensagens π Uma rede Bayesiana é um sistema de inferência que possui algumas semelhanças com o modelo empregado pelos seres-humanos. Vale destacar dois pontos importantes neste contexto. Primeiro, a dimensão do vetor de entrada não possui tamanho fixo, as evidências fornecidas à rede podem ocorrer em qualquer nó e em número variável, não necessariamente em um conjunto de nós que formam uma camada de entrada. Ao trabalhar com entradas de dimensão fixa introduz-se distorções decorrentes da normalização em escala, o que pode ser evitado quando a entrada do sistema possui dimensão variável. O segundo ponto destacado é a potencial contribuição que pode ser obtida com o conteúdo das mensagens π, que são associadas a diagnósticos ou conclusões. Na imagem da Figura 7.2 vê-se nos vazios entre os quadrados adjacentes uma formação nebulosa ilusória em tons de cinza. Esta ilusão é causada pelo preenchimento dos espaços vazios pelos prolongamentos dos segmentos de reta dos quadrados vizinhos. Semelhantemente ao modelo biológico, mensagens π enviadas por nós ancestrais induzem os nós descendentes a atu-

alizarem seus vetores de crença, eventualmente mudando o valor do estado mais provável. A retro-alimentação destes valores leva os nós em níveis superiores a gerar interpretações baseadas em evidências imaginárias, semelhantes àquelas produzidas pelo sistema de interpretação humano. A construção de um sistema de reconhecimento de imagens usando múltiplos classificadores combinados com uma rede Bayesiana pode ser empregada de uma forma em que os classificadores sejam especializados em regiões da imagem ao invés de serem especializados em regiões do espaço de atributos. Neste contexto, as mensagens π podem ser usadas para identificar as regiões que não contribuem efetivamente para o processo de reconhecimento e que demandam a utilização de classificadores mais especializados.

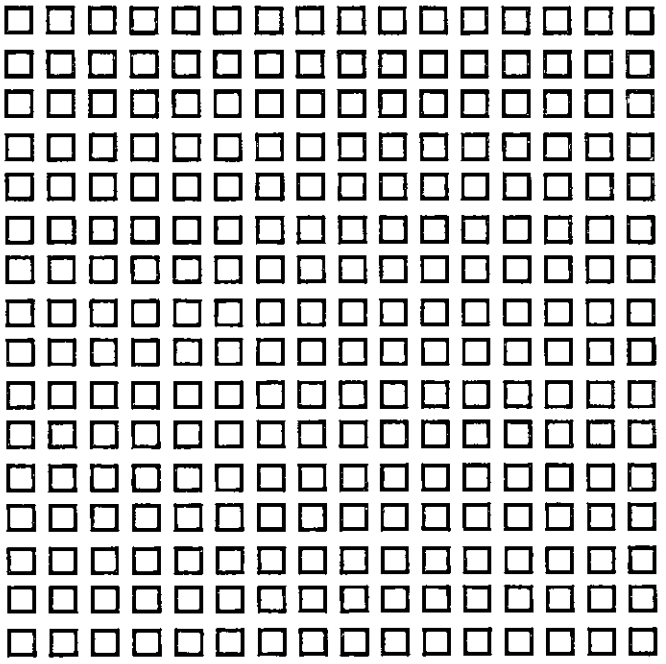


Figura 7.2: Ilusão originada pelo prolongamento de segmentos de reta

Investigação mais extensa do algoritmo de extração de características O processo de extração de características proposto no Capítulo 6 mostrou-se promissor apesar de não ser este o foco de atenção principal do trabalho desenvolvido. Em razão de ter se mostrado efetivo como ferramenta para melhorar a sperabilidade inter-classes uma investigação mais extensa deve ser conduzida. Algumas investidas neste sentido podem ser iniciadas pelo estudo da quantidade e disposição das zonas e pelo estudo de estratégia de composição do vetor de atributos que produza resultados mais promissores do que o obtido pela simples concatenação das características fase e curvatura.

Referências Bibliográficas

[1] B. Abramson. The design of belief network-based systems for price forecasting. *Computers and Eletronic Engineering*, 20:163–180, 1994.

[2] R.L. Afonso. Combinação de classificadores atuando em diferentes atributos de imagens multispectrais de ressonância magnética nuclear de frutas com avaliação de desempenho pelo coeficiênte kappa. Master's thesis, Universidade Federal de São Carlos, 2002.

[3] D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[4] C. Andrieu, N. de Freitas, A. Doucet, and M.I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

[5] N. Arica and F.T. Yarman-Vural. An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 31(2):216–232, 2001.

[6] C.H. Atkeeson, A.W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

[7] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

[8] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[9] L. Bottou and V.N. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.

[10] R.R. Bouckaert. Probablistic network construction using the minimum description length. Technical Report RUU-CS-94-27, Ultrecht University, Departament of Computer Science, Utrecht University, The Netherlands, 1994.

[11] I. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Klower Academic Publishers, 1984.

[12] L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.

[13] J.S. Bridle. *Neurocomputing: Algorithms, Architectures and Applications*, chapter Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition, pages 227–236. Springer-Verlag, 1989.

[14] A.S. Britto Jr. *A Two-Stage HMM-Based Method for Recognizing Handwritten Numeral Strings*. PhD thesis, PUC-PR, Curitiba - Brazil, 2001.

[15] W. Buntine. A guide to the literature on learning probabilistic networks from data. *IEEE Transactions On Knowledge And Data Engineering*, 8:195–210, 1996.

[16] W.L. Buntine. Learning with graphical models. Technical Report FIA-94-02, NASA Ames Research Center, 1994.

[17] A.L. Carneiro. Aprendizado automático em redes bayesianas. Master's thesis, Universidade de Brasília, 1999.

[18] K. Cheung and D. Yeung. A bayesian framework for deformable pattern recognition with application to handwritten character recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12), 1998.

[19] S. Cho and J.H. Kim. Combining multiple neural networks by fuzzy integral for robust classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(2):380–384, 1995.

[20] S. Cho and J.H. Kim. Multiple network fusion using fuzzy logic. *IEEE Transactions on Neural Networks*, 6(2):497–501, 1995.

[21] R. Collobert, S. Bengio, and J. Mariéthoz. Torch: a modular machine learning software library. IDIAP-RR 46, IDIAP, 2002.

[22] G.F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[23] L.K. Cormack. Computation models of early human vision. In Al Bovik, editor, *Handbook of Image and Video Processing*, pages 271–287. Academic Press, 2000.

[24] S.E.N. Correia. Validação e otimização de um sistema de reconhecimento de numerais manuscritos usando wavelets. Technical Report RT00292/02, Depto. de Eng. Elétrica — Universidade Federal da Paraíba, Campina Grande - PB, 2001.

[25] S.E.N. Correia. Reconhecimento de caracteres numéricos manuscritos baseado em uma rede neural com o número de nós de saída reduzido. Technical Report RT00355/03, Universidade Federal de Campina Grande, Miniblio-Copele, 2003.

[26] T.M. Cover and P.E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[27] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[28] P. Dayan, G.E. Hinton, R.M. Neal, and R.S. Zemel. The helmholtz machine. *Neural Computation*, 7:889–904, 1995.

[29] A.P. Depster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.

[30] T.G. Dietterich. Machine-learning research. *Artifitial Intelligence Magazine*, 18(4):97–136, 1997.

[31] T.G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924, 1998.

[32] J. Dong. Comparison of algorithms for handwritten numeral recognition. Technical report, CENPARMI, Concordia University, 1999.

[33] R.O. Duda and P.E. Hart. *Pattern Classification*. Wiley-Interscience, New York, 2nd edition, 2000.

[34] F. Evans. An investigation into the use of maximum likelihood classifiers, decision trees, neural networks and conditional probabilistic networks for mapping and predicting salinity. Master's thesis, School of Computing, Curtin University of Technology, Western Australia, 1998.

[35] K.J. Ezawa and T. Schuermann. Fraud/uncollectable debt detection using a bayesian network based learning system: A rare binary outcome with mixed data structures. In P. Besnard and S. Hanks, editors, *Proceedings of the Elevnth Conference on Uncertainty in Artificial Intelligence (UAI'95)*, pages 157–166. Morgan Kaufmann, 1995.

[36] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the 13th International Conference*, pages 148–156, 1996.

[37] B.J. Frey. *Bayesian Networks for Pattern Classification, Data Compression and Channel Coding*. PhD thesis, Unversity of Toronto, 1997.

[38] P.W. Frey and D.J. Slate. Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6, 1991.

[39] J.H. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19(1):1–141, 1991.

[40] J.H. Friedman. Local learning based on recursive covering. Technical report, Department of Statistics, Stanford University, 1996.

[41] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 27:131–163, 1997.

[42] N. Friedman and L. Getoor. Efficient learning using constrained sufficient statistics. In *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics (AISTATS-99)*, 1999.

[43] A. Garg, V. Pavlović, and T.S. Huang. Bayesian networks as ensemble of classifiers. In *Proceedings of the 16th International Conference on Pattern Recognition*, pages 779–784, October 2002.

[44] D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and bayesian multinets. *Artificial Intelligence*, 82:45–74, 1996.

[45] S. Geman, E. Bienstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[46] S. Geman and D. Geman. Stochastic relaxion, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

[47] Z. Ghahramani and M.I. Jordan. Learning from incomplete data. Technical Report A.I. Memo No. 1509, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1994.

[48] G. Giacinto and F. Roli. Dynamic classifier based on multiple classifier bahaviour. *Pattern Recognition*, 34(9):1879–1881, 2001.

[49] H. Gish. A probabilistic approach to the understanding and training of neural network classifiers. In *Proc. IEEE Int. Conf. Acoustic, Speech, Signal Processing*, pages 1361–1364, 1990.

[50] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[51] P.J. Grother. Nist special database 19 - handprinted forms and characters database. Technical report, National Institute of Standards and Technology (NIST), 1995.

[52] P.W. Hamilton, N. Anderson, P.H. Bartels, and D. Thompson. Expert system support using bayesian belief networks in the diagnosis of fine needle aspiration biopsy specimens of the breast. *Journal of Clinical Pathology*, 47:329–336, 1994.

[53] L.K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

[54] S. Hashem and B. Schmeiser. Improving model accuracy using optimal linear combinations of trained neural networks. *IEEE Transactions on Neural Networks*, 6(3):792–794, 1995.

[55] S. Haykin. *Neural Networks. A Comprehensive Foundation.* Prentice Hall, 2nd. edition edition, 1998.

[56] D. Heckerman. *Learning in Graphical Models*, chapter A Tutorial on Learning with Bayesian Networks. MIT Press, 1999.

[57] E. Herskovits. *Computer-based probabilistic-network construction.* PhD thesis, Medical Informatics, Stanford University., 1991.

[58] Y.S. Huang and C.Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):90–94, 1995.

[59] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

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

[61] F. V. Jensen. *Bayesian Networks and Decision Graphs.* Springer, 2001.

[62] D.M. Johannes. *One-class classification.* PhD thesis, Delft University of Technology, 2001.

[63] R.A. Johnson and D.W. Wichern. *Applied Multivariate Statistical Analysis.* Prentice Hall, 1998.

[64] M.I. Jordan. Why the logistic function? a tutorial discussion on probabilities and neural networks. Technical Report 9503, Massachusetts Institute of Technology, 1995.

[65] M.I Jordan and R.A Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

[66] K.G.Olesen, U.Kjaerulft, F.Jensen, F.V.Jensen, B.Falck, S.Anreassen, and S.K.Andersen. A muni network for the median nerve - a case study on loops. *Applied Artificial Intelligence*, 3:385–403, 1989.

[67] J. Kim, K. Seo, and K. Chung. A systematic approach to classifer selection on combining multiple classifiers for handwritten digit recognition. In *Proceendings of the Fourth International Conference on Document Analysis and Recognition (ICDAR'97)*, volume 2, pages 459–462, 1997.

[68] F. Kimura, T. Wakabayashi, and Y. Miyake. On feature extraction for limited class problem. In *Proceedings of the 13th ICPR*, volume II, pages 191–194, 1996.

[69] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[70] A.N. Kolmogorov. *Foundations of the Theory of Probability*. Chelsea, Nova York, 1950.

[71] P. Krause. Learning probabilistic networks. Technical report, Philips Research Labs, Redhill, England, 1998.

[72] W. Lam and F. Bachus. Learning bayesian networks. an approach based on the MDL principal. *Computational Intelligence*, 10(3):269–293, 1994.

[73] S.L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201, 1995.

[74] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structure and their application to expert systems. *Journal of the Royal Statistical Society B.*, 50(2):157–224, 1988.

[75] Y. LeCun, Y. Bottou, L. Bengio, and P. Haffner, editors. *Gradient-Based Learning Applied to Document Recognition*, 1998.

[76] D. A. Lelewer and D. S. Hirschberg. Data compression. *ACM Computing Surveys*, 19(3):261–296, 1987.

[77] M. Leshno, V.Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861–867, 1991.

[78] X. Li. Simultaneous approximatins of multivariate functions and their derivatives by neural networks with one hidden layer. *Neurocomputing*, 12:327–343, 1996.

[79] X. Lin, X. Ding, and Y. Wu. Handwritten numeral recognition using mfnn-based multiexpert combination strategy. In *Proceendings of the Fourth International Conference on Document Analysis and Recognition (ICDAR'97)*, volume 2, pages 471–474, 1997.

[80] C.L. Liu and H. Sako, H. Fujisawa. Performance evaluation of pattern classifiers for handwritten character recognition. *International Journal on Document Analysis and Recognition*, 4:191–204, 2002.

[81] D. Marr. *Vision*. W.H. Freeman and Company, 1980.

[82] P. McCullagh and J.A. Nelder. *Generalized Linear Models*. Chapman and Hall, 1983.

[83] R.M Moraes. *Uma arquitetura de sistemas especialistas nebulosos para classificação de imagens utilizando operadores da morfologia matemática*. PhD thesis, Instituto de Pesquisas Espaciais - INPE, 1998.

[84] R.M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, University of Toronto, Department of Computer Science, 1993.

[85] R.E. Neapolitan. *Learning Beaysian Networks*. Pearson Prentice Hall, 2004.

[86] L.E.S. Oliveira. *Automatic Recognition of Handwritten Numerical Strings*. PhD thesis, Université du Québec, 2003.

[87] J. Pearl. Fusion, propagation and structuring in belief networks. *Artificial Intelligence*, 29(3):241–288, 1986.

[88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Academic Press, 1997.

[89] J. Peng and B. Bhanu. Local discriminative learning for pattern recognition. *Pattern Recognition*, 34:139–150, 2001.

[90] L.R. Rabiner and R.W. Schafer. *Digital Process of Speech Signal*. Pearson Education, 1978.

[91] M. Ramoni and P. Sebastiani. Robust learning with missing data. Technical Report KMI-TR-28, Knowledge Media Institute, 1996.

[92] C.E. Rasmussen, R.M. Neal, G.E. Hinton, and D. van Camp. The delve manual. Technical report, University of Toronto, 1996.

[93] S.O. Rezende. *Sistemas Inteligentes Fundamentos e Aplicações*. Manoele, 2003.

[94] J. Rissanen. Modeling by Shortest Data Description. *Automatica*, 14:465–471, 1978.

[95] G. Rogova. Combining the results of several neural network classifiers. *Neural Networks*, 7(5):777–781, 1994.

[96] D.G. Rossiter. Statistical methods for accuracy assessment of classified thematic maps. Technical report, International Institute for Geo-Information Science and Earth Observation, Enschede, NL, April 2004.

[97] D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley, and B.W. Suter. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1(4):296–298, 1990.

[98] S.L. Salzberg. On comparing classifiers: A critique of current research and methods. *Data Mining and Knowledge Discovery*, 1:1–12, 1999.

[99] T. Saul, L.K. anad Jaakkola and M.I. Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.

[100] R. Schalkoff. *Pattern Recognition. Statistical Structural and Neural.* Wiley, New York, 1992.

[101] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.

[102] M. Shi, Y. Fujisawa, T. Wakabayashi, and F. Kimura. Handwritten numeral recognition using gradient and curvature of gray scale image. *Pattern Recognition*, 35:2051–2059, 2002.

[103] M. Singh. *Learning Bayesian networks for solving real-world problems.* PhD thesis, University of Pennsylvania, Department of Computer and Information Science, 1998.

[104] M. Singh and Provan G.M. A comparison of induction algorithms for selective and non-selective bayesian classifiers. In A. Prieditis and S. Russel, editors, *Proceedings of he Twelfth International Conference on Machine Learning*, pages 497–505. Morgan Kaufmann, 1995.

[105] M. Singh and M. Valtorta. Construction of bayesian network structures from data: a brief survey and an efficient algorithm. *International Journal of Approximate Reasoning*, 12:111–131, 1995.

[106] S.L.Zabell. W. e. Johnson's sufficientness postulate. *Annals of Statistics*, 10(4):1091–1099, 1982.

[107] Joe Suzuki. Learning bayesian belief networks based on the minimum description length principle: an efficient algorithm using the B&B technique. In *Proc. 13th International Conference on Machine Learning*, pages 462–470. Morgan Kaufmann, 1996.

[108] R.Y. Theo and R.A. Shinghal. A hybrid classifier for recognizing handwritten numerals. In *Proceendings of the Fourth International Conference on Document Analysis and Recognition (ICDAR'97)*, volume 2, pages 283–287, 1997.

[109] J. Tian. A branch-and-bound algorithm for MDL learning bayesian networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, pages 580–588. Morgan Kaufmann, 2000.

[110] O.D. Trier, A.K. Jain, and T. Taxt. Feature extraction methods for character recognition - a survey. *Pattern Recognition*, 29(4):641–662, 1996.

[111] K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognation*, 29(2):341–348, 1996.

[112] V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[113] P. Vitanyi and M. Li. Minimum description length induction, bayesianism, and kolmogorov complexity. *IEEE Transactions on Information Theory*, 46(2):446–464, 2000.

[114] S.R. Waterhouse. *Classification and Regression using Mixtures of Experts*. PhD thesis, Department of Engineering, University of Cambridge, 1997.

[115] A. Webb. *Statistical Pattern Recognition*. John Wiley and Sons, 2002.

[116] L. Xu and A. Krżyzak. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(3):418–435, 1992.

[117] B. Zhang, M. Fu, and H. Yan. A modular classification scheme with elastic net models for handwritten digit recognition. In *Proceendings of the Fourteenth International Conference on Pattern Recognition (ICPR'98)*, pages 1859–1861, 1998.