MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
CENTRO ESTADUAL DE PESQUISAS EM SENSORIAMENTO REMOTO E METEREOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM SENSORIAMENTO REMOTO

Dissertação de Mestrado

EXTRAÇÃO DE FEIÇÕES EM DADOS IMAGEM COM ALTA DIMENSÃO POR OTIMIZAÇÃO DA DISTÂNCIA DE BHATTACHARYYA EM UM CLASSIFICADOR DE DECISÃO EM ÁRVORE

Por

Denis Altieri de Oliveira Moraes

Orientador: Prof. Vitor Haertel Ph.D.

Co-orientador: Prof. Robin Thomas Clarke Ph.D.

Porto Alegre, julho de 2005.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
CENTRO ESTADUAL DE PESQUISAS EM SENSORIAMENTO REMOTO E METEREOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM SENSORIAMENTO REMOTO

Dissertação de Mestrado

EXTRAÇÃO DE FEIÇÕES EM DADOS IMAGEM COM ALTA DIMENSÃO POR OTIMIZAÇÃO DA DISTÂNCIA DE BHATTACHARYYA EM UM CLASSIFICADOR DE DECISÃO EM ÁRVORE

por

Denis Altieri de Oliveira Moraes
Bacharel em Estatística (2001 - UFRGS)

Sob orientação do
Prof. Vitor Haertel Ph.D.

Dissertação submetida ao Programa de Pós-Graduação em Sensoriamento Remoto do Centro Estadual de Pesquisas em Sensoriamento Remoto e Metereologia – UFRGS, como requisito parcial para a obtenção do grau de

Mestre em Sensoriamento Remoto

Linha de Pesquisa: Reconhecimento de Padrões
Orientador: Prof. Vitor Haertel Ph.D.
Co-orientador: Prof. Robin Thomas Clarke Ph.D
Banca Examinadora:
Prof. Dr. Dante Barone
Prof. Dr. João Comba
Prof[a]. Dr[a]. Patrícia Ziegelmann

Dissertação apresentada e aprovada em 13 de julho de 2005.

Agradecimentos

À Sabedoria, "que não é dependente dos fenômenos externos ou internos, que não pode ser destruída, que não é impermanente, que nada contém de errado, que é mais profunda que tudo, e que a tudo permeia".

Aos meus pais, Delci Moraes e Eneli Moraes, que cultivaram esse trabalho desde antes de haver iniciado, com inigualável amor e dedicação.

Aos meus companheiros, colegas e professores. Especialmente ao Prof. Vitor Haertel, o qual me orientou como exemplo de competência e dedicação ao trabalho e à ciência.

À minha namorada, Cláudia Cavalcante, pela beleza, doçura e incontáveis lições de vida.

Àquela que caminha livremente no espaço básico dos fenômenos, Chagdud Khadro. Possa ela ter uma longa vida e possam seus benefícios ser ilimitados a todos os seres.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

EXTRAÇÃO DE FEIÇÕES EM DADOS IMAGEM COM ALTA DIMENSÃO POR OTIMIZAÇÃO DA DISTÂNCIA DE BHATTACHARYYA EM UM CLASSIFICADOR DE DECISÃO EM ÁRVORE *

Denis Altieri de Oliveira Moraes

Orientador: Prof. Vitor Haertel Ph.D.

Neste trabalho é investigada uma abordagem para extração de feições baseada na otimização da distância de Bhattacharyya em um classificador hierárquico de estrutura binária. O objetivo é mitigar os efeitos do fenômeno de Hughes na classificação de dados imagem hiper-espectrais. A utilização de um classificador em múltiplo-estágio, analisando um sub-conjunto de classes em cada etapa ao invés do conjunto total, permite modos mais eficientes para extrair as feições mais adequadas em cada etapa do procedimento de classificação. Em uma abordagem de árvore binária, somente duas classes são consideradas em cada etapa, permitindo a implementação da distância de Bhattacharyya como um critério para extração de feições em cada nó da árvore. Experimentos foram realizados utilizando dados imagem do sensor AVIRIS. A performance da metodologia proposta é comparada com métodos tradicionais para extração e seleção de feições.

* Dissertação de Mestrado em Sensoriamento Remoto, Centro Estadual de Pesquisas em Sensoriamento Remoto e Metereologia, Curso de Pós-Graduação em Sensoriamento Remoto da Universidade Federal do Rio Grande do Sul. Porto Alegre/RS, (98 p.). Julho de 2005.

FEATURE EXTRACTION IN HIGH DIMENSIONAL IMAGE DATA BY OPTIMIZING THE BHATTACHARYYA DISTANCE IN A TREE STRUCTURED CLASSIFIER*

Denis Altieri de Oliveira Moraes

Advisor: Prof. Vitor Haertel Ph.D.

In this work we investigate an approach to feature extraction based on the optimization of the Bhattacharyya distance in a tree structured classifier. The aim is to mitigate the effects of the Hughes phenomenon in hyperspectral image data classification. The use of a multi-stage classifier, analyzing one sub-set of classes at each stage rather than the full set at once, allows for a more efficient way to extract the most adequate features at each step of the classification procedure. In a binary tree approach, only two classes are considered at each node, allowing the implementation of the Bhattacharyya distance as a criterion for feature extraction at each tree node. Experiments are performed using AVIRIS image data. The performance of the proposed methodology is compared against more traditional methods for feature selection and extraction.

* Master of Sciences Dissertation in Remote Sensing, Center for Remote Sensing and Meteorology, Graduate Program in Remote Sensing at Federal University of Rio Grande do Sul. Porto Alegre/RS, (98 p.). July 2005.

1 INTRODUÇÃO

Uma grande área de interesse em sensoriamento remoto é a classificação de padrões contidos em imagens digitais. Sejam localizados em plataformas orbitais ou aéreas, atualmente existem vários sensores coletando informações sobre a Terra. Com o propósito de investigar toda a informação disponibilizada por tais sensores, diversas técnicas de classificação de padrões têm sido utilizadas, tais como o uso de classificadores supervisionados, não-supervisionados, em estágio único e múltiplo-estágio. Classificadores em múltiplo-estágio também são conhecidos como classificadores hierárquicos, ou classificadores de decisão em árvore.

Nas duas últimas décadas, com o advento dos sensores hiper-espectrais - sensores com centenas de bandas - surgiu também a necessidade de elaborar métodos eficientes para diminuir a dimensão dos dados sem uma perda significativa de informação. Um exemplo desses sensores é o AVIRIS, com 224 bandas espectrais, o qual utilizaremos na seção dos experimentos.

Para tais sensores, a redução de dimensões é necessária e tem três propósitos básicos: 1) minimizar o efeito do fenômeno de Hughes, 2) aumentar o grau de acurácia da classificação e, 3) otimizar o tempo de processamento e as limitações computacionais decorrentes do grande volume de dados.

O uso desses sensores é muito útil quando as classes em estudo possuem características espectrais muito similares entre si. A classificação de dados em agricultura, como a discriminação da mesma espécie de milho tratada com três tipos diferentes de manejo do solo, é um exemplo dessas aplicações. Nesses casos, diferenças muito sutis entre classes, que dificilmente poderiam ser verificados por sensores multi-espectrais - como exemplo citamos o sensor Landsat TM que possui sete bandas de larga amplitude e descontínuas - tornam-se evidentes quando analisadas sob a visão hiper-espectral, a qual cobre praticamente toda a região do espectro visível com centenas de bandas adjacentes, captando muito mais informação do espectro eletromagnético.

Quanto às técnicas de classificação supervisionadas em estágio único, o classificador Bayesiano é ótimo no sentido de minimizar a esperança do risco. O classificador da Máxima Verossimilhança é um caso particular do anterior quando se

assume valores 0 ou 1 para a função custo e probabilidades *a priori* iguais para todas as classes.

O método do classificador em estágio único é largamente utilizado, entretanto ocorrem problemas quanto à estimação dos parâmetros das funções de densidade de probabilidade condicionadas, vetores de média e matrizes de variância e covariância. O problema de estimação de uma grande quantidade de parâmetros deve-se ao fato de que na medida em que a dimensão dos dados aumenta, e mantendo-se constante o número de amostras de treinamento, o que geralmente acontece em situações reais, a estimação dos parâmetros vai se tornando menos estatisticamente significante e conseqüentemente menos acurada. Esse fato se traduz na diminuição no valor da acurácia de classificação final a partir de uma determinada dimensão dos dados. Tal efeito é conhecido como fenômeno de Hughes, autor do artigo que pela primeira vez expôs o problema.

Quanto aos classificadores hierárquicos, uma das técnicas conhecidas desde a década de 1970 é o Classificador de Decisão em Árvore (CDA). Nessa abordagem, o problema global envolvendo todas as classes simultaneamente é particionado em problemas menores – subconjuntos de classes - ao longo dos ramos e níveis da árvore. Além disso, os CDA's também proporcionam selecionar um subconjunto N de feições de um total de L feições ($N < L$) para classificar os subgrupos de classes. Dado que o tamanho amostral é sempre limitado, tais procedimentos minimizam os efeitos do fenômeno de Hughes, pois em cada nó, teremos um número menor de classes a serem comparadas e também um número menor de feições para estimar os parâmetros.

Dessa forma, com o uso do classificador hierárquico, o problema de seleção ou extração de feições resulta mais eficiente adotando-se um subconjunto das feições mais adequadas àquele particular subconjunto de classes. Esta abordagem permite, em princípio, obter probabilidades de erro ainda menores do que no caso do classificador em estágio único de Bayes.

O objetivo desse trabalho é então, investigar métodos para extração de feições através do uso de CDA's, de forma a se obter uma acurácia de classificação superior àquela fornecida pelo classificador em estágio único de Bayes.

Nesta dissertação, inicialmente é apresentada no capítulo 2 a **Revisão Bibliográfica,** localizando o assunto do problema de estimação de parâmetros em dados

hiper-dimensionais no campo de reconhecimento de padrões e o estado da arte das técnicas envolvidas. No capítulo 3, **Metodologia,** será revisada a teoria envolvendo a metodologia de classificadores hierárquicos, especialmente os algoritmos de decisão em árvore binária e os métodos para seleção e extração de feições.

O primeiro método para extração de feições que explicaremos brevemente é a técnica tradicional de Análise de Discriminante Canônica. O segundo método é a otimização da distância de Bhattacharyya para os diversos casos especiais, concluindo com o método proposto nos experimentos, que é a otimização dessa distância para diferenças predominantes entre matrizes de covariâncias.

Assim, será aplicada no capítulo 4, sobre os resultados dos **Experimentos**, a metodologia proposta para extração de feições em conjunto com o CDA. A maior parte dessa dissertação está concentrada no estudo e aplicação desses métodos em dados do sensor AVIRIS, bem como a comparação dos mesmos com a metodologia usual para seleção de feições, aplicada em um classificador de estágio único. No capítulo 5, sobre a **Conclusão**, é apresentado um resumo dos principais resultados. Em particular, verificamos que o uso do CDA em conjunto com a Análise de Discriminante Canônica mostrou-se superior no caso de pequenas amostras devido sua eficiência computacional e grau médio de acurácia. Por outro lado, o CDA em conjunto com a otimização da distância de Bhattacharyya otimizada para diferenças predominantes entre matrizes de covariância é conveniente no caso de tamanhos de amostras moderados a grandes.

Desse modo, o capítulo 2 iniciará abordando alguns dos fatores que explicam a importância da busca pela melhor otimização dos métodos em reconhecimento de padrões para extração de feições no caso de dados hiper-dimensionais, conforme veremos a seguir.

2 REVISÃO BIBLIOGRÁFICA

Devido à grande diversidade de sensores com capacidades e aplicações específicas, vários métodos em reconhecimento de padrões vêm sendo desenvolvidos com vistas a aprimorar a classificação dos dados obtidos em sensoriamento remoto. No caso de dados em dimensão muito alta (da ordem de 100 a 300 variáveis) como aqueles obtidos por sensores hiper-espectrais, surgem novos desafios a serem transpostos a fim de tornar possível explorar todo o potencial da informação obtida. Como veremos, tais desafios não podem ser superados utilizando procedimentos tradicionais como aqueles desenvolvidos originalmente para dados fornecidos por sensores multi-espectrais com cerca de uma dezena de bandas ou menos. Ao analisar dados em alta dimensão além de lidar com uma tarefa computacionalmente maior, é conveniente observar se o significativo incremento no número de variáveis, não implica no surgimento de problemas adicionais não enfrentados anteriormente.

Embora a complexidade decorrente do aumento no valor da dimensão dos dados tenha sido explorada ao longo das últimas três décadas, seu impacto varia de um campo de aplicação para outro. Por exemplo, em processos de otimização combinatória é sabido que o esforço computacional cresce exponencialmente na classificação de dados em alta dimensão. Em Estatística, tal complexidade se manifesta quando é necessário estimar parâmetros ou funções densidade de probabilidade de variáveis aleatórias, pois dados em alta dimensão apresentam características que necessitam ser endereçadas adequadamente. Nesse capítulo são revistas as principais características dos espaços em dimensão alta (hiper-espaços) e suas implicações no processo de classificação de dados.

2.1 As propriedades do hiper-espaço

Inicialmente, é necessário dizer que nossas percepções relativas ao espaço tri-dimensional, freqüentemente não podem ser extrapoladas para espaços em dimensões mais elevadas. A distribuição dos dados nas funções densidade de probabilidade normais ou uniformes, por exemplo, apresentam características que se contrapõe às percepções mais intuitivas (Landgrebe, 2003).

Nas seções seguintes serão examinadas algumas características de natureza geométrica e estatística pertinente a dados em hiper-espaços, as quais são relevantes ao processo de classificação empregando dados imagem em altas dimensões.

2.1.1 O volume de um hiper-cubo se concentra nos vértices

Com o aumento das dimensões, foi mostrado por D. W. Scott (1992) que o volume de uma hiper-esfera de raio *r* e dimensão *d* é dado pela equação:

$$V_S(r) = \frac{2r^d}{d}\frac{\pi^{\frac{d}{2}}}{\Gamma(d/2)}$$

E o volume de um hiper-cubo em $[-r,r]^d$ é dado por:

$$V_C(r) = (2r)^d$$

Assim, a fração do volume da hiper-esfera inscrita no hiper-cubo de mesmas dimensões é:

$$f_{d1} = \frac{V_S(r)}{V_C(r)} = \frac{\pi^{\frac{d}{2}}}{d2^{d-1}\Gamma(d/2)}$$

Onde, *d* representa a dimensão do espaço. A Figura 2.1 mostra como a razão f_{d1} decresce enquanto a dimensão aumenta. Note que $\lim_{d\to\infty} f_{d1} = 0$, o que implica que o volume do hiper-cubo está cada vez mais concentrado nos vértices a medida em que *d* aumenta.

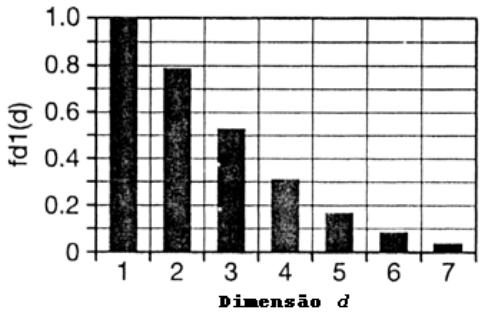


Figura 2.1 Fração do volume de uma hiper-esfera inscrita em um hiper-cubo de dimensão *d* (Landgrebe, 2003)

2.1.2 O volume de uma hiper-esfera se concentra na camada externa

Sendo ε uma fração do raio *r*, Scott (1992) e Wegman (1990) mostram que a fração do volume definido pela esfera de raio *r*-ε, inscrita dentro da esfera de raio *r* é:

$$f_{d2} = \frac{V_d(r) - V(r-\varepsilon)}{V_d(r)} = \frac{r^d - (r-\varepsilon)^d}{r^2} = 1 - \left(1 - \frac{\varepsilon}{r}\right)^2$$

A Figura 2.2 ilustra o caso ε = *r*/5. Nota-se que enquanto a dimensão aumenta, o volume da esfera de dimensão *d* concentra-se na superfície mais externa.

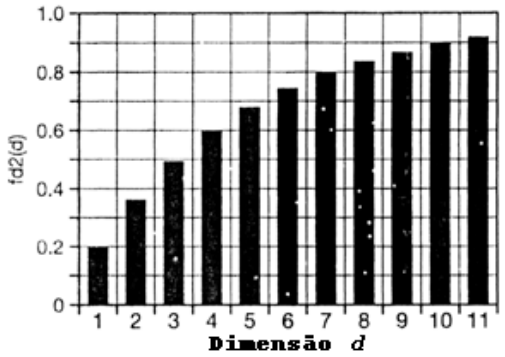


Figura 2.2 Volume contido na camada externa de uma hiper-esfera para um ε = *r*/5 e dimensão *d* (Landgrebe, 2003)

Note que $\lim_{d\to\infty} f_{d2} = 1, \forall \varepsilon > 0$, implicando em que a maior parte do volume da hiper-esfera está concentrado na sua camada externa.

2.1.3 O volume de um hiper-elipsóide se concentra na camada externa

Os resultados anteriores também podem ser generalizados para um hiper-elipsóide. A equação de um hiper-elipsóide em *d* dimensões pode ser escrita como:

$$\frac{X_1^2}{\lambda_1^2} + \frac{X_2^2}{\lambda_2^2} + \ldots + \frac{X_d^2}{\lambda_d^2} = 1$$

E o seu volume é calculado pela expressão (Kendall, 1961):

$$V_e(\lambda_i) = \frac{2\prod_{i=1}^{d} \lambda_i}{d} \frac{\pi^{d/2}}{\Gamma(d/2)}$$

O volume de um hiper-elipsóide definido pela equação:

$$\frac{X_1^2}{(\lambda_1-\delta_1)^2}+\frac{X_2^2}{(\lambda_2-\delta_2)^2}+\ldots+\frac{X_d^2}{(\lambda_d-\delta_d)^2}=1$$

Onde, $0<\delta_i<\lambda_i$ para todo *i* é calculado por:

$$V_e(\lambda_i-\delta_i)=\frac{2\prod_{i=1}^{d}(\lambda_i-\delta_i)}{d}\frac{\pi^{d/2}}{\Gamma(d/2)}$$

Assim, a fração do volume $V_e(\lambda_i-\delta_i)$ inscrito no hiper-elipsóide com $V_e(\lambda_i)$ é:

$$f_{d3}=\frac{\prod_{i=1}^{d}(\lambda_i-\delta_i)}{\prod_{i=1}^{d}\lambda_i}=\prod_{i=1}^{d}\left(1-\frac{\lambda_i}{\delta_i}\right)$$

Seja $\gamma_{\min}=\min\left(\delta_i/\lambda\right)$, então:

$$f_{d3}=\prod_{i=1}^{d}\left(1-\frac{\lambda_i}{\delta_i}\right)\le\prod_{i=1}^{d}(1-\gamma_{\min})=(1-\gamma_{\min})^d$$

Utilizando o fato de que $f_{d3}\ge 0$, conclui-se que $\lim_{d\to\infty} f_{d3}=0$, demonstrando assim que o volume do hiper-elipsóide concentra-se na sua camada mais externa quanto maior o seu número de dimensões.

2.2 Resultados gerais do hiper-espaço

Através do que foi descrito, pelo menos duas importantes conseqüências relativas ao emprego de dados em alta dimensão no contexto desta dissertação são descritas a seguir.

2.2.1 Um hiper-espaço está praticamente vazio

A primeira conseqüência diz respeito ao fato de que um hiper-espaço é praticamente vazio. Isso implica em que os dados multivariados em $\Re^d$ estão na verdade

em uma estrutura dimensional menor. Como conseqüência, dados em alta dimensão podem ser projetados num subespaço reduzido sem perda significativa de informação em termos da separação entre as diferentes classes estatísticas. A segunda conseqüência é que os dados normalmente distribuídos possuem a tendência de se concentrar nas caudas da distribuição multivariada. Similarmente, dados uniformemente distribuídos irão igualmente concentrar-se nos extremos, fazendo assim a estimação das distribuições uma tarefa mais difícil. Assim, vizinhanças locais são geralmente vazias, produzindo perda na estimação detalhada da função densidade em espaços hiper-dimensionais.

É conhecido que distribuições normais apresentam forma simétrica de sino e que a maior parte dos dados concentra-se próximo da média. Como pode ser então verdade que dados normais em alta dimensão concentrem-se nas caudas da distribuição, dado que para as funções gaussianas os valores que possuem maior probabilidade são aqueles que estão próximos da média e não das caudas? Conforme Landgrebe (2003), esse paradoxo foi explicado pela primeira vez por Pi-fuei Hsieh, tal como exposto a seguir.

Primeiramente, note pela Figura 2.3 o que acontece com a magnitude da função densidade gaussiana com média zero enquanto a dimensão aumenta. É sabido que enquanto a forma da curva se mantém em forma de sino, sua magnitude se torna cada vez menor com o aumento da dimensão, pois o volume total precisa permanecer igual a 1. Assim, a magnitude decresce exponencialmente enquanto aumenta o raio *r* em relação à média.

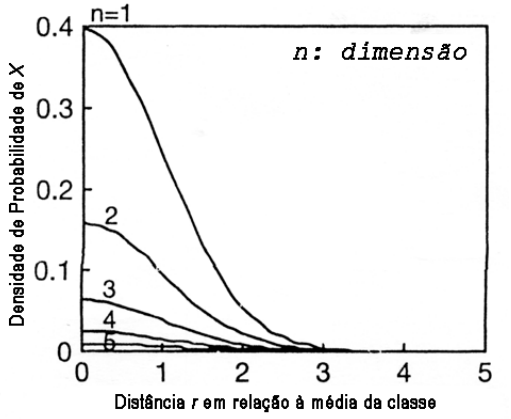


Figura 2.3 Magnitude da densidade gaussiana em função da distância da média para várias dimensões (Landgrebe, 2003)

A seguir, considere como o volume varia com a dimensão do espaço. O volume da hiper-esfera de raio *r* como função da dimensão dos dados é dado por:

$$V_S(S) = \text{volume da hiper-esfera} = \frac{2r^d}{d}\frac{\pi^{d/2}}{\Gamma(d/2)}$$

Assim, o volume em uma camada diferencial como função do raio *r* é:

$$\frac{dV}{dr} = \frac{2\pi^{d/2}}{\Gamma(d/2)} r^{(d-1)}$$

A Figura 2.4 ilustra o volume diferencial na superfície de uma hiper-esfera em função da distância ao seu centro para diferentes valores na dimensão do espaço.

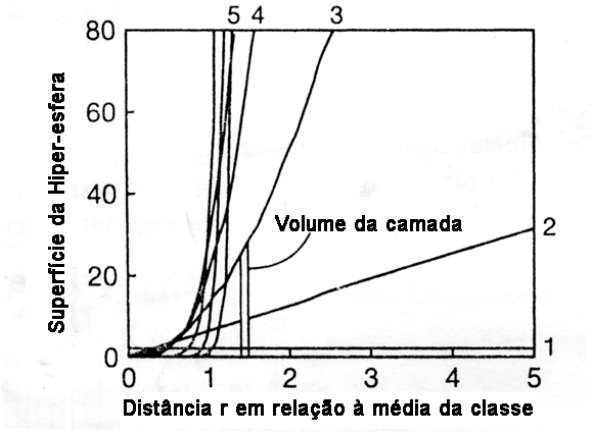


Figura 2.4 Volume da superfície diferencial de uma hiper-esfera em função da distância ao centro e da dimensão *d* (Landgrebe, 2003)

Nota-se que o volume existente em uma camada diferencial de raio *r* aumenta rapidamente, na medida em que a dimensão do espaço aumenta. Então, a massa de probabilidade como uma função do raio *r* pode ser mostrada como:

$$f_r(r) = \frac{r^{d-1}e^{-(r/2)}}{2^{(d/2)-1}\Gamma(d/2)}$$

Essa função está ilustrada na Figura 2.5. Pode ser mostrado que o pico dessa função ocorre em $\sqrt{d-1}$.

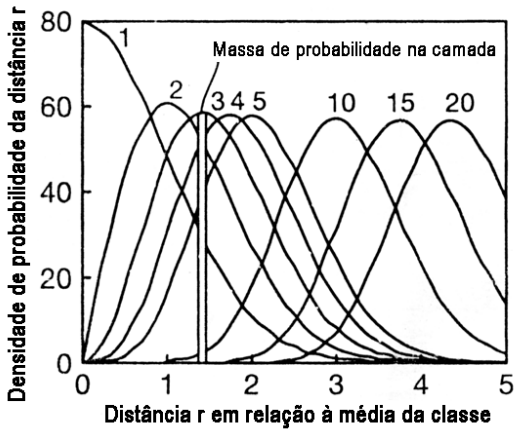


Figura 2.5 Massa de probabilidade em função da distância da média, para densidades gaussianas em várias dimensões (Landgrebe, 2003)

Devido ao volume de uma camada diferencial em uma hiper-esfera aumentar muito mais rapidamente com o raio *r* do que a diminuição da função densidade, o efeito líquido desse aumento apresenta um comportamento como o ilustrado na Figura 2.5. Assim, nota-se que o pico da massa de probabilidade afasta-se da média enquanto a dimensão dos dados aumenta, indicando que a maior parte dos dados concentra-se nas caudas da função densidade, mesmo para dados normalmente distribuídos.

Para elucidar essas características dos dados multivariados, Hsieh gerou amostras com distribuição normal (*N*) e uniforme (*U*) para o experimento descrito a seguir. As variáveis são independentes e identicamente distribuídas, cujos parâmetros são respectivamente *N*(0,1) e *U*(-1,1).

Através desse exemplo, nota-se que a média e o desvio-padrão dessas variáveis aleatórias são funções do número de dimensões. Hsieh (Landgrebe, 2003) calculou a média e o desvio padrão das variáveis aleatórias $r = \sqrt{\sum_{i=1}^{d} x_i^2}$ e $R = \sum_{i=1}^{d} x_i^2$, e as Figuras 2.6 e 2.7 ilustram seus histogramas. Essas variáveis são funções de vetores normais e uniformes para diferentes dimensões.

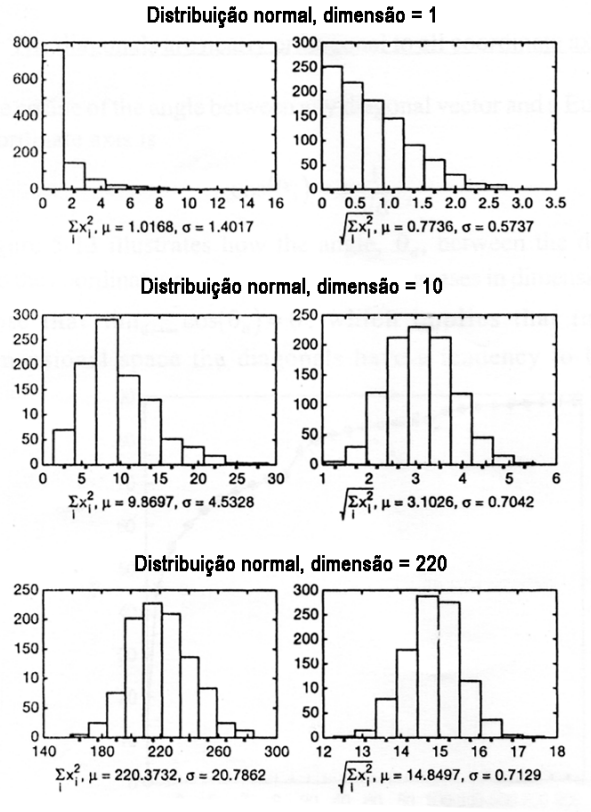


Figura 2.6 Histogramas das funções de variáveis aleatórias normalmente distribuídas para várias dimensões (Landgrebe, 2003)

Nota-se que com o aumento da dimensão, a distância das duas variáveis em relação à coordenada zero aumenta. A variável R tem distribuição χ^2 com d graus de liberdade, onde os x_i's são amostras de uma distribuição $N(0,1)$.

São resultados conhecidos da probabilidade (Sharf, 1991) que a média e variância de R são $E(R) = d$, $Var(R) = 2d$.

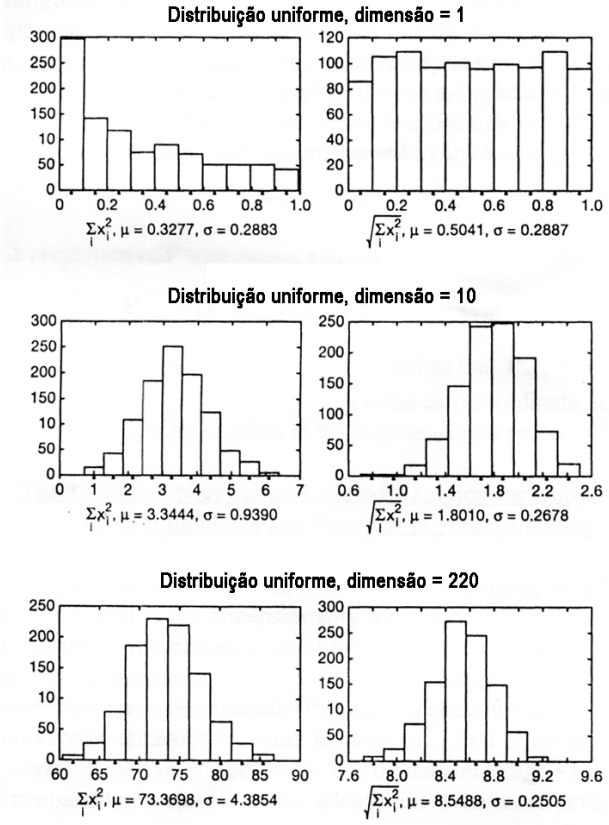


Figura 2.7 Histogramas das funções de variáveis aleatórias uniformemente distribuídas para várias dimensões (Landgrebe, 2003)

Sob essas circunstâncias, seria difícil implementar qualquer procedimento para estimação das funções densidade e obter resultados precisos. Geralmente as abordagens não-paramétricas em alta dimensão enfrentam problemas ainda maiores que as abordagens paramétricas, sendo assim o seu uso nesses casos impraticável.

2.2.2 As diagonais são aproximadamente ortogonais a todos os eixos coordenados

Scott (1992), mostra que o cosseno do ângulo entre qualquer vetor diagonal e um eixo coordenado euclidiano é:

$$\cos(\theta_d) = \pm\frac{1}{\sqrt{d}}$$

A Figura 2.8 ilustra como o ângulo θ_d, entre a diagonal e as coordenadas se aproxima de 90° com o aumento da dimensão. Note que $\lim_{d\to\infty}\cos(\theta_d) = 0$, o qual implica

que, em espaços de alta dimensão, as diagonais possuem tendência a se tornar ortogonais às coordenadas euclidianas.

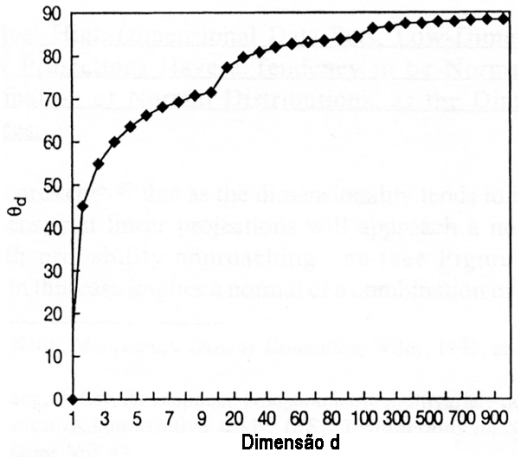


Figura 2.8 Ângulo (em graus) entre uma diagonal e o eixo coordenado euclidiano em função da dimensão d (Landgrebe, 2003)

Essa observação é importante porque a projeção de qualquer agrupamento de feições em qualquer diagonal, por exemplo, fazendo a média das feições, poderia destruir a informação contida nos dados hiper-dimensionais. Para elucidar isso, tome $\mathbf{a}_{\mathbf{diag}}$ como qualquer diagonal em um espaço *d*-dimensional. Tome também $\mathbf{ac}_i$ como a *i*-ésima coordenada desse espaço. Qualquer ponto no espaço pode ser representado pela forma:

$$\mathrm{P} = \sum_{i=1}^{d} \alpha_i \mathrm{ac}_i$$

A projeção de **P** sobre $\mathbf{a}_{\mathbf{diag}}$, $\mathbf{P}_{\mathbf{diag}}$ é:

$$\mathrm{P}_{\mathrm{diag}} = (\mathrm{P}^T \mathrm{a}_{\mathrm{diag}})\mathrm{a}_{\mathrm{diag}} = \sum_{i=1}^{d} \alpha_i (\mathrm{ac}_i^T \mathrm{a}_d)\mathrm{a}_d$$

Mas enquanto *d* aumenta, $\mathbf{ac}_i^{\mathbf{T}}\mathbf{a}_{\mathbf{diag}} \approx 0$, o que implica que $\mathbf{P}_{\mathbf{diag}} \approx 0$. Como uma conseqüência, $\mathbf{P}_{\mathbf{diag}}$ está sendo projetado sob a coordenada zero, perdendo assim informação sobre a sua localização no espaço *d*-dimensional.

2.2.3 O tamanho amostral

Outro ponto importante a ser observado é como o número necessário de amostras para os métodos de classificação supervisionados aumenta como função da dimensão.

Fukunaga (1990) prova que a quantidade de amostras necessárias para treinamento aumenta linearmente em relação à dimensão para um classificador linear e quadraticamente para um classificador quadrático. Esse fato é muito importante, especialmente porque os experimentos demonstram que existem circunstâncias em que as estatísticas de segunda ordem são mais relevantes que as de primeira ordem para discriminar dados em alta dimensão (Choi, 2002). Nesse caso, a utilização de procedimentos não-paramétricos torna o processo ainda mais severo. Foi estimado que, enquanto o número de dimensões aumenta, o tamanho da amostra precisa aumentar exponencialmente para que as funções densidade sejam estimadas com eficiência (Scott, 1992).

É razoável esperar que dados em alta dimensão contenham maior informação, de forma que mais classes possam ser diferenciadas com maior acurácia. Ao mesmo tempo, as características mencionadas acima mostram que as técnicas usuais baseadas em cálculos na dimensão total podem não trazer vantagens, a menos que o número de amostras de treinamento seja substancialmente grande. Isso foi provado por Hughes (1968), o qual demonstrou que a partir de um determinado momento, com um número limitado de amostras de treinamento, há uma penalização na acurácia de classificação enquanto o número de feições aumenta.

2.2.4 Projeções lineares de dados em alta dimensão tendem a se distribuir normalmente em baixa dimensão

Pode-se mostrar (Scott, 1992) que para a maioria dos dados em alta dimensão, suas projeções em baixas dimensões possuem tendência a ser normais, ou uma combinação de distribuições normais, enquanto a dimensão aumenta com probabilidade aproximadamente igual a 1. A normalidade nesse caso implica em uma distribuição normal ou uma combinação de distribuições normais.

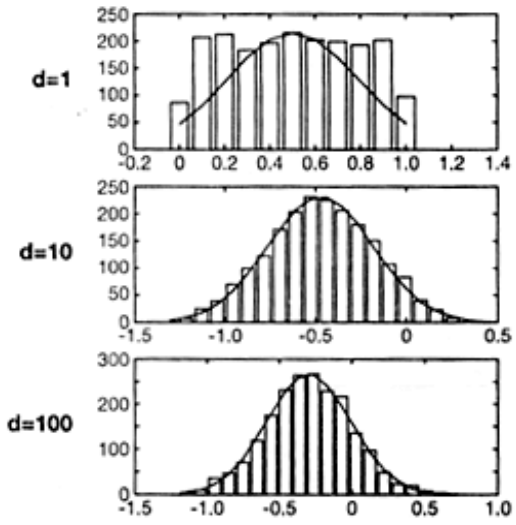


Figura 2.9 Dados simulados de uma classe com distribuição uniforme projetada de um espaço *d*-dimensional para um espaço de dimensão unitária (Landgrebe, 2003)

A literatura relata experimentos diversos neste tópico, utilizando dados reais e simulados. Nesse experimento, os dados foram projetados desde um espaço de alta dimensão para um sub-espaço de dimensão igual à unidade. O comportamento dos dados projetados foi examinado usando dimensões originais iguais a 1, 10 e 100. O método utilizado para as projeções foi o de multiplicar por um vetor normal com ângulos aleatórios às coordenadas. Através de um histograma pode-se ver o comportamento da distribuição dos dados. Uma função com densidade normal foi projetada sobre os histogramas para compará-los (Landgrebe, 2003).

Pode-se observar através da Figura 2.9 que dados que no espaço original apresentam uma distribuição não-normal, no sub-espaço projetado tendem a apresentar uma distribuição normal. A Figura 2.10 ilustra dados hiper-espectrais obtidos pelo sistema AVIRIS, apresentando uma distribuição normal. Na medida em que o número de dimensões no espaço original aumenta, o histograma dos dados projetados tende a se aproximar cada vez mais da distribuição normal. Note que os dados de ambas distribuições (Figura 2.9 e 2.10) têm comportamento similar.

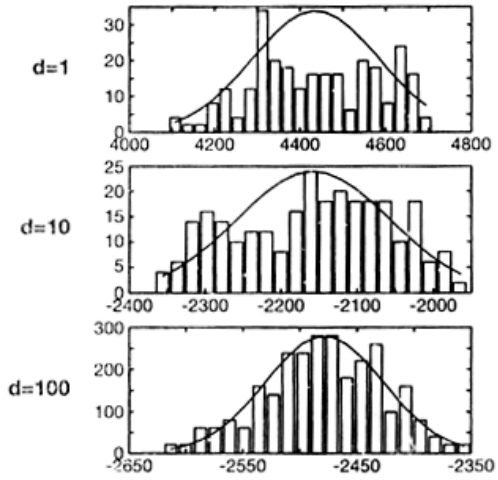


Figura 2.10 Classe espectral com distribuição normal projetada de um espaço *d*-dimensional para um espaço de dimensão unitária (Landgrebe, 2003)

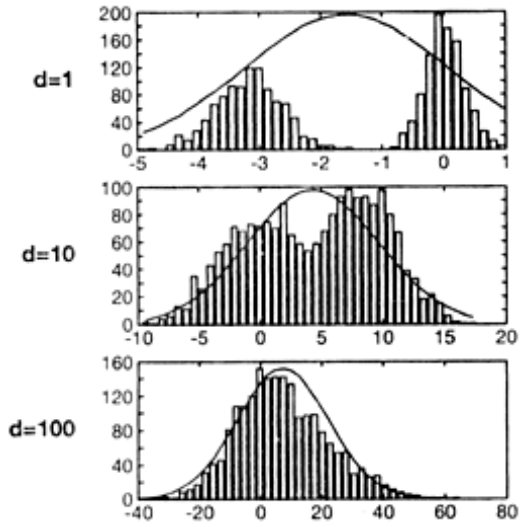


Figura 2.11 Duas classes espectrais simuladas com distribuições normais projetadas de um espaço *d*-dimensional para um espaço de dimensão unitária (Landgrebe, 2003)

Esse resultado sugere que os dados podem ser assumidos como normais ou uma combinação de distribuições normais no espaço projetado sem maiores problemas. Outros experimentos mostram que uma mistura de distribuições normais, onde cada uma representa uma classe estatística diferente, poderia colapsar em uma distribuição normal, o que poderia implicar em perda de informação.

As Figuras 2.11 e 2.12 mostram a repetição do experimento para o problema de duas classes. Ambas ilustram o risco de degradar a informação, tal como a separação entre classes, quando projetados em um sub-espaço. No caso da Figura 2.12 os dados são do sensor AVIRIS, sendo uma das classes, milho, e a outra soja.

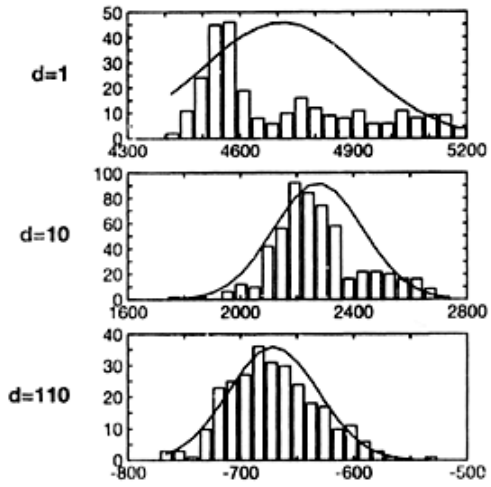


Figura 2.12 Duas classes espectrais simuladas com distribuições normais projetadas de um espaço *d*-dimensional para um espaço de dimensão unitária (Landgrebe, 2003)

Em todos esses casos se podem perceber as vantagens em desenvolver algoritmos capazes de estimar as projeções que melhor separam classes distintamente definidas, realizando os processos de classificação em um espaço de dimensão menor. Os vetores calculados irão separar as classes e, ao mesmo tempo, as classes definidas irão apresentar comportamentos convergindo para distribuições normais. As suposições de normalidade também serão mais consistentes em um sub-espaço do que na dimensão total.

Estatisticamente, as feições originais podem ser resumidas em um número mais reduzido por meio de transformações, lineares ou não, que transportam o conjunto de dados originais para outro sistema de coordenadas espaciais. Tal fato é possível porque as bandas dos sensores hiper-espectrais apresentam alta correlação entre si, sendo assim vantajoso trabalhar com um número menor de feições. Além disso, outra vantagem da redução de dimensões é que geralmente os processos de classificação são muito dispendiosos em termos de tempo de processamento, especialmente os métodos

de classificação hierárquicos. Desse modo, a redução de dimensões também acelera o tempo de processamento final.

2.3 Propriedades assintóticas das estatísticas de primeira e segunda ordem

Nessa seção, é revisto o uso das estatísticas de primeira e segunda ordem no processo de classificação. Os resultados estão baseados no comportamento assintótico dos dados em alta dimensão.

É razoável assumir que enquanto o número de feições cresce, a informação potencial contida nos dados hiper-espectrais também aumenta. Em processos de classificação supervisionada, esse aumento de informação permite a separação de classes que apresentam características muito similares. Nos exemplos mencionados na literatura e aqui reportados, a distância de Bhattacharyya é usada como uma medida de separação entre classes. Essa medida provê um limite de erro para a acurácia de classificação tendo em conta as estatísticas de primeira e segunda ordem. Pode-se interpretar a distância de Bhattacharyya como a soma de duas componentes, uma baseada principalmente na diferença entre médias e outra baseada na diferença entre covariâncias.

A distância de Bhattacharyya, sob a suposição de normalidade é calculada através da expressão:

$$B = \frac{1}{8}(M_2 - M_1)^T \left[\frac{\Sigma_1 + \Sigma_2}{2}\right]^{-1} (M_2 - M_1) + \frac{1}{2}\ln \frac{\left|\frac{\Sigma_1 + \Sigma_2}{2}\right|}{\sqrt{|\Sigma_1||\Sigma_2|}}$$

Onde M_1 e M_2 são os vetores média e Σ_1 e Σ_2 são as matrizes de covariância das classes.

Neste estudo é utilizada a notação *B = Bhatt Mean + Bhatt Cov*, onde *Bhatt Mean*, estima a separação entre duas classes devido à diferenças entre os vetores média enquanto *Bhatt Cov* estima a separação devido as matrizes de covariância.

Dois experimentos ilustrando essas características da distância de Bhattacharyya são relatados em Landgrebe (2003). No primeiro caso, a separação entre as classes é devida principalmente às diferenças entre as estatísticas de segunda ordem (matrizes de covariância), enquanto no segundo experimento a separação é devida

principalmente as estatísticas de primeira ordem (vetores de média). O primeiro, em condições onde as estatísticas de segunda ordem são mais relevantes na discriminação entre as classes e o segundo experimento, apresentando condições favoráveis à predominância das estatísticas de primeira ordem.

Experimento 1

Nesse experimento, os dados foram simulados para duas classes. Ambas possuem distribuições normais com médias e covariâncias diferentes. Cada classe dispõe de 500 amostras de treinamento. Seus parâmetros (vetor média e matriz de covariância) respectivos são:

$$\mu_1 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$\mu_2 = [1.5 \quad 1.5 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$\Sigma_1 = \begin{bmatrix} 1 & & & & & & & & & 0 \\ & 1 & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 1 & & & & & & \\ & & & & 1 & & & & & \\ & & & & & 1 & & & & \\ & & & & & & 1 & & & \\ & & & & & & & 1 & & \\ & & & & & & & & 1 & \\ 0 & & & & & & & & & 1 \end{bmatrix}^T, \quad \Sigma_2 = \begin{bmatrix} 1.5 & & & & & & & & & 0 \\ & 1.9 & & & & & & & & \\ & & 3 & & & & & & & \\ & & & 3 & & & & & & \\ & & & & 3 & & & & & \\ & & & & & 3 & & & & \\ & & & & & & 3 & & & \\ & & & & & & & 3 & & \\ & & & & & & & & 3 & \\ 0 & & & & & & & & & 3 \end{bmatrix}^T$$

Os dados foram classificados usando três classificadores diferentes, o classificador da Máxima Verossimilhança (ML), o classificador ML sob a restrição de utilizar somente a diferença entre covariâncias (ML Cov) e o classificador da Mínima Distância Euclidiana (Min Dist). Os resultados são mostrados na Figura 2.13. Observe que o classificador ML Cov torna-se mais acurado que Min Dist quando são utilizadas mais que duas dimensões.

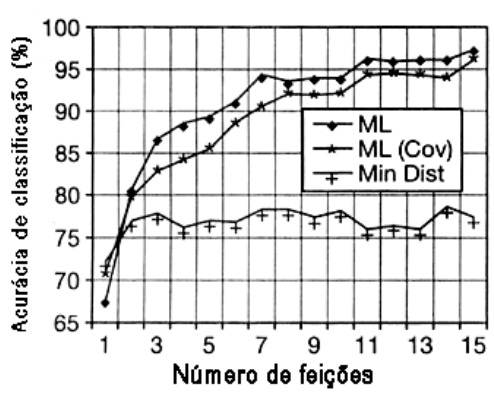


Figura 2.13 Acurácia de classificação para dados simulados - Diferença predominante nas estatísticas de segunda ordem (Landgrebe, 2003)

Para demonstrar as características das estatísticas de primeira e segunda ordem das componentes da distância de Bhattacharyya (*Bhatt Mean, Bhatt Cov* e *Total Dist = B*), as mesmas foram calculadas por Hsieh e ilustradas na Figura 2.14. A razão entre *Bhatt Mean*/*Bhatt Cov* é ilustrada na Figura 2.15.

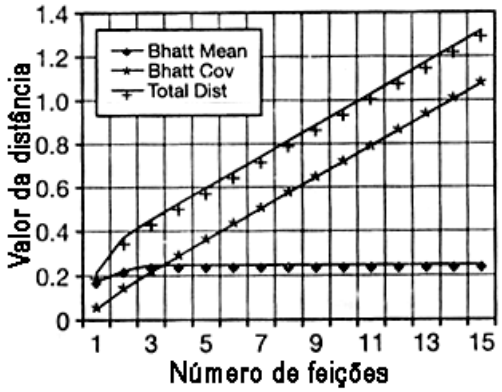


Figura 2.14 Distância de Bhattacharyya, componentes de média, covariâncias e a soma das duas - Diferença predominante nas estatísticas de segunda ordem (Landgrebe, 2003)

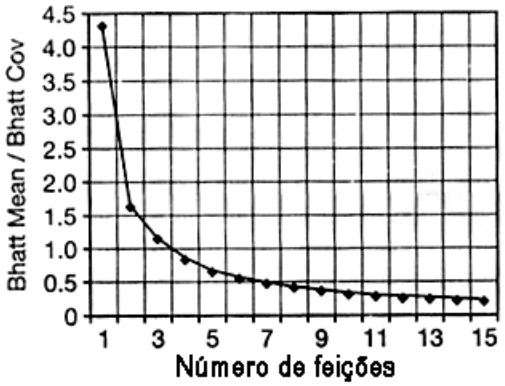


Figura 2.15 Razão entre a componente de média sobre a componente de covariância - Diferença predominante nas estatísticas de segunda ordem (Landgrebe, 2003)

Ambas as figuras exibem o relacionamento entre a predominância das estatísticas de segunda ordem e a relevância do termo *Bhatt Cov*. Enquanto o número de dimensões aumenta, a razão entre o termo *Bhatt Mean/Bhatt Cov* decresce significantemente e o classificador ML Cov torna-se mais efetivo que o Min Dist. Isso mostra que se enquanto a dimensão aumenta, a razão *Bhatt Mean/Bhatt Cov* diminui, as estatísticas de segunda ordem são mais relevantes em dados de hiper-dimensionais mesmo se essa situação não for verdadeira em baixa dimensão.

Experimento 2

Esse experimento é similar ao anterior, com a diferença da predominância das estatísticas de primeira ordem. Os parâmetros (vetor média e matriz de covariância) das duas classes são:

$$\mu_1 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$\mu_2 = [1.5 \quad 1.5 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$\Sigma_1 = \begin{bmatrix} 1 & & & & & & & & 0 \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ 0 & & & & & & & & 1 \end{bmatrix}^T, \quad \Sigma_2 = \begin{bmatrix} 2.5 & & & & & & & & 0 \\ & 2 & & & & & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & & \\ & & & & 1 & & & & \\ & & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & & & & & 1 & \\ 0 & & & & & & & & 1 \end{bmatrix}^T$$

Os resultados de classificação são ilustrados na Figura 2.16. Observe que o classificador Min Dist torna-se mais acurado que ML Cov para dimensões acima de seis. As componentes da distância de Bhattacharyya para a média (*Bhatt Mean*), covariância (*Bhatt Cov*) e suas somas (*Total Dist*) foram calculadas e ilustradas na Figura 2.17.

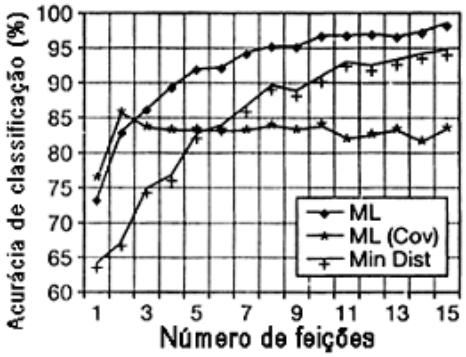


Figura 2.16 Acurácia de classificação de dados simulados - Diferença predominante das estatísticas de primeira ordem (Landgrebe, 2003)

A razão de *Bhatt Cov/Bhatt Mean* está ilustrada na Figura 2.18. Nota-se que enquanto o número de dimensões aumenta, a razão *Bhatt Cov/Bhatt Mean* diminui, mostrando que as estatísticas de primeira ordem são mais relevantes para classificar esse tipo de dados.

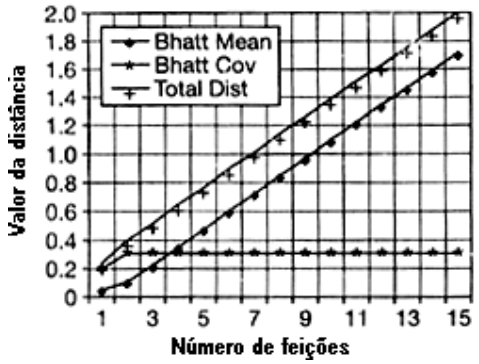


Figura 2.17 Distância de Bhattacharyya, componentes de média, covariâncias e a soma das duas - Diferença predominante das estatísticas de primeira ordem (Landgrebe, 2003)

Esses resultados indicam diretamente que a predominância das componentes de média ou covariância na distância de Bhattacharyya se relaciona com as estatísticas de primeira ou segunda ordem em termos da acurácia de classificação.

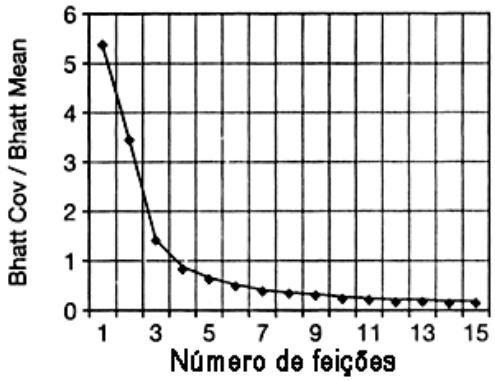


Figura 2.18 Razão entre a componente das médias sobre a componente de covariância - Diferença predominante das estatísticas de primeira ordem (Landgrebe, 2003)

2.4 Implicações da alta dimensão na classificação supervisionada

Baseados nas características observadas sobre os dados em alta dimensão, tais como o volume de hiper-cubos concentrando-se nos vértices, e o volume de hiper-elipsóides nas camadas externas, torna-se evidente que os espaços em alta dimensão são em grande parte vazios, estando assim os dados hiper-dimensionais geralmente em

estruturas de dimensões menores. Como conseqüência, teoricamente é possível reduzir a dimensão dos dados sem perdas significativas de informação e separação.

Devido às dificuldades para estimar funções densidade com abordagens não-paramétricas, é esperado que uma versão paramétrica dos algoritmos produza melhores resultados em casos onde somente um número limitado de amostras conhecidas está disponível para suprir as informações necessárias à priori.

2.4.1 O fenômeno de Hughes

Como já mencionado anteriormente, uma das principais conseqüências do problema de estimação de parâmetros, frente ao número limitado de amostras, é a diminuição no valor da acurácia da classificação a partir de uma determinada dimensão dos dados (Richards, 1998). Tal efeito é conhecido como fenômeno de Hughes (1968), o qual é ilustrado na Figura 2.19.

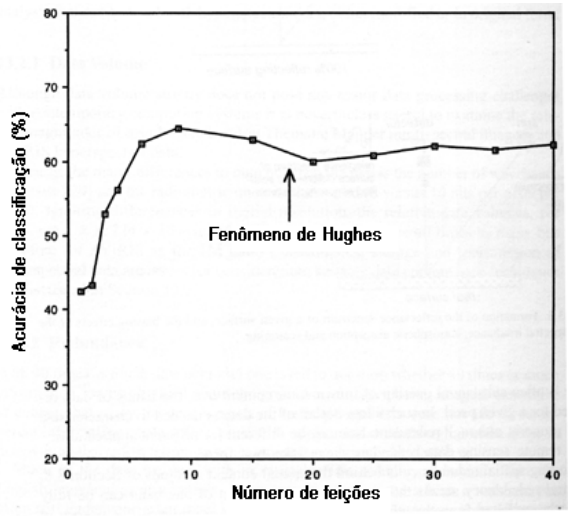


Figura 2.19 Fenômeno de Hughes: decréscimo na acurácia de classificação com o aumento da dimensão (Richards, 1998)

2.5 Principais abordagens de pesquisa para classificação de dados hiper-espectrais

Dentre os atuais processos para mitigar as conseqüências na classificação dos dados hiper-espectrais, destacam-se três vias principais: a análise de Discriminante Regularizada (Aeberhard, 1994), a técnica de amostras semi-rotuladas (Shahshahani, 1994) e os processos para redução de dimensões através de seleção de feições ou extração de feições (Tadjudin, 1998). Nesse trabalho, será tratada somente da última linha de pesquisa: uma metodologia para extração de feições com o intuito de encontrar o melhor subconjunto de combinações lineares das feições originais, conforme um determinado critério de optimalidade.

Outra forma de complementar com uma contribuição significativa ao aumento da acurácia de classificação é a metodologia dos classificadores hierárquicos - ou classificadores de decisão em árvore (CDA) - ao invés dos classificadores de estágio único. Esta abordagem, em conjunto com os métodos para extração de feições permite, em princípio, obter probabilidades de erro ainda menores do que no caso do classificador em estágio único de Bayes.

Dentre os vários métodos que têm sido propostos para o delineamento da estrutura do CDA, destacam-se principalmente três formas: *top-down* (topo para baixo), *bottom-up* (base para cima) e a união destes dois, o método híbrido.

O método *bottom-up* inicia com todas as classes reunidas no mesmo nó raiz. Então, tomando-se alguma medida de dissimilaridade, as classes mais distantes entre si são separadas em nós descendentes distintos ao longo da árvore. As demais classes são então classificadas em algum dos nós descendentes até que sejam alcançados os terminais da árvore, onde o processo termina. O método *top-down* inicia nos terminais da árvore, com todas as classes separadas entre si. Então, a partir de uma medida de similaridade, as classes mais semelhantes são agrupadas ascendentemente até que todas pertençam a um único nó terminal no topo da árvore.

O método *bottom-up* foi o primeiro proposto por Breiman *et al*. (1984) incorporado em um algoritmo conhecido como *Classification and Regression Trees* (CART). O procedimento CART se resume em testar uma combinação linear de feições de forma que essas feições sejam oblíquas no espaço. A idéia fundamental em particionar as classes de um nó, é a de que os dados de cada um dos nós descendentes sejam mais "puros" que os dados do grupo original. Uma maneira de realizar essa tarefa

é definir uma função de impureza $i(t)$. Supondo-se que para o nó t exista uma possível divisão S que divide este nó em dois nós descendentes, esquerdo (L) e direito (R) tal que uma fração p_L dos casos vai para t_L e uma fração p_R para t_R. Pode-se então definir a qualidade dessa divisão pelo decréscimo na função de impureza do nó original:

$$\Delta i(S,t) = i(t) - i(t_L)\,p_L - i(t_R)\,p_R$$

O critério para divisão consiste em selecionar uma divisão S que maximize $\Delta i(S,t)$. Uma das formas conhecidas para o índice de impureza é o índice de Gini (Breiman *et al*., 1984), definido como:

$$i(t) = \sum_{i \neq j} p(i/t)\,p(j/t)$$

Onde $p(i/t)$ é a probabilidade de uma amostra aleatória $\boldsymbol{X}$ pertencer à classe i, dado que está no nó t, enquanto $p(j/t)$ é a probabilidade dessa amostra pertencer à classe j.

A segunda etapa consiste em definir um critério de parada no processo de divisão dos nós, isto é, um critério de término no processo. Um possível critério consiste em definir um limiar $\beta > 0$ e declarar o nó como terminal se:

$$\max_{s \in S} \Delta i(S(t),t) < \beta$$

A terceira etapa consiste em estimar a acurácia a ser obtida no processo de classificação. Essa estimativa é normalmente obtida por meio de amostras disponíveis, empregando procedimentos como *holdout* e *leave-one-out* (Fukunaga, 1990), ou ainda por resubstituição da taxa de erro de classificação de uma árvore T, definida por Breiman *et al*. (1984) como:

$$R(T) = \sum_{t \in \tilde{T}} r(t)\,p(t) = \sum_{t \in \tilde{T}} R(t)$$

Tal que,

$R(T)$: taxa de erro de classificação de uma árvore T;

$r(t)$: taxa de erro de classificação em um nó t pertencente a T;

$p(t)$: probabilidade de uma amostra pertencer ao nó t.

Algumas desvantagens do CART são, primeiramente, que o procedimento permite o uso de uma única feição ou uma combinação linear de feições a cada nó.

Segundo, o CART é computacionalmente pesado, pois requer a criação de múltiplas árvores auxiliares. Finalmente, e talvez o mais importante, embora seja selecionada uma árvore final da família paramétrica de sub-árvores, a árvore ótima no sentido de melhor acurácia final de classificação, pode não estar contida nessa família.

Na literatura são citados ainda outros métodos para delineamento de CDA's.

You e Fu (1976) sugerem uma árvore linear binária, a qual combina classes em dois subgrupos disjuntos a cada nó usando estatísticas das classes.

Gelfand *et al*. (1991) propõem um algoritmo interativo de ramo e poda dividindo os dados em dois grupos aproximadamente iguais e interativamente gerando um ramo da árvore com um dos subgrupos e podando com o outro subgrupo, após isso sucessivamente inverter os papéis. Esse algoritmo de poda é uma simples e intuitiva abordagem *top-down*. Os autores provam a convergência do seu algoritmo e os resultados experimentais em reconhecimento de ondas, com suporte teórico, sugerem a superioridade desse método sobre o método proposto por Breiman *et al*. (1984).

Outro método paramétrico é proposto por Kim e Landgrebe (1990) usando uma abordagem mista *bottom-up* e *top-down* (abordagem híbrida) seqüencialmente. O método *bottom-up* calcula a distância de Bhattacharyya entre cada par de classes e as duas classes com a menor distância são agrupadas, formando um novo grupo, até que restem apenas dois grupos no topo da árvore. Então, calcula-se o vetor média e a matriz de covariância para cada grupo e essa informação é utilizada em um algoritmo *top-down* para gerar dois novos grupos. Cada grupo é verificado para ver se ele contém apenas uma classe. Se sim, esse nó é rotulado como um terminal, se não, o procedimento anterior é repetido.

Destaca-se, finalmente, o método *branch-and-prunning* (ramo e poda), o qual foi desenvolvido para obter soluções ótimas para problemas combinatórios sem que seja necessário recorrer a buscas exaustivas, garantindo que o subconjunto de feições selecionadas seja o melhor dentre todas as combinações possíveis. Nesse método, dado que t é um nó em uma árvore T, define-se como *ramo* da arvore T em t, ao segmento de T_t formado pelo nó t, e todos os seus descendentes; Podar um ramo T_t consiste em remover da árvore T o nó t e todos os seus descendentes, de forma que a árvore final seja formada por $T - T_t$. O principal problema desse método é que o número

de árvores a ser testado é geralmente muito grande, sendo que a melhor árvore pode mesmo nem existir.

No capítulo 3 é apresentada a metodologia do CDA utilizado nessa dissertação em conjunto com o método para extração de feições baseado em um caso especial da otimização da distância de Bhattacharyya. A seguir, é dada uma breve explanação sobre o processo de extração de feições através da Análise de Discriminante Canônica, a qual é utilizada em conjunto com o CDA proposto. Também é abordado um processo para seleção de feições em conjunto um classificador de estágio único. Assim, essas duas técnicas são utilizadas nos experimentos finais com o objetivo de comparar a metodologia proposta com os métodos mais convencionais de classificação.

3 METODOLOGIA

3.1 Introdução

Em sensoriamento remoto, imagens obtidas por sensores hiper-espectrais produzem um conjunto de medidas em diferentes regiões do espectro eletromagnético (bandas espectrais) que exprimem, em termos de radiância, a energia eletromagnética refletida e/ou emitida pelos elementos de cobertura do solo na cena imageada. Nos dados imagem, os valores de radiância espectral em cada pixel individual podem ser convenientemente organizados na forma de um vetor de observações com dimensão igual ao número de bandas espectrais disponíveis no sistema. O processo de classificação consiste em rotular cada pixel individual, definido pelo seu vetor de observações, atribuindo-o a uma das classes existentes. Deve-se notar que o referido vetor de observações é uma *variável aleatória*, com uma distribuição descrita por uma *função densidade de probabilidade* multivariada. Os parâmetros nessa função densidade devem ser estimados para cada classe individualmente, e o problema de reconhecimento de padrões torna-se um problema clássico de teste de hipóteses.

Neste capítulo, é inicialmente revisto o problema de decisão envolvendo duas classes, o qual pode ser posteriormente estendido para o problema multi-classe.

3.2 O Classificador de Bayes

Dado um vetor de observações X, a regra de classificação de Bayes fornece um critério de decisão para fins de alocação de um individuo a uma determinada classe. Uma regra de decisão baseada simplesmente nas probabilidades de X pertencer a cada uma das classes ω_i pode ser escrita da seguinte forma (Fukunaga, 1990):

$$X \in \omega_i \quad \text{se} \quad p(\omega_i / X) > p(\omega_j / X) \quad \forall\ i \neq j \tag{1}$$

Onde $p(\omega_i /X)$ é a probabilidade *a posteriori* de ω_i dado X.

A probabilidade *a posteriori* $p(\omega_i/X)$ pode ser obtida de uma forma conveniente, utilizando o teorema de Bayes:

$$p(\omega_i / X) = \frac{p\left(X / \omega_i\right) P(\omega_i)}{p(X)}$$

Onde $p(X)$ é a função densidade ponderada das duas classes, ou *função mistura de densidade*, $P(\omega_1)$ é a probabilidade *a priori* da classe ω_1 e $p(X/\omega_2)$ é a probabilidade condicional de X dado que pertence à classe ω_1. Nesse caso, a regra de decisão (1) pode ser escrita como:

$$X \in \omega_1 \quad \Leftrightarrow \quad \frac{p\left(X / \omega_1\right) P(\omega_1)}{p(X)} \; > \; \frac{p(X / \omega_2)\, P(\omega_2)}{p(X)} \qquad (2)$$

Caso contrario, $X \in \omega_2$.

Como $p(X)$ é positiva e comum a ambos os lados da desigualdade (2), a regra de decisão pode ser simplificada para:

$$X \in \omega_1 \quad \Leftrightarrow \quad p\left(X / \omega_1\right) P(\omega_1) \; > \; p(X / \omega_2)\, P(\omega_2) \qquad (3)$$

Caso contrário, $X \in \omega_2$.

A regra de decisão (3) pode ser expressa de uma outra forma por:

$$X \in \omega_1 \quad \Leftrightarrow \quad l(X) = \frac{p\left(X / \omega_1\right)}{p\left(X / \omega_2\right)} \; > \; \frac{P(\omega_2)}{P(\omega_1)} \qquad (4)$$

Caso contrário, $X \in \omega_2$.

O termo $l(X)$ é denominado de *razão de verossimilhança* e é a quantidade básica no teste de hipóteses. Nesta expressão, a razão entre as duas probabilidades *a priori* faz a função de um limiar (*threshold*) para a razão de verossimilhança no processo de decisão. As regras de decisão (3) e (4) não serão alteradas se ambos os membros forem substituídos por funções monotonicamente crescentes dos mesmos. Para o caso de dados apresentando uma distribuição normal multivariada, torna-se mais prático, especialmente para fins computacionais, empregar a função logaritmo natural (ln) como mostrado abaixo.

$$X \in \omega_1 \quad \Leftrightarrow \quad \ln\left(\, l(X)\,\right) = \ln p(X / \omega_1) - \ln p(X / \omega_2) \; < \; \ln \frac{P(\omega_1)}{P(\omega_2)}$$

Para o caso multi-classe, assumindo-se probabilidades *a priori* iguais, esta regra de classificação pode ser escrita como:

$$X \in \omega_i \quad \Leftrightarrow \quad l_i(X) > l_j(X) \qquad \forall\ j \neq i \tag{5}$$

O classificador em (4) implementa uma abordagem probabilística ao processo de classificação. A probabilidade de erro ε neste caso pode ser estimada por:

$$\varepsilon = P(\omega_1)\int_{L_2} p(X/\omega_1)dX + P(\omega_2)\int_{L_1} p(X/\omega_2)dX = P(\omega_1)\,\varepsilon_1 + P(\omega_2)\,\varepsilon_2$$

Onde L_1 e L_2 são as regiões de domínio das classes ω_1 e ω_2, respectivamente.

O classificador Bayesiano é um classificador *ótimo* no sentido de que minimiza a probabilidade de erro. Esta probabilidade mínima de erro é conhecida como *erro de Bayes*. A Figura 3.1 ilustra o caso unidimensional dessa regra, onde o erro bayesiano dado pelo limiar *t* é o menor possível, conforme a figura, $\varepsilon = \mathrm{A} + \mathrm{B} + \mathrm{C}$. Caso seja tomado qualquer outro limiar, digamos *t'*, a região de erro sob as duas curvas é aumentada pela quantidade D, sendo assim maior que o erro de Bayes.

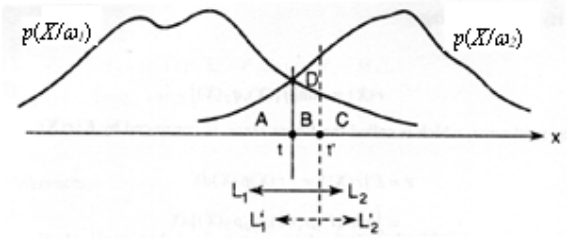


Figura 3.1 A regra de decisão de Bayes (Therrien, 1989)

A abordagem anterior, contudo, é baseada implicitamente no entendimento de que a perda ou penalidade que se incorre por um erro de rotulagem de pixel seja constante e igual para todas as classes. Para desenvolver o método geral, é necessário introduzir o conceito de uma função de penalidade, ou função perda:

$$\lambda(i \mid k) \quad i, k = 1,\ldots M$$

Essa é uma medida de penalidade incorrida quando se rotula erroneamente um pixel como pertencente à classe ω_i quando na realidade ele pertence à classe ω_k.

A função perda total ou risco, estima a penalidade total que se incorre ao rotular um pixel *X* na classe ω_i:

$$L_X(\omega_i) = \sum_{k=1}^{M} \lambda(i \mid k)\, p(\omega_k / X)$$

Sendo *M,* o número total de classes no processo.

A função risco pode ser expressa de uma forma mais adequada fazendo uso do teorema de Bayes:

$$L_X(\omega_i) = \frac{1}{p(X)} \sum_{k=1}^{M} \lambda(i \mid k)\, p(X / \omega_j)\, P(\omega_j) \tag{6}$$

A regra de decisão Bayesiana consiste em rotular *X* de forma a minimizar a perda total (ou risco) $L_X(\omega_i)$. A utilização de (6) requer uma definição com relação às funções perda $\lambda(i|k)$. Uma escolha, freqüentemente sugerida na literatura, consiste em atribuir um mesmo valor para todos os possíveis erros de rotulagem e zero para o acerto, isto é:

$$\lambda(i|k) = 0 \quad \text{se} \quad k = i$$

$$\lambda(i|k) = 1 \quad \text{se} \quad k \neq i$$

Nesta suposição a função perda total (6) fica:

$$L_X(\omega_i) = \frac{1}{p(X)} \sum_{\substack{j=1 \\ j \neq i}}^{M} p(X / \omega_j)\, P(\omega_j)$$

A estratégia bayesiana consiste em minimizar $L_X(\omega_i)$ ou equivalentemente maximizar uma função decisão $G_i(X)$:

$$G_i(X) = - L_X(\omega_i)$$

Isto é, o rótulo para *X* é escolhido de forma a maximizar $G_i(X)$. Sendo que $p(X)$ independe de ω_i, a função decisão pode ser escrita como:

$$G_i(X) = -\sum_{\substack{j=1 \\ j \neq i}}^{M} p(X / \omega_j)\, P(\omega_j) \tag{7}$$

Sendo o conjunto das *M* classes exaustivo, pode-se escrever em geral:

$$p(X) = \sum_{j=1}^{M} p(X / \omega_j)\, P(\omega_j) = p(X / \omega_i)\, P(\omega_i) + \sum_{\substack{j=1 \\ j \neq i}}^{M} p(X / \omega_j)\, P(\omega_j)$$

Então:

$$\sum_{\substack{j=1 \\ j \neq i}}^{M} p(X / \omega_j)\, P(\omega_j) = p(X) - p(X / \omega_i)\, P(\omega_i)$$

Substituindo em (7) e lembrando que $p(X)$ independe de ω_i, pode-se escrever a função decisão como:

$$G_i(X) = p(X / \omega_i)\, P(\omega_i) \tag{8}$$

A função decisão (8) é conhecida como *função discriminante da máxima verossimilhança*.

Assim, a regra da máxima verossimilhança atribui um vetor de observações X a uma classe ω_i, se e somente se, o valor da função discriminante $G_i(X)$ for maior que o valor de $G_j(X)$, isto é:

$$X \in \omega_i \quad \Leftrightarrow \quad G_i(X) > G_j(X) \qquad \forall \qquad j \neq i \tag{9}$$

A expressão acima é usualmente conhecida como a regra incondicional da máxima verossimilhança.

Sabe-se que dados imagem de cenas naturais podem ser adequadamente modelados por distribuições gaussianas multivariadas. Neste caso, a função densidade de probabilidade assume a forma:

$$p(X \mid \omega_i) = \frac{1}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2} (X - M_i)^T \Sigma_i^{-1} (X - M_i) \right]$$

Sendo n a dimensão dos dados (número de bandas espectrais), M_i e Σ_i respectivamente o vetor média e a matriz de covariância associados à classe ω_i.

Assim, substituindo-se na função discriminante $G_i(X)$, o estimador quadrático da Máxima Verossimilhança Gaussiana na forma matricial é dado por:

$$G_i(X) = \frac{p(\omega_i)}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2} (X - M_i)^T \Sigma_i^{-1} (X - M_i) \right] \tag{10}$$

A decisão do classificador (10) não será alterada se for tomada como função decisão uma função monotonicamente crescente de $G_i(X)$:

$$G_i(X) = \ln[P(\omega_i)] - \frac{1}{2} \ln|\Sigma_i| - \frac{1}{2} (X - M_i)^T \Sigma_i^{-1} (X - M_i) \tag{11}$$

A função (11) é a forma mais utilizada da função de decisão da Máxima Verossimilhança Gaussiana. Na prática, com freqüência assumem-se iguais valores para as probabilidades *a priori* $P(\omega_i)$. Nesse caso, a função (11) torna-se:

$$G_i(X) = -\ln|\Sigma_i| - (X - M_i)^T \Sigma_i^{-1} (X - M_i) \quad (12)$$

3.3 Estimação dos parâmetros das funções discriminantes

Os parâmetros em um classificador paramétrico são estimados a partir de um conjunto de amostras disponíveis para cada uma das classes envolvidas, as quais são denominadas de *amostras de treinamento.* No caso do classificador da máxima verossimilhança gaussiana, a estimação da matriz de covariância requer que o número de amostras de treinamento seja no mínimo igual à dimensão dos dados mais um, caso contrário, a matriz de covariância resulta singular. Esse número mínimo de amostras, entretanto, produz estimativas pouco confiáveis para os parâmetros do classificador, em especial para a matriz de covariância. Uma regra simples proposta na literatura estipula que o número de amostras de treinamento para cada uma das classes envolvidas deve ser da ordem de dez vezes maior do que a dimensão dos dados, para que as estimativas amostrais dos parâmetros do classificador sejam aceitáveis. Na prática, entretanto, o analista geralmente não dispõe de um número suficiente de amostras para gerar boas estimativas dos parâmetros em dados imagem hiper-dimensionais, sendo necessário recorrer aos métodos para redução de dimensão dos dados que sejam mais eficazes, como os que serão abordados a seguir.

3.4 Métodos para redução de dimensões

Um problema freqüentemente encontrado no processo de classificação de imagens digitais em sensoriamento remoto consiste no número muitas vezes insuficiente de amostras de treinamento para a estimação dos parâmetros do classificador. A coleta de amostras em campo tem um alto custo financeiro e em algumas vezes inviabiliza a realização do projeto. É necessário assim, encontrar metodologias que produzam melhores resultados de classificação frente à quantidade limitada de amostras e ao número grande de parâmetros a serem estimados.

Existem duas abordagens gerais para fins de redução da dimensão nos dados de imagens. A primeira delas é por seleção de feições e a segunda é por extração de feições. Nesta dissertação é investigada a utilização do critério da otimização da distância de Bhattacharyya (ODB) e de princípios da Análise de Discriminante Canônica (ADC) para fins de extração de feições em um CDA em estrutura binária. A eficiência dessa metodologia para redução da dimensão dos dados é então comparada a um processo clássico de redução de feições via seleção de feições, o *Sequential Forward Selection* (SFS).

3.4.1 Seleção de Feições

As técnicas para seleção de feições geralmente empregam um algoritmo de busca e uma função critério de decisão. O algoritmo de busca gera e compara possíveis soluções, isto é, subgrupos de feições, aplicando uma função critério de decisão como uma medida de adequacidade da solução. Segundo Serpico *et al.* (1995), é dito que "uma busca exaustiva pela solução ótima é proibitiva do ponto de vista computacional, mesmo para um número moderado de feições". Inicialmente são usadas todas as bandas originais e, especificando-se um número desejado de feições menor que a dimensão total, aquelas que não contribuem para a discriminação das classes são removidas. Um critério usualmente empregado é a separação entre as classes. Se a remoção de uma banda, ou de um conjunto de bandas, não diminuir o valor dessa medida de separação substancialmente, então esta banda será redundante.

3.4.1.1 *Sequential Forward Selection*

Dentre os vários métodos para seleção de feições conhecidos, uma metodologia simples e amplamente utilizada é aquela denominada de *Sequential Forward Selection* (SFS). Suponhamos que se queira selecionar N bandas de um total de M disponíveis. Na metodologia SFS, inicialmente é selecionada entre todas as M bandas aquela que melhor discrimina as classes envolvidas, de acordo com alguma medida de separação. Essa primeira banda não será mais descartada e fará conjunto com a próxima banda selecionada dentre as M-1 disponíveis, e assim por diante, até que sejam selecionadas as N bandas desejadas. Nesse trabalho a medida utilizada foi a

otimização da distância de Jeffries-Matusita para o problema multi-classe, a qual é dada por (Serpico *et al*, 1995):

$$J_{Bh} = \sum_{i=1}^{M} \sum_{j>i} \sqrt{P(X/\omega_i)P(X/\omega_j)} J_{ij}^2$$

Deve-se considerar que a seleção ótima de feições envolve uma busca exaustiva envolvendo todas as feições, resultando em um alto custo computacional, principalmente no caso de dados hiper-espectrais.

3.4.2 Extração de Feições

A extração de feições envolve a transformação das bandas originais em um número reduzido de feições, enquanto mantém a separação das classes tanto quanto possível. Essa transformação é geralmente linear e baseada em algum critério de otimização. Na próxima seção é abordada brevemente a metodologia para extração de feições através da Análise de Discriminante Canônica, e a seguir, a otimização da distância de Bhattacharyya, sendo essa última explorada em detalhes para todos os possíveis casos especiais.

3.4.2.1 Análise de Discriminante Canônica

A Análise de Discriminante Canônica utiliza o quociente da matriz de dispersão entre-classes Σ_b sobre a matriz de dispersão intra-classes Σ_w como uma função de critério e calcula um vetor *d* que maximize:

$$\frac{d^T \Sigma_b d}{d^T \Sigma_w d}$$

Onde, Σ_b é a matriz de dispersão entre-classes, definida por:

$$\Sigma_b = \sum_{i=1}^{L} \alpha_i (M_i - M_0)(M_i - M_0)^T$$

Σ_w é a matriz de dispersão intra-classes, tal que:

$$\Sigma_w = \sum_{i=1}^{L} \alpha_i \Sigma_i$$

Σ_i e α_i são a matriz de covariância e a probabilidade *a priori* da classe *i*, respectivamente. *L* é o número total de classes e M_0 é o vetor média, definido por:

$$M_0 = \sum_{i=1}^{L} \alpha_i M_i$$

Embora a análise de discriminante tenha uma performance boa para a maior parte dos casos, o número de feições fornecido por essa abordagem é limitado e igual ao mínimo entre o número de classes menos um e o número de variáveis menos um, isto é:

Número de feições pelo método ADC = min(*L-1*, *dim*(M_i *-1)*)

Assim, para um problema envolvendo *L* classes e *M* feições, com $M > L$, podem ser extraídas somente (*L*-1) feições. Outro aspecto é que se os vetores de médias são muito próximos, ou iguais, a extração de feições não é possível. Por último, para o caso multi-classe, se uma classe tem vetor média muito diferente das outras classes, a matriz de dispersão entre-classes é enviesada para essa classe, resultando em feições ineficientes.

3.4.3 Medidas estatísticas de separação entre classes

Nessa seção, será apresentada uma introdução sobre duas medidas de separação entre classes relacionadas, conhecidas como *Divergência* e *distância de Bhattacharyya*. Para essa última, é apresentada sua forma geral e gaussiana.

3.4.3.1 Divergência

A razão de verossimilhança pode ser considerada como uma estatística, sendo o logaritmo da razão de verossimilhança dado por:

$$\Lambda(X) = \ln \frac{p(X/\omega_1)}{p(X/\omega_2)},$$

Essa medida pode ser descrita com o par de funções densidade de probabilidade ilustrados na Figura 3.2, conforme exposto por Therrien (1989). Em uma regra de decisão empregando o logaritmo da razão de verossimilhança, comparado a um

limiar τ, a probabilidade de erro será menor quando as funções densidade forem bem separadas entre si, e maior caso contrário.

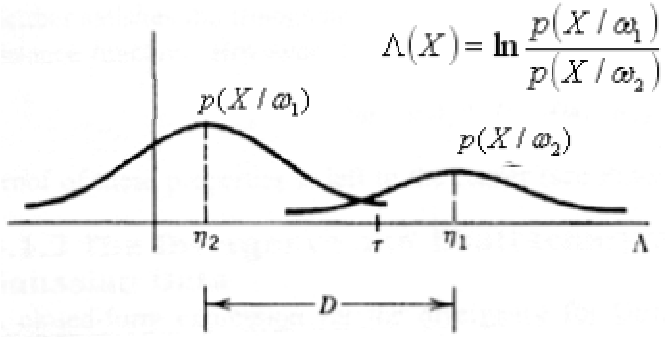


Figura 3.2 Funções densidade típicas para o logaritmo da razão de verossimilhança (Therrien, 1989)

Uma medida de separação entre classes baseada na diferença entre as médias η_1 e η_2, definida por *D,* está ilustrada na Figura 3.2. Essa quantidade é conhecida como *Divergência.* A Divergência é formalmente definida como a esperança da probabilidade condicional de *X*, dada uma classe *i,* de forma que:

$$D = E[\Lambda(X)|\omega_1] - E[\Lambda(X)|\omega_2]$$

Então, substituindo-se $\Lambda(X)$ pela sua definição, a Divergência pode ser re-escrita como:

$$D = \int_{-\infty}^{\infty} \left(\ln \frac{p(X/\omega_1)}{p(X/\omega_2)} \right) p(X/\omega_1) dX - \int_{-\infty}^{\infty} \left(\ln \frac{p(X/\omega_1)}{p(X/\omega_2)} \right) p(X/\omega_2) dX$$

3.4.3.2 Distância de Bhattacharyya: Forma Geral

Outra medida de separação entre classes muito utilizada é a distância de Bhattacharyya, definida como:

$$B = -\ln\left[\int_{-\infty}^{\infty}\sqrt{p(X/\omega_1)\,p(X/\omega_2)}\,\mathrm{d}X\right]$$

A Figura 3.3 mostra um par típico de funções densidade $p(X/\omega_1)$ e $p(X/\omega_2)$. Para interpretar a distância de Bhattacharyya, note que se as funções originais estão bem separadas e a probabilidade de *X* com respeito à classe *i* for alta, a probabilidade de *X* com respeito à classe *j* será muito próxima de zero. Assim, o produto $p(X/\omega_1)\,.\,p(X/\omega_2)$ tenderá a zero e *B* a mais infinito, pois $\lim_{X\to 0}(-\ln(X)) = +\infty$. Por outro lado, se as densidades se sobrepõem, então o produto das duas densidades tenderá a 1 e *B* a zero, pois $\lim_{X\to 1}(-\ln(X)) = 0$.

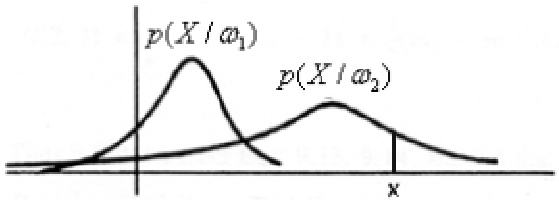


Fig 3.3 Exemplo típico de separação entre duas funções densidade (Therrien, 1989)

A distância de Bhattacharyya é invariante frente a uma transformação linear do vetor *X* e também é aditiva quando os componentes de *X* são independentes, isto é, pode ser expressa como uma soma dos termos similares com cada termo envolvendo somente uma das componentes de *X*. Em adição a isso, se $J_m(\omega_1,\omega_2)$ representa a distância entre as duas classes baseada em uma feição *X* com *m* componentes, então as seguintes propriedades métricas de uma função distância são apropriadas:

$$J_m(\omega_1,\omega_2) > 0; \omega_1 \neq \omega_2$$

$$J_m(\omega_1,\omega_1) = J_m(\omega_2,\omega_2) = 0$$

$$J_m(\omega_1,\omega_2) = J_m(\omega_2,\omega_1)$$

Tais propriedades não satisfazem a desigualdade triangular e, assim não podem ser classificadas como funções verdadeiras de distâncias. Contudo, ambas satisfazem a propriedade adicional:

$$J_m(\omega_1, \omega_2) \leq J_{m+1}(\omega_1, \omega_2)$$

3.4.3.3 Distância de Bhattacharyya: Forma Gaussiana

Embora não seja trivial, a forma da distância de Bhattacharyya, sob a suposição de normalidade dos dados, pode ser obtida através da seqüência de procedimentos a seguir.

Considerando-se a função densidade gaussiana e aplicando-a diretamente na equação da distância de Bhattacharyya, se obtém:

$$B = -\ln \int_{-\infty}^{\infty} p(X/\omega_1)^{1/2}\, p(X/\omega_2)^{1/2}\, dX$$

$$B = -\ln \int_{-\infty}^{\infty} \frac{1}{(2\pi)^{m/2} |\Sigma_1|^{1/4} |\Sigma_2|^{1/4}} \exp\left\{ -\frac{1}{4}\left[(X-M_1)^T \Sigma_1^{-1} (X-M_1) + (X-M_2)^T \Sigma_2^{-1} (X-M_2) \right] \right\} dX \quad (13)$$

O termo exponencial pode ser expandido e reescrito como:

$$-\frac{1}{4}\left[(X-M_1)^T \Sigma_1^{-1} (X-M_1) + (X-M_2)^T \Sigma_2^{-1} (X-M_2) \right]$$

$$= \frac{1}{4}\left[X^T \Sigma_1^{-1} X - 2M_1^T \Sigma_1^{-1} X + M_1^T \Sigma_1^{-1} M_1 + X^T \Sigma_2^{-1} X - 2M_2^T \Sigma_2^{-1} X + M_2^T \Sigma_2^{-1} M_2 \right]$$

$$= \frac{1}{2}\left[X^T \Sigma_p^{-1} X - 2M_p^T \Sigma_p^{-1} X + M_p^T \Sigma_p^{-1} M_p \right] + C \quad (14)$$

Onde,

$$\Sigma_p^{-1} = \frac{1}{2}\left(\Sigma_1^{-1} + \Sigma_2^{-1} \right)$$

$$M_p = \frac{1}{2}\Sigma_p \left(\Sigma_1^{-1} M_1 + \Sigma_2^{-1} M_2 \right)$$

$$C = \frac{1}{4}\left(M_1^T \Sigma_1^{-1} M_1 + M_2^T \Sigma_2^{-1} M_2 - 2M_p^T \Sigma_p^{-1} M_p \right) \quad (15)$$

Então, de acordo com (14), a equação (13) pode ser escrita como:

$$B = -\ln\left\{ \frac{\left|\Sigma_p\right|^{1/2}}{\left|\Sigma_1\right|^{1/4}\left|\Sigma_2\right|^{1/4}} e^{-C} \int_{-\infty}^{\infty} \frac{1}{(2\pi)^{m/2}\left|\Sigma_p\right|^{1/2}} \exp\left[-\frac{1}{2}\left(X - M_p\right)^T \Sigma_p^{-1}\left(X - M_p\right)\right] dX \right\}$$

$$= C + \frac{1}{2}\ln\frac{\left|\Sigma_1\right|^{1/2}\left|\Sigma_2\right|^{1/2}}{\left|\Sigma_p\right|}$$

Desde que sua integral seja igual a 1. Pode ser mostrado através de manipulações algébricas que *C* tal qual definido pela equação (15) se reduz a:

$$C = \frac{1}{8}\left(M_1 - M_2\right)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} \left(M_1 - M_2\right)$$

E que:

$$\frac{\left|\Sigma_1\right|^{1/2}\left|\Sigma_2\right|^{1/2}}{\left|\Sigma_p\right|} = \frac{\left|\frac{1}{2}\left(\Sigma_1 + \Sigma_2\right)\right|}{\left|\Sigma_1\right|^{1/2}\left|\Sigma_2\right|^{1/2}}$$

A forma fechada para a expressão da distância de Bhattacharyya para dados com distribuição gaussiana pode então ser lida como:

$$B = \frac{1}{8}\left(M_1 - M_2\right)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} \left(M_1 - M_2\right) + \frac{1}{2}\ln\left(\frac{\left|\frac{1}{2}\left(\Sigma_1 + \Sigma_2\right)\right|}{\left|\Sigma_1\right|^{1/2}\left|\Sigma_2\right|^{1/2}}\right) \qquad (16)$$

Onde o primeiro termo da expressão (16) à direita estima a contribuição dos vetores média na separação das duas classes e o segundo estima a contribuição das matrizes de covariância.

Note que quando as matrizes de covariância para as duas classes são iguais, a distância de Bhattacharyya (*B*), e a distância da Divergência (*D*) são medidas equivalentes, tal que:

$$D = 8B = \left(M_1 - M_2\right)^T \Sigma^{-1}\left(M_1 - M_2\right)$$

Essa quantidade é também reconhecida como a distância de Mahalanobis entre duas classes.

3.5 Extração de feições via otimização da distância de Bhattacharyya

O método proposto para fins de extração de feições, implementa uma transformação linear cujo critério é o de otimizar a separação entre duas classes pelo critério da distância de Bhattacharyya. A otimização da distância de Bhattacharyya, entretanto, não é uma tarefa trivial, pelo fato de estarem presentes dois tipos diferentes de funções: o traço e o determinante de matrizes. Dessa forma para esse estudo, conforme Fukunaga (1990), são considerados casos sub-ótimos de otimização dessa distância.

3.5.1 As matrizes de covariância de ambas as classes são iguais: $\Sigma_1 = \Sigma_2$

Quando as matrizes de covariância são iguais, a expressão (16) pode ser resumida ao primeiro termo:

$$B_1 = \frac{1}{8}(M_2 - M_1)^T \bar{\Sigma}^{-1}(M_2 - M_1) \tag{17}$$

Note-se que o produto no membro da direita tem dimensão (1x1) e, portanto:

$$B_1 = \frac{1}{8} tr\left\{(M_2 - M_1)^T \bar{\Sigma}^{-1}(M_2 - M_1)\right\} \tag{18}$$

Dadas duas matrizes A e B, sabe-se que $tr(A.B) = tr(B.A)$. Aplicando-se em (18) tem-se:

$$B_1 = \frac{1}{8} tr\left\{\bar{\Sigma}^{-1}(M_2 - M_1)(M_2 - M_1)^T\right\} \tag{19}$$

No membro da direita da equação (19), observa-se que $(M_2 - M_1)(M_2 - M_1)^T$ é uma matriz (n x n) gerada pelo produto de um vetor por si próprio, tendo portanto *rank* = 1. Neste caso, $\bar{\Sigma}^{-1}(M_2 - M_1)(M_2 - M_1)^T$ também é uma matriz (n x n) com *rank* = 1, isto é, apenas um autovalor distinto de zero:

$$\lambda_1 \neq 0\,,\ \lambda_2 = \lambda_3 = ... = \lambda_n = 0$$

Então, de (8) tem-se:

$$B_1 = \frac{1}{8}\lambda_1$$

E,

$$\lambda_1 = (M_2 - M_1)^T \overline{\Sigma}^{-1} (M_2 - M_1)$$

Portanto, no espaço das feições, B_1 está alinhado segundo o primeiro autovetor e_1 de $\overline{\Sigma}^{-1}(M_2 - M_1)(M_2 - M_1)^T$, associado ao primeiro autovalor. Esta é, portanto, a direção da feição que maximiza B_1 entre duas classes.

Dado um vetor X no espaço das feições, com dimensão (n x 1), a feição ótima Y no sentido de máxima separação entre duas classes pelo critério da distância de Bhattacharyya (B_1) pode ser obtido projetando-se X sobre e_1:

$$\underset{(1x1)}{Y} = \underset{(1xn)}{e_1^T} . \overline{\underset{(nx1)}{X}}$$

$$Y = \underset{(1xn)}{(M_2 - M_1)^T} \overline{\Sigma}^{-1} \underset{(nx1)}{X}$$

A direção de e_1 pode ser obtida da seguinte forma: do problema geral de autovalores/autovetores em uma matriz genérica M:

$$M.e_1 = e_1.\lambda_1$$

No presente caso: $M = \overline{\Sigma}^{-1}(M_2 - M_1)(M_2 - M_1)^T$

Então:

$$\overline{\Sigma}^{-1}(M_2 - M_1)(M_2 - M_1)^T.e_1 = e_1.(M_2 - M_1)\overline{\Sigma}^{-1}(M_2 - M_1)^T$$

E, portanto:

$$e_1 = \overline{\Sigma}^{-1}(M_2 - M_1)$$

Assim, a feição ótima é:

$$Y = \left(\overline{\Sigma}^{-1}(M_2 - M_1)\right)^T .X$$

3.5.2 Os vetores média de ambas as classes são iguais: $\mu_1 = \mu_2$

Somente o segundo termo da distância de Bhattacharyya permanece, a contribuição das diferenças entre matrizes de covariância.

$$B_2 = \frac{1}{2}\ln\frac{\left|\frac{\Sigma_{1X} + \Sigma_{2X}}{2}\right|}{\sqrt{\left|\Sigma_{1X}\right|\left|\Sigma_{2X}\right|}}$$

$$B_2 = \frac{1}{4}\left[\ln\left|\Sigma_{2X}^{-1}\Sigma_{1X} + \Sigma_{1X}^{-1}\Sigma_{2X} + 2I\right| - n\ln(4)\right] \tag{20}$$

Onde, os subscritos 1 e 2 indicam as duas classes e o subscrito *X* indica que a matriz de covariância é estimada no espaço das feições originais *X*.

O objetivo aqui consiste em encontrar uma transformação linear que maximize a separação entre um par de classes. A forma geral de uma transformação linear é dada por:

$$Y = A\,X$$

Sendo, *X* o espaço das feições original, *Y* o espaço transformado, e *A* uma matriz ortogonal que realiza a transformação. Pode-se mostrar que a matriz de covariância no espaço transformado (Σ_Y) está relacionada com a matriz de covariância no espaço original (Σ_X) por:

$$\Sigma_Y = A^T\Sigma_X A$$

No espaço transformado, a expressão (20) fica:

$$(B_2)_Y = \left[\ln\left|\left(A^T\Sigma_{2X}A\right)^{-1}\left(A^T\Sigma_{1X}A\right) + \left(A^T\Sigma_{1X}A\right)^{-1}\left(A^T\Sigma_{2X}A\right) + 2I\right| - n\ln(4)\right]$$

A transformação linear *A* que maximiza B_2 pode ser obtida maximizando-se o termo:

$$J(n) = \ln\left|\left(A^T\Sigma_{2X}A\right)^{-1}\left(A^T\Sigma_{1X}A\right) + \left(A^T\Sigma_{1X}A\right)^{-1}\left(A^T\Sigma_{2X}A\right) + 2I\right|$$

Ou retendo-se apenas *m* componentes ($m < n$):

$$J(m) = \left[\ln\left|\left(A^T\Sigma_{2X}A\right)^{-1}\left(A^T\Sigma_{1X}A\right) + \left(A^T\Sigma_{1X}A\right)^{-1}\left(A^T\Sigma_{2X}A\right) + 2I_m\right|\right] \tag{21}$$

Maximizando-se *J(m)*:

$$\frac{\partial J(m)}{\partial A} = -2[*]^{-1}\left[\left\{\Sigma_{2X} A\Sigma_{2Y}^{-1}\Sigma_{1Y}\Sigma_{2Y}^{-1} - \Sigma_{1X} A\Sigma_{2Y}^{-1}\right\} + \left\{\Sigma_{1X} A\Sigma_{1Y}^{-1}\Sigma_{2Y}\Sigma_{1Y}^{-1} - \Sigma_{2X} A\Sigma_{1Y}^{-1}\right\}\right] = 0 \quad (22)$$

Sendo,

$$[*] = \left(A^T\Sigma_{2X}A\right)^{-1}\left(A^T\Sigma_{1X}A\right) + \left(A^T\Sigma_{1X}A\right)^{-1}\left(A^T\Sigma_{2X}A\right)$$

Para tornar $\frac{\partial J(m)}{\partial A}$ nulo, deve-se anular ambas as parcelas no segundo termo em (22):

$$\Sigma_{2X} A\Sigma_{2Y}^{-1}\Sigma_{1Y}\Sigma_{2Y}^{-1} - \Sigma_{1X} A\Sigma_{2Y}^{-1} = 0 \quad (23)$$

$$\Sigma_{1X} A\Sigma_{1Y}^{-1}\Sigma_{2Y}\Sigma_{1Y}^{-1} - \Sigma_{2X} A\Sigma_{1Y}^{-1} = 0 \quad (24)$$

Pré-multiplicando ambos os membros em (23) por Σ_{2X}^{-1} e pós-multiplicando por Σ_{2Y}, tem-se que:

$$A\,\Sigma_{2Y}^{-1}\Sigma_{1Y} = \Sigma_{2X}^{-1}\Sigma_{1X}\,A \quad (25)$$

Pré-multiplicando-se ambos os membros de (25) por Σ_{1X}^{-1} e pós-multiplicando-se por Σ_{1Y}, tem-se que:

$$A\left(\Sigma_{1Y}^{-1}\Sigma_{2Y}\right) = \left(\Sigma_{1X}^{-1}\Sigma_{2X}\right)A \quad (26)$$

Sendo *A* ortogonal, $A^T = A^{-1}$ e lembrando que:

$$\Sigma_Y = A^T\Sigma_X A$$

Assumindo uma matriz de transformação *A* comum para ambos Σ_{1X} e Σ_{2X}:

$$\Sigma_{1Y} = A^{-1}\Sigma_{1X}A \text{ e } \Sigma_{2Y} = A^{-1}\Sigma_{2X}A \quad (27)$$

E,

$$\Sigma_{2Y}^{-1} = \left(A^{-1}\Sigma_{2X}A\right)^{-1} = A^{-1}\Sigma_{2X}^{-1}A \quad (28)$$

Multiplicando (27) por (28):

$$\left(\Sigma_{2Y}^{-1}\Sigma_{1Y}\right) = A^{-1}\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)A$$

Conclui-se que a matriz *A* é a matriz de autovetores de $\Sigma_{2X}^{-1}\Sigma_{1X}$.

De uma maneira similar pode-se mostrar que:

$$\left(\Sigma_{2Y}^{-1}\Sigma_{1Y}\right) = A^{-1}\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)A$$

Isto é, *A* é a matriz de autovetores de $\Sigma_{1X}^{-1}\Sigma_{2X}$.

Assim, as condições de (23) e (24) são satisfeitas pela mesma matriz *A*, resultando que a matriz *A* satisfaz a condição de mínimo (22).

Note que $\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)$ e $\left(\Sigma_{1X}^{-1}\Sigma_{2X}\right)$ são relacionadas na forma $\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)^{-1} = \left(\Sigma_{1X}^{-1}\Sigma_{2X}\right)$. Ambas compartilham a mesma matriz de autovetores e as correspondentes matrizes de autovalores são uma a inversa da outra.

Conclui-se que tomando *A* como a matriz de autovetores de $\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)$, tem-se um ponto de extremo, isto é, a condição de mínimo é atingida e $\frac{\partial J(m)}{\partial A} = zero$. A matriz *A* implementa uma transformação que maximiza a distância de Bhattacharyya entre duas classes no caso de vetores de médias idênticos. Nesse caso, a redução de dimensões pode ser atingida tomando-se um número menor de componentes (*m*) no espaço transformado.

Um primeiro caso para um problema *n*-dimensional, é quando ambas as classes possuem mesmo vetor média, isto é, $\mu_1 = \mu_2$. Primeiro selecione *m* (*m* < *n*) autovetores de $\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)$. Então, a questão a ser formulada é quais *m* < *n* autovetores de $\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)$ devem ser selecionados.

Note que $\Sigma_{2X}^{-1}\Sigma_{1X} = \left(\Sigma_{1X}^{-1}\Sigma_{2X}\right)^{-1}$. Em geral,

$$A^{-1}\Sigma A = \Lambda \text{ e } \left(A^{-1}\Sigma A\right)^{-1} = \Lambda^{-1}$$

Assim,

$$A^{-1}\Sigma^{-1}A = \Lambda^{-1}$$

Conclui-se que a matriz de autovetores de $\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)$ é a inversa da matriz de autovalores de $\left(\Sigma_{1X}^{-1}\Sigma_{2X}\right)$. A função critério (21) pode então ser re-escrita como:

$$J(n) = \ln\left\{\left|\left(A\Sigma_{2X}A\right)^{-1}\left(A\Sigma_{1X}A\right) + \left(A^T\Sigma_{1X}A\right)^{-1}\left(A^T\Sigma_{2X}A\right)^{-1}\left(A\Sigma_{2X}A\right) + 2I\right|\right\} \qquad (29)$$

$$= \ln\left\{\left|A^{-1}\Sigma_{2X}^{-1}AA^{-1}\Sigma_{1X}A + A^{-1}\Sigma_{1X}^{-1}AA^{-1}\Sigma_{2X}A + 2I\right|\right\}$$

$$= \ln\left\{\left|\Sigma_{2X}^{-1}\Sigma_{1X} + \Sigma_{1X}^{-1}\Sigma_{2X} + 2I\right|\right\}$$

$$= \ln\left\{\left|\Lambda + \Lambda^{-1} + 2I\right|\right\}$$

$$= \ln\left\{\left|\Lambda\right| + \left|\Lambda\right|^{-1} + 2\left|I\right|\right\}$$

Como em uma matriz diagonal o determinante é igual ao somatório dos elementos ao longo da diagonal, segue que:

$$J(n) = \sum_{i=1}^{n} \ln\left(\lambda_i + \frac{1}{\lambda_i} + 2\right)$$

O objetivo aqui consiste em reduzir a dimensão dos dados com perda mínima de informação. Adota-se então a estratégia de selecionar os $m < n$ autovetores λ_i que geram os m maiores valores para a expressão acima de $J(n)$, isto é, os m maiores termos $\left(\lambda_i + \frac{1}{\lambda_i} + 2\right)$.

3.5.3 Distintos vetores média e matrizes de covariância: $\mu_1 \neq \mu_2$ e $\Sigma_1 \neq \Sigma_2$

Para o caso geral, quando temos ambas as componentes da distância de Bhattacharyya diferentes, isto é, $\mu_1 \neq \mu_2$ e $\Sigma_1 \neq \Sigma_2$, não há solução ótima. Dessa forma é necessário analisar qual dos termos apresenta diferenças dominantes e adotar um dos seguintes procedimentos sub-ótimos.

3.5.3.1 A diferença dominante ocorre entre os vetores média μ_1 e μ_2

Nesse caso, pode-se propor uma abordagem implementando seletivamente ambos os critérios anteriores. Este procedimento pode ser sumarizado pelas seguintes etapas:

a) Extrair uma feição contendo a contribuição devido à diferença entre as médias. Para isso, calcular os autovetores φ_i e autovalores λ_i de

$$\overline{\Sigma}^{-1}(M_2 - M_1)(M_2 - M_1)^T,$$

onde somente $\lambda_1 \neq 0$.

Utilize o correspondente φ_1 como a primeira feição y_1, dada por:

$$y_1 = \varphi_1^T X$$

Esta feição y_1 captura toda a separação disponível pela diferença entre os vetores média.

b) Mapeie *X* na direção dos (*n* – 1) autovetores restantes

$$y_i = \varphi_i^T X \qquad i = 2,3,\ldots,n.$$

Estas feições não contribuem para a separabilidade devida à diferença entre médias (B_1), mas contém a contribuição devido à diferença entre as matrizes de covariância (B_2).

c) Otimize B_2 no espaço (n–1)-dimensional, isto é, no espaço *Y*, calculando $\left(\Sigma_{2Y}^{-1}\Sigma_{1Y}\right)$. A seguir calcule seus autovalores e autovetores λ_i, φ_i, com $i = 1,\ldots,n-1$.

d) Selecione os *m* autovetores φ_i que estão associados com os maiores $\left(\lambda_i + \frac{1}{\lambda_i} + 2\right)$.

e) Mapeie *Y* em *Z*, tal que:

$$Z_i = \varphi_i^T Y\ ,\ i = 1,\ldots,m.$$

f) As (m+1) feições ótimas segundo o critério da distância de Bhattacharyya são, portanto: $\{y_1, z_1, z_2, \ldots, z_m\}$.

3.5.3.2 A diferença dominante ocorre entre matrizes de covariância Σ_1 e Σ_2

Novamente aqui, pode-se propor uma abordagem que implementa seletivamente ambos os critérios anteriores. Nesse caso, a idéia é utilizar estas direções para extrair as feições, mas selecionando aquelas que apresentam não somente alto valor para contribuição da componente B_2, mas também a contribuição de B_1, isto é, das médias. Este procedimento pode ser sumarizado pelas seguintes etapas:

a) Calcular Φ, a matriz de autovetores de $\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)$.

b) Calcular a matriz de autovalores Λ de $\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)$.

c) Aplicar a transformação $Y = \Phi^T X$

d) No sistema *Y*, a distância de Bhattacharyya é dada por:

$$B_Y = \sum_{i=1}^{n}\left[\frac{1}{4}\frac{\left\{\varphi_i^T\left(\mu_2 - \mu_1\right)\right\}^2}{1+\lambda_i} + \frac{1}{4}\left\{\ln\left(\lambda_i + \frac{1}{\lambda_i} + 2\right) - \ln(4)\right\}\right] \tag{30}$$

Na equação (30), o segundo termo no membro da direita é o critério para extração de feições com base na contribuição da diferença entre covariâncias e o primeiro termo inclui a contribuição da diferença entre médias.

e) Selecionar os *m* autovetores de $\left(\Sigma_{2X}^{-1}\Sigma_{1X}\right)$ que correspondem aos *m* maiores autovalores de B_Y.

3.6 Classificadores em estágio único e múltiplo-estágio

A Figura 3.4 ilustra a metodologia tradicional utilizada para classificação de padrões. A função de decisão $G_i(X)$ é estimada para cada uma das classes envolvidas no processo, sendo então o vetor *X* atribuído à classe vencedora.

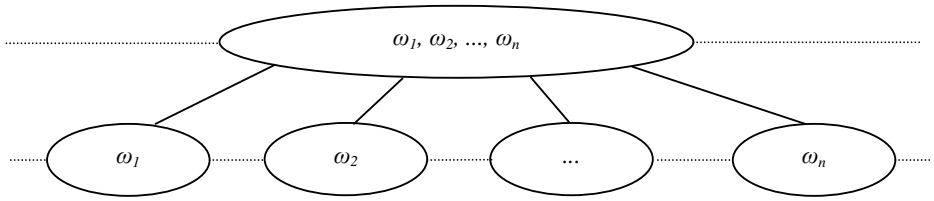
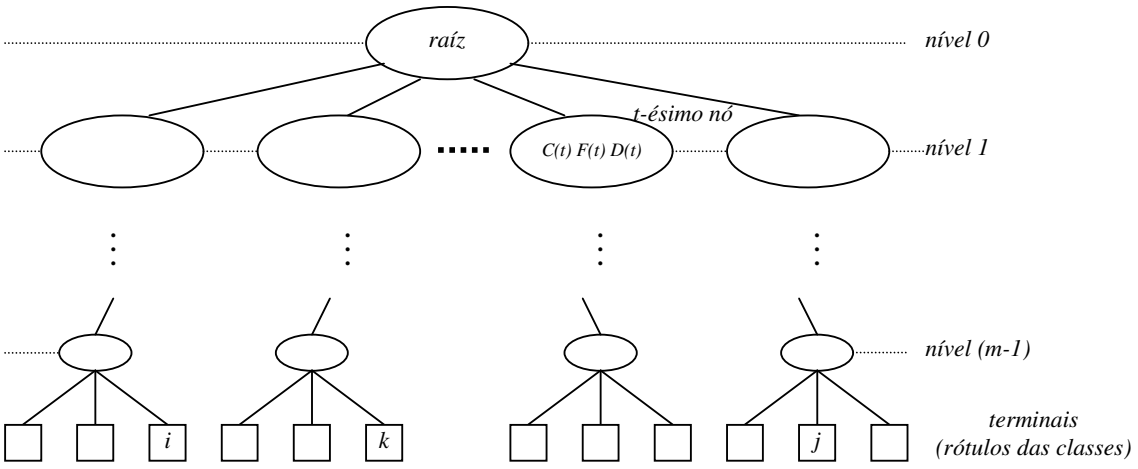


Figura 3.4 Classificador em estágio único

Esse classificador, que implementa a função de decisão (12) simultaneamente para todo o conjunto das classes envolvidas é denominado de classificador em estágio único. Nesta abordagem, o processo de classificação é efetuado de uma forma global, isto é, considerando todas as classes em uma única etapa. Esse fato tem uma conseqüência direta no problema de seleção das feições a serem empregadas no processo de classificação, isto é, as feições selecionadas devem satisfazer a um critério de otimização global na performance do classificador, considerando-se simultaneamente todas as classes envolvidas.

Uma outra possível abordagem para os casos multi-classe consiste no uso dos classificadores hierárquicos, onde o problema global é dividido em etapas, envolvendo um número menor de classes a cada etapa. Nessa abordagem, no processo de classificação de cada individuo, somente um subconjunto de classes é considerado a cada etapa.

O Classificador de Decisão em Árvore (CDA) é um tipo especial de método de classificação em estágio múltiplo, ou classificador hierárquico. Uma das características desse classificador é que ele permite a rejeição de classes em estágios intermediários, simplificando a solução final do problema, pois cada padrão só precisa ser comparado com um conjunto reduzido de classes. No caso de dados em alta dimensão, como em imagens hiper-espectrais, esse aspecto assume especial relevância, pois permite selecionar a cada etapa o subconjunto de feições com maior poder discriminante relativo ao subconjunto de classes consideradas. Esse processo também permite otimizar o método de redução na dimensão dos dados via seleção ou extração de feições, reduzindo os efeitos do fenômeno de Hughes.



C(t) – subconjunto de classes no t-ésimo nó

F(t) – subconjunto de feições no t-ésimo nó

D(t) – regra de decisão no t-ésimo nó

Figura 3.5 Classificador de decisão em árvore (Safavian, 1991)

Safavian (1991), define a estrutura geral de um CDA conforme o esquema apresentado na Figura 3.5. Nessa estrutura, a raiz da árvore é apresentada no nível zero por um único nó, contendo todos os padrões não discriminados pertencentes às *n* classes. Cada nó *t* é composto por uma terna (*C(t)*, *F(t)*, *D(t)*), onde *D(t)* representa a regra de decisão que utiliza o subconjunto de feições *F(t)* para discriminar os padrões contidos no nó *t* entre as *C(t)* classes. Esse processo é repetido ao longo dos ramos da árvore, até que não seja mais possível discriminar as amostras entre si. Neste caso, o nó torna-se um nó terminal e recebe um rótulo correspondendo a uma classe específica.

Recentemente, Tadjudin e Landgrebe (1998) propuseram um algoritmo para CDA na forma binária, similar ao que será adotado nessa dissertação. Diferentemente do exemplo geral ilustrado na Figura 3.4, esse método considera apenas um par de classes a cada nó, subdividindo todas as classes presentes em um nó em apenas dois subconjuntos de classes, similares ao par escolhido. Dessa forma, o CDA binário pode ser resumido nas seguintes etapas:

a. Definir um critério para estimar o grau de separação entre pares de classes;

b. Selecionar o par que apresentar a maior separação. Estimar o vetor média e matriz de covariância para estas duas classes e utilizá-los como estatísticas. Estas estatísticas servirão para definir os dois nós subseqüentes (esquerdo e direito).

c. Classificar as amostras em um dos dois nós utilizando a regra de decisão da Máxima Verossimilhança Gaussiana (12):

$$G_i(X) = -\ln|\Sigma_i| - (X - M_i)^T \Sigma_i^{-1} (X - M_i)$$

Então:

$$X \in n_L \qquad \text{se} \qquad G_{nL}(X) < G_{nR}(X)$$

Onde,

$i = n_L \ e \ n_R$, representam o nó esquerdo e direito, respectivamente;

X: amostra com *p* feições;

M_i: média amostral estimada do nó *i*;

Σ_i: matriz de covariância estimada no nó *i*;

d. Se desse processo de classificação resultar que o número de amostras pertencentes a uma determinada classe supera um limiar previamente estabelecido, a classe é considerada como pertencente a este nó e a totalidade de suas amostras são atribuídas a este nó. Caso contrário, a classe é atribuída a ambos os nós descendentes. Esse processo continua até que os nós terminais sejam atingidos.

O método para delineamento do CDA descrito acima tem as seguintes características desejáveis:

i. Em se utilizando uma abordagem *top-down*, na qual a raiz da árvore consiste de um grupo pré-definido de classes, o CDA garante que as classes têm valor informacional, isto é, os nós terminais corresponderão a classes conhecidas;

ii. Separando-se primeiro as classes com maior distância, a ocorrência de sobreposições de classes pode ser reduzida, diminuindo-se assim o tamanho da árvore;

iii. Utilizando duas classes, ao invés de dois subgrupos de classes, evita-se o problema de emparelhamento, que é a união de duas classes em uma só. Em adição a isso, desde que as classes são assumidas como normalmente distribuídas, a regra de classificação da Máxima Verossimilhança Gaussiana pode ser prontamente aplicada como uma regra de divisão do nó;

iv. Na estrutura do CDA binário, somente duas classes são consideradas em cada nó. Essa estrutura apresenta a notável vantagem de permitir a utilização direta de distâncias estatísticas como, por exemplo, a distância de Bhattacharyya, no processo de seleção de feições, a qual não seria viável no caso de múltiplas classes.

No capítulo 4, serão implementados os experimentos com a aplicação da otimização da distância de Bhattacharyya para esse último caso, onde a diferença predominante se dá entre as matrizes de covariâncias. Esse método será usado em conjunto com o CDA em estrutura binária, de forma a possibilitar o uso da otimização da distância de Bhattacharyya entre pares de classes.

A escolha dessa otimização específica é devida às características espectrais das classes escolhidas para análise, as quais possuem vetores de médias muito semelhantes e matrizes de covariâncias distintas. Dessa forma, utilizando dados hiperespectrais do sensor AVIRIS, procurou-se obter uma melhor separação entre as classes com base na contribuição predominante das diferenças entre as matrizes de covariâncias.

4 EXPERIMENTOS

4.1 Introdução

Neste capítulo são descritos e analisados os experimentos realizados de acordo com a metodologia proposta no capítulo 3, isto é, a classificação de dados imagem hiper-espectrais, empregando um classificador de decisão em árvore (CDA) com estrutura binária e redução na dimensão dos dados em cada nó por meio de um processo de extração de feições sob o critério de otimização da distância de Bhattacharyya no caso especial de diferença predominante entre as matrizes de covariâncias. Os experimentos foram desenvolvidos empregando-se dados imagem em alta dimensão, cobrindo uma área teste no Estado de Indiana (USA). Na cena imageada estão presentes classes espectralmente muito semelhantes, isto é, classes que apresentam vetores de média muito semelhantes entre si, sendo portanto de difícil separação. O desenvolvimento desse capítulo inicia com a apresentação dos dados sobre a imagem hiper-espectral utilizada. A seguir, são definidas as características do CDA com estrutura binária e o critério de pertinência das classes em cada nó.

Inicialmente foi investigada a estrutura do CDA mais adequada aos dados utilizados, isto é, aquela que produziu a acurácia mais elevada. Nesse procedimento inicial, foi empregada a cena inteira, com as seis classes disponíveis. Os demais experimentos foram desenvolvidos em uma área mais restrita utilizando cinco classes, devido ao fato de uma das classes não possuir amostras suficientes para os testes com amostras de tamanho igual a 500. Nesses experimentos foi empregado o mesmo número limitado de amostras para todas as classes, a fim de isolar possíveis fatores de variação na comparação entre as metodologias visando a redução na dimensão dos dados. O tamanho dos conjuntos de amostras de treinamento utilizados foi igual a 500, 350, 150, 100 e 70. Com isso, buscou-se verificar a evolução gradual do comportamento das metodologias de classificação testadas.

A eficiência da metodologia proposta nesta dissertação é avaliada comparando-se os resultados obtidos com aqueles produzidos por outros métodos bem conhecidos de classificação para redução na dimensão dos dados. Desta forma, os experimentos foram repetidos empregando-se um CDA com extração de feições em cada nó via técnicas de Análise Discriminante Canônica e também um classificador em estágio único com redução na dimensão dos dados via seleção de feições. Nesse ultimo caso, foi empregado o processo conhecido como *Sequential Forward Selection* (SFS).

No final do capítulo são feitas as considerações sobre o tempo de processamento das metodologias em função da dimensão utilizada para classificação.

Para a implementação de todos os experimentos, foi utilizada a linguagem do aplicativo comercial MATLAB®, sendo que as rotinas desenvolvidas estão inseridas nos anexos finais.

4.2 Dados sobre a imagem utilizada

Os experimentos foram realizados utilizando-se dados do sensor AVIRIS, o qual captura 224 bandas espectrais com a largura 10nm cada, cobrindo uma região do espectro eletromagnético que se estende de 0.4 μm a 2.5 μm em comprimento de onda (λ). Os dados imagem são, portanto, dados em alta dimensão (dados hiper-espectrais). A cena compreende uma área teste da Purdue University, mostrando uma região agrícola no Estado de Indiana, USA. As Figuras 4.1a e 4.1b ilustram respectivamente uma composição colorida (RGB) produzida com três bandas espectrais da imagem analisada e o mapa temático com a verdade terrestre.

Figura 4.1a Imagem hiper-espectral AVIRIS

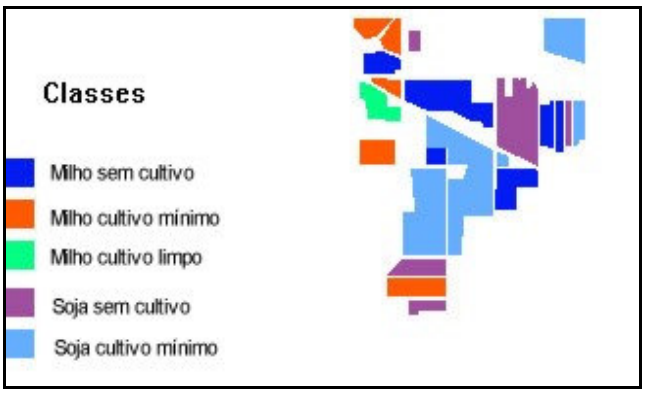


Figura 4.1b Verdade terrestre das classes

Em especial, cinco classes de culturas agrícolas foram analisadas nesse trabalho por apresentarem alta semelhança espectral, constituindo-se em um desafio maior para o classificador. A Figura 4.2 ilustra o alto grau de semelhança entre as curvas de resposta espectral das cinco classes, as quais se referem a três tipos de manuseio do solo para o plantio de milho e dois tipos de manuseio para o plantio de soja.

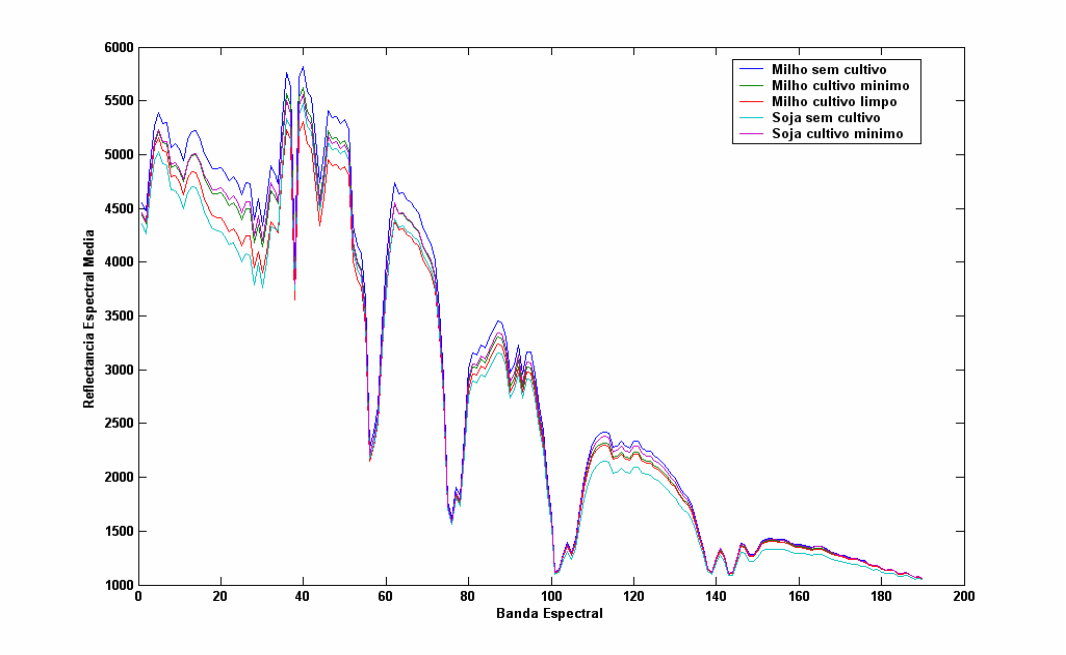


Figura 4.2 Curva de resposta espectral média das cinco classes em estudo

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130
131	132	133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170
171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190
191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220
221	222	223	224						

Figura 4.3 Bandas utilizadas do sensor AVIRIS

Das 224 bandas, foram excluídas as que apresentaram ruído causado pela presença de vapor d'água na atmosfera, restando 190 bandas ilustradas na Figura 4.3. Após o processamento dos primeiros resultados, foi verificada alta correlação entre bandas adjacentes ($r \approx 0.99$), sugerindo assim a seleção sistemática de um terço das bandas

originais para o processamento dos resultados principais. Com apenas as 64 bandas, foi possível uma estimação mais precisa dos vetores de médias e das matrizes de covariâncias, minimizando dessa forma os efeitos do *fenômeno de Hughes* e aumentando conseqüentemente a acurácia final de classificação, além de reduzir expressivamente o tempo de processamento.

4.3 Estrutura binária do CDA

O primeiro passo no processo de escolha do modelo hierárquico de classificação, consistiu no estudo do algoritmo para o CDA. Essa fase inicial utiliza todas as 190 bandas disponíveis do sensor. A distância de Bhattacharyya oferece um medida conveniente de separação entre duas classes. Um processo de extração de feições com base na otimização da distância de Bhattacharyya mostra-se portanto adequado. O CDA com estrutura binária apresenta, dessa forma, a vantagem de permitir em cada nó uma extração otimizada de feições. O método escolhido para a criação do CDA foi o *top-down*, conforme descrito no capítulo 2. É importante considerar que existem muitas possibilidades de subdivisões em uma árvore binária, tornando-se necessário introduzir a notação ilustrada na Figura 4.4.

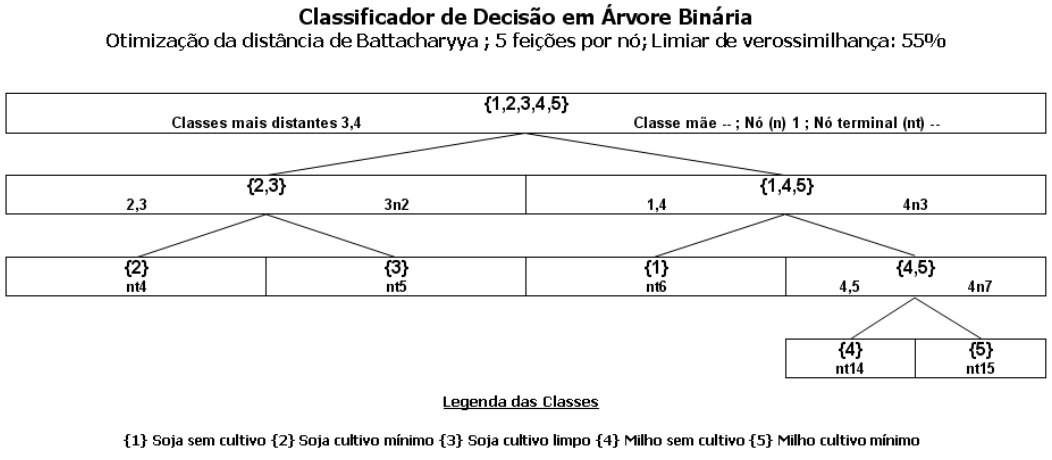


Figura 4.4 Exemplo de CDA com estrutura binária e cinco classes

Nesta figura, em cada nó do CDA são apresentadas as informações relevantes do processo. Os números no primeiro conjunto {1,2,3,4,5} se referem à legenda numérica das classes a que pertencem as amostras nesse nível inicial da árvore. O par 3,4 à esquerda, expressa o par de classes que apresenta a maior separação entre si. O número situado à

direita do nó refere-se à classe cujos parâmetros (vetor média e matriz de covariância) foram utilizados na seleção das amostras que o compõe. O algarismo inicial refere-se à classe cujos parâmetros foram utilizados no processo de seleção de amostras alocadas ao nó. Esta convenção não se aplica ao nó inicial (classe mãe). O número que define o nó encontra-se indicado após o símbolo 'n' e os nós terminais ficam identificados pelo símbolo 'nt'.

4.4 Limiar de verossimilhança (LV)

Para a definição dos nós descendentes do CDA, um outro conceito que se mostrou necessário nos experimentos é o do limiar de verossimilhança (LV). O LV é um parâmetro introduzido no estudo com a finalidade de definir o critério de pertinência de uma classe aos nós descendentes. O LV estipula a fração dos indivíduos na amostra de treinamento que devem ser alocados em um nó, para que a classe em questão seja alocada unicamente a este nó. Os experimentos mostraram que diferentes valores para o LV apresentam uma influência significativa na forma do CDA, isto é, na quantidade de nós terminais, resultando em diferentes valores para a acurácia produzida pelo classificador. Este conceito é descrito mais detalhadamente por meio do exemplo que segue.

Suponha-se que num determinado nó estejam presentes amostras pertencentes a três classes distintas, **A**, **B** e **C**, conforme ilustrado na Figura 4.5. Suponha-se ainda que o par de classes **B** e **C** apresenta o maior grau de separação entre todos os possíveis pares. Todas as amostras dessas duas classes, portanto, serão alocadas nos nós descendentes direito e esquerdo, respectivamente. Devemos decidir agora em qual nó descendente as amostras da classe **A** serão alocadas com base nos parâmetros das classes **B** e **C**.

Neste processo, três possíveis casos podem ocorrer com relação às amostras pertencentes à classe **A:** podem ser alocadas apenas no nó descendente esquerdo (Figura 4.5, Caso 1), apenas no direito (Caso 2) ou em ambos (Caso 3). Nos primeiros dois casos, a classe **A** passa a pertencer ao conjunto de apenas um dos nós descendentes, o que tornaria o outro nó, um nó terminal. No terceiro caso, a classe **A** estaria presente em ambos nós descendentes, de forma que o processo descendente continuaria no ramo esquerdo e direito do CDA.

O LV define assim, a fração mínima de amostras da classe **A** que precisa ser classificada nos nós descendentes para que a classe seja considerada como pertencente a esses nós. Nesse caso, a totalidade das amostras de **A** são então, atribuídas ao nó

descendente vencedor. Assim, por exemplo, se a fração das amostras pertencentes à classe **A,** atribuídas ao nó descendente esquerdo, for maior que o LV adotado, assume-se que este nó é o mais representativo para a classe **A** e conseqüentemente a totalidade das amostras dessa classe são alocadas nesse nó, o que torna o nó descendente direito, terminal (Caso 1). Caso contrário, assume-se que nenhum dos dois nós descendentes seja suficientemente representativo para a classe **A**, e suas amostras são, portanto, repetidas nos dois nós e o processo de partição se repete para ambos os nós descendentes (Caso 3).

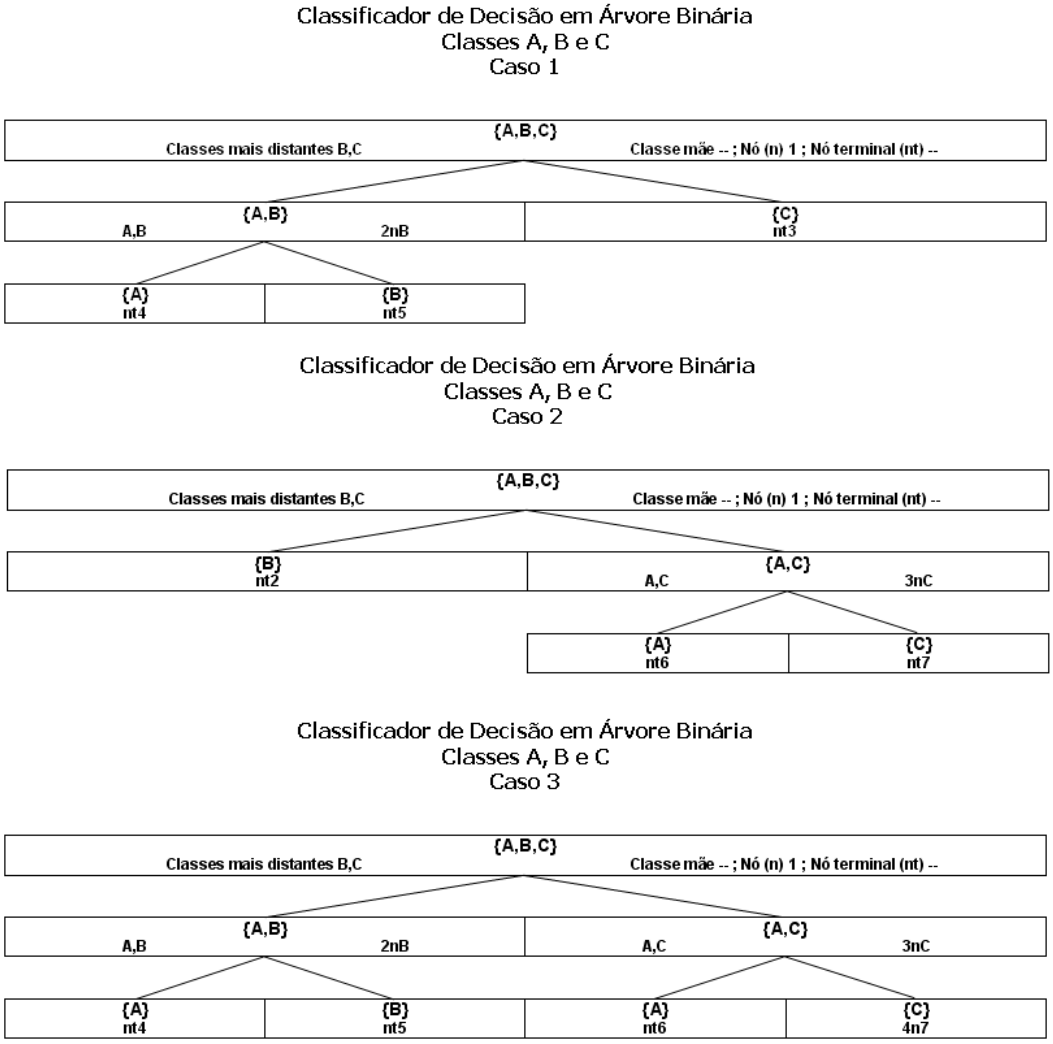


Figura 4.5 Formas de sub-dividir uma árvore binária com três classes A, B e C

4.4.1 Experimentos do LV envolvendo seis classes

Conforme foi comentado na introdução desse capítulo, os primeiros experimentos foram desenvolvidos utilizando a totalidade dos dados imagem disponíveis, isto é, toda a cena imageada. Seis classes são disponíveis conforme ilustrado na Tabela 4.1.

Tabela 4.1 Total de amostras nos experimentos com seis classes

Código da classe	Nome	Amostras de treinamento
1	Milho sem cultivo	1.006
2	Milho cultivo mínimo	656
3	Milho cultivo limpo	1.821
4	Soja sem cultivo	881
5	Soja cultivo mínimo	1.671
6	Soja cultivo limpo	373

A estrutura do CDA produzido com LV igual a 55% e 99% é ilustrada nas Figuras 4.6 e 4.8, respectivamente. As Figuras 4.7 e 4.9 mostram os resultados de acurácia individual para cada uma das seis classes nas diferentes dimensões e para os dois níveis de LV.

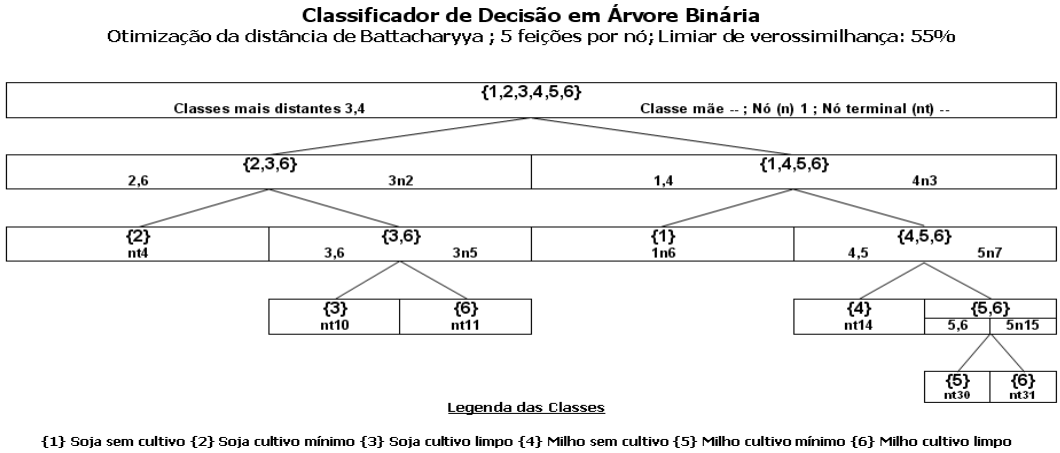


Figura 4.6 Árvore de classificação com seis classes e LV = 55% e 5 feições

Os experimentos evidenciaram (Figura 4.7) que com um LV igual a 55%, os valores estimados para a acurácia apresentam grandes variações entre as seis classes e em uma mesma classe em função da dimensão dos dados. Para dados com dimensão igual a 25, por exemplo, a acurácia estimada para a classe soja cultivo limpo foi próxima de 25%, enquanto para uma dimensão igual a 30, a acurácia estimada dessa

classe foi superior a 85%. Note-se que a estrutura do CDA (número de nós terminais) difere também para esses dois valores na dimensão dos dados.

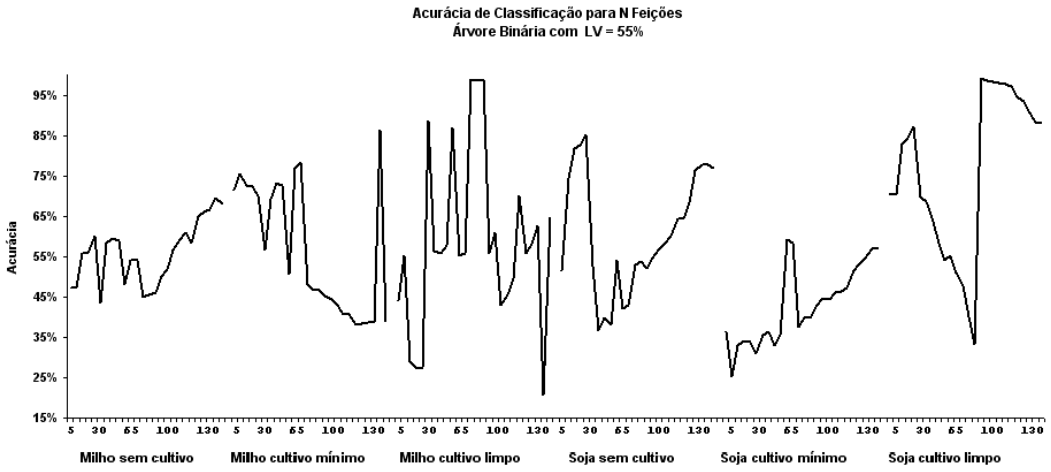


Figura 4.7 Acurácia de classificação das classes no CDA com LV igual a 55%

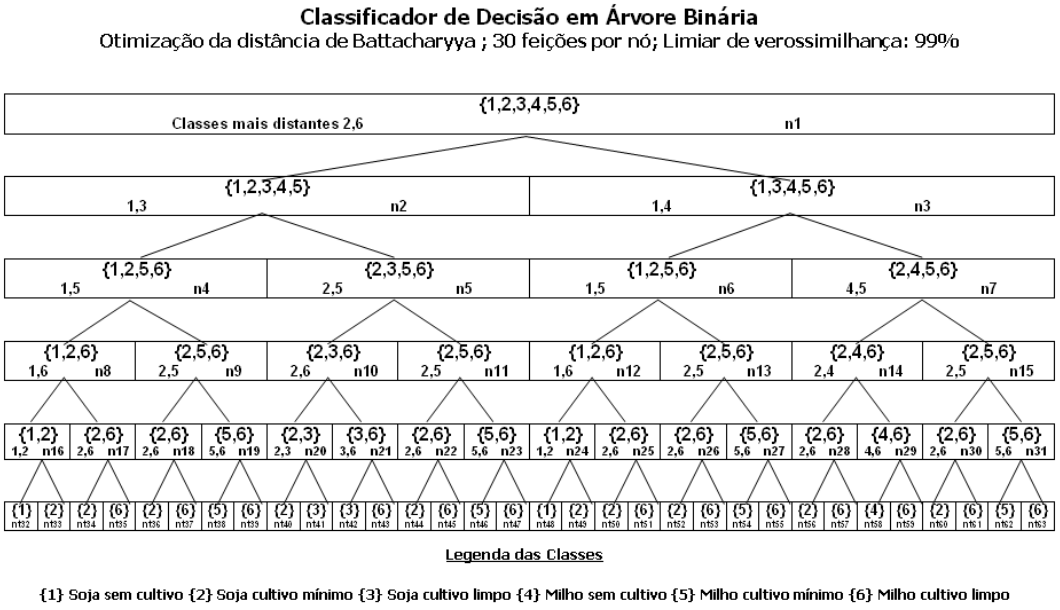


Figura 4.8 Árvore de classificação com seis classes e LV = 99% e 30 feições

Os resultados do experimento implementando um LV igual a 99% encontram-se ilustrados nas Figuras 4.8 e 4.9. Examinando a estrutura do CDA produzido por esse nível de LV (Figura 4.8), verifica-se que a mesma não é afetada pela dimensão dos dados, resultando sempre em um número máximo de nós terminais. A Figura 4.8 ilustra a estrutura

do CDA com um LV de 99% e 30 feições por nó, sendo que o número de nós terminais permanece constante para valores diferentes na dimensão dos dados.

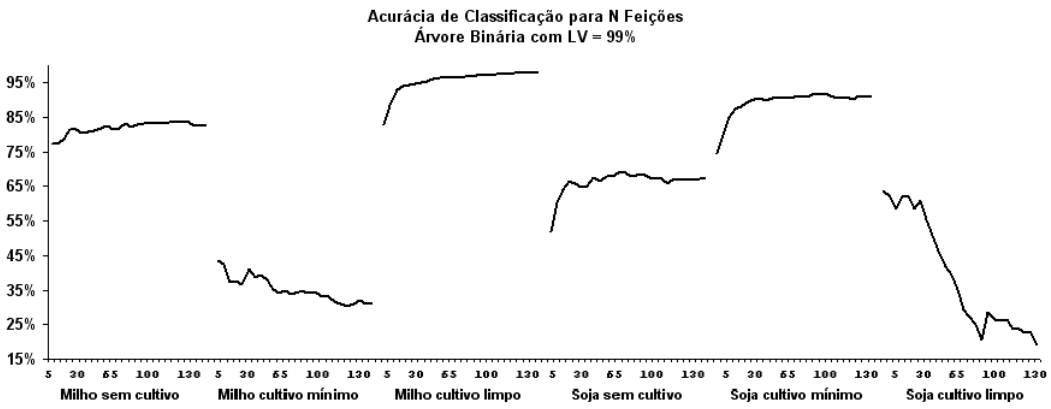


Figura 4.9 Acurácia de classificação das classes no CDA com LV igual a 99%

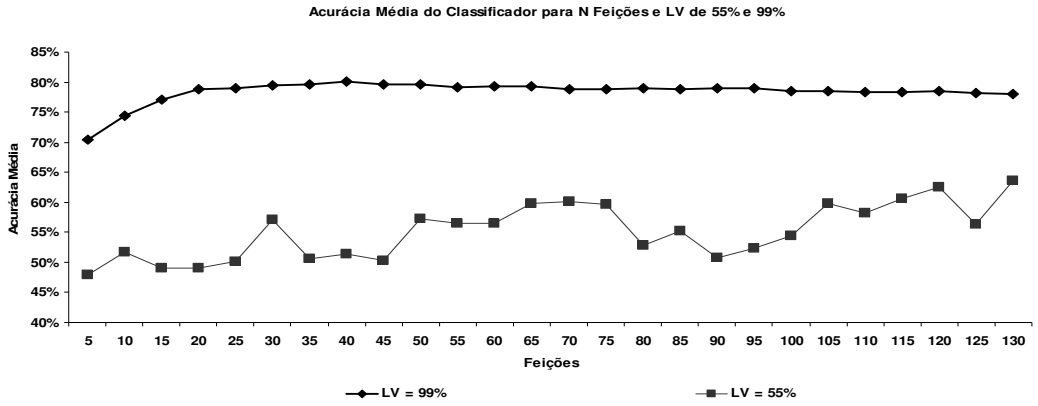


Figura 4.10 Acurácia média entre as seis classes para o CDA com LV igual a 55% e 99%

A acurácia média de classificação envolvendo as seis classes, em função da dimensão e dois valores distintos para LV é ilustrada na Figura 4.10. Nota-se que o CDA com LV igual a 99%, com a estrutura fixa igual a que é apresentada na Figura 4.8, apresenta melhores resultados do que o CDA com estrutura variável, exemplificado na Figura 4.6, para qualquer dimensão utilizada.

4.4.2 Experimentos do LV envolvendo cinco classes

Na segunda fase dos experimentos decidiu-se eliminar a classe *soja cultivo mínimo* devido ao pequeno tamanho do conjunto de amostras disponíveis para treinamento do classificador. Da mesma forma que para os experimentos envolvendo seis classes, foram inicialmente utilizadas todas as amostras e bandas disponíveis (Tabela 4.2). Para o parâmetro LV, foram utilizados nestes experimentos três valores, (55%, 75% e 100%). Embora os resultados com LV igual 99% tenham sido satisfatórios no caso anterior, os resultados com cinco classes evidenciaram que esse LV foi insuficiente para gerar um CDA com número máximo de nós terminais. Como resultado disso foram geradas árvores "incompletas", sendo então necessário fixar o LV em 100% para que fosse obtida sempre a maior estrutura possível, ou seja, o número máximo de nós terminais.

Tabela 4.2 Total de amostras nos experimentos com cinco classes

Código da classe	Classe	Amostras de treinamento
1	Milho sem cultivo	530
2	Milho cultivo mínimo	610
3	Milho cultivo limpo	566
4	Soja sem cultivo	1257
5	Soja cultivo mínimo	1163

Analisando as Figuras 4.11, 4.12 e 4.13, observa-se que valores mais altos para o LV produzem, uma menor variabilidade no valor estimado da acurácia de cada classe individual, em função da dimensão dos dados. Dessa forma, conforme observado também nos experimentos com seis classes, os resultados produzidos com LV igual a 100% produziram estruturas maiores para o CDA, com o número máximo de nós terminais.

Deve-se observar que, na estrutura do CDA proposta nesse estudo, somente o número de feições é idêntico em cada nó. As feições extraídas são peculiares a cada nó em particular, conforme a metodologia descrita no capítulo 3.

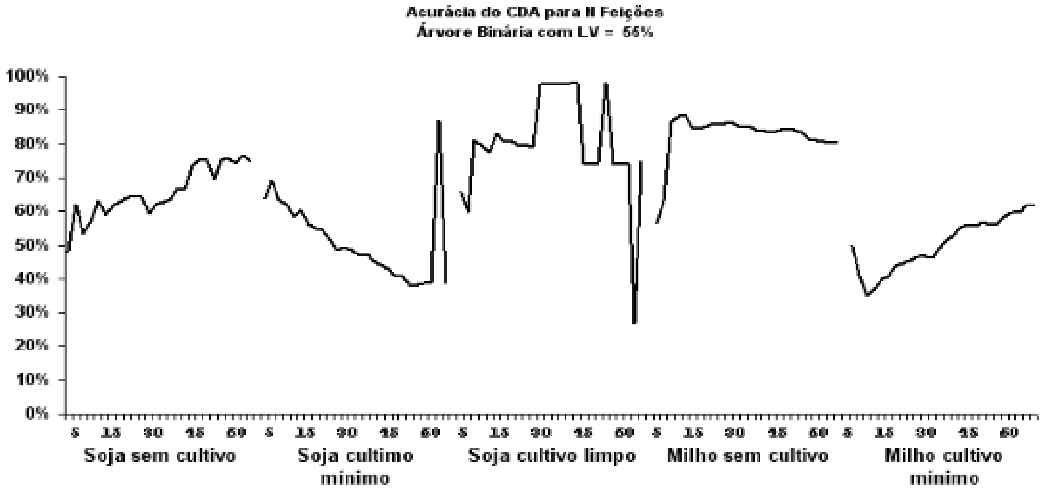


Figura 4.11 Acurácia de classificação para o CDA com LV igual a 55%

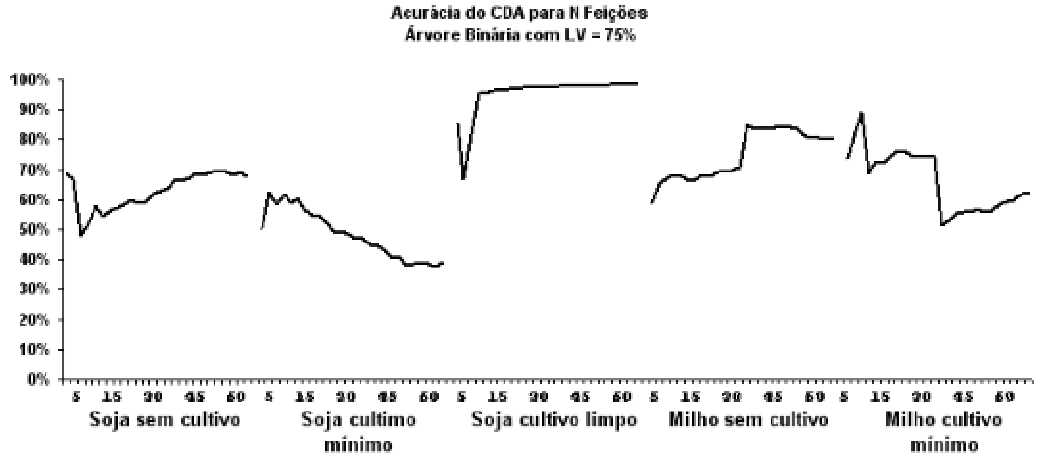


Figura 4.12 Acurácia de classificação para o CDA com LV igual a 75%

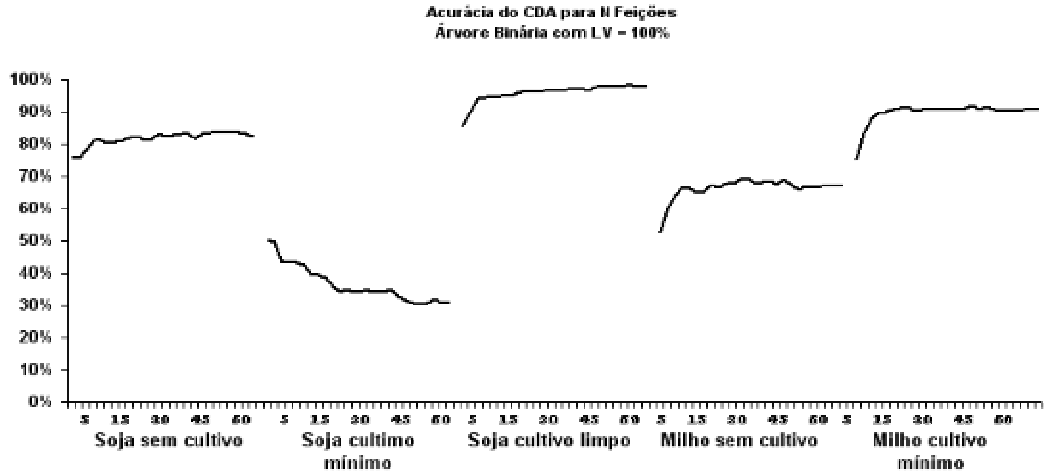


Figura 4.13 Acurácia de classificação para o CDA com LV igual a 100%

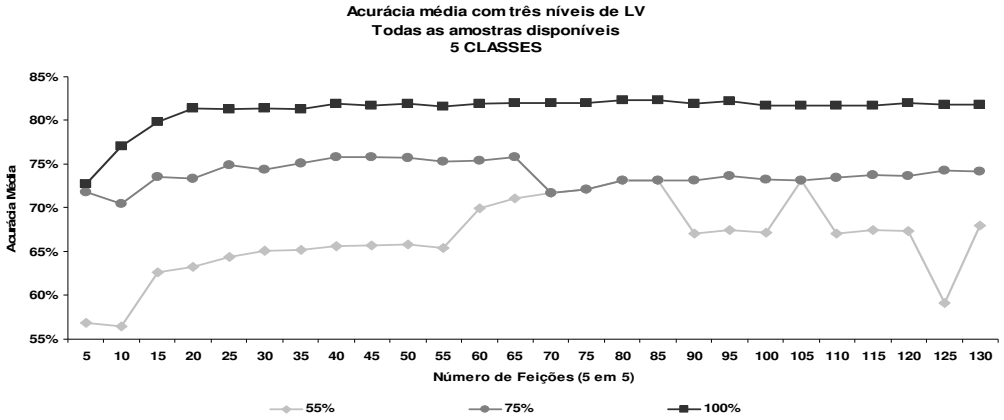


Figura 4.14 Acurácia média de classificação para o CDA com LV igual a 55%, 75% e 100%

Analisando os valores estimados para a acurácia média entre as cinco classes (Figura 4.14), observou-se que o CDA com a estrutura ilustrada na Figura 4.15, apresentou um padrão de acurácia bem definido e superior ao CDA com um número inferior de nós terminais. Decidiu-se então, optar pela estrutura do CDA que apresentou número máximo de nós terminais, ficando assim o trabalho resumido apenas em definir a dimensão a ser utilizada no processo de classificação.

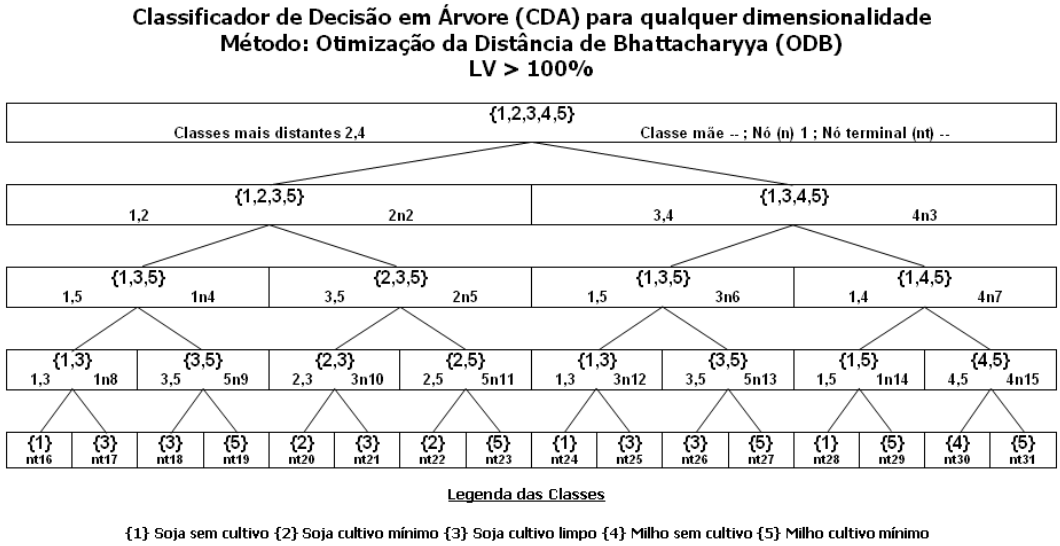


Figura 4.15 Forma final da árvore de classificação (LV = 100%)

4.5 Número de feições utilizadas no CDA com a ADC

O comportamento do CDA com diferentes níveis de LV foi testado também com o método para extração de feições da ADC, o qual apresentou resultados similares aos obtido pelo método da ODB. Dessa forma, optou-se também nesse caso pela utilização da estrutura para o CDA com o mesmo formato obtido via ODB, isto é, uma árvore binária com o maior número de terminais possível, similar ao ilustrado pela Figura 4.15.

Entretanto, não é viável neste caso empregar um número de feições em cada nó idêntico ao dos experimentos envolvendo a abordagem ODB, devido às restrições impostas pela ADC. Assim, a quantidade de feições extraídas foi decrescendo ao longo dos níveis da árvore, sendo c-1, c-2, c-3 e c-4, com c igual ao número de classes em cada nó. Nesse caso, como inicialmente o conjunto de classes era igual a cinco, foram utilizadas desde quatro feições para o nó inicial, até uma única feição no penúltimo nível do classificador. Tal fato é devido à restrição da ADC, que define o número de variáveis canônicas como sendo o mínimo entre o número de classes menos um e o número de variáveis menos um, conforme comentado no capítulo anterior.

Dessa forma, foi estipulado para o CDA em conjunto com a ADC utilizar sempre o número máximo de feições possível a cada nó. Para os experimentos com cinco classes então, o número de feições extraídas nos níveis do CDA foi igual a quatro, três, dois e um, respectivamente.

4.6 Redução sistemática das bandas originais

Concluída a análise da estrutura definitiva do CDA em função do valor de LV, procedeu-se ao estudo da possibilidade de redução sistemática das bandas originais. Devido à alta correlação entre bandas adjacentes ($r \approx 0.99$), as bandas utilizadas foram reduzidas a um terço das 190 originais, sendo assim utilizadas 64 bandas. Tal redução sistemática de bandas proporcionou melhores estimativas para os parâmetros, frente ao número limitado de amostras. Salienta-se que essa redução foi importante também devido ao tempo de processamento necessário para gerar uma única árvore com as amostras de treinamento e classificar as amostras de teste. A título de exemplo, com 120 feições extraídas em cada nó, o CDA demandou um tempo superior à uma hora de processamento. Como será mostrado mais adiante, o tempo de processamento

necessário para o CDA em conjunto com a ODB aumenta exponencialmente com o aumento da dimensão utilizada.

4.7 Padronização do número de amostras de treinamento

O prosseguimento das análises seguiu com a padronização do número de amostras de treinamento em todas as classes, em todos os experimentos, com o propósito de isolar fatores externos e também tornar possível a comparação das metodologias, bem como testar a acurácia de classificação em situações cada vez mais críticas, isto é, com um número de amostras de treinamento cada vez menor. A seguir, são discutidos os resultados dos experimentos envolvendo a metodologia proposta para redução de feições (ODB) e também com o método tradicional da ADC.

Também com o propósito de comparar a eficiência do classificador proposto com um número limitado de amostras, foi implementado outro classificador tradicional em estágio único, utilizando a técnica para seleção de feições conhecida como *Sequential Forward Selection* (SFS). Nesse processo foi utilizada a distância de Jeffries-Matusita (Serpico *et al*., 1995) como medida de separação entre todas as classes conjuntamente.

4.7.1 Resultados utilizando conjuntos de amostras de treinamento de igual tamanho para todas as classes

Inicialmente, o número de amostras de treinamento foi padronizado em 500 *pixels* por classe. Observou-se que com este número, o classificador CDA/ODB produziu melhores resultados a partir da dimensão igual a 10 feições por nó (74,4%), conforme ilustrado na Figura 4.16. A curva representando o valor estimado para a acurácia em função da dimensão dos dados apresenta o padrão esperado. Inicialmente ocorre uma tendência de aumento na acurácia, na medida em que aumenta a dimensão dos dados, isto é, na medida em que informação adicional é acrescentada ao classificador. O valor estimado para a acurácia atinge eventualmente um máximo, passando a declinar com o contínuo aumento da dimensão, numa clara caracterização do fenômeno de Hughes.

O classificador CDA/ADC, que utiliza no máximo 4 feições no primeiro nível da árvore, produziu um valor estimado para a acurácia média entre todas as classes igual a 74,1%. Este classificador apresenta ainda a vantagem de ser mais eficiente em termos de tempo de processamento do que o método da ODB. O método SFS no classificador em estágio único mostrou-se inferior ao uso do CDA em qualquer dimensão, com sua melhor acurácia média observada com 55 feições (74,5%).

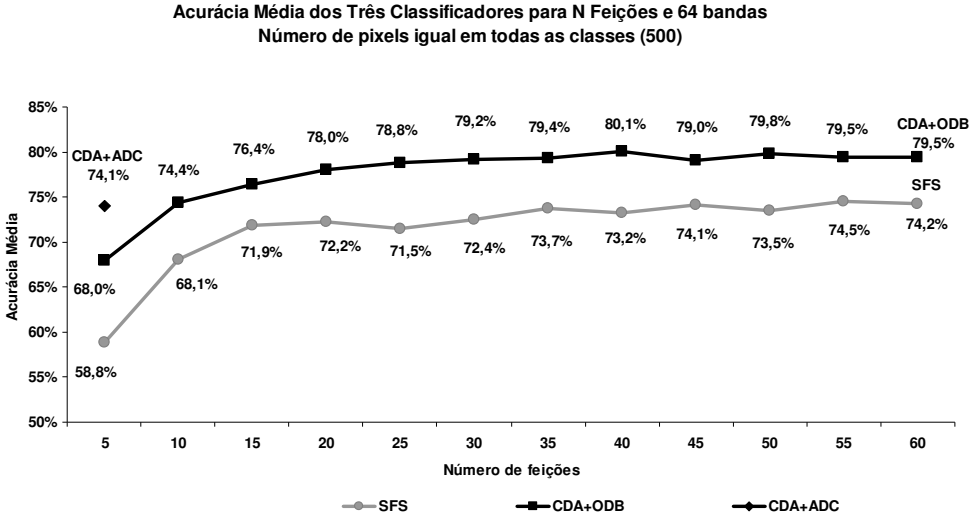


Figura 4.16 Acurácia média dos três métodos com N feições e amostras de tamanho 500

Reduzindo o tamanho da amostra de treinamento para 300 *pixels*, pode-se observar através da Figura 4.17, que o método CDA/ODB inicia com produzindo uma acurácia superior àquela produzida pelo classificador SFS e inferior ao CDA/ADC. A partir da dimensão igual a 10 feições, o SFS supera a performance do CDA/ODB, mas apresenta claramente o efeito do fenômeno de Hughes a partir de 25 feições. Nota-se que o método CDA/ODB evolui lentamente e chega a superar o SFS quando utilizadas 60 feições, sugerindo ser mais robusto com respeito ao fenômeno de Hughes do que o classificador em estágio único empregando a técnica SFS.

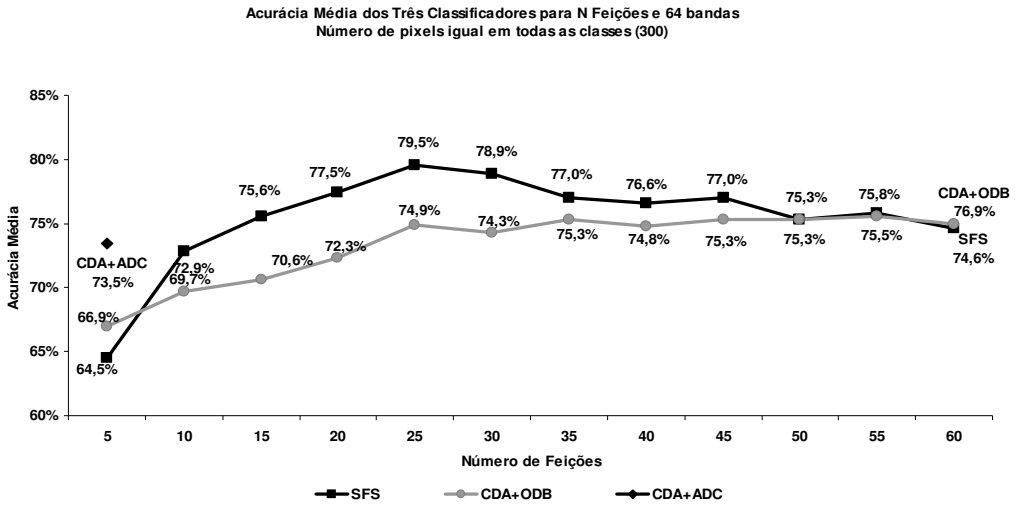


Figura 4.17 Acurácia média dos três métodos com N feições e amostras de tamanho 300

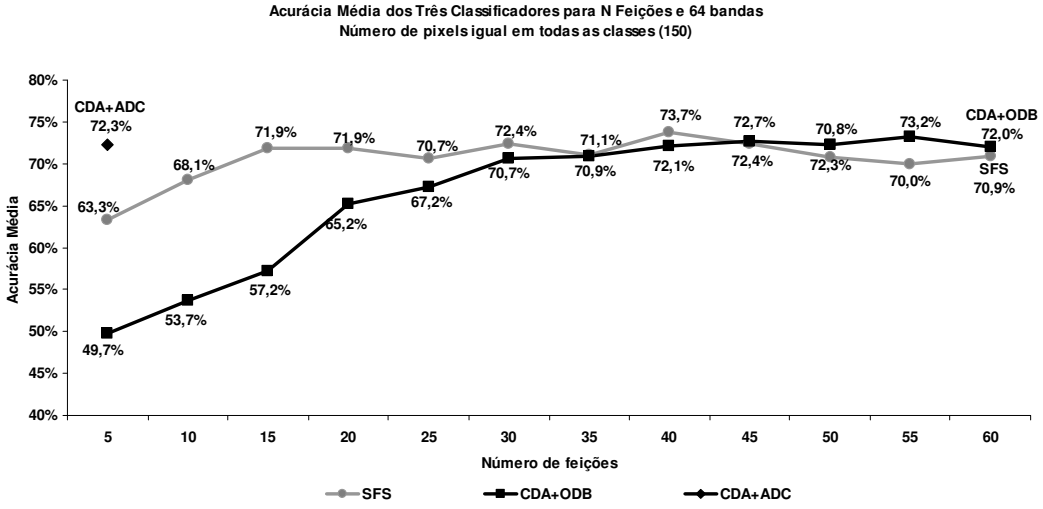


Figura 4.18 Acurácia média dos três métodos com N feições e amostras de tamanho 150

Nos experimentos envolvendo amostras de treinamento com 150 *pixels*, nota-se que o classificador CDA/ODB apresentou uma acurácia média baixa, com um crescimento lento na medida em que a dimensão dos dados aumenta, vindo a igualar-se ao SFS quando utilizadas 35 feições. Novamente aqui, se percebe que a técnica CDA/ODB é mais robusta com relação ao efeito de Hughes. Observa-se também que o método CDA/ADC apresentou acurácia média muito superior ao produzido pelo outros dois classificadores, mesmo com o uso de poucas feições.

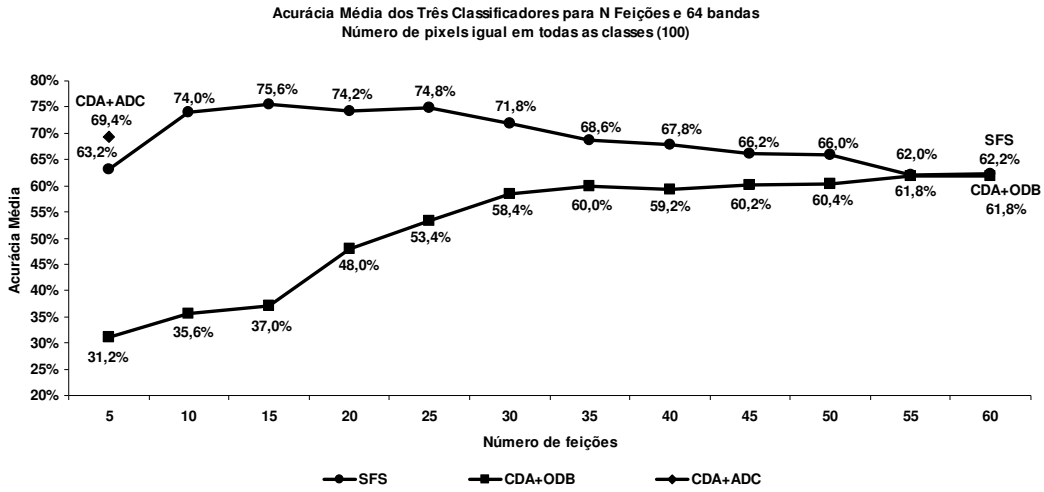


Figura 4.19 Acurácia média dos três métodos com N feições e amostras de tamanho 100

A Figura 4.19 exibe os resultados com amostras de treinamento de tamanho igual a 100. Analisando esses resultados, observou-se que a metodologia proposta é sensível a um número muito pequeno de amostras de treinamento. Conjuntos de amostras de treinamento iguais a 100 *pixels* por classe produziram para o classificador CDA/ODB uma acurácia média inferior àquela produzida pelo classificador em estágio único com a técnica SFS, se igualando somente na dimensão máxima (60 feições). Neste experimento, o classificador CDA/ADC exibiu novamente valores mais altos de acurácia média com o uso de poucas feições do que aqueles obtidos através dos outros dois processos.

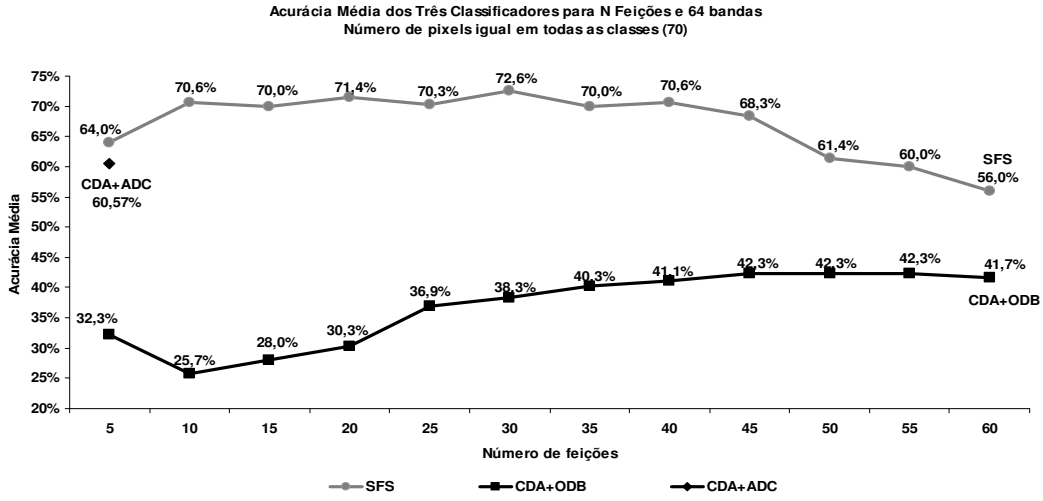


Figura 4.20 Acurácia média dos três métodos com N feições e amostras de tamanho 70

Os experimentos com amostras de treinamento de tamanho igual a 70 (Figura 4.20), evidenciaram que o classificador CDA/ODB é muito sensível a conjuntos de treinamento com número de amostras pequeno, isto é, um número de amostras próximo ao número de parâmetros estimados. Esse fato se deve ao processo de otimização da distância de Bhattachayya escolhido, no qual é priorizada a contribuição das diferenças entre as matrizes de covariância determinada pela equação (30) do capítulo 3. Enquanto isso, na ADC, a estimação das matrizes de covariância tem uma importância menor, pois a mesma prioriza as diferenças entre os vetores de médias e utiliza uma estimação ponderada das matrizes de covariância, isto é, a mesma matriz de covariância para as duas classes.

Tabela 4.3 Síntese comparativa dos três métodos em relação ao número de feições utilizadas e amostras de treinamento

Nº de Feições	Amostras de treinamento				
	500	300	150	100	70
5	CDA/ADC	CDA/ADC	CDA/ADC	CDA/ADC	SFS
10	CDA/ODB	CDA/ADC	CDA/ADC	SFS	SFS
15	CDA/ODB	SFS	CDA/ADC	SFS	SFS
20	CDA/ODB	SFS	CDA/ADC	SFS	SFS
25	CDA/ODB	SFS	CDA/ADC	SFS	SFS
30	CDA/ODB	SFS	SFS	SFS	SFS
35	CDA/ODB	SFS	CDA/ADC	CDA/ADC	SFS
40	CDA/ODB	SFS	SFS	CDA/ADC	SFS
45	CDA/ODB	SFS	SFS	CDA/ADC	SFS
50	CDA/ODB	CDA/ODB	CDA/ADC	CDA/ADC	SFS
55	CDA/ODB	CDA/ODB	CDA/ODB	CDA/ADC	CDA/ADC
60	CDA/ODB	CDA/ODB	CDA/ADC	CDA/ADC	CDA/ADC

A Tabela 4.3 exibe uma síntese comparativa da qualidade dos três métodos implementados em relação ao número de feições utilizadas no processo de classificação e a quantidade de amostras de treinamento disponíveis. Nota-se que, se forem utilizadas até 5 feições, o método CDA/ADC é mais vantajoso na maior parte dos casos, com exceção ao uso de 70 amostras. Esse método também é praticamente igual ao SFS quando são utilizadas menos que 150 amostras de treinamento (cerca de duas vezes e meia a quantidade de parâmetros estimada) e superior ao SFS a partir de 35 feições quando são utilizadas menos que 100 amostras para treinar o classificador devido ao efeito do fenômeno de Hughes sobre o método SFS. Enquanto isso, observa-se que a metodologia CDA/ODB é a mais eficiente dentre todas quando se dispõe de um número

razoável de amostras de treinamento, ou então quando é utilizada uma quantidade grande de feições.

4.8 Tempo de processamento em função da dimensão dos dados

Quanto ao tempo de processamento, foi observado que o classificador CDA é muito mais penalizado do que o classificador em estágio único. Essas conclusões podem ser verificadas através da Figura 4.5, a qual ilustra o tempo de processamento para o caso de um conjunto de treinamento com 500 amostras. Enquanto o SFS seleciona *N* feições no processo global de classificação, independentemente do número de classes envolvidas, a estrutura de classificação em árvore exige o uso efetivo de um número de feições superior a *N* em todo o processo de classificação, conforme o que será demonstrado a seguir.

Pode-se observar pela análise da Figura 4.21, que o classificador que requer maior tempo de processamento para qualquer dimensão dos dados é o CDA/ODB. Isso é principalmente devido à necessidade de procura das duas classes mais distantes entre si e de extrair as feições para esse par de classes a cada nó. Em termos reais, o número total de feições extraídas ao longo de todo o CDA varia de acordo com o número de classes envolvidas e também com a dimensão *N* escolhida em cada nó. Sendo adotado um número de feições igual em cada nó, o total de feições extraídas pelo método CDA/ODB é igual a:

$$TF_{CDA/ODB} = \frac{2^C}{2} - 1 \cdot N$$

Onde,

$TF_{CDA/ODB}$: total de feições extraídas através do método CDA/ODB;

C: número de classes;

N: dimensão em cada nó;

A Tabela 4.4 ilustra alguns exemplos da quantidade total de feições extraídas pelo método CDA/ODB para *C* classes e *N* feições. Nota-se através desse exemplo que com *N* igual a 60 feições, num problema com 5 classes, o processo todo estará selecionando, na verdade, 1800 feições.

Tabela 4.4 Total de feições extraídas pelo método CDA/ODB em função da dimensão e do número de classes

	Dimensão			
Nº de Classes	**5**	**30**	**60**	**130**
2	5	30	60	130
3	15	90	360	390
4	35	210	840	910
5	75	450	1800	1950
6	155	930	3720	4030

Enquanto isso, como a dimensão máxima utilizada pela ADC é limitada primeiramente pelo número de classes, o número total de feições extraídas ao longo do CDA é dado por:

$$TF_{CDA/ADC} = \sum_{i=2}^{C} (C - i + 1)$$

Onde,

$TF_{CDA/ADC}$: total de feições extraídas pelo método CDA/ADC;

C: número de classes.

Nesse caso, o total de feições selecionadas através do método CDA/ADC, é dado apenas em função do número de classes envolvidas, conforme exemplificado na Tabela 4.5. O tempo de processamento do método CDA/ADC, utilizando inicialmente 4 feições, foi semelhante ao tempo tomado pelo método CDA/ODB, quando utilizadas 5 feições por nó. Deve-se notar que esse tempo é ligeiramente inferior, pois o número total de feições selecionadas no método CDA/ADC com cinco classes é na verdade 26, enquanto o CDA/ODB, com cinco classes e dimensão igual a cinco feições por nó, seleciona 75 feições ao todo (Tabela 4.4).

O tempo de processamento utilizado pelo método SFS foi inferior, pois a cada incremento no número de feições, todas as feições selecionadas anteriormente permanecem no conjunto, não sendo assim necessário re-processamento, mas apenas testar qual dentre as bandas restantes mais contribui para a classificação. Como o conjunto de bandas disponíveis a selecionar é cada vez mais reduzido, o tempo necessário para a seleção de uma nova feição também é menor.

Tabela 4.5 Total de feições extraídas pelo método CDA/ADC em função do número de classes

Nº de Classes	Total de feições
2	1
3	4
4	11
5	26
6	57

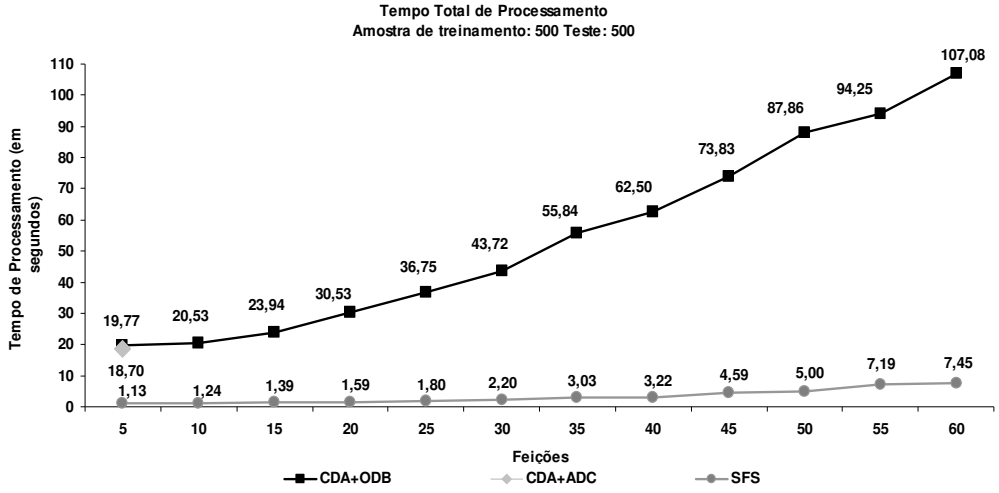


Figura 4.21 Tempo de processamento dos três métodos com 5 classes e amostras de tamanho 500

Como exemplo, observa-se que para uma dimensão de classificação igual a 30, o método SFS seleciona apenas 30 feições, enquanto o método do CDA/ODB seleciona ao todo 450 feições. Devido às suas restrições, o método CDA/ADC não pode ser comparado diretamente com os demais, pois novamente nesse caso com cinco classes, seleciona no máximo 26 feições.

O método CDA/ODB proposto apresentou então, características superiores e inferiores em relação à combinação CDA/ADC e também ao classificador tradicional em estágio único combinado com o método SFS. Inicialmente, observou-se que o conjunto CDA/ODB demandou maior tempo de processamento e apresentou acurácia final de classificação inferior às demais metodologias testadas em casos críticos, isto é, frente a tamanhos cada vez mais limitados de amostras.

No entanto, os experimentos demonstraram também que, a partir de conjuntos de amostras oito vezes maiores que o número de bandas originais (500 amostras de treinamento e 64 bandas), o método proposto é consideravelmente mais

eficiente que os métodos tradicionais. Outra característica observada é que, mesmo para tamanhos amostrais críticos, o método é robusto frente aos efeitos do fenômeno de Hughes, apresentando uma acurácia final de classificação crescente com o aumento da dimensão utilizada.

Além disso, nota-se que a implementação de métodos para extração de feições em conjunto com o CDA contribui para um ganho efetivo na acurácia final de classificação. Embora a metodologia da ADC não tenha sido testada em um classificador de estágio único, a aplicação da mesma em conjunto com o CDA, sugere que é possível adaptar qualquer outro procedimento conhecido para extração/seleção de feições em um classificador hierárquico, obtendo resultados superiores aos obtidos através do método de classificação em estágio único. Assim, sendo que a maior parte do trabalho desenvolvido consistiu na elaboração do algoritmo para a construção do CDA que produzisse resultados confiáveis, o CDA por si só torna-se um dos resultados mais importantes dessa dissertação, dadas suas potenciais aplicações.

5 CONCLUSÃO

Nesta dissertação é investigada a eficiência de um processo de extração de feições com base no critério de otimização da distância de Bhattacharyya (ODB), para fins de classificação de dados imagem em dimensão muito alta (dados hiper-espectrais). Como a distância de Bhattacharyya encontra-se definida para apenas duas classes de cada vez, utilizou-se aqui um classificador de decisão em árvore (CDA) em estrutura binária. Dessa forma, em cada nó é considerado um único par de classes a cada vez, o que permite a utilização do critério acima. Para fins de comparação, o processo de classificação empregando a estrutura em árvore binária foi repetido, empregando-se também o método tradicional de extração de feições para fins de classificação, conhecido como Analise de Discriminante Canônica (ADC). A metodologia proposta nesta dissertação para extração de feições é também comparada com um método tradicional de classificação em estágio único, implementando o conceito de seleção de feições (*Sequential Forward Selection* – SFS).

Na seção dos experimentos, é utilizada uma imagem do sensor hiper-espectral AVIRIS, o qual disponibiliza 190 bandas espectrais (variáveis) cobrindo em detalhes o espectro eletromagnético. A cena utilizada contém seis tipos de culturas agrícolas, as quais correspondem a seis classes com alta semelhança espectral, sendo assim um grande desafio ao classificador. Com a totalidade das bandas disponíveis e essas seis classes, é obtido um dos principais resultados através do estudo do algoritmo para a construção e definição do classificador em estrutura binária. Esse estudo revela, com a análise de um parâmetro chamado de Limiar de Verossimilhança (LV), que a acurácia de classificação em função da dimensão utilizada torna-se estável somente quando o CDA possui o número máximo de nós terminais. É verificado que o CDA com um número reduzido de nós terminais produz resultados de classificação pouco previsíveis e inferiores aos obtidos com o LV no seu nível máximo. Dessa forma, é definida a forma do CDA adotada nos experimentos finais.

Com o propósito de aumentar a precisão das estimativas dos parâmetros, especialmente na matriz de covariância das classes, são tomadas duas medidas. A primeira consiste na eliminação de áreas que apresentam excesso de solo exposto, que aqui funcionou como ruído, restando assim cinco classes com número de amostras suficientes para comparação. Devido à alta correlação entre as bandas adjacentes, a

segunda medida é a redução sistemática das mesmas para um terço das originais, restando 64 bandas.

Com o propósito de fixar possíveis fatores de variação externos ao processo de classificação, a quantidade de amostras utilizadas para treinar o classificador é fixada igualmente em 500, 300, 150, 100 e 70 para todas as classes. Esse procedimento permite também verificar a evolução gradual do comportamento da acurácia de classificação entre todas as metodologias testadas. A análise dos resultados finais revela que a metodologia proposta (CDA/ODB) é mais sensível quando o número de amostras de treinamento se aproxima ao número de parâmetros estimados nas matrizes de covariâncias. Entretanto, esse método mostra-se mais robusto em relação aos efeitos do fenômeno de Hughes, apresentando um aumento gradual na acurácia de classificação com o aumento da dimensão utilizada. Enquanto isso, apesar do método SFS apresentar valores mais elevados na acurácia final de classificação, o mesmo também evidencia claramente os efeitos do fenômeno de Hughes. Quanto ao método CDA/ADC, o mesmo apresenta resultados medianos, com uma acurácia final de classificação relativamente alta frente a tamanhos amostrais críticos, dadas suas limitações quanto ao número máximo de feições possíveis para extração.

Destaca-se ainda que, conforme trabalhos realizados anteriormente (Tadjudin, 1998), o caso especial da otimização da distância de Bhattacharyya para diferenças predominantes entre os vetores de médias apresenta resultados superiores aos obtidos através do método proposto nessa dissertação (diferença predominante entre as matrizes de covariância). Todavia, os resultados aqui obtidos são inéditos e impossíveis de serem previstos sem a experimentação conduzida.

Outrossim, é indicado que futuros trabalhos referentes à redução de dimensões em dados com um grande número de variáveis sejam aplicados em conjunto com a metodologia do CDA proposto, pois o mesmo além de ser um possível mitigador do fenômeno de Hughes, sugere que existe um aumento na acurácia de classificação obtida através dos métodos tradicionais para extração de feições em classificadores de estágio único.

6 BIBLIOGRAFIA

Aeberhard, S. and Coomans, D. e De Vel, O., **"Comparative Analysis of Statistical Pattern Recognition Methods in High Dimensional Settings"**, Pattern Recognition, Vol. 27, nº 8, pp. 1065-1077, 1994.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J., **Classification and Regression Trees (CART)**. Belmont, CA: Wadsworth Int., 1984.

Choi, E. and Lee, C., "**Feature extraction based on the Bhattacharrya distance",** Patt. Recog. vol. 36, pp. 1703-170, 2002.

Fukunaga, K., **Introduction to Statistical Pattern Recognition**. 2nd. Ed. Boston: Academic Press, 1990.

Gelfand, S. B., Ravishankar, C. S., and Delp, E. J., **"An iterative growing and pruning algorithm for classification tree design"**, IEEE Trans. Patt. Anal. Mach. Intell., pp. 163-174, 1991.

Gonzalez, R. C. and Woods, R. E., **Digital Image Processing**, Addison-Wesley,1993.

Hughes, G. F., **"On the mean accuracy of statistical pattern recognizers"**, IEEE Trans. Inform. Theory, vol. IT-14, pp. 55-63, 1968.

Kendall, M. G., **A Course in the Geometry of n-Dimensions**, Hafner Publishing, 1961.

Kim, B. and Landgrebe, D. A., **"Hierarquical decision tree classifiers in high-dimensional and large class data**", Ph.D. dissertation and Tech. Rep. TR-EE-90-47, School of Elec. Eng. Purdue Univ., W. Lafayette, IN, 1990.

Landgrebe, D. A., **Signal Theory Methods in Multispectral Remote Sensing**, Wiley Interscience, 2003.

Lee, C. and Landgrebe, D. A., **"Analyzing high dimensional multispectral data"**, IEEE Transactions on Geoscience and Remote Sensing, Vol. 31, No. 4, pp. 792-800, July, 1993.

Richards, J. A. and Jia, X., **Remote Sensing Digital Image Analysis: an introduction**, 3rd. ed. Springer, 1998.

Safavian, S. R. and Landgrebe, D. A., **"A Survey of Decision Tree Methodology"**, IEEE Trans. Systems, Man and Cybernetics, vol. 21, no. 3, May/Jun 1991.

Scharf, L.L., **Statistical Signal Processing. Detection, Estimation, and Time Series Analysis**, Addison-Wesley, 1991.

Scott, D. W., **Multivariate Density Estimation**. John Wiley & Sons, 1992.

Serpico, S. B., Roli, Bruzzone, **"Extension of the Jeffreys-Matusita distance to multiclass cases for feature selection"**, IEEE Transactions on Geoscience and Remote Sensing, v 33, n 6, p 1318-1321, Nov. 1995.

Shahshahani, B. M. and Landgrebe D. A., **"The Effect of Unlabeled Samples in Reducing the Small Sample Size Problem and Mitigating the Hughes**

Phenomenon", IEEE Transactions on Geoscience and Remote Sensing, vol. 32, nº 5, 1087-1095, sept. 1994.

Tadjudin, S. and Landgrebe, D. A., **"Classification of high-dimensional data with limited training samples"**, Ph.D. dissertation and Tech. Rep. TR-ECE-98-8, School of Elec. Eng. Purdue Univ., W. Lafayette, IN, 1998.

Therrien, C. W., **Decision Estimation and Classificaton, An Introduction to Pattern Recognition and Related Topics**. John Wiley & Sons, 1989.

Wegman, E. J. **"Hyperdimensional data analysis using parallel coordinates"**, Journal of the Statistical Association, Vol. 85, Nº 411, pp. 664-675, 1990.

You, K. C. and Fu, K. S., **"An approach to the design of a linear binary tree classifier"**, in 3rd. Symp. Machine Processing of Remotely Sensed Data, Purdue Univ., W. Lafayette, IN, 1976.

ANEXOS

Algoritmos criados em MATLAB para execução dos experimentos.

```
function w = treestats500(varargin)
% Calculo das estatisticas das classes.
% Run:
%
treestats500(train_soy_notill_190,train_soy_min_190,train_soy_190,train_corn_notill_190,train_corn_mi
n_190);

Inicio = clock; % Tempo de Inicio.

for k = 1:nargin
   w(k).sample = varargin{k};     % Referência automática das amostras (este método duplica o espaço
utilizado).
   w(k).filename = inputname(k);  % Nomes dos arquivos das classes.
   w(k).mean = mean(w(k).sample); % Vetor medias.
   w(k).cov = cov(w(k).sample);   % Matriz de covariancias.
end

% Nomes das classes.
w(1).name = 'Soja sem cultivo';
w(2).name = 'Soja cultivo minimo';
w(3).name = 'Soja cultivo limpo';
w(4).name = 'Milho sem cultivo';
w(5).name = 'Milho cultivo minimo';
% w(6).name = 'Milho cultivo limpo';

save w w;

% Gráfico comparativo dos vetores média das classes.
x = [1:65];
plot(x,w(1).mean,x,w(2).mean,x,w(3).mean,x,w(4).mean,x,w(5).mean)
legend(w(1).name,w(2).name,w(3).name,w(4).name,w(5).name)
xlabel('Banda Espectral');
ylabel('Reflectancia Espectral Media');

% Tempo total de processamento.

Fim = clock;
etime(Fim,Inicio)
```

```
% Criação do arquivo com a verdade terrestre.

img_GT = [];
for i = 1:500
   img_GT = [img_GT;1];
end
for i = 1:500
   img_GT = [img_GT;2];
end
for i = 1:500
```

```
   img_GT = [img_GT;3];
end
for i = 1:500
   img_GT = [img_GT;4];
end
for i = 1:500
   img_GT = [img_GT;5];
end

save img_GT img_GT;
load img_GT img_GT;
```

```
function d = treebuild(dist,nfeat,T,w)
% TREEBUILD gera uma arvore de decisao binaria para
% classificaçao de imagens hiperespectrais.
%
%   Sintaxe:
%   TREEBUILD(DIST,NFEAT,T,W)
%
%   DIST e a funçao discriminante entre classes, podendo
%   ser: 'eucl' ou 'Euclidiana' e 'batt' ou 'Battacharyya'.
%   NFEAT e´ o numero de feiçoes extraidas, NFEAT < numero de bandas.
%   T e´ um limiar (threshold) usado para limitar o percentual
%   necessario de pixels para atribuir um pixel a uma determinada classe.
%   W e uma variavel estruturada contendo todas as estatisticas das
%   classes.
%

Inicio = clock;
nclasses = size(w,2);
par.maxdist = [];

% Inicio da arvore.
for node = 1:31
  if node == 1
     tree(node).nodes = [1:nclasses];
  end
  par.max = [];

  % Distancias entre pares de classes .
  if (size(tree(node).nodes,2) > 1)
     for i = tree(node).nodes
       d.par = [];
       for j = tree(node).nodes
          if i == j
             d.par(j) = NaN;                    % Evita que a distancia minima seja zero.
          else
             d.par(j) = treedist(dist,i,j,w); % Distancia entre todos os pares, i ~= j.
          end
       end
       [val,pos] = max(d.par(:));                % Identifica qual par (i,j) possui
       par.max = [par.max; [val i pos]];          % a distancia maxima e salva na variavel estruturada 'par'.
     end
     [val,pos] = max(par.max(:,1));

     % Extração de feições.
     i = par.max(pos,2);
     j = par.max(pos,3);
```

```
    fprintf(1,'Node%2d: %1d %1d %1d %1d %1d %1d\t',node,tree(node).nodes);
    fprintf(1,'Next pair: i = %1d   j = %1d\t\r',[i j]);
    par.maxdist = [par.maxdist; par.max(pos,2:3)];                % Maior distancia entre classes.

        % Covariance dominant difference.
                batt(node).matrix = inv(w(j).cov) *  w(i).cov;
                [V,L] = eig(batt(node).matrix);
                batt(node).eigenvectors = V;
                batt(node).eigenvalues  = L;

                % Compute each term of BD.
                batt(node).b = [];
                for n = 1:size(batt(node).eigenvectors,1)
       db = (1/4 * ((batt(node).eigenvectors(:,n)' * (w(j).mean - w(i).mean)')^2/(1 +
batt(node).eigenvalues(n,n)))) +...
            (1/4 * (log(batt(node).eigenvalues(n,n) + inv(batt(node).eigenvalues(n,n)) + 2 ) - log(4)));
%           db = log(batt(node).eigenvalues(n,n) + inv(batt(node).eigenvalues(n,n)) + 2);  % APENAS
CONTRIBUICAO DE COVARIANCIAS.

        batt(node).b = [batt(node).b; db];
                end

                % Selecting the highests eigenvectors that maximizes the BD
                batt(node).b_temp = batt(node).b;
                batt(node).evectors = [];
                for k = 1:nfeat
        [val,pos] = max(batt(node).b_temp);
        batt(node).bposicao(k) = pos;
        batt(node).b_temp(pos) = NaN;
        batt(node).evectors = [batt(node).evectors batt(node).eigenvectors(:,pos)];
                end

     % The transformed features and statistics of the node.
    for classe = tree(node).nodes
       y(classe).y(node).sample = w(classe).sample * batt(node).evectors;
       y(classe).y(node).mean = mean(y(classe).y(node).sample);
       y(classe).y(node).cov = cov(y(classe).y(node).sample);
       y(classe).y(node).T = - ((chi2inv(0.95,size(batt(1).evectors,2)))/2) - (1/2) *
log(det(y(classe).y(node).cov)) +...
                        log(1/size(y,2));
    end

    % Classificaçao pela Maxima Verossimilhança.
    for k = tree(node).nodes
       w(k).class_maxver = []; % Incializa a matriz que recebe a classificação dos pixeis.
       for pixel = 1:size(y(k).y(node).sample,1) % Executa a classificação da imagem.
          max_ver = [];
          for classe = [par.maxdist(node,:)]
             mv = maxima_verossimilhanca(k,pixel,classe,y,node);
             max_ver = [max_ver;mv];
          end
          [valor,pos] = max(max_ver);
          if pos == 1
             w(k).class_maxver = [w(k).class_maxver;par.maxdist(node,1)];
          else
             w(k).class_maxver = [w(k).class_maxver;par.maxdist(node,2)];
          end
       end
    end
```

```
      % Classificaçao geral.
      tree(node*2).nodes = [];
      tree(node*2+1).nodes = [];
      for k = tree(node).nodes
         fprintf(1,'Class%2d \t\r',k);
         tabulate(w(k).class_maxver(:))
         x= tabulate(w(k).class_maxver(:));
         if (x(i,3) > T) | (k == i)
            tree(node*2).nodes = [tree(node*2).nodes k];
         elseif x(j,3) > T | (k == j)
            tree(node*2+1).nodes = [tree(node*2+1).nodes k];
         else
            tree(node*2).nodes = [tree(node*2).nodes k];
            tree(node*2+1).nodes = [tree(node*2+1).nodes k];
         end
      end
      fprintf(1,'Node%2d: %1d %1d %1d %1d %1d %1d\t',node*2,tree(node*2).nodes);
      disp('+++...');
      fprintf(1,'Node%2d: %1d %1d %1d %1d %1d %1d\t',node*2+1,tree(node*2+1).nodes);
      disp('...+++');

      % Terminal nodes.
   else
      par.maxdist = [par.maxdist; tree(node).nodes tree(node).nodes];
      tree(node*2).nodes = [tree(node).nodes];
      tree(node*2+1).nodes = [tree(node).nodes];
      fprintf(1,'Node%2d: %1d %1d %1d %1d %1d %1d\t \n',node,tree(node).nodes);
      disp('=== Terminal node ===');
   end
   save tree tree;
end

% batt(node) = batt(node)extract2(nfeat,par,w);

save y y;
save batt batt;
save tree tree;
save w w;
save d d;
save par par;
Fim = clock;
TempoDeProcessamento = etime(Fim,Inicio)
```

```
function y = treeclassifier(tree,y,par,batt,img)
% Classifica um pixel em determinada classe i ou j
% ao longo de uma arvore binaria ate os nos terminais.
% Classificadores: Maxima Verossimilhança ('maxver') ou Mahalanobis ('maha').
% treeclassifier(tree,y,par,batt,img);

Inicio = clock;
img_class = [];
[lin,col] = size(par.maxdist(:,1));

for pixel = 1:size(img,1)
   node = 1;
   while (node <= lin)
%       node
      max_ver = [];   % Initialize maxver vector (witch contains two values).
      i = par.maxdist(node,1);
      j = par.maxdist(node,2);
      if i == j  % Terminal node.
         img_class = [img_class;i];
         node = lin+1;   % Terminate loop.
      else
         pixel_reduced = img(pixel,:) * batt(node).evectors;   % Compute pixel in the new coordinates.

         for classe = par.maxdist(node,:)
            % The transformed features: and statistics of the node.
            mv = maxima_vero(pixel_reduced,classe,y,node);         % Compute the pixel maxver.
            max_ver = [max_ver mv];
         end
         [val,pos] = max(max_ver);
         parentnode = node;
         if (pos == 1) % Go to left node.
            node = node*2;
%                if ((size(tree(node).nodes,2)) == 2) & (val > y(par.maxdist(node,1)).y(node).T)
%                   img_class = [img_class;7];
%                   node = lin+1;
%                end
         else % Go to right node.
            node = node*2+1;
%                if ((size(tree(node).nodes,2)) == 2) & (val > y(par.maxdist(node,2)).y(node).T)
%                   img_class = [img_class;7];
%                   node = lin+1;
%                end
         end
      end
   end
end

save img_class img_class;
Fim = clock;
TempoDeProcessamento = etime(Fim,Inicio)
```

```
function y = maxima_verossimilhanca(k,pixel,classe,y,node)
y = -log(det((y(classe).y(node).cov))) - (y(k).y(node).sample(pixel,:) - ...
    y(classe).y(node).mean) * inv(y(classe).y(node).cov) * (y(k).y(node).sample(pixel,:) - ...
   y(classe).y(node).mean)' + 2 * log(1/size(y,2));
```

```
% Script de automatização de todo o processo de construção do CDA
% com as amostras de treinamento e utilização do CDA para classificação das amostras de teste.

loaddata500;
treestats500(train_corn_notill_500,train_corn_min_500,train_corn_500,train_soy_notill_500,train_soy_m
in_500);
clear;
load w;
load img;
makegt500;
load img_GT;

for x = 5:5:60
   x
     for z = [100]
       Inicio = clock;
       z
       treebuild('batt',x,z,w);
       Fim = clock;
       Tempobatbuild500(x,z) = etime(Fim,Inicio); % Elapsed time to treebuild('batt',x,y,w).
       save Tempobatbuild500 Tempobatbuild500;
       load par;
       load tree;
       trees100_5a15_500(x).trees(z).tree = tree;   % Tree structure.
       load y;
       load batt;
       Inicio = clock;
       treeclassifier(tree,y,par,batt,img);
       load img_class;
       trees100_5a15_500(x).trees(z).crosstab = crosstab(img_GT,img_class);  % Confusion Matrix.
       trees100_5a15_500(x).trees(z).crosstab
       Fim = clock;
       Tempobatclass500(x,z) = etime(Fim,Inicio); % Elapsed time to treeclassifier(tree,y,par,batt,img).
       save Tempobatclass500 Tempobatclass500;
       Tempobattotal500(x,z) = Tempobatbuild500(x,z) + Tempobatclass500(x,z);
       save Tempobattotal500 Tempobattotal500;
   end
end
save img_class img_class;
save trees100_5a15_500 trees100_5a15_500;
```

```
function d = treebuildcan_nfeat(dist,nfeat,T,w)
%   Adaptaçao do CDA para o metodo da Analise Discriminante Canonica

Inicio = clock;
nclasses = size(w,2);
par.maxdist = [];


% Inicio da arvore.
tree(1).nodes = [1:nclasses];
% par.maxdist = s(63,2);
for node = 1:63
  par.max = zeros(6,3);

  % Distancias entre pares de classes .
  if (size(tree(node).nodes,2) > 1)
     for i = tree(node).nodes
       d.par = [];
       for j = tree(node).nodes
          if i == j
             d.par(j) = NaN;                    % Evita que a distancia minima seja zero.
          else
             d.par(j) = treedist(dist,i,j,w); % Distancia entre todos os pares, i ~= j.
          end
       end
       [val,pos] = max(d.par(:));                % Identifica qual par (i,j) possui
       par.max = [par.max; [val i pos]];          % a distancia maxima e salva na variavel estruturada 'par'.
     end
     [val,pos] = max(par.max(:,1));

     % Extração de feições.
     i = par.max(pos,2);
     j = par.max(pos,3);
     fprintf(1,'Node%2d: %1d %1d %1d %1d %1d %1d\t',node,tree(node).nodes);
     fprintf(1,'Next pair: i = %1d   j = %1d\t\r',[i j]);
%      disp(' ');
     par.maxdist = [par.maxdist; par.max(pos,2:3)];    % Maior distancia entre classes.

     % Analise Discriminante Canonica.
     can(node).cov_w = zeros(size(w(1).cov,2));
     can(node).mean_zero = zeros(1,size(w(1).cov,2));
     for k = tree(node).nodes
        can(node).cov_w = can(node).cov_w + ((1/size(k,2)) * (w(k).cov));
        can(node).mean_zero = can(node).mean_zero + (1/size(tree(node).nodes,2)) * (w(k).mean);
     end
     can(node).cov_b = zeros(size(w(1).cov,2));
     for k = tree(node).nodes
        can(node).cov_b = can(node).cov_b + ...
                  (1/size(tree(node).nodes,2)) * ((w(k).mean - can(node).mean_zero)' * (w(k).mean -
can(node).mean_zero));
     end
     can(node).matrix = pinv(can(node).cov_w) *  can(node).cov_b;  % Discriminant Matrix.
           [V,D] = eig(can(node).matrix);
           can(node).e = V;         % Eigenvectors.
           can(node).d  = diag(D)'; % Eigenvalues.

     % Selecting the highests eigenvectors that maximizes the CA
     [values,index] = sort(can(node).d);
     nfeat = size(tree(node).nodes,2) - 1;
     can(node).eordered = zeros(size(w(1).cov,1),nfeat);
```

```
    for k = 1:nfeat
       can(node).eordered(:,k) = can(node).e(:,index(1,size(index,2)+1-k));
    end

    % The transformed features and statistics of the node.
    for classe = tree(node).nodes
       y(classe).y(node).sample = w(classe).sample * can(node).eordered;
       y(classe).y(node).mean = mean(y(classe).y(node).sample);
       y(classe).y(node).cov = cov(y(classe).y(node).sample);
       y(classe).y(node).T = - ((chi2inv(0.95,size(can(node).eordered,2)))/2) - (1/2) *
log(det(y(classe).y(node).cov)) + log(1/size(y,2));
    end

    % Classificaçao pela Maxima Verossimilhança.
    for k = tree(node).nodes
       w(k).class_maxver = []; % Incializa a matriz que recebe a classificação dos pixeis.
       for pixel = 1:size(y(k).y(node).sample,1) % Executa a classificação de cada pixel da imagem.
          max_ver = [];
          for classe = par.maxdist(node,:)    % Classifica o pixel em uma das duas classes mais distantes
entre si.
             mv = maxima_verossimilhanca(k,pixel,classe,y,node);  % Calcula a verossimilhança do pixel
`as duas classes.
             max_ver = [max_ver;mv]; % Armazena os dois valores de verossimilhança.
          end
          [valor,pos] = max(max_ver);
          if pos == 1
             w(k).class_maxver = [w(k).class_maxver;par.maxdist(node,1)];
          else
             w(k).class_maxver = [w(k).class_maxver;par.maxdist(node,2)];
          end
       end
    end

    % Classificaçao geral.
    tree(node*2).nodes = [];
    tree(node*2+1).nodes = [];
    for k = tree(node).nodes
       fprintf(1,'Class%2d \t\r',k);
       tabulate(w(k).class_maxver(:))
       x= tabulate(w(k).class_maxver(:));
       if (x(i,3) > T) | (k == i)
          tree(node*2).nodes = [tree(node*2).nodes k];
       elseif x(j,3) > T | (k == j)
          tree(node*2+1).nodes = [tree(node*2+1).nodes k];
       else
          tree(node*2).nodes = [tree(node*2).nodes k];
          tree(node*2+1).nodes = [tree(node*2+1).nodes k];
       end
    end
    fprintf(1,'Node%2d: %1d %1d %1d %1d %1d %1d\t',node*2,tree(node*2).nodes);
    disp('+++...');
    fprintf(1,'Node%2d: %1d %1d %1d %1d %1d %1d\t',node*2+1,tree(node*2+1).nodes);
    disp('...+++');

    % Terminal nodes.
  else
    par.maxdist = [par.maxdist; tree(node).nodes tree(node).nodes];
    tree(node*2).nodes = [tree(node).nodes];
    tree(node*2+1).nodes = [tree(node).nodes];
    fprintf(1,'Node%2d: %1d %1d %1d %1d %1d %1d\t \n',node,tree(node).nodes);
```

```
    disp('=== Terminal node ===');
  end
  save tree tree;
end

% batt(node) = batt(node)extract2(nfeat,par,w);

save y y;
save can can;
save tree tree;
save w w;
save d d;
save par par;
Fim = clock;
TempoDeProcessamento = etime(Fim,Inicio)
```

```
function d = sfs(nfeat,w)
% Sequential Forward Selection (SFS)
% Metodo para seleçao sequencial de feiçoes.
%
Inicio = clock;
classes = [1:size(w,2)];
d.bandas_disp = [1:size(w(1).sample,2);ones(1,size(w(1).sample,2))];
d.bandas_selecionadas = [];

for k = 1:nfeat                         % Repete o processo para quantas bandas forem desejadas.
   d.j_max = 0;                         % Zera o criterio maximo da ditancia de jefries-matusita.
          for i = 1:size(d.bandas_disp,2)       % Inicia a procura sequencial das bandas, da primeira a
ultima.
      d.bandas_temporarias = [d.bandas_selecionadas];
      d.j_ave = 0;                    % Zera o distancia de jefries-matusita para a banda em questao.
      if d.bandas_disp(2,i) == 1        % Utiliza somente se ela nao foi selecionada ainda.
        d.bandas_temporarias = [d.bandas_selecionadas d.bandas_disp(1,i)]      % Incorpora a banda
temporariamente ao conjunto.
        for c = classes               % Calcula as medias e covariancias com o conjunto temporario de bandas.
          w(c).mean = mean(w(c).sample(:,d.bandas_temporarias));
          w(c).cov = cov(w(c).sample(:,d.bandas_temporarias));
        end
        for c1 = 1:size(classes,2) - 1 % Calcula a distancia de bhattacharyya e j-m entre todos os pares de
classes.
          for c2 = c1+1:size(classes,2)
            d.bhatt(c1,c2) = (1/8) * (w(c1).mean - w(c2).mean) * inv((w(c1).cov + w(c2).cov)/2) *
(w(c1).mean - w(c2).mean)' + ...
                        (1/2) * (logdet((w(c1).cov + w(c2).cov)/2) - ((1/2) * (logdet(w(c1).cov) +
logdet(w(c2).cov))));
            d.j(c1,c2) = 2 * (1 - exp(-d.bhatt(c1,c2)));  % Distancia de jefries-matusita entre os pares de
classes.
            d.j_ave = d.j_ave + d.bhatt(c1,c2);  % Distancia de jefries-matusita media (na verdade eh
apenas a soma) para o problema multiclasses.
          end
        end
      end
      if d.j_ave > d.j_max        % Testa se a nova banda supera o valor da banda anterior.
        d.j_max = d.j_ave;       % Caso positivo, substitiu o valor maximo pelo resultante dessa banda.
        x_ast = i;              % Informa qual foi a banda que superou a anterior.
      end
   end                              % Salta para a proxima banda.
   d.bandas_selecionadas = [d.bandas_selecionadas x_ast];      % Incorpora definitivamente a banda que
maximizou o criterio da dist. j-m.
```

```
   d.bandas_disp(2,x_ast) = 0;                    % Informa que a banda ja foi selecionada.
end
save d d;
save w w;
Fim = clock;
d.TempoDeProcessamentoSFS = etime(Fim,Inicio)
```

```
function y = classifier(img,d,w)
% Classificador em estágio único para o algoritmo SFS.
Inicio = clock;
img_class = [];
classes = [1:size(w,2)];
for c = classes              % Calcula as medias e covariancias com o conjunto temporario de bandas.
   w(c).mean = mean(w(c).sample(:,d.bandas_selecionadas));
   w(c).cov = cov(w(c).sample(:,d.bandas_selecionadas));
end

for i = 1:size(img,1)
   max_ver = [];
   pixel = img(i,d.bandas_selecionadas);
   for classe = 1:size(w,2);
      mv = maxima_vero_sfs(pixel,classe,w);
      max_ver = [max_ver mv];
   end
   [val,pos] = max(max_ver);
   img_class = [img_class;pos];
end

save img_class img_class;
Fim = clock;
TempoDeProcessamento = etime(Fim,Inicio)
```