

20 e 21 de outubro
Instituto Nacional de Pesquisas Espaciais - INPE
São José dos Campos - SP

Árvore de Decisão Oblíqua

Carlos A. P. Castro Filho

Doutorando do Curso de Pós-Graduação em Computação Aplicada (CAp)
Instituto Nacional de Pesquisas Espaciais (INPE) – São José dos Campos – SP – Brazil
pires@dpi.inpe.br

Abstract. *The Oblique Decision Tree (ODT) classification method has the advantage of dividing feature spaces using multidimensional hyperplanes that are oblique to the Cartesian axes. This paper aims to present a model of an ODT. The ODT developed is binary and uses the Exchange Method for splitting each node into two subsets of classes, along with Fisher's Linear Discriminant to calculate the parameters for the hyperplanes. Tests on synthetic and remote sensing images have both showed better results than those obtained with traditional classifiers such as the Maximum Likelihood.*

Resumo. *O método de classificação de árvore de decisão oblíqua tem a vantagem de particionar espaços de atributos multidimensionais utilizando hiperplanos com orientação oblíqua aos eixos cartesianos. Este trabalho tem por objetivo apresentar uma modelagem de árvore de decisão oblíqua. A árvore de decisão desenvolvida é binária e utiliza o método da troca para dividir cada nó em dois subconjuntos de classes, juntamente com o discriminante linear de Fisher para cálculo dos parâmetros desses hiperplanos. Testes realizados sobre imagens sintéticas e de sensoriamento remoto apresentaram resultados superiores aos obtidos com classificadores tradicionais como, por exemplo, o de Máxima Verossimilhança.*

Palavras-chave: reconhecimento de padrões, classificação, árvore de decisão.

1. Introdução

Classificação é um método no qual um rótulo ou identificador de classe é associado a um conjunto de dados [Theodoridis e Koutroumbas 2006]. Esta associação ocorre em função das características do objeto a ser classificado. No caso de imagens digitais estas características podem ser atributos associados às respostas espectrais, textura, forma e contexto, dentre outros. Este processo de rotular pode ser implementado através de procedimentos de reconhecimento de padrões que, aplicados às imagens digitais, passa a se chamar de classificação [Pal e Mather 2001]. Um desses procedimentos de classificação é o de árvore de decisão.

De acordo com Quinlan (1993), este procedimento tem como vantagem o fato de possuir natureza e propriedades não-paramétricas, podendo classificar imagens com distribuições estatísticas diferentes da gaussiana, heterogêneas e possuidoras de ruídos. Além disso, podem também ser manipulados outros tipos de dados cujos valores são diferentes dos numéricos. Já, segundo Friedl e Brodley (1997), as árvores de decisão têm como vantagem o fato de serem simples e flexíveis, realizando testes sequenciais e de fácil compreensão cuja semântica é praticamente intuitiva.

Uma árvore de decisão é constituída internamente de:

- um nó inicial (chamado “raiz”) que contém todo o conjunto de dados a ser classificado;
- nós de decisão que dividem os dados através de testes, gerando os “galhos” que dão sequência à árvore; e
- nós terminais, isto é, “folhas” que representam as classes a serem extraídas no processo de classificação.

A Figura 1.1 ilustra o caso de uma árvore de decisão univariada, suas regras de classificação (ou testes) e a respectiva representação da classificação dos dados em um espaço de atributos. Este é o caso mais simples de árvore de decisão, sendo chamada de univariada em função de suas regras estarem associadas somente a um único atributo.

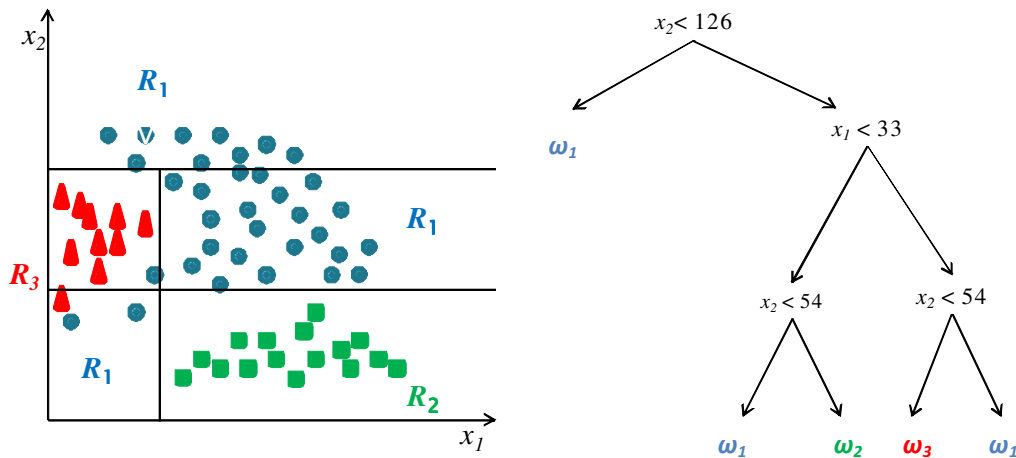


Figura 1.1 – Exemplo de árvore de decisão univariada.
Adaptado de Duda et al. (2001)

A regra geral da árvore de decisão univariada apresentada na Figura 1.1 é dada pela Equação 1.1. Neste tipo de árvore, cada reta que divide os dados é perpendicular ao respectivo eixo de atributo j , interceptando-o no valor ω_0 .

$$f(x/j, \omega_0) = x_j + \omega_0 \quad (1.1)$$

Já, na Figura 1.2 a árvore de decisão é chamada de oblíqua, ou multivariada linear [Yildiz e Alpaydin 2005], porque os testes são equivalentes a hiperplanos com orientação oblíqua aos eixos cartesianos, ou eixos dos atributos. Na respectiva representação da classificação no espaço dos atributos x_1 e x_2 observa-se na Figura 1.1 e 1.2 uma divisão das classes através de retas (ou hiperplanos, em casos de dimensões maiores) cujas equações são testes matemáticos. Caso um dado observado tenha um valor verdadeiro ao do teste ele será conduzido para um dos “galhos” da árvore, caso

contrário será conduzido para o outro. Ao término do processo, ao chegar em uma “folha”, aquele dado será associado à uma classe.

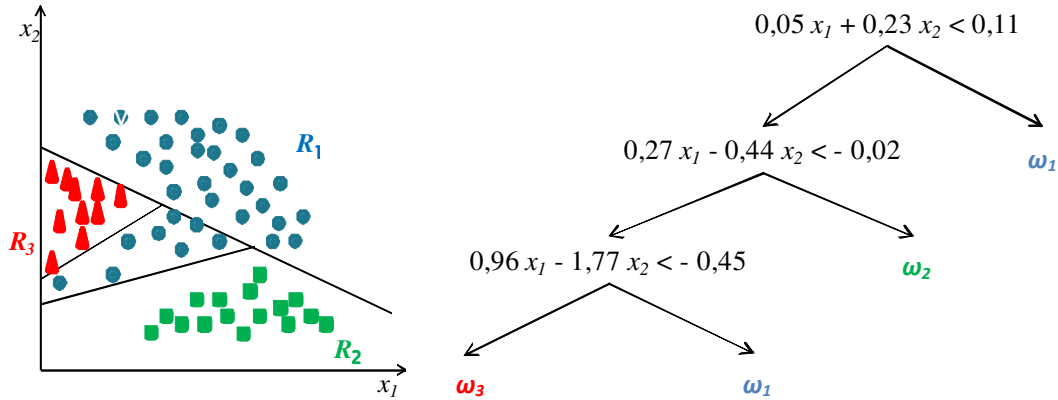


Figura 1.2 – Exemplos de árvores de decisão oblíqua.
Adaptado de Duda et al. (2001)

Tanto no caso da Figura 1.1 como nos da Figura 1.2 as árvores de decisão podem ser classificadas como binárias. No caso de árvores binárias, ou OBCT (*Ordinary Binary Classification Trees*) cada nó possui como saída somente dois galhos seguindo uma função booleana do tipo $f_m(x|\Phi) > 0$ com duas saídas possíveis, verdadeiro ou falso [Yildiz e Alpaydin 2005]. Esta função é parametrizada por Φ , com duas saídas, verdadeiro ou falso, referentes aos galhos da direita e da esquerda. Cada $f_m(x)$ define uma função discriminante e uma respectiva forma de modelo. No caso oblíquo o modelo é dado por:

$$f_m(x/\omega_m, \omega_{m0}) = \sum_{j=1}^d \omega_{mj} x_j + \omega_{m0} = \omega_m^T x + \omega_{m0} \quad (1.2)$$

sendo ω_m a direção do hiperplano divisor (d-dimensional) e ω_{m0} o valor limitante quando x está projetado sobre ω_m .

Este trabalho tem por objetivo apresentar uma modelagem e implementação de árvore de decisão oblíqua. Visando atingir o objetivo será utilizada a linguagem de programação IDL (*Interactive Data Language*).

2. O Programa

O programa é composto por diversos módulos, estando o algoritmo principal organizado conforme consta da Figura 2.1. Neste diagrama de blocos é possível observar as etapas de construção de uma árvore de decisão oblíqua, sendo cada um dos blocos possuidor de uma determinada cor:

- em verde estão os módulos que encontram-se discriminados separadamente;
- em azul estão as funções ou simples operações matemáticas;
- em amarelo estão as tomadas de decisão realizadas pelo próprio algoritmo através de um determinado critério; e
- em magenta estão as interfaces com o usuário.

O programa permite a entrada de imagens multi-canais, admitindo resolução radiométrica de até 8 bits por pixel. Logo após a entrada desses dados é necessário que o usuário realize a coleta (ou seleção) das amostras de treinamento, visto que o método de classificação a ser utilizado é supervisionado. Para maiores detalhes sobre tipos de resolução e de classificação sugere-se consultar Theodoridis e Koutroumbas, 2006.

O algoritmo principal foi decomposto em módulos, sendo os principais: módulo de construção da árvore de decisão; módulo de construção de nós de decisão; módulo de cálculo de ω e ω_0 ; módulo de classificação; e módulo de avaliação.

Por apresentarem peculiaridades em suas implementações, cada módulo, com exceção do módulo de avaliação, será detalhado a seguir. O módulo de avaliação segue o descrito por Congalton e Green (1999), onde é construída uma matriz de confusão buscando avaliar os dados da classificação.

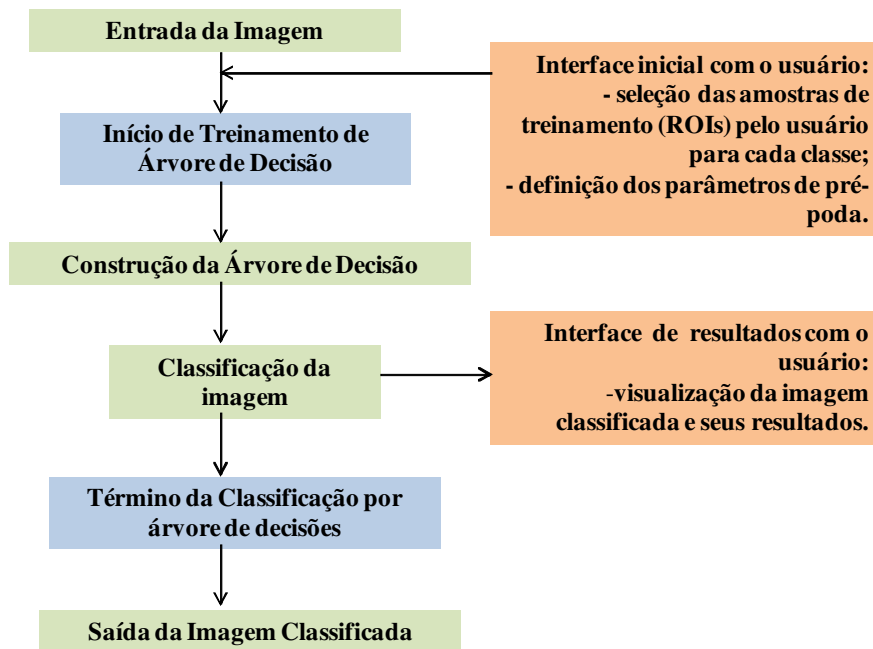


Figura 2.1 – Fluxograma do Módulo Principal

2.2 Módulos Principais

2.2.1 Módulo de construção da árvore de decisão

O módulo de construção da árvore de decisão encontra-se ilustrado na Figura 2.2. Nele, inicialmente são identificados os nós que são criados. A árvore de decisão desenvolvida possui característica de ser binária (OBCT), o que possibilita a identificação dos novos nós em função da identidade do nó que os gerou. De acordo com a Figura 2.3 observa-se que os nós são identificados com um número que depende do respectivo “nó mãe”, ou nó de origem.

Iniciando a construção da árvore de decisão pelo algoritmo desenvolvido, o número do primeiro nó será sempre 1. Consequentemente os próximos nós a serem gerados serão sempre os de número 2 e 3. Porém a divisão de um nó de origem

(qualquer que seja o número dele) em dois novos nós somente ocorrerá caso sejam atendidos 3 (três) critérios:

- caso o nó não seja puro, isto é, somente poderá ser dividido se naquele nó houver instâncias rotuladas a duas ou mais classes distintas;
- caso a porcentagem mínima de instâncias estiver dentro da tolerância definida pelo usuário; e
- caso a porcentagem mínima de diminuição de entropia (ou impureza) estiver dentro do valor definido pelo usuário.

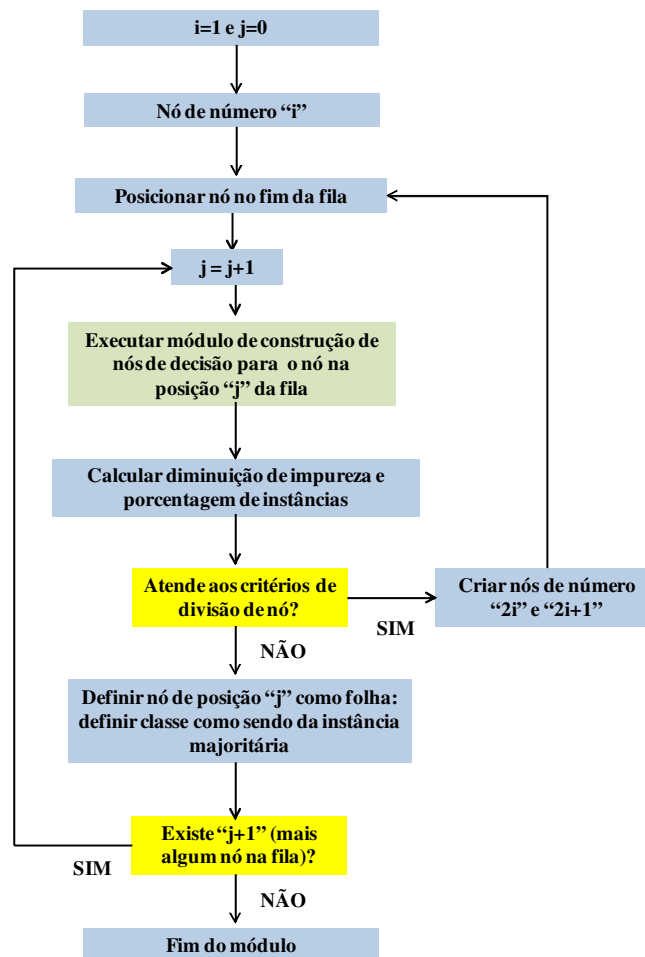


Figura 2.2 – Módulo de Construção de Árvore de Decisão

Nota-se que nesta etapa de divisão dos nós são utilizados parâmetros definidos pelo usuário e que visam limitar o crescimento da árvore de decisões, realizando a técnica de pré-poda. Além deste fato, a pré-poda também evita o *overtraining* do classificador e a forte influência que *outliers* poderiam ter no treinamento do mesmo.

Caso os critérios para geração de novos nós sejam cumpridos, os “nós filhos” são gerados e os números identificadores destes novos nós são inseridos em uma fila. Esta fila tem por objetivo organizar os nós gerados de maneira a percorrer a árvore em largura, e que cada nó seja, por sua vez, submetido ao critério triplo para divisão de nós.

2.2.2 Módulo de construção de nós de decisão.

Neste módulo, conforme pode-se observar na Figura 2.4, as amostras de treinamento relativas a K classes previamente selecionadas pelo usuário serão os dados de entrada. Para que seja construído o nó primeiramente é necessário transformar o problema de K classes (sendo K maior do que duas) em um problema de duas classes, visto que a árvore que se deseja implementar é binária (OBCT).

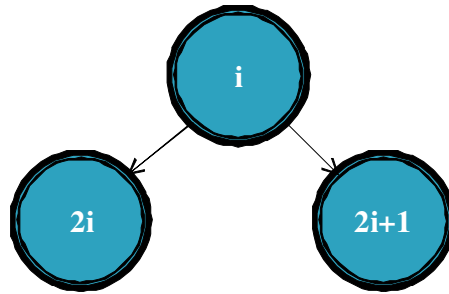


Figura 2.3 – Geração e identificação de novos nós

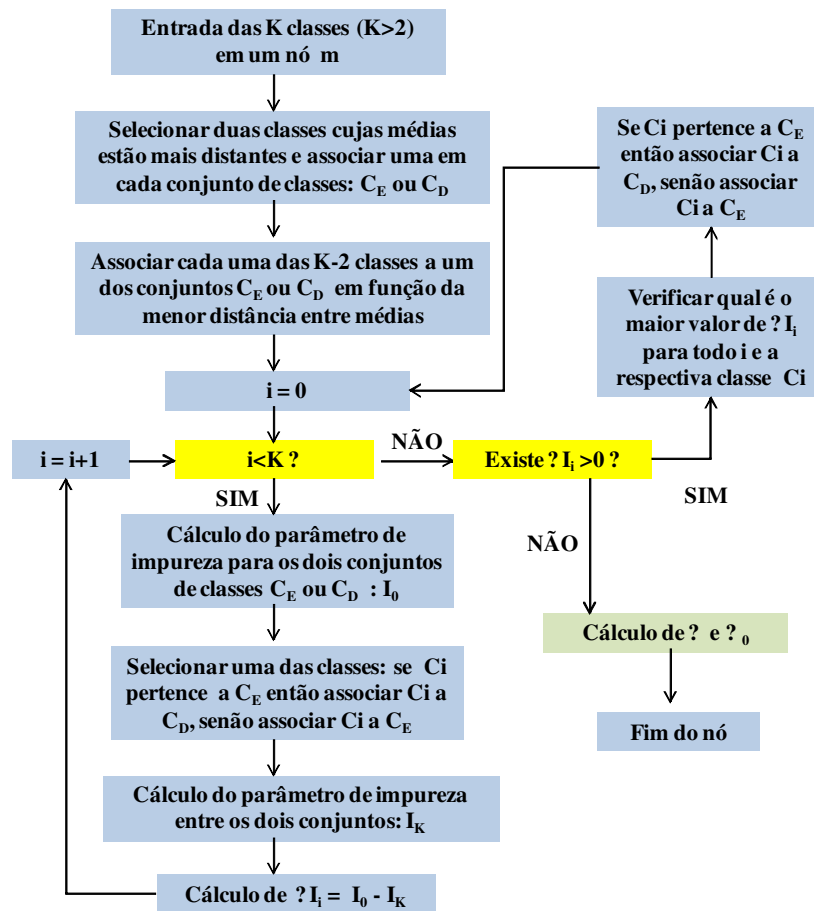


Figura 2.4 – Módulo de Construção de Nós de Decisão (Método da Troca)

Neste caso optou-se por utilizar o método de troca, onde inicialmente são formados dois conjuntos de classes cujas médias das classes “semente” encontram-se mais distantes entre si. Após a formação desses conjuntos, inicia-se o processo de troca onde cada uma das classes é retirada do seu conjunto original e alocada no outro, calculando-se a variação no valor de impureza que ocorre neste processo. A classe que, em caso de troca, ocasionar a maior diminuição no valor de impureza é então efetivamente trocada, passando a fazer parte do outro conjunto de classes. O processo então reinicia calculando mais uma vez os valores de variação de impureza para cada uma das classes e analisando se haverá uma nova troca.

Ao término do processo, quando nenhuma outra troca de classes entre os dois conjuntos diminui o valor de impureza, inicia-se um novo módulo, de cálculo dos valores de ω e ω_0 . O módulo de construção de nós de decisão chega ao fim assim que são obtidos os valores dos parâmetros para construção da regra de decisão.

No presente trabalho, para avaliar a impureza, será utilizado o cálculo de entropia. Este valor é dado conforme a Equação 2.1, sendo $P(\omega_i | t)$ a probabilidade de ocorrência de ω_i em X_t . Computacionalmente o valor desta probabilidade será estimado através da razão entre o número de instâncias de uma determinada classe i e o número total de instâncias. Já o valor da redução de impureza no nó será dado pela Equação 2.2 onde N_t é o número total de instâncias no nó e N_{tE} e N_{tD} é o número de instâncias no nó da esquerda e da direita respectivamente.

$$I(t) = -\sum_{i=1}^M P(\omega_i | t) \log_2 P(\omega_i | t) \quad (2.1)$$

$$\Delta I(t) = I(t) - \left(\frac{N_{tE}}{N_t} I(t_E) + \frac{N_{tD}}{N_t} I(t_D) \right) \quad (2.2)$$

2.2.3 Módulo de cálculo de ω e ω_0

Os valores de ω e ω_0 serão calculados neste módulo. Para que isto ocorra inicialmente é necessário que se tenha duas classes distintas (ou dois conjuntos de classes), o da direita e o da esquerda, as quais foram obtidas no módulo anterior através do Método da Troca.

Com base nos dados dessas duas diferentes classes serão calculadas as médias, o número de amostras de cada uma das classes e a respectiva matriz de covariância. Após isto é possível então aplicar as fórmulas do discriminante linear de Fisher, conforme as Equações 2.3, 2.4, 2.5, 2.6 e 2.7, obtendo-se os valores de ω e ω_0 . Nas equações, \mathbf{m}_E e \mathbf{m}_D são respectivamente os vetores de médias dos conjuntos de classes da esquerda e da direita, \mathbf{m}^* é o vetor de média projetado no eixo ω^T , n é o número de instâncias (ou observações) e S é a matriz de espalhamento.

$$\mathbf{m}_E^* = \frac{1}{n_E} \sum_{x^t \in C_E} \omega^T x^t = \omega^T \mathbf{m}_E \quad (2.3)$$

$$S_E = \sum_{x^t \in C_E} (x^t - \mathbf{m}_E)(x^t - \mathbf{m}_E)^T \quad (2.4)$$

$$S = S_E + S_D \quad (2.5)$$

$$\omega = S^{-1}(\mathbf{m}_E * -\mathbf{m}_D *) \quad (2.6)$$

$$\omega_0 = -\frac{1}{2}(\mathbf{m}_E * +\mathbf{m}_D *)^T S^{-1}(\mathbf{m}_E * -\mathbf{m}_D *) + \log \frac{n_E}{n_D} \quad (2.7)$$

2.2.4 Módulo de Classificação

A etapa de classificação de uma imagem no algoritmo desenvolvido é realizada pixel a pixel. Cada pixel, sendo considerada uma instância definida por um vetor com os respectivos valores dos atributos, é submetido às regras lógicas do tipo se-então-senão de cada nó. Desta forma cada instância percorre um caminho na árvore até atingir uma folha correspondente à uma determinada classe a qual aquele pixel ficará associado. Este fluxo encontra-se ilustrado na Figura 2.5.

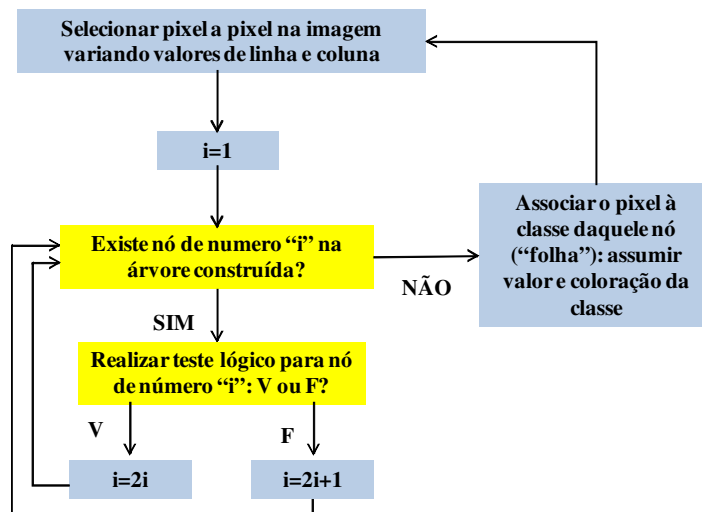


Figura 2.5 – Módulo de Classificação da Imagem

2.3 Análise do Algoritmo Desenvolvido

Para analisarmos o algoritmo de classificação desenvolvido inicialmente são identificadas as variáveis que influenciam em seu processamento. Neste caso foram identificadas 3 (três) variáveis: N (número total de pixels na imagem), k (número de classes definidas na etapa de treinamento) e n (número total de pixels na amostra de treinamento).

A partir destas variáveis foi possível analisar a complexidade de cada módulo, verificando sua ordem. Neste caso observa-se que o algoritmo completo pode ser dividido em duas etapas: módulo de construção de árvore de decisão e módulo de classificação da imagem.

Ao analisarmos o módulo de construção de árvore de decisão, verificamos que a complexidade do mesmo é de $O(n)$. Entretanto, dentro deste módulo existe o módulo de construção de nós de decisão que, por sua vez, possui complexidade de $O(k^2)$, dependente do número de classes definida pelo usuário na etapa de treinamento.

Multiplicando ambos os termos temos que a etapa referente ao módulo de construção de árvore de decisão tem complexidade de $O(n.k^2)$.

A próxima etapa do algoritmo refere-se ao módulo de classificação da imagem. Por ser baseada em uma árvore binária, neste caso a complexidade é de $O(N \cdot \log_2^N)$.

Somando ambas as etapas do algoritmo desenvolvido, temos que o mesmo possui complexidade de $O(n.k^2 + N \cdot \log_2^N)$. Neste caso observa-se que o desempenho do algoritmo depende das três variáveis anteriormente definidas, sendo mais influenciada pelo número de classes em trabalho.

3 Testes, Resultados e Avaliações

Visando avaliar o algoritmo desenvolvido, foram realizados alguns testes. Estes testes podem ser divididos em duas categorias: sobre imagens sintéticas e sobre imagens de sensoriamento remoto. Em ambos os casos a avaliação dos resultados foi realizada somente quantitativamente através do uso das matrizes de confusão, acurácia global e respectivos índices Kappa.

3.1 Imagens Sintéticas

Foram utilizadas 02 (duas) imagens sintéticas para que fossem feitos os testes do classificador desenvolvido: xxx10 e sintético-RGB. O objetivo destes testes foi de avaliar o classificador utilizando-se imagens controladas, isto é, imagens cujas quantidades de classes e nível de ruído eram conhecidos.

Para que fosse possível comparar os resultados, foram definidos alguns procedimentos como padrões:

- para a classificação por árvore oblíqua os parâmetros de divisão de nós (ou pré-poda) tiveram as porcentagens de 5% do tamanho mínimo do conjunto e de 10% de diminuição mínima no valor de entropia;
- para as etapas de treinamento foram utilizadas amostras de 100 pixels por classe; e
- para as etapas de avaliação foram utilizadas amostras de 10.000 pixels por classe.

Inicialmente, aplicando-se o algoritmo de árvore oblíqua sobre as amostras de treinamento coletadas da imagem xxx10, foi construída uma árvore de com as seguintes regras de classificação:

- Nó 1 $\{-0.0172510 * R + 0.0414885 * G - 0.0212578 * B - 0.521221 \text{ LE } 0\}$;
- Nó 2 $\{0.0582046 * R - 0.0399582 * G - 0.0235878 * B + 1.10185 \text{ LE } 0\}$;
- Nó 3 $\{-0.0497875 * R + 0.0550983 * G - 0.00703979 * B + 1.17127 \text{ LE } 0\}$;
- Nó 4 {Classe majoritária 2};
- Nó 5 {Classe majoritária 4};
- Nó 6 {Classe majoritária 1}; e
- Nó 7 {Classe majoritária 3}.

Nesta regra de classificação os valores das bandas de composição colorida do vermelho, do verde e do azul são dadas respectivamente pelos valores de R, G e B. Já, a regra binária LE se refere ao termo “menor ou igual”.

A avaliação quantitativa desta classificação apresentou acurácia global de 60,76% e índice Kappa de 0,4767, havendo maior confusão entre as classes 1 e 2.

No caso da imagem sintético-RGB, aplicou-se o algoritmo obtendo-se a árvore de decisão com as seguintes regras de classificação:

- Nó 1 $\{-0.176577 * R + 0.00846551 * G + 0.0402511 * B + 6.6728 \text{ LE } 0\}$;
- Nó 2 $\{0.0942192 * R - 0.343239 * G + 0.0980717 * B + 24.7215 \text{ LE } 0\}$;
- Nó 3 $\{0.00687872 * R - 0.233677 * G + 0.103446 * B + 15.0405 \text{ LE } 0\}$;
- Nó 4 {Classe majoritária 1};
- Nó 5 $\{-0.248126 * R + 0.127104 * G + 0.0554282 * B - 3.69978 \text{ LE } 0\}$;
- Nó 6 {Classe majoritária 3};
- Nó 7 {Classe majoritária 4};
- Nó 10 {Classe majoritária 2}; e
- Nó 11 {Classe majoritária 4}.

A avaliação quantitativa neste caso apresentou acurácia global de 99,71% e índice Kappa de 0,9960, apresentando um resultado visualmente bastante homogêneo, sem confusão nas bordas que separam as classes.

3.2 Imagem de Sensoriamento Remoto

Os testes sobre uma imagem real foram realizados sobre dados de Radar de Abertura Sintética – SAR – do Projeto Cartografia da Amazônia, mais especificamente o Subprojeto Cartografia Terrestre em execução pela Diretoria de Serviço Geográfico (DSG) do Exército Brasileiro.

A área de trabalho foi selecionada de acordo com a disponibilidade das imagens SAR. Neste sentido foram analisadas cenas de uma região próxima a cidade de Barcelos – AM, com coordenadas centrais de 63°15'11''w/ 0°44'46''s, no sistema geodésico WGS-84, e datadas de 30 de março de 2009.

Após o preparo das imagens SAR e extração de novos atributos, foram selecionadas as bandas P-HH, P-HV, P-VV, P-Coh, X-HH, Hint e BMI para a extração das seguintes classes a serem trabalhadas no algoritmo de árvore de decisão oblíqua: água, antropizada, campinarana arbórea, campinarana arbustiva, primária alagada e primária firme. Estas classes foram definidas de acordo com o mapa temático fitoecológico do Projeto RADAMBRASIL [Brasil 1978].

Na etapa de coleta de amostras, foram selecionados 50 (cinquenta) polígonos como amostras de treinamento e 30 (trinta) como amostras de teste (ou validação) para cada classe. Os polígonos foram selecionados com uma distribuição homogênea pela imagem e cada um continha cerca de 200 (duzentos) pixels.

A árvore construída, após diversos testes, obteve como o valor máximo de acurácia global de 89,87%, tendo como parâmetros de pré-poda as porcentagens de 30% de diminuição mínima de impureza e de 16% de tamanho mínimo de instâncias para divisão de cada nó.

4 Comparações e Análises

Além dos testes utilizando o classificador por árvores oblíquas desenvolvido no presente trabalho, foram também realizadas classificações utilizando outros métodos tradicionais visando a comparação entre resultados. A comparação foi realizada sobre a

avaliação quantitativa aplicando os métodos de classificação de MaxVer (Máxima Verossimilhança), de Distância de Mahalanobis e SVM (*Support Vector Machine*). Estes métodos foram selecionados e aplicados em função da disponibilização de seus algoritmos no sistema de processamento de imagens ENVI (*ENvironment for Visualizing Images*) versão 4.7.

Em todos os testes onde foi utilizada a classificação pelo método de MaxVer, adotou-se a distribuição normal (Gauss). Já, para a classificação pelo SVM foi adotado o método linear com parâmetro de penalidade de 100. Os resultados encontram-se na Tabela 4.1.

Tabela 4.1 – Comparação de acurácia global (índice Kappa) a partir de diferentes Classificadores

Classificador\Imagem	Sintético-RGB	XXX10	SAR
Árvore Oblíqua	99,71 (0,9960)	60,76 (0,4767)	89,87 (0,8582)
MaxVer	99,85 (0,9980)	60,67 (0,4752)	78,08 (0,7570)
Dist Mahalanobis	99,60 (0,9947)	60,06 (0,4675)	75,58 (0,7072)
SVM Linear	99,73 (0,9964)	62,87 (0,4979)	92,67 (0,9120)

Comparando os resultados apresentados na Tabela 4.1 pode-se observar o seguinte:

- as acurácias globais de todos os classificadores aplicados à imagem sintético-RGB foram bastante semelhantes, ultrapassando os 99% de acerto. Este bom resultado ocorreu em função da pouca quantidade de ruído na imagem. Apesar do melhor resultado ter sido obtido com o classificador MaxVer, a árvore oblíqua apresentou desempenho satisfatório;
- o classificador que apresentou melhor resultado sobre a imagem XXX10 foi o SVM Linear, se aproximando dos 63% de acurácia global. Neste caso a árvore oblíqua também obteve bons resultados superando os classificadores MaxVer e de Distância de Mahalanobis;
- sobre a imagem de SAR o melhor resultado também foi obtido através do classificador SVM Linear, seguido da Árvore Obliqua. Neste caso, além de ter sido observada a presença de ruído no dado a ser classificado, o mesmo não possui distribuição gaussiana, possível motivo pelo qual os resultados nos métodos MaxVer e de Distância de Mahalanobis terem sido inferiores.

5 Conclusões

Os resultados obtidos pelo classificador de Árvore Oblíqua obtiveram resultados semelhantes aos de outros classificadores tradicionais. O algoritmo foi capaz de classificar todas as imagens testadas obtendo resultados que variaram em função da quantidade de ruído presente.

Nos testes sobre imagens sintéticas, observou-se que quanto menor a quantidade de ruído inserido na imagem controlada, maior a acurácia global atingida.

Consequentemente, na imagem com pouca presença de ruído obteve-se um percentual acima de 99% de acerto na classificação.

Além das imagens sintéticas, o resultado obtido no teste sobre uma imagem de sensoriamento remoto comprova que o algoritmo pode também ser aplicado sobre estes dados. O processo de classificação através de árvores de decisão independe do tipo de dado e de distribuição estatística para obter bons resultados.

Comparando com outros classificadores é possível observar que o método de classificação por árvore de decisão, por ser não-paramétrico, obteve destaque sobre imagens ruidosas e com distribuição diferente da gaussiana. Apesar de não ter obtido resultados superiores ao do SVM Linear, em todos os testes realizados o tempo de processamento foi inferior ao daquele método. Este baixo custo computacional é evidenciado pela simplicidade na complexidade do algoritmo desenvolvido.

Novos testes sobre o algoritmo desenvolvido ainda são necessários. É importante realizar a comparação entre resultados obtidos com o uso desta árvore oblíqua com o de árvores univariadas. Esta comparação deverá ser feita sobre o tamanho e complexidade das árvores construídas, acurácia global e índice Kappa obtidos e custos computacionais envolvidos.

Referências

- Brasil (1978). Departamento Nacional da Produção Mineral. “Projeto RADAMBRASIL”. Folha AS.20 Manaus; geologia, geomorfologia, pedologia, vegetação e uso potencial da terra. 628 p. Rio de Janeiro. CDD 558.1
- Congalton, R. G.; Green, K. (1999). “Assessing the Accuracy of Remotely Sensed Data: Principles and Practices”. Lewis Publishers. Denver. EUA. 180 p. ISBN 0873719867.
- Duda, R.O.; Hart, P.E.; Stork, D.G. (2001). “Pattern Classification”. 2nd ed. John Wiley & Sons. NY.
- Friedl, M. A.; Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. “Remote Sensing of Environment”. 61. Pg 399-409.
- Pal, M.; Mather, P. M. (2001). “Decision Tree Classification of Remotely Sensed Data”. Anais da 22a Asian Conference on Remote Sensing. Singapore. Disponível em: <http://www.crisp.nus.edu.sg/~acrs2001/pdf/046PAL.PDF> . Acesso em 02 Out 2009.
- Quinlan. J. R. (1993). “C4.5: Programs for Machine Learning”. Morgan Kaufmann. California.
- Theodoridis, S.; Koutroumbas, K. Pattern Recognition. 3. ed. San Diego: Academic Press, 2006. 651 p.
- Yildiz, O. T.; Alpaydin, E. (2005). Linear discriminant trees. “International Journal of Pattern Recognition and Artificial Intelligence”. Vol 19, No. 3.