Anais XIV Simpósio Brasileiro de Sensoriamento Remoto, Natal, Brasil, 25-30 abril 2009, INPE, p. 7219-7225.

# Bagging e árvore de decisão para a classificação de imagens no sistema SACI

Carlos Frederico de Sá Volotão [1,2]

[1] Diretoria de Serviço Geográfico - DSG/EB
QGEx – SMU – 70630-901 - Brasília – DF, Brasil

[2] Instituto Nacional de Pesquisas Espaciais - INPE
Caixa Postal 515 - 12245-970 - São José dos Campos - SP, Brasil
volotao@dpi.inpe.br

**Abstract.** Classification trees are used to classify remote sensing images. Trees sort instances down a branch of the tree, from root to leaf, each node on the way questioning a specific attribute of that instance. Each branch leaving each node represents a value of the attribute that was tested at that node. Usually the trees are represented upside down, with the root at the top and the leaves at the bottom. Trees are widely used algorithms and therefore there is a big number of literature concerning to this, which is an advantage in the implementation of this classifier. The over-fitting of the training samples may occur if both the stop criteria and the pruning are not effective to generalize the tree. To enhance the classification performance the bagging is an alternative where several trees are constructed by randomly sorting the training samples to obtain different subsamples called bootstraps. Each tree provides a classification result and the final result is the class frequency majority chosen by vote. This makes the final classification more generic and can enhance the overall accuracy. An academical system for research purposes called "Sistema de Análise e Classificação de Imagens" (SACI) was developed. This environment provides the basic operations for choosing the files, defining the regions of interest and displaying the results by charts and images. The best visual classification was achieved by the last instances although with the worse numeric results. The image results corresponding to the cases above 90% are very homogeneous and much useful than the best kappa of 0,93, which resulted in more "granulated" classes. The tests pointed the best time achieved by the last instance, 25 seconds to train, classify and display the results, but about 5 minutes in the worst case.

**Keywords:** image processing, remote sensing, bagging, decision tree, processamento de imagens, sensoriamento remoto, sistema de análise e classificação de imagens, árvore de decisão.

## 1. Introdução

Para testar diferentes métodos de processamentos de imagens, diversas ferramentas foram integradas em um sistema, construído em linguagem IDL, chamado Sistema de Análise e Classificação de Imagens (SACI), que hoje está sendo remodelado no INPE.

O SACI é uma versão expandida de um sistema denominado Sistema de Classificação de Imagens Digitais (SCID) e apresenta diversos tipos de classificadores supervisionados e não-supervisionados, ferramentas de manipulação de amostras e de análise de imagens de sensores passivos e ativos, onde pode-se agregar vários módulos de maneira independente (Rosa et al., 2005).

Em 2007 foram adicionados ao sistema o *bootstrap aggregating* (*bagging)* e a árvore de decisão dentre outros classificadores. Este trabalho apresenta uma breve descrição desses dois novos recursos e a influência dos parâmetros disponibilizados.

## 2. Bagging no SACI

Busca-se para um classificador sempre o melhor desempenho de classificação (Zhu et al., 2004). O *bagging* é um método que interfere na aprendizagem, realizando combinações a partir do conjunto das amostras em novos conjuntos que apresentam repetições e omissões de pixels. O *bagging* consiste no sorteio com reposição de B conjuntos chamados *bootstraps* de amostras produzindo B classificadores, cada um com o seu valor de classificação para o

mesmo dado. A votação define o valor final. Isto faz com que a classificação torne-se mais genérica e pode melhorar a exatidão geral.

No SACI, o *bagging* foi escrito associado ao classificador do tipo árvore de decisão ou classificação, cuja implementação está voltada prioritariamente para a classificação de imagens de sensoriamento remoto. Em breve será possível utilizá-lo com outros classificadores disponíveis no mesmo ambiente, como apresentado por Rogers (2003), por meio de combinação de classificadores.

O *bagging* aplicado à árvore de decisão pode evitar que o treinamento dos dados produza uma árvore muito especializada, pois produz novos conjuntos de dados a partir dos dados originais aleatoriamente repetindo-se alguns dados e desconsiderando-se outros. Gerando-se vários novos conjuntos define-se o mesmo número de novas árvores. A maneira de se utilizar múltiplas árvores é aplicar cada classificador aos dados e utilizar um critério para a seleção da classe final. Tradicionalmente esse método é a votação por maioria.

Se classes distintas contiverem número de elementos muito distinto, essa diferença de pixels pode influenciar a classificação das classes. Esse desbalanço de número de elementos, a priori, dos conjuntos de amostra pode ser indesejada e fazer reduzir o número de pixels que recebe uma dada classificação.

Para isso o bagging apresenta a possiblidade de se escolher o número de amostras por classe. Na implementação do *bagging* pode-se, para as amostras:

(a) fazer um sorteio de todas as amostras indistintamente;

(b) forçar com que cada classe tenha o mesmo número original de elementos das amostras, fazendo-se os sorteios internos a cada classe;

(c) fazer com que todas as classes tenham o mesmo número de elementos, e que esse número seja igual ao número de amostras:

- da menor classe;
- da maior classe;
- médio por classe;
- definido pelo usuário;

Tendo as amostras de pixels em imagens forte correlação entre os atributos dos vizinhos, é importante considerar a representatividade de uma subclasse de pixels: amostras de respostas diferentes dentro de uma mesma classe podem sofrer com o desbalanceamento numérico de representantes daquela subclasse. É importante saber desse efeito pois o efeito da classe pode ser minimizado pelo procedimento de igualar as classes, mas as diferenças nas classes não são resolvidas.

Os resultados do *bagging* apresentaram melhora, mas essa melhora se mostrou percentualmente baixa em relação à qualidade da classificação. Em valores isso será abordado na seção 4 deste trabalho.

## 3. Árvore de Decisão com bagging no SACI

A classificação por árvores de decisão foi também implementada. Por ser um classificador vastamente utilizado torna-se importante a existência do presente módulo no SACI.

Uma árvore de decisão é equivalente a uma série de condicionais a respeito dos atributos dos dados. A Figura 1 apresenta de modo claro e simples um exemplo de árvore de decisão de 1 nível de profundidade e 4 folhas para classificar em duas classes C1 e C2 imagens de 3 bandas B1, B2 e B3.

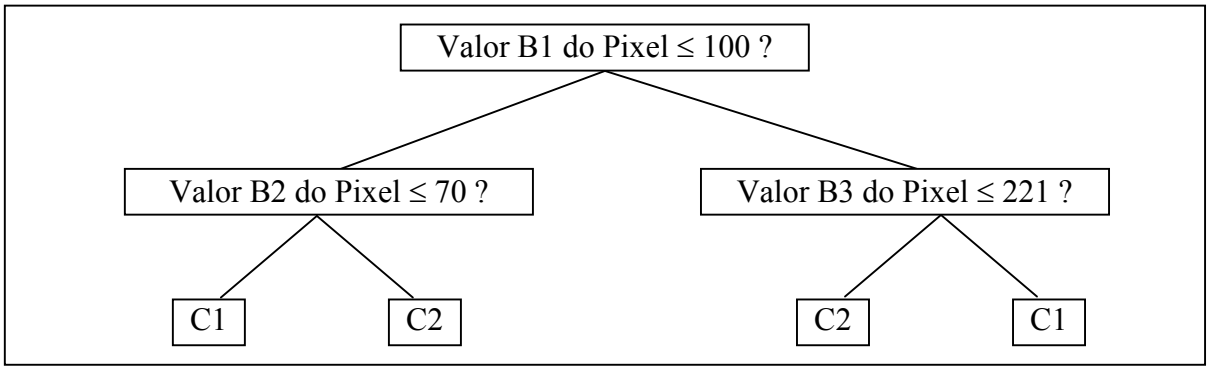


**Figura 1**. Um exemplo de árvore de decisão

A classificação por árvores ocorre da raiz para as folhas, com a raiz para cima e as folhas para baixo. No caso de árvores binárias univariadas de classificação, a partir da raiz há sempre duas opções a seguir. Cada nó ou ramo define uma banda e um valor de divisão. Se o valor do pixel para a banda definida pelo ramo for superior ao valor de divisão, passa-se para um dos lados. Se inferior, para o outro. Um lado pode ser uma folha ou um novo ramo. Se for um novo ramo o processo se repete. Se for uma folha, o pixel é associado a uma classe definida pela folha (Rogers, 2003). Um exemplo simples de classificação de imagem de 3 bandas é visto na Figura 1.

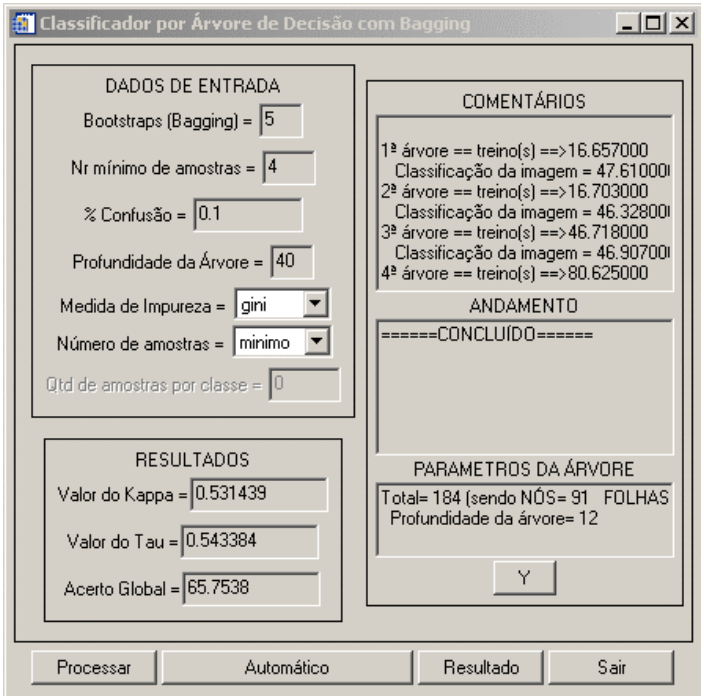


**Figura 2.** Janela de parâmetros para a classificação por árvore com *bagging* no SACI.

A fase de construção da árvore é crítica e a generalização pode não atingir um grau ideal. Para evitar o *overfitting,* são disponibilizados alguns critérios de parada, como número máximo de nós, grau de impureza aceitável para um nó, profundidade máxima da árvore e número de elementos de cada classe no conjunto de amostras. Isso pode evitar que alguns nós sejam criados, sendo os parâmetros um recurso importante para que os resultados.possam ser melhorados.

Pode-se definir um número de *bootstraps,* fazendo com que mais ou menos árvores sejam construídas a partir da combinação de pixels presentes nas amostras.

A modificação desses parâmetros permite que o processo de construção da árvore torne-se muito mais rápida, mas a precisão pode ser totalmente comprometida de acordo com os valores fornecidos.

Na Figura 2 é apresentada a janela de execução de árvore com *bagging*. Os parâmetros são: o número de *bootstraps* (B); o mínimo número de amostras para definir uma folha; a tolerância de pureza para definir uma folha; o limite de profundidade da árvore; o método de medição da impureza; e o número de pixels por classe a ser usada. O método de impureza pode ser escolhido entre "gini" ou "entropia".

Os cinco botões abaixo têm o seguinte significado: "Y" inverte o eixo Y, "Processar" executa a classificação, "Automático" executa uma seqüência classificações com parâmetros distintos, "Resultado" abre a janela de resultados, e "Sair" volta ao menu básico do SACI.

Entre muitos critérios de particionamento apresentados na literatura científica foi selecionado o método de entropia para definir o ponto de divisão. A entropia é uma medida de impureza. O valor exato do ponto de divisão é definido pelo ganho da divisão pelo índice utilizado.

Sem dispositivos aceleradores o treinamento é exaustivo. Deve-se comparar cada valor de atributo de cada banda entre si e a que proporcionar o maior ganho geral é a escolhida para aquele ramo. Um nó na árvore corresponde a uma divisão nos dados e todos os cálculos são refeitos.

Os índices usados foram: ganho de entropia e ganho pelo "Gini Index" devido aos bons resultados indicados na literatura (Brodley and Utgoff, 1992). Breiman et al. (1984), Quinlan (1986), Mingers (1989), Safavian e Landgrebe (1991), Buntine e Niblett (1992), e Fayyad e Irani (1992) comparam e discutem diferentes critérios de mérito de partição.

## 4. Resultados e Discussão

Os testes não sejam exaustivos, mas um breve estudo de caso. Correspondem à imagem da Figura 3, de "Lena" com 5 classes, que corresponde a uma classificação usando 3 bootstraps, 15 amostras por folha no mínimo, 30% de tolerância para considerar homogênea a folha, sem limite de profundidade máxima e método de impureza "gini index".

A Tabela 1 fornece um caso com o objetivo é analisar o efeito dos parâmetros da árvore e do *bagging*. Os casos 4-5 e 18-19 apresentam redução numérica do erro global ao se aumentar o número de *bootstraps*, em 1,5% e 1,2% respectivamente. O caso 21-23 ocorre uma inversão quando se passa de 10 para 5 e um aumento de 5 para 3 *bootstraps*. Isso demonstra que para valores baixos de amostras por classe (200) uma variação de 5 bootstraps não garante uma melhora significativa.

O efeito aproximado de *bagging* para outros casos similares estudados variou levemente, ficando entre 1% a 4%, dependendo dos parâmetros.

O caso 11 deveria ter se aproximado de 100%, entretanto o valor global de acerto foi de 93,3%, tendo a árvore atingido uma profundidade de 49 níveis, não excedendo o limite. Não havia outro critério de parada impedindo que chegasse a 100% entretanto o limiar atingido mostra a existência de confusão nos conjuntos de amostra. Somente uma nova criteriosa seleção de amostras pode melhorar esse conjunto. Isso demonstra o quanto é sensível a ruídos e que a escolha das amostras é de fundamental importância.

Durante a execução dos testes notou-se que as melhores classificações em termos visuais foram obtidas nas últimas linhas da Tabela 1, com resultados muito mais homogêneos, embora com os piores resultados numéricos. Os melhores tempos também foram o dos últimos, chegando a 25 segundos, em comparação a 5 minutos do mais demorado.

Não foi implementado algoritmo de poda, Helmbold e Schapire (1995) apresentam um algoritmo que declaram ser eficiente para podar, mas requer um segundo conjunto de treinamento, sendo mais uma opção para melhorar o resultado. Kearns and Mansour (1998) apresentam um algoritmo bottom-up para poda que somente requer uma passagem e não necessita de outro conjunto de treinamento, para aumentar a robustez do classificador. A poda apresentada por Rastogi e Shim (1998) pode ser aplicada ainda na fase construção.

Para evitar *overfitting* pode-se adicionar ruído nos dados de treinamento e usar um método descrito por Ho (1998).

**Tabela 1.** Resultado de teste da imagem "Lena" com diferentes parâmetros e 27 casos.

| item | B | mpix | impurity | maxdepth | method | ppc | nodes | depth | kappa | tau | global |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | - | 15 | gini | original | 820 | 16 | 0,923 | 0,925 | 94,0% |
| 2 | 0 | 0 | - | 50 | gini | original | 1248 | 40 | 0,927 | 0,928 | 94,3% |
| 3 | 0 | 0 | - | 50 | entropia | original | 1096 | 51 | 0,925 | 0,926 | 94,1% |
| 4 | 1 | 0 | - | 50 | gini | min | 800 | 47 | 0,901 | 0,902 | 92,2% |
| 5 | 10 | 0 | - | 50 | gini | min | 844 | 51 | 0,921 | 0,922 | 93,7% |
| 6 | 1 | 0 | - | 50 | gini | original | 1168 | 51 | 0,903 | 0,904 | 92,3% |
| 7 | 1 | 0 | - | 50 | gini | aleatorio | 1010 | 29 | 0,908 | 0,910 | 92,8% |
| 8 | 1 | 0 | - | 50 | gini | med | 1056 | 51 | 0,912 | 0,913 | 93,1% |
| 9 | 1 | 0 | - | 50 | gini | max | 1556 | 51 | 0,921 | 0,922 | 93,7% |
| 10 | 1 | 0 | - | 50 | gini | max | 1450 | 51 | 0,915 | 0,916 | 93,3% |
| 11 | 1 | 0 | - | 100 | gini | max | 1428 | 49 | 0,916 | 0,917 | 93,3% |
| 12 | 1 | 2 | - | 100 | gini | max | 418 | 17 | 0,900 | 0,901 | 92,1% |
| 13 | 1 | 3 | - | 100 | gini | max | 348 | 15 | 0,900 | 0,901 | 92,1% |
| 14 | 1 | 0 | 0,05 | 100 | gini | max | 1076 | 52 | 0,903 | 0,904 | 92,3% |
| 15 | 1 | 0 | 0,10 | 100 | gini | max | 420 | 43 | 0,886 | 0,888 | 91,0% |
| 16 | 1 | 0 | 0,15 | 100 | gini | max | 310 | 37 | 0,874 | 0,876 | 90,1% |
| 17 | 1 | 3 | 0,15 | 100 | gini | max | 146 | 14 | 0,861 | 0,863 | 89,0% |
| 18 | 1 | 3 | 0,15 | 100 | gini | min | 58 | 12 | 0,858 | 0,860 | 88,8% |
| 19 | 20 | 3 | 0,15 | 100 | gini | min | 78 | 10 | 0,873 | 0,875 | 90,0% |
| 20 | 10 | 3 | 0,15 | 100 | gini | 200 | 48 | 8 | 0,868 | 0,870 | 89,6% |
| 21 | 10 | 6 | 0,15 | 100 | gini | 200 | 40 | 9 | 0,860 | 0,862 | 89,0% |
| 22 | 5 | 6 | 0,15 | 100 | gini | 200 | 38 | 9 | 0,863 | 0,865 | 89,2% |
| 23 | 3 | 6 | 0,15 | 100 | gini | 200 | 34 | 8 | 0,857 | 0,858 | 88,7% |
| 24 | 3 | 10 | 0,15 | 100 | gini | 200 | 34 | 7 | 0,843 | 0,845 | 87,6% |
| 25 | 3 | 6 | 0,30 | 100 | gini | 200 | 20 | 8 | 0,732 | 0,735 | 78,8% |
| 26 | 3 | 10 | 0,30 | 100 | gini | 200 | 16 | 6 | 0,747 | 0,749 | 79,9% |
| 27 | 3 | 15 | 0,30 | 100 | gini | 300 | 15 | 5 | 0,751 | 0,753 | 80,2% |

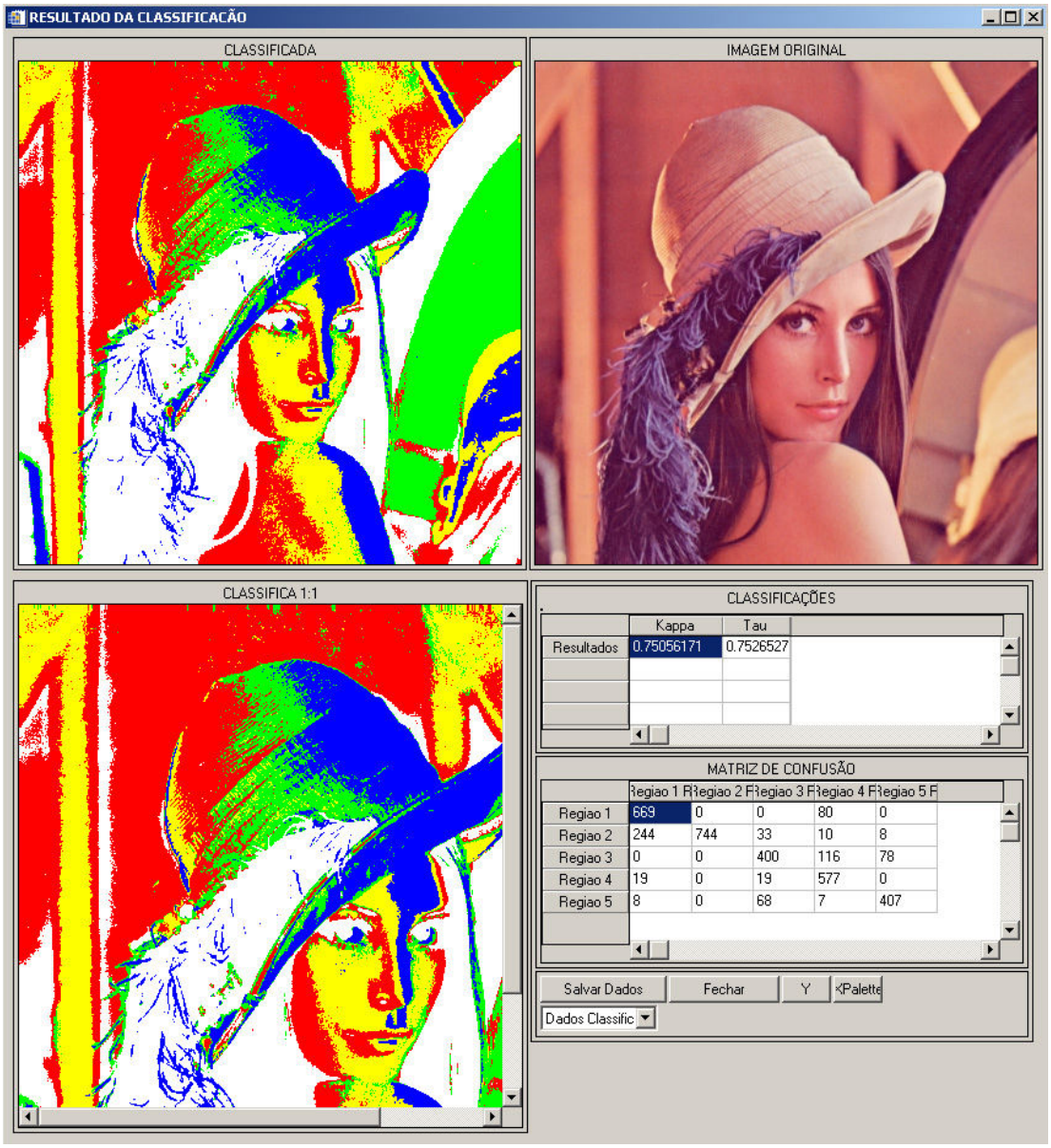


**Figura 3.** Resultado da classificação por árvore com *bagging* no SACI.

## 5. Conclusão

Este trabalho apresenta a implementação de bagging e árvore de decisão no sistema SACI, e descreve suas variáveis, apresentando uma breve descrição de seus parâmetros.

A qualidade de classificação em termos de acertos em relação aos próprios dados das amostras é um pouco melhorada com o uso do *bagging*.

Os parâmetros reduzem a complexidade e são capazes de tornar a imagem mais homogênea e aumentar o efeito de generalização visual porém reduzem um pouco os índices de exatidão.

O sistema SACI foi escrito em linguagem de programação IDL e é parte de um sistema acadêmico do INPE de estudo de ferramentas de processamento de imagens.

## References

Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. **Classification and regression trees**. Belmont, CA: Wadsworth International Group. 1984

Brodley, C. E.; Utgoff, P. E. Multivariate decision trees. **COINS Technical Report 92-82**, University of Massachussets, Dec, 1992. (online: 04-mar-2008) http://www.cs.iastate.edu/~honavar/brodley-dtree.pdf

Buntine, W.; Niblett, T. A further comparison of splitting rules for decision-tree induction. In: **Machine Learning**, 8, pp.75-85. 1992.

Fayyad, U. M.; Irani, K. B. The attribute selection problem in decision tree generation. In: **Proceedings of the Tenth National Conference on Artificial Intelligence**, pp. 104-110. San Jose, CA: MIT Press, 1992.

Helmbold, D. P.; Schapire, R. E. Predicting nearly as well as the best pruning of a decision tree. In: **Proceedings of the Eighth Annual Conference on Computational Learning Theory**, ACM Press, pages 61-68, 1995.

Ho, T. K. The random subspace method for constructiong decision forests. In: **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 20, 8-aug-1998, pp.832-844.

Kearns, M.; Mansour, Y. A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In: **Proc. 15th International Conf. On Machine Learning**, 1998. (online: 04-mar-2008) http://citeseer.ist.psu.edu/82070.html

Mingers, J. An empirical comparison of selection measures for decision tree induction. In: **Machine Learning**, 3, pp. 319-342. 1989.

Quinlan, J. R. Induction of decision trees. In: **Machine Learning**, 1, pp.81-106. 1986.

Rastogi, R.; Shim, K. PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning. In: **Proceeding of the 24 th VLDB Conference**, New York, 1998.

Rogers, S. **Non linear transformations for the creation of diverse classifier ensembles**. Dissertation Project, University of Sheffield, England, 2003. (Online: 04-mar-2008) http://www.dcs.shef.ac.uk/intranet/teaching/projects/archive/ug2003/pdf/u8sr.pdf

Rosa, G.; Dutra, L. V.; Freitas, C. C. Extração de atributos via modelos Arma 1D/2D, avaliação estatística da distância JM e classificador condicional contextual. In: **Anais** do V Worcap, INPE, São José dos Campos, 2005.

Safavian, S. R.; Langrebe, D. A survey of decisión tree classifier methodology. In: **IEEE Transactions on Systems, Man and Cybernetics**, 21, pp.660-674. 1991.

Zhu, X.; Wu, X.; Yang, Y. Effective classification of noisy data streams with attribute-oriented dynamic classifier selection. In: **Proceedings of the 4th IEEE International Conference on Data Mining**, pp. 305-312, Brighton, UK, 2004.