



C4.5: Um Recurso para Geração de Árvores de Decisão

Ricardo Martins Bernardes¹

1. Introdução

Nos últimos anos muitos produtos de software tradicionais como Oracle (*Oracle Data Mining*), SPSS (*Clementine*), SAS (*SAS Data Mining*) tem incorporado módulos voltados à exploração de dados com o objetivo de descobrir conhecimento útil a partir de banco de dados. Estes módulos geralmente empregam técnicas de *mineração de dados (data mining)*, e são a base de uma área emergente da ciência da computação denominada *Knowledge Discovery in Databases (KDD)*. Além da adição de módulos específicos para este processo em pacotes já consagrados, também tem surgido uma infinidade de sistemas específicos para esta finalidade.

KDD combina técnicas de aprendizado de máquina, reconhecimento de padrões, estatísticas, banco de dados e visualização para extrair padrões interessantes a partir de bancos de dados. A tarefa básica é extrair conhecimento (ou informação) de dados brutos, e as ferramentas básicas usadas para extrair padrões de dados utilizam técnicas e métodos de mineração de dados. Dentre as várias técnicas utili-

zadas para mineração de dados destacam-se as redes neurais, os algoritmos genéticos, o aprendizado de máquina, as estatísticas tradicionais e as árvores de decisão. Já o processo de usar estas técnicas e ferramentas (incluindo pré-processamento, seleção e transformação de dados), interpretar e transformar padrões em "conhecimento", é o processo de KDD em si.

O conhecimento extraído é subsequente utilizado para suportar tomadas de decisão, tarefas de predição e de classificação, sumarização de bases de dados ou explicar fenômenos observados. O uso de sistemas para KDD alivia o problema de analisar grandes montantes de dados manualmente, sendo empregados atualmente em áreas como finanças, economia, biologia, astronomia, análise de mercado, diagnósticos, detecção de fraudes e manufatura, inteligência de mercado entre outros.

Um método para elaboração de árvores de decisão e para geração de regras amplamente utilizado e incorporado por outras ferramentas para mineração de dados é o C4.5, que é implementado por um software

¹ Mestre em Administração de Empresas, Assistente de Operações II, Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo – 13083-970 – Campinas, SP. (E-mail: ricardo@cnptia.embrapa.br)

com o mesmo nome, desenvolvido pelo Prof. Ross Quinlan (Quinlan, 2001). Este método é uma evolução do método anterior, conhecido como ID3 (Iterative Dichotomizer 3). Por ter sua teoria descrita em livro (Quinlan, 1993) e ser disponibilizado com fontes, o método disseminou-se rapidamente e hoje é incorporado em várias ferramentas comerciais da categoria.

Segundo Quinlan (1993) muitas aplicações da inteligência artificial, para tarefas de importância prática, são baseadas na construção do modelo de conhecimento usado por um especialista humano. Esta abordagem que teve impacto maior somente a partir de meados da década de 1980 está ilustrada por numerosos estudos de casos reportados em *The Rise of the Expert Company* (Feigenbaum et al., 1998). Em alguns casos, a tarefa que um expert executa pode ser pensada como uma *classificação* – alocação de coisas em categorias ou classes determinadas pelas suas propriedades. Por exemplo, Feigenbaum et al. (1998) citam um sistema desenvolvido pela American Express para assistir os autorizadores de crédito. As propriedades para esta aplicação eram os detalhes da transação proposta e os detalhes da história de crédito do cliente. As classes correspondentes eram a recomendação de aprovar ou desaprovar a transação.

Para o domínio agropecuário, o software pode ser utilizado por pesquisadores de todas as áreas como ferramenta auxiliar às ferramentas tradicionais de análise estatística, assim como para testar modelos de classificação biológicos, econômicos ou sociais, ou como auxiliar na modelagem de sistemas especialistas.

Esta instrução visa introduzir pesquisadores interessados em mineração de dados na utilização do software para geração de regras e árvores de decisão C4.5. O algoritmo embutido neste software é utilizado intensivamente por outros produtos comerciais da categoria. C4.5 é o nome coletivo do software - o qual também é o nome do programa principal. O software constrói modelos de classificação indutivamente a partir dos dados analisados por meio de generalização de exemplos específicos.

O escopo desta instrução restringe-se aos aspectos de execução do software, e não aos aspectos computacionais e matemáticos que o embasam. Os algoritmos utilizados pelo software estão descritos detalhadamente no livro *C4.5 Programs for Machine Learning*, de autoria de Quinlan (1993), da Universi-

dade de Sidney. Maiores detalhes sobre o software e seus fundamentos podem, ser encontrados em <http://www.cse.unsw.edu.au/~quinlan/>, onde é possível, também, realizar download do sistema. Uma explanação didática dos algoritmos computacionais do software pode ser encontrada em <http://www.cs.uregina.ca/~dbd/cs831/cs831.html>.

2. O Sistema de Indução C4.5

O sistema C4.5 consiste de quatro programas principais:

1. o gerador de árvores de decisão ('c4.5'),
2. o gerador de regras ('c4.5rules'),
3. o interpretador de árvores de decisão ('consult'), e
4. o interpretador de regras ('consultr').

Nesta instrução aborda-se apenas os dois primeiros.

A estrutura de pastas do software (Release 8) é a seguinte²:

../R8/Data – contém alguns exemplos de aplicações:

hypo.names:	<i>dataset</i> de 2514 casos médicos sobre dados de tiróide, amplamente documentado, com bibliografia.
labor-neg.names:	<i>dataset</i> de 40 casos sobre acordos trabalhistas entre trabalhadores e a indústria canadense, com fonte.
soybean.names:	<i>dataset</i> de 683 casos sobre doenças da soja, amplamente documentado e com bibliografia.
vote.names:	<i>dataset</i> de 300 casos sobre tendência dos votos no congresso americano, com fonte.

../R8/Doc – contém páginas de manual no estilo Unix, nos seguintes arquivos³:

c4.5.1:	manual do gerador de árvores de decisão (<i>c4.5</i>)
c4.5rules.1:	manual do gerador de regras (<i>c4.5rules</i>)
consult.1:	manual do interpretador de árvores de decisão
consultr.1:	manual do interpretador de regras
verbose.1:	manual sobre os níveis de saída (<i>verbose</i>) do programa <i>c4.5</i>
verbrules.1:	manual sobre os níveis de saída do programa <i>c4.5rules</i>

../R8/Src – contém os fontes – em C - e binários do sistema.

² Na Embrapa Informática Agropecuária, o software está instalado na máquina calixto, no filesystem */usr/local*.

³ Consulte as dicas para a leitura dos manuais *online* no final desta Instrução.

3. Requerimentos-chave para a Utilização do Software

A seguir são descritos alguns requerimentos considerados importantes para a utilização do software. Estes requerimentos são explicados detalhadamente na seção 4.

Descrição atributo-valor: os dados a serem analisados devem estar em um arquivo texto. Toda a informação sobre um objeto ou *caso* deve ser expresso como uma coleção fixa de propriedades ou *atributos*. O valor de cada atributo pode ser discreto ou contínuo, mas os atributos que descrevem um caso não devem variar de um caso para outro, ou seja, a estrutura definida deve ser válida para todo o conjunto de dados.

Classes pré-definidas: as categorias para as quais os casos são classificados devem ser estabelecidas previamente. Na terminologia de aprendizado de máquina isto se chama *aprendizado supervisionado*, em contraste ao *aprendizado não supervisionado* no qual o grupos de casos são encontrados por análise.

Classes Discretas: este requerimento diz respeito às categorias para as quais os casos são alocados. As classes devem estar exatamente delineadas – um caso pertence ou não a uma determinada classe – e deve haver muito mais casos do que classes.

Dados suficientes: a generalização indutiva funciona identificando padrões em dados. Como a diferenciação de padrões usualmente depende de testes estatísticos de um tipo ou de outro, deve existir casos suficientes para permitir que esses testes sejam efetivos. Isto também evita que padrões identificados sejam meras coincidências. O montante de dados requeridos é afetado por fatores como o número de propriedades e classes, bem como pela complexidade do modelo de classificação; a medida que isto aumenta, mais dados são necessários para construir um modelo confiável. Um simples modelo pode algumas vezes ser identificado a partir de poucas dezenas de casos, mas um modelo de classificação detalhado usualmente requer centenas ou até milhares de casos.

Modelos de classificação lógica: os programas constróem somente classificadores que podem ser expressados como árvores de decisões ou conjuntos de regras. Esta forma, essencialmente restringe a descrição de uma classe para uma expressão lógica. Uma forma comum de modelo de classificação

que não satisfaz este requerimento é a *discriminante linear*, na qual os pesos das contribuições dos atributos são combinados aritmeticamente e comparados com um limite; as correspondentes descrições das classes são, então, aritméticas ao invés de lógicas.

4. Um Exemplo Ilustrativo

As linhas de comandos apresentadas a seguir pressupõem que o software esteja instalado abaixo de uma pasta - ou filesystem - chamada */usr/local*, e que o usuário esteja posicionado na pasta */usr/local/R8/Data*. Entretanto, o software é facilmente carregável através da URL <ftp://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz>. Após baixá-lo na estação, utilize os comandos a seguir para descompactar, desagrupar e compilar o software (é necessário ter instalado um compilador C):

```
tar xvzf c4.5r8.tar.gz
cd R8/Src
make all
```

Ou, se o tar do seu Unix não aceitar a opção 'z':

```
gunzip c4.5r8.tar.gz
tar xvf c4.5r8.tar
cd R8/Src
make all
```

Com este procedimento o software ficará instalado abaixo da pasta *R8*, no seu diretório *home*. Consulte no item 2 a estrutura de pastas do software.

Para demonstração será usado um exemplo ilustrativo que procura indicar quais condições de tempo são propícias ao jogo de golfe.

Os dados submetidos ao sistema devem estar em dois arquivos com o mesmo nome, porém com extensões diferentes. Um deve possuir a extensão *".names"* (Fig. 1), e outro, a extensão *".data"* (Fig. 2). No primeiro são descritas as classes, atributos e valores que compõem um caso individual; no segundo são colocados os dados de cada caso, um por linha. Os arquivos devem estar em formato texto Unix.

Um terceiro arquivo, também com o mesmo nome, mas com extensão *".test"* pode ser usado para validar os resultados obtidos na análise do arquivo de dados analisado. Caso um arquivo de teste esteja presente e a opção *-u* seja utilizada, o software tenta classificar os casos constantes neste arquivo, usando as conclusões obtidas no arquivo de dados *".data"*. É recomendável o uso de um *test-set* a fim de aumentar a confiabilidade das regras. Ele deve

ser elaborado com casos escolhidos aleatoriamente no conjunto total de casos. Durante a execução o programa *c4.5* gera dois outros arquivos, um para a árvore não podada (***.unpruned**) e outro para a árvore podada (***.tree**), os quais serão usados posteriormente pelo programa *c4.5rules*.

```
Jogar, Nao Jogar.

Condicoes do Dia: ensolarado, coberto,
chuvoso.
Temperatura: continuous.
Umidade: continuous.
Vento: verdadeiro, falso.
```

Fig. 1. Listagem do arquivo *golf.names*.

Note que na primeira linha da listagem da Fig. 1 estão descritas as classes utilizadas para classificar os dados (*Jogar* e *Nao Jogar*) na sintaxe:

```
classe1, classe2, ..., classen.
```

Após uma linha em branco, são definidos os atributos de um caso e definidos seus valores possíveis, obedecendo à sintaxe:

```
Nome do Atributo: valor1, valor2, ..., valorn.
ou
Nome do Atributo: continuous.
ou
Nome do Atributo: discrete n.
```

Notas:

- "Nome do Atributo" pode conter espaços, maiúsculas e minúsculas;
- o valor "continuous" pode ser usado para definir valores contínuos;
- é possível também utilizar o valor "discrete *n*" quando existem muitos valores discretos para definir. Neste caso o valor *n* deve ser o máximo número de valores diferentes possíveis para este atributo;
- é possível, também, usar o valor "ignore" para ignorar o atributo durante a análise. É útil para re-submeter análises, permitindo o isolamento de atributos sem a necessidade de reformatar o formato dos casos.

```
ensolarado, 85, 85, falso, Nao Jogar
ensolarado, 80, 90, verdadeiro, Nao Jogar
coberto, 83, 78, falso, Jogar
chuvoso, 70, 96, falso, Jogar
chuvoso, 68, 80, falso, Jogar
chuvoso, 65, 70, verdadeiro, Nao Jogar
coberto, 64, 65, verdadeiro, Jogar
ensolarado, 72, 95, verdadeiro, Nao Jogar
ensolarado, 69, 70, verdadeiro, Jogar
chuvoso, 75, 80, falso, Jogar
ensolarado, 75, 70, verdadeiro, Jogar
coberto, 72, 90, verdadeiro, Jogar
coberto, 81, 75, falso, Jogar
chuvoso, 71, 80, verdadeiro, Nao Jogar
```

Fig. 2. Listagem do arquivo *golf.data*.

Note que os atributos listados na Fig. 2 estão dispostos na mesma ordem em que foram declarados no arquivo *golf.names*. A classe atribuída ao caso, entretanto, é colocada por último na linha. Na ausência de algum valor (valor perdido) é possível utilizar o valor "?" para representá-lo.

Ex: coberto, ?, ?, verdadeiro, Jogar

Roteiro para Utilização:

- executar o programa *c4.5*, para gerar a árvore de decisão sobre os casos analisados, com o seguinte comando:

```
/usr/local/R8/Src/c4.5 -f golf
```

Após a execução, será produzida uma listagem conforme exibida na Fig. 3.

```
user@calixto[1] /usr/local/R8/Src/c4.5 -f golf
C4.5 [release 8] decision tree generator
-----
Thu Dec 27 14:19:04 2001

Read 14 cases (4 attributes) from golf.data

Decision Tree:

Condicoes do Dia = coberto: Jogar (4.0)
Condicoes do Dia = ensolarado:
|  Umidade <= 75 : Jogar (2.0)
|  Umidade > 75 : Nao Jogar (3.0)
Condicoes do Dia = chuvoso:
|  Vento = verdadeiro: Nao Jogar (2.0)
|  Vento = falso: Jogar (3.0)

Tree saved

Evaluation on training data (14 items):

  Before Pruning          After Pruning
  -----
  Size   Errors           Size   Errors   Estimate
-----
    8    0 ( 0.0%)      8    0 ( 0.0%)   (38.5%) <<
```

Fig. 3. Árvore de Decisão gerada pelo software C4.5.

A Fig. 3 pode ser vista como uma árvore de decisão, a qual pode ter:

- uma folha indicando uma classe, ou
- um nó de decisão que especifica algum teste para ser feito sobre um simples valor de atributo, com um galho e subárvores para cada possível saída do teste.

Os números, após as folhas, são explicados detalhadamente nos capítulos 3 e 4 do livro *C4.5: programs for machine learning* – ver referência - e são por demais extensos para serem apresentados neste documento. Entretanto eles podem ser apresentados, resumidamente, nas formas (N) ou (N/E). Na primeira forma N representa a soma fracional dos casos que pertencem a classe (folha). Na segunda, E representa o número de casos que não foram classificados pela classe. Estes números não podem ser entendidos como frequência.

Na seqüência da Fig. 3, as árvores - não podada e podada - são avaliadas novamente contra o conjunto de dados para testar a consistência das regras geradas. A primeira tabela ilustra a consistência da árvore não podada, em duas colunas:

1. **Size:** o tamanho da árvore não podada. Isto é, o número de nós que a compõe.

2. **Errors:** o número de erros de classificação e seu correspondente percentual de erros em relação ao número total de casos.

A Segunda tabela ilustra a consistência da árvore podada, em três colunas:

1. **Size:** o tamanho da árvore podada. Deve ser ou menor ou igual ao da árvore não podada, dependendo da extensão da poda realizada pelo C4.5.
2. **Errors:** o número de erros de classificação e seu correspondente percentual real de erro após a poda.
3. **Estimate:** o percentual estimado de erro da árvore após a poda, útil quando comparado com a percentagem real.

- b) executar o programa C4.5rules para gerar as regras - a partir da árvore de decisão produzida na etapa a) - sobre os casos analisados. Utilizar o seguinte comando:

```
/usr/local/R8/Src/c4.5rules -f golf
```

Após a execução, será produzida uma listagem conforme exibida na Fig. 4.

```

user@calixto[2] /usr/local/R8/Src/c4.5rules -f golf
C4.5 [release 8] rule generator Thu Dec 27 14:24:23 2001
-----
Read 14 cases (4 attributes) from golf
-----
Processing tree 0

Final rules from tree 0:

Rule 2:
  Condiçoes do Dia = coberto
  -> class Jogar [70.7%]

Rule 4:
  Condiçoes do Dia = chuvoso
  Vento = falso
  -> class Jogar [63.0%]

Rule 1:
  Condiçoes do Dia = ensolarado
  Umidade > 75
  -> class Nao Jogar [63.0%]

Rule 3:
  Condiçoes do Dia = chuvoso
  Vento = verdadeiro
  -> class Nao Jogar [50.0%]

Default class: Jogar

Evaluation on training data (14 items):

Rule      Size  Error   Used      Wrong      Advantage
-----
  2         1   29.3%   4         0 (0.0%)   0 (0|0)   Jogar
  4         2   37.0%   3         0 (0.0%)   0 (0|0)   Jogar
  1         2   37.0%   3         0 (0.0%)   3 (3|0)   Nao Jogar
  3         2   50.0%   2         0 (0.0%)   2 (2|0)   Nao Jogar

Tested 14, errors 0 (0.0%)  <<

      (a) (b)      <-classified as
      -----
          9          (a): class Jogar
          5          (b): class Nao Jogar

```

Fig. 4. Regras geradas pelo software c4.5rules.

A saída gerada pelo `c4.5rules` no nível default de saída, pode ser interpretado como segue.

Após os cabeçalhos que identificam o conjunto de dados, são listadas as regras geradas. Um conjunto de regras é gerado para cada árvore de decisão dada. Um conjunto de regras usualmente consiste de no mínimo uma regra, a qual é usada para classificar casos desconhecidos quando nenhuma outra regra se aplica (ex.: Jogar). Toda a regra é composta de atributos, valores e a resultante classificação, seguida pela percentagem que representa a acurácia da regra.

Ex.:

```
Se "Condicoes do Dia" = "ensolarado"
mas a "Umidade" > 75
então "Nao Jogue" (63,0%)
```

De acordo com o programa `c4.5rules`, esta regra é certa em 63% das vezes e, logo, tem uma margem de erro de 37%. Após a listagem, as regras são avaliadas contra o conjunto total de casos a fim de testar a validade de cada uma. A tabela de regras possui seis colunas:

1. **Rule:** O número atribuído para cada regra pelo programa `c4.5rules`.
2. **Size:** O tamanho da regra. Isto é, o número de antecedentes – ou testes - pelos quais ela é formada.
3. **Error:** A margem de erro da regra.
4. **Used:** O número de vezes em que a regra foi usada, desconsiderando sua correção, para classificar os casos.
5. **Wrong:** o número de vezes em que a classificação do caso não satisfaz a regra e o correspondente percentual.
6. **Advantage:** a diferença entre o número de vezes em que a regra foi usada corretamente e o número de vezes em que ela foi usada incorretamente. Representa uma medida de efetividade da regra. Por exemplo, suponhamos que uma regra tenha gerado os seguintes valores para a tabela:

Used	Wrong	Advantage
4	1 (25.0% = 1 wrong / 4 used)	2 (3 1)

Wrong = 1: o número de vezes em que a regra foi usada incorretamente.

Right = 3: o número de vezes em que a regra foi usada corretamente (4 total used - 1 wrong).

Advantage = 2: a diferença entre 3 certas e 1 errada.

A tabela também mostra o número total de classificações erradas e calcula o percentual de erro em relação ao número total de casos testados.

A última tabela da Fig. 4 é resultado da aplicação das regras elaboradas durante a análise contra as classes especificadas no conjunto original de casos. Na seqüência, o programa soma o número de classificações corretas e incorretas na tabela. Tanto as colunas quanto as linhas possuem o mesmo cabeçalho, mas existe uma distinção entre eles:

- as colunas da tabela são as classes originais dos casos;
- as linhas da tabela são as classes alocadas pelo software no processo de classificação;
- a célula onde existe uma intersecção entre linha e coluna pode conter um número ou não. Se a célula não contém um número, então não foi classificada nenhum caso sob aquela célula pela correspondente classe na coluna. De outra forma, o número representa o número de casos da classe da linha que foram classificados como membros da correspondente classe na coluna. Desclassificações podem ocorrer quando as classes da linha e da coluna de uma célula não conferem. Por exemplo, a última tabela da Fig. 4 mostra que 9 casos da classe conhecida "Jogar" foram corretamente classificadas, usando as regras geradas, como membros da classe "Jogar", 5 casos da classe "Não Jogar" foram corretamente classificados, usando as regras geradas, como membros da classe "Não Jogar".

A quantidade de informações na saída dos resultados depende do nível de saída (*verbose*) utilizado. Neste exemplo foi utilizado o nível de saída default. Consulte as páginas do manual online do software para maiores informações sobre os níveis possíveis.

5. Algumas Dicas de Uso

a) Uso da documentação:

- exemplos de aplicações:

vá para a pasta `../R8/Data`, utilize os comandos **ls** e **pg**, ou **more**, para visualizar os arquivos com extensão `.names` e `.data`:

```
pg soybean.names (Solaris)
```

```
ou
```

```
more soybean.names (Linux)
```

- manual *online*:

vá para a pasta `../R8/Doc` e use o comando:

```
nroff -man <nome-arq> | more
```

onde <nome-arq> pode ser:

```
c4.5.1 manual do gerador de árvores de decisão
c4.5rules.1 manual do gerador de regras
consult.1 manual do interpretador de árvores de decisão
consultr.1 manual do interpretador de regras
verbose.1 manual sobre os níveis de verbose do c4.5
verbrules.1 manual sobre os níveis de verbose do c4.5rules
```

b) Edição dos arquivos de casos:

- edite seus arquivos com um editor texto do Windows e transfira para seu diretório home na estação, via ftp. Lembre-se que o formato deve ser texto. A conversão para o formato Unix é realizada pelo próprio ftp;
- é aconselhável e preferível ter os dados em uma base que suporte operações usando SQL, ou outro mecanismo de filtragem, seleção e classificação, que permita a exportação para o formato texto, delimitado por vírgula. Isto vai facilitar a filtragem, transformações, conversões e reagrupamentos de valores de atributos ou de classes, possibilitando novas análises sobre os casos de forma mais rápida e eficiente.

c) saída:

- como na maioria das vezes a saída do software excede o número de linhas de um terminal, o usuário pode utilizar as seguintes facilidades do Unix:

```
/usr/local/R8/Src/c4.5 -f golf | more
faz paginação no terminal
```

```
/usr/local/R8/Src/c4.5 -f golf > analise1.txt
redireciona a saída para o arquivo analise1.txt
```

```
/usr/local/R8/Src/c4.5 -f golf | mailx
```

```
usuario1
```

envia o resultado via e-mail para o usuário *usuario1* da rede local

```
/usr/local/R8/Src/c4.5 -f golf | mailx -
sAnalise_1 usuario1
```

envia o resultado via e-mail, com o assunto *Analise_1*, para o usuário *usuario1* da rede local (endereços remotos também são permitidos)

6. Referências Bibliográficas

FEIGENBAUM, E. A.; MCCORDUCK, P.; NII, H. P. **The rise of the expert company**: how visionary business are using intelligent computers to achieve higher productivity and profits. [S.l.]: Times Books, 1998.

HAMILTON, H.; GURAK, E.; FINDLATER, L.; OLIVE, W. **Machine learning/decision trees - C4.5 tutorial**. Disponível em: <http://www.cbi.msstate.edu/faculty/dvance/ml/C4_5%20Tutorial.htm>. Acesso em: 03 jan. 2002.

INFORMATION DISCOVERY. **A characterization of data mining technologies and processes**. Disponível em: <<http://www.datamining.com/dm-tech.htm>>. Acesso em: 03 jan. 2002.

LE BRET, C. **Have you heard about data mining?** Disponível em: <<http://www.tribunes.com/tribune/art97/lebe.htm>>. Acesso em: 19 dez. 2001.

QUINLAN, J. R. **C4.5: programs for machine learning**. São Mateo: Morgan Kauffman, 1993.

QUINLAN, R. **Ross Quinlan - AI Group, CSE**: [home page]. Disponível em: <<http://www.cse.unsw.edu.au/~quinlan/>>. Acesso em: 19 dez. 2001.

Instruções Técnicas, 7

MINISTÉRIO DA AGRICULTURA,
PECUÁRIA E ABASTECIMENTO



Embrapa Informática Agropecuária Área de Comunicação e Negócios

Av. Dr. André Tosello s/nº
Cidade Universitária - "Zeferino Vaz"
Barão Geraldo - Caixa Postal 6041
13083-970 - Campinas, SP
Telefone/Fax: (19) 3789-5743
E-mail: sac@cnptia.embrapa.br

1ª edição

© Embrapa 2001

Comitê de Publicações

Presidente: Francisco Xavier Hemerly

Membros efetivos: Amarindo Fausto Soares, Ivanilde Dispatto, Marcia Izabel Fugisawa Souza, José Ruy Porto de Carvalho, Suzilei Almeida Carneiro

Suplentes: Fábio Cesar da Silva, João Francisco Gonçalves Antunes, Luciana Alvim Santos Romani, Maria Angélica de Andrade Leite, Moacir Pedroso Júnior

Expediente

Supervisor editorial: Ivanilde Dispatto

Normalização bibliográfica: Marcia Izabel Fugisawa Souza

Capa: Intermídia Publicações Científicas

Editoração Eletrônica: Intermídia Publicações Científicas