UM SISTEMA DE PREDIÇÃO USANDO ÁRVORES DE MODELOS

Paulo Sérgio de Souza Coelho
Faculdades Ibmec. Avenida Rio Branco, 108, 2º andar, Rio de Janeiro – RJ.
psergio@ibmecrj.br

Gerson Lachtermacher
FCE/UERJ e Faculdades Ibmec – RJ.
glachter@uerj.br

Nelson Francisco Favilla Ebecken
COPPE / UFRJ - Cx. Postal 68506, Rio de Janeiro – RJ.
nelson@ntt.ufrj.br

Resumo

Indução de Modelos é uma das atividades de Mineração de Dados onde uma base de dados é utilizada para treinar um modelo que represente o conhecimento contido naquela base. Os modelos de predição modelam variáveis dependentes numéricas. As Árvores de Modelos são específicos para predição e usam uma estrutura computacional de árvore para armazenar um modelo de regressão em cada folha da árvore. A estrutura da árvore determina uma partição do conjunto de dados utilizado para induzir o modelo. Esta partição determina diversos subconjuntos de dados utilizados para induzir cada modelo de regressão. Este trabalho apresenta um novo algoritmo de indução de árvores de modelos chamado M5.5'. Este algoritmo foi desenvolvido com base na teoria estatística de estimação de regressão baseada no Método de Mínimos Quadrados aliada a diversas técnicas conhecidas de Mineração de Dados. A implementação foi feita em linguagem Java, dentro da suíte de código aberto Weka (Waikato University). O sistema obtido foi comparado com sistemas similares que possuem posição de destaque no cenário atual usando bases de dados já conhecidas academicamente. Os principais resultados são mostrados.

Palavras Chave: Mineração de Dados, Modelos de Indução, Árvores de Modelos, Modelos de Regressão.

Abstract

Model Induction is a Data Mining activity that uses a data base to train a model which represents the knowledge in that base. The prediction models are specific to numeric dependent variables. The Model Trees are prediction models using a computational structure of a tree to store a lot of regression models in the tree nodes. The tree structure defines a partition of the data set used to induct the model, and this partition defines the data subset used to induct each regression model. This work shows the model tree induction algorithm called M5.5'. This algorithm was developed using the statistic theory to regression model estimation based on Minimum Squares Method, allied to known Data Mining techniques. A Java implementation was made, inside the Weka Data Mining suite open code. This system was compared with similar systems having special positions in current scene using academic known data bases, and the main results are shown.

Keywords: Data Mining, Induction Models, Model Trees, Regression Models.

1. INTRODUÇÃO

A área de conhecimento composta por diversas ciências, dentre as quais, Estatística, Matemática e Computação em torno do objeto de estudo que é a manipulação de grandes volumes de dados é conhecida por Mineração de dados. Mais genericamente, podemos considerar a área chamada de Descoberta de Conhecimento em Banco de Dados – *Knowledge Discovery in Databases*, também indicado como KDD, da qual a mineração de dados é uma das sub-áreas.

Neste texto está descrito um novo algoritmo de Indução de Árvores de Modelos (M5.5´) que foi desenvolvido em linguagem Java e implementado dentro de uma suíte de código aberto denominada Weka (Cf. Weka Project, 2001). A implementação foi feita com o intuito de realizar os primeiros testes do algoritmo desenvolvido (testar as heurísticas introduzidas), sem intenção de que o código obtido estivesse em sua versão final. Após esta fase, uma revisão e uma adaptação completa nos códigos serão desenvolvidas, respeitando os preceitos da Engenharia de Software e de desenvolvimento integrado para projetos de código aberto.

Árvore de Modelos é uma teoria relativamente nova (em torno de 20 anos de existência desde seus primórdios – Cf. Breiman, 1984), e apresenta uma grande aceitação como ferramenta de predição, considerando os principais critérios de comparação de modelos, apresentando simultaneamente níveis aceitáveis de precisão, interpretabilidade, robustez, velocidade e escalabilidade.

De uma maneira geral, a área de predição é uma das áreas dentro daquelas de *Data Mining* que tem sido pouco desenvolvida. Curiosamente, é uma área que tem forte aplicabilidade prática. Esta dicotomia sugere que todos os estudos feitos nesta subárea serão bem vindos e aproveitados pela comunidade academia.

Este artigo está dividido em mais 5 seções além desta introdução. A Seção 2 dá uma visão panorâmica do desenvolvimento da Tecnologia da Informação e o surgimento da Mineração de Dados. A Seção 3 faz uma Revisão Bibliográfica dos principais modelos de classificação e predição, focando nos modelos de predição baseados em árvore e mais especialmente nas chamadas árvores de modelos, fundamento deste trabalho e em especial no modelo M5´ ponto de partida deste novo algoritmo. A Seção 4 descreve o algoritmo desenvolvido, o M5.5', mostrando suas diferenças do M5´. Alguns testes, de desempenho do novo algoritmo, são mostrados na Seção 5 e a conclusão do artigo e sugestões para novos desenvolvimentos são feitas na Seção 6 do trabalho.

2. DESENVOLVIMENTO DA TI E MINERAÇÃO DE DADOS

Cientistas históricos determinam que o desenvolvimento da civilização humana se dá em função da posse da comunicação (Cf. Habermas, 2002), pois a conquista da comunicação permite que os aprendizados e as conquistas de uma geração sejam transmitidos para as gerações seguintes.

O armazenamento dos dados feito eletronicamente tem inicialmente apenas para registro dos acontecimentos e futuras consultas. O armazenamento de diferentes informações permite que as consultas extrapolem as estatísticas das operações. Além disso, e diante do crescimento do volume dos bancos de dados, torna-se necessário que análises mais profundas das informações armazenadas sejam realizadas.

Estes modelos estão relacionados às medidas de similaridade, co-ocorrência (correlação), relações de dependência, de ordem, etc. Estes modelos apresentam forte viés matemático e estatístico, e é a área que está sendo chamada de **Mineração de Dados** (***Data Mining***), conforme Han & Kamber (2000). Para estimar os parâmetros destes modelos, as

ferramentas de Mineração de Dados utilizam massas de dados vindas de qualquer origem e a partir de dados multidimensionais, temporais ou não, com variáveis de qualquer natureza, o que inclui as estruturas de texto, gráficas, de áudio e até vídeo.

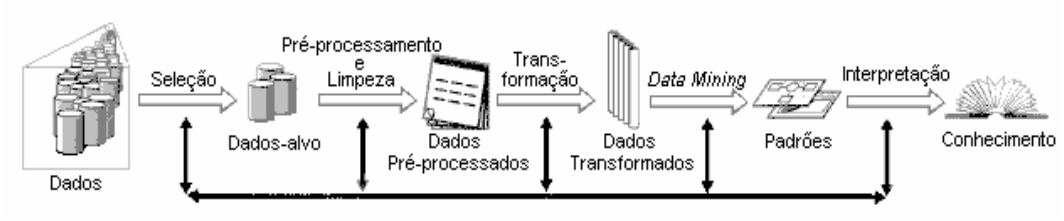

Figura 1: Etapas do KDD.

O modelo pode ser expresso através de um conjunto de regras, de equações ou ainda estruturas gráficas como grafos, redes ou árvores. O processo total, que vai da localização e extração dos dados até a compreensão do conhecimento modelado, passando pela etapa de construção do modelo de conhecimento – Mineração dos dados, é chamado de ***Knowledge Discovery in Databases* (*KDD*)**, ou Descoberta de Conhecimento em Banco de Dados (Cf. Berry, 1997, Berson, 1999, Han & Kamber, 2001 e Weiss, 1998).

A Figura 1 apresenta um esquema com as etapas contidas no processo de KDD. A Mineração de Dados é uma etapa deste processo total, chamado de KDD.

2.1. SUÍTE WEKA

O Weka (Cf. Weka Project, 2001) é um projeto de desenvolvimento de ferramentas de *Machine Learning* e *Data Mining*, da Universidade de Waikato, na Nova Zelândia. O projeto gerou um software, que é uma suíte que inclui diversas técnicas e algoritmos para preparação, mineração de dados e ferramentas para manipulação dos modelos obtidos. Manuais podem ser obtidos diretamente no site do sistema, e Witten (2000) é uma importante referência.

O software é desenvolvido de acordo com o projeto GNU (GNU, 2005), e seu código em Java está disponível, podendo ser compilado em qualquer plataforma. O projeto não está fechado e é possível realizar alterações no código e submeter estas alterações para uma nova versão oficial do software.

A interface é bastante peculiar, comum entre sistemas desenvolvidos nesta linguagem. O software (versão 3.4.4) lê dados armazenados em um formato específico chamado arff, ou em formato csv (Excel). Para avaliar a precisão do modelo obtido cinco estatísticas são estimadas baseadas nas técnicas de amostra de teste ou validação cruzada (maiores detalhes em Breiman, 1984).

A facilidade de acesso ao Weka, inclusive aos códigos que geram o software, tem tornado a ferramenta extremamente popular. Uma rápida consulta à internet mostra diversos cursos de graduação, pós-graduação, técnicos e de extensão em geral que estão utilizando a ferramenta, como na *New York University*, e a *Imperial College London*.

O algoritmo ora proposto foi desenvolvido a partir do código do algoritmo de indução de árvores conhecido como M5´ disponível na suíte Weka. Este código foi implementado a partir do algoritmo M5 (Quilan, 1992) disponível apenas comercialmente.

3. MODELOS DE CLASSIFICAÇÃO E PREDIÇÃO

De uma maneira geral, os algoritmos de indução de modelos de classificação ou predição buscam modelar uma determinada variável chamada de **variável de resposta** ou **variável dependente** através da relação desta com um conjunto de outras variáveis chamadas de **variáveis explicativas** ou **variáveis independentes**.

A indução de modelos é um problema antigo, e muitas técnicas foram desenvolvidas na busca de soluções viáveis. Por exemplo, as Redes Neurais são desenvolvidas desde 1943 (Cf. McCulloch e Pitts, 1943), e a teoria estatística de decisão surge a partir dos anos 1940 (Cf. Pearl, 1988), dentre outras. As técnicas que foram desenvolvidas (e que ainda continuam sendo) dão origem a um grande conjunto de ferramentas computacionais, que são de grande utilidade para aqueles que precisam analisar cenários incertos. Estas ferramentas possuem perfil comercial ou acadêmico.

No universo das técnicas de indução de modelos destacam-se os modelos baseados em árvores, tanto por sua eficiência quanto por sua eficácia. Estes modelos tornaram-se bastante populares e têm sido largamente desenvolvidos nos últimos 20 anos, desde os trabalhos de Breiman (1984). Diferentemente dos **Modelos de Regressão**, que é uma metodologia consolidada, muitas metodologias diferentes foram desenvolvidas usando estruturas de árvores.

É importante definir que a técnica de Indução de Modelos pode realizar duas ações que para o nosso propósito precisam ser distinguidas pela natureza da variável dependente (variável de resposta), que são as ações de classificação ou predição. Dizemos que o modelo induzido é um Modelo de Classificação quando a variável dependente é uma **variável categórica**, e que é um Modelo de Predição quando a variável dependente é uma **variável numérica**.

As **variáveis numéricas** assumem valores em um conjunto numérico, que por natureza é ordenado e é potencialmente infinito. Por outro lado, quando os valores que a variável pode assumir estão em um conjunto finito sem ordenação natural ou suficientemente pequeno consideramos que é uma **variável categórica**. O termo categórico está relacionado a categorias ou classes que possivelmente não devem ser representadas numericamente pois não é possível explicar qualquer posicionamento em um eixo. Categorias podem não admitir ordenação como, por exemplo, cor, bairro ou dia da semana.

Alguns autores preferem chamar os modelos de predição de modelos de regressão. Outros (Cf. Han & Kamber, 2000) discordam desta definição. A razão para isto é que o modelo de regressão é apenas uma das conhecidas técnicas de modelos de predição (baseado na **Teoria de Regressão Estatística)**. Esta nomenclatura pode provocar certa confusão, que só pode ser evitada em situações específicas ou quando alguma descrição acompanhar o conceito, e deve ser evitada. Assim, o termo “Modelo de Regressão” neste texto, refere-se à técnica estatística específica de regressão, podendo ser dito que o Modelo de Regressão é um Modelo de Predição.

O objetivo deste texto não nos permite maiores comentários sobre ferramentas de classificação, que são possivelmente mais diversas do que as ferramentas de predição. Maiores detalhes sobre as técnicas de classificação, inclusive uma comparação de ferramentas pode ser vista em Coelho & Ebecken (2002).

3.1. MODELOS DE PREDIÇÃO

De uma maneira simplificada, podemos definir que os modelos de predição tentam modelar o comportamento da variável dependente fazendo uso de um conjunto de casos onde os valores da variável dependente são conhecidos. Este conjunto de casos costuma ser chamado de **conjunto de treinamento** do modelo. Quanto maior o conjunto de treinamento, mais aleatória a sua composição e mais extenso o seu espectro, melhores chances temos de obter um melhor modelo, pois este depende, de uma maneira geral, da representatividade da amostra. O uso do conjunto de treinamento para estimar os parâmetros do modelo é comumente chamado de treinamento do modelo.

Os métodos de predição diferenciam-se a partir da estrutura do modelo. Estruturas

diferentes são treinadas de maneira diferente, e a partir daí cada método possui suas próprias peculiaridades. Os modelos de predição mais conhecidos são as regressões. Estruturas baseadas em árvores são importantes, não só para predição como para classificação também, assim como os modelos baseados em Redes Neurais (Cf. Haykin, 2000 e Lachtermacher 1992). Existem também os modelos baseados em similaridade de exemplos conhecidos. São métodos derivados do algoritmo de classificação conhecido como *KNN – K-Nearest Neighbourhood* (*K* vizinhos mais próximos) (Cf. Han & Kamber, 2001).

Quando os tomadores de decisão estão diante de cenários, notadamente o planejamento, que exigem a predição de valores desconhecidos, e quando esta predição é possível, entra em ação alguma ferramenta computacional. Questões relativas à qualidade técnica do modelo preditor (precisão, velocidade, escalabilidade, robustez e interpretabilidade) são misturadas com questões de interface ao usuário (que muitas vezes desconhece os detalhes técnicos envolvidos nos modelos) e preço no processo de seleção de qual ferramenta será aplicada.

3.2. ÁRVORES DE PREDIÇÃO

Uma árvore, no sentido computacional, é uma estrutura de representação gráfica de objetos (dados, condições, informações, etc.). Uma definição formal e completa é fornecida por Szwarcfiter (1994). Existem diversas ferramentas computacionais para realizar atividades de predição (e classificação) utilizando estruturas de árvore, ou estruturas tão similares que possam ser expressas em forma de árvores. Estas ferramentas apresentam metodologias diferentes para detalhes dos algoritmos envolvidos, mas a maneira que as estruturas de árvores são usadas reúne algumas características que são as mesmas para todas as metodologias.

Um nó interno é sempre um teste sobre um conjunto de atributos e uma folha representa sempre um valor para a variável dependente. O uso do modelo já induzido equivale a tentar estabelecer o valor da variável dependente para um determinado caso. Este caso é primeiramente testado na raiz, e este teste determina qual sub-árvore o nó deve seguir. Se esta sub-árvore possuir apenas um nó, então este nó é uma folha, e será obtido o valor desejado. Se esta sub-árvore possuir mais de um nó, então teremos uma raiz para a sub-árvore, que será um novo teste. Como a árvore é um conjunto finito de nós, temos que este processo resulta em uma folha e, portanto, num valor para a variável dependente.

Os processos de construção dos modelos baseados em árvores também apresentam características comuns para as mais diferentes metodologias de predição. O procedimento é sempre recursivo. O passo recursivo realiza uma de duas tarefas: 1) cria um nó interno – um teste que realiza a divisão do conjunto de registros (casos) usando alguma característica dos atributos, ou 2) cria um nó folha – determina um valor para a variável dependente (ou um modelo, como veremos posteriormente) – quando uma condição de parada é observada. As condições de parada (parada do crescimento da árvore) mais comumente utilizadas são duas: quando um limite para a função objetivo é alcançado ou quando o conjunto de casos é menor que um limite pré-definido.

Deve-se considerar que existem dois tipos de modelos diferentes. Esta diferença fica muito evidente na estrutura da árvore, principalmente quando se observa o que está armazenado nas folhas da árvore que já foi induzida. As folhas de uma **Árvore de Regressão** apresentam valores para a variável dependente, e/ou possivelmente, valores para algumas estatísticas do conjunto de valores da variável dependente que caíram naquele nó durante a fase de indução (treinamento) do modelo. As Árvores de Regressão foram originalmente propostas por Breiman (1984). As folhas de uma **Árvore de Modelos** apresentam modelos para que os valores da variável dependente sejam obtidos. Estes modelos podem ser modelos

de regressão estimados através do conjunto de casos que alcança a folha, como foi originalmente proposto por Quinlan (1992).

Grosso modo, esta diferenciação é bem aceita. Na verdade, a estrutura da Árvore de Modelos proposta por Quinlan (1992) é bem mais complexa do que a estrutura de uma Árvore de Regressão, pois apresentam algumas características que ainda não foram relatadas. Além disso, as diferenças entre as estruturas obrigam que as metodologias sejam mais específicas, conforme veremos nos capítulos subseqüentes.

Os trabalhos de Breiman (1984) representam um importante ponto de referência para o uso de estruturas de árvores para indução de modelos. É, certamente, o primeiro trabalho relevante nesta linha. Posteriormente, uma ferramenta computacional foi desenvolvida com base nas idéias presentes no livro, e comercialmente leva o nome de CART (*Classification And Regression Trees*). Esta ferramenta é atualmente muito bem aceita comercialmente, inclusive ganhou a KDD Cup (importante evento anual realizado por um grupo de pesquisadores de *Data Mining*) em 2000 (Cf. CART, 2005). Atualmente, o termo CART designa o livro original, as metodologias de árvores de classificação e de regressão, e o software.

3.3. ÁRVORES DE MODELOS

Historicamente Quinlan (1986) foi o primeiro texto em que se usa o termo Árvore de Decisão para designar um modelo de aprendizado de máquina (neste caso modelos de classificação), chamado de ID3. Este trabalho é posteriormente desenvolvido gerando o conhecidíssimo C4.5 no texto Quinlan (1993). Este livro é um texto bem avançado sobre o problema de classificação utilizando árvores. O texto inclui o problema de geração de regras, agrupamento de valores de atributos e interação com outros modelos de classificação. Uma cópia dos arquivos fontes para compilação, em ambiente UNIX, do software foram disponibilizados juntamente com o livro e atualmente estão na *Home Page* do autor, (Cf. Quinlan, 2005).

Quinlan (1992) define importantes alterações na estrutura original de Breiman (1984) para realizar predição utilizando árvores. O sistema é chamado de M5, e está fundamentado nos trabalhos iniciais de Breiman. A estrutura criada foi denominada *Model Trees*, ou Árvores de Modelos. Este nome foi utilizado porque as árvores representam modelos lineares por partes. Todos os nós da árvore apresentam equações lineares. Esta é uma grande diferença para o CART (Cf. Breiman, 1984), que apresenta valores fixos, e apenas nas folhas.

O sistema completo apresenta as etapas de simplificação do modelo, poda da árvore, baseados no CART. A contribuição é a técnica de suavização das descontinuidades dos modelos. Esta característica tornou-se necessária para evitar que dois exemplos muito semelhantes (tipicamente com todos os atributos iguais exceto um), apresentem saída muito diferente.

Depois que a árvore foi completamente desenvolvida, uma equação de regressão é estimada para cada nó interior. A equação é estimada utilizando os procedimentos padrão para regressão (mínimos quadrados), usando apenas os atributos que foram utilizados como teste na sub-árvore abaixo deste nó (por isso não existem modelos nas folhas). Segundo Quinlan (1992), esta medida é para assegurar que, a equação no nó que é a raiz da sub-árvore e a própria sub-árvore, possuam a mesma informação, pois a precisão da equação de regressão será comparada com a precisão da sub-árvore. Depois que a equação de regressão é estimada, ela pode ser simplificada, eliminando parâmetros para minimizar o seu erro estimado. A estimativa de erro é feita a partir da média dos resíduos da regressão. Finalmente, o uso dos modelos de regressão tornou necessário tratar problemas de descontinuidade, que surgem quando dois casos muito semelhantes (tipicamente casos possuindo valores iguais para todos

atributos exceto um) caem em diferentes folhas. É importante garantir que os resultados não sejam tão diferentes. É aceitação geral o fato que o uso deste procedimento de suavização aumenta substancialmente a precisão da predição (Cf. Quinlan, 1992 e Wang & Witten, 1997).

Outros algoritmos,para indução de árvores de modelos estão disponíveis na literatura, como o Retis, descrito em Karalic & Cestnik (1991). A idéia básica deste sistema é o uso de uma técnica que chamada de *m-distribution estimate*, baseada em uma abordagem Bayesiana. Esta abordagem Bayesiana não está relacionada com o famoso Teorema de Bayes, mas apenas na idéia de que para estimar o valor de um determinado parâmetro parte-se de um valor inicial (*prior*) que é alterado em função de alguns experimentos, chegando ao valor final (*posteriori*). O algoritmo CORE, que está descrito em Robnik-Sikonja (1997), é um sistema que induz uma Árvore de Modelos que pode possuir vários tipos de modelos nas suas folhas. O sistema foi criado durante os trabalhos de pesquisa para o título de mestrado de Robnik-Sikonja e, pelo que está relatado. Árvores de Regressões Locais (*Local Regression Trees*) são o resultado da integração de árvores de regressão com modelagem local, profundamente descritos em Torgo (1999). A questão central nos modelos locais é a noção de similaridade, que permite isolar os dados em subgrupos – chamaremos vizinhanças – para estimação das regressões. Esta similaridade é determinada considerando o conjunto de treinamento como um espaço vetorial e usando uma métrica particular. Diferentes métricas de distância podem ser usadas. Estes algoritmos são, via de regra, menos utilizados.

3.4. ALGORÍTMO M5'

O algoritmo de Árvores de Modelos M5' foi apresentado à comunidade acadêmica em 1997 (Cf. Wang & Witten, 1997). O seu desenvolvimento e publicação se deram como conseqüência da implementação do algoritmo M5 de Quinlan dentro da suíte Weka. Por ocasião da implementação do algoritmo M5 proposto por Quinlan (1992) pela equipe de desenvolvimento do Weka, percebeu-se que a descrição do algoritmo era insuficiente, sem descrições de detalhes fundamentais para a transcrição do algoritmo completo para uma linguagem computacional. Diante disto, Wang & Witten (1997) buscaram diretrizes mais substanciais para a implementação do M5. Algumas soluções que estavam ocultas nos trabalhos originais de Quinlan foram desenvolvidas (e estão descritas), e algumas outras foram obtidas a partir da metodologia do CART.

É possível estudar o algoritmo M5' além da tradicional e comum leitura das publicações realizadas pelos seus autores e desenvolvedores. Como o código que implementa o algoritmo dentro da suíte Weka está disponível, é possível obter informações sobre o algoritmo. Curiosamente, acontece que a descrição oficial disponível na literatura inclui detalhes que não estão implementados. Por outro lado, existem alguns procedimentos implementados que não estão descritos na literatura. A lista completa destas discrepâncias pode ser vista em Coelho (2005).

O algoritmo está dividido em dois passos recursivos. O primeiro passo é a construção da estrutura da árvore, e o segundo passo é a simplificação da estrutura anterior. Os passos estão descritos a seguir:

1. **Crescimento da Árvore:**
   - **Pré-seleção dos atributos para o Modelo de Regressão em cada nó;**
2. **Poda da Árvore:**
   - **Estimação, Simplificação e Suavização dos Modelos de Regressão;**

As heurísticas deste algoritmo estão descritas a seguir. O primeiro passo envolve o critério de parada do crescimento e a metodologia de busca do melhor atributo para realizar a divisão em cada nó interno da árvore. É feita uma pré-seleção dos atributos, de modo que os modelos induzidos em cada nó utilizem apenas os atributos que foram utilizados como

divisão em algum nó abaixo do nó corrente (esta pré-seleção é feita na hora que a pilha de recursão está sendo desfeita).

O segundo passo (poda da árvore) usa uma função ponderada de erro, que compara o erro total do modelo de regressão em um nó com o erro total da sub-árvore que está abaixo deste nó. Depois, os modelos são simplificados e suavizados, conforme descrito abaixo.

3.4.1. Critério de Parada

Existem dois critérios de parada no procedimento de crescimento da árvore. O primeiro critério é em função do tamanho do conjunto de casos que alcança o nó, e este tamanho pode ser controlado a partir da interface do sistema. Este critério é colocado apenas por questão de segurança de que o procedimento recursivo é finito sem inconsistência lógica.

O segundo critério de parada é em função da dispersão da variável dependente. O algoritmo pára o processo recursivo de divisão quando a sub-amostra apresenta dispersão da variável dependente menor do que 5% da dispersão da amostra inicial. Esta dispersão é medida em termos do desvio padrão.

3.4.2. Escolha do teste de divisão

Durante o crescimento da árvore, o algoritmo escolhe o teste que dividirá o conjunto de registros definindo o atributo a ser usado para o teste e o nível do atributo, ou seja, o limite que define os dois subconjuntos. Esta busca é feita exaustivamente, considerando todos os valores possíveis para os atributos. Usa-se uma função que mede a qualidade de cada possível divisão, e a que apresenta maior valor é escolhido. A expressão que representa esta função é chamada de Redução da Dispersão, e é dada por:

$$SDR = \frac{m}{n}\beta\left[sd(t) - \sum_i \frac{n_i}{n} sd(t_i) \right]$$

onde $sd(t) = \sqrt[5]{\sigma_y^2}$ mede a dispersão da variável de resposta do conjunto avaliado, n_i é a cardinalidade do subconjunto *i* e *n* é a cardinalidade do superconjunto, que é igual a soma das cardinalidades dos subconjuntos, que são sempre dois. Os parâmetros *m* e β são descritos originalmente para controlar valores faltantes e atributos categóricos, respectivamente, mas não são efetivamente utilizados.

O sistema faz uso de uma medida de *off-set* para o procedimento de busca do ponto de divisão, de modo que os dois primeiros e os dois últimos decis não são examinados. Como espera-se que o ponto ótimo de divisão esteja próximo da mediana do conjunto, esta é uma medida que traz o benefício da escalabilidade e velocidade do algoritmo, com pequenas perdas.

3.4.3. Função Ponderada de Erro

É utilizada uma função de erro absoluto dos modelos, e esta função é ponderada por um fator. Esta ponderação é feita para compensar o erro em função da quantidade de atributos, de maneira que modelos de regressão com menos atributos possam, dependendo do nível de contribuição dos atributos removidos, apresentar estimativa de erro menor. Esta função de erro tem a forma:

$$\frac{n+v}{n-v} \times \frac{\sum |y_i - \hat{y}_i|}{n}$$

A ponderação também serve como uma compensação ao fato da medida de erro ser uma subestimação do erro que se obtêm quando amostras não utilizadas no treinamento forem submetidas ao modelo para predição.

3.4.4. Simplificação dos Modelos

O uso de algum método de simplificação dos modelos está diretamente relacionado com o ganho no critério de interpretabilidade do modelo. Estes métodos podem ser particularmente importantes quando as heurísticas que pré-selecionam os atributos a serem considerados como variáveis independentes nos modelos de regressão forem comprovadamente falhas.

Depois que o modelo de regressão foi estimado, alguns atributos são selecionados para eliminação. É definida uma estatística chamada de *Akaike*, usada para comparar o modelo completo, com todos os atributos, com o modelo onde foi retirado um atributo – aquele com menos coeficiente padronizado:

$$Akaike = \frac{SSE_C}{SSE_R}(n-k)+2g$$

sendo n a quantidade de casos (registros) usado para estimar ambas regressões. SSE_C refere-se ao somatório dos quadrados dos resíduos da regressão completa, com k atributos, e SSE_R é o mesmo somatório, só que da regressão reduzida, com g atributos, sendo portanto $g < k$.

A estatística Akaike é baseada na Teoria da Informação, e é comumente chamada de AIC (Akaike Information Criterion). Existem várias expressões diferentes para estabelecer o valor da estatística, mas de maneira geral a magnitude da estatística está inversamente relacionado com a qualidade do modelo, ou seja, escolhe-se o modelo que apresenta menor AIC (Cf. Hair et. al, 2003 e Motulsky & Christopoulos, 2003).

O procedimento busca o atributo i que apresenta menor coeficiente padronizado. É estimada uma regressão com todos os atributos exceto este, um conjunto com g atributos (neste primeiro passo, $g = k-1$). Calcula-se a estatística Akaike, como definida anteriormente, e esta estatística é comparada com a estatística *Akaike* anterior, que no primeiro passo é calculada supondo $SSE_C = SSE_R$. Se A estatística *Akaike* for menor do que a estatística *Akaike* anterior, o atributo é eliminado, a estatística *Akaike* calculada passa a ser considerada a estatística *Akaike* anterior, e o procedimento repete a busca do atributo com menor coeficiente padronizado e cálculo da nova estatística *Akaike*. Para efeito de cálculo, SSE_C e k permanecem os mesmos. O procedimento só para quando o atributo que apresenta menor coeficiente padronizado não é eliminado.

3.4.5. Suavização dos Coeficientes

O uso dos modelos de regressão tornou necessário tratar problemas de descontinuidade, que surgem quando dois casos muito semelhantes (tipicamente casos possuindo valores iguais para todos atributos exceto um) caem em diferentes folhas. É importante garantir que os resultados não sejam tão diferentes.

Este é o último processo a ser considerado. Esta suavização pode ser interpretada como uma média ponderada de várias predições para o mesmo caso. Cada nó da árvore apresenta um teste sobre os valores dos atributos do caso, e estes testes determinam o caminho que o caso fará até uma folha da árvore. Ao longo deste caminho, cada nó, inclusive a folha, possui um modelo de regressão. As predições de cada um destes modelos são ponderadas, de maneira que casos semelhantes (registros próximos) possuam valores de predição não muito diferentes.

A técnica de suavização prediz o valor de um caso realizando uma soma ponderada dos valores preditos por todos os modelos que estão armazenados em cada um dos nós que estão no caminho desde a raiz até a folha que o caso alcança. Esta soma ponderada é obtida enviando valores de um nó para seu pai, seqüencialmente desde a folha até a raiz. O primeiro valor é calculado na folha, através do modelo lá armazenado. O valor predito pelo nó S (e que

será enviado para seu pai) é calculado usando o valor S_i, predito pelo filho de S onde o caso caiu, usando a equação:

$$PV(S) = \frac{n_i PV(S_i) + kM(S)}{n_i + k}$$

onde PV é o valor predito pelo nó, M é o valor de resposta estabelecido pelo modelo linear no nó em questão, n_i é o número de casos que caem no nó S_i durante a fase de treinamento, e k é uma constante de suavização, definida com valor fixo e igual a 15.

4. M5.5'

Os algoritmos de indução de árvores de modelos, incluindo o M5´, apresentam alguns processos não otimizados ou não devidamente testados. Os argumentos para revisar estes algoritmos estão fundamentados nas Teorias de Estimação de Modelos de Regressão, principalmente o Método dos Mínimos Quadrados (maiores descrições desta técnica podem ser vistas em Neter, 1996 e Draper, 1998). Vemos grandes lacunas entre tais teorias e o que está praticado em especial no algoritmo M5'. Neste sentido é que foi desenvolvido e implementado o algoritmo de indução de árvores de modelos chamado de M5.5' que teve como ponto de partida o algoritmo M5' implementado na suíte Weka. Estão descritas aqui apenas as modificações feitas no algoritmo original.

A pré-seleção dos atributos só é utilizada quando o conjunto de dados tem muitas variáveis, que é a situação em que o sistema pode se tornar extremamente complexo. Além disso, é possível controlar se os atributos pré-selecionados são apenas os utilizados abaixo ou também aqueles que foram utilizados em algum nó acima do nó atual.

4.1. CRITÉRIO DE PARADA

São considerados também dois critérios de parada no procedimento de crescimento da árvore. O primeiro critério também é em função do tamanho do conjunto de casos que alcança o nó, ainda por motivo de segurança, mas este tamanho pode ser controlado a partir da interface do sistema.

O outro critério de parada é em função do erro do modelo de regressão do nó. Este erro é estimado usando o subconjunto de casos que alcança o nó através do $SSE = \sum (y_i - \hat{y}_i)^2$, sendo y_i o valor observado da variável dependente (na base de dados) e $\hat{y}_i$ o valor estimado pelo modelo de regressão para a variável dependente. O critério de parada é quando o *SSE* do subconjunto for menor do que uma fração do *SSE* do conjunto completo de casos. Esta fração é controlável a partir da interface.

Além disso, um teste de significância do modelo (Cf. Mendenhall & Sincich, 1996) é utilizado, com nível de significância definido pelo usuário.

4.2. ESCOLHA DO TESTE DE DIVISÃO

A escolha do teste de divisão é ainda uma busca exaustiva, considerando todos os valores possíveis para todos os atributos. Entretanto, a função que mede a qualidade de cada possível divisão, é chamada de Redução da Dispersão, e dada por:

$$RD = d(t) - \sum_i \frac{n_i}{n} d(t_i)$$

onde $d(t)$ representa uma função de dispersão da variável de resposta do conjunto avaliado, n_i é a cardinalidade do subconjunto i e n é a cardinalidade do superconjunto, que é igual a soma das cardinalidades dos subconjuntos, que são sempre dois.

Estão descritos alguns estudos em Wang & Witten (1997) que concluem que o uso

do desvio padrão e da variância como função objetiva no critério de divisão não apresentam diferenças significativas. Usamos como medida de dispersão ou a variância da variável independente ou uma razão entre a variância e a média.

A medida de *off-set* utilizada para a busca do ponto de divisão pode ser controlada a partir da interface do sistema, de maneira que a perda em função do ganho computacional pode ser controlado.

4.3. FUNÇÃO PONDERADA DE ERRO

O fator multiplicativo para a ponderação do erro total do modelo foi modificado. Na estimação dos modelos de regressão baseado no método dos mínimos quadrados, o desvio da variável dependente é comparado com a soma dos resíduos quadráticos do modelo através da estatística R^2, e existe uma estatística chamada de R^2 ajustado que faz esta compensação através dos graus de liberdade do desvio da variável dependente (n – 1) com o desvio do modelo estimado com v atributos ($n - v - 1$). O fator multiplicativo pra esta estatística ajustada é o mesmo utilizado pelo novo algoritmo, que tem a forma:

$$\frac{n-1}{n-v-1}$$

que tem as mesmas características do fator original, e a vantagem de lidar com graus de liberdade, de maneira padronizada.

4.4. SUAVIZAÇÃO DOS COEFICIENTES

Os valores da constante de suavização k passam a ser definidos em função do tamanho do conjunto de dados inicial. De acordo com a expressão de suavização, os pesos de ponderação são o tamanho do conjunto de treinamento que alcança o nó filho para a predição feita por ele e o valor de k para a predição feita pelo modelo de regressão no nó pai. Se o conjunto de treinamento do nó filho for menor que k, então a equação de regressão terá peso maior que a predição do filho. E se o conjunto de treinamento do nó filho for maior do que k, então a predição do filho será mais importante. O valor de k funciona como um limite que determina a quantidade máxima de casos que torna o modelo de regressão de um nó mais relevante do que a predição do seu filho. Fixado um valor para k, quanto mais distante da folha, maior a relevância da predição vinda do filho, visto que a quantidade de casos que alcança o filho cresce.

4.5. IMPLEMENTAÇÃO

A implementação do algoritmo foi feita dentro do ambiente Weka. Foi utilizado o IDE NetBeans da Java, versão 3.6 para desenvolvimento e compilação, bem como para gerar a documentação final em Javadoc.

5. TESTES

Alguns testes bem sucedidos do sistema estão demonstrados. Os resultados são comparados com o sistema M5', descrito anteriormente. A estimativa de erro foi feita por validação cruzada, com 10 subgrupos. As bases utilizadas estão disponíveis no Repositório de bases de dados de aprendizado de máquina da UCI (*University of California, Irvine*) (UCI, 2005), e têm as características fundamentais descritas na Tabela 1. Referências importantes que já usaram estas bases são Quinlan (1993) e Wang & Witten (1997).

Tabela 1: características das bases de dados

Nome da Base	Atributos Numéricos	Atributos Categóricos	Casos
auto-mpg	5	2	398
cpu	6	1	209
servo	2	2	167

As diferentes medidas de precisão, utilizadas neste trabalho, estão descritas na Tabela 2. Elas medem, com exceção da primeira, o erro do modelo. Estas medidas de precisão não têm comportamento monótono, ou seja, o melhor modelo para um caso específico usando um dos critérios pode não ser o melhor usando outro critério. É importante observar que a medida quadrática acentua os erros maiores que a unidade e reduz os erros menores.

Tabela 2: Medidas de performance para predição fornecidas pela Suíte Weka

Estatística	*Expressão*
Correlation coefficient	$\frac{s_{PA}}{\sqrt{s_P s_A}}$
Mean absolute error	$\frac{1}{n}\sum_i \lvert p_i - a_i \rvert$
Root mean squared error	$\sqrt{\frac{1}{n}\sum_i (p_i - a_i)^2}$
Relative absolute error	$\frac{\sum_i \lvert p_i - a_i \rvert}{\sum_i \lvert a_i - \bar{a} \rvert}$
Root relative squared error	$\sqrt{\frac{\sum_i (p_i - a_i)^2}{\sum_i (a_i - \bar{a})^2}}$

A Tabela 3 apresenta os resultados obtidos para as diversas medidas de performance, para cada base de dados utilizada, tanto do algoritmo proposto como para o sistema M5´.

Tabela 3: resultados dos testes comparativos

Base	Auto-mpg				
Sistema	Corr	MAE	RMSE	RAE	RRSE
M5.5'	0,9319	2,0028	2,8324	30,5824	36,1723
M5'	0,9238	2,0872	2,9912	31,8716	38,2
Base	cpu				
Sistema	Corr	MAE	RMSE	RAE	RRSE
M5.5'	0,9828	10,4089	29,2121	11,8744	18,8753
M5'	0,9766	13,6917	35,3003	15,6194	22,8092
Base	Servo				
Sistema	Corr	MAE	RMSE	RAE	RRSE
M5.5'	0,9689	0,2034	0,3867	17,5912	24,7879
M5'	0,9353	0,3059	0,5745	26,4577	36,8312

Como podemos observar, para todas as bases analisadas, houve uma melhora de performance do algoritmo proposto com relação ao algoritmo M5´, no que tange sua precisão. Vale ressaltar que a melhor performance de erro teve efeitos de piora da performance de tempo, em função, fundamentalmente, da implementação não otimizada.

6. CONCLUSÃO

A leitura de um algoritmo algumas vezes nos leva a propor modificações nas heurísticas empregadas. Algumas vezes estas modificações não são profundas e não são traduzidas em termos de ganhos de qualidade no produto final – o algoritmo modificado. Entretanto, quando as modificações se fazem notar, o resultado pode ser chamado de um novo algoritmo. Este é o processo de melhoria contínua da tecnologia – no caso de *software*. Este processo está diretamente ligado ao avanço da ciência rumo ao ainda não alcançado, que é, em última análise, a razão de sua existência.

No caso da Ciência da Computação, existe um fator agravante para o seu desenvolvimento, que é as inovações tecnológicas. O crescimento da capacidade computacional e da velocidade de processamento faz com que diversas limitações sejam desfeitas, o que torna diversas heurísticas e metodologias obsoletas. Diante da obsolescência, é inevitável que os algoritmos sejam revistos.

A proposta que foi feita aqui para o novo algoritmo de indução de modelos se encaixa neste perfil. Analisando historicamente, desde o CART até o M5.5', uma série de desenvolvimentos contínuos foram feitos, adequando o antigo sistema às tecnologias emergentes. Os sistemas apresentam maior poder de tratamento de casos extremos, como atributos categóricos, base escaladas, valores faltantes, entre outros. E trazendo sempre vantagens qualitativas, segundo mais importantes critérios de comparação de modelos.

A aproximação da técnica de indução de modelos de predição com as teorias estatísticas traz diversas vantagens. Além da padronização obtida ao usar heurísticas bem conhecidas na comunidade científica, uma outra importante vantagem foi o ganho de precisão observado nas bases de dados estudadas.

Por outro lado, a existência do M5.5' como um outro sistema de indução de modelos de predição, além do CART, M5 e M5' dentre outros, têm também a importante função de popularizar técnicas de predição. Importante ressaltar também que a popularização da suíte Weka também traz benefícios para o desenvolvimento das técnicas de *Data Mining*, pois a suíte é bastante estável e robusta, o que, juntamente com a sua acessibilidade torna-a uma excelente ferramenta.

Uma revisão de todo o código se faz necessária para que o sistema possa ser disponibilizado a nível acadêmico. Esta revisão teria que considerar os preceitos da Engenharia de Software e da Orientação a Objetos, de maneira que o reuso das classes e pacotes internos da suíte Weka fosse realizado corretamente. Esta revisão permitiria que o algoritmo desenvolvido fosse disponibilizado para a comunidade acadêmica juntamente com a suíte. Ainda sobre a implementação, uma tarefa que pode vir a ser desenvolvida é a implementação do sistema de regras, que é alternativo à árvore.

7. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Berry, M. J. A.; Linoff, G. *Data Mining Techniques*. Nova York, EUA: John Wiley & Sons, 1997. 454p.

[2] Berson, A., Smith, S. & Thearling, K. *Building data mining applications for CRM*. McGraw-Hill, 1999.

[3] Breiman, L. et al. *Classification and Regression Trees*. Boca Raton, Florida, EUA: Chapman & Hall/CRC, 1984. 358 p.

[4] CART. Cart Decision Tree Software for Data Mining Web Mining and Business. Disponível em: < http://www.salford-systems.com/cart.php >. Acesso em: 28 abr. 2005.

[5] Coelho, Paulo Sérgio de Souza. Um Sistema de para Indução de Modelos de Predição Baseados em Árvores. Tese de Doutorado, COPPE – UFRJ, Rio de Janeiro, 2005.

[6] Coelho, Paulo Sérgio de Souza; Ebecken, Nelson. *A comparison of some classification techniques*. In Zanasi, A., Brebbia, C. A., Ebecken, N. F. F. & Melli, P. *Data Mining III*. Southamptom, Inglaterra: WIT Press, 2002.

[7] Draper, N. R. & Smith Jr., H. *Apllied Regression Analysis*. 3ª Edição. Nova Iorque, EUA: John Wiley & Sons, 1998. 706 p.

[8] GNU. *GNU General Public License - GNU Project - Free Software Foundation (FSF)* <http://www.gnu.org/copyleft/gpl.html>. Acessado em: 21 de março de 2005.

[9] Habermas, Jurgen. *Racionalidade e Comunicação*. Editora Edições 70, 2002.

[10] Han, Jiawei; Kamber, Micheline. *Data Mining concepts and Techniques*. San Francisco: Morgan Kaufmann, 2001. 550 p.

[11] Haykin, Simon S. *Redes Neurais: princípios e prática*. 2 ed. São Paulo: Bookman, 2000. 900 p.

[12] Karalic, Aram & Cestnik, Bojan. *The Bayesian Approach to Tree-Structured Regression*. Proceedings of ITI'91 (Information Technology Interfaces). 155-160, 1991.

[13] Lachtermacher, Gerson. *Sistema de Previsão de Séries Temporais Utilizando Redes Neurais*. Pesquisa Operacional, 2 vol. 12 15-44, 1992.

[14] McCulloch, W. S. e Pitts, W. *A logical calculus of the ideias immanent in nervous activity*. Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943.

[15] Mendenhall, William & Sincich, Terry. *A Second Course in Statistics: regression analysis*. 5ª edição. Prentice Hall: New Jersey, 1996.

[16] Neter, J. et al. *Applied Linear Statistical Models*. 4 ed. Chicago: Irwin, 1996.

[17] Pearl, Judea. *Probabilistic Reasoning in Inteligent Systems: networks of plausible inference*. São Francisco, Califórnia, EUA: Morgan Kaufmann Publishers, Inc., 1998.

[18] Quinlan, John Ross. *Induction of Decision Trees*. Machine Learning, 1:81-106, 1986.

[19] Quinlan, John Ross. *Learning With Continuous Classes*. In Proceedings AI'92 (Adams Sterling, Eds), 343-348, Singapore: World Scientific, 1992.

[20] Quinlan, John Ross. *C4.5: programs for machine learning*. San Mateo, California, EUA: Morgan Kaufmann Publishers, 1993.

[21] Quinlan, John Ross. *Ross Quinlan's personal homepage*. Disponível em: <http://www.rulequest.com/Personal>. Acesso em: 28 abr. 2005.

[22] Robnik-Sikonja, Marko & Kononenko, Igor. An adaptation of Relief for attribute estimation in regression. In Fisher, D., editor, Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97), 296-304. Morgan Kaufmann Publishers, 1997.

[23] Szwarcfiter, Jayme Luiz & Markenzon, Lilian. *Estrutura de Dados e seus Algoritmos*. 2ª Edição. Rio de Janeiro: LTC Editora, 1994.

[24] Torgo, Luís Fernando Raínho Alves. *Inductive Learning of Tree-based Regression Models*. Tese (doutorado em Ciência dos Computadores), Faculdade de Ciências – Universidade do Porto. Porto, Portugal, 1999.

[25] UCI. *UCI Machine Learning Repository*. Disponível em: <http://www.ics.uci.edu/~mlearn/MLRepository.html>. Acesso em 24 de março de 2005.

[26] Weiss, Sholom M.; Indurkhya, Nitin. *Predictive Data Mining: a pratical guide*. São Francisco, CA: Morgan Kaufmann Publishers, 1998.228 p.

[27] Weka Machine Learning Project. Disponível em: <http://www.cs.waikato.ac.nz/~ml>. Acesso em: 10 abr. 2001.

[28] Weka Software. *Weka 3 - Data Mining with Open Source Machine Learning Software in Java*. Disponível em: < http://www.cs.waikato.ac.nz/~ml/weka/index.html >. Acesso em: 10 abr. 2001.

[29] Wang, Yong and Witten, Ian H. *Inducing Model Trees for Continuous Classes*. Proc of Poster Papers, 9 th European Conference on Machine Learning, Prague, Czech, 1997.

[30] Witten, Ian H. & Frank, Eibe. Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann: San Francisco, 2000.