REDES NEURAIS E ÁRVORES DE DECISÃO NA ANÁLISE DO CRÉDITO BANCÁRIO

Maria Teresinha Arns Steiner
UFPR – Departamento de Matemática, CP: 19081-CEP: 81531-990, Curitiba, PR; tere@mat.ufpr.br

Nei Yoshihiro Soma
ITA – Divisão da Ciência da Computação, Pça Mal. Eduardo Gomes, 50, Vl. das Acácias
CEP: 12228-990, São José dos Campos, SP; nysoma@comp.ita.br

Tamio Shimizu
USP – Departamento de Engenharia de Produção, São Paulo, SP; tmshimiz@usp.br

Júlio Cesar Nievola
PUC-PR – Programa de Pós-Graduação em Informática Aplicada, Av. Imaculada Conceição, 1155, CEP 80215-901, Curitiba, PR; nievola@ppgia.pucpr.br

Fábio Mendonça Lopes e **Andréia Smiderle**
Doutorandos do Programa de Pós-Graduação em Métodos Numéricos em Engenharia-UFPR,
CP: 19081-CEP: 81531-990, Curitiba, PR; fmendoncal@uol.com.br; andreiasmiderle@brturbo.com.br

Resumo
Reconhecer e prever quais clientes serão "bons ou maus pagadores" de crédito é tarefa importante e difícil para as instituições bancárias e os serviços de proteção ao crédito. Com registros históricos de 2.855 clientes de um banco alemão, foram abordadas neste artigo, comparativamente, as técnicas de Redes Neurais e Árvores de Decisão, utilizando os *softwares MatLab-Neural Networks Toolbox* e *WEKA*, respectivamente. Essas técnicas permitem o reconhecimento de padrões e, também, a sua utilização em diagnósticos posteriores, servindo como uma ferramenta auxiliar para o tomador de decisões (gerente de crédito). Vale salientar que o presente trabalho é uma extensão do trabalho apresentado em STEINER et al., 1999.
Palavras-chave: Reconhecimento de Padrões, Redes Neurais e Árvores de Decisão.

Abstract
Recognising and foreseeing credit customers as "good or bad payers" is an important and difficult task to bank institutions and to credit protection services. Based on 2,855 historical records from a german bank, we compared, in this work, the Neural Networks and Decision Trees techniques, comparatively, through MatLab-Neural Networks Toolbox and WEKA softwares. These techniques allow us to recognise patterns, and their further use in later evaluations as well, serving as a auxiliary tool for decision making (credit manager). We would like to emphasise that this present work is an continuation of the work presented by STEINER et al., 1999.
Keywords: Pattern Recognition, Neural Networks and Decision Trees.

1. Introdução

Parte das receitas de um banco comercial são compostas pelas operações de crédito como: concessão de empréstimos, financiamentos, fianças, cartões de crédito e cheques especiais, dentre outras. Qualquer erro na decisão de concessão de crédito pode significar que em uma única operação haja a perda do ganho obtido em várias outras bem-sucedidas. A correta decisão de crédito é essencial para a

sobrevivência das empresas bancárias. É sempre desejável e necessário, portanto, analisar uma proposta de negócio e comparar o custo de conceder com o custo de negar a operação.

A relação risco/retorno está implícita em qualquer concessão de crédito. O volume de incobráveis, assim como sua rentabilidade são efeitos da política adotada pela organização e de seus critérios de concessão de crédito. A otimização dos resultados é, portanto, decorrência de eficiente política de crédito, associada, evidentemente, à política de cobrança e às demais políticas da empresa (SILVA, 1993).

A análise do processo decisório quanto a concessão de crédito é bastante complexa envolvendo, além da experiência anterior do analista do banco, instrumentos e técnicas que possam auxiliá-lo nessa tarefa. Os métodos quantitativos, muito utilizados nesse tipo de análise, consideram registros históricos para a decisão sobre a concessão de crédito. Essas técnicas, entre as quais podem ser citadas Árvores de Decisão e Redes Neurais, se empregadas corretamente, constituem eficientes ferramentas auxiliares dos gestores de crédito.

São muitas as vantagens da utilização de técnicas quantitativas na administração de crédito, dentre as quais pode-se citar (ROSENBERG & GLEIT, 1994; CURNOW et al., 1997):

- maior número de merecedores de crédito receberá o crédito (ou crédito adicional), aumentando os lucros;
- maior número de não-merecedores de crédito terá o crédito negado (ou reduzido), diminuindo as perdas;
- os pedidos de crédito podem ser processados rapidamente;
- as decisões são objetivas e não passam por critérios subjetivos;
- menor número de pessoas é necessário para administrar o crédito e maior número com experiência pode concentrar-se nos casos mais difíceis.

O objetivo no presente trabalho é, dando continuidade ao trabalho apresentado por STEINER et al., 1999, utilizar as técnicas de Redes Neurais e de Árvores de Decisão visando classificar clientes adimplentes dos inadimplentes. Feito o reconhecimento desses padrões (fase da aprendizagem), essas técnicas podem ser utilizadas no diagnóstico de novos clientes, servindo como ferramentas auxiliares na tomada de decisão quanto ao crédito bancário (FU, 1982).

2. Descrição do Problema do Crédito Bancário

Serão utilizados no decorrer deste trabalho dados históricos referentes à concessão de empréstimos, entre 20.000 e 50.000 DM, por um banco alemão (RÖDDER, KOPITTKE & KULMANN, 1996). Estes dados correspondem a sete informações referentes a 2.855 clientes quanto ao seu comportamento bancário, com respostas binárias (sim ou não), e à informação fornecida pelo banco com relação ao cumprimento ou não do crédito (pagou ou não pagou). Estas oito informações (de A a G e K) comporão o banco de dados que será usado pelas técnicas aqui abordadas.

A. Existência de rendimentos compatíveis? (sim/não) – (AREND)

Existem rendimentos compatíveis quando:

$0{,}95Y - \max\{F, C\} - R \geq 0$, onde:

Y = rendimento mensal líquido;

C = gastos fixos (aluguéis, manutenção e outros);

F = limite de liberação de garantia;

R = prestações mensais referentes ao crédito.

B. Existência de patrimônio? (sim/não) – (BPATR)
É considerado o patrimônio compatível com a operação.

C. Estabilidade empregatícia há mais de três anos? (sim/não) – (CEMPR)
D. O solicitante é cliente? (sim/não) – (DCLIEN)
Este critério é preenchido quando o candidato apresenta uma conta corrente sem problemas há seis meses ou mais.

E. Inexistência de crédito insolvente pendente? (sim/não) – (EINSOL)

F. Inexistência de problemas no SPC? (sim/não) – (FSPC)
Qualquer tipo de problema conduz a uma resposta não.

G. Existência de fiador? (sim/não) – (GFIAD)

K. Crédito honrado? (sim/não) – (KPAGOU)
Foram considerados insolventes aqueles casos que assim podem ser lançados contabilmente. Não foram considerados atrasos de até um mês ou prorrogações de até três meses.

De forma resumida, os 2.855 casos utilizados estão representados no Quadro 1. Das 128 (2^7) possíveis combinações para as respostas às perguntas de A a G, só foram consideradas aquelas que efetivamente ocorreram, ou seja, 48 combinações.

Na terceira linha deste quadro, por exemplo, lê-se o seguinte: dos dez clientes (**K**PAGOUtotal) com **A**REND=0, **B**PATR=0, **C**EMP=1, **D**CLIEN=0, **E**INSOL=0, **F**SPC=0, **G**FIAD=0, oito casos tiveram um desenrolar positivo (**K**PAGOU=sim), enquanto em dois casos não foram honrados os pagamentos (**K**PAGOU=não).

Como pode ser observado, as combinações de observações constantes do Quadro 1 foram preenchidas em ordem crescente de ocorrência: primeiro os casos raros e depois os mais freqüentes. Mesmo havendo a tendência de o número de observações positivas de clientes (respostas para as perguntas de A a G serem iguais a **1**) correlacionar-se com um desenrolar positivo (**K**PAGOU=sim), esse fato não foi considerado neste trabalho.

3. A Aplicação de Redes Neurais ao Problema do Crédito Bancário

Para aplicar a técnica de Redes Neurais ao problema abordado, utilizou-se o pacote computacional *MatLab-Neural Network Toolbox* (DEMUTH & BEALE, 1994), onde optou-se por fazer uso de uma Rede Neural de Múltiplas Camadas (FAUSETT, 1995), (HAYKIN, 2002), na qual aplicou-se o algoritmo de aprendizagem retro-propagação ("*back-propagation*"). Como este assunto já é bem conhecido da comunidade científica, não serão apresentados maiores detalhes sobre o mesmo.

Para a topologia da rede em pauta, utilizou-se apenas uma camada escondida com i unidades, onde $0 \leq i \leq 20$. A função de ativação utilizada na camada escondida e na de saída foi a função sigmoidal (GORNI, 1993). Testes foram feitos para determinar o melhor número de neurônios i da camada escondida, que fornecesse o menor erro. Seguindo o procedimento apresentado em STEINER, 1995, o número de neurônios i para essa camada escondida é determinado durante a execução do programa da seguinte forma: começa-se com $i=0$ e verifica-se o número de padrões classificados corretamente quando o processo de aprendizagem converge e, assim, prossegue-se até $i=20$. Dessas *21* tentativas escolhe-se,

para *i*, aquela topologia que classificou o maior número de padrões corretamente, denotada por *i**. Os pesos *W* e as bias θ, para cada uma dessas topologias $0 \leq i \leq 20$, são aleatórios no início do processo.

Todo o procedimento é repetido para 5 conjuntos de pesos diferentes. Vale salientar que a cada época na aplicação do algoritmo *back-propagation* os $(m + k)$ padrões são apresentados em uma ordem aleatória, ou seja, a cada época é modificado o seqüenciamento com que os $(m + k)$ padrões são apresentados para a rede neural.

Número	**A** REND	**B** PATR	**C** EMP	**D** CLIEN	**E** INSOL	**F** SPC	**G** FIAD	**K** PAGOU=sim	**K** PAGOU=não	**K**PAGOU Total
1	0	0	0	0	0	0	0	8	2	10
2	0	0	0	0	0	0	1	8	2	10
3	0	0	1	0	0	0	0	8	2	10
4	1	0	1	0	0	0	0	8	4	12
5	1	0	0	0	0	0	1	17	2	19
6	0	0	0	0	0	1	0	17	2	19
7	0	0	1	0	0	0	1	17	3	20
8	1	0	0	0	0	0	0	17	3	20
9	1	1	0	0	0	0	0	17	5	22
10	1	0	0	1	1	1	0	26	2	28
11	1	0	0	0	0	1	1	26	2	28
12	1	0	1	0	0	1	0	26	2	28
13	1	0	1	1	0	1	0	26	2	28
14	0	0	0	1	0	1	0	26	3	29
15	1	0	0	0	0	1	0	26	3	29
16	1	1	1	0	0	0	0	26	4	30
17	1	1	1	1	0	1	0	34	1	35
18	1	0	0	1	0	1	0	34	2	36
19	1	1	1	0	0	1	0	34	2	36
20	1	1	1	1	1	1	0	34	2	36
21	0	0	0	1	1	1	0	34	3	37
22	0	0	1	0	0	1	0	34	4	38
23	1	1	0	0	0	0	1	34	5	39
24	1	0	0	1	0	1	1	43	1	44
25	1	0	0	1	1	1	1	43	2	45
26	0	0	0	0	0	1	1	43	3	46
27	0	0	1	1	0	1	0	43	3	46
28	1	1	1	0	0	0	1	43	4	47
29	1	0	1	0	0	0	1	43	7	50
30	1	1	1	1	0	1	1	52	1	53
31	1	1	0	1	1	1	0	52	2	54
32	0	0	1	1	1	1	0	52	3	55
33	1	0	1	1	1	1	0	60	2	62
34	0	0	1	0	0	1	1	60	3	63
35	0	0	0	1	0	1	1	64	2	66
36	0	0	0	1	1	1	1	78	2	80
37	1	1	0	0	0	1	0	78	2	80
38	1	1	0	1	0	1	0	86	3	89
39	0	0	1	1	1	1	1	86	4	90
40	1	0	1	0	0	1	1	95	2	97
41	1	1	0	0	0	1	1	95	3	98
42	0	0	1	1	0	1	1	95	4	99
43	1	1	0	1	1	1	1	121	2	123
44	1	1	0	1	0	1	1	130	2	132
45	1	1	1	0	0	1	1	156	3	159
46	1	0	1	1	1	1	1	173	2	175
47	1	1	1	1	1	1	1	190	2	192
48	1	0	1	1	0	1	1	208	3	211
								2.726	129	2.855

Quadro 1. Respostas da Amostra para os Casos de Concessão de Crédito Pesquisados (RÖDDER, KOPITTKE & KULMANN, 1996)

Como regra de parada do algoritmo para cada uma das topologias (número de neurônios na camada escondida $0 \leq i \leq 20$ e conjunto de pesos W) adotou-se a seguinte alternativa: sempre que o algoritmo apresentar uma variação de erro entre duas épocas consecutivas menor do que um valor ε (ε=0.01, por exemplo), finaliza-se o processo de aprendizagem. Repete-se esse processo para os *5* conjuntos de pesos, obtendo-se *5* valores para i^*. Entre os *5* conjuntos de pesos, escolhe-se o conjunto de pesos W, correspondente a i^*, que classifique o maior número de padrões corretamente, finalizando assim todo o processo.

Pelo procedimento acima descrito, para o problema real abordado neste trabalho ficou definida uma topologia composta por oito neurônios na camada escondida, a qual apresentou a menor porcentagem de erros na classificação dos padrões. Para essa técnica foram efetuadas 10 mil épocas (uma época corresponde a uma passagem dos $(m + k)$ padrões através da rede) utilizando o *MatLab*, sendo que o valor do erro sofreu alterações consideráveis no início da aprendizagem, mas a partir da qüingentésima época, aproximadamente, ele pouco se modificou.

Encerrada a aprendizagem, pôde-se obter os resultados dos diagnósticos para todos os 2.855 clientes enquadrados nos 48 casos mostrados no Quadro 1. Esses resultados constam na Tabela 1, onde constam também, os resultados obtidos através das Árvores de Decisão, assunto da próxima seção.

4. A Aplicação de Árvores de Decisão ao Problema do Crédito Bancário

As Árvores de Decisão utilizam a estratégia *dividir-e-conquistar* ("*divide-and-conquer*"), tendo como resultado um subconjunto do conjunto total de atributos. As Árvores de Decisão constituem uma das formas de aprendizado de máquina ("*machine learning*"), onde um problema complexo é decomposto em subproblemas mais simples. Recursivamente a mesma estratégia é aplicada a cada sub-problema (GAMA, 2002).

Quinlan, da Universidade de Sidney, é considerado o "pai das Árvores de Decisão". A sua contribuição foi a elaboração de um novo algoritmo chamado *ID3*, desenvolvido em 1983. O *ID3* e suas evoluções (*ID4*, *ID6*, *C4.5*, *See 5*) são algoritmos muito utilizados para gerar Árvores de Decisão. O atributo mais importante é apresentado na árvore como o primeiro nó, e os atributos menos importantes, segundo o critério utilizado, são mostrados nos nós subseqüentes. As principais vantagens das Árvores de Decisão são que elas "tomam decisões" levando em consideração os atributos que são considerados mais relevantes, segundo a métrica escolhida, além de serem compreensíveis para as pessoas. Ao escolher e apresentar os atributos em ordem de importância, as Árvores de Decisão permitem aos usuários conhecer quais fatores mais influenciam os seus trabalhos (QUINLAN, 1993).

Segundo GARCIA, 2000, as Árvores de Decisão consistem de:
- nodos (nós) que representam os atributos;
- arcos (ramos), provenientes dos nodos e que recebem os valores possíveis para estes atributos (cada ramo descendente corresponde a um possível valor deste atributo) e
- nodos folha (folhas da árvore), que representam as diferentes classes de um conjunto de treinamento, ou seja, cada folha está associada a uma classe.

Cada percurso na árvore (da raiz à folha) corresponde a uma regra de classificação.

O problema de construir uma Árvore de Decisão pode ser expresso recursivamente: primeiro selecione um atributo para colocar no nó raiz e faça um ramo para cada possível valor. Isto divide o problema em sub-conjuntos, um para cada valor do atributo. Agora o processo pode ser repetido recursivamente para cada ramo. Se a qualquer instante todos os exemplos em um nó têm a mesma

classificação, pare de desenvolver aquela parte da árvore. Como determinar, no entanto, qual atributo dividir? Escolhe-se o atributo que gere uma árvore menor e que tenha chances de classificar melhor, ou seja, precisamos medir o grau de pureza de cada nó. Com isto, poderemos escolher o atributo que produz os nós filhos mais puros (CARVALHO, 2002). A medida de pureza mais utilizada é chamada de informação e é medida em *bits*.

Construção de uma Árvore de Decisão

O processo de construção de uma Árvore de Decisão inicia-se a partir de um conjunto de treinamento, que contém exemplos com classes previamente conhecidas (dados históricos).

Para gerar uma árvore de decisão com uma alta taxa de predição é necessário fazer a escolha correta dos atributos que serão usados como teste no agrupamento dos casos. Estes testes devem gerar uma árvore com o menor número possível de subconjuntos, fazendo com que cada folha da árvore contenha um número significativo de casos. O ideal é escolher os testes de modo que a árvore final seja a menor possível.

Como analisar todas as possibilidades seria algo absurdo, foram desenvolvidos vários métodos aplicados na escolha dos atributos e dos testes a serem utilizados, sendo que todos concordam em dois pontos: uma divisão que mantém as proporções de classes em todas as partições é inútil e uma divisão onde em cada partição todos os exemplos são da mesma classes tem utilidade máxima. Uma vez feita a escolha, as outras possibilidades não são mais exploradas (LEMOS, 2003).

Para melhor esclarecer os critérios que levam à escolha de um atributo, faz-se necessário o conhecimento de dois conceitos: **Entropia** e **Ganho de Informação (**CARVALHO, 2000**).**

Entropia: É a medida que indica a homogeneidade dos exemplos contidos em um conjunto de dados. Ela permite caracterizar a "pureza" (e impureza) de uma coleção arbitrária de exemplos (OSÓRIO, 2000).

Dado um conjunto *S* contendo exemplos positivos ("+") e exemplos negativos ("-") que definem o conceito a ser aprendido, a entropia relativa dos dados deste conjunto *S* é indicada pela expressão (4.1) a seguir (WITTEN e FRANK, 2000):

(4.1) $$Entropia(S) = - P_{(+)} \cdot \log_2 P_{(+)} - P_{(-)} \cdot \log_2 P_{(-)}$$

onde:

$P_{(+)}$ = Proporção entre os exemplos positivos e o total de exemplos do conjunto, ou seja, número de casos positivos / número total de casos.

$P_{(-)}$ = Proporção entre os exemplos negativos e o total de exemplos do conjunto, ou seja, número de casos negativos / número total de casos.

É assumido que: $0 \cdot \log_2 0 = 0$, por definição.

A equação (4.1) apresentada é usada para calcular a entropia levando-se em conta duas classes. Fazendo a generalização para "N" Classes, tem-se a a equação (4.2):

(4.2) $$Entropia\ (S) = - \sum_{i=1}^{N} P_i \ \log_2 \ P_i$$

A *Entropia (S)* tem máximo valor para $(log_2 P_i)$ se $P_i = P_j$ para qualquer $i \neq j$ (caso em que o número de casos positivos é igual ao número de casos negativos) e a *Entropia (S) = 0*, se existe um *i* tal que $P_i = 1$ (caso em que todos os exemplos são da mesma classe).

Ganho de Informação (Critério *GAIN*): Segundo OSÓRIO, 2000, o ganho de informação é a medida que indica o quanto um dado atributo irá separar os exemplos de aprendizado de acordo com a sua função objetivo (classes). O ganho de informação é a redução esperada no valor da Entropia devido à ordenação do conjunto de treinamento segundo os valores do atributo escolhido (ANTUNES, 2000).

GAIN (S, A) = Redução esperada na entropia de *S*, causada pelo particionamento dos exemplos em relação a um atributo escolhido (*A*).

(4.3) $$Gain\ (S, A) = Entropia\ (S) - \sum_{v=1}^{N} \frac{|S_v|}{|S|} . Entropia(S_v)$$

onde:
A = Atributo considerado; *N* = Número de valores possíveis que este atributo pode assumir;
S_v = Sub-conjunto de *S* onde o atributo *A* possui o valor *V*.

O método de particionamento recursivo para geração das Árvores de Decisão, que subdivide o conjunto de casos de treinamento até que cada subconjunto em cada partição contenha casos de uma única classe ou até que nenhum outro teste ofereça qualquer melhora, pode gerar árvores complexas que acabam perdendo o seu poder de generalização. Faz-se necessário então, adotar algumas medidas para transformar árvores complexas em árvores mais simples (QUINLAN, 1993).

Existem dois caminhos pelos quais este particionamento recursivo pode ser modificado para produzir árvores mais simples: decidindo não continuar a dividir o conjunto de dados de treinamento ou removendo retrospectivamente alguma estrutura já construída pelo método. O primeiro caminho pode causar o término da divisão antes que o benefício das divisões subseqüentes se tornem evidentes. Na segunda alternativa, o processo de *dividir-e-conquistar* segue até o fim e então, a árvore é "podada". Este processo é mais lento, mas muito mais seguro. O processo de poda irá, em geral, causar união de alguns exemplos de classes diferentes em um mesmo nó.

Pode-se dizer que uma das maiores motivações para podar Árvores de Decisão é no sentido de se evitar o ajuste demasiado / sobreajuste ("*overfitting*") da árvore aos dados. Neste caso, a árvore poderia se ajustar a peculiaridades dos dados, que talvez não ocorram em dados ainda não vistos (FREITAS, 2000). Ainda segundo FREITAS, 2000, deve-se, no entanto, ter cuidado para que a poda não seja muito agressiva, a fim de não gerar um sub-ajustamento ("*underfitting*") da árvore aos dados. A realização da "poda" ou simplificação das Árvores de Decisão é baseada em "erros".

Muitos são os algoritmos de classificação que constróem Árvores de Decisão. Não há uma forma de determinar qual é o melhor algoritmo, sendo que um algoritmo pode ter melhor desempenho em determinada situação e outro pode ser mais eficiente em outros tipos de situações. O algoritmo *J4.8*, por exemplo, é a implementação em Java do algoritmo *C4.5*. Existe ainda uma versão melhorada da *C4.5* que é *C4.5 Revision8*, que é a última versão pública desta família de algoritmos antes do *C5.0*, uma implementação comercial (WITTEN e FRANK, 2000). A essência do algoritmo para a Árvore de Decisão *C4.5* trabalha de acordo com o que foi relatado no decorrer desta seção 4, sendo que o mesmo apresenta resultados satisfatórios para a maioria dos problemas.

A utilização de Árvores de Decisão apresenta as seguintes vantagens: não assume nenhuma distribuição particular para os dados; as características ou atributos podem ser categóricos (qualitativos) ou numéricos (quantitativos); pode construir modelos para qualquer função desde que o número de exemplos de treinamento seja suficiente; possui elevado grau de interpretabilidade.

Após a construção de uma Árvore de Decisão é importante avaliá-la. Esta avaliação é realizada através da utilização de dados que não tenham sido usados no treinamento. Esta estratégia permite estimar como a árvore generaliza os dados e se adapta a novas situações, podendo, também, se estimar a proporção de erros e acertos ocorridos na construção da árvore (BRADZIL, 2002).

Para aplicar a técnica de Árvores de Decisão, mais especificamente a técnica *J4.8*, ao problema abordado, foi utilizado o *software* livre *WEKA* (*Waikato Environment for Knowledge Analysis*, disponível no site www.cs.waikato.ac.nz/ml/weka).

5. Análise dos Resultados Obtidos

Na Tabela 1 pode-se observar e comparar os diagnósticos (probabilidade de pagamento) fornecidos pelas técnicas pesquisadas, Rede Neural (RN, Múltiplas Camadas) e Árvore de Decisão (AD, *J4.8*). Nessa tabela tem-se que, por exemplo, para os dez clientes da linha três do Quadro 1, a Rede Neural diagnostica que o cliente pagará o seu crédito com uma probabilidade de 78,93% e a Árvore de Decisão fornece uma probabilidade de 80.00% para esse caso. Pode-se observar que as probabilidades para todas as 48 combinações são bastante próximas para as duas técnicas.

Deve-se salientar, no entanto, que a Árvore de Decisão *J4.8* não chegou a formar uma árvore efetivamente; ela simplesmente classificou todos os clientes como sendo adimplentes. Isto pode ter ocorrido devido ao fato do número de clientes adimplentes (2.726) ser significativamente maior do que o número de clientes inadimplentes (129), dando um percentual de acerto de 95,48% (2726/2855) ao classificar todos os clientes como adimplentes. Por este motivo é que as probabilidades contidas na Tabela 1 para esta técnica, não passam de um simples cálculo de probabilidade que podem ser obtidas fazendo-se:
[(1 – KPAGOU=não) / KPAGOUTotal]. 100.]

Já na Tabela 2 constam os resultados dos diagnósticos para algumas combinações não constantes naquelas apresentadas no Quadro 1, evidenciando que a tanto Rede Neural, por meio do *MatLab*, como a Árvore de Decisão, por meio do *WEKA*, conseguem diagnosticá-los através do aprendizado efetuado com os 48 casos. Aqui, novamente, tem-se que para a Árvore de Decisão *J4.8*, como todos os casos foram classificados como adimplentes, tem-se que qualquer novo caso é classificado como adimplente com 100% de probabilidade de pagamento.

6. Conclusões

O objetivo deste trabalho é apresentar duas metodologias para o reconhecimento de padrões de comportamento de clientes com relação à adimplência de crédito, examinando o caso concreto de um banco alemão. Com base nesse reconhecimento, o objetivo passa a ser a previsão do comportamento de futuros clientes, buscando minimizar as perdas bancárias, além do esforço gerencial na decisão quanto a concessão de crédito. Este trabalho dá uma continuidade ao trabalho apresentado por STEINER et al., 1999.

Neste trabalho, especificamente, tendo por base dados históricos, foram determinadas as probabilidades de adimplência (diagnósticos) de 2.855 clientes pesquisados, através das técnicas de Redes Neurais e Árvores de Decisão, comparativamente, através do treinamento das referidas técnicas com o uso dos *softwares MatLab* e *WEKA*, respectivamente. Os diagnósticos (probabilidade de pagamento) para cada um dos clientes enquadrados nas combinações do Quadro 1 estão contidos na Tabela 1, onde pode-se

verificar que os resultados foram bastante semelhantes para ambas as metodologias. Feito o treinamento das técnicas pode-se obter os diagnósticos de quaisquer clientes (não necessariamente contidos no Quadro 1) os quais encontram-se contidos na Tabela 2.

A idéia principal é pesquisar técnicas capazes de fazer o reconhecimento de padrões de forma eficiente e eficaz com o objetivo de poder oferecer ao especialista da área (gerente de crédito, neste caso), um sistema computacional contendo uma técnica de performance satisfatória como uma ferramenta adicional para a tomada de decisão (quanto a concessão de crédito, neste trabalho).

Número	Diagnóstico RN (%) Múltiplas Camadas	Diagnóstico AD (%) *J4.8*
1	82,18	80,00
2	80,93	80,00
3	78,93	80,00
4	69,79	66,67
5	88,70	89,47
6	86,14	89,47
7	84,90	85,00
8	83,48	85,00
9	78,65	77,27
10	94,54	92,86
11	95,76	92,86
12	91,51	92,86
13	94,11	92,86
14	90,50	89,65
15	89,01	89,65
16	85,83	86,67
17	96,79	97,14
18	93,35	94,44
19	95,00	94,44
20	93,98	94,44
21	90,74	91,89
22	91,14	89,47
23	87,53	87,18
24	96,92	97,73
25	96,90	95,55
26	93,00	93,48
27	93,96	93,48
28	89,62	91,49
29	86,16	86,00
30	99,06	98,11
31	97,41	96,30
32	94,81	94,54
33	96,75	96,77
34	96,46	95,24
35	95,41	96,97
36	96,78	97,50
37	97,62	97,50
38	96,50	96,63
39	97,70	95,56
40	97,06	97,94
41	98,08	96,94
42	96,51	95,96
43	98,61	98,37
44	98,72	98,48
45	97,98	98,11
46	98,19	98,86
47	98,66	98,96
48	98,56	98,58

Tabela 1. Probabilidade de pagamento dos clientes enquadrados nas 48 combinações contidas no Quadro 1, sendo que para a obtenção dos resultados foram utilizados os *softwares MatLab* (RN) e *WEKA* (AD)

Número	**A** REND	**B** PATR	**C** EMP	**D** CLIEN	**E** INSOL	**F** SPC	**G** FIAD	Diagnóstico RN (%)	Diagnóstico AD (%)
1	0	1	1	0	0	0	0	76,72	100,00
2	0	1	1	1	1	1	1	98,54	100,00
3	0	0	0	0	1	1	1	96,32	100,00
4	1	1	1	0	1	1	1	88,91	100,00
5	1	1	1	1	1	0	1	85,62	100,00
6	0	0	0	0	0	1	0	86,14	100,00
7	0	0	0	1	0	0	0	86,92	100,00
8	0	1	0	1	0	1	0	93,92	100,00
9	1	0	1	0	1	0	1	71,81	100,00
10	0	0	1	1	1	0	0	63,17	100,00

Tabela 2. Probabilidade de pagamento de observações não aprendidas
(combinações que não participaram do treinamento)

Referências:

ANTUNES, C. M. Árvores de Decisão, 2002. Disponível em: <http://mega.ist.utl.pt/~ic.apr/doc/aulas/arvoresdecisão.pdf> Acesso em: 21 jul. 2002.

BRADZIL, P. B. Construção de Modelos de Decisão a partir de dados, 1999. Disponível em: http://www.nacc.up.pt/~pbrazdil/Ensino/ML/ModDecis.html. Acesso em: 21 jul. 2002.

CARVALHO, I. C. *Uma Contribuição ao Estudo do Efeito das Insconsistências em Bases de Dados usadas no Treinamento de Sistemas Simbólicos e Conexionistas.*Dissertação de Mestrado, CEFET-PR, Curitiba, PR, 2000.

CARVALHO, I. C. *Métodos de Mineração de Dados (Data Mining) como Suporte à Tomada de Decisão.*Dissertação de Mestrado, ITA, São José dos Campos, SP, 2002.

CURNOW, G.; KOCHMAN, G.; MEESTER, S.; SARKAR, D.; WILTON, K. Automating credit and collections decisions at AT&T capital corporation. *Interfaces*, v.27, p.29-52, 1997.

DEMUTH, H. & BEALE, M. *Neural network toolbox for use with MATLAB (user's guide)*. Natick, Massachusetts, The Math Works, Inc., 1994.

FAUSETT, L. *Fundamentals of neural networks - architectures, algorithms, and applications*. Florida Institute of Technology. Prentice Hall, Upper Saddle River, New Jersey, 1995.

FREITAS, A. A. *Uma Introdução a Data Mining.Informática Brasileira em Análise*. CESAR - Centro de Estudos e Sistemas Avançados do Recife. Ano II, n. 32, mai./jun. 2000.

FU, K.S. *Syntatic pattern recognition and applications*. New Jersey, Prentice-Hall, 1982.

GORNI, A.A. *Redes neurais artificiais - uma abordagem revolucionária em inteligência artificial*. São Paulo, Micro Sistemas, 1993.

GAMA, J. Árvores de Decisão, 2000.
Disponível em: <http://www.liacc.up.pt/~jgama/Mestrado/ECD1/Arvores.html>. Acesso em: 14 ago. 2002.

GARCIA, S. C. *O uso de Árvores de Decisão na Descoberta de Conhecimento na Área da Saúde*. SEMANA ACADÊMICA. Universidade Federal do Rio Grande do Sul, 2000.

HAYKIN, S. *Redes neurais. Princípios e Prática.* Porto Alegra, Bookman, 2002.

LEMOS, E. P. *Análise de Crédito Bancário com o uso de Data Mining: Redes Neurais e Árvores de Decisão*. Dissertação de Mestrado, UFPR, Curitiba, PR, 2003.

OSÓRIO, F. Sistemas Adaptativos Inteligentes - Indução de Árvores de Decisão, 2000. Disponível em: <http://www.inf.unisinos.br/~osorio/sadi.html> Acesso em: 12 ago. 2002.

QUINLAN, J. C. *C4.5: Programs for machine learning*. San Mateo: Morgan Kaufmann, 1993. 302p.

RÖDDER, W.; KOPITTKE, B.; KULMANN, F. *Sistemas especialistas probabilísticos*. Texto para o Ensino à Distância feito em Cooperação com a Fern Universität Hagen, 1996.

ROSENBERG, E. & GLEIT, A. Quantitative methods in credit management: a survey. *Operations Research*, v.42, n.4, p.589-613, 1994.

SILVA, J. PEREIRA. *Análise e decisão de crédito*. São Paulo, Atlas, 1993.

STEINER, M. T. A., CARNIERI, C., KOPITTKE, B. H. & STEINER NETO, P. J. *Sistemas especialistas probabilísticos e redes neurais na análise do crédito bancário*. São Paulo, RAUSP, vol. 34, n.3, 1999.

STEINER, M.T.A. *Uma metodologia para o reconhecimento de padrões multivariados com resposta dicotômica*. Santa Catarina, 1995. Tese (Doutorado) em Engenharia de Produção – Universidade Federal de Santa Catarina.

WITTEN, I. H.; FRANK, E. *Data Mining: Pratical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers. San Francisco, Califórnia, 2000.