

Modelos de Regressão Logística, Redes Neurais e *Support Vector Machine* (SVMs) na Análise de Crédito a Pessoas Jurídicas

Logistic Regression Models, Neural Networks and Support Vector Machine (SVMs) in the Credit Analysis Corporate Entities

Vania Gryczak Gevert

Departamento de Matemática

Universidade Estadual do Centro-Oeste – UNICENTRO, Guarapuava, PR

vaniagg2@hotmail.com

Arinei Carlos Lindebck da Silva

Engenharia de Produção

Universidade Federal do Paraná - UFPR, Curitiba, PR

arineicls@gmail.com

Fabiano Gevert

Universidade Estadual do Centro-Oeste – UNICENTRO, Guarapuava, PR

gevert_fg@yahoo.com.br

Vanessa Terezinha Ales

Engenharia de Produção

Universidade Federal do Paraná - UFPR, Curitiba, PR

vanessa.ales@gmail.com

Resumo: Na área de crédito bancário, ao tomar um empréstimo ou realizar qualquer negócio, o passo seguinte ao preenchimento do cadastro na unidade concedente é a análise de crédito. O uso e a posse de instrumentos que auxiliem na tarefa de classificar possíveis solventes ou insolventes podem tornar-se uma vantagem competitiva. O presente trabalho foi desenvolvido para prever empresas adimplentes e inadimplentes. Para isso, são analisados o registro de 199 clientes (pessoas jurídicas)

Recebido em 06/01/2011 - Aceito em 01/04/2011.

RECEN Guarapuava, Paraná v. 12 n° 2 p. 269-293 jul/dez 2010

de uma agência bancária, utilizando três técnicas: Regressão Logística, Redes Neurais de Múltiplas Camadas e *Support Vector Machine*. O objetivo é a comparação do desempenho das três técnicas escolhidas. Os resultados encontrados demonstram uma superioridade de *Support Vector Machine* em relação às outras duas técnicas. Alguns dos motivos para esse sucesso estão relacionados ao fato de essa técnica exibir bom desempenho de generalização em muitas bases de dados reais, ter boa fundamentação teórica, a eliminação da possibilidade de mínimos locais pelo processo de treinamento e a existência de poucos parâmetros livres para ajuste.

Palavras-chave: análise de crédito; modelos de previsão; regressão logística; redes neurais artificiais; *support vector machine*.

Abstract: In the banking area, in order to grant a loan or do any business, the cadastre is followed by credit analysis. The use of instruments that aid in solvency classification may become a competitive advantage. This paper was developed to predict if companies will pay or not the loan. Therefore, 199 clients (companies) of a bank agency were analyzed using three different techniques: Logistic Regression, Multilayer Perceptron Neural Network and Support Vector Machine. The objective is to compare their performances. The Support Vector Machine presented better results than the other two techniques. Some reasons for this success are related to the fact that this technique exhibits good generalization performance in many real datasets, its good mathematical foundation, eliminating the possibility of local minima for the training process and the low number of free parameters to adjust.

Key words: artificial neural networks; credit analysis; forecasting models; logistic regression; support vector machine.

1 Introdução

Em finanças, o vocábulo crédito define um instrumento de política de negócios a ser utilizado por uma empresa comercial ou industrial na venda a prazo de seus produtos, ou por um banco comercial, por exemplo, na concessão de empréstimo,

financiamento ou fiança.

O crédito a pessoas jurídicas, nas diversas modalidades financeiras, tem sido tratado empiricamente ou por sistemas de *credit scoring* baseados em análise discriminante ou de regressão logística. No entanto, nos últimos anos, em razão do crescimento do mercado de crédito às empresas no Brasil, o assunto passou a merecer maior atenção e preocupação, notadamente em razão da forte elevação das taxas de inadimplência [1].

Após a 2ª Guerra Mundial, alguns analistas perceberam que os modelos estatísticos poderiam trazer benefícios nas decisões de crédito e começaram a utilizá-los. Nas décadas seguintes, a evolução do *credit scoring* continuou e foi impulsionada pela chegada dos cartões de crédito no fim dos anos 60 [2].

O *credit scoring* pode ser definido como um método de avaliar o risco em empréstimos, através dos proponentes, dados históricos e técnicas estatísticas [3].

Na área financeira das indústrias, do comércio e de demais atividades econômicas, os homens de finanças estão constantemente envolvidos com tomadas de decisão que podem representar o sucesso ou fracasso próprio, de um projeto específico ou mesmo da empresa [4]. Conceder ou não crédito, tomar recursos ou aplicá-los são decisões constantes a que estão sujeitos os executivos financeiros, qualquer que seja o negócio da empresa.

Os bancos precisam distinguir, entre seus clientes, quais pagarão a dívida contraída, e quais não honrarão o compromisso assumido.

Quando a empresa tem, à sua disposição, uma regra de reconhecimento de padrões e classificação que indique previamente a chance de inadimplência de um futuro cliente, a decisão de concessão de crédito é facilitada, podendo-se utilizar argumentos quantitativos em substituição aos argumentos subjetivos e decidir com maior confiança [5].

As técnicas estatísticas multivariadas têm sido consideradas como ferramentas poderosas na administração do risco de inadimplência existente na concessão de crédito [6]. Para a construção de um modelo de previsão de risco de crédito, é importante identificar qual a técnica estatística mais eficiente para modelar os dados da empresa, de forma a conseguir a melhor previsão de comportamento dos clientes.

O objetivo final da instituição ou banco é que seja efetuado o pagamento do empréstimo. Porém, nem sempre isso é possível, levando os tomadores de empréstimos à inadimplência. Alguns métodos quantitativos foram desenvolvidos por estudiosos na área para prever e classificar prováveis adimplentes e inadimplentes. Como o trabalho a ser realizado pretende classificar empresas nestas duas classes, é necessário conhecimento sobre os diversos modelos desenvolvidos, bem como suas vantagens e limitações.

Nenhum modelo consegue precisão absoluta, ou seja, acertar totalmente as previsões. Sabendo disto, qualquer avanço em acuracidade da previsão gera ganhos financeiros para a instituição concedente. Daí vem o interesse em analisar diferentes tipos de modelos e apontar quais apresentam maior precisão.

Na literatura pesquisada, encontram-se poucos estudos que abordam *Support Vector Machine* como ferramenta de modelos de *credit scoring* [7, 8]. Em contrapartida, Redes Neurais e Regressão Logística são largamente empregadas neste tipo de problema, como pode ser visto em vários trabalhos [9–11]. Por esta razão, julgou-se oportuno apresentar as três técnicas para a utilização em um mesmo banco de dados e comparar suas vantagens e desvantagens.

Os modelos de previsão de risco de crédito vêm como uma ferramenta de auxílio massificado, que é caracterizado pela avaliação de um grande número de solicitações de pequenos valores, já que a competitividade do mercado exige decisões rápidas. O analista informa os dados de seu potencial cliente no sistema de crédito e, imediatamente, o computador fornece a informação quanto à aprovação do crédito. O método estatístico utilizado para a construção do modelo leva em conta o histórico da instituição com seus clientes, possibilitando a identificação das características capazes de diferenciar o bom do mau pagador [12].

Portanto, predição e classificação são as tarefas a serem realizadas. Após análise de diversas técnicas de predição estudadas na literatura, optou-se por três delas, por se adequarem melhor ao banco de dados disponível e oferecerem uma margem de erro aceitável, além de propiciarem rápido processamento computacional e leitura de dados acessíveis para usuários leigos. Desta forma, as técnicas selecionadas foram:

- Regressão Logística;
- Redes Neurais Artificiais de Múltiplas Camadas (RNA);
- *Support Vector Machine* (SVM).

Seguem detalhamentos de cada técnica e dos procedimentos metodológicos utilizados.

2 Regressão logística

Nos modelos de regressão logística, a variável dependente é, em geral, uma variável binária (nominal ou ordinal) e as variáveis independentes podem ser categóricas (desde que dicotomizadas após transformação) ou contínuas.

Considere o caso em que as observações podem ser classificadas em uma de duas categorias mutuamente exclusivas (0 e 1). Como exemplo, as categorias poderiam representar uma empresa que pode ser classificada como adimplente ou inadimplente.

A variável dependente binária Y pode assumir valores:

$$Y_i = \begin{cases} 1 & \text{Se o } i\text{-ésimo indivíduo pertencer a categoria dos adimplentes} \\ 0 & \text{Se o } i\text{-ésimo indivíduo pertencer a categoria dos inadimplentes} \end{cases}$$

O modelo de regressão logística é também conhecido como modelo logístico e é baseado na função sigmóide, $f(z)$, dada por:

$$f(z) = \frac{1}{1 + e^{-z}} \quad (1)$$

ou ainda, habitualmente mais usada

$$f(z) = \frac{e^z}{1 + e^z} \quad (2)$$

Portanto, pode-se observar que a função logística varia entre 0 e 1, $0 \leq f(z) \leq 1$ e essa é a principal razão do modelo logístico ser usado para descrever uma dicotômica probabilidade de algo acontecer ou não acontecer, ou seja, variável dicotômica.

Existem, ainda, outras duas características de $f(z)$ que tornam o modelo logístico largamente utilizado:

- z representa um índice que combina a contribuição de diversos fatores de risco, e $f(z)$ representa o risco (probabilidade) de que um evento ocorra, para um dado z ;
- possui forma de “S”, indicando que o efeito de z em $f(z)$ é mínimo até que algum “gatilho” seja disparado, depois aumenta rapidamente até que algum nível seja alcançado, voltando a crescer lentamente.

2.1 O modelo matemático logístico

A partir da função logística $f(z)$, pode-se obter o modelo logístico, escrevendo z como a soma linear das variáveis independentes e substituindo na função:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (3)$$

$$E(Y|x) = \pi(x) = \frac{e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n}}{1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n}} \quad (4)$$

onde $E(Y|x)$ é a esperança de Y dado x .

3 Redes neurais artificiais

Rede Neural Artificial (RNA) é uma das técnicas de tratamento de dados mais recentes e que tem despertado grande interesse tanto de pesquisadores da área de tecnologia quanto da área de negócios [13].

RNA pode ser definida como um método de solucionar problemas de inteligência artificial, constituída de um sistema de circuitos que simulam o cérebro humano, inclusive seu comportamento. São modelos matemático-computacionais inspirados no funcionamento das células neuronais, isto é, na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência (a rede aprende, corrige erros e faz descobertas com a alimentação de dados) [14].

Assim como o cérebro humano, as RNAs se destacam pelo alto nível de interconexão e paralelismo maciço, isto é, vários dados sendo operados simultaneamente, onde suas informações são executadas por unidades denominadas neurônios. A informação entre os neurônios é transmitida por meio de conexões denominadas sinapses ou pesos sinápticos. Os pesos sinápticos são alterados através de processos algorítmicos, para que possam representar o conhecimento adquirido. Estas alterações caracterizam o processo de aprendizagem de uma rede neural artificial [14].

Uma rede neural é formada por um conjunto de neurônios, também chamados de elementos de processamento. As unidades de processamento são distribuídas em um conjunto de camadas. Uma rede neural possui uma estrutura que pode ser escrita a partir de quatro conceitos [15]: o tipo de modelo de rede neural; as unidades de processamento que tratam a informação que entra na rede; um sistema de nós que interligam as unidades de processamento; uma função de aprendizado, a partir da qual o sistema calibra o modelo.

Muitos são os modelos de redes neurais, mas os modelos básicos são: *perceptron*, Redes Lineares e Redes de Múltiplas Camadas (MLP).

As Redes de Múltiplas Camadas são modelos de redes que apresentam uma ou mais camadas de entrada e saída, chamadas camadas intermediárias. Este tipo de Rede Neural Artificial é o modelo mais utilizado atualmente; geralmente são treinadas através do algoritmo de *backpropagation*.

A estrutura de uma rede neural é composta pelos fatores:

- Arquitetura;
- Topologia;
- Seleção da função de ativação dos neurônios da camada escondida e de saída;
- Transformação dos dados ou normalização;
- Seleção do conjunto de dados;
- Algoritmo de treinamento e seus respectivos parâmetros;
- Critério de parada do treinamento.

A arquitetura da RNA refere-se ao formato da rede. Toda rede é dividida em camadas, usualmente classificadas em três grupos, onde os neurônios de uma camada são conectados apenas aos neurônios da camada imediatamente posterior, não havendo conexões entre neurônios da mesma camada. Porém, caracteristicamente, as camadas são totalmente conectadas, sendo assim definidas: camada de entrada (onde os padrões são apresentados à rede), camada(s) intermediária(s) (onde é feita a maior parte do processamento, por meio das conexões ponderadas; podem ser consideradas como extratoras de características) e camada de saída (onde o resultado final é concluído e apresentado).

A figura 1 ilustra a arquitetura de uma RNA com duas camadas de entrada, quatro neurônios em duas camadas intermediárias e apenas uma camada de saída.

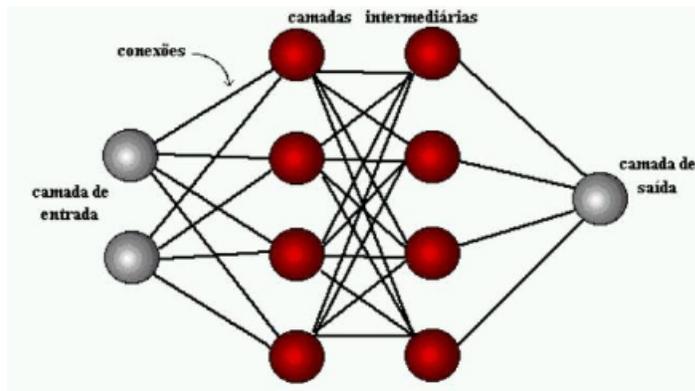


Figura 1. Conexões em uma Rede Neural Artificial [16]

Os três tipos principais de arquiteturas são [17]: redes *feedforward* (redes em avanço/para frente) com uma única camada, redes *feedforward* com múltiplas camadas, e redes recorrentes ¹.

Com referência à topologia da rede, deve-se ter:

- o número de nós da camada de entrada, que corresponde ao número de variáveis do problema;
- o número de camadas escondidas e o número de neurônios a serem colocados

¹RNAs que apresentam conexões com *loops*. São mais utilizadas em sistemas dinâmicos. Em uma rede corrente, as entradas passadas podem ser lembradas e influenciar a próxima saída gerada [18].

nesta camada: inexistente um critério pré-determinado sobre o número de neurônios que deve ser utilizado; em geral, redes neurais com poucos neurônios são preferidas, pois tendem a possuir um maior poder de generalização, evitando, assim, o sobreajuste (*overfitting*). Porém, se a rede possui poucos neurônios, a rede não aprende, podendo ocorrer *underfitting* (a rede não converge durante o treinamento);

- o número de neurônios da camada de saída.

Acerca da função de ativação, também chamada de função de transferência, é uma função matemática que, aplicada à combinação linear entre as variáveis de entrada e os pesos que chegam a determinado neurônio, retorna ao seu valor de saída. Existem várias funções matemáticas utilizadas. Porém, comumente utiliza-se a função linear, a função logística ou a função tangente hiperbólica.

Em relação à seleção dos dados, ao implementar uma rede neural, o conjunto de dados é separado em dois subconjuntos: o de treinamento e o de teste. O conjunto de treinamento é utilizado para ajustar os parâmetros da rede, e o conjunto de teste para validar o modelo.

A um conjunto de regras bem definidas para a solução de um problema de aprendizado denomina-se algoritmo de aprendizado. Existem vários tipos de algoritmos de aprendizado, tais como, *backpropagation*, *perceptron*, entre outros, específicos para determinados modelos de redes neurais, como a MLP, Hopfield, etc. Normalmente, estes algoritmos diferem entre si, pelo modo como os pesos são modificados [19].

Com relação ao algoritmo de treinamento, Santos et al [20] destacam que existem vários métodos para o treinamento de uma rede neural, mas que o algoritmo mais popularmente usado é *backpropagation*. A aplicação deste algoritmo requer a escolha de um conjunto de parâmetros (número de iterações do algoritmo, critério de parada, pesos iniciais, taxa de aprendizado), cuja influência pode ser decisiva para a capacidade de generalização da rede.

Desta forma, após o treinamento, o conjunto de teste é utilizado para verificar a habilidade de generalização da rede sob condições reais de utilização. A habilidade de generalização da rede se refere ao seu desempenho ao classificar padrões do conjunto

de teste [20]. Como critério de parada do algoritmo (*backpropagation*), em geral, é utilizado um número máximo de ciclos ou iterações.

4 *Support vector machine*

Support Vector Machines (SVM) são sistemas de aprendizagem de máquina que usam espaços de hipóteses de funções lineares na alta dimensionalidade característica do espaço, treinado como algoritmo de aprendizagem da Teoria de Otimização, que implementa um limite derivado da Teoria da Aprendizagem Estatística. Esta aprendizagem estratégica, introduzida por Vapnik e colaboradores, vem superando, em poucos anos desde sua introdução, outros sistemas de aprendizagem em ampla variedade de aplicações com maior desempenho [21].

Basicamente o funcionamento de uma SVM pode ser descrito da seguinte forma: dadas duas classes e um conjunto de pontos que pertencem a essas classes, uma SVM determina o hiperplano que separa os pontos de forma a colocar o maior número de pontos da mesma classe do mesmo lado, enquanto maximiza a distância de cada classe a esse hiperplano. A distância de uma classe a um hiperplano é a menor distância entre ele e os pontos dessa classe e é chamada de margem de separação. O hiperplano gerado pela SVM é determinado por um subconjunto dos pontos das duas classes, chamados de vetor de suporte.

Tratando-se de classificação binária, o problema é achar uma função paramétrica, linear ou não, para um hiperplano de separação dos pontos em dois conjuntos no R^m , em que m é o número de dimensões existentes. No caso onde o problema seja separável por um hiperplano linear com um conjunto de N observações $x_i = (x_{i1}, \dots, x_{im})$ e respostas binárias $y_i \in \{-1, 1\}$ têm-se três hiperplanos:

1. Hiperplano de Separação: $H_0 : y = w^t x + b = 0$ que separa as observações.
2. Hiperplano Superior: $H_1 : y = w^t x + b = +1$ que é definido por pelo menos 1 ponto pertencente ao grupo com $y = +1$.
3. Hiperplano Inferior: $H_1 : y = w^t x + b = -1$ que é definido por pelo menos 1 ponto pertencente ao grupo com $y = -1$.

A figura 2 mostra os hiperplanos de separação, superior e inferior no espaço $m = 2$.

Os pontos que definem os hiperplanos H_1 e H_2 são chamados de *support vectors* (vetores suporte) e a orientação do plano de separação (H_0) é feita de forma que a distância entre H_1 e H_2 seja máxima.

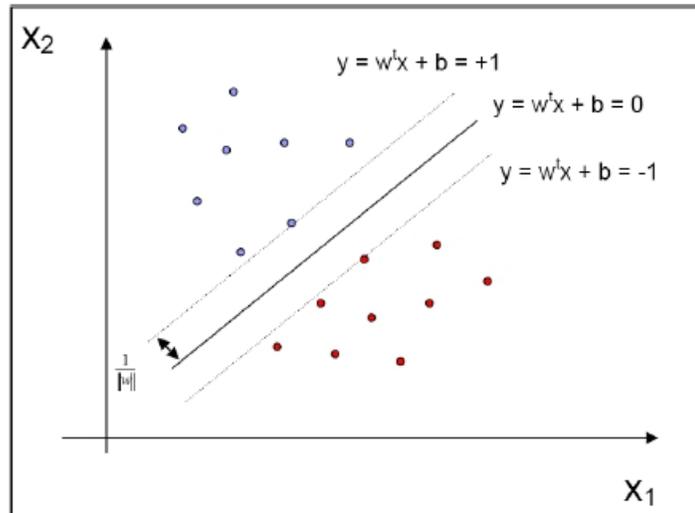


Figura 2. Hiperplanos de Separação, superior e inferior formados no espaço $m = 2$

Uma vez que a distância entre os hiperplanos H_1 e H_2 é calculada por $D(H_1, H_2) = \frac{2}{\|w\|} = \frac{2}{w^t w}$ e, o objetivo é encontrar os parâmetros w que maximizem essa distância, define-se a função objetivo do problema primal do SVM como sendo:

$$\text{Minimizar } \frac{1}{2} w^t w$$

Como restrição, no problema separável com um hiperplano linear, para que não haja pontos entre H_1 e H_2 faz-se $w^t x - b \geq +1$ para $y = +1$ e $w^t x - b \leq -1$ para $y = -1$. Essas duas restrições podem ser combinadas fazendo com que a formulação matemática desse problema tenha como restrição:

$$y(w^t x - b) \geq +1$$

Assim, esse problema de separação tem $m + 1$ incógnitas (w_1, \dots, w_m, b) . A es-

timitativa dos parâmetros é definida pelos pontos sobre H_1 e H_2 , chamados *support vectors* (vetores suporte), de forma que os outros pontos podem ser movidos livremente sem alterar o resultado da otimização.

A extensão natural desse modelo é o tratamento dos problemas não separáveis por um hiperplano linear. Nesse caso, introduz-se N variáveis de folga ($\xi_i \geq 0, i = 1, \dots, N$), de forma a criar uma penalidade na função objetivo e uma folga nas restrições. Portanto, a formulação do problema dual de separação no caso inseparável por um hiperplano linear é:

$$\text{Minimizar } \frac{1}{2} \omega^t \omega + C \left(\sum_{i=1}^N \xi_i \right)$$

$$\text{s.a. } y_i(\omega^t x_i - b) \geq +1 - \xi_i$$

$$\xi_i \geq 0, i = 1, \dots, N \tag{5}$$

em que C é uma constante de penalização ($C > 0$). Esse problema tem $N + m + 1$ incógnitas ($\xi_i, \dots, \xi_N, \omega_i, \dots, \omega_M, b$).

Esse modelo permite encontrar um hiperplano separador aceitando pontos classificados incorretamente, porém com a maior margem de separação entre as classes.

O primeiro termo da função tem o objetivo de maximizar a margem, enquanto que o segundo termo, $C \cdot (\sum_{i=1}^N \xi_i)$ minimiza o valor das variáveis de folga ξ_i , reduzindo o número de pontos que ficam do lado incorreto. Ou seja, o parâmetro C enfatiza maior ou menor importância das variáveis de folga, possibilitando que o modelo do SVM não perca a capacidade de generalização.

Devido à natureza das restrições do problema primal, pode-se ter dificuldades na obtenção da solução do problema primal, por isso, utiliza-se a teoria do Lagrangeano para obter a forma dual do SVM. O problema dual possui a mesma solução do primal, ou seja, pode-se revolver o problema primal indiretamente através da resolução do problema dual.

O Lagrangeano é obtido por meio de Multiplicadores de Lagrange, que estão associados as restrições de desigualdade do problema primal. Esses multiplicadores de Lagrange devem ser positivos. As restrições do problema dual ficam mais simples do que as do problema primal, sendo dado pela formulação a seguir:

$$\begin{aligned}
 \text{Max}L(w, b, x, a) &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x^i)(x^j) \\
 \text{s.a.} \quad &\sum_{i=1}^N \alpha_i y_i = 0 \\
 &0 \leq \alpha_i \leq C
 \end{aligned} \tag{6}$$

Na formulação do problema dual existe um produto interno $(x^i)(x^j)$ que pode ser substituído por uma função $K(x^i, x^j)$, denominada função kernel, obedecendo o Teorema de Mercer. Essas funções kernel mapeam os dados de entrada em um espaço de dimensão maior, no qual os dados podem ser separados por meio de um hiperplano, ou seja, os dados tornam-se linearmente separáveis, como pode ser observado na figura 3.

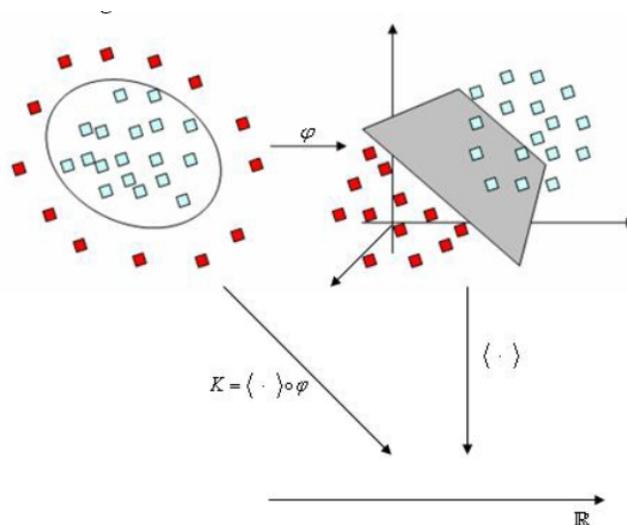


Figura 3. Representação da função Kernel [22]

As funções kernel realizam um mapeamento implícito, portanto, nas formulações do SVM, os exemplos de treinamento nunca aparecem isolados, mas em forma de produto interno, que pode ser substituído por uma função kernel.

Os vetores suportes são os dados x^i cujos multiplicadores de Lagrange α_i possuem valores não nulos. Estes são os valores que contribuem para a construção do hiperplano ótimo.

Uma característica das soluções do SVM é a esparsidade dos Multiplicadores de Lagrange, ou seja, apenas uma fração dos pontos será vetor suporte.

Para classificar os padrões do conjunto de treinamento como vetores suporte verificam-se os valores dos respectivos Multiplicadores de Lagrange conforme as condições de Karush-Kuhn-Tucker (KKT), ou seja:

Se $\alpha_i = 0, y_i \cdot f(x_i) > 1$ então x_i é considerado um vetor comum, que se situa do lado correto, na região da sua classe.

Se $0 < \alpha_i < C, y_i \cdot f(x_i) = 1$ então x_i é um vetor suporte, situa-se sobre a margem da região da sua classe. É conhecido como vetor suporte *non-bound* (VS-NB).

Enfim, se $\alpha_i = 0, y_i \cdot f(x_i) < 1$, então x_i é um vetor suporte *bound* (VS-bound). Ele pode se localizar entre a margem e o hiperplano separador, caso $0 < \xi_i < 1$. Ele pode estar na própria superfície de separação, caso $\xi_i = 1$. Ou pode estar na região de classe oposta à sua, do outro lado da superfície de separação, caso $\xi_i > 1$ [21].

Com isso, o SVM é considerado uma ferramenta que detecta automaticamente os vetores mais relevantes para a construção de um hiperplano ótimo, que possa fazer o reconhecimento de novos dados.

5 Procedimentos metodológicos

No desenvolvimento desta pesquisa, foram utilizados dados reais sobre os clientes de uma agência bancária, num total de 199 empresas, das quais 64 são inadimplentes e 135 são adimplentes, de onde foram extraídas as 21 informações. Tais informações foram escolhidas porque há a necessidade delas para o preenchimento do cadastro requerido pelo banco.

Os dados analisados referem-se às seguintes informações: existência de restrições em nome da empresa, restrições baixadas nos últimos cinco anos, tempo de conta no

banco, setor de atividade (comércio, indústria, serviços), tempo de atividade, número de funcionários, imóvel (próprio, alugado ou cedido), bairro (centro ou outros), principais clientes (pessoa física, jurídica ou mista), faturamento anual, se possui conta em outro banco, valor de bens imóveis e móveis, possui seguro empresarial, possui aplicações financeiras na instituição, porcentagem de vendas a prazo, experiência de crédito na instituição, histórico da conta corrente, existência de restrição no nome dos sócios, restrições baixadas nos últimos cinco anos no nome dos sócios, sociedade de cônjuges, valor de bens móveis e imóveis em nome dos sócios. Juntamente com esses dados, a informação se a empresa é adimplente ou inadimplente.

Para a obtenção de um bom resultado, devem ser feitas várias divisões diferentes no mesmo conjunto de dados em conjuntos de treinamento e de teste. Este método, chamado de *cross validation*, fornece uma estimativa mais realista do erro a ser cometido pelo método de classificação utilizado. Deve-se manter, nesta divisão, a mesma proporção entre as classes para os conjuntos de treinamento e validação [18].

Conforme recomenda Haykin [17], o número de divisões diferentes no mesmo conjunto é 10, escolhendo-os de forma aleatória para a separação das amostras de treinamento e teste. Então, optou-se em utilizar o mesmo número de divisões proposto pelo autor em um único conjunto de dados.

Em relação à divisão entre treinamento e teste, não existe nenhuma regra rígida estabelecida, e alguns autores preferem partições na proporção 60-40 ou 75-25 entre os grupos de treinamento e validação, respectivamente [15]. Pesquisando trabalhos anteriores, observaram-se diversas divisões: Cunico [23] optou em dividir os conjuntos na proporção 70-30, Lemos [9] na proporção 90-10, ambos obtendo resultados satisfatórios. Neste trabalho, optou-se em separar os dados na proporção 80-20 para o conjunto de treinamento e teste, respectivamente, para observar se os resultados também seriam satisfatórios.

5.1 Implementação da técnica Regressão Logística

Para a implementação da técnica de Regressão Logística, utilizou-se o *software Statistica*, pois, além de apresentar recursos de cálculos e gráficos, é de fácil manuseio e utiliza pouco tempo de processamento.

A análise dos dados foi feita alocando no *software* a matriz de treinamento acrescentada da coluna das respostas. Portanto, a matriz a ser analisada ficou com formato 108 x 21.

O *software* possui um pacote especial na opção *Nonlinear Estimation* que, depois de alimentado com a matriz, oferece a possibilidade de várias análises estatísticas, e entre elas está a Regressão Logística (*Logistic Regression*). Para a utilização do *software*, deverão ser informadas as variáveis independentes, dependente e o método utilizado para estimativa dos parâmetros.

5.2 Implementação da técnica Redes Neurais

Para a implementação da técnica de Redes Neurais, foi utilizado o software *MATLAB – Neural Networks Toolbox*.

Optou-se por uma rede *backpropagation* com correção de erro pelo algoritmo Levenberg-Marquadt ², que faz parte do *default* da função utilizada no *MATLAB*. Também foi necessário informar ao *software* o número de neurônios na camada de entrada, na camada escondida e na saída, bem como a função de ativação, o número de iterações e a margem de erro desejada.

A quantidade de neurônios da camada de entrada é igual ao número de variáveis utilizadas que, no caso deste trabalho, é igual a 21; a quantidade de neurônios na camada de saída é igual a 1; a função de ativação foi a sigmoideal (*logsig*) e utilizada apenas uma camada intermediária ou escondida.

A técnica foi executada várias vezes, sempre variando a topologia da rede, ou seja, o número de neurônios na camada escondida ou ainda o número de iterações.

5.3 Implementação da técnica *Support Vector Machine*

Para a implementação do *Support Vector Machine* foi utilizado um programa desenvolvido por Ales [22] em linguagem *Visual Basic* ³. O valor da variável para as

²Levenberg-Marquardt: é um algoritmo de aproximação que utiliza o método de Gauss-Newton modificado pela introdução de um parâmetro μ que funciona como fator de estabilização no treinamento da rede.

³O *Microsoft Visual Basic* é uma linguagem para desenvolvimento de aplicações visuais para ambiente *Windows* baseado em *Basic (Beginners Allpurpose Symbolic Instruction Code)*.

empresas adimplentes é 1, e para as inadimplentes é -1.

Inicialmente, para o treinamento, foi utilizado o botão **Variando Parâmetros**, onde os valores dos parâmetros e das funções *kernel* mudavam sem a interferência do usuário. Após esse treinamento foram analisados os resultados com menor taxa de erros (0%) e treinadas as funções *kernel* com os mesmos parâmetros, porém, separando os dados nos conjuntos de treinamento e teste.

6 Resultados

Os resultados finais da técnica de regressão logística podem ser visualizados na tabela 1.

Tabela 1. Resultado final da regressão logística

Regressão Logística					
Média acerto treinamento	Desvio padrão	Média acerto teste	Desvio padrão	Média de acerto do modelo	Tempo computacional (s)
100,00%	0,00%	96,99%	3,02%	98,50%	menor que 0,1

Observa-se, na tabela 1, que o modelo teve desempenho de 100% na fase de treinamento e desvio padrão de 0% e, na fase de teste, teve 96,99% de desempenho e desvio padrão de 3,02%. Logo, a média de acerto do modelo é de 98,50%. O tempo computacional mostrou-se satisfatório pois, nos 10 conjuntos treinados, o tempo foi menor que 0,1 segundo.

A tabela 2 apresenta os resultados obtidos com a técnica de Redes Neurais. Na tabela 2, pode-se destacar a rede 5, que possui acerto de 93,39% e 90,16% e desvio padrão de 7,99% e 10,55%, nas fases de treinamento e teste, respectivamente.

O tempo de execução das redes também pode ser destacado, pois as redes treinadas obtiveram tempos menores do que 0,1 segundo, o que significa baixo tempo computacional. Sendo assim, a rede eleita é a número 5, por ter o melhor desempenho dentre as redes treinadas, baixo custo computacional e média de acerto do modelo de 91,78%.

Tabela 2. Performance das redes treinadas

Rede	Média Acerto Treinamento	Desvio padrão	Média acerto teste	Desvio padrão	Número de neurônios na camada intermediária	Número de iterações	Função	Tempo comp.(s)
1	91,37%	5,60%	87,78%	7,78%	6	30	logit	>0,1
2	86,53%	12,75%	82,45%	14,90%	6	50	logit	>0,1
3	91,42%	5,67%	89,09%	6,08%	6	100	logit	>0,1
4	87,49%	13,92%	79,89%	17,91%	8	30	logit	>0,1
5	93,39%	7,97%	90,16%	10,55%	8	50	logit	>0,1
6	90,80%	7,85%	84,33%	11,56%	8	100	logit	>0,1
7	91,08%	4,56%	84,69%	10,40%	10	100	logit	>0,1
8	86,36%	10,16%	78,56%	14,65%	12	100	logit	>0,1

Os resultados da técnica SVM estão destacados na tabela 3. As diferentes funções *kernel*, variando os parâmetros, obtiveram resultados satisfatórios. Então, foram escolhidas as funções que obtiveram desempenho de erro de treinamento de 0% com todo o conjunto de dados.

Os dados foram novamente treinados, porém, dividindo o conjunto em dois subconjuntos, o de treinamento (80%) e o de teste (20%), escolhendo as funções *kernel* e os parâmetros obtidos na tabela 3.

Para reforçar os resultados obtidos nesta etapa, o programa disponibiliza a estratégia validação cruzada (*cross validation*). O usuário escolhe o número de testes e a porcentagem do teste. O número de teste escolhido foi $k = 10$, conforme recomenda Haykin [17], e a porcentagem do teste foi a mesma utilizada nos outros modelos, ou seja, 20%. O tempo computacional das funções destacadas foi satisfatório, com 1,24, 66,77 e 54,50 segundos, respectivamente.

Na tabela 4, verifica-se que as funções *kernel* destacadas obtiveram os melhores desempenhos dentre as funções treinadas, pois as taxas de erro de treinamento foram de 0%, 0,32% e 0,76%, com desvio padrão de 0%, 0,81% e 0,40%, respectivamente. As taxas de erro de teste foram de 0%, 0,95% e 0,71%, respectivamente com desvio padrão de 0%, 1,66% e 1,15%, respectivamente.

Observa-se também, na tabela 4, que as outras funções *kernel* (gaussiano e poli-

nomial não homogêneo) possuem baixas taxas de erros, com desvio padrão pequeno e com poucos vetores suporte.

Verifica-se, na tabela 3, que o número de vetores de suporte *bound* e número de vetores suporte não *bound* são iguais a 0 e 100 para o *kernel* sigmoidal, causando *overfitting* ou um superajustamento do modelo, não generalizando a superfície de separação, podendo desconsiderá-lo para comparação. O número de vetores de suporte para o *kernel* polinomial homogêneo é 19 (VS) e 0 (VSB) e, para o kernel sigmoidal, 17 (VS) e 21(VSB). O número de vetores da função *kernel* polinomial homogêneo é aproximadamente 10% do total dos dados de treinamento, e isso gerou uma superfície de separação bem generalizada.

Tabela 3. Funções *kernel* treinadas (157), número de vetores suporte *bound* (VS), número de vetores suporte não *bound* (VSB), parâmetros C, grau da função polinomial e constante, *kapa* para a função sigmoidal, *sigma* para a função Gaussiana, 157 dados para o conjunto de treinamento e 42 para o conjunto de teste

VS	VSB	C	kernel	Grau	Cte	kapa	Sigma
20	0	10	Polinomial Não Homogêneo	2	5	0	0
0	100	1	Sigmoidal	0	0	10	0
19	18	1	Gaussiano	0	0	0	1
12	0	10	Polinomial Não Homogêneo	6	0	0	0
14	0	10	Polinomial Não Homogêneo	6	1	0	0
12	5	10	Gaussiano	0	0	0	1
4	14	100	Gaussiano	0	0	0	10
16	1	100	Produto interno	0	0	0	0
12	0	10	Polinomial Homogêneo	5	0	0	0
0	100	20	Sigmoidal	0	-10	0,1	0
19	0	8	Polinomial Homogêneo	3	0	0	0
22	0	3	Polinomial Não Homogêneo	8	2	0	0
1	98	10	Sigmoidal	0	-15	0,8	0
13	41	10	Sigmoidal	0	-2	0,1	0
3	99	1	Gaussiano	0	0	0	15
13	0	100	Polinomial Homogêneo	5	0	0	0
13	0	100	Polinomial Não Homogêneo	6	1	0	0
17	21	1000	Sigmoidal	0	-5	0,2	0

Observando os resultados da tabela 3, o *kernel* polinomial homogêneo pode alcançar a média de erro de 0,99% na etapa de classificação. Isto representa 99,01%

de acerto do modelo, sendo eleito o *kernel* polinomial homogêneo, com parâmetros $C=8$ e grau do polinômio 3, para realizar a comparação.

Tabela 4. Performance das funções kernel, desvio padrão e tempo computacional

Média erro trein.	Desvio padrão	kernel	Média erro teste	Desvio padrão	Média do modelo	Tempo computacional (s)
0,00%	0,00%	Polinomial Não Homogêneo	1,49%	1,77%	0,74%	78,44
0,00%	0,00%	Sigmoidal	0,00%	0,00%	0,00%	1,24
1,59%	0,34%	Gaussiano	2,38%	0,00%	1,98%	54,02
1,34%	1,25%	Polinomial Não Homogêneo	4,05%	2,52%	2,69%	5,30
2,10%	1,90%	Polinomial Não Homogêneo	4,52%	5,55%	3,31%	4,75
2,04%	0,84%	Gaussiano	4,05%	2,52%	3,04%	51,08
3,82%	1,88%	Gaussiano	3,33%	1,23%	3,57%	3,69
0,13%	0,27%	Produto interno	2,81%	2,79%	1,47%	73,56
1,02%	1,00%	Polinomial Homogêneo	2,86%	3,33%	1,94%	6,36
31,85%	0,00%	Sigmoidal	33,33%	0,00%	32,59%	0,67
0,32%	0,81%	Polinomial Homogêneo	0,95%	1,66%	0,63%	66,77
0,70%	0,92%	Polinomial Não Homogêneo	2,14%	2,37%	1,42%	1,80
31,85%	0,00%	Sigmoidal	33,33%	0,00%	32,59%	0,97
1,47%	0,43%	Sigmoidal	2,38%	0,00%	1,92%	14,80
31,85%	0,00%	Gaussiano	33,33%	0,00%	32,59%	1,78
1,78%	1,30%	Polinomial Homogêneo	3,13%	0,00%	2,45%	5,77
0,96%	1,57%	Polinomial Não Homogêneo	2,38%	3,37%	1,67%	5,28
0,76%	0,40%	Sigmoidal	0,71%	1,15%	0,73%	54,50

6.1 Comparação entre as técnicas utilizadas

Os resultados obtidos nas três técnicas de predição permitiram a comparação entre elas e a análise da melhor técnica.

Os percentuais de acerto nos modelos de regressão logística e redes neurais mostraram-se apropriadas na classificação, atingindo 100% e 93,39% na fase de treinamento, e 96,99% e 90,16% na fase de teste, respectivamente.

A técnica *Support Vector Machine* (SVM) apresentou resultados bastante próximos entre a fase de treinamento (99,68%) e de teste (99,05%).

Tabela 5. Comparação entre as técnicas

	Média de Acertos		
	Treinamento	Teste	Geral
Regressão Logística	100,00%	96,99%	98,50%
Redes Neurais	93,39%	90,16%	91,78%
SVM	99,68%	99,05%	99,37%

A tabela 5 mostra os resultados de classificação obtidos pelos três métodos. Todos eles apresentaram bons resultados de classificação, pois, segundo Picinini et al [24]: “Modelos de *credit scoring* com taxas de acerto acima de 65% são considerados bons por especialistas”.

Desta forma, a técnica eleita como a mais indicada para a predição de empresas adimplentes e inadimplentes é a técnica *Support Vector Machine*, que proporcionou a média de resultados de 99,37%. Contudo, deve ser ressaltado, mais uma vez, que a adoção de qualquer um dos modelos traria bons resultados à instituição financeira.

7 Conclusões

O objetivo deste estudo foi construir um modelo de previsão de risco de crédito comparando as técnicas Regressão Logística, Redes Neurais e *Support Vector Machine*. As características especiais de cada uma das técnicas foram analisadas com o objetivo de encontrar o modelo com melhor desempenho na predição de clientes adimplentes e inadimplentes.

As três técnicas apresentadas e testadas neste trabalho - Regressão Logística, Redes Neurais e *Support Vector Machine* - mostraram ser ferramentas de grande valia para os analistas de crédito bancário, obtendo respectivamente 98,50%, 91,78% e 99,37% de acerto na classificação.

Vale destacar que, a técnica *Support Vector Machine* teve um melhor desempenho em relação às outras técnicas, seguido da Regressão Logística e finalmente das Redes Neurais.

Support Vector Machine é uma área de pesquisa que está em franco desenvolvimento, restrita ainda a poucos grupos, porém já alcançando um público maior. Como a teoria é totalmente formalizada, embora ainda haja muito a ser explorado, o enfoque da maioria dos trabalhos são as aplicações. Busca-se um dimensionamento do poder de abrangência de SVM em problemas práticos. Um dos problemas investigados é o reconhecimento de padrões de empresas adimplentes e inadimplentes que pode ser útil ao analista de crédito.

A técnica SVM mostrou ser uma ferramenta de grande utilidade para auxiliar os analistas de crédito a decidir sobre a sua concessão, pois classificou as empresas em solventes e insolventes com baixo índice de erro.

Esses testes contribuíram para uma compreensão mais prática da técnica, reforçando o conhecimento teórico e confirmando a aplicabilidade do SVM à análise de crédito bancário.

As SVMs são robustas diante de dados de grande dimensão, sobre os quais outras técnicas de aprendizado comumente obtêm classificadores super ou sub ajustados. Outra característica atrativa é a convexidade do problema de otimização formulado em seu treinamento, que implica na existência de um mínimo global. Essa é uma vantagem das SVMs sobre, por exemplo, as Redes Neurais Artificiais, em que há mínimos locais na função objetivo minimizada. Além disso, o uso de funções *kernel* na não-linearização das SVMs torna o algoritmo eficiente, pois permite a construção de simples hiperplanos em um espaço de dimensão de forma tratável do ponto de vista computacional. Porém, entre as principais limitações das SVMs, encontra-se a sua sensibilidade a escolhas de parâmetros e a dificuldade de interpretação de modelo gerado por essa técnica.

Os bancos e instituições de concessão de crédito, utilizando suas informações cadastrais, têm condições de avaliar novas empresas que estão em busca de crédito.

Não se pode afirmar que a empresa que obteve um sinal favorável ao crédito vai ser adimplente, pois fatores externos como a economia do país, ou problemas inesperados como enchente, incêndio, roubo, podem alterar o comportamento da empresa em razão dos compromissos adquiridos.

A experiência profissional do analista, aliada à técnica SVM, são instrumentos

que podem ajudar na tarefa de tomada de decisão. Porém, nunca por si só essas técnicas poderão substituir a figura do especialista no contexto da análise de crédito.

Sugestões que poderão ser objeto de estudos, de modo a aperfeiçoar a metodologia aqui abordada:

- Implementar essa técnica de reconhecimento de padrões para maior número de casos;
- Aprofundamento teórico das funções *kernel*;
- As mesmas técnicas poderiam ser utilizadas na análise de crédito para pessoas físicas;
- Avaliar e implementar outros métodos de Aprendizado de Máquina na predição de empresas que foram citados na revisão de literatura;
- Fazer outros tipos de testes de validação para os métodos utilizados.

8 Referências

- [1] MENDES FILHO, E. F.; CARVALHO, A. C. P. L. F.; MATIAS, A. B. Utilização de redes neurais artificiais na análise de risco de crédito a pessoas físicas. In: III Simpósio Brasileiro de Redes Neurais. Recife, Novembro 1996.
- [2] THOMAS, C. L. A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *Int J Forecasting*, v.16, n. 2, p. 149–172, 2000.
- [3] MESTER, L. J. What's the point of crediscoring? *Bus Review*. p. 3-16, 1997.
- [4] SECURATO, J. R. Crédito – Análise e Avaliação do Risco – Pessoas Físicas e Jurídicas. São Paulo: Saint Paul, 2002, 356 p.
- [5] GUIMARÃES, I. A.; CHAVES NETO, A. Reconhecimento de padrões: metodologias estatísticas em crédito ao consumidor. *RAE eletrônica*, v.1, n. 1, 2002. Disponível em:

<http://www.rae.com.br/electronica/index.cfm?FuseAction=Artigo&ID=1215&Secao=FINANÇAS2&Volume=1&Numero=2&Ano=2002>. Acesso em 02/06/2009.

- [6] SILVA, J. P. Gestão e Análise de Risco de Crédito. São Paulo: Atlas, 2003.
- [7] HUANG, Z.; CHEN, H.; HSU, C.; CHEN, W.; WU, S. Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decis Support Syst*, v. 37, p. 543–558, 2004.
- [8] LAI, K. K.; YU, L.; WANG, S.; ZHOU, L. Credit risk analysis using a reliability-based neural network ensemble model. *ICANN, Part II*, LNCS 4132, p. 682–690, 2006.
- [9] LEMOS, E. P. Análise de crédito bancário com o uso de data mining: redes neurais e árvores de decisão. Dissertação de Mestrado, Universidade Federal do Paraná, PR, 2003.
- [10] STEINER, M. T. A.; CARNIERI, C.; KOPITTTKE, B. H.; STEINER NETO, P. J. Sistemas especialistas probabilísticos e redes neurais na análise do crédito bancário. *Rev Adm*, v. 34, n. 3, p. 56–67, 1999.
- [11] GONÇALVES, E. B. Análise de risco de crédito com o uso de modelos de regressão logística, redes neurais e algoritmos genéticos. Dissertação de Mestrado, Universidade de São Paulo, SP, 2005.
- [12] SILVA, J. P. Os dois lados do crédito. *GV Executivo*, v. 5, n. 3, p. 68–72, 2006.
- [13] CORRAR, L. J.; PAULO, E.; DIAS FILHO, J. M. Análise multivariada para os cursos de Administração, Ciências Contábeis e Economia. São Paulo, Ed. Atlas, 2007, 568 p.
- [14] FAUSETT, L. Fundamentals de Neural Networks: architectures, algorithms and applications. Ed Prentice-Hall: New Jersey, 1994.
- [15] HAIR JR., J. F.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E. Multivariate data analysis. New Jersey: Prentice Hall, 1998, 816 p.

- [16] CARVALHO, A. P. L. F. Redes neurais artificiais. Disponível em: <http://www.icmc.usp.br/andre/research/neural>. Acesso em 29/05/09.
- [17] HAYKIN, S. Neural Networks: a comprehensive foundation. New Jersey: Prentice-Hall, 1999.
- [18] MARTINELLI, E. Extração de conhecimento de redes neurais artificiais. Dissertação de Mestrado, Universidade de São Paulo – USP, SP, 1999.
- [19] TATIBANA, C. Y.; KAETSU, D. Y. Uma introdução às redes neurais. Disponível em: <http://www.din.uem.br/ia/neurais/#links>. Acesso em 10/07/09.
- [20] SANTOS, A. M.; SEIXAS, J. M.; PEREIRA, B. B.; MEDRONHO, R. A. Usando redes neurais artificiais e regressão logística na predição da hepatite A. *Rev Bras Epidemiol*, v. 8, n. 2, p. 117–126, 2005.
- [21] SHAWE-TAYLOR, J.; CRISTIANINI, N. Kernel Methods and Pattern Analysis. United Kingdom: Cambridge, 2000.
- [22] ALES, V. O Algoritmo Sequential Minimal Optimization de Support Vector Machine: Uma Técnica de Reconhecimento de Padrões. Dissertação de Mestrado, Universidade Federal do Paraná, PR, 2008.
- [23] CUNICO, L. H. B. Técnicas em data mining aplicadas na predição de satisfação de funcionários de uma rede de lojas do comércio varejista. Dissertação de Mestrado, Universidade Federal do Paraná, PR, 2005.
- [24] PICININI, R.; OLIVEIRA, G. M. B.; MONTEIRO, L. H. A. Mineração de critério de credit scoring utilizando algoritmos genéticos. Anais do VI Simpósio Brasileiro de Automação Inteligente, p. 463–466, Bauru, SP, 2003.

