1. Introdução

Predição de valores de ações é uma tarefa desafiadora na área de predição de séries temporais financeiras, devido à grande quantidade de variáveis que envolvem essas predições. No passado, métodos baseados em estatística foram propostos para atacar este problema, como o modelo autoregressivo (AR), o modelo autoregressivo de médias móveis (ARMA) e o modelo autoregressivo integrado de médias móveis (ARIMA). Esses são modelos lineares que frequentemente se mostram inadequados para predição no mercado de ações, já que as ações possuem muitos ruídos e são séries temporais não-estacionárias. Recentemente, o uso de abordagens não lineares têm sido propostas, tais como *AutoRegressive Conditional Heteroskedasticity* (ARCH), *Generalized AutoRegressive Conditional Heteroskedasticity* (GARCH), Redes Neurais Artificiais (RNAs) e Máquinas de Vetor Suporte (SVMs). Todos esses métodos têm mostrado resultados promissores (Yeh et al,2011).

O objetivo deste trabalho é avaliar o comportamento das RNAs e SVMs, ainda pouco conhecidas no meio financeiro, para dois índices do mercado de ações. Foram utilizadas implementações desses algoritmos na ferramenta Weka, bastante utilizada na área de aprendizado de máquina e mineração de dados, que facilita bastante a construção dos modelos e análise dos resultados. Os resultados obtidos com as SVMs se mostraram bastante promissores.

O trabalho está dividido como segue: Na Seção 2, são descritas as séries temporais. Na Seção 3, são descritos os regressores, modelos utilizados para predição de ações, e os dois algoritmos utilizados para construção dos regressores – Backpropagation para indução de uma Rede Neural Artificial e as SVMs. Na Seção 4 são descritos os dados utilizados e os resultados obtidos. Por fim, na Seção 5 são descritas algumas conclusões do trabalho realizado.

2. Séries Temporais

Segundo Shumway & Stoffer (2003) séries temporais são uma sequência de observações de dados feitas ao longo do tempo, portanto essas observações possuem o que é chamado de dependência serial. As correlações incluídas nos dados ao longo do tempo podem restringir a aplicabilidade de muitos métodos estatísticos convencionais. Por isso, são exigidas técnicas estatísticas diferentes para lidar com a correlação serial dos dados, sendo tais técnicas chamadas análise de séries temporais.

No mercado de ações, podemos modelar a evolução do preço de uma determinada carteira de ativos ou de um determinado ativo ao longo do tempo por uma série temporal. As séries temporais desse tipo de fenômeno têm algumas características próprias, como por exemplo, alta volatilidade dos preços e comportamento não linear ao longo do tempo. Essas características tornam necessária a utilização de modelos robustos de regressão, que sejam capazes de responder rápido a constantes mudanças.

Segundo Tsay (2005), análise de séries temporais financeiras está preocupada com a teoria e a prática de valoração de bens através do tempo. Como estamos lidando com um conjunto de dados referentes a um fenômeno econômico, existem muitas variáveis que podem interferir na variação dos dados ao longo do tempo. Por exemplo, condições políticas em um país e mudança na taxa de juros são condições macroeconômicas que podem influenciar o valor das ações da empresa. Devido a muitas condições não mensuráveis ou mesmo grande número de variáveis, torna-se inviável construir um modelo matemático que levasse em consideração tais variáveis. Por isso, torna-se útil aplicar técnicas para análise de séries temporais que tentem

encontrar um padrão de comportamento utilizando um regressor, a partir dos valores anteriores da própria série.

Fonte: http://sdw.ecb.europa.eu

Figura 1: Exemplo de série temporal financeira,

3. Regresssores

Para construir o regressor, também denominado modelo, podem ser usados métodos clássicos para problemas de regressão em séries temporais, como os modelos AutoRegressivos, Média Movel, AutoRegressivos de Média Móvel, além dos modelos ARCH e GARCH. Estes dois últimos muito utilizados em fenômenos econométricos, pois foram construídos para trabalhar com heteroscedasticidade, ou seja, quando a variabilidade dos dados não é sempre a mesma.

Além dos métodos citados, existem métodos computacionais de aprendizado de máquina, que buscam por um modelo para oferecer predições a dados não vistos. Tais algoritmos de aprendizado também podem ser usados para construir modelos a partir de dados existentes. Dois algoritmos interessantes são o de backpropagation para construção de uma Rede Neural *Multilayer Perceptron* , que tem a propriedade de construir uma combinação não-linear dos atributos de entrada, e o algoritmo SMO para construir uma máquina de vetores suporte (SVM) .

3.1. Redes Neurais

Uma Rede Neural Artificial (RNA) é um processador massivo distribuído paralelamente construído por unidades simples de processamento que tem pré-disposição natural para acumular conhecimento empírico e transformar-lo de um modo que este seja disponível para uso (Haykin, 2005).

Para o problema em questão utilizamos como arquitetura da RNA a *MultiLayer Perceptron*, que possui *n* atributos na camada de entrada e *m* classes, que são os neurônios de saída e no mínimo uma camada intermediária. A rede muda o peso dos seus neurônios após cada dado ser processado, baseando-se no erro da saída quando comparado ao resultado esperado. Isto é feito através do *backpropagation*, que é um algoritmo de aprendizado em redes neurais, sendo este o algoritmo mais usado em redes neurais para tarefa de regressão (Russel & Norvig,

2005). O algoritmo de *backpropagation* busca a minimização do gradiente do erro de saída, formado pelos pesos de cada neurônio.

Redes *Multilayer Perceptron* têm uma camada de entrada, onde os padrões são apresentados à rede, uma ou mais camadas intermediárias (também chamadas de camadas escondidas) que fazem a maior parte do processamento e uma camada de saída que retorna o resultado.

Na figura 2 é ilustrada uma RNA com 2 atributos (neurônios) na camada de entrada, 1 neurônio na camada de saída e 5 neurônios na camada intermediária.

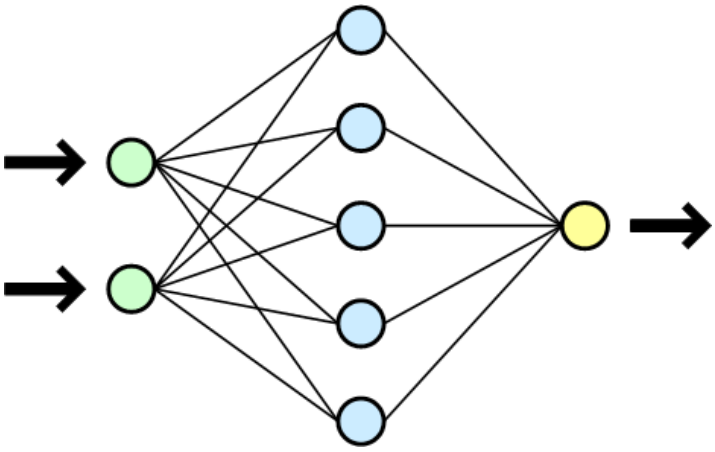

Figura 2: Ilustração de uma RNA

3.1. SVM

Máquinas de Vetores Supporte (SVM) são embasados na teoria de aprendizado estatístico (Vapnik, 1995). A tarefa de aprendizado de uma SVM envolve a otimização de uma função de custo convexa, pois não existem falsos mínimos locais para complicar o processo de aprendizado de uma SVM (Campbell, 2000).

A indução de uma SVM envolve resolver um problema de otimização de programação quadrática. Para simplificar o processo de otimização para construção (aprendizado) de uma SVM, foi proposto o algoritmo SMO – *Sequential Minimal Optimization* –, que transforma o processo de aprendizado da SVM em uma série de problemas de otimização menores (Platt, 1999). Há muitos resultados promissores utilizando este algoritmo de aprendizado (Campbell, 2000).

4. Aplicações e Resultados

4.1. Metodologia

Para avaliar os dois algoritmos de aprendizado, utilizou-se duas bases de dados disponíveis em http://robjhyndman.com/TSDL, que é um repositório de séries temporais online. A base de dados mostra flutuações de diversas bolsas do mercado de ações internacionais todos os dias do ano, onde houve a marcação de 1987 a 1997.

Foram escolhidos, aleatoriamente, dois índices do mercado de ações, que foram o DAX e o Nikkei 225. O DAX é um índice do mercado de ações de empresas alemãs na Bolsa de Valores de Frankfurt, e o Nikkei 225 é um índice do mercado de ações de empresas japonesas na Bolsa de valores de Tóquio.

Antes de usar os dados, é necessário realizar o pré-processamento, para que os modelos possam ter maior eficiência. O filtro utilizado foi o retorno logarítmico, onde $R_i = \ln \frac{x_i}{x_{i-1}}$,

sendo x_i é o valor do índice no tempo i.

Para trabalhar tanto com o modelo de previsão por redes neurais como para o modelo SVM, precisa-se primeiro definir dois parâmetros, que são quantas unidades de tempo serão previstas pelo modelo (janela de previsão), e o segundo diz respeito a quantidade de unidades de tempo serão usadas na previsão (parâmetros). Encontrar valores tanto para os parâmetros como para a janela de previsão é um exercício prático, portanto foram testados possibilidades de 5 a 3 unidades de tempo para os parâmetros, e de 1 a 3 para a janela de previsão.
A seguir estão os modelos que apresentaram menor erro relativo quadrático médio.

Modelo	Índice	Janela de previsão	Parâmetros
Redes Neurais	DAX_t	DAX_t	DAX_{t-4} ,DAX_{t-3} ,DAX_{t-2}, DAX_{t-1}
	NIK_t	NIK_t	NIK_{t-4} , NIK_{t-3} , NIK_{t-2}, NIK_{t-1}
SVM	DAX_t	DAX_t	DAX_{t-3} ,DAX_{t-2}, DAX_{t-1}
	NIK_t	NIK_t	NIK_{t-3} , NIK_{t-2}, NIK_{t-1}

Tabela 1: Relação de janela de previsão e parâmetros por índice e algoritmo

Para realizar o pré-processamento, (retorno logarítmico), o processamento (aplicação dos regressores) e pós processamento (avaliação dos resultados e estimação das taxas de erro), foi utilizado o software Weka. O Weka é um software desenvolvido pela Universidade de Waikato, na Nova Zelândia, cujo nome vem de *Waikato Enviroment for Knowledge Analysis*. O Weka é uma ferramenta promissora na área de aprendizado de máquina e mineração de dados, pois reúne uma série de algoritmos de aprendizado implementados, diversas ferramentas de pré-processamento, e possibilidade de avaliar e validar os modelos em uma interface bem intuitiva, além de ter uma licença de uso GNU (General Public Licence) (Witten & Frank, 2005).

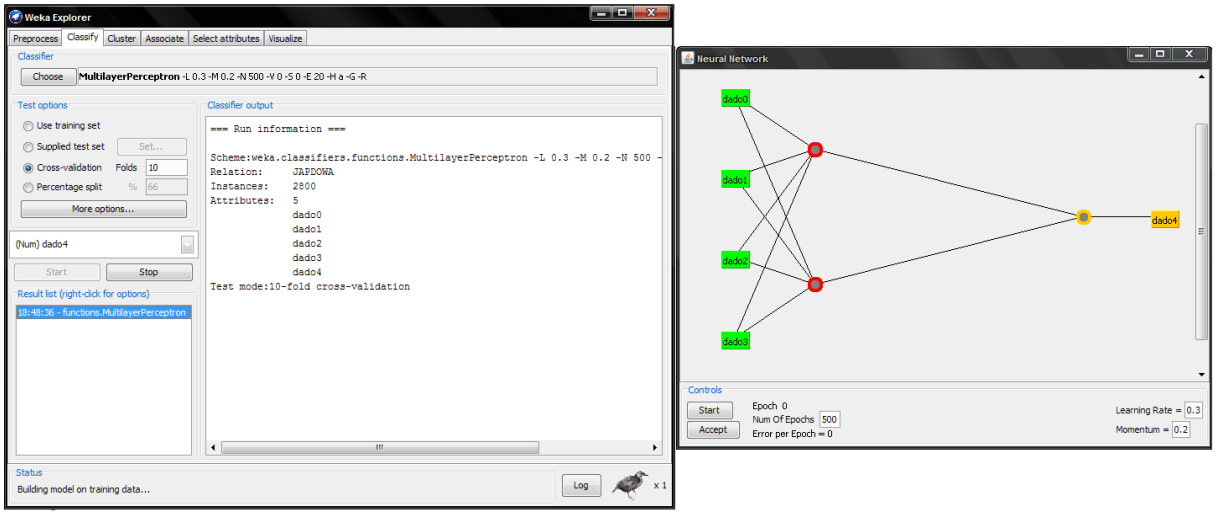

Figura 3: Interface de utilização do Weka para Redes Neurais

Os regressores foram gerados a partir do retorno logarítmico diário, que totalizaram 2800 instancias. Para as redes neurais foi considerado o número de neurônios escondidos $e = \sqrt{mn}$, onde *n* é o numero de parâmetros de entrada e *m* o número de saídas, e também

foram feitos testes com o dobro de neurônios escondidos, para avaliar como os as taxas de erros iriam reagir.

A avaliação dos resultados foi baseada em 3 medidas de erros principais: o r-quadrado, que mede a proporção da variação da previsão que é explicada pelas entradas, ou seja, quão maior for o valor do r-quadrado, melhor o modelo de regressão pode explicar o fenômeno. Vale ressaltar que o r-quadrado está compreendido entre -1 e 1, sendo -1 um grau de correlação perfeita mas de direção oposta, e 1 um grau de correlação perfeita e mesma direção. A segunda medida de erro utilizada é o erro relativo absoluto médio (ERAM), que é dado por:

$$ERAM(\%) = \frac{1}{n}\sum_{i=1}^{n}\frac{|x_i - \hat{x}_i|}{x_i} \times 100$$

A terceira medida é o erro relativo quadrático médio (ERQM), que é dado por:

$$ERQM(\%) = \frac{1}{n}\sum_{i=1}^{n}\frac{(x_i - \hat{x}_i)^2}{x_i} \times 100$$

Onde n é número de instâncias utilizadas para calcular o erro total, x_i é o valor previsto pelo

regressor para a instancia i, e $\hat{x}_i$ é o valor real da instância i.

Como referência para as taxas de erro geradas, o ideal é que o erro relativo quadrático médio e o erro relativo absoluto médio se aproximem de 0% e o r-quadrado se aproxime de 1, para que o valor previsto seja o mais próximo do valor real.

Foram usados dois métodos para gerar as taxas de erros, o método hold out, e o método de *k-fold cross validation* com 10 partições. O método de *hold out* consiste em separar os dados em dois grupos, um chamado grupo de treinamento e o outro grupo de teste. Primeiramente os regressores são induzidos com os parâmetros dados, e a cada iteração o valor gerado pelo regressor é comparado com o valor real, e então o regressor se reajusta, afim de minimizar a taxa de erro. Em seguida, o regressor já treinado tem suas taxas de erro calculadas com os dados do grupo de teste. Foram utilizados *hold outs* de 75%/25% e 85%/15% para divisão entre porcentagem de dados utilizados no treinamento/teste.

Já no método de *k-fold cross validation* a base de dados é dividida em *k* partições (k=10), e depois o regressor é induzido com *k-1* partes da base de dados e uma das partes da base de dados é utilizada para calcular as taxas de erro. O processo é repetido *k* vezes para que todas as partes sejam utilizadas tanto induzindo o regressor como para calcular as taxas de erro. No final, é calculado uma média aritmética das *k* taxas de erro.

Modelo	Ìndice	Método de erro	ERQM (%)	ERAM (%)	R-quadrado
Redes Neurais	DAX	Hold out 0.75	6.00426	4.84064	0.89745
		Hold out 0.85	7.02458	6.14001	0.83547
		k-fold	6.01561	5.42430	0.86339
	NIK	Hold out 0.75	4.94537	4.05812	0.93698
		Hold out 0.85	4.61362	3.95425	0.90128
		k-fold	4.80023	4.00325	0.91567
SVM	DAX	Hold out 0.75	2.73719	3.32832	0.92588
		Hold out 0.85	2.39936	1.86587	0.97655
		k-fold	3.23571	2.10019	0.95668
	NIK	Hold out 0.75	4.29636	3.38232	0.93007
		Hold out 0.85	4.35505	3.62541	0.94552
		k-fold	5.08375	4.00258	0.92421

Tabela 2: Medidas de erro por índice e algoritmo de aprendizado

Conforme a tabela acima demostra, os melhores resultados foram obtidos no índice DAX, utilizando o SVM, tendo os menores valores de ERQM e ERAM, e os valores mais próximos de 1 para o R-quadrado, demostrando assim, a adequação do modelo.

Foi realizado o teste estatístico t-pareado para verificar se o erro quadrático médio dos regressores gerados pelos algoritmos de aprendizado são diferentes, com um determidado grau de confiança. Para a base de dados DAX, podemos afirmar com 95% de confiança que o SMO obteve melhor resultado que a Rede Neural usando backpropagation como algoritmo de aprendizado. Já para a base NIK, o mesmo não se pode afirmar. Ainda sim, os resultados são considerados promissores.

7. Conclusões

Podemos notar que os regressores gerados por técnicas de inteligência artificial como as redes neurais e o SVM podem nos fornecer poderosas ferramentas para a predição de variações no mercado de ações. Contudo, devemos observar que estas técnicas são relativamente novas, e que não devem ser usadas para substituir técnicas mais consolidadas. No entanto, tais ferramentas, como sugere Samanez (2007), são importantes complementos para auxiliar investidores em projetos de decisão de investimentos, sobretudo em cenários de alta incerteza.

Segundo Chen (1997), o campo de aplicação de novas ferramentas computacionais para modelar fenômenos econômicos, como séries temporais de indíces do mercado de ações ainda é um campo que se tem muito a explorar, que surgem como alternativas aos métodos estatísticos clássicos, como o ARIMA e o GARCH.

Referências

CAMPBELL, C. *Algorithmic Approaches to Training Support Vector Machines: A Survey*. In: Proc. 8th European Symposium on Artificial Neural Networks, Belgium. 2000

CHEN, A.; LEUNG, M.T. & DAOUK, H. - *Application of neural networks to an emerging financial market, Forecasting and trading the Taiwan Stock Index,* Vol. 30, p. 901-923. 69, 1997.

HAYKIN, S. *Neural Networks A Comprehensive Foundation* Second Edition; Pearson; Hamilton, 2005.

PLATT, J.C. *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*. Chapter. In: Scholkopf,B. et al (eds). Advances in Kernel Methods: Support Vector Learning. MIT Press. 1999.

REZENDE, S. O. *Sistemas Inteligentes - Fundamentos e Aplicações*; Primeira Edição;Barueri; Manole; 2005.

RUSSEL E NORVIG. *Inteligência Artificial*; Segunda Edição; São Paulo; Elsevier; 2004.

SAMANEZ, C.P. *Gestão de Investimentos e Geração de Valor*. São Paulo: Pearson, 2007.

SHUMWAY, R. H.\& STOFFER, D. S. *Time Series Analysis and Its Applications With R Examples;* Second Edition; Springer, 2003.

TSAY, R. S. *Analysis of Financial Time Series* Second Edition; New Jersey, Wiley-Interscience, 2005.

VAPNIK, V.N. *The nature of Statistical Learning Theory*. Springer-Verlag, New York. 1995.

YEH, C.-Y.; HUANG, C.-W.; LEE, S.-J. *A multiple-kernel support vector regression approach for stock market price forecasting*. Expert Systems with Applications, 38, pp. 2177–2186, 2011.