

Trabalho Final de Conclusão de Disciplina

MPE/MPF - EESP

Disciplina: Métodos Quantitativos Computacionais Aplicados a Economia e Finanças

Professor: Ernesto Coutinho Colla

Alunos: Gustavo Falcim, Raphael Castro da Costa Ferreira e Rodolfo Rosina

Data: 28/06/2015

Descrição do Problema

No início de cada dia no mercado, operadores e analistas se reúnem para comentar os eventos relevantes do dia que se inicia, bem como o fechamento do dia anterior, sendo uma das variáveis centrais o índice da bolsa brasileira (Ibovespa). Observa-se, também, os índices de fechamento de bolsas de outros mercados, sobretudo o americano, bem como das bolsas asiáticas, as quais por uma questão de fuso horário, refletem o que aconteceu entre o fechamento das bolsas brasileiras e americanas.

Motivação do Problema

Além dos fechamentos, os agentes de mercado costumam utilizar-se de projeções como parâmetros para suas operações. Algumas projeções/recomendações são realizadas por modelos ou por análises grafistas. O nosso objetivo é realizar previsões fazendo uso de modelos de redes neurais, que permite a não linearidade nas relações entre as variáveis, diferenciando, dessa forma, de métodos econométricos lineares.

Modelagem

Input: dados diários de fechamento do índice Ibovespa (BVSP), variável a ser projetada, bem como os fechamentos dos índices de mercados americano e asiáticos, Dow Jones (DJI), Nikkei (N225) e Shanghai (000001.SS).

Drivers:

Obtenção dos dados:

Usamos o pacote quantmod função getSymbols, para qual há de se fornecer o ticker do ativo que se quer. A função, via XML, do YAHOO! Finance, traz uma série de informações diárias de cada ativo, como por exemplo preço de abertura, fechamento, volume de negociações, maior preço e menor preço.

Entendimento dos dados:

Para cada ativo, tem-se um objeto com: abertura, máxima, mínima, volume, fechamento e ajuste, no formato xts. Horizonte temporal de 2007 até a data atual, sendo que há dias que tem mercado em alguns países, mas não em outros. Além disso, interessa-nos apenas o fechamento de cada índice.

Data Preparation:

Manipulamos de tal forma que ficamos apenas com os dados de fechamento, ainda em xts.

Posteriormente, por meio da função merger (pacote xts), unimos os vetores do fechamento de cada bolsa, seguindo as datas do Ibovespa, transformando em um data.frame e

adicionamos uma coluna de datas (Date), construída a partir do índice da fusão entre os vetores.

Em seguida, percebemos que o número de NaN, quando não houve pregão pelo menos em um dos mercados, seria por volta de 350, o que representaria cerca de 13% da amostra (gira em torno de 2724 observações, dependendo do Begin Date). Dessa forma, preferimos usar a fechamento do dia útil imediatamente anterior, pois, embora não houvesse mercado para um determinado índice, pode ter havido para os demais, e não desprezaríamos essa informação na modelagem. Para tanto, desenvolvemos um loop (for) que percorria as linhas e colunas substituindo os NaN pela cotação do dia útil anterior.

Com a tabela pronta, sem NaN, criamos as defasagens para BVSP e DJI, usando a função Lag(.), do quantmod, e recombinaamos a tabela de dados, retirando DJI em t=0 e adicionando as variáveis defasadas.

Dividimos os dados em treinamento (80% do volume de dados, independente de qual volume seja) e teste, para ver a performance do modelo.

Modelagem

Usamos redes neurais de apenas uma camada para modelar o BVSP diário, com o objetivo de prever qual seria o índice de fechamento no final do dia (pensando que o modelo será rodado antes da abertura do pregão no Brasil, idealmente. Todavia, isso não impede que o modelo seja rodado em outro momento, quando alguma das outras bolsas, ou até mesmo a bolsa brasileira já esteja em funcionamento – o getSymbols pega a última cotação de um ativo para aquele instante em que ela foi acionada).

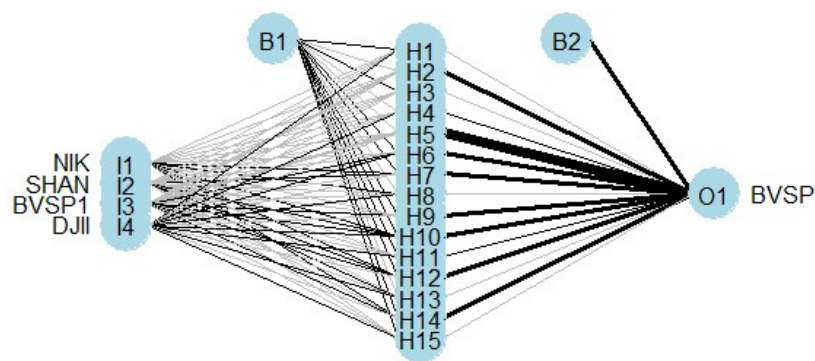
Os motivos de se usar redes neurais são: oportunidade de aprender este tipo de modelagem (machine learning), ela ser do tipo não linear (diferentemente dos métodos econométricos tradicionais), há vários pacotes já desenvolvidos e ela permite uma automatização mais fácil (usando variáveis explicativas, quando se compara os pacotes de time series). Utilizamos o pacote nnet, rodando BVSP contra BVSP t-1, DJI t-1, Nikkey e Shanghai, usando tamanho de 15, para interação na camada oculta, com 1 output, de tal sorte que a especificação do modelo ficou: 4 – 15 – 1, com 91 pesos.

```
a 4-15-1 network with 91 weights
options were - linear output units decay=0.05
b->h1 i1->h1 i2->h1 i3->h1 i4->h1
0.06 -0.06 -0.06 -0.06 0.00
b->h2 i1->h2 i2->h2 i3->h2 i4->h2
-0.12 -1109.38 -203.73 -4681.45 -1224.79
b->h3 i1->h3 i2->h3 i3->h3 i4->h3
0.01 -0.04 -0.05 -0.06 -0.03
b->h4 i1->h4 i2->h4 i3->h4 i4->h4
-0.06 -0.06 -0.01 -0.07 0.03
b->h5 i1->h5 i2->h5 i3->h5 i4->h5
-0.19 -2156.73 -376.41 -5915.28 -2008.64
b->h6 i1->h6 i2->h6 i3->h6 i4->h6
0.07 -0.04 -0.06 0.05 0.04
b->h7 i1->h7 i2->h7 i3->h7 i4->h7
0.07 0.08 0.07 0.00 -0.04
b->h8 i1->h8 i2->h8 i3->h8 i4->h8
```

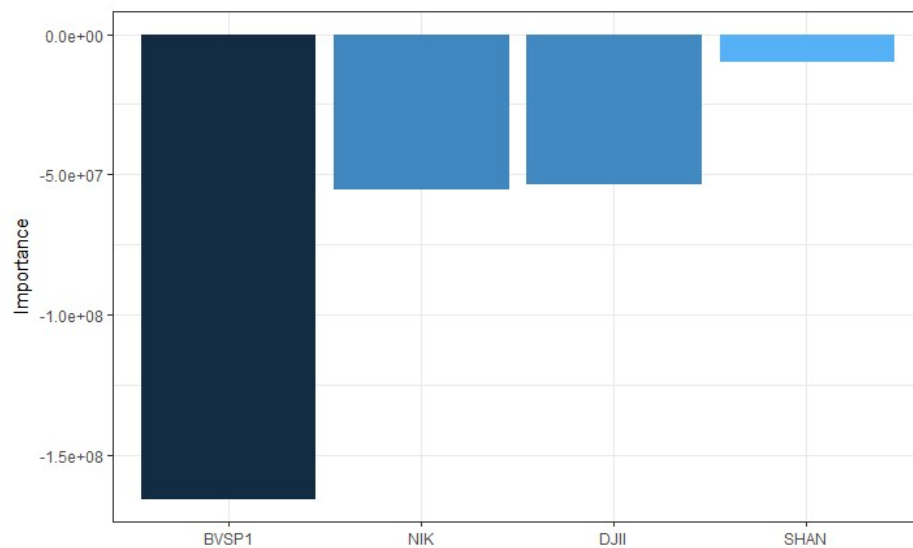
-0.04	0.00	0.00	-0.03	0.07				
b->h9	i1->h9	i2->h9	i3->h9	i4->h9				
0.02	-0.06	-0.01	0.07	-0.01				
b->h10	i1->h10	i2->h10	i3->h10	i4->h10				
0.07	0.02	0.02	0.06	0.00				
b->h11	i1->h11	i2->h11	i3->h11	i4->h11				
0.08	-0.03	0.00	-0.02	0.02				
b->h12	i1->h12	i2->h12	i3->h12	i4->h12				
0.04	0.01	0.08	0.01	-0.01				
b->h13	i1->h13	i2->h13	i3->h13	i4->h13				
-0.04	-0.07	0.06	-0.04	0.08				
b->h14	i1->h14	i2->h14	i3->h14	i4->h14				
0.02	0.08	-0.02	0.01	-0.01				
b->h15	i1->h15	i2->h15	i3->h15	i4->h15				
0.01	-0.01	-0.04	-0.07	0.02				
b->o	h1->o	h2->o	h3->o	h4->o	h5->o	h6->o	h7->o	
h8->o	h9->o							
8108.58	-0.07	9143.32	-0.03	0.04	20776.08	8108.63	8108.58	
-0.03	8108.64							
h10->o	h11->o	h12->o	h13->o	h14->o	h15->o			
8108.58	0.01	8108.53	-0.03	8108.59	-0.03			

Com o pacote NeuralNetTools, é possível observar graficamente como ficou o modelo, conforme abaixo:

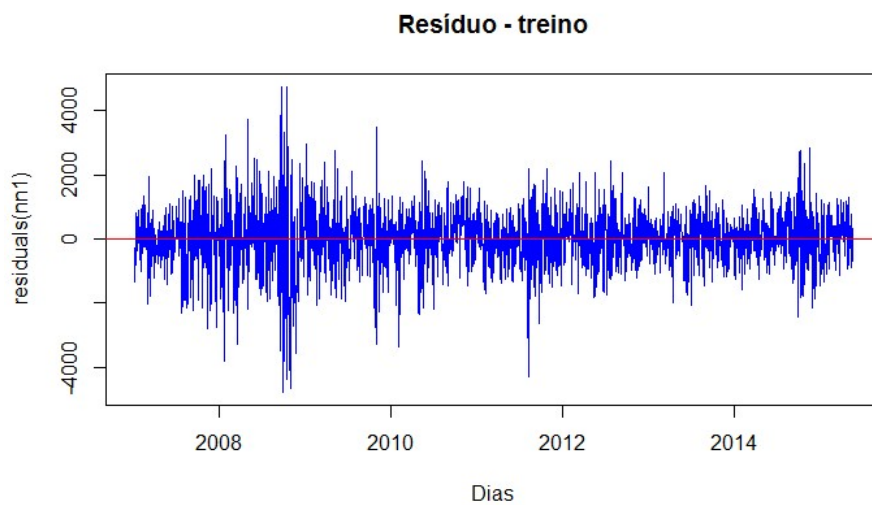
Arquitetura da rede



Importância de cada variável dentro do modelo



Resíduos do treino

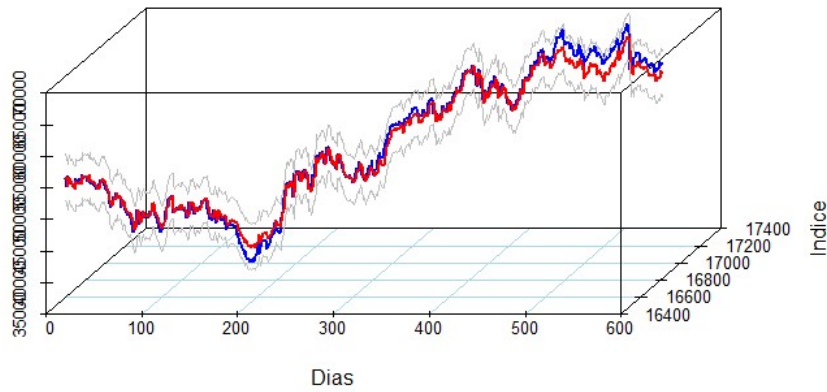


Média do desvio absoluto = 673.9826
 Coeficiente de Variação (média/desv. Padrão) = 0.1468525

Previsão com a base de teste (20% das observações) e criação dos intervalos da faixa de previsão (com meio desvio padrão da série projetada, em relação ao valor projetado para aquele dia), bem como erro quadrático médio, máximo e mínimo.

Abaixo o gráfico do projetado (vermelho) e real (azul), bem como os intervalos de projeção.

BVSP:Previsao X Real



Abaixo o tail da tabela de projetado, real e intervalos (superior e inferior)

```

Date PrevBVSP RealBVSP up low
2017-06-16 2017-06-16 61860.00 61626 65775.82 57944.19
2017-06-19 2017-06-19 61547.02 62014 65462.84 57631.21
2017-06-20 2017-06-20 61960.50 60766 65876.32 58044.68
2017-06-21 2017-06-21 60621.45 60762 64537.26 56705.63
2017-06-22 2017-06-22 60619.63 61272 64535.45 56703.82
2017-06-23 2017-06-23 61169.59 61087 65085.40 57253.77

```

>

Ou nesta tabela:

Arquivo Editar Ajuda

	row.names	Date	PrevBVSP	RealBVSP	up	low	var7
1	2017-06-16	2017-06-16	61860	61626	65775.82	57944.19	
2	2017-06-19	2017-06-19	61547.02	62014	65462.84	57631.21	
3	2017-06-20	2017-06-20	61960.5	60766	65876.32	58044.68	
4	2017-06-21	2017-06-21	60621.45	60762	64537.26	56705.63	
5	2017-06-22	2017-06-22	60619.63	61272	64535.45	56703.82	
6	2017-06-23	2017-06-23	61169.59	61087	65085.4	57253.77	
7							
8							
9							
10							
11							
12							
13							
14							

Em seguida, ele printa o último valor no console

```
> print(finall[nrow(finall), "PrevBVSP"])  
[1] 61169.59
```

Também criamos uma tabela de saída, na qual colocamos, além dos erros quadráticos da parte de treino, os erros quadráticos e resultados da parte de teste, bem como o último valor projetado.

Arquivo Editar Ajuda							
	Erro. Qd. Medio Treino	Desv Pad Treino	Erro Qd. Medio Teste	Max Erro Qd	Min Erro Qd	ProjBVSP	var7
1	673.9826	0.1468525	660.4046	5967.702	2.888019	61169.59	
2							
3							
4							
5							
6							

Outcomes:

Como outcomes, além dos já apresentados, gráfico, tabelas e erros quadráticos, que também servem para o monitorar a performance do modelo, há o script que criar uma planilha em excel na pasta Database (nome final1.xlsx). Nesta, contém uma aba com o real, projetado e bandas de desvio, e outra aba com a tabela com os erros quadráticos, bem como o último valor projetado.

Referências Bibliográficas

Todos os *papers* descritivos dos pacotes que foram utilizados.

Sites de suporte e discussão tais como <https://pt.stackoverflow.com/> e <https://github.com>