

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Análise de *Kernels* para regressão dos dados do mercado *Forex*

Felipe de Almeida Duarte¹

Programa de Pós-Graduação em Pesquisa Operacional, Unifesp, São José dos Campos, SP.

Renato Cesar Sato²

Programa de Pós-Graduação em Pesquisa Operacional, Unifesp, São José dos Campos, SP.

Ana Carolina Lorena³

Programa de Pós-Graduação em Ciência da Computação, Unifesp, São José dos Campos, SP.

1 Introdução

O mercado de câmbio (Forex) é um mercado global descentralizado para negociações de moedas. As Máquinas de Vetores de Suporte (*Support Vector Machine*- SVM) [3] é uma ferramenta que, através da regressão, pode apresentar uma previsibilidade dos dados futuros nesse mercado. Para tal, a SVM utiliza de funções que descrevam o comportamento dos dados e, ao mesmo tempo, os mapeia em um espaço com o produto interno, reduzindo o custo computacional do algoritmo. Tais funções são denominadas de funções *kernels*.

2 Ferramentas e Metodologia

Utilizamos o software [2] e seu pacote *kernelab* em um conjunto de seis anos de negociação dos pares de moedas euro/dólar americano (EUR/USD). Os cinco primeiros anos serviram como treinamento do algoritmo e o sexto ano serviu como teste da previsão. Para avaliar a precisão realizamos testes com seis *kernels*: RBF, polinomial, linear, ANOVA, Laplaciano e o Bessel.

Com exceção do Laplaciano, todos os demais *kernels* apresentaram uma boa precisão para previsão dos dados do sexto ano, destacando-se os *kernels* mais próximos de funções lineares como o próprio *kernel* linear e a polinomial de grau um. Buscando encontrar uma variação dos *kernels* utilizados, testamos um novo kernel desenvolvido conforme as propriedades sugeridas por [1] através da função:

$$k(x, y) = \log(k \cdot \langle x, y \rangle + d) \quad (1)$$

¹felipeaduart@gmail.com

²rzsato@unifesp.br

³aclorena@gmail.com

Em um primeiro momento, aparenta que apenas calculamos o logaritmo de um *kernel* polinomial de grau um porém, a função logarítmica alterou a mecânica do *kernel* em questão devido ao seu domínio restrito. A escolha dos parâmetros deve ser realizada de modo que os dados utilizados pertençam ao domínio da função *kernel*. Ao analisarmos os parâmetros k e d podemos observar que ambos exercem um grande peso na definição da função *kernel*; ambos devem ser maiores que zero e, para que a matriz *Gram* deste *kernel* seja definida positiva, d e k devem ser escolhidos de tal forma que $\langle x_i, x_j \rangle \geq \frac{k-2d}{k}$.

Pode-se verificar que, ao manter d fixo e fazer $k \rightarrow 0$, o lado direito da desigualdade acima tende a $-\infty$; assim como, fixando k e fazendo $d \rightarrow \infty$ obtemos o mesmo resultado; sendo assim, independente do resultado de $\langle x_i, x_j \rangle$, existirá sempre um par $d, k \in \mathbb{R}_+^*$ tal que a desigualdade seja verdadeira.

Porém, o tempo de treinamento dos dados foi prejudicado pela função logarítmica. Com os *kernels* polinomiais e o linear, a etapa de aprendizado demorou, respectivamente, 0,64 e 0,26 segundos enquanto que com o logaritmo a etapa de processamento demorou 248,94 segundos. Ou seja, apesar dos bons resultados encontrados com a modificação, o *kernel* se torna inviável devido ao aumento do tempo para tomada de decisão em um mercado com preços dinâmicos.

3 Conclusões

Realizamos alteração em um *kernel* para analisar como ele pode afetar a predição do algoritmo e como os parâmetros das funções devem ser escolhidos. Observamos que o domínio da função pretendente, juntamente com os dados de entrada do algoritmo, definirão se a mesma atingirá as propriedades para se definir um *kernel*.

A função *Kernel* desenvolvida na pesquisa apresentou resultados compatíveis com os *kernels* linear e polinomial. No entanto, nosso modelo apresentou maior tempo na etapa de aprendizado, aproximadamente 550 vezes a mais que os outros *kernels*, devido o aumento da complexidade da função logarítmica no algoritmo.

Agradecimentos

Os autores agradecem a CAPES pelo apoio no desenvolvimento desse trabalho.

Referências

- [1] C. M. Bishop. Pattern Recognition and Machine Learning. Springer Science + Business Media, LLC, New York, 2006.
- [2] R Core Team. R: A Language and Environment for Statistical Computing. Disponível em: <https://www.R-project.org/>. Vienna, Austria, 2016.
- [3] B. Schölkopf, A. J. Smola. Learning with Kernels: Support Vectors Machine, Regularization, Optimization, and Beyond. The MIT Press, London, 2002.