Research on the Price Features of Oil Stochastic Model Based on the Continuous Jump Model

Mengmeng Hou [1,*] and West Gabllian [2]

[1] School of Management, Wuhan University of Technology, Wuhan, China, 430074
[2] School of Engineering, University of Florida, Gainesville, FL, US
[*] Corresponding author: 2426935187@qq.com

Abstract. Aiming at calculating the price changes under the price features of oil stochastic model, the continuous jump model is proposed in this paper for data processing. The procedure is flexible, may be used with market prices of any oil contingent claim with closed form pricing solution, and easily deals with missing data problems. The results show that the accuracy can thus be improved overall the proposed system substantially.

1 Introduction

This paper develops a parsimonious three-factor model of the term structure of oil futures prices that can be easily estimated from available futures price data. In addition, it proposes a new simple spreadsheet implementation procedure. The procedure is flexible, may be used with market prices of any oil contingent claim with closed form pricing solution, and easily deals with missing data problems. The approach is implemented using daily prices of all futures contracts traded at the New York Mercantile Exchange between 1991 and 2001. In-sample and out-of-sample tests indicate that the model fits the data extremely well. Though the paper concentrates on oil, the approach can be used for any other commodity with well-developed futures markets [1-2].

Enlightening as they may be, these stochastic models have been adopted rather slowly by practitioners. One possible reason for this could be that even though two-factor models behave reasonably well most of the time, for some market conditions they behave poorly, making daily estimations somewhat unreliable[3-5]. Also, most parameter estimation procedures proposed in the literature are rather involved and require extensive data aggregation, which translates into substantial information loss. The Kalman filter approach, however, has relatively high implementation requirements. One of the main issues is the handling of missing observations in the data, which is particularly important when using financial instruments, like futures, that only trade for specific maturities. In typical implementations of this procedure contracts with close, but different maturities are grouped together to complete the data sets used in the estimation. The consequence of these groupings is that a substantial amount of data is discarded with its associated information loss. This problem is particularly acute when new contracts are introduced in the market and there are no complete time series data for a given maturity. For example, oil markets currently trade futures contracts for over thirty-five different maturities while only a few years ago they only traded futures contracts for less than twenty different maturities [6].

Current implementations of the Kalman filter methodology for this type of models require linearity in the pricing expressions, which restricts its applications to linear payoffs such as futures. Nonlinear Kalman filter estimations do exist in the literature, but are much harder to apply. Even though in the current implementation of our model we only use futures prices in the estimation, it would be easy to include contracts with nonlinear payoffs such as options.

2 The Algorithm and Data Analysis

This modification considerably simplifies the estimation procedure since we only have to estimate seven parameters. In what follows we will be using this modified model in all the calculations. The basic equation for the routing algorithm is shown below [7-8]:

$$\overline{x_i} = \frac{x_i - b_i}{a_i - b_i} \tag{1}$$

Experimental data to determine, we can also experience the value of the formula (2) the decision.

$$n = \log_2 m \tag{2}$$

Hidden node output is calculated as follows:

$$h_j = f\left(\sum_{i=1}^{m} w_{ij}x_i - \theta_j\right) \tag{3}$$

The output of the output node is calculated as follows:

$$f\left(\sum_{i=1}^{m} w_{ij}x_i - \theta_j\right) = f\left(f\left(\theta_j\right)\right) \tag{4}$$

Where in θ is an output node threshold.
Put Equation (3) into Equation (4), then we can get the S-type function:

$$f\left(\sum_{i=1}^{m} w_{ij}x_i - \theta_j\right) = f\left(f\left(\theta_j\right)\right) \tag{5}$$

In the structure of GA algorithm, we can get the optimization equation as the following equation (6):

$$h_{\ j} = \exp\left(-\frac{\left\|X - C_j\right\|}{2b_j^2}\right), \quad j = 1, 2, ..., m \tag{6}$$

The output of the network is given as:

$$y_m\left(k\right) = wh = w_1h_1 + w_2h_2 + ... + w_mh_m \tag{7}$$

Assuming the ideal output is y (k), the performance index function is:

$$E\left(k\right) = \frac{1}{2}\left(y\left(k\right) - y_m\left(k\right)\right)^2 \tag{8}$$

The equation of basic function is as equation (9) as follows:

$$\partial_j(C_{ijkl}\partial_k u_l + e_{kij}\partial_k\varphi) - \rho\ddot{u}_i = 0 \tag{9}$$

Under the linear relationship, basic equation is shown in equation (2):

$$\partial_j(e_{ijkl}\partial_k u_l - \eta_{kij}\partial_k\varphi) = 0 \tag{10}$$

The linear differential equation can be expressed into the following simplified forms:

$L(\nabla, \omega) f(x, \omega) = 0$,

$$L(\nabla, \omega) = T(\nabla) + \omega^2 \rho \mathrm{J} \tag{11}$$

In which,

$$T(\nabla) = \begin{Vmatrix} T_{ik}(\nabla) & t_i(\nabla) \\ t_k^T(\nabla) & -\tau(\nabla) \end{Vmatrix}, \quad \mathrm{J} = \begin{Vmatrix} \delta_{ik} & 0 \\ 0 & 0 \end{Vmatrix},$$

$$f(x, \omega) = \begin{Vmatrix} u_k(x, \omega) \\ \varphi(x, \omega) \end{Vmatrix} \tag{12}$$

$$T_{ik}(\nabla) = \partial_j C_{ijkl}\partial_l, \quad t_i(\nabla) = \partial_j e_{ijk}\partial_k, \quad \tau(\nabla) = \partial_i \eta_{ik}\partial_k$$

Consider an infinite situation, we have the equation (5) in the following:

$$L^0 = \begin{Vmatrix} C^0_{ijkl} & e^0_{kij} \\ e^{0T}_{ikl} & -\eta^0_{ik} \end{Vmatrix} \tag{13}$$

Consider the propagation, instead the equation (13) with the following form:

$$C(\mathrm{x}) = C^0 + C^1(\mathrm{x}), \quad e(\mathrm{x}) = e^0 + e^1(\mathrm{x}), \quad \eta(\mathrm{x}) = \eta^0 + \eta^1(\mathrm{x}), \quad \rho(\mathrm{x}) = \rho_0 + \rho_1(\mathrm{x}) \tag{14}$$

Then we have equation (15):

$$C^1 = C - C^0, \quad e^1 = e - e^0,$$

$$\eta^1 = \eta - \eta^0, \quad \rho_1 = \rho - \rho_0 \tag{15}$$

The proposed procedure provides results reasonably close to those of the more formal Kalman filter estimation. For example, applying both estimation procedures to the weekly oil futures prices traded at NYMEX between 1990 and 1995, we find very similar mean square errors (MSE). Figure 1 shows that the estimated values for the state variables are almost indistinguishable between the two procedures. Figure 2 shows that the percentage differences between the estimated values, using both procedures, for the six and twelve-month futures prices are very small and generally smaller than 0.5%.

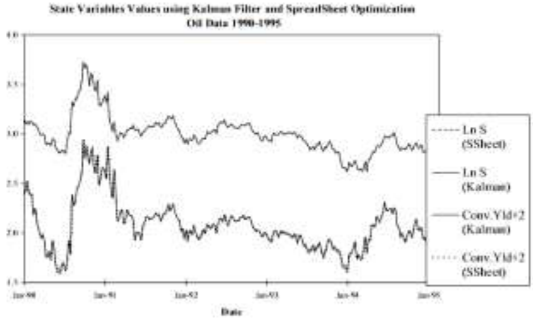

Fig. 1. State variables values of the two-factor model in Schwartz.

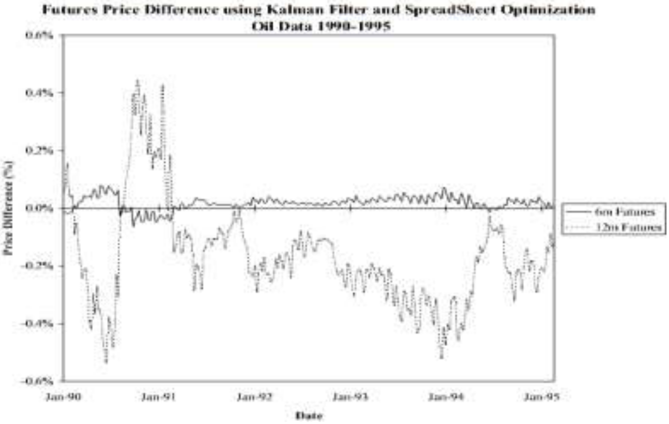

Fig. 2. Percentage differences in 6 month and 12 month futures price estimation.

3 Conclusion

To make our out-of-sample analysis we use Data-set 2 and analyze the performance of the model over the last three years (1999 to 2001). Our initial model calibration, for transactions in the first quarter of 1999, is done with Data-set 1. The model is then recalibrated on a quarterly basis (on the last day of March, June, September and December of each year) and parameter values are not updated for the following three months. This out-of-sample analysis is done using only past information, available at the time of the calibration and represents an upper bound for the errors. More frequent calibrations could reduce model errors, since more recent prices would be included.

Results for the three-year out-of-sample data are now presented. We start by noting that the model is very flexible in that it can deal with very different shapes of the term structure of futures prices. To illustrate this point we select arbitrary dates in the sample period in which futures oil prices exhibit either backwardation or contango. The consequence of these groupings is that a substantial amount of data is discarded with its associated information loss. This problem is particularly acute when new contracts are introduced in the market and there are no complete time series data for a given maturity. In this article we develop a parsimonious three-factor model of the term structure of futures oil prices, which fits the data extremely well. In addition, and very importantly, we propose an implementation procedure that significantly simplifies the estimation methods proposed in the literature. These factors make the proposed approach ideally suited for practical applications in the valuation and hedging of real and financial oil-contingent claims. The method can be used for other commodities as well; for example, we have also implemented the model using copper futures price data with similar results.

References

1. Wei Dai, Mostafa Bassiouni. An improved task assignment scheme for Hadoop running in the clouds. Journal of Cloud Computing, (2013), pp. 21-33.
2. Wei Kuang Lai, Yi-Uan Chen, Tin-Yu Wu, Mohammad S. Obaidat. Towards a framework for large-scale multimedia data storage and processing on Hadoop platform. The Journal of Supercomputing, (2014), pp. 681-696.

3. Zhijian Chen, Wenhai Luo, Dan Wu, Xiang Huang, Jian He, Yuanhuan Zheng, Di Wu. Exploiting application-level similarity to improve SSD cache performance in Hadoop. The Journal of Supercomputing, (2014), pp. 703-715.
4. Y. Mishing, in: Diffusion Processes in Advanced Technological Materials, edtied by D. Gupta Noyes Publications/William Andrew Publising, Norwich, NY (2004), in press.
5. Zhijian Chen, Wenhai Luo, Dan Wu, Xiang Huang, Jian He, Yuanhuan Zheng, Di Wu. Exploiting application-level similarity to improve SSD cache performance in Hadoop. The Journal of Supercomputing, (2014), pp. 703-715.
6. Y. Mishing, in: Diffusion Processes in Advanced Technological Materials, edtied by D. Gupta Noyes Publications/William Andrew Publising, Norwich, NY (2004), in press.
7. Xie Qun, Zhang Yubin, Zhang Jibiao, etc. Cluster analysis and comprehensive evaluation on summer coastal marine environment quality of Leizhou Peninsul. Marine Environmental Science ,33 (2014): 543-549.
8. Zhao Dongling, Feng Yanruo, Pan Zhengyun. Research on clustering analysis based Algorithm K- grid, Journal of Technology, 30(2014):175-178.
9. Gong Liyan, Meng Xianjun. Quality evaluation of apple processing based on Principal Component Analysis and Cluster Analysis, Agricultural Engineering, 30 (2014): 276-285.