機械学習による金融時系列モデルのパラメータ推定

Parameter estimation of financial time series models by machine learning

高石哲弥
Tetsuya Takaishi

広島経済大学
Hiroshima University of Economics

Abstract: 金融資産の実証分析において、ボラティリティは金融資産のリスクを表す重要な量である。ボラティリティの推定において、よく利用される手法は、資産価格の収益率時系列をモデル化し、ボラティリティを推定する方法である。このとき、モデルのパラメータは収益率時系列に合うように推定される。本研究では、金融時系列モデルの１つである GARCH モデルのパラメータ推定に機械学習の手法を用い、パラメータの推定が正しく行えることを示す。また、学習率の違いによる収束率についても述べる。

１．はじめに

金融における実証分析では、日々変動する金融資産のリスクを見積もり、そして将来のリスクを予測することは保有する金融資産を安全に管理するために必要不可欠である。リスクを見積もる代表的な指標としては価格変動の大きさを表すボラティリティがある。ボラティリティの変動を予測するための手法としては、価格時系列の性質を捉えたモデルを構築し、そのモデルから将来のボラティリティを予測する方法がある。金融資産価格の時系列の統計的性質は精力的に調べられており、その結果、多くの金融資産価格に共通に現れる性質があることが知られている。それらの性質は“Stylized facts”と呼ばれている[1]。Stylized facts として特に重要な性質は次である。

（１）収益率分布は裾野の厚いファットテイル分布である。
（２）収益率の自己相関は短期の相関である。
（３）絶対値収益率の自己相関は長期相関である。
（４）ボラティリティクラスタリングを示す。

実証分析では、多くの Stylized facts を再現でき、かつ予測精度の高いモデルが有用とされる。そのようなモデルの例として GARCH モデル[2-6]があり、実証分析でよく利用されている。その他に確率的ボラティリティ変動モデルもよく利用されるモデルである。

モデルを利用する場合、モデルのパラメータを時系列データにもっとも合うように推定する必要がある。そして、推定されたパラメータの下で将来のボラティリティ変動を予測する。

推定方法としては、最尤法やベイズ推定等が存在する。確率的ボラティリティ変動モデルのように、ベイズ推定が有効な推定法とされているモデルもある。ベイズ推定の実行には、マルコフ連鎖モンテカルロ法が良く利用される。そのため、有効なマルコフ連鎖モンテカルロ法の開発も重要な課題として研究がなされている。

本研究では、GARCH モデルのパラメータ推定を機械学習によって実行し、パラメータが正しく決定できるかどうかを調べる。

２．GARCH モデル

本研究では、GARCH モデルの中で最もシンプルな３つのパラメータを持つ GARCH(1,1)モデル[2]を用いる。価格時系列を r_t {t=1,…,N} とすると、GARCH(1,1)モデルは以下で記述される。

$$r_t = \sigma_t \varepsilon_t \quad (１)$$

$$\sigma_t{}^2 = \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2 \quad (２)$$

ここで、$\sigma_t^{\ 2}$ は分散、ε_t は正規乱数を表している。

このモデルにおいて、パラメータは α, β, ω の３つである。これらのパラメータを時系列データにもっとも合うように推定する。

３．ベイズ推定

ベイズの定理を利用することによって、時系列 r_t が与えられたもとでパラメータ $\theta = (\alpha, \beta, \omega)$ の従う確率分布 $p(\theta \mid r_t)$ が以下のように与えられる。

$$p(\theta \mid r_t) = \frac{L(r_t \mid \theta)\pi(\theta)}{f(r_t)} \quad (３)$$

ここで、$L(r_t \mid \theta)$ は尤度関数、$\pi(\theta)$ はパラメータの事前分布である。$f(r_t)$ は規格化定数にあたるが、ベイズ推定をマルコフ連鎖モンテカルロ法で実行するときには、重要ではない。

GARCH モデルの $L(r_t \mid \theta)$ は、以下で与えられる。

$$L(r_t \mid \theta) = \prod_{t=1}^{N} \frac{1}{(2\pi\sigma_t^{\ 2})^{1/2}} \exp(-\frac{r_t^{\ 2}}{2\sigma_t^{\ 2}}) \quad (４)$$

ベイズ推定において、パラメータの推定値は確率分布 $p(\theta \mid r_t)$ のもとでのパラメータの期待値 $\langle \theta_i \rangle$ として与えられる。ここで、$(\theta_1, \theta_2, \theta_3) = (\alpha, \beta, \omega)$ と置いた。

$$\langle \theta_i \rangle = \int \theta_i p(\theta \mid r_t) d\theta \quad (５)$$

一般には、上記の積分は解析的には行えないので、マルコフ連鎖モンテカルロ法によって期待値 $\langle \theta_i \rangle$ を見積もる。機械学習による結果との比較のために、ベイズ推定よるパラメータ推定も実行する。マルコフ連鎖モンテカルロ法には様々な手法があり、GARCH モデルにもいくつかの手法が試されている[7,8]が、本研究では多変量スチューデント分布を提案分布に用いた Metropolis-Hastings 法を用いた[9-14]。

４．機械学習による推定

機械学習によるパラメータ推定が正しく行えるかどうかを見るために、まずパラメータを $\alpha, \beta, \omega = (0.3, 0.6, 0.05)$ に設定した人工の時系列データを生成し、そのデータを利用して正しいパラメータ推定ができるかどうかを決定する。

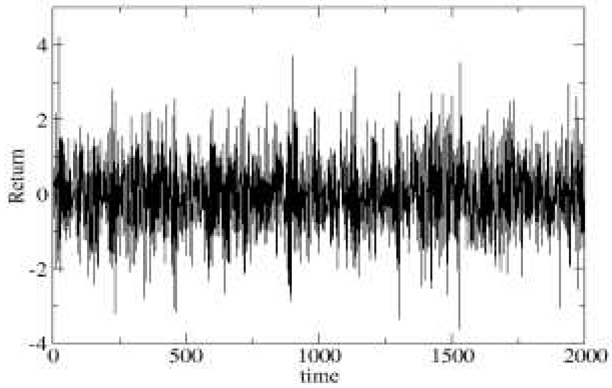

図１：人工時系列データ $\alpha, \beta, \omega = (0.3, 0.6, 0.05)$

図１は $\alpha, \beta, \omega = (0.3, 0.6, 0.05)$ に設定し、（１）及び（２）式によって生成した時系列長 N=2000 のデータである。このデータから逆に生成に使ったパラメータを推定する。

モデルの対数尤度から損失関数を以下のように定義し、機械学習によって最適化を行う。

$$\text{LOSS} = \sum_{t=1}^{N} \frac{1}{2} \left(\log(\sigma_t^{\ 2}) + \frac{r_t^{\ 2}}{\sigma_t^{\ 2}} + \log(2\pi) \right) \quad (６)$$

最適化には Adam optimizer[15]を利用した。実行に際し利用した計算機環境は表 1 である。

表 1: 計算機環境

CPU	Intel Core i7-8700 3.20GHz
Memory	16GB
OS	Windows 10
TensorFlow[16]	Version 1.9.0
Python	Version 3.6.6

パラメータの初期値は $\alpha, \beta, \omega = (0.1, 0.8, 0.1)$ に設定し、Adam optimizer の学習率(LR)は様々な値を用いてパラメータの収束を調べた。その結果、LR=0.0001～0.005 の値においては正しい値に収束することを確認した。そして、LR=0.005 を超

えるあたりから不安定となり収束しない結果となった。

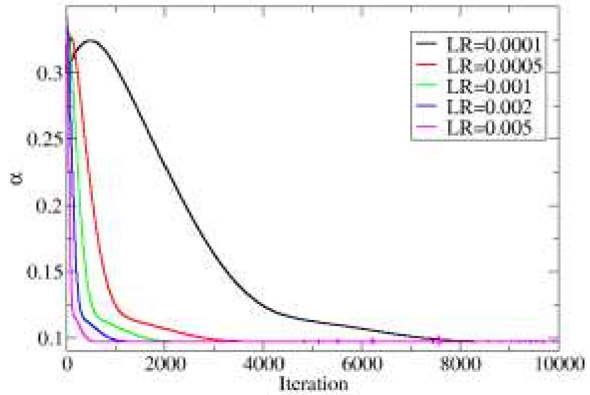

図 2：パラメータ α

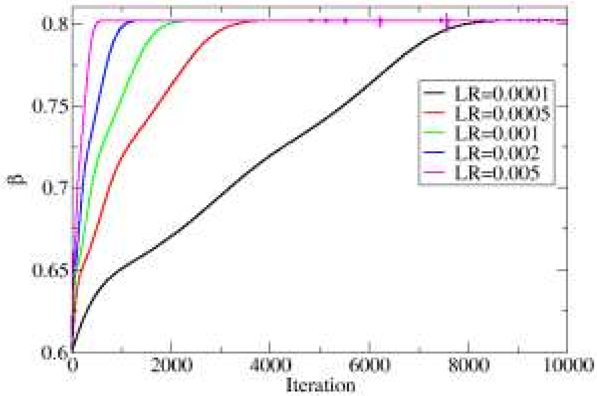

図 3：パラメータ β

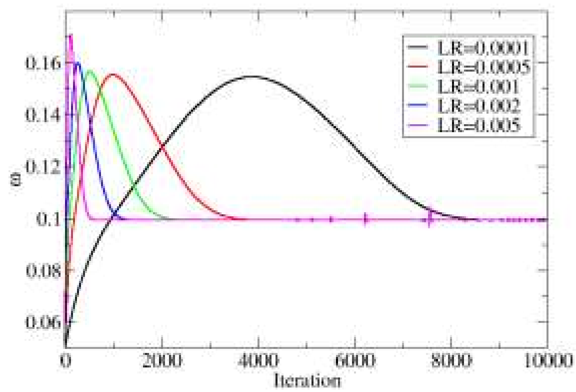

図 4：パラメータ ω

図 2～4 は様々な LR に対するパラメータ α, β, ω の収束の様子を表している。LR=0.005 まで、LR が大きいほど速く正しい値に収束している。

機械学習とベイズ推定による実行結果は、表２にまとめてある。機械学習によって求められた値と設定値は非常に似た結果となっており、機械学習によって正しい結果が求められている。

表２：推定結果

	α	β	ω
設定値（真値）	0.1	0.80	0.1
機械学習	0.0967	0.802	0.0991
ベイズ推定	0.103	0.779	0.115

次に、学習率による収束の速さについて調べる。学習率ごとの収束の速さ（収束率）を以下のように測定した。まず、真値に収束した後の損失関数の値を最小値として、誤差を損失関数の最小値からの乖離として定義する。
Error=損失関数（iteration）—損失関数の最小値

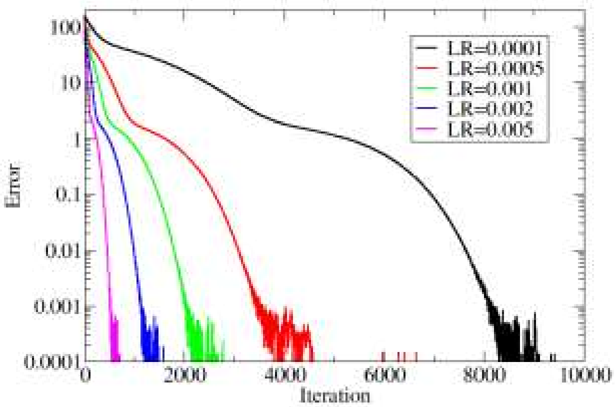

図 5：学習率の違いによる誤差減少の違い

図５はこのように定義した誤差がどのように減少していくかを表している。収束は LR が大きいほど早く、誤差の小さい領域では指数関数的に減少しているように見える。そこで、誤差の小さい領域で誤差を $a\exp(-\mu x)$ の関数でフィットする。ここで、a と μ はフィッテイングパラメータ、x は iteration である。フィッテイングによって求めた μ の値をプロットしたのが図６である。μ が大きいほど収束が速いことを意味するので、学習率が大きいほど収束が速くなることを示し

ている。但し、学習率が 0.005 を超えると不安定となり収束しない。

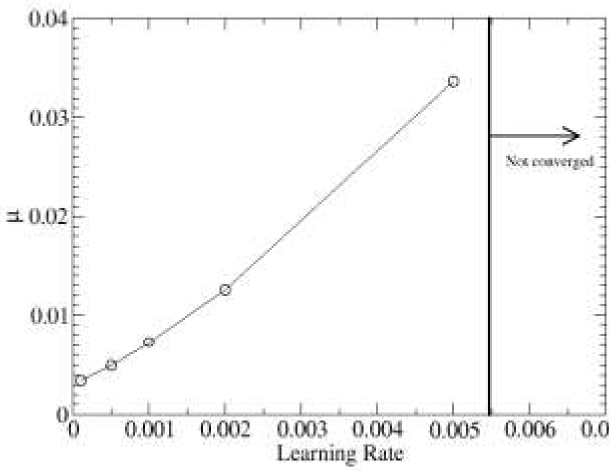

図６：μ と学習率(LR)の関係

５．まとめ

本研究では、金融時系列モデルの１つである GARCH モデルのパラメータ推定を機械学習の手法（Adam optimizer）によって実行した。予め、パラメータ値の分かっている時系列を利用し、機械学習によるパラメータ推定を行った結果、正しい結果が得られることを確認した。また、Adam optimizer の学習率を変えて実行し、学習率が大きいほど正しい値への収束が速いことが分かった。但し、学習率がある値を超えると、収束が不安定となることも分かった。

参考文献

[1] R.Cont, Empirical properties of asset returns: stylized facts and statistical issues. Quantitative. Finance 1, 223–236, (2001)

[2] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, Journal of Econometrics 31, 307-327, (1986)

[3] R.F. Engle, “Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of the United Kingdom inflation,” Econometrica, vol.50, 1982, pp.987-1007.

[4] T. Bollerslev, R.Y. Chou and K.F. Kroner, “ARCH modeling in finance,” *Journal of Econometrics*, vol.52, 1992, pp.5–59.

[5] T. Takaishi, Rational GARCH model: An empirical test for stock returns,” *Physica A*, vol.473, 2017, pp.451-460.

[6] T.Takaishi, “Volatility estimations using a rational GARCH model”, Quantitative Finance and Economics, vol.2, pp.127-136.

[7] L. Bauwens, M. Lubrano, “Bayesian inference on GARCH models using the Gibbs sampler,” *Econometrics Journal*, vol.1, 1998, pp.23-46.

[8] T. Takaishi, “Bayesian Estimation of GARCH model by Hybrid Monte Carlo,” *Proceedings of the 9th Joint Conference on Information Sciences*, 2006, CIEF-214. Available: https://doi.org/10.2991/jcis.2006.159

[9] T. Takaishi, “An Adaptive Markov Chain Monte Carlo Method for GARCH Model,” *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Complex Sciences*, vol.5, 2009, pp.1424-1434. Available: https://doi.org/10.1007/978-3-642-02469-6_22

[10] T. Takaishi, “Bayesian Estimation of GARCH Model with an Adaptive Proposal Density,” *New Advances in Intelligent Decision Technologies, Studies in Computational Intelligence*, vol.199, 2009, pp.635–643. Available: https://doi.org/10.1007/978-3-642-00909-9_61

[11] T. Takaishi, “Bayesian Inference on QGARCH Model Using the Adaptive Construction Scheme,” *Proceedings of 8th IEEE/ACIS International Conference on Computer and Information Science*, 2009, pp.525-529. Available: https://doi.org/10.1109/ICIS.2009.173

[12] T. Takaishi, “Bayesian inference with an adaptive proposal density for GARCH models,” *J. Phys.: Conf. Ser.*, vol.221, 2010, 012011.

[13] T. Takaishi and T.T. Chen, “Bayesian Inference of the GARCH model with Rational Errors,” *International Proceedings of Economics Development and Research*, vol.29, 2012, pp.303–307. Available: http://www.ipedr.com/vol29/55-CEBMM2012-R00014.pdf

[14] T. Takaishi, “Markov Chain Monte Carlo versus Importance Sampling in Bayesian Inference of the GARCH model,” *Procedia Computer Science*, vol.22, 2013, pp.1056 – 1064.

[15] D.P. Kingma and J.L. Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,” arXiv:1412.6980. Available: https://arxiv.org/pdf/1412.6980.pdf

[16] TensorFlow https://www.tensorflow.org/