Differential Machine Learningを用いたラフ・ボラティリティモデルのキャリブレーション

Calibration of Rough Volatility Models Using Differential Machine Learning

黒崎地大
Chihiro Kurosaki

指導教員 劉慶豊

法政大学大学院理工学研究科システム工学専攻修士課程

1. はじめに

金融市場におけるデリバティブの価格付けモデルとして，ラフ・ボラティリティモデル [1] が注目を集めている．従来の価格付けモデルでは，デリバティブ市場の動きを十分に表現することが困難である一方，ラフ・ボラティリティモデルは，少ないパラメータでこれを十分に表現することができる．しかしながら，ラフ・ボラティリティモデルの市場に合うパラメータを特定するキャリブレーションにおいては，未だ効率的な手法が確立されていない．そのため，実務的な適用において効率的な手法の開発が求められている．

最近では，効率的なキャリブレーション手法としてディープラーニングを利用したディープ・キャリブレーション手法が提案されている．しかしながら，ディープラーニング手法を用いた従来のラフ・ボラティリティモデルのキャリブレーションは，学習データセットの作成において多大な時間を要するため，実務適用が難しいという欠点がある．本研究の目的は，この欠点に対を克服するために，Differential Machine Learning，(以下 DML[2]) を利用したキャリブレーション手法を開発することである．

2. rBergomi モデル

分析対象のラフ-ボラティリティモデルとして rBergomi[1] モデルを扱う．rBergomi モデルにおける株価 S と分散プロセス $v = (v_t)_{t\in[0,T]}$ は，次のようにあらわされる．

$$\frac{dS_t}{S_t} = \sqrt{v_t}d\left(\rho W_t + \sqrt{1-\rho^2}W_t^{\perp}\right) \tag{1}$$

$$v_t = \xi_0(t)\exp\left(\eta W_t^H - \frac{1}{2}\eta^2 t^{2H}\right), \quad t\in[0,T]. \tag{2}$$

ここで，$(W, W^{\perp}) = (W_t, W_t^{\perp})_{t\in[0,T]}$ は二つの独立したブラウン運動であり，$\rho \in (-1,1)$ は株価とボラティリティの間の相関を表す．$\eta > 0$ は，分散の変動を表し，$\xi_0(t) : \mathbb{R}_+ \to \mathbb{R}_+$ は $\xi_0(t) = \mathbb{E}(v_t)$ によって与えられるもので，$t \in [0,T]$ における forward variance curve と呼ばれ，通常定数または区分定数として扱われる．さらに，W_t^H は次のように与えられるフラクショナルブラウン運動である．

$$W_t^H = \sqrt{2H}\int_0^t (t-s)^{H-\frac{1}{2}}dW_s, \quad t\in[0,T] \tag{3}$$

ここで，$H \in [0,1]$ は Hurst 指数と呼ばれ，フラクショナルブラウン運動の荒さを表す．

3. ニューラルネットワークによるキャリブレーション問題

従来の方法では，ニューラルネットワークを用いて価格付けモデルのパラメータ θ に基づく価格関数 $F(\theta, w)$ を学習しキャリブレーションを行う．この場合，ニューラルネットワークの重み w を最適化する目的関数は，次で与えられる．

$$\hat{w} = \arg\min_{w\in\mathbb{R}^r} \sum_{u=1}^{N_{\mathrm{Train}}} \left(F(\theta_u, w) - F^*(\theta_u)\right)^2 \tag{4}$$

ここで $F^*(\theta_u)$ はシミュレーションの結果によって得られる，学習ターゲットとなるオプション価格を表す．次に，学習したニューラルネットワークを使用し，市場データに基づいて価格付けモデルのパラメータ θ を次式によって決める．

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \sum_{i=1}^{n} \left(\tilde{F}(\theta)_i - P^{\mathrm{MKT}}(\zeta_i)\right)^2 \tag{5}$$

ここで，$\tilde{F}(\theta)_i$ はモデルにより予測される価格，ζ_i は満期，ストライク価格などの条件，$P^{\mathrm{MKT}}(\zeta_i)$ はそれに対応する市場価格に相当する．過去の研究では，ニューラルネットワークを用いた価格関数の近似に関して，Pointwise approach[2] 図 1 など様々な手法が提案されてきたが，最も大きな困難はトレーニングデータセットの作成である．ラフボラティリティモデルによるオプション価格 $F^*(\theta_u)$ の算出には，大規模なモンテカルロシミュレーションを行う必要がある．そのため，1000 組の小さなデータセットを例に挙げても，1000 回のモンテカルロ計算のコストでシミュレートされるため非常に非効率的である．本研究では，(1) で定義される価格関数の学習に重きを置き，データセットの作成コストを改善するために DML 手法によるキャリブレーションを提案する．

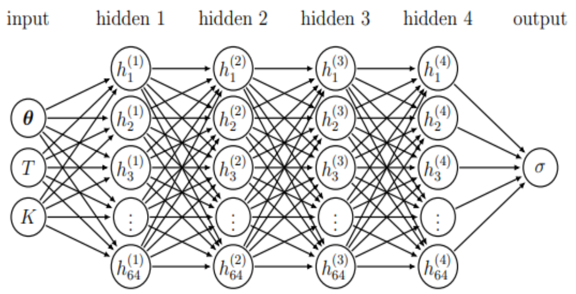

図 1 Pointwise approach（[3] より引用）．入力として価格付けモデルのパラメータ θ，満期 T，ストライク K．出力はそれに対応するボラティリティσ．

従来の研究では、学習ターゲットとしてボラティリティを対象としたもの，またはオプション価格を対象としたものが存在する。ファイナンス理論では，以下の式によりブラックショールズの価格公式を用いて，オプション価格 $P^{(\mathrm{mkt})}$ とボラティリティ $\sigma_{\mathrm{BS}}^{(\mathrm{mkt})}$ の間に一対一の関係が成り立っている．

$$P^{(\mathrm{mkt})}(K,T) = P_{\mathrm{BS}}\left(K,T,\sigma_{\mathrm{BS}}^{(\mathrm{mkt})}(K,T)\right) \quad (6)$$

したがって，オプション価格がわかればボラティリティを逆算することができる．本研究では価格をターゲットとして扱う．

4. DML

従来のニューラルネットワークによる方法においては，入力 $x \in \mathbb{R}^n$(モデルパラメータなど) と出力 $y \in \mathbb{R}$(多数のモンテカルロシミュレーションの平均をとることで得たオプション価格) の例のみを用いて訓練される．DML では，予測対象を個々のサンプルパスに応じたペイオフ $p \in \mathbb{R}$ とし，追加としてペイオフに対しての感応度 $\frac{\partial p}{\partial x}$ も含めたデータセット $\{(x^{(i)}, p^{(i)}, \frac{\partial p^{(i)}}{\partial x^{(i)}})\}_{i=1}^m$ を用いて学習を行う．結果として，多数のシミュレーションにより真値を求める必要がなく学習データセット作成における計算負荷が大幅に削減される．DML のコスト関数は以下のように定義される．

$$C = MSE + \lambda\overline{\mathrm{MSE}} \quad (7)$$

ここで，前半の MSE は通常の入力と価格に対しての平均二乗誤差である．ニューラルネットワークによる出力を $f(x^{(i)};w)$ とおくと，MSE は

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(f(x^{(i)};w) - p^{(i)})^2 \quad (8)$$

とあらわされる．後半の $\lambda\overline{\mathrm{MSE}}$ は，予測されるペイオフに関する感応度 $\frac{\partial f(x^{(i)};w)}{\partial x^{(i)}}$ と実際のペイオフに対する感応度 $\frac{\partial p^{(i)}}{\partial x^{(i)}}$ との誤差を用いて，

$$\lambda\overline{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\partial f(x^{(i)};w)}{\partial x^{(i)}} - \frac{\partial y^{(i)}}{\partial x^{(i)}}\right)^2 \quad (9)$$

とあらわされる．DML は入力データに対する予測値だけでなく，その感応度も同時に学習することで，その正則化効果によりノイズの多いデータセットに対してもモデルの汎化能力が向上する．また，グリークスの予測への応用も行うことができるというメリットがある．DML の学習には，図 2 のような twin-network が用いられる．前半のネットワークでオプション価格を学習し，後半のネットワークでオプション価格の感応度を学習する構造である．

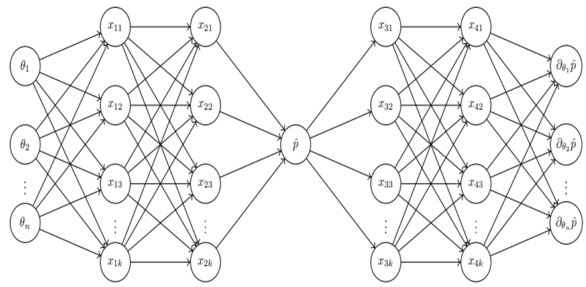

図 2 twin-network

5. 実験手順

(1) 学習データの作成

3 節 (1) の rBergomi モデルに関し，$H = 0.1$，$T = 1.0$ で固定した．その他のパラメータに関し，表 1 の範囲でパラメータセット $\theta = (\xi, \eta, \rho, S_0)$ のうち ξ,η,ρ の値を一様分布からランダムに 100 個選んだ．選んだ値に対して S_0 を変化させてサンプルパスを 5000 本ずつ合計 500,000 本発生させた．その後，個々のサンプルパスに対して，固定されたストライク価格 $K = 1.1$ に関するペイオフ $p = (S_T - K)_+$ とペイオフに関する感応度 $\frac{\partial p}{\partial \theta}$ を pytorch の自動微分ライブラリを用いて計算し，学習データセット $\{\theta, p, \frac{\partial p}{\partial \theta}\}$ を作成した．

表 1 パラメーター範囲

パラメータ	範囲
ξ	(0.01, 0.16)
η	(0.50, 4.00)
ρ	(-0.95, -0.10)
S_0	(1.0, 2.0)

(2) モデルの学習

(1) で作成したデータセットを標準化し，400,000 サンプルを学習データ，100,000 サンプルを検証データとした．学習は Twin-network(図 2) を用いて 50 エポック行った．各層の活性化関数には Softplus 活性化関数を用いた．最適化手法には Adam を用いた．その他ハイパーパラメタは表 2 の値で設定した．

表 2 ハイパーパラメタ

ハイパーパラメタ	設定値
隠れ層	3×3
各層のノード数	64
学習率	0.0001
バッチサイズ	512

6. 実験結果

実験結果を図 3 および表 3 に示す．青線はトレーニングデータにおけるコスト関数 (6) の値．オレンジ線は検証データにおけるコスト関数である．これらの値が小さく収束するほど良い学習結果となる．エポック 10 付近までは順調に学習が進んでいる．その後，学習が完了しコスト関数の値が収束していることがわかる．最終的なコスト関数の値は表 3 のようになった．

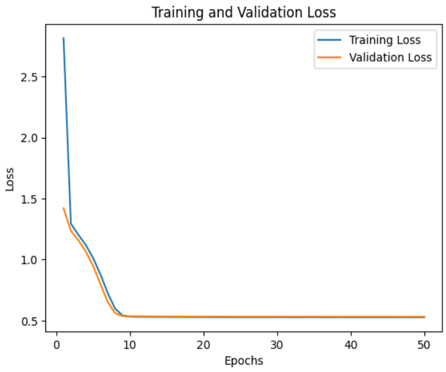

図 3 学習結果

表 3 最終的なコスト関数の値

$TrainLoss$	0.5274
$ValidationLoss$	0.5314

7. 考察とこれからの研究

本研究では，DML を用いてラフボラティリティモデルをキャリブレートする方法を提案して，実験を行った．研究成果として，オプション価格を用いずともペイオフと感応度のみで学習が可能であることを示した．しかし現状では，ペイオフを使用したことによるノイズの影響，ランダムネスによる微分ラベルへのノイズの影響がどの程度存在するのか明らかになっていない．これからの研究として，サンプルパスの平均を取った場合の学習結果への影響やパラメータを変えた場合の影響を実験を通して検証する予定である．

参考文献

1) Bayer, C., Friz, P., and Gatheral, J.,“Pricing under Rough Volatility”, *Quantitative Finance*, 16(6), pp.887–904, 2016.
2) Huge, B., and Savine, A.,“Differential machine learning”, arXiv preprint arXiv:2005.02347, 2020.
3) Baschetti, F., Bormetti, G., and Rossi, P.,“Deep calibration with random grids”, *Quantitative Finance*, pp 1–23, 2024.