Calibration of European Option Pricing Models using Neural Networks

Peetu Luotonen

Overview

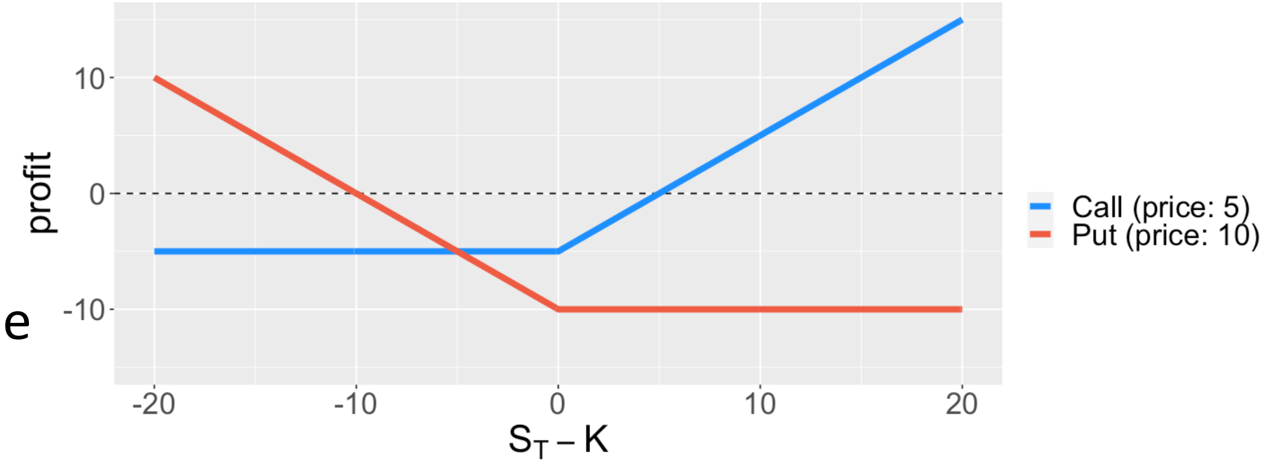
- Basics of Option Pricing
- Option Pricing Models
- Model Calibration
- Numerical Results
  - Data and Model Evaluation
  - Pricing Results
  - Model Parameters
- Applications
- Summary

Overview

- Basics of Option Pricing
- Option Pricing Models
- Model Calibration
- Numerical Results
  - Data and Model Evaluation
  - Pricing Results
  - Model Parameters
- Applications
- Summary

Option Basics

- European option: right to buy (*call*) or sell (*put*) a specific amount of an underlying asset at a specific time and (strike) price
- Variables affecting the option price:
  - underlying asset price S_0
  - maturity (time to expiration) T
  - strike price K
  - risk-free rate r
  - dividends (here, dividend yield q) of the underlying asset
  - distribution parameters of the underlying asset price (e.g., volatility σ)



Option payoff example

Option Pricing

- General principle: assume parameterized price process for the underlying asset price, and then derive a fair (arbitrage-free) price for the option
- Fair price can be derived in two ways:
  1. By constructing a *replicating portfolio* of the option from the underlying asset and the risk-free asset, and then solving the resulting stochastic PDE with suitable boundary conditions
  2. By calculating the expected payoff of the option under a risk-neutral probability measure (*risk-neutral valuation)*
- Corresponding European put/call price can be determined using the *put-call parity*

$$c + Ke^{-rT} = p + S_0 e^{-qT}$$

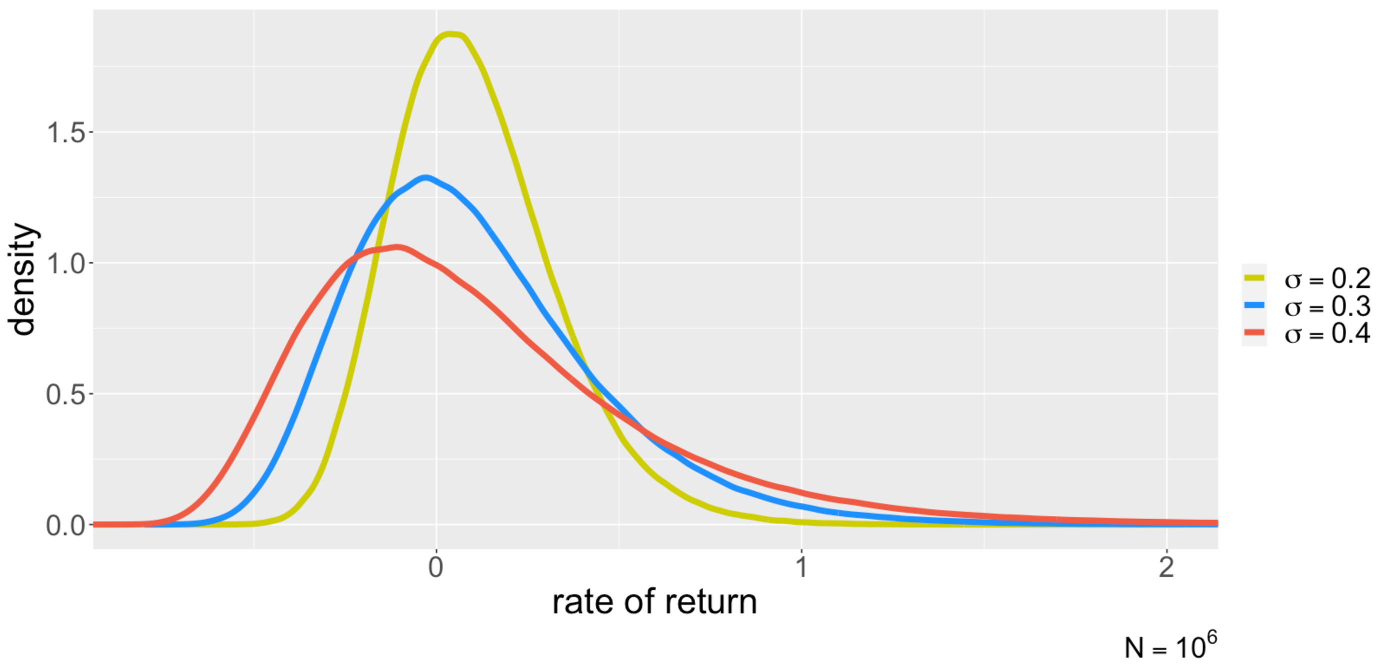
Overview

- Basics of Option Pricing
- Option Pricing Models
- Model Calibration
- Numerical Results
  - Data and Model Evaluation
  - Pricing Results
  - Model Parameters
- Applications
- Summary

The Black-Scholes Model

- Assumption: asset price follows a *geometric Brownian motion*

$$\frac{\mathrm{d}S_t}{S_t} = \underbrace{\mu \mathrm{d}t + \sigma \mathrm{d}W_t^{\mathbb{P}}}_{\text{'real'}}$$
$$= \underbrace{r\mathrm{d}t + \sigma \mathrm{d}W_t^{\mathbb{Q}}}_{\text{risk-neutral}}$$



Asset rate of returns with different volatilities

- One unobservable parameter: σ
- Limitations:
  - Gaussian log-returns
  - constant volatility

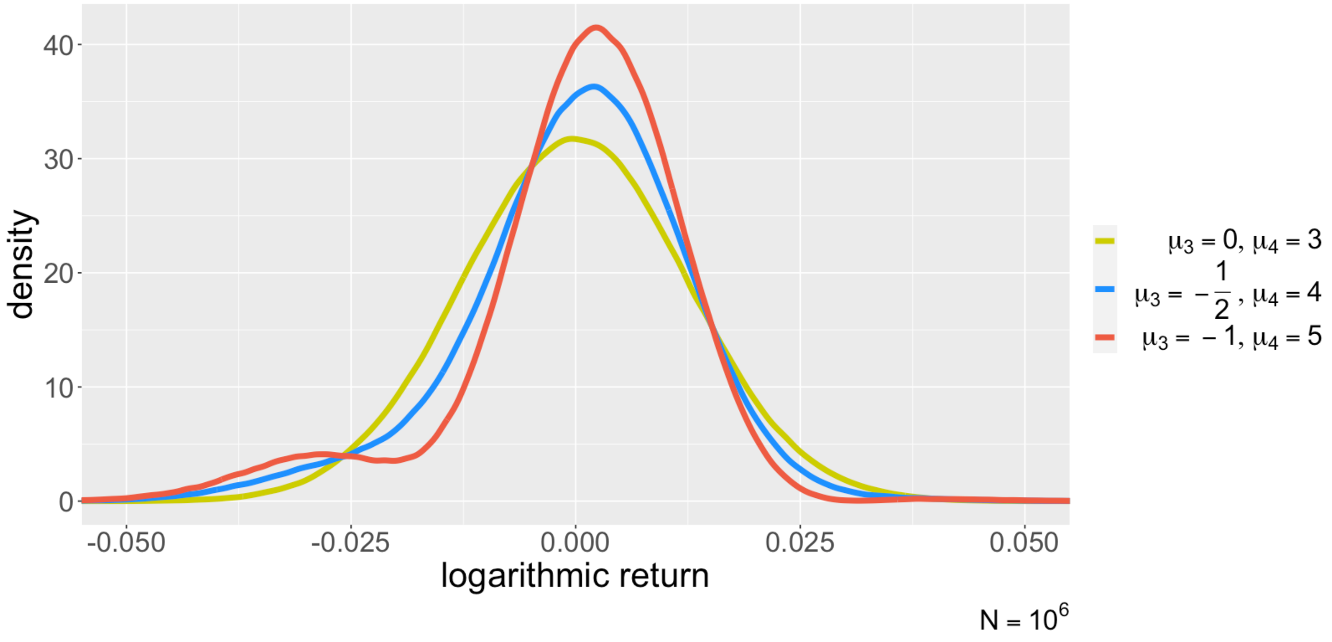
The Corrado-Su Model

- Density Gram-Charlier expansion to allow nonzero skewness and excess kurtosis:

$$g(z) = n(z)\left[1 + \frac{\mu_3}{3!}He_3(z) + \frac{\mu_4 - 3}{4!}He_4(z)\right]$$

$$z = \frac{\ln(S_t/S_0) - (r + q - \sigma^2/2)T}{\sigma\sqrt{T}}$$

- Additional parameters: skewness μ_3 and kurtosis μ_4
- Asset volatility remains constant



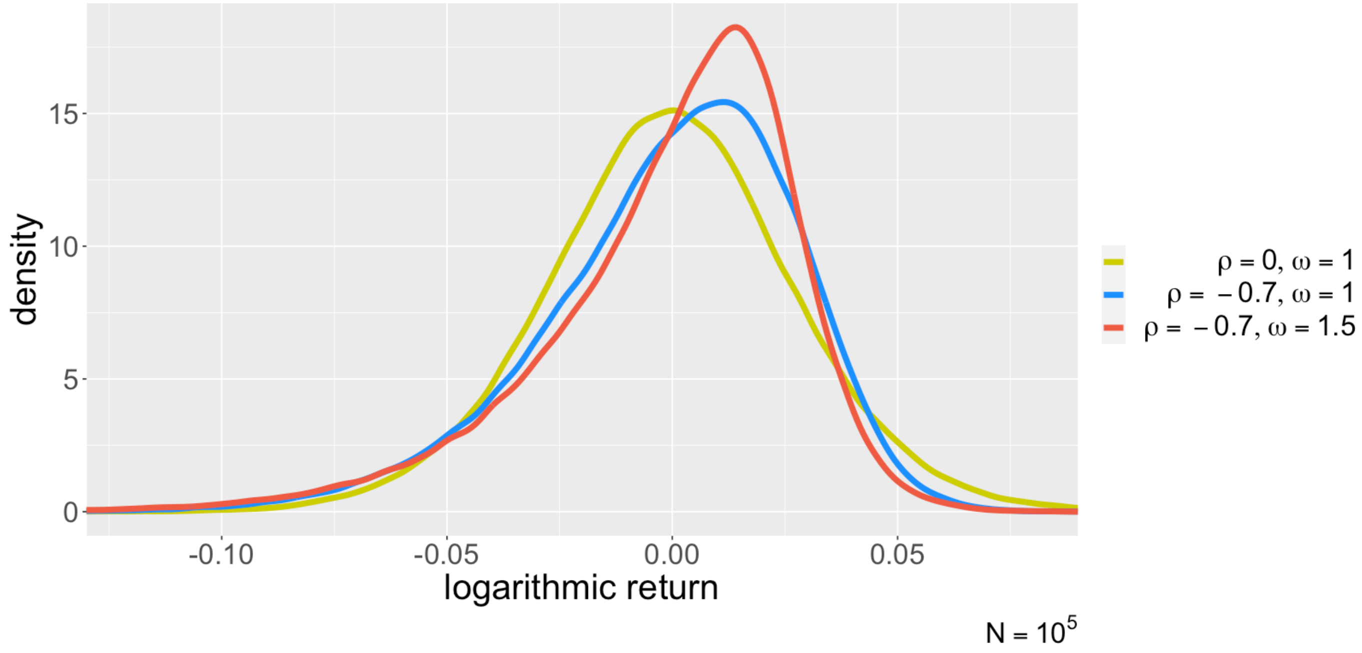
Log-returns with different μ_3 and μ_4

The Heston Model

- Asset volatility as a separate stochastic process:

$$\mathrm{d}S = \mu S \mathrm{d}t + \sqrt{V} S \mathrm{d}W^{(1)}$$
$$\mathrm{d}V = \kappa(\theta - V)\mathrm{d}t + \omega\sqrt{V}\mathrm{d}W^{(2)}$$
$$\mathrm{d}W^{(1)}\mathrm{d}W^{(2)} = \rho \mathrm{d}t$$

- 5 parameters:
  - mean variance θ
  - mean reversion coefficient κ
  - initial variance V_0
  - volatility of volatility ω
  - correlation between the asset price and volatility ρ



Log-returns with different ρ and ω

The Bates Model

- Stochastic volatility + jumps:

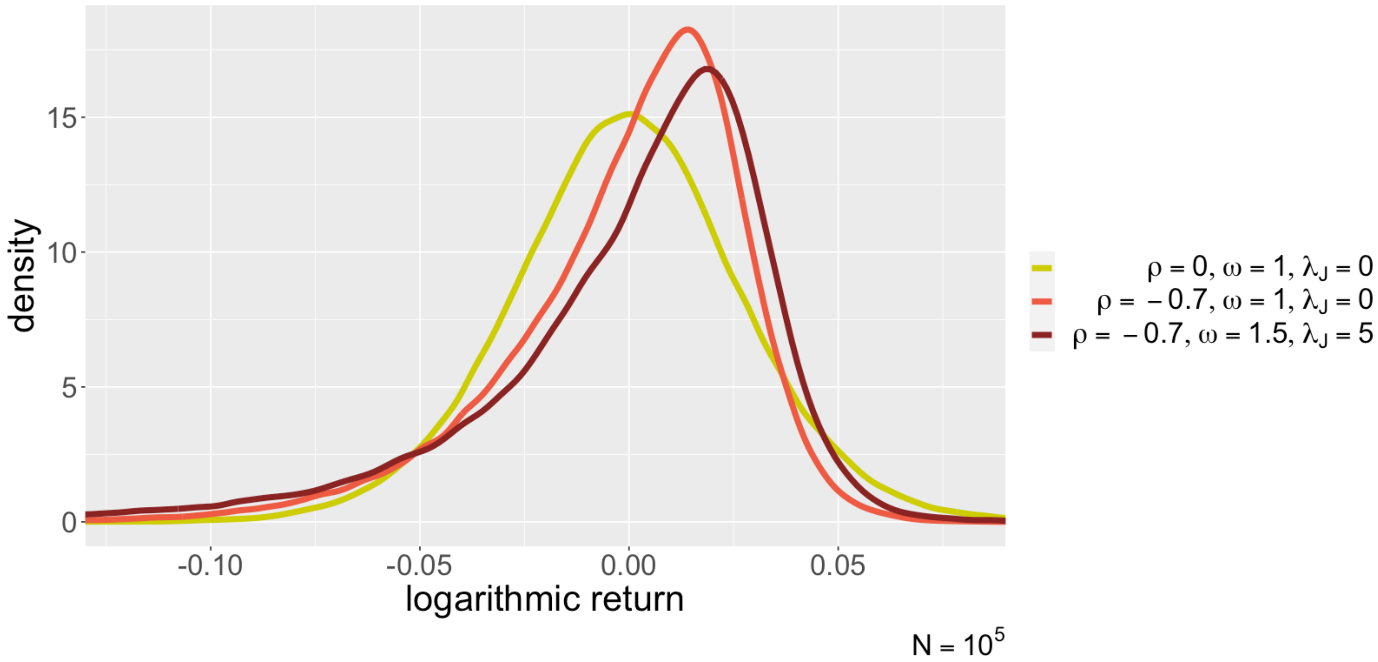
$$\frac{\mathrm{d}S}{S} = (\mu - \lambda_J \mu_J)\mathrm{d}t + \sqrt{V}\mathrm{d}W^{(1)} + k\mathrm{d}Q$$

$$\mathrm{d}V = \kappa(\theta - V)\mathrm{d}t + \omega\sqrt{V}\mathrm{d}W^{(2)}$$

$$\mathrm{d}W^{(1)}\mathrm{d}W^{(2)} = \rho\mathrm{d}t$$

$$\mathbb{P}(\mathrm{d}Q = 1) = \lambda_J\mathrm{d}t$$

- 8 parameters: Heston parameters + μ_J (jump mean), σ_J (jump standard deviation), λ_J (average number of annual jumps)



Log-returns with different ρ, ω and λ_J (with negative σ_J)

Overview

- Basics of Option Pricing
- Option Pricing Models
- Model Calibration
- Numerical Results
  - Data and Model Evaluation
  - Pricing Results
  - Model Parameters
- Applications
- Summary

Model Calibration

- Goal: select model parameters θ by minimizing some error between the market prices y_i and the model prices f(x_i;θ) given by the parametric pricing function f
- x_i is a set of other variables (S_0, K, T, r, q, put-call flag)
- Problem formulation:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i \in [n]} w_i \ell(y_i, f(x_i; \boldsymbol{\theta}))$$

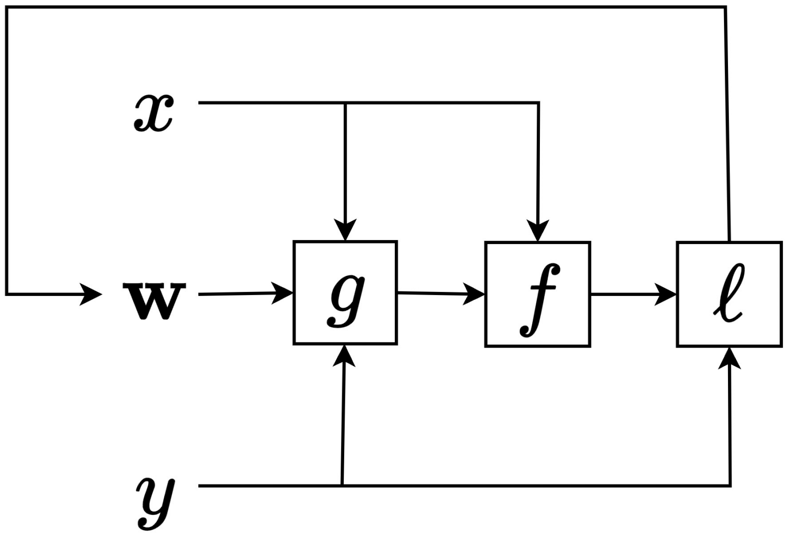
Model parameters

Model	$\boldsymbol{\theta}$
Black-Scholes	σ
Corrado-Su	σ, μ_3, μ_4
Heston	$\kappa, \theta, V_0, \omega, \rho$
Bates	$\kappa, \theta, V_0, \omega, \rho, \lambda_J, \mu_J, \sigma_J$

“Inverse map” Approach

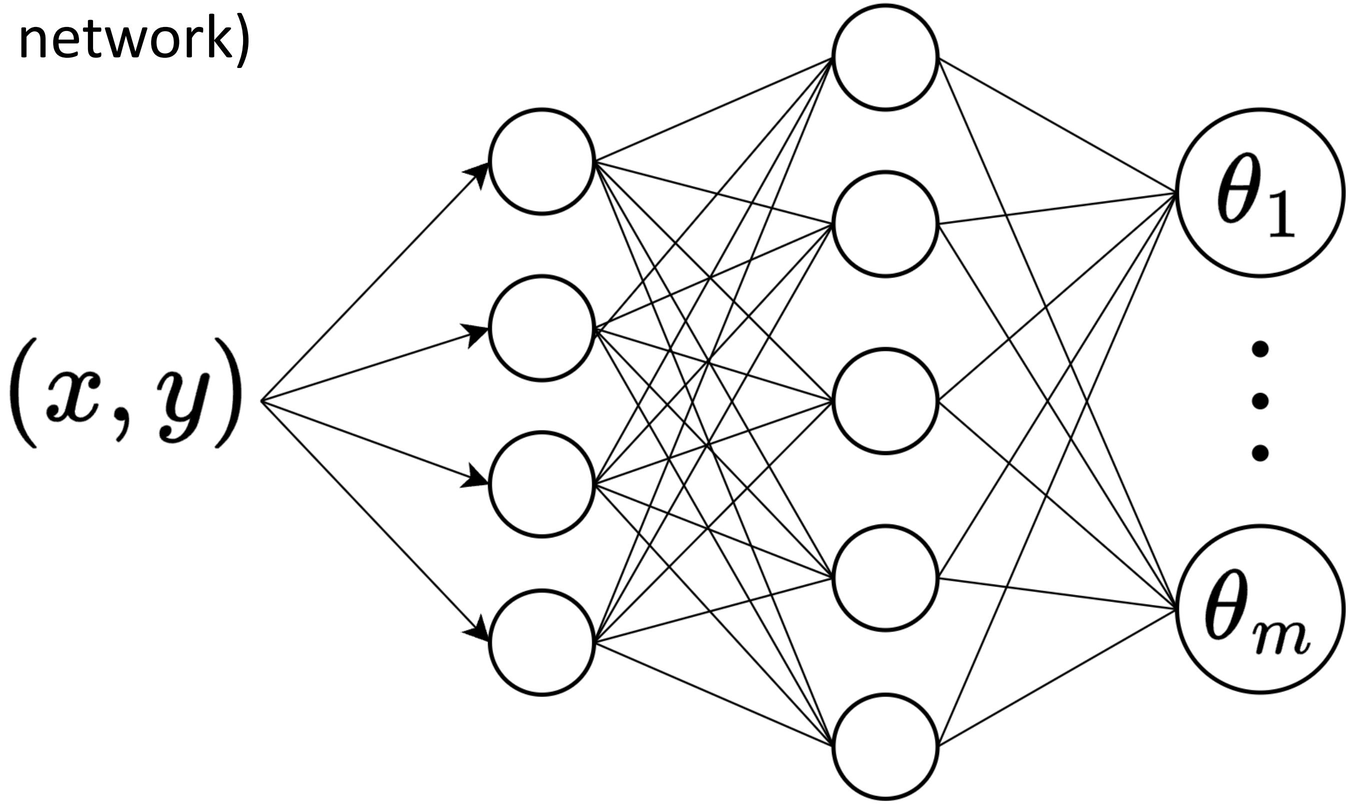
- Goal is to learn a mapping g from option prices y_i and other market variables x_i to the model parameters θ
- The inverse map g is parameterized by weights w, and its outputs are used as inputs to the parametric pricing function f
- Formulation:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{i \in [n]} w_i \ell(y_i, f(x_i; g(x_i, y_i; \mathbf{w})))$$



Inverse Map Approach using Neural Networks

- Model the inverse map g as a multilayer perceptron (dense feedforward neural network)

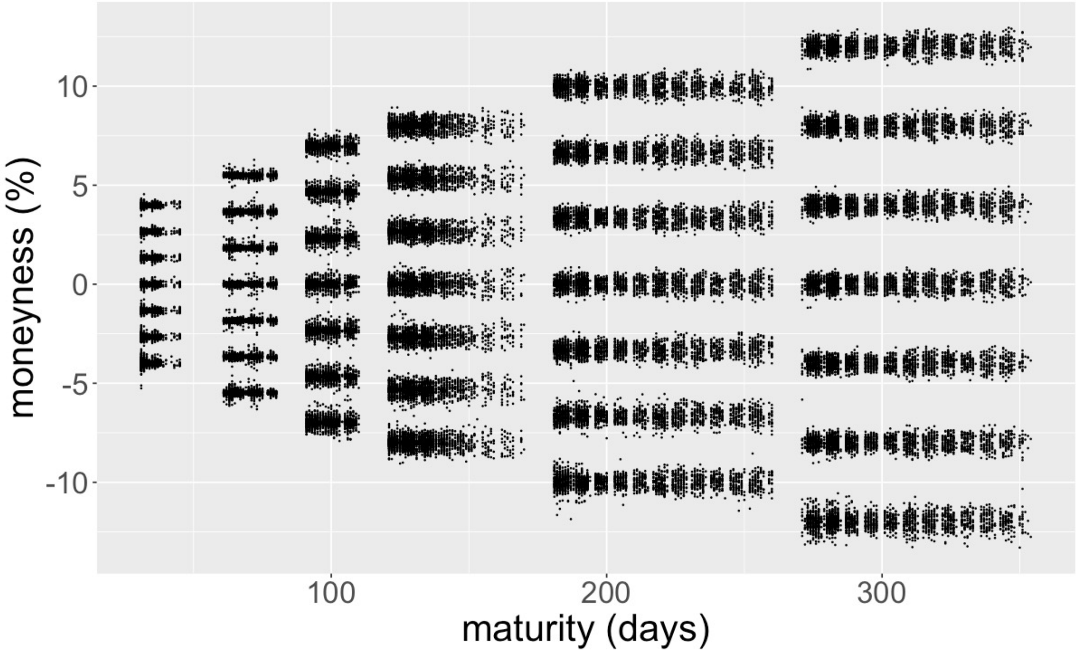


Overview

- Basics of Option Pricing
- Option Pricing Models
- Model Calibration
- Numerical Results
  - Data and Model Evaluation
  - Pricing Results
  - Model Parameters
- Applications
- Summary

Data

- European SPX (S&P 500) index options
  - Market price: mid price between the bid and ask prices
- Risk-free rate: US Treasury bond yields interpolated at the option’s maturity
- Dividend yield: last yearly dividends of companies in the S&P 500 index, normalized by stock price and weighted by market capitalization



data points in maturity-moneyness dimension

(moneyness = relative difference between the underlying price and the strike price)

Model Evaluation

- 10 different train-validation-test splits between 2015 and 2022
  - Each split consists of a 2-year train interval, followed by a 5-month validation interval, followed by a 6-month test interval (with 2-week gap between the intervals)
- Validation intervals are used for selecting the best model hyperparameters:
  - The number of hidden network layers: 1 or 2
  - The number of units per hidden layer: 6, 8 or 10
- The stochastic training process is repeated for each model, time split and set of hyperparameters 5 times using different random seeds

Number of unique data points for the time splits

	Train	Valid.	Test
Min.	15750	2688	3276
Avg.	17686	3713	4423
Max.	18774	4326	5124
Total	56322	37128	44226

Model Evaluation Metrics

- Absolute (dollar) error:

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$$

- Percentage error:

$$\mathrm{MAPE} = 100\% \cdot \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

- Metrics related to the bid-ask spread:

$$\mathrm{errSpread} := \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{a_i - b_i}$$

$$\mathrm{pSpread} := 100\% \cdot \frac{1}{n}\sum_{i=1}^{n} I(b_i \leq \hat{y}_i \leq a_i)$$

Overview

- Basics of Option Pricing
- Option Pricing Models
- Model Calibration
- Numerical Results
  - Data and Model Evaluation
  - Pricing Results
  - Model Parameters
- Applications
- Summary

Pricing Results

- The results presented next have been obtained by fitting the models to *only call options*
- Overall, the results for calls and puts are similar, but slightly better for puts
- When fitting the models to calls and puts simultaneously, the results are noticeable worse than in the case where calls and puts are fitted separately
- In the case of all models, and both calls and puts, the largest allowed network (2 hidden layers, 10 units per hidden layer) gives the best validation performance (measured in errSpread)

Pricing Results

- More complex models have better train results on average
- Overall, the average validation and test results are the best for the Bates and Corrado-Su models, and the worst for the Black-Scholes model
- The average validation and test results are worse than the train results, but the out-of-sample performance of all models is still relatively good:
  - Average errSpread around one or lower (predictions are relatively close to the bid and ask prices)
  - Over half of the out-of-sample predictions inside the bid-ask spread

Average model metrics over all train, validation and test sets, and the random seeds. For each model, only the best version (set of hyperparameters) is considered, and the best version is trained with 12 different seeds. For each metric, the best and worst test values are colored green and red, respectively.

Model	RMSE	MAPE	errSpread	pSpread
Black-Scholes	train : 0.80 valid : 2.35 test : 2.79	train : 2.61 valid : 2.81 test : 1.74	train : 0.60 valid : 0.93 test : 1.01	train : 65.51 valid : 53.95 test : 50.24
Corrado-Su	train : 0.70 valid : 2.11 test : 2.07	train : 2.37 valid : 2.50 test : 1.45	train : 0.52 valid : 0.81 test : 0.83	train : 70.29 valid : 57.99 test : 55.11
Heston	train : 0.66 valid : 2.86 test : 3.17	train : 1.74 valid : 2.10 test : 1.42	train : 0.43 valid : 0.81 test : 0.88	train : 74.54 valid : 62.05 test : 58.51
Bates	train : 0.63 valid : 2.90 test : 3.08	train : 1.58 valid : 1.96 test : 1.34	train : 0.40 valid : 0.78 test : 0.85	train : 76.50 valid : 63.86 test : 59.70

Pricing Results

- The model performance varies significantly between different years, maturities, strike prices and random seeds
- The average model performance is the weakest for short maturity and out of the money options, and 2020 is the worst out-of-sample year in the case of all models
- Correlations of the metrics over 12* random seeds are clearly positive on average

* the 5 seeds used in validation + 7 additional seeds

correlations of the metrics over the random seeds

(a) train-validation

Model	RMSE	MAPE	errSpread	pSpread	Avg.
B-S	0.203	0.950	0.766	0.963	0.721
C-S	0.343	0.691	0.604	0.886	0.631
H	0.298	0.808	0.677	0.960	0.686
B	-0.214	0.840	0.618	0.951	0.549
Avg.	0.157	0.822	0.666	0.940	0.646

(b) train-test

Model	RMSE	MAPE	errSpread	pSpread	Avg.
B-S	0.188	0.732	0.629	0.967	0.629
C-S	0.340	0.545	0.579	0.883	0.587
H	0.132	0.495	0.551	0.944	0.531
B	-0.237	0.207	0.370	0.924	0.316
Avg.	0.106	0.495	0.533	0.929	0.516

(c) validation-test

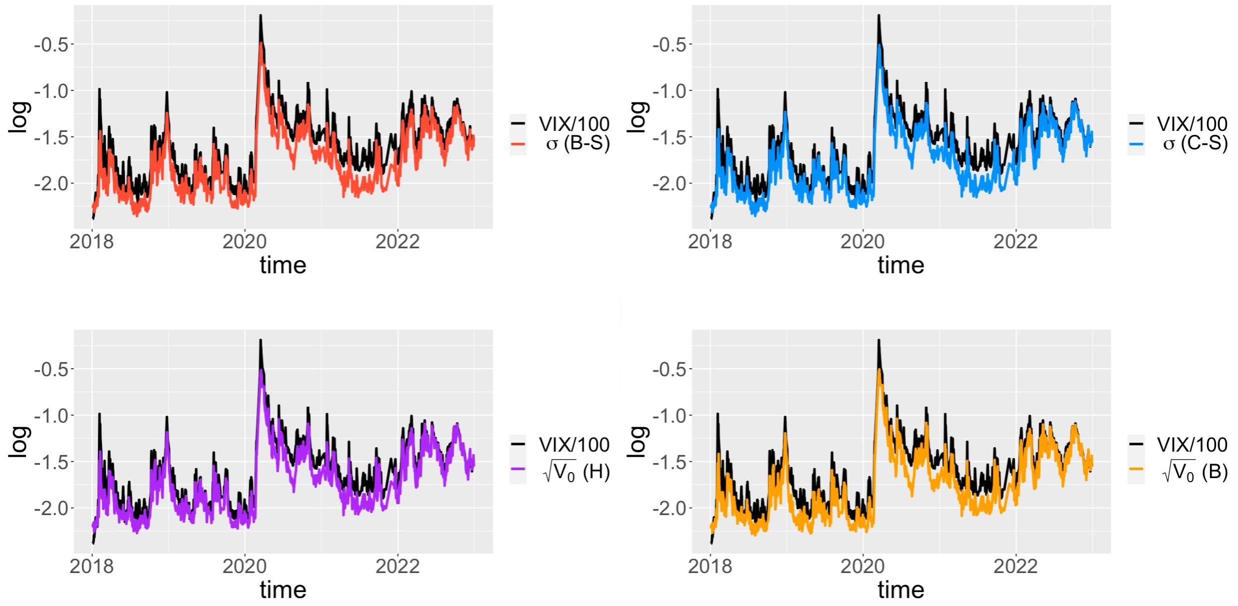
Model	RMSE	MAPE	errSpread	pSpread	Avg.
B-S	0.472	0.843	0.780	0.973	0.767
C-S	0.808	0.712	0.841	0.875	0.809
H	0.885	0.839	0.942	0.964	0.907
B	0.921	0.447	0.848	0.965	0.795
Avg.	0.772	0.710	0.853	0.944	0.820

Overview

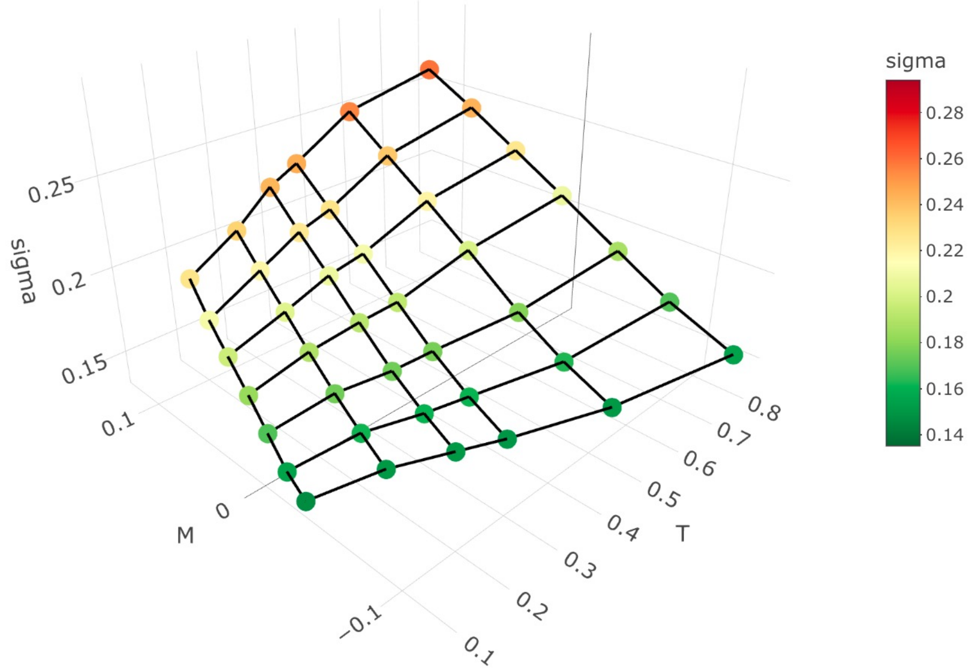
Model Parameters

- For models with multiple parameters (Corrado-Su, Heston, Bates), the average parameter values can vary significantly between different random seeds
  - Not advisable to interpret parameters of a model separately
- The average parameter values also vary between different years, maturities and strike prices
  - This type of behavior is well-known
- The average volatility parameters (σ for Black-Scholes and Corrado-Su, $\sqrt{V_0}$ for Heston and Bates) are similar to each other, and to the VIX volatility index

Model Parameters



Short maturity (1 month) implied volatilities of each model compared to the VIX index



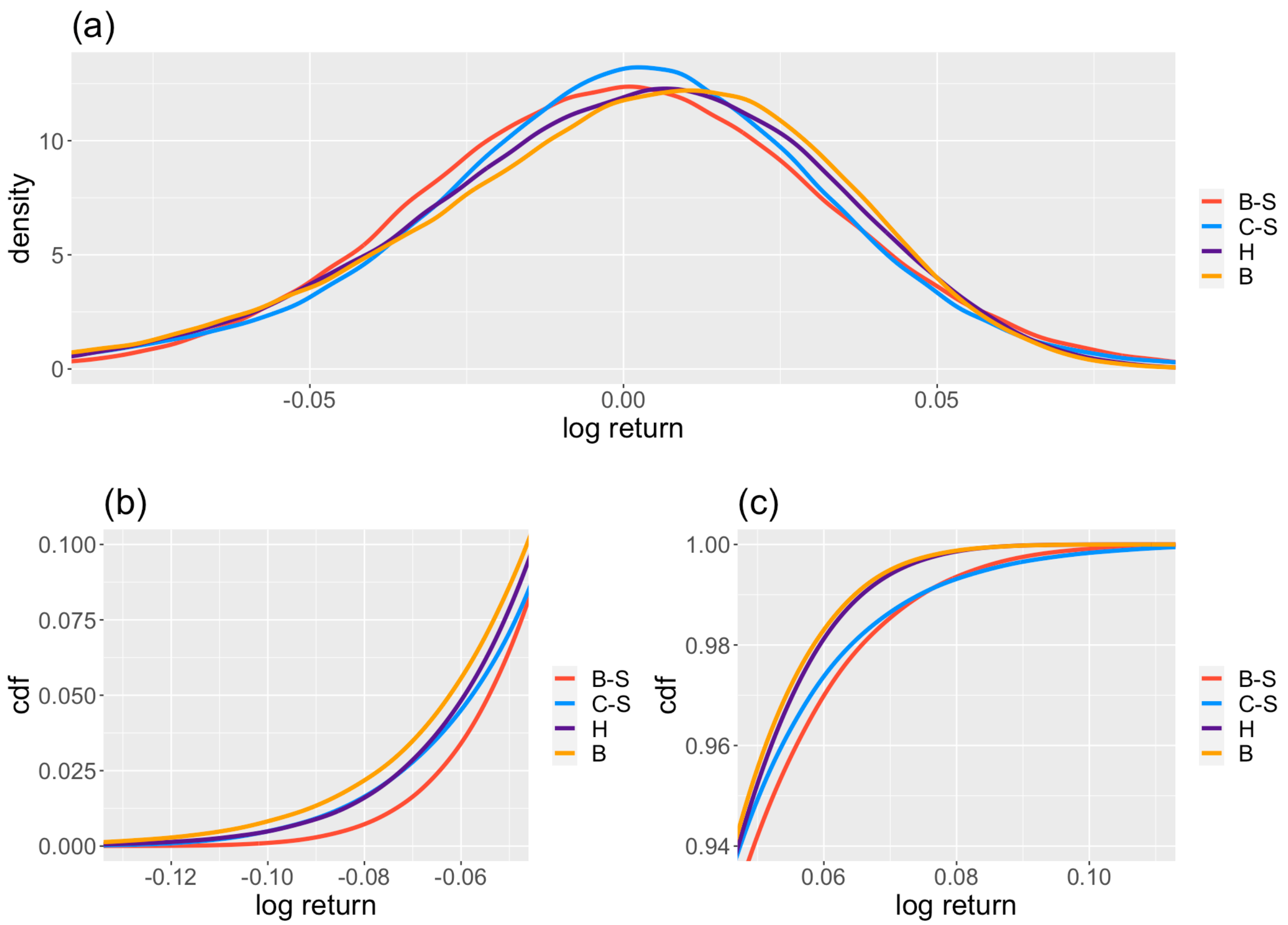
Black-Scholes implied volatility as a function of moneyness (M) and maturity (T)

Overview

- Basics of Option Pricing
- Option Pricing Models
- Model Calibration
- Numerical Results
  - Data and Model Evaluation
  - Pricing Results
  - Model Parameters
- Applications
- Summary

Computing Implied Distributions

- Simulate log-returns from the discretized stochastic process using the parameter values given by the inverse maps
- Corrado-Su, Heston and Bates models are able to produce samples similar to empirical log returns (with negative skewness and positive excess kurtosis)



Hedging

- Idea: reduce variance of returns by constructing a portfolio of options and shares of the underlying asset
- Common example: sell a call option, and hedge the position by buying shares of the underlying asset
- The amount of shares bought is determined by *delta* of the call option (partial derivative of the option price w.r.t. the underlying price)
  - In the case of the Heston model, one should also consider the vega of the option (partial derivative w.r.t. the asset volatility)
  - Theoretically perfect hedge not possible under the Bates model
- In neural network setting, computation of the partial derivatives (*Greeks)* is straightforward due to automatic differentiation

Overview

Summary

- There exist many parametric pricing models for European options, and the parameters can be chosen by minimizing some error between the market and model prices
- When the parameters have been obtained, they can be used in different applications, such as hedging, or calculation of the implied asset distribution
- In the so-called inverse map approach, one not only obtains the model parameters, but also a mapping from the market variables to the model parameters
- Here, the inverse map is modeled as a simple feedforward neural network (MLP)
- When the models are fitted to SPX index options (calls and puts separately), the overall results are promising, but the model performance
    - is weaker out-of-sample than in-sample
    - varies between different time intervals, option maturities and strikes, and seeds that control the randomness of the training process