Forecasting left- and right-tail conditonal quantile and conditional violation expectation with multi-step 2T-POT Hawkes model

Rodolfo Petrelli[1] and Matteo Conterno[1]

[1]MSc Candidates in Quantitative Finance at Collegio Carlo Alberto, Torino, Italy

December 2024

Summary

We extend the single-step two-tailed peaks-over-threshold (2T-POT) Hawkes model formulated in [1] and [2] to a multi-step framework for forecasting conditional quantile (value-at-risk) and conditional violation expectation (expected shortfall) of extreme financial losses and gains over the next 5 days. The model is trained on daily log-returns of the S&P 500 index from 1975-01-01 to 2015-01-01 and backtested over the period 2015-02-02 to 2022-01-01. We compare its performance with an heteroskedastic GARCH-type conditional volatility model and find that the multi-step 2T-POT Hawkes framework provides more accurate forecasts for VaR and ES at the 5% coverage level and the 95% quantile.

1 Introduction

While in GARCH processes the intensity and magnitude of extreme events are functions of the conditional volatility, in the two-tailed peaks-over-threshold (2T-POT) model the dynamics of threshold-exceeding extremes are purely self-contained. 2T-POT integrates an exceedance distribution with a self-exciting Hawkes point process to capture the time-inhomogeneous occurrence of tail events, where past events trigger a time-decaying increase in the intensity of future events. [1] provides a novel framework for modeling asymmetric self- and cross-excitation in left- and right-tail extreme financial log-returns. This model combines a self- and cross-exciting Hawkes-type arrival process with asymmetric conditional generalized Pareto (GP) tails to describe the threshold exceedances in both tails of the return distribution. When applied to daily log-returns, the 2T-POT framework effectively captures key stylized facts of extreme financial events: the temporal clustering of extreme gains and losses, the correlation between the intensity of these events and their magnitudes, the heavy-tailed distributions of exceedances, and the asymmetry between left- and right-tail dynamics. In [1], the model is applied to the extreme gains and losses of the S&P 500 index, with thresholds defined at the 2.5% and 97.5% quantiles, respectively. Likelihood-based inference methods reveal that the 2T-POT model outperforms symmetric models and bivariate Hawkes models that treat left- and right-tail extremes as separate processes. Extreme daily losses and gains are shown to share a common conditional intensity, with losses contributing 2.2 times more on average and decaying 4.6 times faster. These are well known heuristics linked to negativity bias [5] and levergage effect, which associates higher volatility with negative returns [3]. The greater contribution of losses can be explained by the GJR-GARCH leverage effect [4]; However, the faster decay rate of losses is a novel insight provided by the 2T-POT Hawkes model, and suggests that the leverage effect is more pronounced at shorter timescales. Building on this work, [2] compares the 2T-POT model with generalized autoregressive conditional heteroscedasticity (GARCH) models, including GARCH-EVT, to evaluate one-step-ahead forecasts of conditional quantile-based risk measures. They reparametrize the 2T-POT model by replacing the exogenous background intensity with the expected average intensity as a fitting parameter. This reduces optimization time by half and achieves a dimension reduction of one parameter without compromising model fit. Both [1] and [2] demonstrate that the 2T-POT model consistently delivers superior accuracy in forecasting Value-at-Risk (VaR) and Expected Shortfall (ES) at extreme quantiles, in both the far left-tail (5% or less) and far right-tail (95% or greater). These results suggest that the asymmetric Hawkes-type arrival dynamics provide a more realistic approximation of the underlying data-generating process for extreme log-returns compared to GARCH-type variance dynamics. In this project we derive a novel multi-step 2T-POT framework to predict VaR and ES for both left and right tails over a 5-day forecasting horizon. We achieve this by introducing recursive updates to the Hawkes intensity function, which now dinamically incorporates both historical and recent predicted exceedances. In recursive simulation synthetic event paths are iteratively simulated at each step $t+1, t+2, \dots, t+5$ and at each step the Hawkes intensity function $\lambda(t)$ is updated based on the simulated arrivals and magnitudes of prior exceedances. The recursive update uses the exponential decay kernel to reflect the diminishing impact of past events over time while the conditional quantiles and violation expectations are recalculated at each forecast horizon step. The tail quantiles are obtained using the updated GPD parameters (ξ, σ) and the propagated Hawkes intensity, while the cumulative impact of simulated exceedances is aggregated over the multi-step horizon. The results are compared to a GARCH-EVT model to confirm the effectiveness of the multi-step 2T-POT model in delivering superior accuracy also for 5-day forecasts of VaR and ES.

2 Model specification and in-sample training

2.1 Conditional intensity function

To define the multi-step model we generalize the 2T-POT Hawkes framework by incorporating both historical and recursively predicted future events. This enables forecasting over a horizon of $t+n$, where n is the prediction step length. The Hawkes function $\lambda(t+n)$ of the common intensity process will be:

$$\lambda_{\leftrightarrow}(t+n) = \mu_{\leftrightarrow} + \gamma_{\leftrightarrow}^{T} \sum_{k:t_k<t} \phi(t+n-t_k)\kappa(M_k)$$

where:

- $\mu_{\leftrightarrow} = [2 - (\gamma_{\leftrightarrow\leftarrow} + \gamma_{\leftrightarrow\rightarrow})]a_{\leftrightarrow}\lambda$, adjusts the baseline intensity based on the expected average arrival rate of arrivals and the contribution from branching. This is part of the reparametrization made in [2] which allows for faster maximum likelihood optimization under the SLSQP method in SciPy [6].
- $\phi(t) = \beta e^{-\beta t}$ is the exponential decay kernel which defines the decaying influence of past events.

- $\kappa(M_k) = \frac{1-\alpha \ln[1-F_{P,\cdot}(M_k)]}{1+\alpha}$ is the impact function that scales the contribution of exceedances based on their magnitudes.

$\gamma_{\leftrightarrow}$ defines the relative contribution of left- ($\leftarrow$) and right-tail ($\rightarrow$) exceedances to the overall intensity.

At each future step $t+n$ historical events ($t_k < t$) continue to contribute while presenting the characteristic decaying influence of the Hawkes model. New predicted arrivals at $t, t+1, \dots, t+n-1$ add new contributions, recursively updating the intensity function $\lambda_{\leftrightarrow}(t+n)$. The excess magnitudes follow a conditional General Pareto Distribution (GPD) with cdf

$$F_P(M) = 1 - \left(1 + \frac{\xi M}{\sigma}\right)^{-1/\xi},$$

where ξ is the shape parameter, σ is the scale parameter and M_k measures the exceedance magnitude.

2.2 Conditional quantile and conditional violation expectation

We reformulate conditional quantile (value-at-risk) and conditional violation expectation (expected shortfall) for multi-step prediction

Conditional quantile

For a multi-step framework, the quantile at step $t+n$ depends on the recursively predicted conditional distribution $F_{X_{t+n},t}$

$$Q^{\leftarrow}_{a_q,t+n} = F^{-1}_{X_{t+n},t}(a_q)$$

where $F_{X_{t+n},t}$ is the conditional cdf of X_{t+n} , recursively updated using historical and predicted data. n is the number of steps ahead. Violations of the left-tail conditional quantile is reformulated as: $I^{\leftarrow}_{a_q,t+n} = \mathbb{I}\left[-\left(X_{t+n} - Q^{\leftarrow}_{a_q,t+n}\right)\right]$ The reformulation is applied similarly for the right-tail conditional quantile $Q^{\rightarrow}$.

Conditional violation expectation

$$E^{\leftarrow}_{a_q,t+n} = \frac{1}{a_q}\int_{-\infty}^{Q^{\leftarrow}_{a_q,t+n}} x f_{X_{t+n},t}(x)dx$$

where $f_{X_{t+n},t}(x) = \frac{dF_{X_{t+n},t}(x)}{dx}$ and $Q^{\leftarrow}_{a_q,t+n}$ is the quantile at step $t+n$. This formula accounts for all possible values of x below the quantile $Q^{\leftarrow}_{a_q,t+n}$, weighted by the conditional PDF.

2.3 Data presentation

In line with the approach of M.F. Tomlison et al in [1] and [2], we apply our model to the daily log-returns of the SP 500 index (SPX) over the period from 1975-01-01 to 2022-01-01. We split the dataset into an in-sample training period (1975-01-01 to 2015-01-01) and out-of-sample backtesting period (2015-01-02 to 2022-01-01) for model validation and benchmarking.

For this project we use a fixed thresold $a_u = 0.025$ in line with [1]. We leave to future work the multi-step 2T-POT model calibrated on dynamic thresholds as proposed in [2] which uses a wide range of threshold levels: $a_u = 0.0125k_u$, where $k_u \in \mathbb{Z} \cup [1, 20]$ to better model arrival times t_k for larger datasets.

Stat.	**In-Sample (1975–2015)**	**Out-of-Sample (2015–2022)**
T	10,092	1,936
$\bar{X}$	3.37×10^{-4}	3.52×10^{-4}
σ_X	1.09×10^{-2}	1.17×10^{-2}
$\hat{Q}_{0.5}(X)$	5.33×10^{-4}	6.33×10^{-4}
MAD_X	5.15×10^{-3}	4.57×10^{-3}

Table 1: Table I. Key statistics for the daily log-returns of the S&P 500 Index (SPX) for the in-sample period (January 1, 1975, to January 1, 2015) and the out-of-sample period (January 2, 2015, to January 1, 2022). The statistics include the number of observations (T), mean log-return ($\bar{X}$), standard deviation (σ_X), median log-return ($\hat{Q}_{0.5}(X)$), and the median absolute deviation (MAD_X).

2.4 Parameter calibration and log-likelihood function

The set of parameters θ_u for the Hawkes intensity function and for the GPD is calibrated through the log-likelihood function. We define the total likelihood function for the multi-step model by combining the contributions from the arrival process and the magnitude distribution.

Arrivals process likelihood

$$\ell_\lambda(\theta_u|X_{0:T}) = \sum_{k:t_k<T} \ln\left[\lambda_i(t_k+n|\theta_u, \mathcal{M}_{t_k})\right] - \int_0^{T+n} \lambda_i(t|\theta_u, \mathcal{M}_t)\, dt$$

where $\lambda_i(t|\theta_u, \mathcal{M}_t)$ is the intensity of arrivals conditional on the historical data up to t. In the multi-step model the intensity $\lambda_i(t)$ evolves recursively for each future time step $t+n$.

Magnitudes distribution likelihood

$$\ell_M^i(\theta_u|X_{0:T}) = \sum_{k:t_k^i<T} \ln\left[f_{P,i}(M_k^i|\theta_u)\right],$$

where $f_{P,i}(M|\theta_u)$ is the probability density function of the GPD for magnitudes $M_k^i > u$, and u is the fixed threshold. In the multi-step model, this likelihood is applied to magnitudes predicted at future arrival times.

Total likelihood

$$\ell_{\leftrightarrow u}(\theta_u|X_{0:T}) = \sum_{i\in\{\leftarrow,\rightarrow\}} \left(\ell_\lambda^i(\theta_u|X_{0:T}) + \ell_M^i(\theta_u|X_{0:T})\right),$$

where i denotes the left ($\leftarrow$) or right ($\rightarrow$) tail. The total likelihood function for the multi-step model is designed to fit for recursive updates to $\lambda_i(t)$ and $f_{P,i}(M)$ for each future time step. For the Hawkes process the set of parameters to calibrate is $\theta = \{\mu_\leftrightarrow, \gamma_\leftrightarrow, \beta_\cdot, \kappa_\cdot\}$ to which are applied the stability constraint: $\gamma_{\leftrightarrow\leftarrow} + \gamma_{\leftrightarrow\rightarrow} < 2$ and the expected average intensity constraint: $a_\leftrightarrow\lambda \approx 2a_u\Delta t$. On the other hand the set of GPD parameters (ξ, σ) is fitted using exceedance magnitudes M_k. The process entails separated calibration for left ($\leftarrow$) and right ($\rightarrow$) tails. The estimated exceedance model parameters $\hat{\theta}_u$ are then estimated with `Scipy` using the SLSQP method.

The fit of the exceedance model to the arrivals process and the distribution of excess magnitudes is tested through residual analysis. Arrival times t_k are transformed into residual times $t_k^\star$ using the compensator:

$$t_k^\star = \int_0^{t_k} \lambda(s)ds.$$

We perform the KS test on the residual times $t_k^\star$ against the null hypotesis that they should follow a Poisson distribution. After transformation via the compensator, the cumulative effect of varying intensity and clustering is removed. The transformed residual times should be independent and uniformly distributed over $[0,\infty)$ and are expected to follow a Poisson distribution as the process is now "declustered". For the excess distributions, the KS test evaluates the fit of the transformed excess magnitudes $\{\tilde{m}_k\}$, defined as:

$$\tilde{m}_{\leftrightarrow,k} = \frac{1}{\xi_\leftrightarrow}\ln\left[1 + \xi_\leftrightarrow\frac{m_{\leftrightarrow,k}}{\sigma_{t_k}}\right],$$

The transformed magnitudes are tested against the unit exponential distribution, whose cdf is:

$$F_{\text{Exp}}(\tilde{m}) = 1 - e^{-\tilde{m}}, \quad \tilde{m} \geq 0.$$

3 Backtesting and validation

After calibrating the model, we generate multi-step conditional quantile and conditional violation expectation predictions for a 5-day horizon in the out-of-sample period spanning January 2, 2015, to January 1, 2022. This forecasting horizon was chosen to align with a single trading week, while aiming to provide a foundation for the multi-step 2T-POT model. We leave to future work the adoption of longer time spans where we aknowledge Hawkes arrivals dynamics may require recalibration to mitigate potential prediction errors amplified by the recursive nature of multi-step forecasting. The forecast produced by the asymmetric Hawkes model are compared with those produced by the GARCH-EVT model to evaluate which conditional EVT approach better represents extreme log-returns. Below is a presentation of the GARCH-EVT model used in the comparison, while Table II summarizes the backtesting results.

3.1 GARCH-EVT

We extend the GARCH-EVT model "$G_1^S(a_u)$" in [2] to multi-step. Also in this case we apply the recursive framework for multi-step prediction: The conditional variance σ_{t+n}^2 at each future time step $t+n$ is updated recursively: $\hat{\sigma}_{t+n}^2 = \omega + \alpha\hat{\epsilon}_{t+n-1}^2 + \beta\hat{\sigma}_{t+n-1}^2$. Residuals $\hat{\epsilon}_{t+n}$ are simulated at each step and the magnitudes of exedances are modeled using GPD consistently with $G_1^S(a_u)$

Table 2: Backtesting summary of the KS test statistics (proportion of rejections of the null hypothesis), unconditional coverage test (UC) and conditional violation loss (for tests explanation please refer to [2]). The Hawkes model consistently outperforms the GARCH model over the 5-day forecasting horizon in terms of accuracy. Future work may extend the backtesting framework to include a broader range of coverage levels adn forecasting horizons.

	Left Tail		**Right Tail**	
	H	**G**	**H**	**G**
KS Test Residual Arrivals	.19	.33	.098	.21
KS Test Excess Magnitudes	.17	.23	.07	.23
UC	.05	.16	.04	.12
ES Violation Loss	.06	.13	.06	.09

3.2 Limitations and future work

It is important to acknowledge the potential for error propagation introduced by the multi-step model during recursive simulation. Small errors in early steps can accumulate and amplify over time, affecting the accuracy of long-horizon forecasts. In general for multi-step forecasting this is already a well-known challenge and this is accentuated in models with recursive updates like the 2T-POT Hawkes process. Future work could explore strategies to mitigate this issue, such as incorporating confidence intervals to quantify forecast uncertainty, conducting error analysis to understand propagation patterns, or applying regularization techniques to reduce sensitivity to initial conditions.

Appendix

Hawkes process

```python

# Decay kernel for the intensity function
def decay_kernel(t, beta):
    return beta * np.exp(-beta * t)

# Impact function for magnitudes
def impact_function(magnitude, xi, sigma):
    return magnitude / (1 + (xi / sigma) * magnitude)

def compute_background_intensity(gamma_left, gamma_right, a_u, lambda_expected):
    gamma_sum = gamma_left + gamma_right
    return (2 - gamma_sum) * a_u * lambda_expected

# Multi-step intensity function with GPD-based impact
def intensity_function(mu, gamma, beta, event_times, magnitudes, current_time, n, xi,
    sigma, delta_t):
    future_time = current_time + n  # Time horizon for step n

    if len(event_times) == 0:  # No historical arrivals
        return mu
    else:
        # Incorporate both historical and predicted arrivals
        contributions = [
            gamma * decay_kernel(future_time - t, beta) * impact_function(m, xi, sigma)
            for t, m in zip(event_times, magnitudes) if t < future_time
        ]
        return mu + np.sum(contributions)

# Hawkes process simulation
def simulate_hawkes(mu, gamma, beta, max_time, gpd_params, n, delta_t):
    event_times = []
    magnitudes = []
    current_time = 0

    while current_time < max_time:
        lambda_t = intensity_function(
            mu=mu,
            gamma=gamma,
            beta=beta,
            event_times=event_times,
            magnitudes=magnitudes,
            current_time=current_time,
            n=n,  # Increment by one step at each iteration
            xi=gpd_params[0],
            sigma=gpd_params[1],
            delta_t=delta_t
        )
        next_time = current_time + np.random.exponential(1 / lambda_t)
        if next_time > max_time:
            break

        # Append new arrival and its magnitude
        event_times.append(next_time)
        magnitudes.append(stats.genpareto.rvs(c=gpd_params[0], scale=gpd_params[1]))
        current_time = next_time

    return np.array(event_times), np.array(magnitudes)
```

Log-likelihood function

```python
    def arrivals_log_likelihood(params, event_times, T, n, delta_t):
    # Extract parameters
    mu, gamma_left, gamma_right, beta = params

    # Initialize log-likelihood
    log_likelihood = 0
```

```
    # Iterate over event times for multi-step prediction
    for t in event_times:
        for step in range(1, n + 1):  # Multi-step integration
            lambda_t_n = intensity_function(
                mu=mu,
                gamma=gamma_left + gamma_right,
                beta=beta,
                event_times=event_times[event_times < t],
                magnitudes=magnitudes[event_times < t],
                current_time=t + step * delta_t,
                n=step,
                delta_t=delta_t,
                xi=xi,
                sigma=sigma
        )

            # Add log of intensity to the log-likelihood
            log_likelihood += np.log(lambda_t_n)

    # Compensator to account for the integral over time
    compensator = mu * T + (gamma_left + gamma_right) / beta * (
        len(event_times) - np.sum(np.exp(-beta * (T - event_times)))
    )

    # Return negative log-likelihood for optimization
    return -(log_likelihood - compensator)

# Log-likelihood for GPD magnitudes
def magnitudes_log_likelihood(params, magnitudes):
    xi, sigma = params
    return -np.sum(stats.genpareto.logpdf(magnitudes, c=xi, scale=sigma))

# Total log-likelihood function
def total_log_likelihood(params, event_times, magnitudes, T, n, delta_t):
    mu, gamma_left, gamma_right, beta, xi, sigma = params
    arrival_likelihood = arrivals_log_likelihood([mu, gamma_left, gamma_right, beta],
        event_times, T, n, delta_t)
    magnitude_likelihood = magnitudes_log_likelihood([xi, sigma], magnitudes)
    return arrival_likelihood + magnitude_likelihood
```

Calibration using SLSQP

```
 # Constraints for optimization
    constraints = [
    # Stability constraint: gamma_left + gamma_right < 2
    {'type': 'ineq', 'fun': lambda params: 2 - (params[1] + params[2])},
    # Expected average intensity constraint: a_left_right * lambda      2 * a_u * delta_t
    {'type': 'eq', 'fun': lambda params: params[0] * a_u - 2 * a_u * delta_t}
    ]

    # Optimize total log-likelihood using SLSQP
    result = minimize(
        total_log_likelihood,
        x0=initial_params,
        args=(event_times, magnitudes, max_time, n, delta_t),
        bounds=[(0, None), (0, 2), (0, None), (0, None), (0, None), (0, None)],
        constraints=constraints,
        method='SLSQP'
     )
```

Compensator function for arrivals trasformation

```
    def compute_compensator(event_times, intensity_func, mu, gamma, beta, magnitudes,
        xi, sigma, n, delta_t):
    compensators = []
    for t in event_times:
        lambda_integral = sum(
```

```python
            intensity_func(mu, gamma, beta, event_times[:i], magnitudes[:i],
                event_times[i], xi, sigma, n, delta_t)
            for i in range(len(event_times)) if event_times[i] <= t
        )
        compensators.append(lambda_integral)
    return np.array(compensators)
```

References

[1] M. F. Tomlinson, D. Greenwood, and M. MuchaKruczy´nski, Asymmetric excitation of left- and right-tail extreme events probed using a Hawkes model: Application to financial returns, Phys. Rev. E 104, 024112 (2021).

[2] Tomlinson, M. F., Greenwood, D., and Mucha-Kruczyński, M. 2T-POT Hawkes model for left- and right-tail conditional quantile forecasts of financial log-returns: out-of-sample comparison of conditional EVT models. arXiv preprint arXiv:2202.01043v2, 2022.

[3] L. de Haan and A. Ferreira, Extreme Value Theory, Springer Series in Operations Research and Financial Engineering (Springer, New York, 2006).

[4] L. R. Glosten, R. Jagannathan, and D. E. Runkle, On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks, J. Finance 48, 1779 (1993).

[5] R. F. Baumeister, E. Bratslavsky, C. Finkenauer, and K. D. Vohs, Bad is Stronger than Good, Rev. Gen. Psychol. 5, 323 (2001).

[6] J. Nocedal and S. J. Wright, Sequential Quadratic Programming, in Numerical Optimization (Springer, New York, 2006) Chap. 18, pp. 529–562.