

稀疏取樣下 High- σ 邊界估計的穩定性On the Stability of High- σ Boundary Estimators Under Sparse Sampling

DigWise Technology Corporation, Ltd.

1. High- σ 邊界估計：有限樣本下的高信賴策略

在實務應用中，特別是半導體製程與測試環境下，我們往往受限於時間、計算資源，或實體樣本的取得成本，無法獲得大量數據。然而，產品品質與異常風險評估仍需對高 σ 區間的邊界（如 3.5σ 甚至 4σ ）做出可靠判斷。這樣的場景，要求我們在有限樣本中，仍能推估出具信心的 rare-event 區域。

為深入探討此問題，我們設計了三組具代表性的統計分佈範例，分別為常態分佈（Normal）、對數常態分佈（Log-normal），以及具結構性的多峰分佈（Multimodal）。每組分佈皆先產生 100,000 筆樣本作為 ground truth，並從中多次隨機抽樣 500 筆資料，模擬有限樣本條件下的邊界推估實驗。如 Example 1 所示。

Example 1 Population and Subsamples

```
# On the Stability of High-Sigma Boundary Estimators Under Sparse Sampling
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats, interpolate
from sklearn.preprocessing import PowerTransformer

def sigma_percentage(sigma):
    '''return left and right % boundary based on the specified sigma value'''
    return (stats.norm.cdf((-sigma, sigma))*100)

def load_data(size, sub, case='multimodal'):
    np.random.seed(42) # mask this line for evaluation
    if case=='normal':
        v0 = np.random.normal(loc=0.2, scale=0.2, size=size)*4 + 5
    elif case=='log-normal':
        v0 = np.random.lognormal(mean=0.2, sigma=0.2, size=size)*4
    else: # multimodal
        v0 = np.concatenate([
            np.random.lognormal(mean=1, sigma=0.2, size=size//2)+2,
            np.random.normal(loc=6, scale=0.3, size=size//2)
        ])
    v1 = np.random.choice(v0, size=sub) # sub samples
    return v0, v1
```

如圖 Figure 1 所示，在常態與對數常態這兩類典型分佈中，即使僅有 500 筆樣本，透過 Power Transform (如 Box-Cox 或 Yeo-Johnson) 後計算出的 $\mu \pm 3.5\sigma$ 邊界，仍能與真實分佈高度吻合。這顯示 Power Transform 在結構單一旦具對稱性的分佈中，具有良好的穩定性與泛化能力。

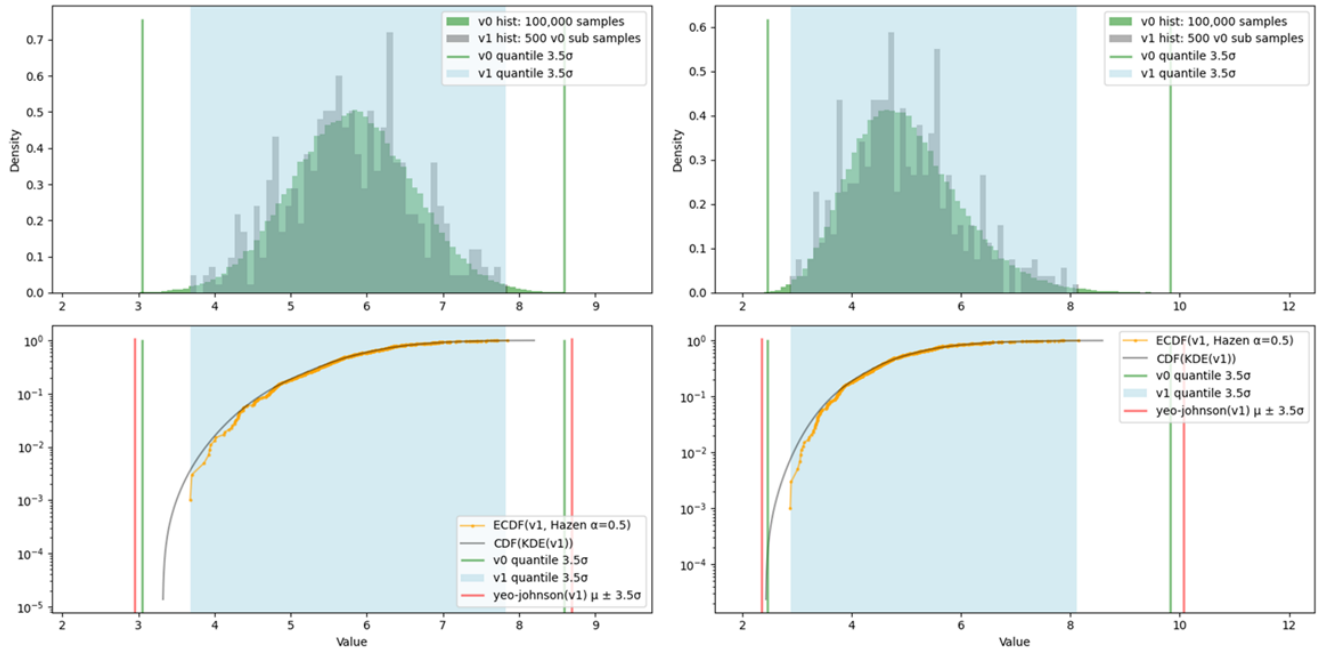


Figure 1 Transform-Based Boundary Estimation

然而，當資料來自複雜或非典型分佈（例如由兩個不同分佈混合而成），Power Transform 的效果便顯得有限。特別是在高 σ 區間，這類變換難以準確捕捉尾部行為，導致推估邊界與真實值產生明顯偏差，進而增加誤判風險。如 Figure 2 所示，此現象在混合分佈中尤為明顯。

這些觀察結果為有限樣本下的 high-sigma 邊界評估提供了策略性選擇的依據，並突顯了統計建模假設與資料特性不符時，可能帶來潛在的高風險決策錯誤。實驗顯示，資料分佈結構對邊界推估方法的穩定性與可信度至關重要。對於接近高斯分佈的資料，Power Transform 是一個輕量且準確的工具；然而，對於如半導體測試等具有高度非線性和結構複雜性的資料，更彈性的非參數方法（如 KDE 或 GMM）可能提供更穩健的邊界估計。

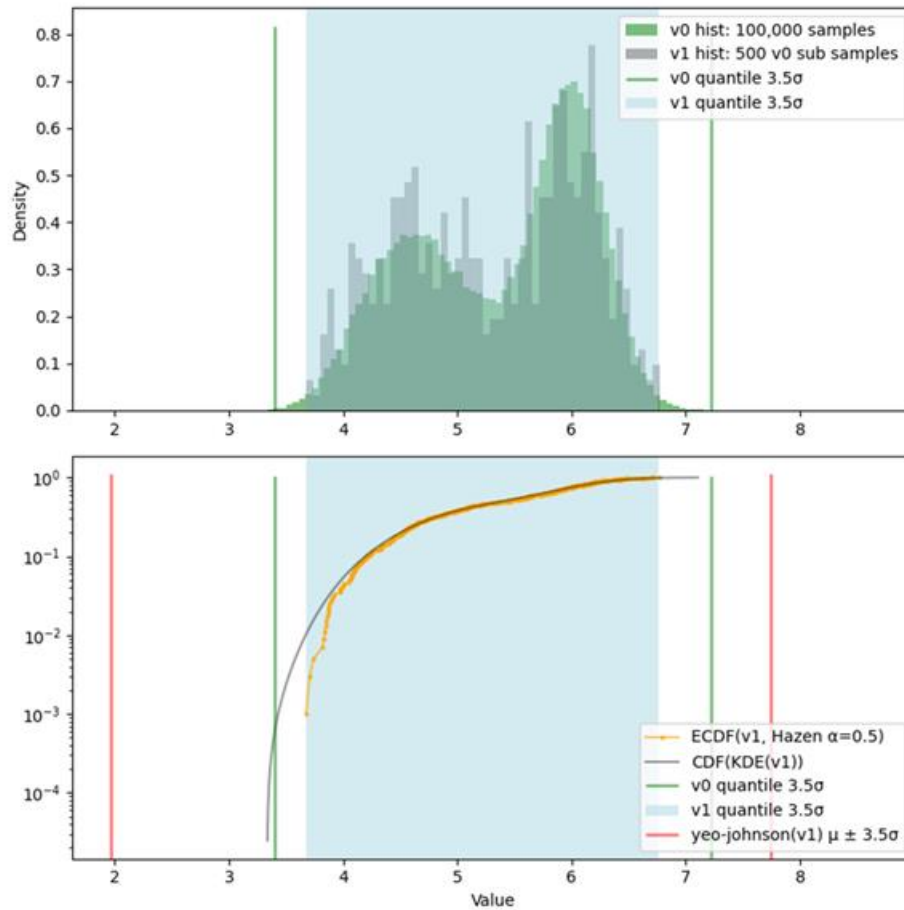


Figure 2 Power Transform Boundary Distortion in Multimodal Distributions

2. ECDF 與 KDE 方法在 High-Sigma 邊界評估的穩定性分析

當樣本數足夠大（例如超過 1,000 筆），由樣本排序構建的經驗累積分佈函數（ECDF）能穩定地逼近真實的母體分佈。此時，反轉 ECDF 以獲得高 sigma 區間的百分位點函數（PPF(ECDF)）邊界，其可信度與準確性均較高，是高效估算 rare-event 的理想方法。如圖 Figure 3 所示。

然而，在樣本數較少的情況下（例如 500 筆），ECDF 本身會出現階梯狀跳躍（下圖中橘色線），導致其反轉函數 PPF(ECDF) 對單點樣本變化高度敏感，特別是在尾端區域，邊界估計可能出現不連續、偏移或高變異的情況。

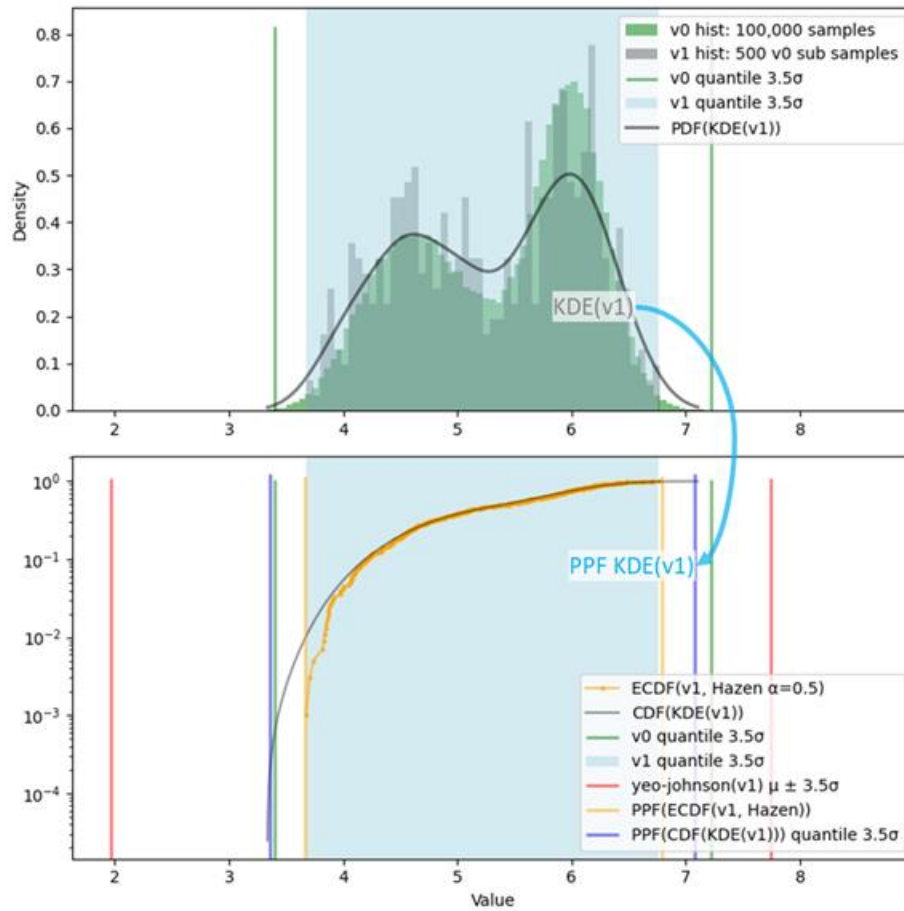


Figure 3 PPF(ECDF) for High-Sigma Boundary Estimation

在這種情境下，KDE (Kernel Density Estimation) 提供了一種平滑替代方案。透過核函數估計機率密度，再進行積分以取得平滑的 CDF，並反轉為 PPF(CDF)，能夠緩解 ECDF 在高 sigma 區域的波動問題。雖然 KDE 本身也依賴核寬參數 (bandwidth) 的選擇，但在多數情況下，它對於稀疏樣本在尾端的估計穩定性通常優於 ECDF。

相較於 Power Transform 方法，雖然 ECDF 與 KDE-based 方法在樣本數較少時可能出現較大波動，但在處理結構複雜的分佈時，能更真實地捕捉尾端行為，提供更可靠的 high-sigma 邊界推估。完整程式碼如 Example 2 所示。

Example 2 Quantifying Rare Events with Limited Observations

```

def order_statistic_ecdf(v, alpha=0.5): #  $F(i) = (i - \alpha)/(n + 1 - 2\alpha)$ 
    # Hazen 1914,  $\alpha=0.5$ , Weibull 1951  $\alpha=0$ , Blom 1958,  $\alpha=0.375$ , Gringorten 1963  $\alpha=0.44$ 
    ecdf = (stats.rankdata(sorted(v), method='average') - alpha) / (len(v) + 1 - 2*alpha)
    return ecdf

def pt_quantile(v, sigma, method='yeo-johnson'): # power transform: box-cox, yeo-johnson
    pt = PowerTransformer(method=method)
    x = pt.fit_transform(v.reshape(-1,1)).ravel()
    b = pt.inverse_transform(
        (x.mean() + np.array([-sigma*x.std(), sigma*x.std()])).reshape(-1,1) ).ravel()
    return x, b

def ppf_CDFKDE(v, q, res=1000):
    kde = stats.gaussian_kde(v)
    margin = 1.5*v.std()*kde.factor
    t = np.linspace(v.min()-margin, v.max()+margin, res)
    pdf = kde(t)
    cdf = np.cumsum(pdf)/pdf.sum()
    ppf = interpolate.interp1d(cdf, t, fill_value='extrapolate')(q) # PPF(CDF((KDE(v))))
    return t, pdf, cdf, ppf

# load data and sub sample
sigma = 3.5
q = sigma_percentage(sigma)/100
v0, v1 = load_data(size=1000_000, sub=500, case='multimodal')

q0 = np.quantile(v0, q=q) # v0 quantile
q1 = np.quantile(v1, q=q) # v1 quantile

# order statistic:  $F(i) = (i - \alpha)/(n + 1 - 2\alpha)$ 
e1 = order_statistic_ecdf(v1, alpha=0.5) # Hazen
p1 = interpolate.interp1d(e1, sorted(v1), fill_value='extrapolate')(q) # PPF ECDF

# power transform: box-cox, yeo-johnson
x1, b1 = pt_quantile(v1, sigma, method='yeo-johnson')

# CDF(KDE), PPF
t1, k1, c1, pq1 = ppf_CDFKDE(v1, q, res=1000) # PPF CDF

# visualization
plt.figure(figsize=(8,8))
ax1 = plt.subplot(211)
ax1.hist(v0, bins=100, alpha=0.5, density=True, color='g', label=f'v0 hist: {len(v0):,} samples')
ax1.hist(v1, bins=50, alpha=0.3, density=True, color='k', label=f'v1 hist: {len(v1):,} sub samples')
ax1.vlines(q0, *ax1.get_ylim(), lw=2, color='g', alpha=0.5, label=f'v0 quantile {sigma}\sigma')
ax1.axvspan(*q1, lw=0, color='lightblue', alpha=0.5, label=f'v1 quantile {sigma}\sigma')
ax1.plot(t1, k1, alpha=0.5, lw=2, c='k', label='PDF(v1)')
ax1.set_ylabel('Density')
ax1.legend()
ax2 = plt.subplot(212, sharex=ax1)
ax2.plot(sorted(v1), e1, c='orange', marker='o', ms=2, alpha=0.6, label='ECDF(v1, Hazen  $\alpha=0.5$ )')
ax2.plot(t1, c1, ms=1, alpha=0.4, c='k', label='CDF(v1)')
ax2.vlines(q0, *ax2.get_ylim(), lw=2, color='g', alpha=0.5, label=f'v0 quantile {sigma}\sigma')
ax2.axvspan(*q1, lw=0, color='lightblue', alpha=0.5, label=f'v1 quantile {sigma}\sigma')
ax2.vlines(b1, *ax2.get_ylim(), lw=2, color='r', alpha=0.5, label=f'yeo-johnson(v1) {sigma}\sigma')
ax2.vlines(p1, *ax2.get_ylim(), lw=2, color='orange', alpha=0.5, label=f'PPF(ECDF(v1)) {sigma}\sigma')
ax2.vlines(pq1, *ax2.get_ylim(), lw=2, color='b', alpha=0.5, label=f'PPF(CDF(v1)) {sigma}\sigma')
ax2.set_xlabel('Value')
ax2.set_yscale('log')
ax2.legend()
plt.tight_layout()
    
```

3. Handling Multimodal Distributions with GMM

為了探討在結構複雜的分佈下，如何在有限樣本中建立穩健且具泛化能力的 high-sigma 邊界估計方法，我們引入 Gaussian Mixture Model (GMM) 作為建模工具。本實驗以先前設計的 multimodal 分佈所產生的 1,000,000 筆資料作為真實樣本空間，並從中隨機抽取 1,000 筆，模擬如 SPICE 模擬等受限於計算資源的實務場景。

我們以這 1,000 筆資料建構三個高斯核的 GMM，學習原始資料的多模態特性。GMM 能有效保留資料內在結構，即使僅使用約 500 筆樣本，也能在 high-sigma 區間提供穩定的邊界推估。

為評估 GMM 的泛化能力與高尾部推估表現，我們利用其生成的 1,000,000 筆 synthetic data，分別透過 PPF(ECDF) 與 PPF(CDF) 方法推估 high-sigma 邊界，並與真實分佈進行比較，如圖 Figure 4 所示，推估的邊界相當一致。

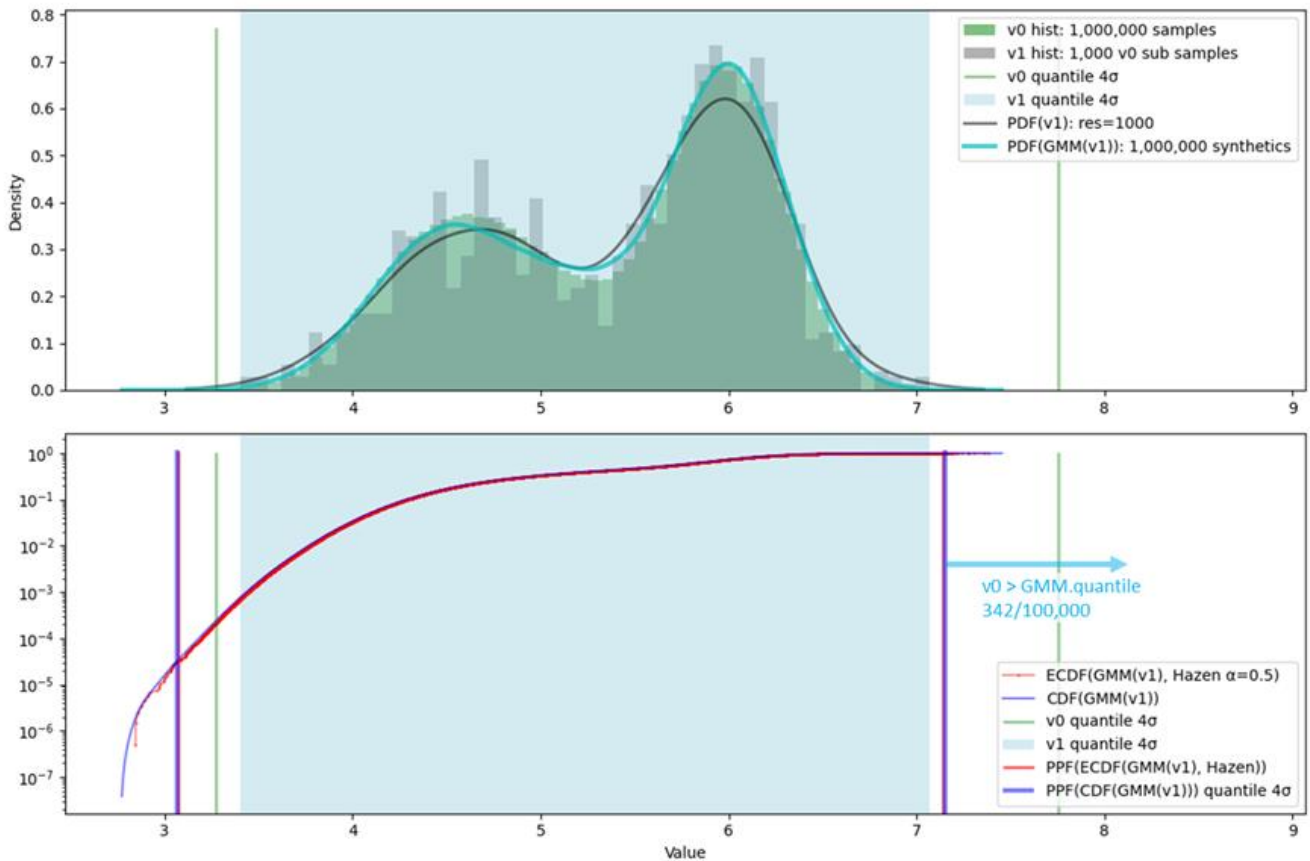


Figure 4 High-Sigma Boundary Estimation via GMM-Synthetic Data

值得注意的是，透過有限樣本建構 GMM 並進一步生成百萬筆 synthetic data 所推估出的邊界，其覆蓋範圍約為百萬分之 342，然而在理論分佈中， 4σ 單尾區域僅佔約百萬分之 31.7。這樣的對比凸顯一個關鍵問題：為了捕捉極端尾部的精度，所需的資料量與建模成本可能遠高於實務所能負荷。

儘管 GMM 推估的尾部邊界與真實分佈仍存差距，然而在樣本有限的情境下，這樣的誤差可被視為兼顧準確性與效率的務實折衷。為進一步提升尾部極值的準確度與信心水準，下一節將介紹極值理論 (EVT) 及結合 tail-aware GMM 技術，以加強對極端事件的建模能力。

4. Modeling Tail Risk Using Extreme Value Theory (EVT)

我們已探討在有限樣本下高信賴邊界評估所面臨的挑戰，並逐步發展出對於多模態結構進行建模的策略。為了更進一步處理極端事件（如極大值或極小值）在尾部的表現，本章引入極值理論 (Extreme Value Theory, EVT) 的概念，作為強化 high-sigma 分析的補充工具。極值理論專注於研究資料的最大（或最小）值行為，其核心可分為兩種主要模型：

- Generalized Extreme Value (GEV)：由非重疊區塊的最大值所建構，適用於 block maxima 模型。
- Generalized Pareto Distribution (GPD)：針對超過某門檻值 (threshold) 的樣本進行擬合，屬於 peaks-over-threshold (POT) 模型，更靈活且在資料量受限時表現更佳。

為說明 GEV 與 GPD 在大樣本下的穩定性，Figure 5 展示了兩者在理想條件下的表現：當樣本數充足時，無論是透過 block maxima (對應 GEV) 或超過 90% 閾值的 tail exceedances (對應 GPD)，其極端行為皆可與理論模型精準吻合，突顯 EVT 在極值建模上的強大表現。

然而實務上常受限於有限樣本（例如僅 500 筆），使得直接估計 GEV 分佈變得困難。為此，我們採用 bootstrap block maxima 策略，例如自原始樣本中重複抽取 10,000 次、每次 100 筆區塊並取最大值，以重建極端分佈。結果顯示，即使資料量有限，透過重採樣所得的最大值分佈仍展現穩定形態，足以近似 GEV 理論，成為實務中建模極值風險的可行替代方案。

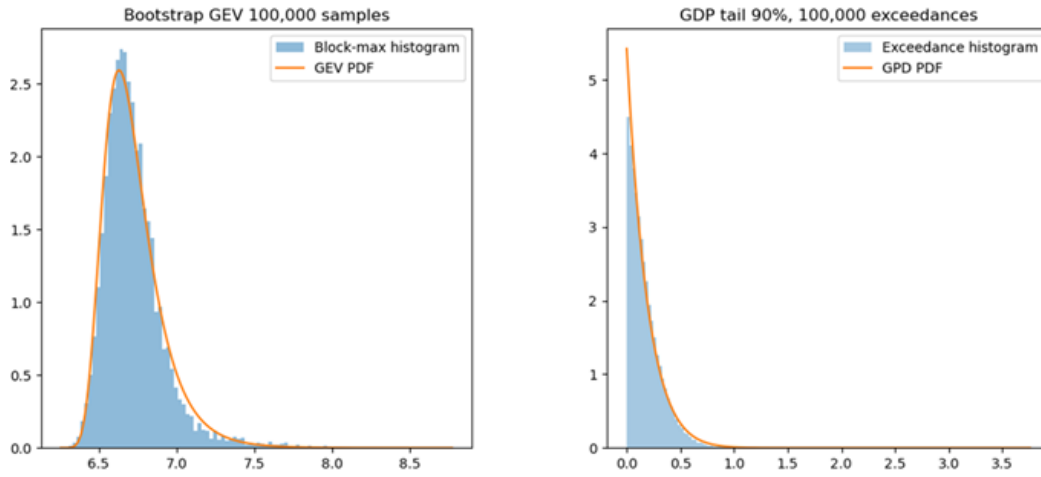


Figure 5 Extreme Value Theory (EVT)

為評估擬合結果的可靠性與極值邊界的穩定性，我們進一步透過 Q-Q plot 比較實際 exceedances 與 EV 理論分佈的分位數差異，以檢驗 tail 擬合的吻合程度；同時藉由 Return Level plot 對應不同重現週期的估計極值，視覺化 EV 模型在高信心水平下的預測表現，如圖 Figure 6。

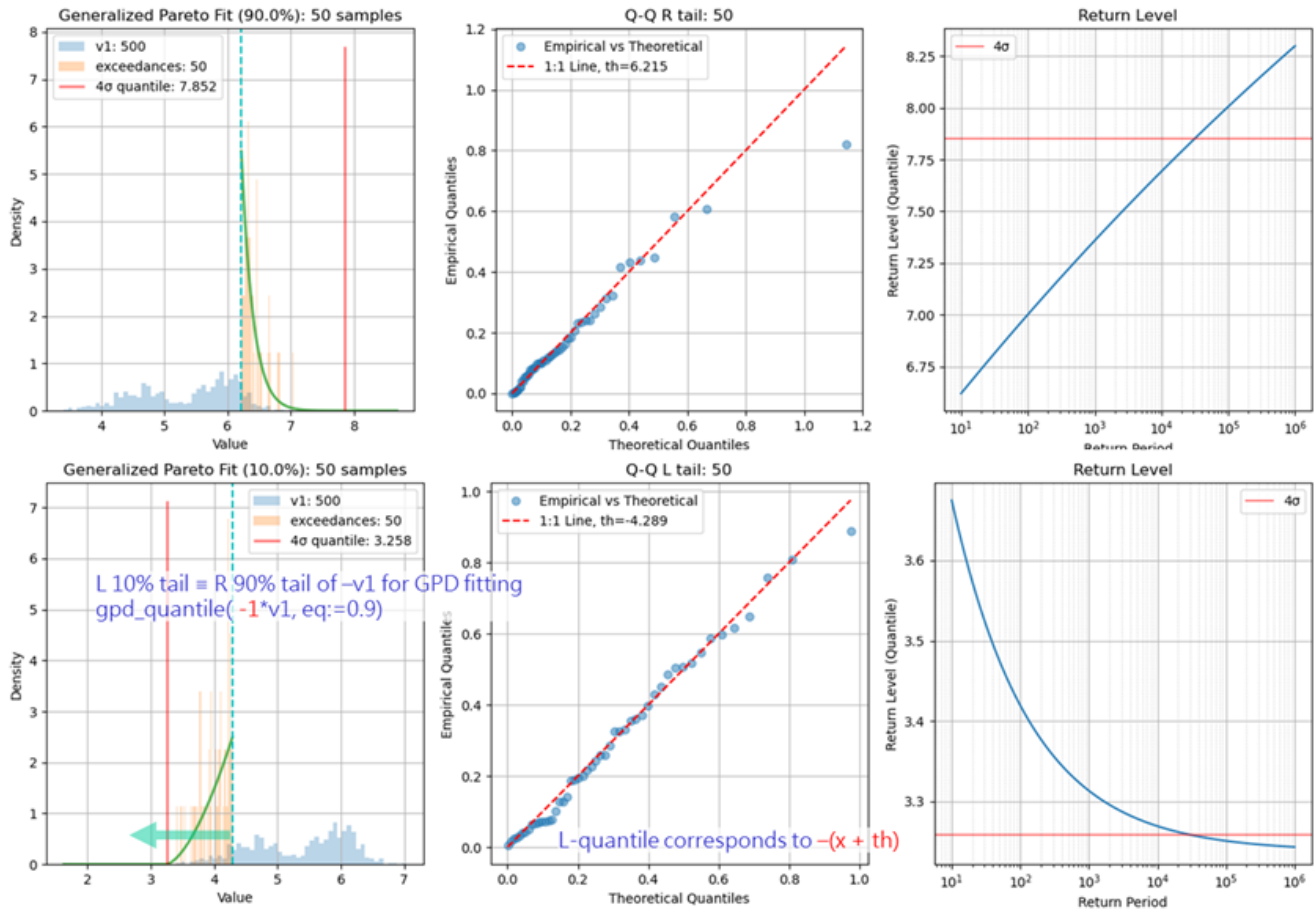


Figure 6 Double-side GPD Tail Fitting & QA

當 Q-Q plot 上的 empirical quantiles 大多落在 1:1 線下方，表示模型對極端值的預測高於實際觀測，屬於較為保守的估計。儘管可能高估風險，在樣本有限的情況下，這種悲觀預測有助於提高對極值事件的信心。而當 return-level plot 的 y 軸趨於收斂，則表示模型在高重現週期下的極值估算已穩定，未再受到樣本波動影響，提供一個更可靠的長期風險評估依據。

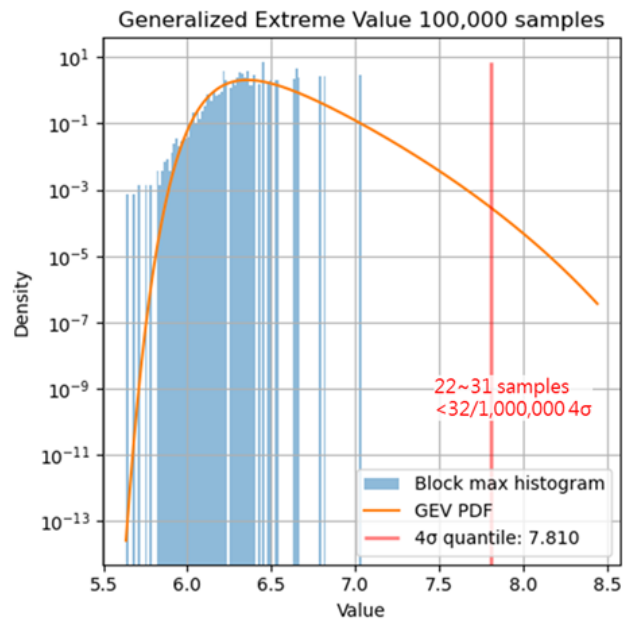


Figure 7 Bootstrap GEV Tail Fitting

當樣本數量有限（如僅有 500 筆觀測值）時，使用 block maxima 方法抽樣可能會導致不穩定結果，進而影響 GEV 模型的擬合效果。如 Figure 7 所示，儘管 GEV 模型能在 4 sigma 分位數上接近原始樣本，但擬合效果不穩定。在這種情況下，Generalized Pareto Distribution (GPD) 可以作為一個折衷方案，聚焦於超越高閾值後的 tail exceedances，雖然 GPD 與 GEV 方法在某些情況下表現有所差異，但在樣本量有限的情境下，提供了一種相對可行的選擇。

範例 Example 3 展示如何從僅有 500 筆的有限樣本中擷取上下 10% tail 的 exceedances，並以 GPD 模型進行擬合。擬合完成後，我們將使用 GPD 的 percent point function (PPF) 來預測左右兩端的高-sigma 分位數邊界（例如 4 σ 、5 σ 等極端值），這些邊界可用於風險評估或異常偵測。對於左邊界，透過對採樣輸入取負值（-v1）可以生成對應的邊界數值；而對應的分位數可以通過反轉公式 $-(x + th)$ 得到，其中 th 是 GPD 模型中的閾值參數，x 是對應的尾端值。這樣的處理方式確保了左右邊界的對稱性，並且便於不同方向上的極端值預測。

Example 3 GPD Tail Modeling and Quantile Reliability

```

%% GPD Tail Fitting
def gpd_quantile(v, eq=0.9):
    th = np.quantile(v, eq) # quantile threshold for exceedances
    ve = v[v > th] - th # exceedances, keep floc=0, loc=0
    c, loc, scale = stats.genpareto.fit(ve, floc=0) # loc=0
    x = np.linspace(0, ve.max()*3, 1000)
    p = stats.genpareto.pdf(x, c=c, loc=loc, scale=scale) # PDF
    return (th, ve), (c, loc, scale), (x, p) # exceedances, gpd, pdf

v0, v1 = load_data(size=1000000, sub=500, case='multimodal')
q = sigma_percentage(4)/100 # 4 $\sigma$  quantile

sign = 1 # (-1, 1): (left, right)
(th, ve), (c, loc, scale), (x, p) = gpd_quantile(v1*sign, eq:=0.9) # negative v1 for left-side
b = sign*(stats.genpareto.ppf(q[1], c=c, loc=loc, scale=scale) + th) # PPF quantile

# GPD fitting
plt.figure(figsize=(14, 5))
ax0 = plt.subplot(131, title=f'Generalized Pareto Fit: {len(ve)} samples')
ax0.hist(v1, bins=50, density=True, alpha=0.3, label=f'v1: {len(v1)}')
ax0.hist(sign*(ve+th), bins=50, density=True, alpha=0.3, label=f'exceedances: {len(ve)}')
ax0.plot(sign*(x+th), p, lw=2, alpha=0.8)
ax0.axvline(sign*th, color='c', linestyle='--')
ax0.vlines(b, *ax0.get_ylim(), lw=2, alpha=0.5, color='r', label=f'{sigma} $\sigma$  quantile: {b:.3f}')
ax0.set_xlabel('Value')
ax0.set_ylabel('Density')
#ax0.set_yscale('log')
ax0.legend()
ax0.grid(which='major', linestyle='-', alpha=0.9)
ax0.grid(which='minor', linestyle=':', alpha=0.6)

# Q-Q
probs = np.linspace(1e-3, 1-1e-3, len(ve))
qt = stats.genpareto.ppf(probs, c, loc=loc, scale=scale) # theoretical quantile

ax1 = plt.subplot(132, title=f'Q-Q {"R" if sign>0 else "L"} tail: {len(ve):,}')
ax1.plot(qt, sorted(ve), 'o', alpha=0.5, label='Empirical vs Theoretical')
ax1.plot(np.quantile(qt, [0,1]), np.quantile(qt, [0,1]), 'r--', label=f'1:1 Line, th={th:.3f}')
ax1.set_xlabel('Theoretical Quantiles')
ax1.set_ylabel('Empirical Quantiles')
ax1.legend()
ax1.grid(which='major', linestyle='-', alpha=0.9)
ax1.grid(which='minor', linestyle=':', alpha=0.6)

# return level
T = np.logspace(1, 6, 100) # occurs once per T observations (expected occurrence interval)
return_levels = sign*(stats.genpareto.ppf(1 - 1/T, c, loc=loc, scale=scale) + th)

ax2 = plt.subplot(133, title='Return Level')
ax2.plot(T, return_levels)
ax2.axhline(b, alpha=0.5, color='r', label='4 $\sigma$ ')
ax2.set_xscale('log')
ax2.set_xlabel('Return Period') # occurs interval
ax2.set_ylabel('Return Level (Quantile)')
ax2.grid(which='major', linestyle='-', alpha=0.9)
ax2.grid(which='minor', linestyle=':', alpha=0.6)
ax2.legend()
plt.tight_layout()
    
```

5. Combining Tail-GMM and GPD for Robust Boundaries

GMM 透過多個高斯分佈組合擬合資料整體形狀，但由於高斯分佈尾部衰減迅速（屬於 exponential tail），在捕捉極端值時存在明顯限制。為改善此問題，處理多模態資料時可採用結合式策略：先以 GMM 建構主要結構，特別適用於多峰（multimodal）分佈，再透過適當初始化（如將 means_init 延伸至尾部）強化尾部辨識，此策略稱為 tail-GMM。

然而，若資料本身具重尾特性（如 lognormal 或 Pareto-like 分佈），GMM 即使增加成分數也難以準確擬合尾部行為，常低估極端區域的機率密度。此時可引入 GPD 補強尾部建模，提供更穩健的 quantile 邊界預測。即便樣本量有限（如僅 500 筆），GPD 仍能有效保留尾部結構。透過 tail-GMM 與 GPD 的整合，可同時保留整體結構與尾部行為，在樣本不足時仍維持高信心的極值估計，是一種兼顧效率與準確度的實用建模策略。

Example 4 中，我們將 GPD 擬合與 quantile 計算整合為一個函式，例如先分別對左右兩側各取 90% 的 exceedances 擬合 GPD 分布，接著透過 PPF（percent point function）計算左右 4σ 邊界。擬合結果如圖 Figure 8 所示。

Example 4 Double-Sided GPD Tail Fitting

```
%% Double-sided GPD Tail Fitting
def gpd_quantile2(v1, eq=0.9, q=0.99997): # pick eq% tail for exceedances
    # right
    v = v1
    th1 = np.quantile(v, eq) # quantile threshold for exceedances
    ve1 = v[v > th1] - th1 # exceedances, keep floc=0, loc=0
    c1, loc1, scale1 = stats.genpareto.fit(ve1, floc=0) # loc=0
    x1 = np.linspace(0, ve1.max()*3, 1000)
    p1 = stats.genpareto.pdf(x1, c=c1, loc=loc1, scale=scale1)
    b1 = stats.genpareto.ppf(q[1], c=c1, loc=loc1, scale=scale1) + th1

    # left
    v = -v1
    th2 = np.quantile(v, eq) # quantile threshold for exceedances
    ve2 = v[v > th2] - th2 # exceedances, keep floc=0, loc=0
    c2, loc2, scale2 = stats.genpareto.fit(ve2, floc=0) # loc=0
    x2 = np.linspace(0, ve2.max()*3, 1000)
    p2 = stats.genpareto.pdf(x2, c=c2, loc=loc2, scale=scale2)
    b2 = -(stats.genpareto.ppf(q[1], c=c2, loc=loc2, scale=scale2) + th2)

    return b2, b1 # left, right
```

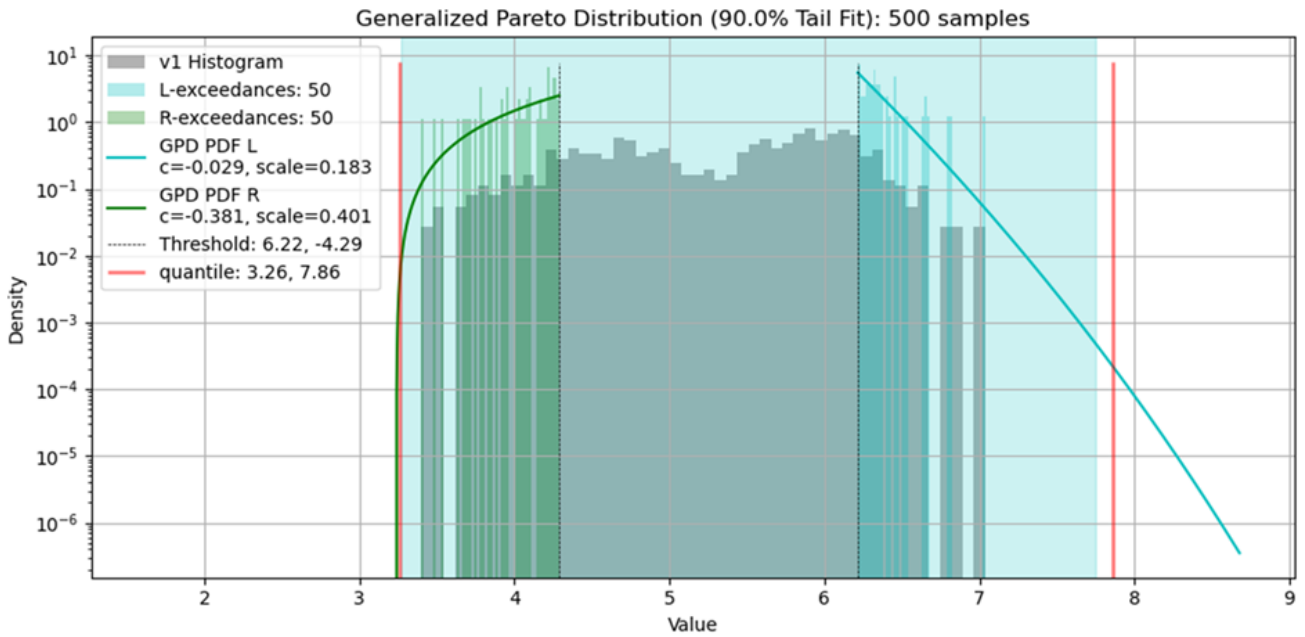


Figure 8 GPD Fitting for Subsample v1 (Log-Scale View)

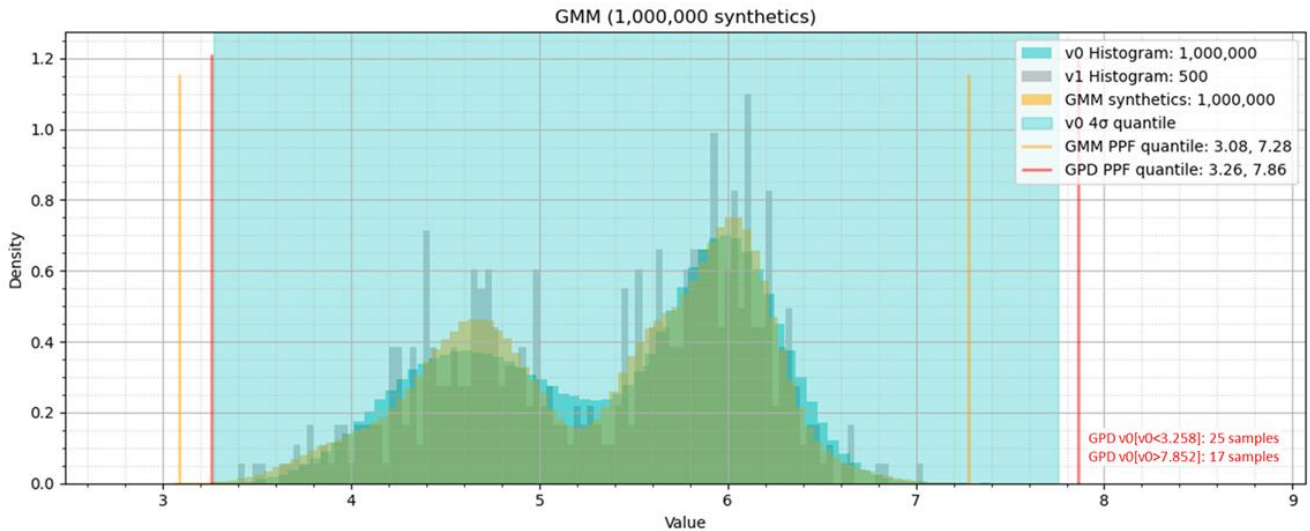


Figure 9 Tail-aware GMM with GPD Strategy for High-Sigma Confidence

如圖 Figure 9 所示，橘色直方圖為由 GMM 生成的樣本分佈。GPD 所預測的 4σ 分位數（紅色邊界）與原始母體 v_0 （100 萬筆資料）相比，僅有 25 與 17 筆樣本落在邊界之外，顯示其估計結果已高度接近 v_0 的實際 4σ 分位（淺藍區間），且略帶保守，具有良好的風險涵蓋能力。

Example 5 展示一個策略範例，整合採用 5 個高斯核 (以 `means_init` 初始化) 的 tail-GMM 與 GPD tail fitting，用以提升尾部事件的建模表現。

儘管 GMM 的尾部可能低估風險，我們可以利用 GPD 所估的 σ 邊界作為決策層，來判定異常樣本或設置警示門檻 (如製程控制或 binning)，以修正 GMM 的風險判斷。當資料量不足以支持 GEV 模型時，這樣的流程可提供高信心且高解析度的尾部行為預測，是 EVT 在工程與風險管理中常見且強大的應用策略。

Example 5 Combining tail-GMM and GPD for Tail Event Modeling

```

%% Tail Event Modeling
v0, v1 = load_data(size=1000000, sub=500, case='multimodal')
q0 = np.quantile(v0, q=q)

# Tail-GMM
means_init = np.quantile(v1, np.linspace(0.01, 0.995, 5)).reshape(-1, 1)
gmm = GaussianMixture(
    n_components = 5,
    means_init = means_init,
    init_params = 'k-means++',
    random_state = 42
)
gmm.fit(v1.reshape(-1, 1))

vg = gmm.sample(1000_000)[0].flatten()

# GMM PPF quantile
x = np.linspace(v1.mean()-5*v1.std(), v1.mean()+5*v1.std(), 1000_000)
logprob = gmm.score_samples(x.reshape(-1, 1))
pdf = np.exp(logprob)
cdf = np.cumsum(pdf) / np.sum(pdf)
qg = interpolate.interp1d(cdf, x, fill_value='extrapolate')(q)
print(f'GMM v0[v0<{qg[0]:.3f}]: {len(v0[v0<qg[0]]):,} samples')
print(f'GMM v0[v0>{qg[1]:.3f}]: {len(v0[v0>qg[1]]):,} samples')

# GPD quantile
qp = gpd_quantile2(v1, eq=0.9)
print(f'GPD v0[v0<{qp[0]:.3f}]: {len(v0[v0<qp[0]]):,} samples')
print(f'GPD v0[v0>{qp[1]:.3f}]: {len(v0[v0>qp[1]]):,} samples')

plt.figure(figsize=(12, 5))
ax = plt.subplot(111, title=f'GMM ({len(vg):,} synthetics)')
ax.hist(v0, bins=100, density=True, color='c', alpha=0.5, label=f'v0 Histogram: {len(v0):,}')
ax.hist(v1, bins=100, density=True, color='k', alpha=0.2, label=f'v1 Histogram: {len(v1):,}')
ax.hist(vg, bins=100, density=True, color='orange', alpha=0.5, label=f'GMM synthetics: {len(vg):,}')
ax.axvspan(*q0, color='c', alpha=0.3, label='v0 4σ quantile')
ax.vlines(qp, *ax.get_ylim(), lw=2, alpha=0.5, color='r', label=f'GPD: {qp[0]:.2f}, {qp[1]:.2f}')
ax.set_xlabel('Value')
ax.set_ylabel('Density')
ax.grid(axis='both', which='major', linestyle='-', alpha=0.9)
ax.grid(axis='both', which='minor', linestyle=':', alpha=0.6)
ax.minorticks_on()
ax.legend()
plt.tight_layout()

```