Goldman Sachs

India Hackathon 2025

Quant

Presented by : **Arjun Verma**

Goldman Sachs

- Optimal Hedging Strategy
- Automated Market Making
- Exotic Option Pricing

Quant

Optimal Hedging Strategy

Keywords : Linear Algebra, Optimization principle, CVaR, LassoCV

Problem Formulation

Minimize the Value at Risk of the hedged portfolio while keeping the total capital cost of the hedge low.

$$\min_{\beta \in \mathbb{R}^n} \mathrm{VaR}_\alpha(\mathrm{PnL}_{\text{unhedged}} + R\beta) + \lambda \cdot \sum_{j=1}^{n} c_j |\beta_j|$$

where, β_j is the position size of stock j, R is return matrix, c_j is cost of the stock j. 309 stocks given in our case.

Assumptions

- Returns reflect future risk
- Stocks are liquid (long/short)
- Hedging cost is linear
- PnL is linear in returns

Constraints

- Positions β_j can be positive (long) or negative (short)
- Total cost: $\sum_{j=1}^{n} c_j |\beta_j| \leq \text{Budget}$

Current Approach Linear Programming

We will Optimize CVaR over VaR, as **CVaR is convex**, unlike VaR and **CVaR captures tail risk**, gives worst-case losses beyond the VaR threshold.

Given the inputs $p_i, \quad i = 1, \dots, T,$ PnL on day i, $C_j, \quad j = 1, \dots, n,$ Cost of stock j, $R_{i,j}$, return of stock j on day i. We define some variables

- $w_j = v_j^+ - v_j^-$, long $v_j^+ \geq 0,$ and short $v_j^- \geq 0,$ positions for stock j.
- t "we expect 95% of days to have total P&L" t (VaR threshold).
- $u_i \geq 0,$ measures how far below the t the total P&L on day i (if it drops).

Total PnL on day i being $p_i + \sum_{j=1}^{n} (v_j^+ - v_j^-) R_{i,j}$. and we define our **OBJECTIVE**

$$\min_{v^+,\, v^-,\, u,\, t} \underbrace{\sum_{j=1}^{n} \left[C_j \cdot (v_j^+ + v_j^-) \right]}_{\text{(A) total hedging cost}} + \underbrace{t}_{\text{(B) VaR threshold}} + \underbrace{\frac{1}{(1-\alpha)T} \sum_{i=1}^{T} u_i}_{\text{(C) average tail shortfall}}$$

Implementation and Observations

$$\underbrace{t - \left[p_i + \sum_j (v_j^+ - v_j^-) R_{i,j} \right]}_{\text{“how far below } t \text{ is day } i\text{?”}} \leq u_i \iff -\sum_{j=1}^{n} (v_j^+ - v_j^-) R_{i,j} - t - u_i \leq -p_i.$$

Writing in "$A\,x \leq b$" form :

$$\left[-R_{i,1}, \dots, -R_{i,n}, \ R_{i,1}, \dots, R_{i,n}, \ (\underbrace{-1}_{u_i \text{ column}}), \ (\underbrace{-1}_{t \text{ column}}) \right] \begin{pmatrix} v^+ \\ v^- \\ u \\ t \end{pmatrix} \leq p_i.$$

and our objective being:

$$c = \left[\lambda c_1, \lambda c_2, \dots, \lambda c_N, \lambda c_1, \lambda c_2, \dots, \lambda c_N, \frac{1}{(1-\alpha)T}, \dots, \frac{1}{(1-\alpha)T}, 1 \right]$$

Decreasing λ shifts focus toward risk minimization, resulting in **lower CVaR** but **higher hedging costs**.

Higher Costs

Better CVaR

Future Improvements Using Lasso CV

LassoCV offers a simpler, faster alternative to the CVaR method by selecting a sparse hedge that minimizes PnL error while controlling cost through L1-penalty, **often outperforming previous method in CvaR - Cost trade-off**.

$$\min_{\beta} \underbrace{\frac{1}{2T} \| \mathbf{p} - R\,\beta \|_2^2}_{\substack{\text{(Hedging error)} \\ \text{unhedged PnL loss}}} + \underbrace{\alpha \|\beta\|_1}_{\substack{\text{(Hedging cost)} \\ \text{sparse positions}}}$$

where, $w_j = \hat{\beta}_j / \sigma_j$, so hedged PnL is

$$\tilde{p}_i = p_i - \sum_j w_j R_{i,j}$$

Goldman Sachs

Optimal Hedging Strategy

Automated Market Making

Exotic Option Pricing

Quant

Automated Market Making

Keywords : Order-book, Market Making, Spread, Quoting Strategy, HFT

Problem Formulation

Let $t_0, \dots, t_{N-1}$ be the timestamps. At each t_n we observe best bid/ask $p_n^b\ p_n^a$ with sizes s_n^b, s_n^a & public trades $\mathcal{T}_n$. We choose quotes (b_n, a_n) and lot ℓ_n so that if a sell trade in $\mathcal{T}_n$ has price we buy $\le b_n$. If a buy trade has price $\ge a_n$ we sell. Inventory evolves as

$$I_n = I_{n-1} \pm \ell_n$$

with $a_n - b_n \ge 0.1$ and $|I_n| \le 20$ we seek Max PnL with no peeking into future data

Observations and Results

Price Downtrend Detected → shift bid/ask downward using trend & inventory adjustments to avoid accumulating long positions.
Large Ask-Size Spikes → Interpreted as sell-side walls; strategy tilts quotes downward using the order book imbalance adjustment.
Downward Momentum → EMA slope captures trend; quotes follow the drift via the momentum adjustment, reducing quote staleness.

70-120 PnL

All tested parameters, realized PnL ranged between 70-120.

-3 std. dev. of inventory size

Inventory remained low risk control despite carrying short positions.

Developments
Using Deep Reinforcement Learning

DRL replaces fixed parameters by **learning to quote** from LOB and trades, inventory, and volatility) and outputs bid/ask offsets from mid-price. Trained on historical data,**RL agents capture nonlinear patterns** (e.g. deep liquidity or momentum) and yield **10–30 %** higher risk-adjusted returns and lower inventory variance, than classical Avellaneda–Stoikov or imbalance-based strategies. Periodic retraining ensures adaptability to evolving market conditions. Can be modeled as a Markov-Decision-Process with:

State : Prices $(p_{t,k}^b, p_{t,k}^a)$,sizes $(s_{t,k}^b, s_{t,k}^a)$ & inventory I_{t-1} .
Action : $a_t = (\delta_t^b, \delta_t^a)$ so that bids are $m_t - \delta_t^b$,ask $m_t + \delta_t^a$
Reward Function : Rewards PnL, Punishes Inventory

$$r_t = \underbrace{1\{\text{ask filled}\}\, \ell_t\, a_t - 1\{\text{bid filled}\}\, \ell_t\, b_t}_{\text{realized PnL at } t} - \alpha\, |I_t| - \beta\, I_t^2 - \gamma\, 1\{b_t \ge a_t\}$$

1. Compute Mid-Price

$$\mu_t = \frac{p_t^a\, s_t^b + p_t^b\, s_t^a}{s_t^b + s_t^a + \varepsilon}$$
$$m_t = \frac{p_t^b + p_t^a}{4} + \frac{\mu_t}{2}$$

weights bid/ask by opposite side size to capture immediate imbalance.

2. Volatility effect

$$\phi_t = f(\sigma_t)$$

widens spreads proportionally to recent mid-price volatility

3. Liquidity effect

$$Q_t = s_t^b + s_t^a$$
$$\lambda_t = g(Q_t)$$

High total depth lets us tighten quotes; low depth forces wider spreads.

4. Momentum

$$\tau_t = \Theta(\{m_{t-L+1}, \dots, m_t\})$$

Measures short-term momentum by smoothing mids (EMA span=10) then extracting slope.

5. Order-Book Imbalance

$$\mathrm{OI}_t = \frac{s_t^b - s_t^a}{s_t^b + s_t^a + \varepsilon},$$

Measures relative strength of bid vs. ask sizes. Guides direction of quotes.

6. Half-Spread

$$\mathrm{DS}_t = h(\phi_t, \lambda_t, \mathrm{OI}_t)$$
$$\Delta_t = \frac{\mathrm{DS}_t \times \tau}{2}$$

Translates "spread in ticks" into a one-sided price offset from mid. High volatility widens, high liquidity narrows.

7. Inventory ratio

$$\rho_t = \frac{I_{t-1}}{I_{\max}}$$

Normalizes position to ±1.

8. Inventory Adjustment

$$\delta_t^{\mathrm{inv}} = \Phi(\rho_t, \Delta_t)$$

Pulls quotes toward mid price when inventory is large.

9. Momentum Adjustment

$$\delta_t^{\mathrm{trend}} = \Gamma(\tau_t)$$

Bias quotes along the detected trend.

10. Imbalance Adjustment

$$\delta_t^{\mathrm{imb}} = \Psi(\mathrm{OI}_t, \Delta_t)$$

Shifts quotes in direction of order-book pressure: buy pressure shifts quotes up

11. Form Quotes and round to ticks and Update Inventory

$$b_t^* = m_t - \Delta_t - \delta_t^{\mathrm{inv}} - \delta_t^{\mathrm{imb}} + \delta_t^{\mathrm{trend}}$$
$$a_t^* = m_t + \Delta_t - \delta_t^{\mathrm{inv}} - \delta_t^{\mathrm{imb}} + \delta_t^{\mathrm{trend}}$$
$$I_t = I_{t-1} + \ell_t\, 1_t^{\mathrm{bid}} - \ell_t\, 1_t^{\mathrm{ask}}$$

If a public trade "crosses" our bid or ask, we lock in that execution: inventory changes by exactly our lot size

Exotic Options Pricing

Keywords : Monte Carlo Pricing, European option, Black-Scholes model

Problem Formulation

Price a set of **36** knock-out European basket options on an equally weighted portfolio of three stocks **(DTC, DFC, DEC)** under a correlated risk-neutral model with **Monte-Carlo Simulation**.

Given , Spot Price of **100**, risk-free rate $r = 5\%$, **15** vanilla call prices for **5** strikes and **3** maturities.

The basket is

$$B_t = \frac{1}{3}(S_t^{\mathrm{DTC}} + S_t^{\mathrm{DFC}} + S_t^{\mathrm{DEC}})$$

Assumptions

- Exercisable only at maturity T.
- Each Stock follows the relation

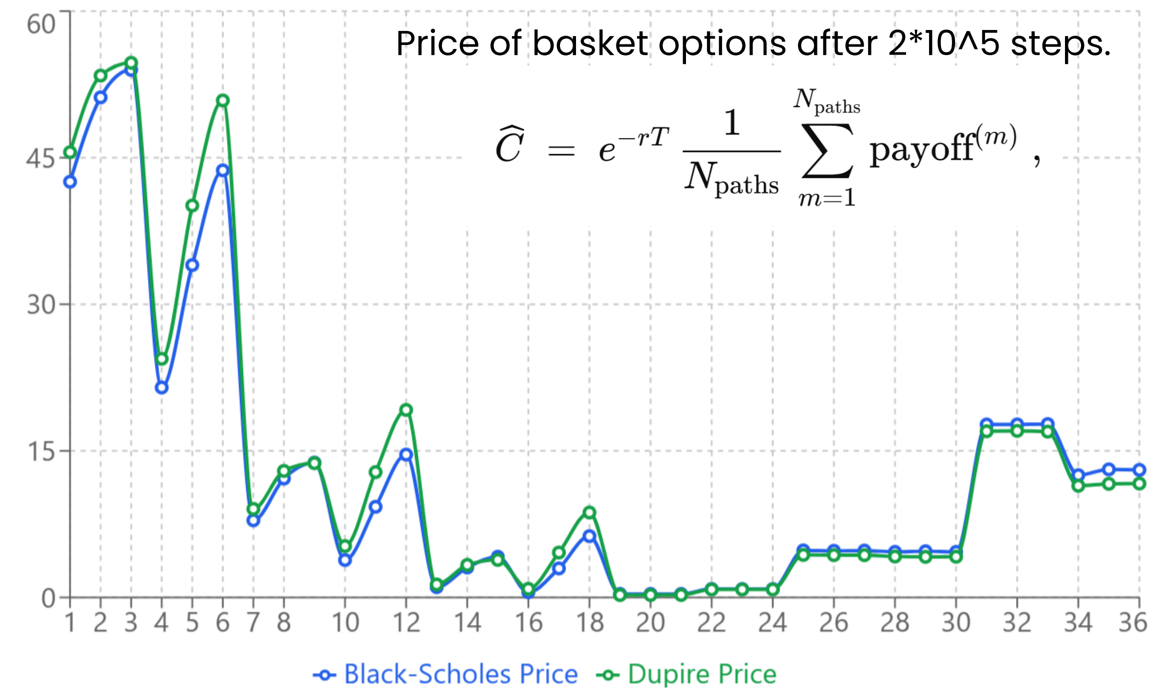
$$dS_t^X = r\, S_t^X\, dt + \sigma^X(S_t^X, t)\, S_t^X\, dW_t^X$$

- All assets are non-dividend-paying.
- Correlation between Stocks

$$\rho = \begin{bmatrix} 1.00 & 0.75 & 0.50 \\ 0.75 & 1.00 & 0.25 \\ 0.50 & 0.25 & 1.00 \end{bmatrix}$$

- Volatility depends on Spot and Time $\sigma = \sigma(S_t, t)$
- We can calibrate volatility with the given 15 option prices.
- Large number of paths, fine time discretization

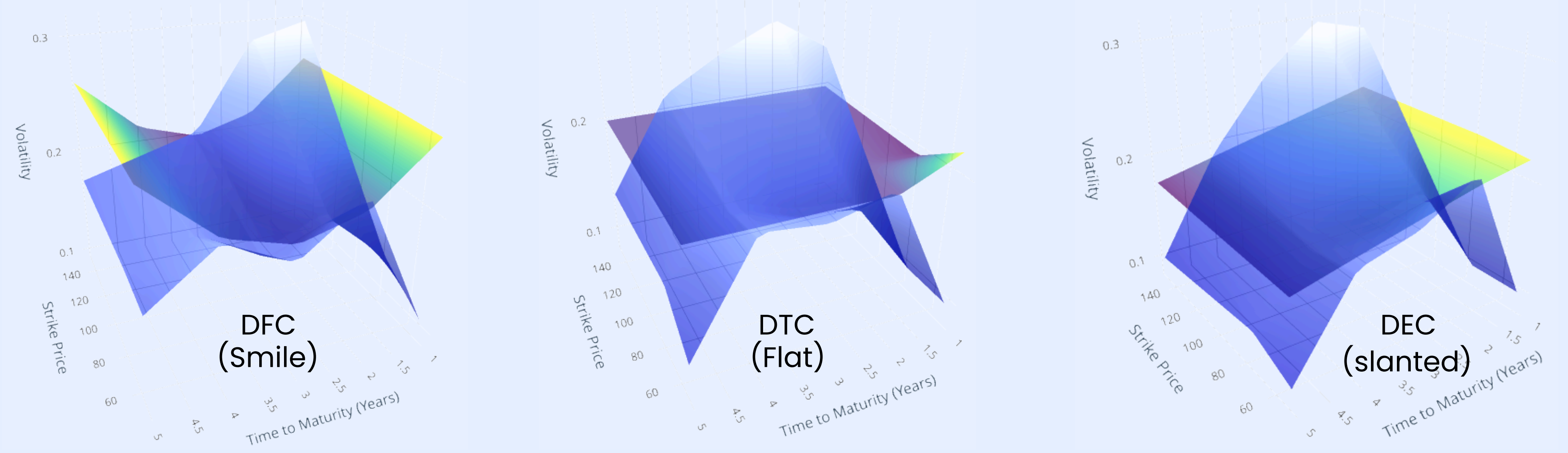
- Black–Scholes gives better results as it uses directly inferred ATM volatilities without requiring derivatives of the entire price grid.
- Dupire's Formula require a smooth surface of spot and maturity.



Calibration
Local Volatility Surfaces

1. From market price, invert Black–Scholes to get the Volatility.
2. Using Dupire's formula calculate Volatility as a function of spot and time.

- 1st method generates smoother, more stable Surface unlike 2nd - considering the sparse data (5*3 options), which makes calculating derivatives difficult.



95% accuracy in pricing the Vanilla Basket options with knock-out using **Black-Scholes model**.

Local Volatility Surface plays important role in pricing **accurately**.

With a bigger real life dataset we could use **Dupire's Formula.**

Quant

Thank You