Learning to Trade II
Deep Hedging

TU Munich
Visiting Prof. Hans Buehler
Prof. Blanka Horvath

TU München 2022

http://deep-hedging.com

Electronic copy available at: https://ssrn.com/abstract=4151041

Lesson 2.1: Vanilla Deep Hedging

AI for Derivatives Trading

Electronic copy available at: https://ssrn.com/abstract=4151041

Recap

- Notation
  - Everything is in discrete time → markets not complete.
  - We call s_t the **state** of the market. It represents all known information.
  - That means that any observable random variable R_t can be written as $R(s_t)$.
  - At any point t we observe the model prices $H_t^i = H^i\ (s_t)$ of n hedging instruments.
  - The (sum of the) model prices of our existing portfolio is $Z_t \equiv Z(s_t)$.
  - The market is observed under the *statistical* measure P.

- Trading cost
  - Trading $a \in \mathrm{R}^n$ units of H at t will **cost** $c_t(a)$ in excess of the model price H_t.
  - The function c is normalized to $c_t(0) = 0$, non-negative, and convex.
  - Convexity excludes fixed trading cost.
  - Trading is limited to where $c < \infty$.

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging [1]

- With Statistical Hedging, we looked at the return $dZ_t = Z_{t+dt} - Z_t$.
- Now do the other extreme:
  - Assume T is such that all instruments in our portfolio Z and any relevant hedging instruments are expired.
  - Valuation of Z_T and H_T is trivial, and *model-independent*: it is the sum of all cashflows until T.
  - Instead of a single action a, we will now solve for an **policy** a which is a function of the state, i.e. at any time t the hedging action is $a_t \equiv a(s_t)$.
- Then

$$G^a := Z_T - Z_t + \sum_{r=t}^{T-1} a_r \cdot (H_T - H_r) - c_r(a_r)$$

[1] Deep Hedging, Buehler et all 2018, https://arxiv.org/pdf/1802.03042.pdf

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- Consider

$$G^a := Z_T - Z_t + \sum_{r=t}^{T-1} a_r \cdot (H_T - H_r) - c_r(a_r)$$

- We note that we will need the model prices H_r of all our trading instruments along the path.
- We do not need the model price of our portfolio Z as its payoff is known in T

(*) We drop Z_t as it is a constant and will disappear from any optimization.

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- Consider

$$G^a := Z_T - Z_t + \sum_{r=t}^{T-1} a_r \cdot (H_T - H_r) - c_r(a_r)$$

- Let $a \star H_T := \sum_{r=t}^{T-1} a_r \cdot (H_T - H_r)$ and $C_T(a) := \sum_{r=t}^{T-1} c_r(a_r)$. Then above becomes essentially*

$$G^a := Z_T + a \star H_T - C_T(a)$$

(*) We drop Z_t as it is a constant and will disappear from any optimization.

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- Let U be a **monetary utility**. Then the **Vanilla Deep Hedging** problem is given as

$$\max_{a}: U(\, G^{a}\,) \quad \text{with} \quad G^{a} \coloneqq Z_T + a \star H_T - C_T(a)$$

- The main difference to our previous formulation is that a is now a *function* which takes the current state and returns the next hedge.
- Natural machine learning approach: write a as **neural network** with network weights θ i.e.

$$a_t \equiv a(s_t) \coloneqq a(\theta; s_t)$$

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- Neural network with weights $\theta \in R^K$ is a function approximator $F(\theta;\cdot)$ for a function, say, $f: R^{n_0} \to R^{n_K}$.
  - Let $x_0 \in R^{n_0}$.
  - K is called **depth.**
  - Define **widths** $n_1, \dots, n_{K-1}$.
  - Define **weights** $w_k \in R^{n_{k+1} \times n_k}$ and **biases** $b_k \in R^{n_{k+1}}$.
    The totality is $\theta = (w_0, \dots, w_{K-1}; b_0, \dots, b_{K-1})$.
  - Let $\sigma: R \to R$ be a suitable function, called **activation function**
  - Define a neural network by iterative application of σ to linear combinations of the nodes x:

$$x_{k+1}^i := \sigma\left(w_k^i \cdot x_k + b_k^i\right)$$

- Point is that the approximator $F(\theta;\cdot) \approx f(\cdot)$ can be arbitrarily close for a wide range of functions f ("universal approximation theorem").

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- AI Deep Hedging problem

$$\max_{\theta}: U\left(Z_T + a(\theta) \star H_T - C_T\big(a(\theta)\big)\right)$$

- Actions a are neural networks with weights θ

$$a_t \equiv a(s_t) \coloneqq a(\theta; s_t)$$

[1] Deep Hedging, Buehler et all 2018, https://arxiv.org/pdf/1802.03042.pdf

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- AI Deep Hedging problem

$$\max_{\theta}: U\left(Z_T + a(\theta) \star H_T - C_T\big(a(\theta)\big)\right)$$

- Numerically this is a Monte Carlo problem if we use K paths for an OCE $U(X) = \sup_{y}: E[u(X+y) - y]$:

$$\max_{\theta,y}: \frac{1}{K}\sum_{\omega=1}^{K} u\big(Z_T + a(\theta) \star H_T - C_T\big(a(\theta)\big) + y\big) - y$$

- In Reinforcement Learning this is called “Periodic Policy Search”

[1] Deep Hedging, Buehler et all 2018, https://arxiv.org/pdf/1802.03042.pdf

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- AI Deep Hedging problem

$$\max_{\theta}: U\left(Z_T + a(\theta) \star H_T - C_T\big(a(\theta)\big)\right)$$

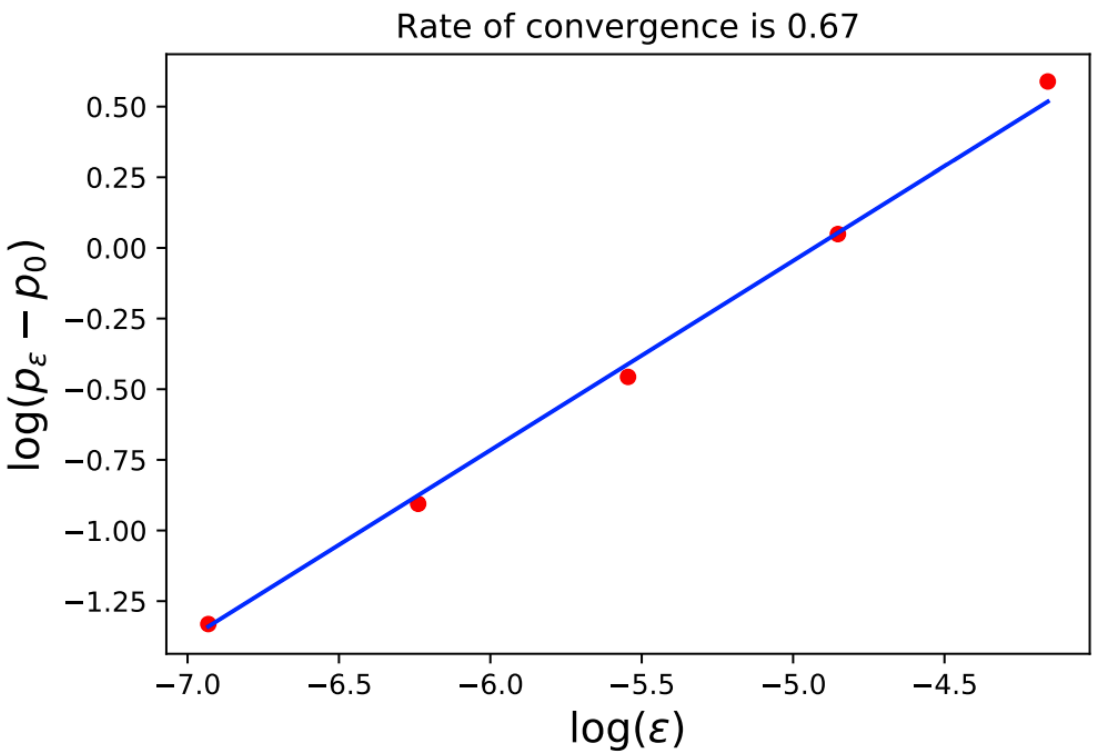
- We have proven in [1] that this scheme theoretically converges to the right optimal policy a^*.
- Requires that the state s is indeed exhaustive and contains all relevant information → "feature engineering"
  - Too big a state: difficult to converge.
  - Too small a state: won't converge to optimum

[1] Deep Hedging, Buehler et all 2018, https://arxiv.org/pdf/1802.03042.pdf

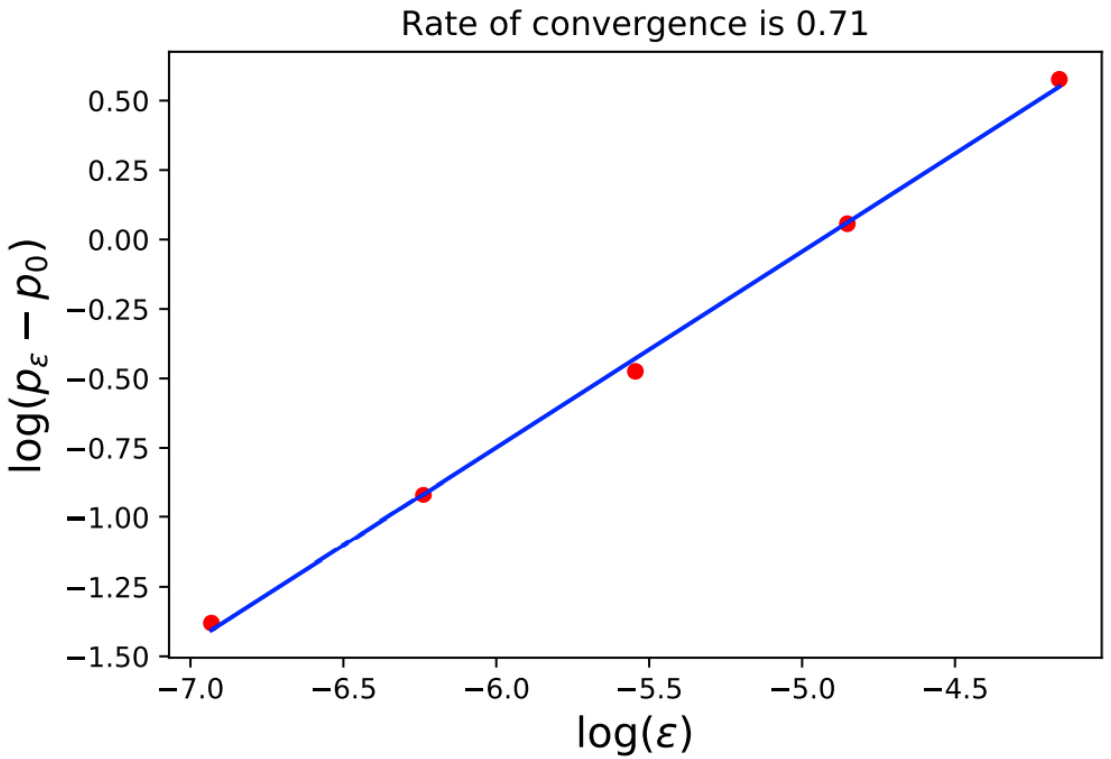
Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- Test first vs cases where we know or guess the theoretical answer [1]



Black-Scholes model price asymptotics.



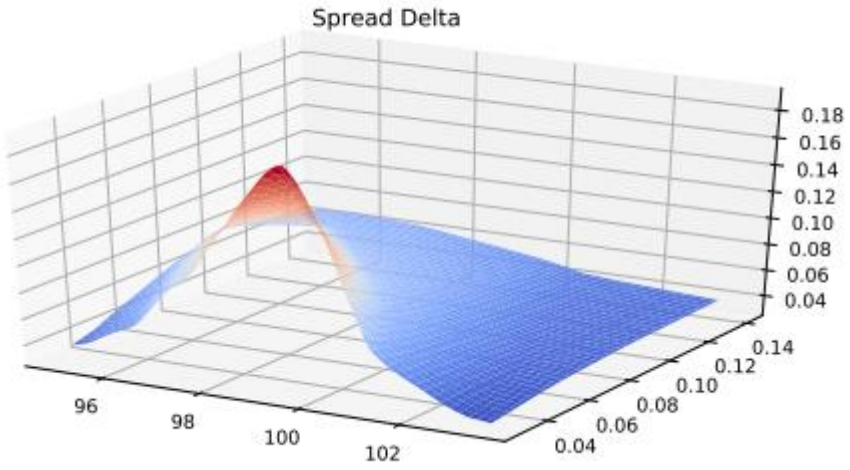
Heston model price asymptotics

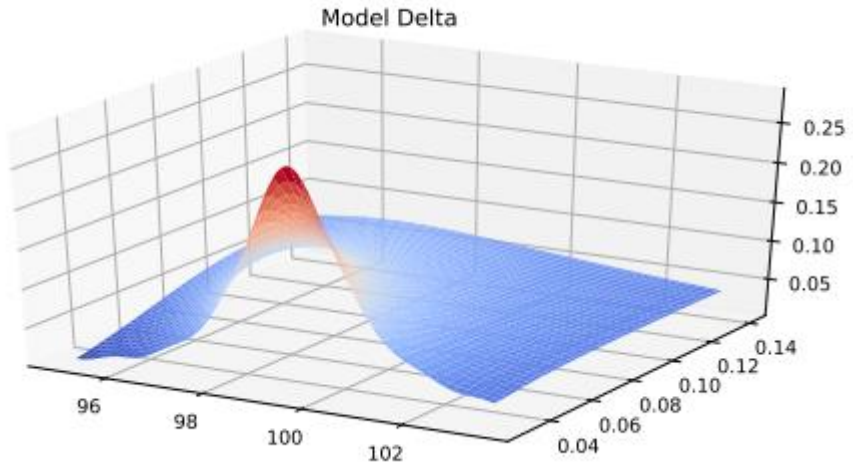
[1] Deep Hedging, Buehler et all 2018, https://arxiv.org/pdf/1802.03042.pdf

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- Delta of a call spread in Black & Scholes [1]





[1] Deep Hedging, Buehler et all 2018, https://arxiv.org/pdf/1802.03042.pdf

Electronic copy available at: https://ssrn.com/abstract=4151041

Deep Hedging

- **Marginal Pricing**: we wish to sell a derivative D to our client
- Define

$$U^*(Z) \coloneqq \max_a : U\left(Z_T + a \star H_T - C_T(a)\right)$$

- The marginal price of selling D is given as

$$\Pi(D) \coloneqq U^*(Z) - U^*(Z - D)$$

- It represents the *minimal price* we should charge to not be worse off with respect to U.

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- So it works … some git code to play around https://github.com/hansbuehler/deephedging
- Problem solved? … not at all !
- No point solving Deep Hedging for classic diffusion dynamics…

Electronic copy available at: https://ssrn.com/abstract=4151041

Lesson 2.2: Market Simulation

Managing Uncertainty

Electronic copy available at: https://ssrn.com/abstract=4151041

Market Simulation

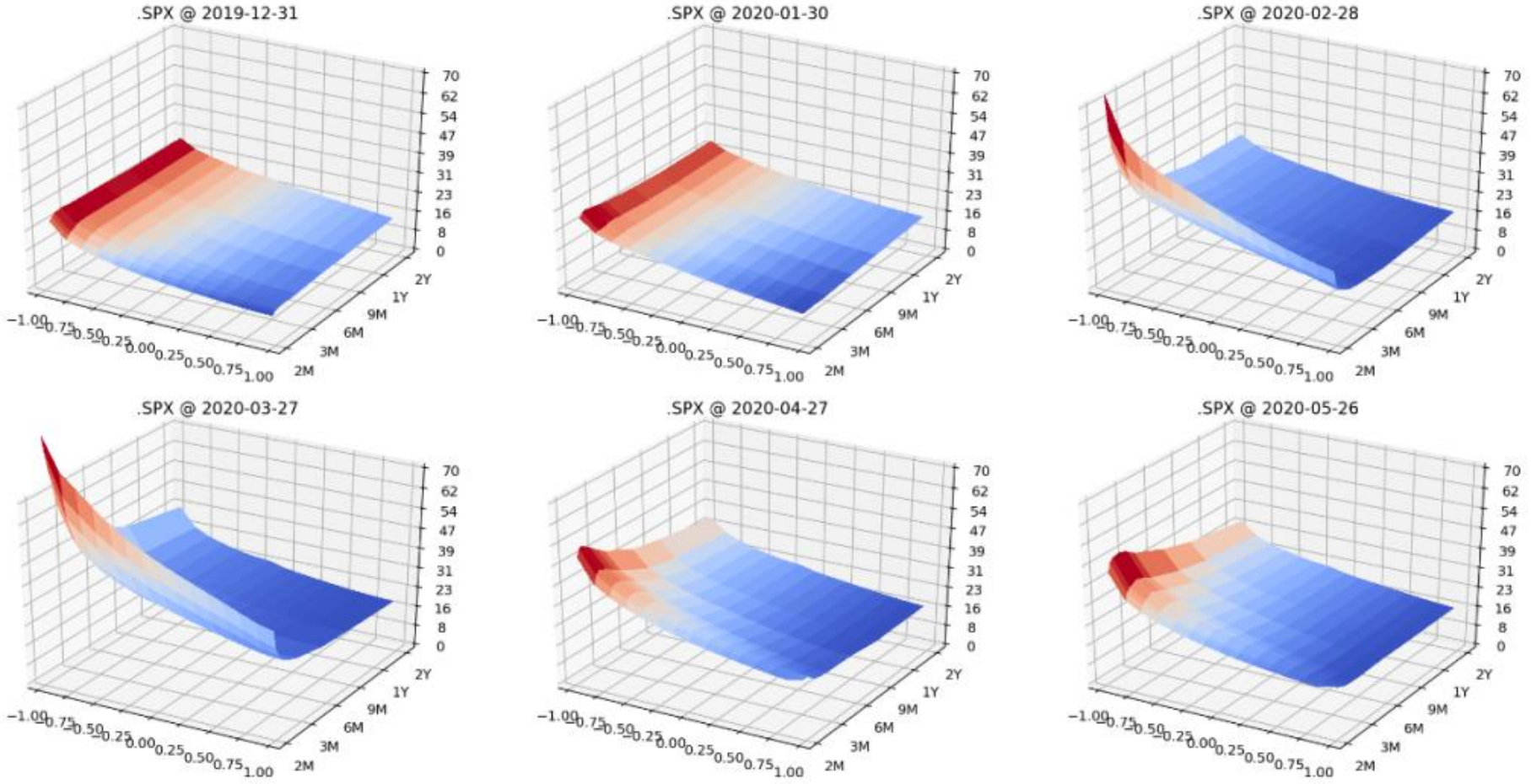
- We are given a history of historic market data
- Aim is to write a simulator for the “entire market”

- Lack of data: for example index spot and option prices.
  - 10Y of data ~ 2500 data points
  - Typically index option surface has 100’s of options

- Arbitrage
  - Need to find a parametrization which ensures absence of static arbitrage
  - Static arbitrage = positive returns for free → Deep Hedging will exploit it emphatically

Electronic copy available at: https://ssrn.com/abstract=4151041

Market Simulation



Data from Learning Risk-Neutral Implied Volatility Dynamics, Buehler et al 2021, https://arxiv.org/pdf/2103.11948.pdf

Electronic copy available at: https://ssrn.com/abstract=4151041

Market Simulation

Market Simulation for Implied Volatilities

- Arbitrage: parameterize grid of option prices $C\left(T_j, K_i\right)$in a space in which we can easily enforce absence of static arbitrage
- Discrete Local Volatility "DLV" [1] is a numerically robust form of local volatility.

$$\sigma_{j,i}^2 := \frac{C\left(T_{j+1}, K_{i+1}\right) - C(T, K_i)}{2\left\{C\left(T_j, K_{i+1}\right) - 2C\left(T_j, K_i\right) + C\left(T_j, K_{i-1}\right)\right\}K_i^2\left(T_{j+1} - T_j\right)}$$

- Bijection between prices C and DLV's σ.
- Positivity of σ is sufficient to define an arbitrage-free option surface

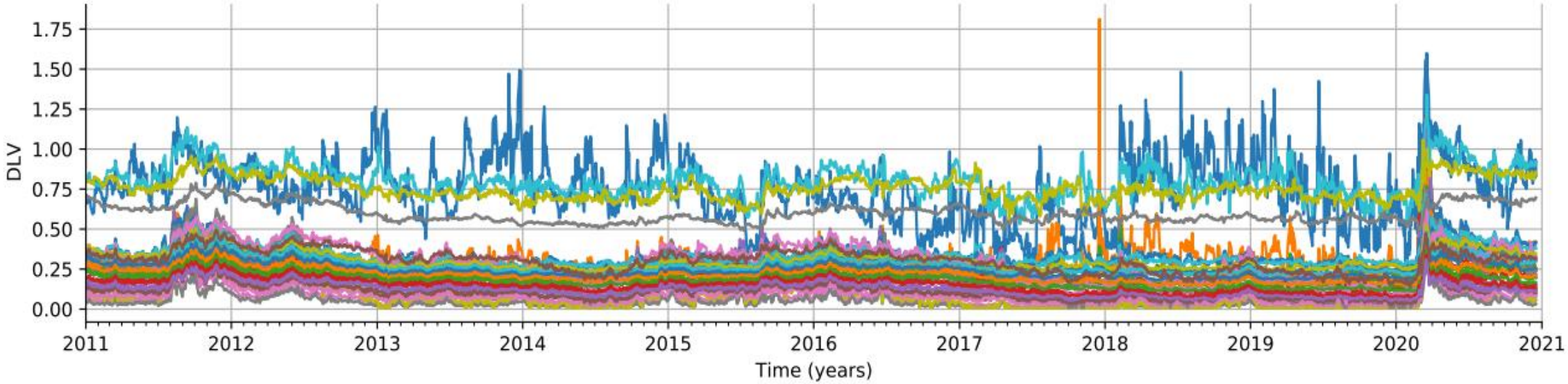
[1] Discrete Local Volatility for Large Time Steps (Extended Version), Buehler et al 2015, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2642630

Electronic copy available at: https://ssrn.com/abstract=4151041

Market Simulation

Market Simulation for Implied Volatilities

Spot and Option Prices at t ⇨ Spot and DLVs at t



[1] graph from Multi-Asset Spot and Option Market Simulation, Wiese et al 2021, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3980817

Electronic copy available at: https://ssrn.com/abstract=4151041

Market Simulation

Market Simulation for Implied Volatilities

- Dimensionality of the vector $Y_t = (\log S_t\,, \log \sigma_{j,i})$ same as spot and options
- However, option prices are managed by humans and simple (non-ML) machines.
- Working assumption the change vector dY_t can be presented by a much lower dimensional state vector ds_t.

Electronic copy available at: https://ssrn.com/abstract=4151041

Market Simulation

Market Simulation for Implied Volatilities

- Classic method: PCA [1] i.e. unconditional linear representation of the form

$$dY_t = Bdt + Ads_t$$

- Pure regression allows us matching mean and variance of dY_t.

[1] Deep hedging: learning to remove the drift, Buehler et al 2022 https://www.risk.net/cutting-edge/banking/7932226/deep-hedging-learning-to-remove-the-drift and https://arxiv.org/abs/2111.07844

Electronic copy available at: https://ssrn.com/abstract=4151041

Market Simulation

Market Simulation for Implied Volatilities

- Autoencoder [1]: learn a machine representation
  - Decoder: non-linear form of PCA: learn weights ξ

$$Y_t = A_d(\xi; s_t)$$

  - Encoder: lean weights η to reconstruct the relevant state space from the full surface Y

$$s_t = A_e(\eta; Y_t)$$

[1] Deep Hedging: Learning to Simulate Equity Option Markets, Wiese et al 2020 https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3470756

Electronic copy available at: https://ssrn.com/abstract=4151041

Market Simulation

Market Simulation for Implied Volatilities

- Autoencoder: learn a machine representation

$$Y_t = A_d(\xi; s_t) \quad s_t = A_e(\eta; Y_t)$$

- Plenty of freedom to build a "realistic" process for ds_t.
- What means realistic
- For example
  - Classic covariance, correlation, auto-correlation …
  - Matching the signature representation → Wasserstein distance [1]
  - … ?

[1] Conditional Sig-Wasserstein GANs for Time Series Generation, Ni et al 2020, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3623086

Electronic copy available at: https://ssrn.com/abstract=4151041

Market Simulation

Market Simulation for Implied Volatilities

- Working assumption the change vector dY_t can be presented by a much lower dimensional state vector ds_t.
  - Variational Autoencoder [1]: learn a machine representation of a general distribution

$$Y_t = A_d(\xi; s_t, dW_t) \quad \text{with} \quad dW_t \sim N^m(\sqrt{dt})$$

  - VAE can match many features of the distribution of dY_t.
  - What are the relevant characteristics … ?
  - Engineer features which we assume are relevant for s_t

[1] Deep Hedging: Learning to Simulate Equity Option Markets, Wiese et al 2020 https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3470756

Electronic copy available at: https://ssrn.com/abstract=4151041

Deep Hedging & Market Simulation

- Used in JP Morgan for Cliquet Options
  https://www.risk.net/derivatives/equity-derivatives/7921526/jp-morgan-testing-deep-hedging-of-exotics

- Good
  - Promising, versatile, intuitive machinery
  - Model-free in the sense that the machine is easily transferrable
  - Takes into account market frictions
  - Results deviate from classic models

- But
  - Very heavy numerically
  - Results deviate from classic models and need explanation
  - Vanilla Deep Hedging requires re-training every time Z changes → "Bellman" Deep Hedging: joint research and patent pending. Not a trivial topic at all.

Electronic copy available at: https://ssrn.com/abstract=4151041

Vanilla Deep Hedging

- Data-Driven Risk Management for Derivatives – a new topic in quant finance
  - Data challenges
    - Not that much relevant data, in particular when expanding into multi-asset simulation across asset classes
    - Data hard to access outside finance making validation of papers hard
    - Trading cost beyond simple bid/ask spread often hard to obtain
  - Modelling challenges
    - Adhere to static no-arbitrage conditions, e.g. butterfly spreads cannot have negative prices.
    - Consistent interpolation onto a large number of hedging instruments
  - Statistical challenges
    - Create realistic data – where “realistic” is defined by “it works in hedging”.
    - Non-stationarity of financial data
  - Machine Learning challenges
    - Non-linear data generators beyond diffusion-like approaches
    - Managing explosions / interpolation / extrapolation
    - Robustness against outliers in data, including outliers which are simply there (e.g. March 2019)

Electronic copy available at: https://ssrn.com/abstract=4151041

Lesson 2.3: Statistical Arbitrage

Managing Uncertainty

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Back to Deep Hedging … on our market simulator
- We have earlier defined

$$U^*(Z) \coloneqq \max_a : U\left(Z_T + a \star H_T - C_T(a)\right)$$

- What about $U^*(0)$ which represents the "value" of an initially empty portfolio?
- We say that the market exhibits **statistical arbitrage** if $U^*(0) > 0$.
- Happens naturally.

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Statistical Arbitrage is an economic fact
  - Selling insurance commands a risk premium
    - Selling puts
    - Selling long-term rates vs. short term rates
    - Selling credit protection
    - …
  - Arbitrageurs exploit opportunities to make the market more efficient
    - Hedge funds' economic role is to find such opportunities and drive markets towards efficiency
    - Some use forms of crowd sourcing to add signals
    - Entire (well paid) industry

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Indeed, naïve application to Deep Hedging to the S&P500 with data for the last 5 years leads to
  - Going long the index
  - Selling puts
  - Ignore hedging any derivative

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

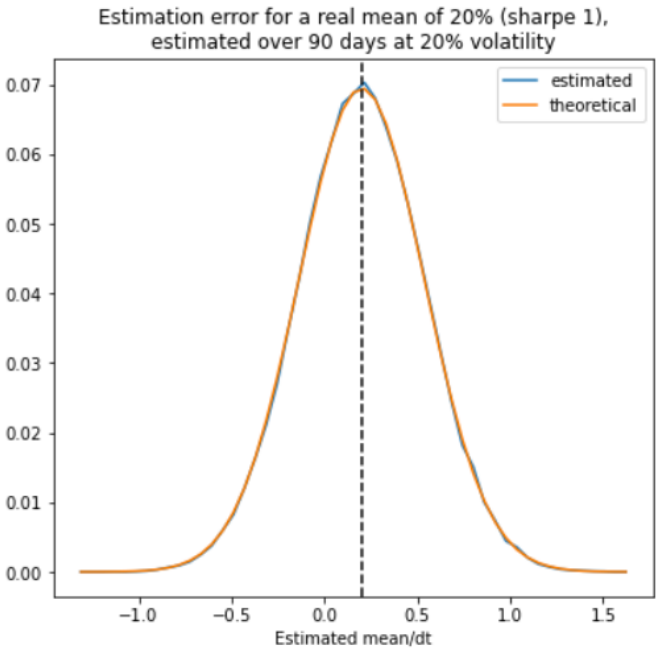
- Statistical Arbitrage is also result of *Model Uncertainty*
  - Assume we observe a market generated by a normal asset price X
    - Annualized volatility 20%
    - Annualized drift 20% i.e. sharpe is 1
  - In practise, we observe *one path.* On that path we estimate the mean of the process.

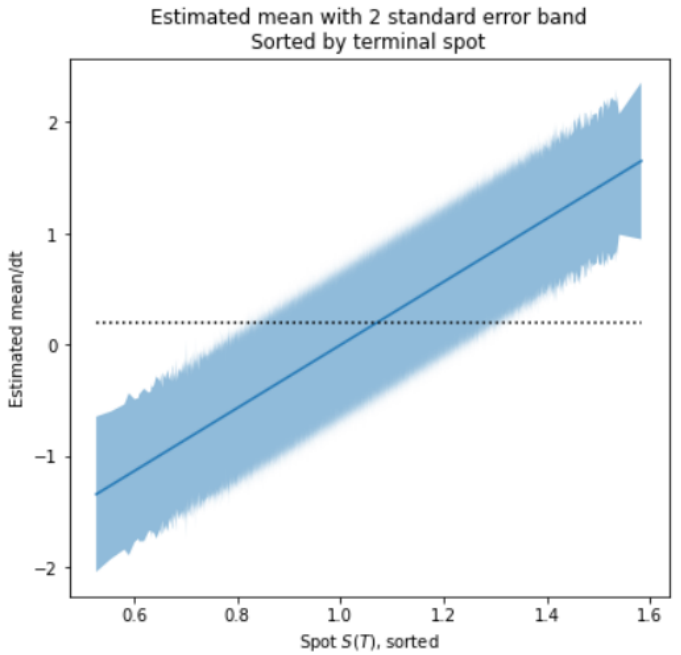
$$\text{mean}(dX) = \frac{1}{m}\sum_{t=1}^{m} dX_t = \frac{1}{m}(X_m - X_0)$$

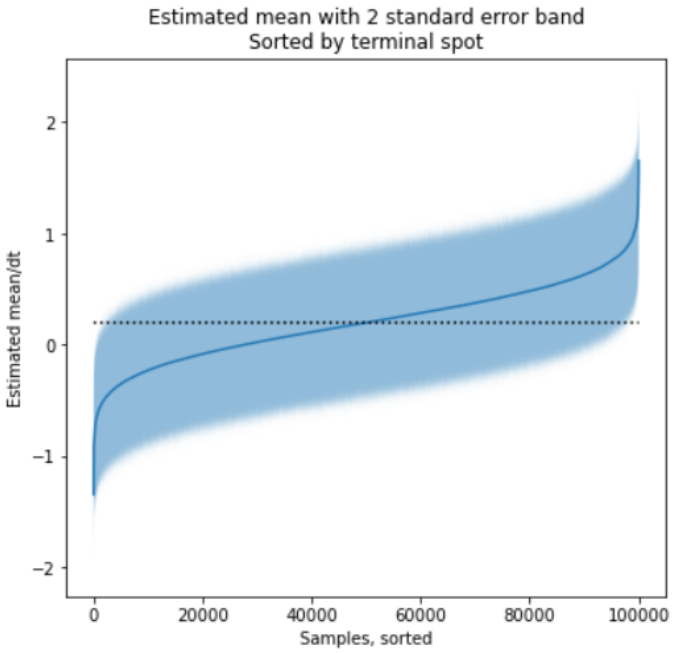
    - The estimator itself is a random variable
    - How does this look like statistically?

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage







The naïve mean estimator is *uncertain* as it depends on the path we are on.
Effect is called *Knightian Uncertainty*

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Why does the *Knightian Uncertainty* of the mean matter?

$$U(X) = E_Q[X] + \underbrace{U\left(X - E_Q[X]\right)}_{Risk\ term}$$

- Finding an optimal hedging strategy is very sensitive to the mean.
- Machine *thinks* it can make money … but it doesn't work because of the estimation error.

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Knightian Uncertainty: classic problem in systematic investment strategies – plenty of heuristics to address the issue even for classic "Markoviz" portfolio construction [1]
- *This is a lot less difficult for estimators of higher moments. That also makes theoretical sense – the arbitrage is in the mean !*

- Two main ideas:
  - Remove the Drift → published work with Imperial College London
    This approach is robust and does not attempt to construct a drift.
  - Managing Knightian Uncertainty → research project with TU Munich
    Use the drift, but appreciate that it is subject to estimation noise

[1] good summary: Marcos Lopez de Prado: Advances in Financial Machine Learning, Wiley, 2018

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

Removing the Drift

- Objective is first remove the sampling drift, and then in a second step set it to a judiciously researched target drift with higher confidence.

- In classic portfolio optimization we only have “linear” assets
- To remove the drift, we simply divide each return by the mean return over the sample period

$$d\tilde{X}_t^i := \frac{dX_t^i}{\frac{1}{m}(X_m - X_0)}$$

(*) strictly speaking we will set the drift to the prevailing overnight interest rate

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

Removing the Drift

- Objective is first remove the sampling drift, and then in a second step set it to a judiciously researched target drift with higher confidence.

- For markets with complex assets removing the drift distorts the co-dependence of the instruments, e.g. stock and options thereon.
- Instead of changing the paths we aim now to *reweight* the observed paths such that the drift disappears – that means constructing a new equivalent measure Q.

[1] Marcos Lopez de Prado: Advances in Financial Machine Learning, Wiley, 2018

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Assume using the entropy $u(x) \coloneqq \frac{1-e^{-\lambda x}}{\lambda}$
- Cost zero for illustration.
- Under a measure Q define

$$U_Q(X) \coloneqq \sup_y : E_Q[u(X+y)] - y = -\frac{1}{\lambda}\log E_Q\left[e^{-\lambda X}\right]$$

- We wish to chose $Q \approx P$ such that

$$0 = \max_a : U_Q(\, a \star H_T\,)$$

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Step 1: under P find the optimal strategy a^0 for the empty portfolio

$$a^0 := \operatorname{argmax}_a : U_P(\, a \star H_T \,) = \operatorname{argmax}_a : \frac{1}{\lambda} \log E_P\left[e^{-\lambda\{a \star H_T\}}\right]$$

- Step 2: define the measure

$$dQ := \frac{e^{-\lambda\{a^0 \star H_T\}}}{E_P\left[e^{-\lambda\{a^0 \star H_T\}}\right]} dP$$

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

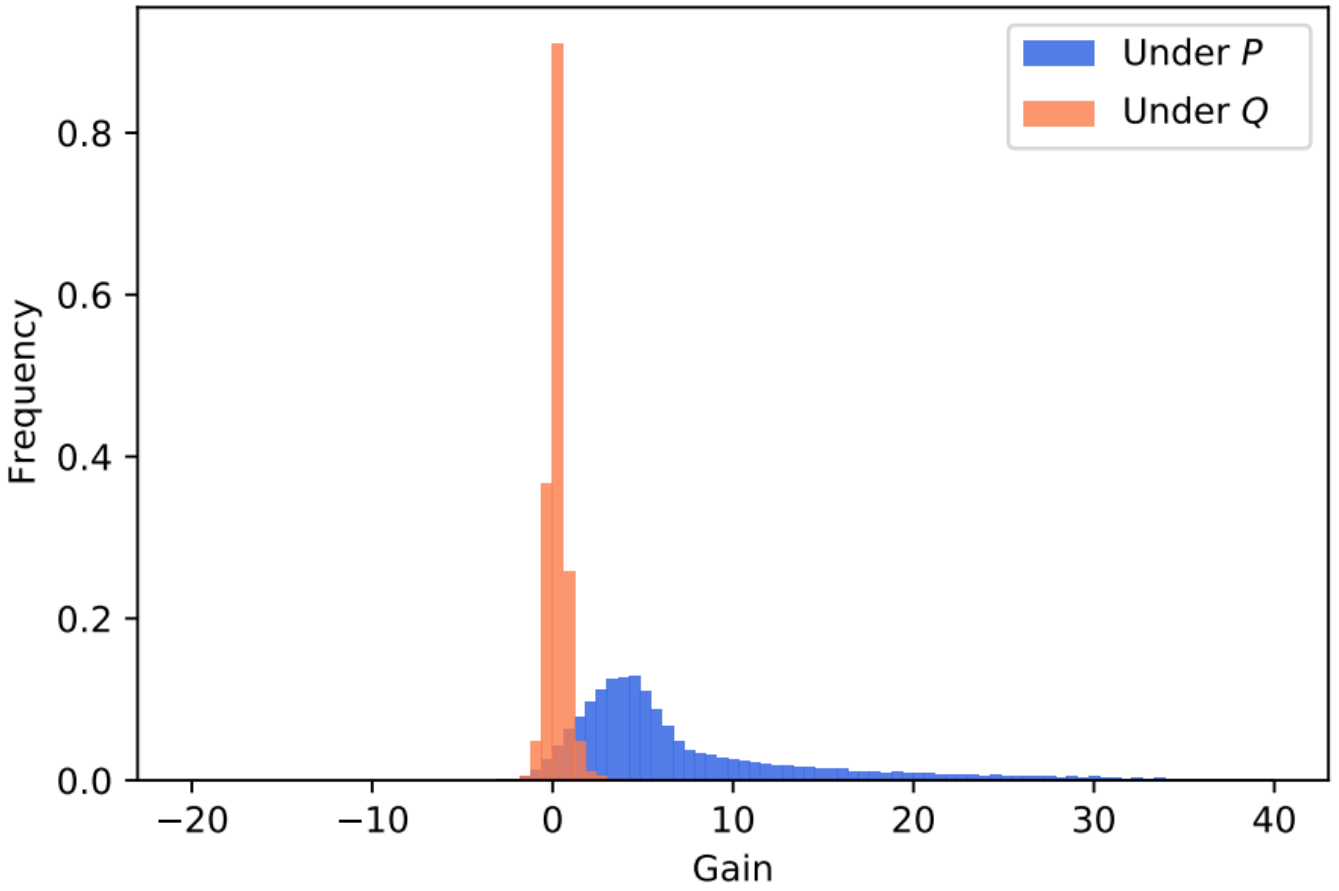
- Step 3: under the monetary utility U_Q:

$$\begin{aligned}
&\max_a: U_Q(\, a \star H_T \,) \\
&= \max_a: \frac{1}{\lambda} \log E_P \left[\frac{e^{-\lambda\{(a+a^0)\star H_T\}}}{E_P\left[e^{-\lambda\{a^0 \star H_T\}}\right]} \right] \\
&\sim \max_a: \frac{1}{\lambda} \log E_P \left[e^{-\lambda\{(a+a^0)\star H_T\}} \right] \\
&= 0
\end{aligned}$$

  - Because a_0 was optimally chosen under P.
- This shows that the optimal a is zero – in other words, the monetary utility U_Q is free of statistical arbitrage

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage



Numerical results of reducing the drift

[1] Learning Risk-Neutral Implied Volatility Dynamics, Buehler et al 2021, https://arxiv.org/pdf/2103.11948.pdf

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- The measure Q is one of many equivalent martingale measures.
- Our specific choice minimizes the entropy of Q with respect to P among all equivalent martingale measures [1] i.e.

$$Q \mapsto E_P\left[\frac{dQ}{dP}\log\frac{dQ}{dP}\right]$$

[1] Deep hedging: learning to remove the drift, Buehler et al 2022 https://www.risk.net/cutting-edge/banking/7932226/deep-hedging-learning-to-remove-the-drift and https://arxiv.org/abs/2111.07844

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- In the case of the entropy, without transaction cost we have the following intuitive result [1]:
- For a portfolio Z the optimal strategy a^* under U_P is given as

$$a^* := a^0 + \tilde{a}$$

- Where:
  - a^0 is the optimal "prop trading" strategy for the empty portfolio under P.
  - $\tilde{a}$ is the optimal "pure hedging" strategy under the risk-neutral Q.

[1] Deep hedging: learning to remove the drift, Buehler et al 2022 https://www.risk.net/cutting-edge/banking/7932226/deep-hedging-learning-to-remove-the-drift and https://arxiv.org/abs/2111.07844

Electronic copy available at: https://ssrn.com/abstract=4151041

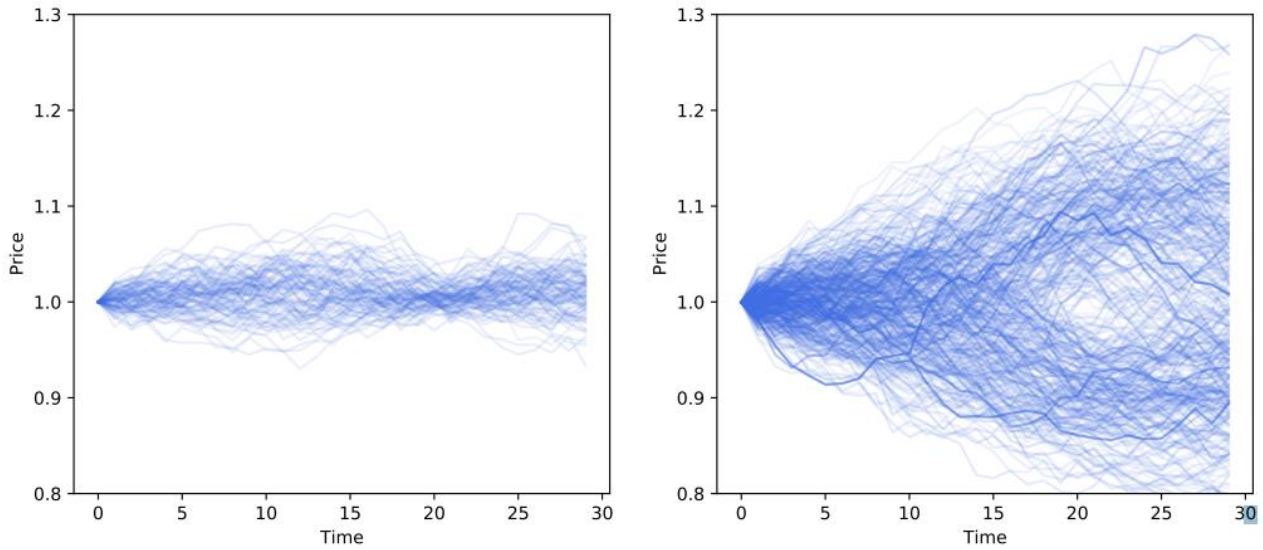
Statistical Arbitrage

- Fun fact: in discrete time, we can change also the *volatility* of a process by changing measure.
  - Experiment: market with 15% annual realized volatility. Option traded with 20% volatility. Statistical arbitrage is selling the option and delta hedge.
  - What happens when we change our measure:
    The measure will put more weight on paths with lower realized (discrete time) variance per path

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Experiment [1]: market with 15% annual realized volatility. Option traded with 20% volatility. Statistical arbitrage is selling the option and delta hedge.
- The measure will put more weight on paths with lower realized (discrete time) variance per path



Left: paths given the highest 0.1% of probabilities under Q; right: lowest 0.1%

[1]Learning Risk-Neutral Implied Volatility Dynamics, Buehler et al 2021, https://arxiv.org/pdf/2103.11948.pdf

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

Generalization to cost and arbitrary u

- Define "marginal cost" m_t as linearization of c in zero as follows [1]:
  - Denote by $\nabla^{\pm} c_t(0)$ the gradients in zero in positive and negative direction;
  - Let $a^{\pm} := \max\{\pm a, 0\}$;
  - Define

$$m_t(a_t) := \sum_i a_t^+ \cdot \nabla^+ c_t(0) - a_t^- \cdot \nabla^-(0)$$

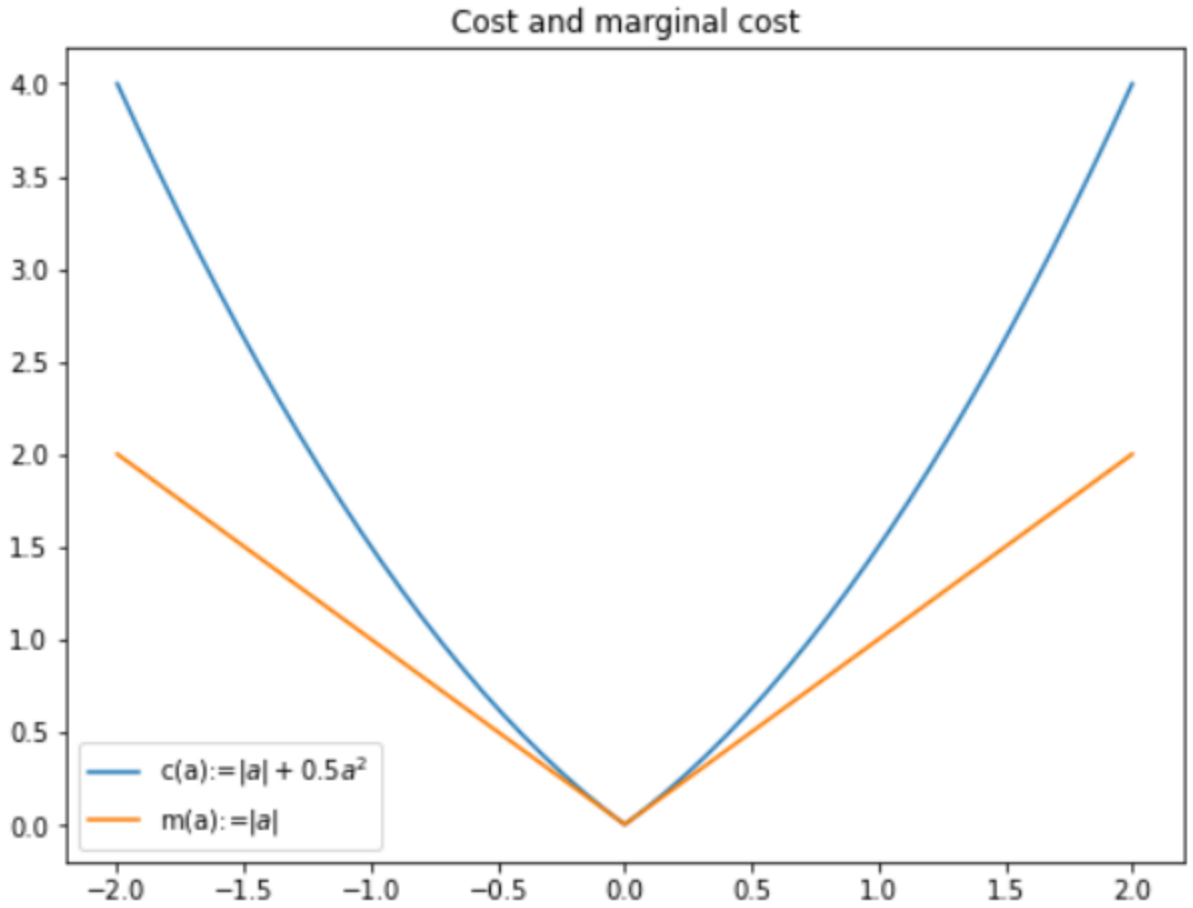
and

$$M_T(a) := \sum_t m_t(a_t)$$

[1] Deep hedging: learning to remove the drift, Buehler et al 2022 https://www.risk.net/cutting-edge/banking/7932226/deep-hedging-learning-to-remove-the-drift and https://arxiv.org/abs/2111.07844

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage



Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Apply similar idea to before [1]: find a^0, y^0 as solution to

$$\max_{a,y}: E_P[\, u(\, a \star H_T - M_T(a) + y^0) - y^0 \,]$$

- Define the measure Q via

$$dQ := u'(\, a^0 \star H_T - M_T(a^0) + y^0 \,)\, dP$$

[1] Deep hedging: learning to remove the drift, Buehler et al 2022 https://www.risk.net/cutting-edge/banking/7932226/deep-hedging-learning-to-remove-the-drift and https://arxiv.org/abs/2111.07844

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Under the new measure the expected returns of all instruments are within marginal bid/ask spread in the following sense:
  - Let $\gamma_t^{i\pm} := \nabla^{\pm}(e^i)$ be the marginal cost of buying/selling a small quantity of the ith asset. Then [1] the measure Q is a *near-martingale measure* in the sense that

$$H_t^i - \gamma_t^{i-} \leq E_Q\left[H_T^i | s_t\right] \leq H_t^i - \gamma_t^{i+}$$

  - There is no statistical arbitrage under Q with full (or marginal) transaction cost:

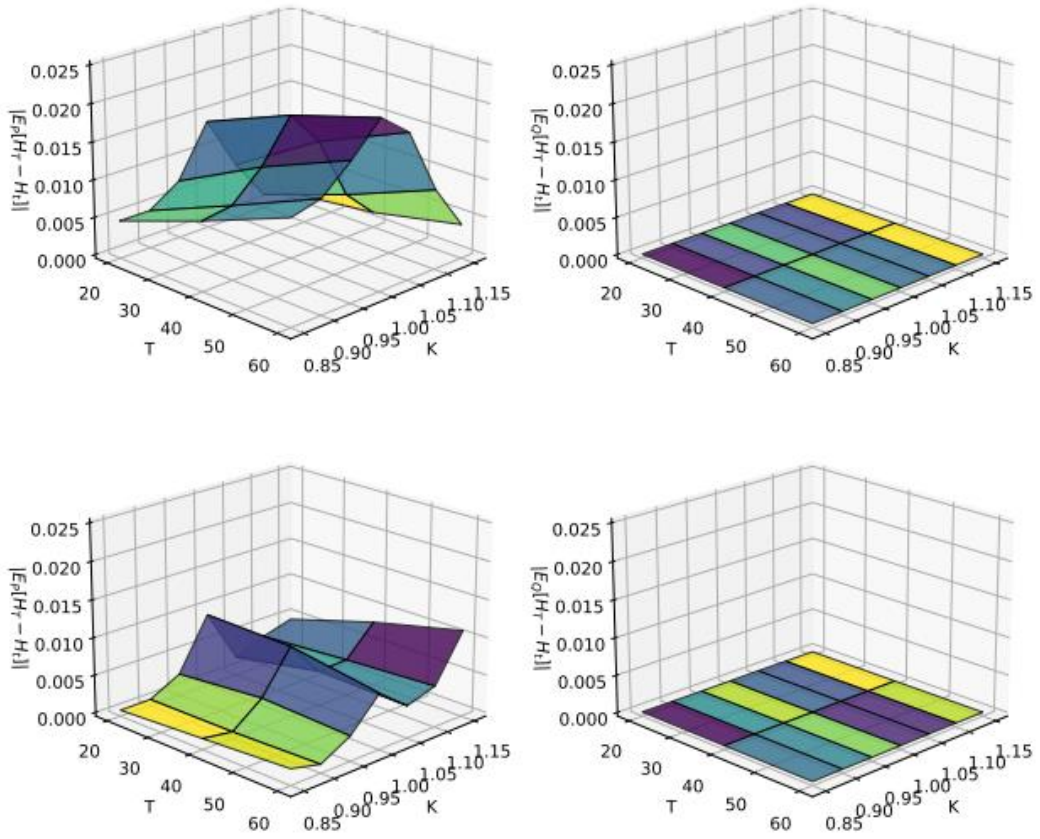
$$0 = \max_a : U_Q(\, a \star H_T - C_T(a)\,)$$

  - The measure Q minimizes the $\tilde{u}$-divergence to P among all near-martingale measures [1].

[1] Deep hedging: learning to remove the drift, Buehler et al 2022 https://www.risk.net/cutting-edge/banking/7932226/deep-hedging-learning-to-remove-the-drift and https://arxiv.org/abs/2111.07844

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage



Full market simulation results [1]: left are expected returns under P, right under Q under transaction cost

[1] Deep hedging: learning to remove the drift, Buehler et al 2022 https://www.risk.net/cutting-edge/banking/7932226/deep-hedging-learning-to-remove-the-drift and https://arxiv.org/abs/2111.07844

Electronic copy available at: https://ssrn.com/abstract=4151041

Stochastic Implied Volatility

- We built a market simulator for (a form of) implied volatilities
- We then removed the drift … such that the price processes become martingales
- We have created with machine learning a *stochastic implied volatility model* … a task not achieved through years of quantitative finance research !

Electronic copy available at: https://ssrn.com/abstract=4151041

Statistical Arbitrage

- Good
  - Nice theoretical framework
  - Model-free approach that works for any market model, any currency etc
  - Same numerical complexity as Deep Hedging

- But
  - Removing the drift is akin to giving up coming up on a better estimate on where the market goes → research into Knightian Uncertainty instead
  - Numerically not quite robust enough
    - Measure Q still has some marginal statistical arbitrage.
    - Static arbitrage and statistical arbitrage are very close to each other.
      For example, lack of sampling of rare events can introduce numerical “static” arbitrage (e.g. short term OTM puts never pay in our MC simulation)

Electronic copy available at: https://ssrn.com/abstract=4151041

Please ask questions

Electronic copy available at: https://ssrn.com/abstract=4151041