Adaptive Robust Stochastic Control and Statistical Surrogates

Tao Chen
Department of Statistics and Applied Probability
University of California, Santa Barbara
tchen@pstat.ucsb.edu

Joint work with Michael Ludkovski

9th Western Conference on Mathematical Finance

University of Southern California, Los Angeles

November 16 2018

Motivations

Adaptive Robust Parametric Markovian Control

- To control the risk due to model uncertainty
  (error in model estimation or model misspecification)
- A robust framework adaptively reduces uncertainty through learning

Numerical Implementation

- Solve discrete time robust Bellman equation
- Three challenges: continuous state space, integration, optimization
- Main hurdle: curse of dimensionality

Main Goals

- To propose and study an **adaptive robust control approach** for solving a discrete time Markovian control problem subject to Knightian uncertainty

- To develop a new numerical methodology in response to the challenges arise when scaling the approach to higher dimensions.

T. R. Bielecki, T. Chen, I. Cialenco, A. Cousin, and M. Jeanblanc *Adaptive Robust Control Under Model Uncertainty.* Submitted for Publication, 2017.

T. Chen and M. Ludkovski *Robust Stochastic Control and Statistical Surrogates.* In preparation, 2018.

Example: Dynamic Optimal Portfolio Selection

An investor is deciding on investing in a risky asset and a risk-free banking account by maximizing the expected utility $U(W_T)$ of the terminal wealth.

- r - the constant risk free rate
- e^{Z_t} - the return on the risky asset
- Assume that $Z_t = \mu^* + \sigma^* \varepsilon_t$, where ε_t are i.i.d. $\mathcal{N}(0,1)$
- The dynamics of the wealth process produced by a s.f. strategy

$$W_{t+1} = W_t(1 + r + u(e^{Z_{t+1}} - 1 - r)).$$

- Stochastic Control Problem if μ^* and σ^* are known:

$$\sup_{\vec{u} \in \mathcal{A}} \mathbb{E}_{\mu*,\sigma*}[U(W_T^{\vec{u}})].$$

Notations

- $(\Omega, \mathcal{F})$ - measurable space
- $\mathcal{T} = \{0, 1, \dots, T\}$ and $\mathcal{T}' = \{0, 1, \dots, T-1\}$
- $\boldsymbol{\Theta} \subset \mathbb{R}^d$ - known parameter space
- $X = \{X_t, t \in \mathcal{T}\}$ - observed process taking values in $\mathbb{R}^k$
- $Z = \{Z_t, t \in \mathcal{T}\}$ - observable i.i.d. or ergodic Markov random noise
- $\mathbb{F} = \{\mathscr{F}_t, t \in \mathcal{T}\}$ - the natural filtration of X
- $\{\mathbb{P}_\theta, \theta \in \boldsymbol{\Theta} \subset \mathbb{R}^d\}$ - measures associated with plausible laws of Z
- $\mathbb{P}_{\theta^*}$ - (unknown) true law of Z.

Classical Robust Approach

Due to model uncertainty, one considers the following robust stochastic control problem

$$\inf_{\vec{u}\in\mathcal{A}} \sup_{\theta\in\mathbf{\Theta}} \mathbb{E}_\theta[L(X,\vec{u})]$$

which (in some cases) is solved by Bellman equation of the form

$$V(t,x) = \inf_{u\in A} \sup_{\theta\in\mathbf{\Theta}} \mathbb{E}_\theta[V(t+1, f(x,u,Z_{t+1}^\theta))]. \tag{1.1}$$

- Select the best strategy u^* over the worst possible model θ_*.
- "static robustness" and no reduction of uncertainty: fixed adversarial choice of θ_* and $\mathbf{\Theta}$.
- If θ_* is far from the true model θ^*, this approach is overly conservative.

Adaptive Robust Approach

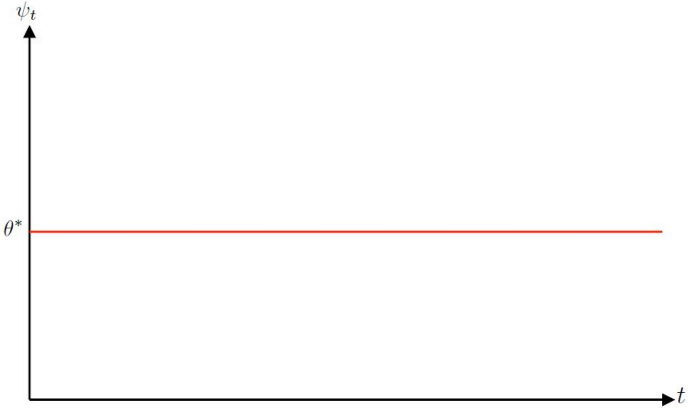
By incorporating learning into the robust approach, we achieve reduction of uncertainty through adaptively updating the parameter set. We have the following dynamic programming equation different from (1.1):

$$V(t, x, \bar{\theta}) = \inf_{u \in A} \sup_{\theta \in \boldsymbol{\Theta}_t(\bar{\theta})} \mathbb{E}_\theta[V(t+1, \mathbf{T}(x, \bar{\theta}, u, Z^\theta_{t+1}))]. \tag{1.2}$$

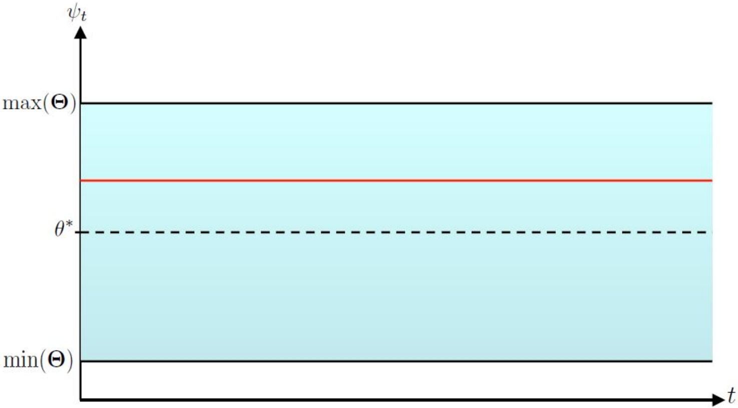
Equation (1.2) solves the robust stochastic control problem with reducing uncertainty:

- $\bar{\vartheta}$ is a point estimator of θ^* and $\bar{\theta}$ is the realization
- augmented state process $Y = (X, \bar{\vartheta})$ with dynamics $\mathbf{T}$
- dynamics of $\bar{\vartheta}$ is given in Bielecki, Chen, Cialenco (2017)
- $\boldsymbol{\Theta}_t$ is the confidence region centered at $\bar{\theta}_t$
- measurable selectors $\check{u}_t$ and $\check{\theta}_t$ exist

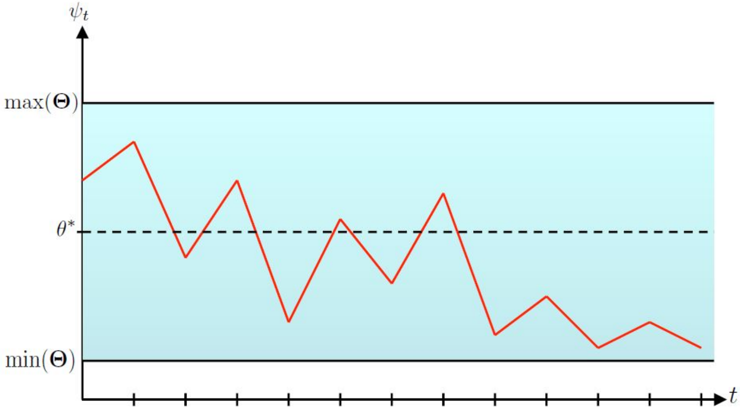
Without uncertainty



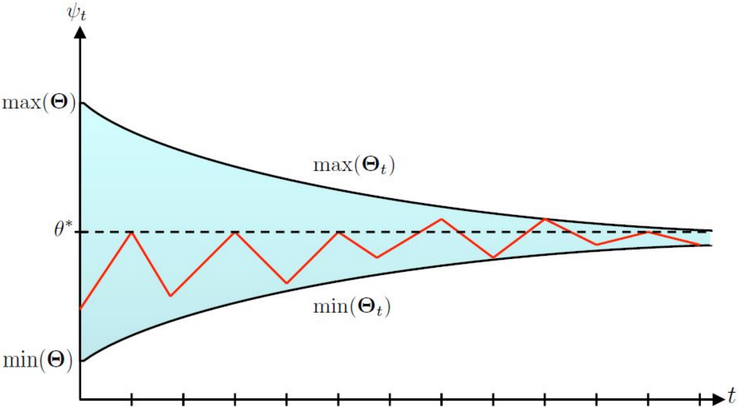
Robust



Strong robust



Adaptive Robust



Outline for Numerical Implementation

We want to find a numerical solver for

$$V(t,x,\bar{\theta}) = \inf_{u\in A} \sup_{\theta\in\tau(t,\bar{\theta})} \mathbb{E}_\theta[V(t+1,\mathbf{T}(t,x,\bar{\theta},a,Z^{\theta}_{t+1}))],$$

for $(x,\bar{\theta}) \in \mathbb{R}^k \times \mathbb{R}^d$.

- Discretization of the continuous state space for $Y = (X, \bar{\vartheta})$, accompanied by interpolation in order to evaluate $V(t,x,\bar{\theta})$
- Approximation of the integral since integrand is not analytically available
- Approximation of the optimizers $\check{u}(x,\bar{\theta})$ and $\check{\theta}(x,\bar{\theta},u)$
- The key idea is to recursively construct a functional approximation $\hat{V}(t,\cdot)$ that is used for interpolation and prediction

Curse of Dimensionality

As far as we know, no existing schemes are available when the state dimension is higher than 2. Specific difficulties include:

- Traditionally one constructs a grid of $(x, \bar{\theta})$-values, which is extremely inefficient for $k + d > 2$ and essentially impossible for $k + d > 4$
- Parametric representation of $\hat{V}$ (eg. in terms of polynomials in x and $\bar{\theta}$) is difficult for $k + d > 2$ and brings the concern for overfitting/underfitting
- The control u affects the evolution of the state x and prevents direct simulation of X as is done in the popular regression Monte Carlo paradigm

Existing Methods for Discretization and Interpolation

- Monte-Carlo based paradigm: simulate N trajectories according to a fixed measure $\mathbb{Q}^0$ and solve Bellman equation pathwise
- Such paradigm uses pre-specified $\mathbb{Q}^0$ that leads to non-adaptive experimental design
- Accompanied linear interpolation works very badly for high dimensional problem
- Estimated value functions are not smooth
- No way to deal with out-of-sample path

Spatial Modeling and Statistical Surrogates

- After discretization, $\hat{V}(t+1,\cdot)$ is only evaluated at sampled sites
- Popular interpolation methods will have to go through all sites all the time, which is very expensive for high dimensions
- If two state points y^1 and y^2 are close, then $\hat{V}(t+1,y^1)$ and $\hat{V}(t+1,y^2)$ should also be close
- Leverage already obtained solutions of similar optimization problems
- Build a spatial statistical model $\hat{V}(t+1,\cdot)$ for $V(t+1,\cdot)$ over the domain by learning the correlation structure of $V(t+1,\cdot)$ at sample sites

Gaussian Process Surrogates

- Non-parametric regression, similar to splines or kernel regression
- Multivariate Gaussian structure to describe the shape of $\hat{V}$ and $\tilde{u}$: covariance matrix $\mathbf{K}_{i,j} := K(y^i, y^j)$
- Train the model corresponds to applying the Gaussian conditional equations, and posterior is still Gaussian
- Statistical model for $\hat{V}$ and $\tilde{u}$ are described by the corresponding posterior means

$$m_*(y_*) = k(y_*)[\mathbf{K} + \sigma^2 \mathbf{I}]^{-1}\vec{e},$$
$$s_*(y_*, y'_*) = K(y_*, y'_*) - k(y_*)[\mathbf{K} + \sigma^2 \mathbf{I}]^{-1} k(y'_*).$$

Fitting the emulator = learning the hyperparameters in $\mathbf{K}$.

Experimental Design

- Accuracy of $\hat{V}(t, y_*)$ at given input y_* is directly related to the density of the design $\mathcal{D}_t$ around y_*
- Extrapolate for inputs far away from the simulated sites and depend on the prior mean
- Design $\mathcal{D}_t$ should be statistically sound by avoiding points that can't be observed in practice
- $\mathcal{D}_t$ should increase in time to avoid extrapolation when solving the Bellman equation backwards in time
- For a design, the choosing sites must fill in the space well
- Adaptively choose the design based on result of the previous step of numerical recursion

Basic Loop

$$V(t,x,\bar{\theta}) = \inf_{u\in A} \sup_{\theta\in\mathcal{T}(t,\bar{\theta})} \mathbb{E}_\theta[V(t+1,\mathbf{T}(t,x,\bar{\theta},a,Z^\theta_{t+1})] =: F(V(t+1,\cdot),x,\bar{\theta})$$

We have a $\text{fit} - \text{predict} - \text{optimize} - \text{fit}$ loop:

1. (Assume that the surrogate $\hat{V}(t+1,\cdot)$ has been fitted)
2. Select an experimental Design $\mathcal{D}_t$ of N_t sites y^n, $n = 1,\ldots,N_t$;
3. Solve the optimization problem at each y^n, using $\text{predict}(\hat{V}(t+1,y'))$ for the expectation. This yields the outputs $e^n = F(\hat{V}(t+1,\cdot),y^n)$ and optimal control $\check{u}^n$ at y^n;
4. Fit $\hat{V}(t,\cdot)$ based on data $(y^{1:N_t},e^{1:N_t})$ and $\tilde{u}(t,\cdot)$ based on $(y^{1:N_t},\check{u}^{1:N_t})$;
5. Goto 1: start the next recursion for $t-1$

Our Contribution

- We develop a machine learning framework tailored to generic stochastic min-max optimization problem.
- We recast the task of solving the Bellman equation as a statistical learning problem of fitting a surrogate (i.e. a statistical model) for $(x, \bar{\theta}) \mapsto \hat{V}(t, x, \bar{\theta})$.
- Gaussian Process surrogate for $\hat{V}$, coupled with an adaptive Experimental Design.
- Gaussian Process surrogate for $\tilde{u}$ allows us to predict the optimal control (for out-of-sample) paths without directly optimizing.

Dynamic Optimal Portfolio Selection

Recall the dynamic optimal portfolio selection problem where an investor wants to maximize the expected utility $U(W_T)$ of the terminal wealth.

- log return of the risky asset $Z_t = \mu + \sigma\varepsilon_t$, where ε_t are i.i.d. $\mathcal{N}(0,1)$
- Dynamics of the wealth process

$$\begin{aligned} W_{t+1} &= W_t(1 + r + \varphi_t(e^{Z_{t+1}} - 1 - r)) \\ &= W_t(1 + r + \varphi_t(e^{\mu+\sigma\varepsilon_t} - 1 - r)), \quad t \in \mathcal{T}',\ W_0 = w_0. \end{aligned}$$

- Adaptive Robust Stochastic Control Problem:

$$\sup_{\vec{u}\in\mathcal{A}} \inf_{\mathbb{Q}\in\mathcal{Q}^{\varphi,\Psi}} \mathbb{E}_{\mathbb{Q}}[U(W_T^{\vec{u}})].$$

Confidence Region

The MLE $\bar{\theta}_t = (\bar{\mu}_t, \bar{\sigma}_t^2)$ of the unknown parameter $\theta = (\mu, \sigma^2)$ can be expressed in the following recursive way:

$$\bar{\mu}_{t+1} = \frac{t}{t+1}\bar{\mu}_t + \frac{1}{t+1}Z_{t+1},$$
$$\bar{\sigma}_{t+1}^2 = \frac{t}{t+1}\bar{\sigma}_t^2 + \frac{t}{(t+1)^2}(\bar{\mu}_t - Z_{t+1})^2$$

Due to asymptotic normality of the MLEs, we have the recursive $1-\alpha$ confidence regions take the form

$$\mathbf{\Theta}_t = \tau_\alpha(t, \bar{\mu}, \bar{\sigma}) := \left\{(\mu, \sigma^2) \in \mathbb{R}^2 : \frac{t}{\bar{\sigma}^2}(\bar{\mu} - \mu)^2 + \frac{t}{2\bar{\sigma}^4}(\bar{\sigma}^2 - \sigma^2)^2 \le \kappa_\alpha\right\}$$

with κ_α being the $(1-\alpha)$–quantile of the χ_2^2 distribution.

Dimension-reduced Bellman Equation

The associated adaptive robust Bellman equation is

$$V(T, w, \bar{\mu}, \bar{\sigma}) = u(w),$$
$$V(t, w, \bar{\mu}, \bar{\sigma}) = \sup_{u \in A} \inf_{(\mu,\sigma)\in\tau_\alpha(t,\bar{\mu},\bar{\sigma})} \mathbb{E}_{\mu,\sigma}\left[V_{t+1}\left(\mathbf{T}(t, w, \bar{\mu}, \bar{\sigma}, u, Z_{t+1})\right)\right]$$

By choosing the CRRA utility function $U(x) = \frac{x^{1-\gamma}}{1-\gamma}$, we can be show that the ratio $\overline{V}$ defined as $\overline{V}(t, \bar{\mu}, \bar{\sigma}) := V(t, w, \bar{\mu}, \bar{\sigma})/w^{1-\gamma}$ satisfy the following backward recursion

$$\overline{V}(T, \bar{\mu}, \bar{\sigma}) = \frac{1}{1-\gamma},$$
$$\overline{V}(t, \bar{\mu}, \bar{\sigma}) = \inf_{u \in A} \sup_{(\mu,\sigma)\in\tau_\alpha(t,\bar{\mu},\bar{\sigma})} \mathbb{E}\Big[\left(1 + r + u(e^{\mu+\sigma\varepsilon_{t+1}} - 1 - r)\right)^{1-\gamma}$$
$$\overline{V}\Big(t+1, \frac{t}{t+1}\bar{\mu} + \frac{1}{t+1}(\mu + \sigma\varepsilon_{t+1}), \frac{t}{t+1}\bar{\sigma}^2 + \frac{t}{(t+1)^2}(\bar{\mu} - \mu - \sigma\varepsilon_{t+1})^2\Big)\Big].$$

Algorithm Part I

Backward recursion over the time steps for $t = T - 1, \dots, 0$

1. Create a design $\mathcal{D}_t = (\mu_t^{1:N_t}, \sigma_t^{1:N_t})$ that will be used to estimate $\hat{V}(t, \cdot)$. The design is based on Monte Carlo paradigm, Sobol space filling, and adaptively adding more points.
2. For $n = 1, 2, \dots, N_t$, let $f_2(u, \mu_t^n, \sigma_t^n) = \inf_{(\mu,\sigma)\in\tau(t,\mu_t^n,\sigma_t^n)} \hat{E}\Big[\hat{V}(t+1, \mathbf{T}(t, \mu_t^n, \sigma_t^n, u, \mu + \sigma\varepsilon))\Big]$. $\hat{E}$ is an approximate operator to estimate the expectation using quadrature rule or Monte Carlo.
3. Let $e_t^n := \sup_{u\in A} f_2(u, \mu_t^n, \sigma_t^n)$. Record the estimated optimal control $\check{u}_t^n$.
4. Build a GP model $\hat{V}(t, \cdot)$ for the link between (μ_t^n, σ_t^n) and (e_t^n). Build a GP model $\tilde{u}(t, \cdot)$ for the link between (μ_t^n, σ_t^n) and $(\check{u}_t^n)$.

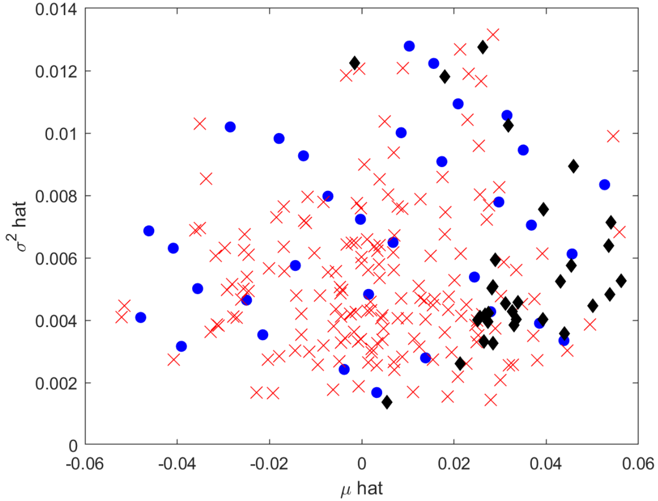
Algorithm Part II

Forward simulation on fresh paths, over the time steps for $t \in \mathcal{T}'$, to evaluate the performance of the strategy assuming the true probability model.

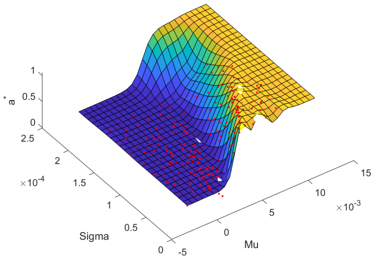
1. For path n, draw i.i.d. sequence of Z_t^n via ε_t^n.
2. Using the GP model to predict the control $u_{t-1}^n = \tilde{u}(t-1, \mu_{t-1}^n, \sigma_{t-1}^n)$.
3. Update the states according to $(w_{t+1}^n, \mu_{t+1}^n, \sigma_{t+1}^n) = \mathbf{T}(t+1, w_t^n, \mu_t^n, \sigma_t^n, u_{t-1}^n, Z_{t+1}^n)$.

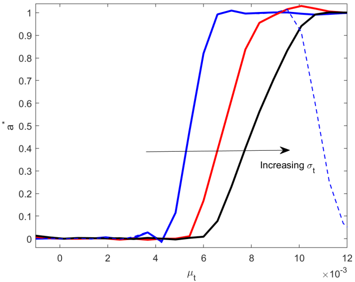
The final answer is the average $\underline{V}(0, w_0, \mu_0, \sigma_0) = \frac{1}{N}\sum_{n=1}^{N} U(w_T^n)$.

Simulation Design



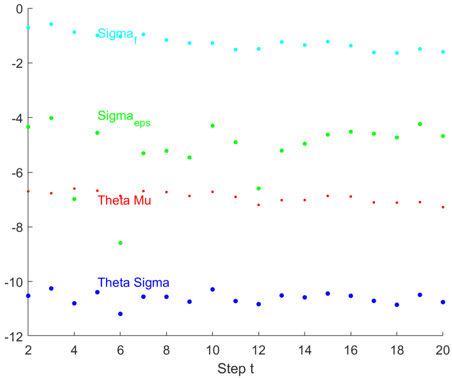
GP Fitting and Extrapolation





- In the area where points are sampled, GP surrogate works very well
- For GP, outputs corresponding to inputs far away from simulated sites are decided by mean function
- We shift $(\check{u}_t^n)$ by a sigmoid function $M(\mu) = (1 + e^{-(A\mu - B)})^{-1}$ and set the mean function as 0
- It improves the extrapolation but doesn't affect predictions inside the training domain

Stability of Hyperparameters



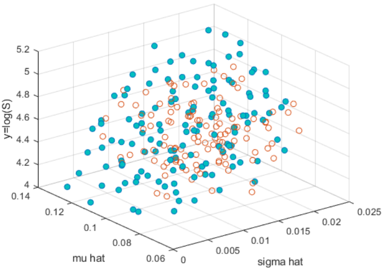
A major worry is that the GP model is mis-estimated, which can be diagnosed by an outlier in the hyperparameters.

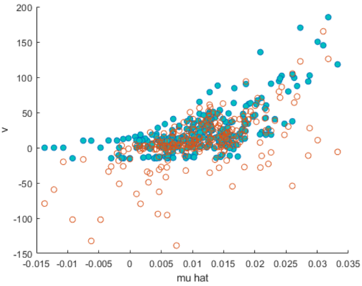
Adaptive Robust Hedging

In a similar setup, the investor wants to minimize the expected superhedging risk $\ell[(\Phi(S_T) - W_T)^+]$.

- $\ell : \mathbb{R}_+ \to \mathbb{R}_+$ is an increasing function such that $\ell(0) = 0$
- State process: $(X_t, \bar{\mu}_t, \bar{\sigma}_t, W_t)$, where $X_t = \log(S_t)$
- Need to modify the Monte Carlo paradigm as state W_t depends on control such that direct simulation is impossible
- $\mathcal{D}_t = (x_t^{1:N_t}, \mu_t^{1:N_t}, \sigma_t^{1:N_t}, \mathsf{BS}^\epsilon(s_t)^{1:N_t})$
- Randomization of the starting simulation sites is important due to strong correlation between X_t and $\bar{\mu}_t$

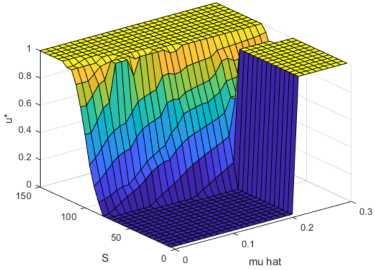
Simulation Design





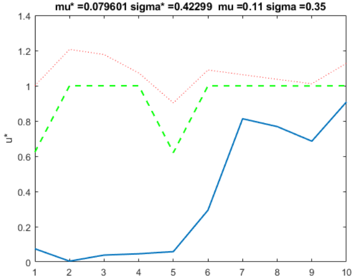
- $\inf_{(\mu,\sigma)\in\tau(t,\mu_t^n,\sigma_t^n)} \hat{E}\Big[\hat{V}(t+1, \mathbf{T}(t, x_t^n, \mu_t^n, \sigma_t^n, w_t^n, u, \mu+\sigma\varepsilon))\Big]$
- fit $\hat{V}(t+1,\cdot)$ at $\mathbf{T}(t, x_t^n, \mu_t^n, \sigma_t^n, w_t^n, u, \mu+\sigma\varepsilon)$ for (μ,σ) on $\partial\tau(t,\mu_t^n,\sigma_t^n)$
- place sites in the area reached by state process under the true $\mathbb{Q}^*$
- use space-filling method to add simulation sites to $\mathcal{D}_{t+1}$
- approximate the subspace of W_t by martingale prices of the option

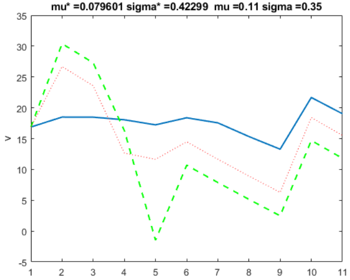
Multi-Start



- In some extreme scenarios, value function is flat in certain regions and `fminbnd` has trouble to pick the "proper" optimizer.
- Does not affect backward computation but the application to the out-of-sample paths.
- Choose the smallest u^* among all possible optimizers.
- Check the boundaries of A and multi-start.

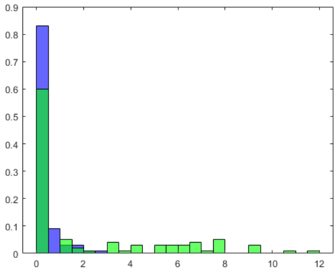
Solution of Adaptive Robust Hedging

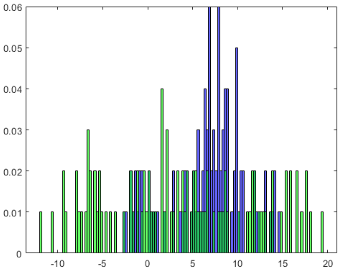




- Adaptive robust hedging strategy versus “adaptive” delta hedging.
- Adaptive robust hedging strategy is conservative in the beginning due to “uncertainty”.
- Strategy gets close to delta hedging near the end.

Solution of Adaptive Robust Hedging





- 100 ω's in the physical world, 100 pairs of (μ^*, σ^*): 10^4 out-of-sample paths
- Left: distribution of the super-hedge error; Right: distribution of hedging portfolio terminal wealth

Thank You !

The end of the talk ...
but not of the story