Bandits for Algorithmic Trading with Signals

Álvaro Cartea, Fayçal Drissi, Pierre Osselin

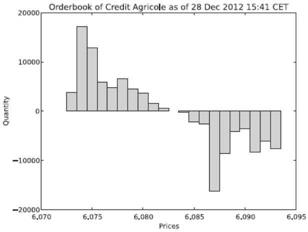
Fields-CFI Conference on Recent Advances in Mathematical Finance and Insurance
Oxford-Man Institute

25 September 2023

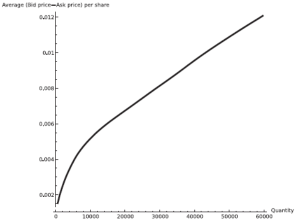
Optimal execution

Optimal execution

- Trading fast is expensive: execution costs and market impact.



(a) Execution costs.



(b) Permanent impact.

- Trading slowly is risky: the price may move adversely.

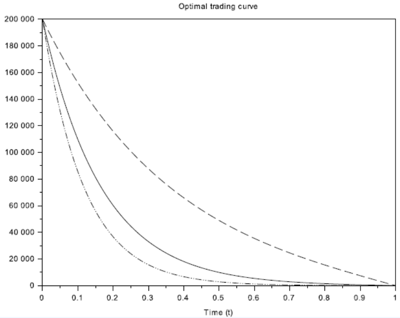
$\Longrightarrow$ Need to find an optimal trading trading schedule.

Optimal execution models

- Inputs: time horizon T, initial inventory y_0.
- Performance criterion: $\mathbb{E}[x_T + y_T\, S_T - \int_0^T \eta(y_t)dt]$.
- Assumptions and dynamics:

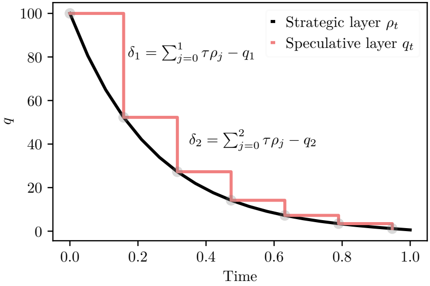
$$\text{Market impact : } dS_t = \sigma dW_t + \kappa\, \nu_t\, dt$$
$$\text{Execution costs: } dx_t = \nu_t S_t dt - L(v_t)dt\,.$$

- Result: an optimal (ideal) schedule $(\rho)_{t\in[0,T]}$.



In practice: slices and child orders

- The trading schedule is sliced into child orders.
- Discrete observation times $t \in \{0, \ldots, T\}$. At time t:
  - The trader compares the target inventory $\sum_{j=0}^{t} \tau \rho_j$ with her current inventory q_t.
  - Child order with quantity: $\delta_t = \sum_{j=0}^{t} \tau \rho_j - q_t$.



Extensions of optimal execution models

- Realistic assumptions: complex representations of the state space of market features and trading decisions.
- Market features (contextual information):
  - Predictive price signals.
  - Volatility and liquidity estimators.
  - Fill probability of limit orders.
- Trading decisions (actions):
  - Order type (limit order, market order, ...).
  - Depth of a limit order.
  - Trading venue (dark pool, lit LOB, OTC).

Extensions of optimal execution models in the literature

- Dynamic programming
  - + Interpretability and robustness.
  - - Need assumptions, dynamics, parameter estimation.
    Curse of dimensionality.
  - Solutions: NN PDE solvers.
    Approximation techniques.

- Machine learning (RL, NN)
  - + Scalability to complex environments.
  - - Poor interpretability and robustness.
    Noisy data.
    Training requires data (simulations).
  - Solutions: focus on interpretable problems and architectures.

Learning in algorithmic trading

Strategic and Speculative layers

Our approach: two layers for execution algorithms

- Strategic layer
  - An agent liquidates $\mathcal{Q}$ shares throughout $[0, T]$.
  - Encodes urgency, risk aversion, execution costs, market impact.
  - Result: trading schedule $(\rho_t)_{t\in\mathbb{R}}$.
  - Split the optimal schedule into child orders.
- Speculative layer
  - Targets ρ.
  - Optimises trading performance of child orders using unsupervised learning.
  - Actions: placement, timing, routing, ..
  - Market features: estimators/predictors.

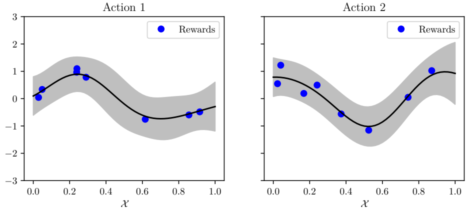
Optimal behaviour of the agent in the speculative layer

- New contextual non-stationary bandit (MTGP-LR).
- Actions: order type, placement, timing, routing of child orders.
- Context: market features (estimators and predictors).
- Reward for each arm k: Trading performance.
- Reward functions: multi-task Gaussian Process.

Multi-task Gaussian Process for the reward functions

- The agent learns the optimal mapping from contextual market features to optimal actions.
- UCB: Trade-off exploration (large posterior variance) / exploitation (large posterior mean).
- Two-layer approach for optimal execution
  - Guarantees of the strategic layer.
  - Interpretability of MTGPs.
  - No need for pre-training.
  - Adapts to non-stationary markets.

Multi-task Gaussian Process for the reward functions



- The agent learns the optimal mapping from contextual market features to optimal actions.
- UCB: Trade-off exploration (large posterior variance) / exploitation (large posterior mean).
- Two-layer approach for optimal execution
  - Guarantees of the strategic layer.
  - Interpretability of MTGPs.
  - No need for pre-training.
  - Adapts to non-stationary markets.

Speculative layer with MTGP-LR

Algorithm 1: MTGP-LR

Input:
Action space $\mathcal{A}$, Contextual features $\mathcal{X}$, time horizon $T > 1$.

while $t \leq T$ **do**
- Observe vector of contexts $\mathbf{x}_t \in \mathcal{X}$;
- Compute the vector **UCB** $\leftarrow \left\{ \tilde{\mu}(a, \mathbf{x}_t) + \tilde{\beta}_t^{1/2}\, \tilde{\sigma}(a, \mathbf{x}_t) \right\}_{a \in \mathcal{A}}$;
- Select action (trading decision) $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \mathbf{UCB}_a$;
- Observe reward (trading performance) y_t;
- **if** Change-point detected **then**
  - Reset the bandit;

$\Longrightarrow$ Sub-linear regret in stationary environments !

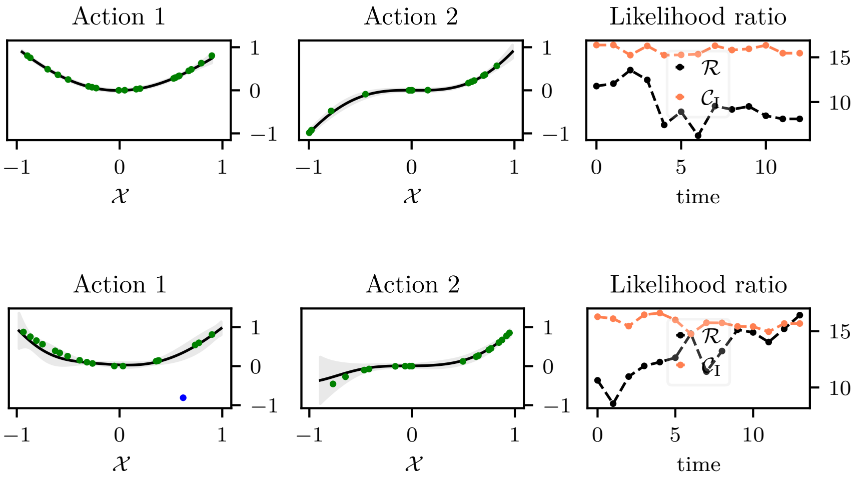
Non-stationary environments: Likelihood ratio test

- Let $\mathcal{W}$ be the time window that contains the last P rewards. Let $\mathcal{W} = \underline{\mathcal{W}} + \overline{\mathcal{W}}$.
- Hypothesis test:
  - $\boldsymbol{H_0}$: f_t in $\overline{\mathcal{W}}$ is the same as in $\underline{\mathcal{W}}$.
  - $\boldsymbol{H_1}$: f_t in $\overline{\mathcal{W}}$ is from a new MTGP.
- Likelihood ratio statistic

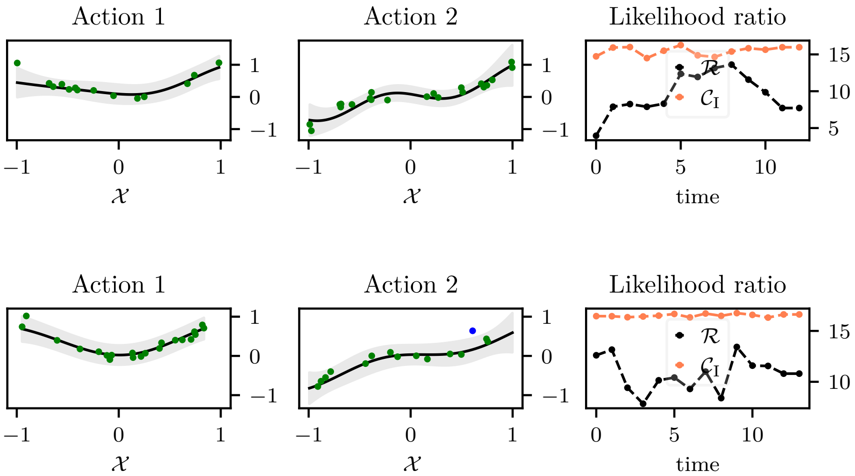
$$\mathcal{R} = 2\log\frac{p(\overline{\boldsymbol{y}}|\boldsymbol{H}_1)}{p(\overline{\boldsymbol{y}}|\boldsymbol{H}_0)} = -\,\overline{\boldsymbol{y}}^T(\overline{K}+\sigma_1^2 I)^{-1}\overline{\boldsymbol{y}} - \log|\overline{K}+\sigma_1^2 I| \\ + (\overline{\boldsymbol{y}}-\tilde{\boldsymbol{\mu}})^T(\tilde{K}+\sigma_0^2 I)^{-1}(\overline{\boldsymbol{y}}-\tilde{\boldsymbol{\mu}}) + \log|\tilde{K}+\sigma_0^2 I|\,.$$

- Statistical test: $\mathcal{R} \geq \mathcal{T} \implies$ reject null hypothesis.
- Inference error:
  - **Type I**: wrong detection of regime change.
  - **Type II**: missed detection of regime change.
- Results: we choose $\mathcal{T}$ to target a specific Type I or Type II inference error.

How to reset in non-noisy environments:
MTGP bandit with $\sigma = 0$, $P = 20$, $p = 10$, $\delta_\mathrm{l} = 0.4$, $\beta = 0.6$
$f^{a_1} : x \mapsto x^2$, $f^{a_2} : x \mapsto x^3$.



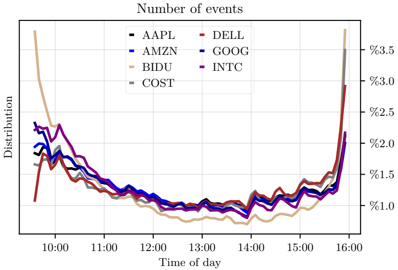
How to reset in noisy environments
MTGP bandit with $\sigma = 0.2$, $P = 20$, $p = 10$, $\delta_{\mathrm{I}} = 0.4$, $\beta = 0.6$
$f^{a_1} : x \mapsto x^2$, $f^{a_2} : x \mapsto x^3$.



Application: optimal timing with short term predictive signals

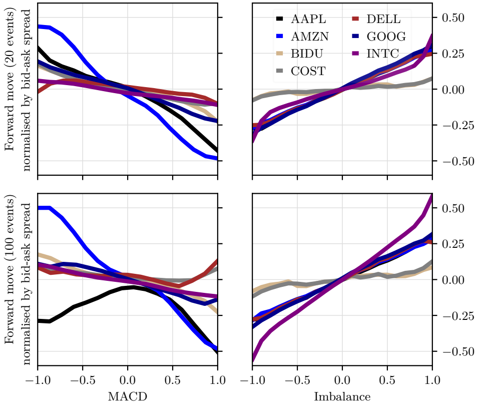
LOB data: 1 October 2022 $\longrightarrow$ 31 December 2022

Ticker	Avg. spread (in ticks)	Avg. queue size best bid	Avg. queue size best ask	Events per minute
AAPL	1.48	532.72	543.38	4472.36
AMZN	1.44	512.35	517.36	3939.42
BIDU	11.37	141.48	152.81	187.09
COST	24.62	73.39	72.33	177.15
DELL	1.60	396.65	390.34	361.98
GOOG	1.43	496.99	514.38	2766.57
INTC	1.16	5247.56	5197.65	1458.23



Short term predictive signals

- Volume imbalance: $I_t = \frac{Q_t^{\mathrm{B}} - Q_t^{\mathrm{A}}}{Q_t^{\mathrm{B}} + Q_t^{\mathrm{A}}}$
- MACD: $\begin{cases} \tilde{S}_t & = \mathrm{EMA}^{\varepsilon_1}(S_t) - \mathrm{EMA}^{\varepsilon_2}(S_t) \\ I_t^2 & = 10^5 \left(\tilde{S}_t - \mathrm{EMA}^{\varepsilon_3}\left(\tilde{S}_t\right) \right) / S_{t-\varepsilon_2-\varepsilon_3}, \\ \mathrm{EMA}^{\epsilon}(x_t) & = \varepsilon x_t + (1-\varepsilon)\,\mathrm{EMA}(x_{t-\Delta t}) \end{cases}$



Signals: predictive power depends on signal values

Ticker	Accuracy MACD	Accuracy IMB	Accuracy MACD extreme values	Accuracy IMB extreme values
AAPL	50.34%	62.84 %	60.24 %	77.25 %
AMZN	49.61 %	64.01 %	60.48 %	77.9 %
BIDU	49.23 %	52.33 %	50.38 %	58.16 %
COST	51.63 %	55.48 %	53.71 %	63.76 %
DELL	48.29 %	62.64 %	57.62 %	77.54 %
GOOG	48.77 %	64.82 %	60.97 %	79.27 %
INTC	52.58 %	81.93 %	72.80 %	98.03 %

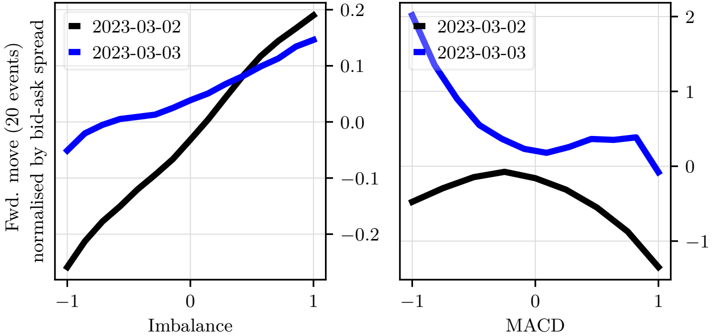
Table: Accuracy of MACD and imbalance computed as the hit ratio between signal predictions 20 events ahead and the realised price move.

Signals: noise

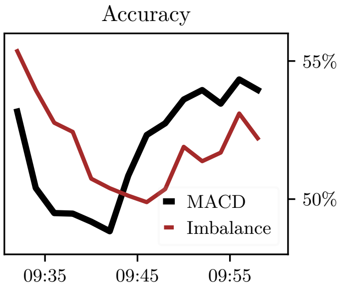
Mixing (very) noisy signals reduces statistical inference.

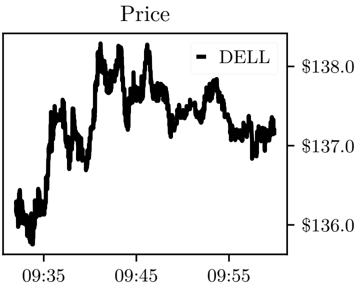
Ticker	Accuracy MACD	Accuracy IMB	Accuracy MACD+IMB
AAPL	50.34%	62.84 %	35.44 %
AMZN	49.61 %	64.01 %	34.21 %
BIDU	49.23 %	52.33 %	29.36 %
COST	51.63 %	55.48 %	31.58 %
DELL	48.29 %	62.64 %	33.01 %
GOOG	48.77 %	64.82 %	33.81 %
INTC	52.58 %	81.93 %	44.19 %

Signals: market regimes



Signals: market regimes

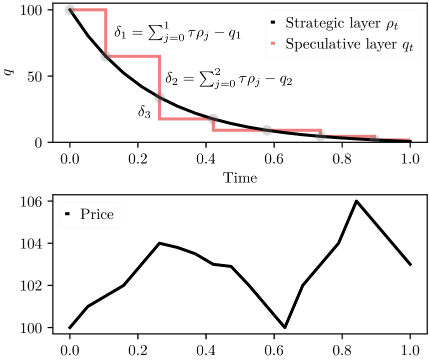




Optimal timing with MTGP-LR

- Context: signal values.
- Action k: trade with an MO of size δ_t immediately, or at $t+1$.
  - If $\delta_t \leq 0$, and $\mathbb{E}[S_{t+1}|I_t^k] \geq 0$, then sell at $t+1$.
  - If $\delta_t \leq 0$, and $\mathbb{E}[S_{t+1}|I_t^k] \leq 0$, then sell at t.
- Reward: improvement in execution price over threshold.

$$\left(S_t - \tilde{S}_{t+1}^k\right) \times \operatorname{sign}(\delta_t),$$



Short-term predictive signals in the literature

- Dynamics of the price and the signals

$$\mathrm{d}S_t = \pi_t\, dt + \sigma dW_t + \kappa\, \nu_t$$
$$\mathrm{d}\pi_t = \theta\,(\overline{\pi} - \pi_t)\,\mathrm{d}t + \gamma \mathrm{d}B_t\,.$$

- Short term (noisy) signals + long term objective.
- Non-stationarity: Continuous-time Markov Chain.
- A model for several signals needs prior knowledge :
  - Multivariate joint dynamics for the signals.
  - Exact form for how each signal drives the price.
  - Number of possible market regimes.
  - Transition probabilities between the regimes.

Performance study

Setup

- 7 securities from Nasdaq.
- 1000 execution programmes.
- Execution programme: 100 shares to liquidate in 10 minutes: buy/sell with probability $1/2$).

Ablation study

- **GP-LR**: bandit without transfer learning (multi-output GP).
- **GP**: bandit without change-point detection and transfer learning (multi-output GP).
- **UCB**: bandit without contextual features, change-point detection, and transfer learning.

- **Imbalance**: only volume imbalance.
- **MACD**: only MACD.

Average performance in USD per 10^6 USD traded.

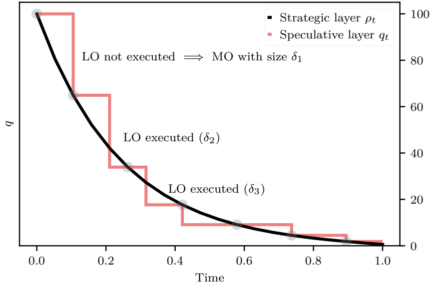
Ticker	Oracle	GP-LR	MTGP-LR	GP	UCB	Imbalance	MACD
BIDU	38.10	**8.63**	8.39	6.02	5.22	6.09	3.72
	$\pm$24.47	$\pm$18.21	$\pm$18.61	$\pm$18.37	$\pm$19.49	$\pm$19.24	$\pm$18.77
COST	16.44	4.56	**4.96**	4.66	3.15	3.76	2.15
	$\pm$12.03	$\pm$8.87	**$\pm$9.07**	$\pm$9.42	$\pm$9.57	$\pm$10.22	$\pm$8.67
AAPL	18.03	4.74	**5.18**	4.32	4.02	4.09	1.90
	$\pm$6.49	$\pm$5.51	**$\pm$6.61**	$\pm$5.01	$\pm$5.25	$\pm$5.04	$\pm$7.17
AMZN	11.72	4.70	**4.74**	4.04	2.80	4.70	0.70
	$\pm$2.66	$\pm$3.46	**$\pm$3.52**	$\pm$4.61	$\pm$4.40	$\pm$3.81	$\pm$3.63
DELL	25.97	3.97	**4.09**	3.45	3.65	3.85	2.85
	$\pm$16.36	$\pm$10.96	**$\pm$10.84**	$\pm$11.04	$\pm$11.64	$\pm$11.75	$\pm$11.16
GOOG	18.46	3.66	**4.08**	2.39	2.20	2.75	0.57
	$\pm$9.59	$\pm$7.11	**$\pm$7.13**	$\pm$7.35	$\pm$7.95	$\pm$7.54	$\pm$7.70
INTC	4.47	2.53	2.78	2.69	1.24	**3.56**	-0.54
	$\pm$4.89	$\pm$3.89	$\pm$4.02	$\pm$4.23	$\pm$3.85	**$\pm$4.14**	$\pm$3.77
Average	19.02	4.68	**4.89**	3.94	3.18	4.11	1.62

Application: optimal limit order placement

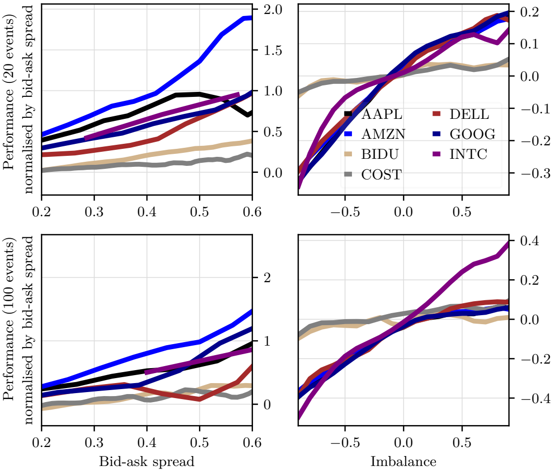
Optimal placement with MTGP-LR

- Context: bid-ask spread and imbalance.
- Action k: post limit order at the k^{th} limit at t, and send MO at $t+1$ if the order is not executed.
- Reward: improvement in execution price over threshold (MO at t).

$$\left(S_t - \tilde{S}^k_{t+1}\right) \times \operatorname{sign}\left(\delta_t\right),$$



First limit sell LOs

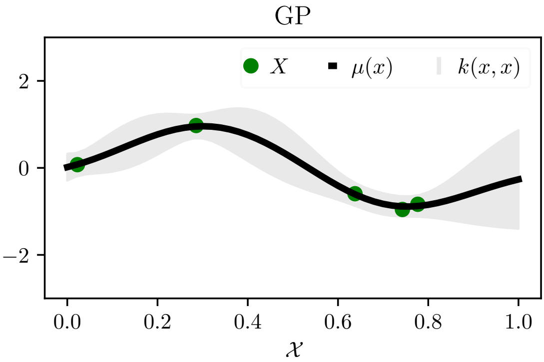


MTGP-LR

Introduction to GPs

Gaussian Processes

- A GP is a random function $f : \mathcal{X} \to \mathbb{R}$ where $\mathcal{X} \subseteq \mathbb{R}^m$.
- For any finite set of locations $\boldsymbol{X}_* \subseteq \mathcal{X}$, the random vector $\boldsymbol{f}_* = f(\boldsymbol{X}_*)$ follows a multivariate Gaussian distribution.

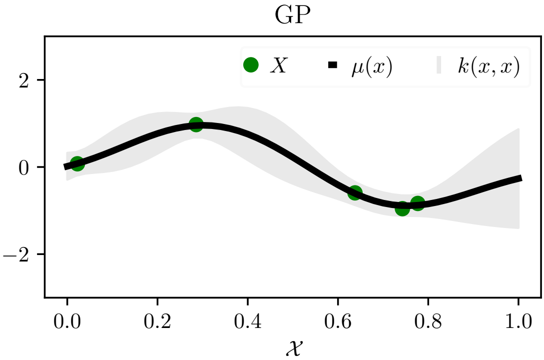


Gaussian Processes

- GPs are determined by a mean function μ and a covariance function $\boldsymbol{k}$.

$$f(X) \sim \mathcal{GP}(\mu(\boldsymbol{x}), \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x})) \implies \boldsymbol{f}_* \sim \mathcal{N}(\mu(\boldsymbol{x}_*), \boldsymbol{k}(\boldsymbol{x}_*, \boldsymbol{x}_*))$$

.

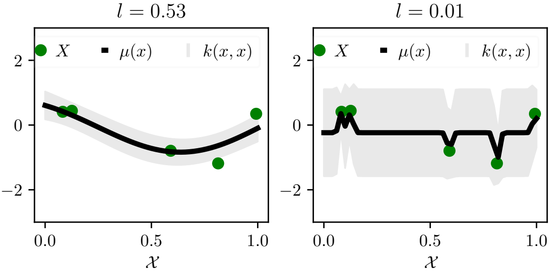


Kernel functions

- Covariance functions encode smoothness of function samples.
- Stationary kernels: $k(x, x') = k(x - x')$.
- Example: Squared Exponential kernel

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \exp\{-\|\boldsymbol{x} - \boldsymbol{x'}\| \,/\, 2\, l^2\},$$

- Small l indicate that sample functions change quickly. Large values indicate that they change slowly.



Learning with GPs

- The GP is determined by a finite set of Gaussian observations $X = \{x_1, \ldots, x_n\}$, where $y_i = f(\boldsymbol{x}_i) + \epsilon_i$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

- The GP's hyperparameters are: noise σ and hyperparameters of the prior's kernel k.

- The GP's hyperparameters are inferred using the marginal likelihood of the data:

$$
\begin{aligned}
L(\boldsymbol{\theta}, \sigma) &= \log p(\ \boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}, \sigma) \\
&= -\frac{1}{2} \log \left(\det \left(K_{\boldsymbol{\theta}} + \sigma^2 \ I \ \right) \right) - \frac{1}{2}\ \boldsymbol{y}^{\intercal} \left(K_{\boldsymbol{\theta}} + \sigma^2 \ I \ \right)^{-1} \boldsymbol{y} \\
&\quad - \frac{n}{2} \ \log(2\,\pi) \ ,
\end{aligned}
$$

Learning with GPs

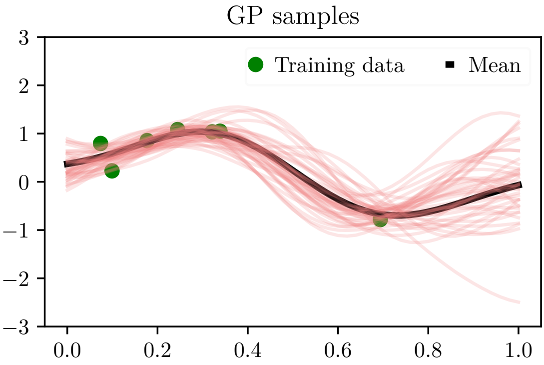
- Inference:

$$\begin{cases} \mu_{post}\left(\boldsymbol{x}_*\right) & = \boldsymbol{k}\left(\boldsymbol{x}_*, \boldsymbol{X}\right)\left(K + \sigma^2 I\right)^{-1} \boldsymbol{y}\,, \\ k_{post}\left(\boldsymbol{x}_*, \boldsymbol{x}'_*\right) & = k\left(\boldsymbol{x}_*, \boldsymbol{x}'_*\right) - \boldsymbol{k}\left(\boldsymbol{x}_*, \boldsymbol{X}\right)\left(K + \sigma^2 I\right)^{-1} \boldsymbol{k}\left(\boldsymbol{X}, \boldsymbol{x}'_*\right)\,, \end{cases}$$

Sampling with GPs

- Posterior sampling:

$$p\left(\boldsymbol{f}_{\star}|\boldsymbol{X}_{\star}, \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_{\star}, \boldsymbol{K}_{\star,\star}\right)$$



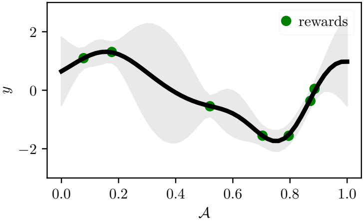
GP bandits (UCB)

- Space of actions: $\mathcal{A}$ (possibly continuous).
- Reward function: sample from a GP $f : \mathcal{A} \mapsto \mathbb{R}$.
- Observation times: $\mathcal{T} = \{1, ..., T\}$.

Algorithm 2: GP-UCB

Data: GP prior $\mu_0 = 0,\ \sigma_0,\ k$.
while $t \leq T$ **do**
  Choose action $a_t^\star \in \mathcal{A}$;
  Sample $y_t = f(a_t^\star) + \epsilon_t$;
  Perform Bayesian update for the GP mean μ_t and variance $\mathbf{K_t}$.
end

GP bandit: reward function



- UCB: $a_t^\star = \arg\max_{a \in \mathcal{A}} \tilde{\mu}_{t-1}(a) + \beta_t^{1/2}\, \tilde{\sigma}_{t-1}(a)$.
- Trade-off: exploration (large posterior variance) / exploitation (large posterior mean).

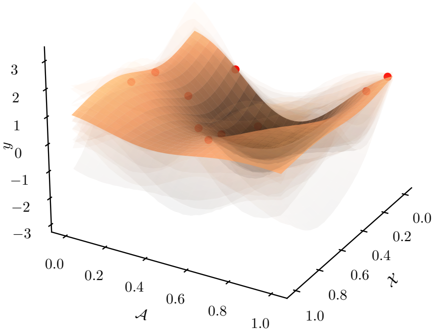
Contextual GP bandits (UCB) (in the literature)

- Space of actions: $\mathcal{A}$ (possibly continuous).
- Space of contextual features: $\mathcal{X} \in \mathbb{R}^m$.
- Reward function: sample from a GP $f : \mathcal{A} \times \mathcal{X} \mapsto \mathbb{R}$.
- Regularity of the reward function in the space of actions: $k^{\mathcal{A}}$.
- Regularity of the reward function in the space of contexts: $k^{\mathcal{X}}$.
- Kernel of the reward function: $k = k^{\mathcal{A}} \otimes k^{\mathcal{X}}$:

$$k((a,\mathbf{x}),(a',\mathbf{x}')) = k^{\mathcal{A}}(a,a') \times k^{\mathcal{X}}(\mathbf{x},\mathbf{x}'), \quad \forall \, (a,\mathbf{x}) \, , \left(a',\mathbf{x}'\right) \in (\mathcal{A},\mathcal{X}) \times (\mathcal{A},\mathcal{X}) \; .$$

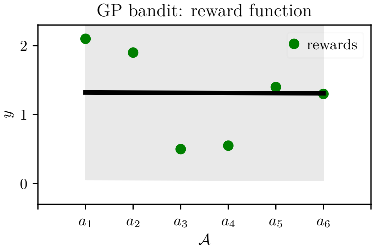
Algorithm 3: CGP-UCB

Data: GP prior $\mu_0 = 0,\ \sigma_0,\ k = k^{\mathcal{A}} \otimes k^{\mathcal{X}}$.
while $t \leq T$ **do**
- Observe context $x_t \in \mathcal{X}$;
- Select $a_t = \arg\max_{a \in \mathcal{A}} \tilde{\mu}_{t-1}(a, \mathbf{x}_t) + \beta_t^{1/2}\, \tilde{\sigma}_{t-1}(a, \mathbf{x}_t)$;
- Sample $y_t = f(a_t, x_t) + \epsilon_t$;
- Perform Bayesian update for the GP mean μ_t and variance $\mathbf{K_t}$;



Our case

- Space of actions is discrete (trading decisions for child orders).
- Similarity between all actions can be harmful (spurious transfer learning).
- Clustering effects between outcomes.

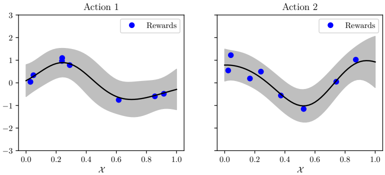


- Transfer learning only between actions that share common causal mechanisms.
- **Non stationarity**: the reward function f changes.

MTGP-LR

Reward function: MTGP

- For every action $a \in \mathcal{A}$, a latent function f_a maps the contexts to the rewards associated to that action.



- The reward function $f = \{f^a\}$ is a sample from a multi-task Gaussian Process (MTGP).

- The MTGP kernel function is

$$k((a,\mathbf{x}),(a',\mathbf{x}')) = k^{\mathcal{X}}(\mathbf{x},\mathbf{x}')\, K^{\mathcal{A}}_{a,a'}\,, \quad \forall\,(a,\mathbf{x})\,,\,(a',\mathbf{x}') \in (\mathcal{A},\mathcal{X}) \times (\mathcal{A},\mathcal{X})$$

- $\left(K^{\mathcal{A}}_{a,a'}\right)_{(a,a')\in\mathcal{A}\times\mathcal{A}}$ is a (non-parametric) positive semi-definite matrix that specifies how actions are related.

$\Longrightarrow$ Transfer learning between tasks without harming performance.

MTGP inference

- Posterior inference for $\boldsymbol{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$ ($\boldsymbol{w} = (a, \mathbf{x})$) is

$$\begin{cases} \boldsymbol{\mu}_{post}\left(\boldsymbol{w}_*\right) & = \boldsymbol{k}\left(\boldsymbol{w}_*, \boldsymbol{W}\right)\left(K + \sigma^2\ I\ \right)^{-1} \boldsymbol{y}\,, \\ k_{post}\left(\boldsymbol{w}_*, \boldsymbol{w}'_*\right) & = \boldsymbol{k}\left(\boldsymbol{w}_*, \boldsymbol{w}'_*\right) - \boldsymbol{k}\left(\boldsymbol{w}_*, \boldsymbol{W}\right)\left(K + \sigma^2\ I\ \right)^{-1} \boldsymbol{k}\ \left(\ \boldsymbol{W}, \boldsymbol{w}'_*\right)\,, \end{cases}$$

  where

  - $K = K^{\mathcal{A}} \odot K^{\mathcal{X}}$
  - $K^{\mathcal{X}}_{T,T} = \left(k^{\mathcal{X}}\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right)\right)_{i,j \in \{1,\ldots,T\}}$
  - The operator $\odot$ is the Hadamard (element-wise) matrix product.

- Kernel hyper-parameters learning:

$$(\boldsymbol{\theta}^*, \sigma^*, K^{\mathcal{A},\star}) = \underset{\boldsymbol{\theta},\sigma,K^{\mathcal{A}}}{\arg\max}\ L(\boldsymbol{\theta}, \sigma, K^{\mathcal{A}}),$$

The setup

- $\mathcal{A}$: finite set of actions. $\mathcal{X}$: set of contextual features.
- $\mathcal{T} = \{1, ..., T\}$: set of discrete observation times.
- The reward function $f_t : \mathcal{X} \times \mathcal{A} \mapsto \mathbb{R}$ is a sample from a MTGP.
- Non-stationarity: f_t is piecewise-stationarity
  - Finite number of change-points: $M = 1 + \sum_{t=1}^{T} \mathbb{1}_{\{f_t \neq f_{t-1}\}}$.
  - Observation times when the reward function changes: $\{\nu_j\}_{j\in\{0,\ldots,M\}} \subset \mathcal{T}$.
  - Segments of stationarity: $[\nu_{j-1}, \nu_j]$, for $j \in \{1, ..., M\}$.
- Bandit: $\left\{\mathcal{A}, \mathcal{X}, \mathcal{T}, (f_t)_{t\in\mathcal{T}}\right\}$.
- Objective: minimise the regret

$$R(T) = \sum_{t=1}^{T} r_t = \sum_{t=1}^{T} f_t(a_t^*, \mathbf{x}_t) - f_t(a_t, \mathbf{x}_t)\,,$$

where $a_t^* = \arg\max_{a\in\mathcal{A}} f_t(a, \mathbf{x}_t)$.

The algorithm in stationary environments

Algorithm 4: MTGP-LR (stationary)

Data: Number of actions N, number of contextual features m, action space $\mathcal{A}$, context space $\mathcal{X}$, $\beta > 0$, kernel $k_{\boldsymbol{\theta}}^{\mathcal{X}}$.
$\mathcal{D} \leftarrow \{\}$;
$f \sim MTGP(0, K_{\boldsymbol{\theta}})$;
while $t \leq T$ **do**
- Observe vector of contexts $\mathbf{x}_t \in \mathcal{X}$;
- Compute vector **UCB** $\leftarrow \left[\tilde{\mu}_{t-1}(a, \mathbf{x}_t) + \beta_t^{1/2}\, \tilde{\sigma}_{t-1}(a, \mathbf{x}_t)\right]_{a \in \mathcal{A}}$;
- $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \mathbf{UCB}_a$;
- Select action a_t; Sample reward $y_{a_t,t}$;
- $\mathcal{D} \mathrel{+}= \{(y_t, \mathbf{x}_t, a_t)\}$;
- Retrain the kernel hyper-parameters $\boldsymbol{\theta}$ with data set $\mathcal{D}$;

Theorem: regret in stationary environments

Fix $\delta \in (0, 1)$. The regret of MTGP-LR is bounded with high probability

$$\mathbb{P}\left(R(T) \leq \sqrt{C_1 \; T \; \beta_T \; \gamma_T} + 2, \quad \forall T \geq 1\right) \geq 1 - \delta \,, \tag{1}$$

where

$$\beta_T = 2 \log(N \, T^2 \, \pi^2 / 3 \, \delta \,). \tag{2}$$

Moreover, if the MTGP's kernel $k^{\mathcal{X}}$ is the Squared Exponential, Matérn, or Linear kernel, then, the regret bound is sub-linear.

Non-stationary environments: Likelihood ratio test

- Let $\mathcal{W}$ be the time window that contains the last P rewards.
- Let $\overline{\mathcal{W}} \subset \mathcal{W}$ be the sub-window containing the most recent $p \leq P$ points of $\mathcal{W}$, and let $\underline{\mathcal{W}} = \mathcal{W} \setminus \overline{\mathcal{W}}$.
- Hypothesis test:
  - Null hypothesis $\boldsymbol{H_0}$: f_t in $\overline{\mathcal{W}}$ and the noise σ_0 are the same as in $\underline{\mathcal{W}}$.
  - Alternative hypothesis $\boldsymbol{H_1}$: f_t in $\overline{\mathcal{W}}$ and the noise σ_1 are from a new MTGP.
- Likelihood ratio statistic

$$\mathcal{R} = 2 \log \frac{p(\overline{\boldsymbol{y}}|\boldsymbol{H}_1)}{p(\overline{\boldsymbol{y}}|\boldsymbol{H}_0)} = -\,\overline{\boldsymbol{y}}^T(\overline{K} + \sigma_1^2 I)^{-1}\overline{\boldsymbol{y}} - \log|\overline{K} + \sigma_1^2 I| + (\overline{\boldsymbol{y}} - \tilde{\boldsymbol{\mu}})^T(\tilde{K} + \sigma_0^2 I)^{-1}(\overline{\boldsymbol{y}} - \tilde{\boldsymbol{\mu}}) + \log|\tilde{K} + \sigma_0^2 I|\,.$$

- Statistical test: $\mathcal{R} \geq \mathcal{T} \implies$ reject null hypothesis.
- Inference error:
  - **Type I**: wrong detection of regime change.
  - **Type II**: missed detection of regime change.

Propositon (Type I error guarantee: wrong detection) Let $\delta_{\mathrm{I}} \in (0,1)$, and let λ_{i,H_0} be the eigenvalues of the matrix $\boldsymbol{V}_{H_0}^{\frac{1}{2}} \Lambda \boldsymbol{V}_{H_0}^{\frac{1}{2}}$ and $\mu_{H_0} = \mathbb{E}[\mathcal{R}|H_0]$. The threshold

$$\mathcal{C}_{\mathrm{I}} = \mu_{H_0} + \max\left\{ \sqrt{8 \ln \frac{1}{\delta_{\mathrm{I}}} \left(\sum_i \lambda_{i,H_0}^2 + \tilde{\boldsymbol{\mu}}^T (\overline{K} + \sigma_1^2 I)^{-1} V_{H_0} (\overline{K} + \sigma_1^2 I)^{-1} \tilde{\boldsymbol{\mu}} \right)}\,, \right.$$
$$\left. - 8 \ln \{\delta_{\mathrm{I}}\} \max_i \{|\lambda_{i,H_0}|\} \right\}$$

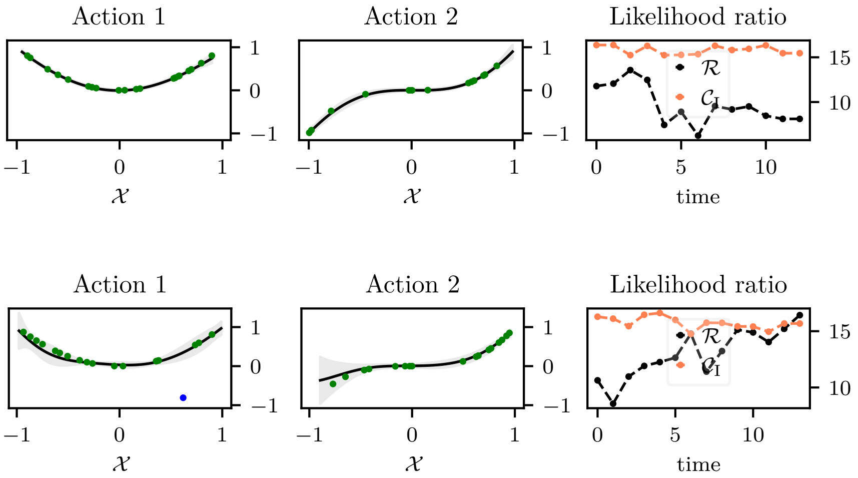
guarantees a type I error probability of at most δ_{I}.

Propositon (Type II error guarantee: missed detection) Let $\delta_{\mathrm{II}} \in (0,1)$, and let λ_{i,H_1} be the eigenvalues of the matrix $\boldsymbol{V}_{H_1}^{\frac{1}{2}} \Lambda \boldsymbol{V}_{H_1}^{\frac{1}{2}}$ and $\mu_{H_1} = \mathbb{E}[\mathcal{R}|H_1]$. The threshold

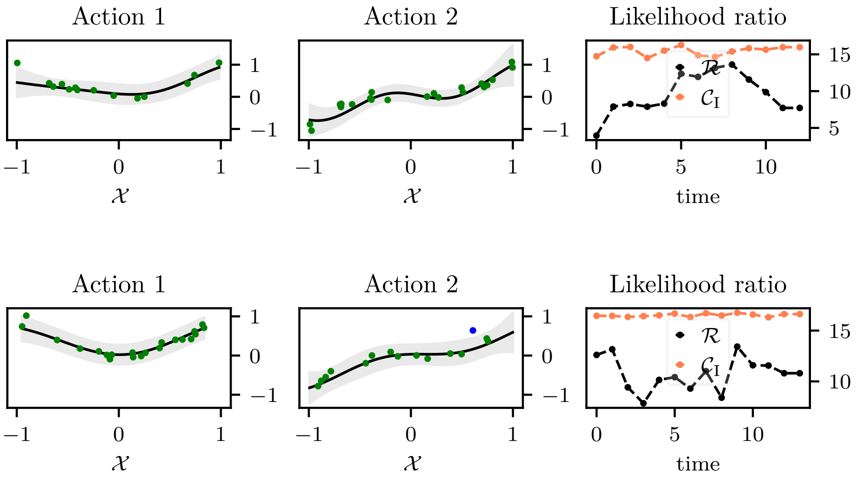
$$\mathcal{C}_{\mathrm{II}} = \mu_{H_1} - \max\left\{ \sqrt{8 \ln \frac{1}{\delta_{\mathrm{II}}} \left(\sum_i \lambda_{i,H_1}^2 + \tilde{\boldsymbol{\mu}}^T (\tilde{K} + \sigma_0^2 I)^{-1} V_{H_1} (\tilde{K} + \sigma_0^2 I)^{-1} \tilde{\boldsymbol{\mu}} \right)}\,, \right.$$
$$\left. - 8 \ln \{\delta_{\mathrm{II}}\} \max_i \{||\lambda_{i,H_1}||\} \right\}$$

guarantees a type II error probability of at most δ_{II}.

How to reset in non-noisy environments:
MTGP bandit with $\sigma = 0$, $P = 20$, $p = 10$, $\delta_{\mathrm{l}} = 0.4$, $\beta = 0.6$
$f^{a_1} : x \mapsto x^2$, $f^{a_2} : x \mapsto x^3$.



How to reset in noisy environments
MTGP bandit with $\sigma = 0.2$, $P = 20$, $p = 10$, $\delta_{\mathrm{l}} = 0.4$, $\beta = 0.6$
$f^{a_1} : x \mapsto x^2$, $f^{a_2} : x \mapsto x^3$.



Algorithm 5: MTGP-LR

Input:
Sequence $(\beta_t)_{t\in\mathcal{T}} > 0$, kernel $k_{\boldsymbol{\theta}}$, number of actions N, number of contextual features m,
$P \in \mathbb{N}^\star$, $p \leq P$, time horizon $T > 1$,
probability bound $\delta_{\mathrm{I}} \in (0,1)$ or probability bound $\delta_{\mathrm{II}} \in (0,1)$.

Initialise:
$\mathcal{D} \leftarrow \{\}$ is the data set for the current stationary segment
$f \sim \mathrm{MTGP}(\mathbf{0}, \boldsymbol{K_\theta})$
$\tilde{\boldsymbol{\beta}}_{1:T} \leftarrow \boldsymbol{\beta}_{1:T}$

while $t \leq T$ **do**
  Observe the vector of contexts $\mathbf{x}_t \in \mathcal{X}$
  Compute the vector **UCB** $\leftarrow \left\{\mu_{t-1}(a, \mathbf{x}_t) + \tilde{\beta}_t^{1/2}\, \sigma_{t-1}(a, \mathbf{x}_t)\right\}_{a\in\mathcal{A}}$
  Select action $a_t \leftarrow \arg\max_{a\in\mathcal{A}}$ $\mathbf{UCB}_a$
  Observe reward y_t
  Add $\{y_t, \mathbf{x}_t, a_t\}$ to data set $\mathcal{D}$
  if $\#\mathcal{D} \geq P$ **then**
    Compute LR statistic $\mathcal{R}$ with data sets $\underline{\mathcal{D}}$ and $\overline{\mathcal{D}}$
    Compute threshold $\mathcal{C} = \mathcal{C}_{\mathrm{I}}$ with the bound δ_{I} for the probability of type I error.
     Alternatively, compute $\mathcal{C} = \mathcal{C}_{\mathrm{II}}$ with δ_{II} for the probability of type II error.
    if $\mathcal{R} \geq \mathcal{C}$ **then**
      $\mathcal{D} \leftarrow \overline{\mathcal{D}}$
      $\tilde{\boldsymbol{\beta}}_{t+1:T} \leftarrow \boldsymbol{\beta}_{1:T-t}$
  Retrain the hyper-parameters $\boldsymbol{\theta}$ with data set $\overline{\mathcal{D}}$ and log marginal likelihood.

Thank you for listening!

Any questions?

Why not RL

- bandit assumption: actions do not affect price dynamics or the predictive power of signals.
- Actions in the RL setting affect both rewards and states.
- RL training requires proper simulation and modeling of market impact.
- Impact is in the strategic layer.
- Possible to penalise reward with model-specific impact parameter.

Why not regression

- Predictive / descriptive power of features depends on the features values.
- Opposing signals can lead to cancellation of predictive power.
- Mixing (very) noisy signals in non stationary environments reduces statistical inference.

Computational complexity of GPs: many options

- Trailing period $\implies$ maximum size for the matrix to inverse.
- Discretise the space of contextual features and group observations: reduces noise.
- Low-rank matrix approximation
  - Lanczos algorithm (iterative method to find the *m* most useful eigenvalues).
  - Control of the size of the low rank decomposition used for samples.

  Pleiss, G., Gardner, J., Weinberger, K., Wilson, A. G. (2018, July). Constant-time predictive distributions for Gaussian processes.