Reinforcement Learning in Continuous Time

Xunyu Zhou

Columbia University

June 2024 Rennes, Workshop Celebrating Ying Hu

Background and Motivation

Theory and Algorithms

Conclusions

Multi-Armed Bandits

- m slot machines in a casino, with different but unknown winning probabilities – in what sequence to play the machines?
- Classical model-based approach: first estimate (explore) and then optimize (exploit) - "Separation principle" or "plug-in"
- Reinforcement learning (RL) approach: explore and exploit *simultaneously* - trades off exploration (learning) and exploitation (optimization)
- ε-greedy strategy (Sutton and Barto 1998): playing the *current* best machine with probability $1 - \varepsilon$ and the other machines at random with probability ε

A Game Changer

- ε-greedy strategy is a *randomized* policy/strategy (trial and error)
- *The gambler learns the best (randomized) policies instead of learning a model*

Key Elements of Reinforcement Learning

- *Exploration* (trial and error): broaden search space via randomization (stochastic policies)
- *Policy evaluation* (PE): estimate value (objective) function of a given policy using samples only
- *Policy improvement* (PI): improve and update current policy based on learned value function, including policy gradient (PG) and Q-learning
- *Convergence and regret analysis*: convergence of the policy parameters and loss of objective value compared with oracle access

Pitfalls of Current RL Study

- Two major limitations in existing study on RL
  - Mainly for discrete-time Markov Decision Processes (MDPs)
  - Many RL algorithms devised in heuristic and *ad hoc* manners
- There seems a lack of an overarching theoretical understanding and a *unified* framework for RL methods

RL in Continuous Time and Spaces

- Bridge these gaps by providing a unified theoretical underpinning of RL in continuous time with possibly continuous state and action spaces
- Carry out all theoretical analysis for the continuous setting and take discrete *observations* at the final, algorithmic stage
- Rule out sensitivity in time step size
- Make use of well-developed tools in stochastic calculus, differential equations, and stochastic control, which enables better interpretability/explainability to underlying learning technologies
- Provide new perspectives on RL overall

Research Questions

- How to explore strategically?
- How to do PE?
- How to do PI generally?
- How to do PG specifically?
- Do we have sublinear regret?

A Pentalogy

- H. Wang, T. Zariphopoulou and X. Zhou, "Reinforcement learning in continuous time and space: A stochastic control approach", *Journal of Machine Learning Research*, 2020.
- Y. Jia and X. Zhou, "Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach", *Journal of Machine Learning Research*, 2022a.
- Y. Jia and X. Zhou, "Policy gradient and actor–critic learning in continuous time and space: Theory and algorithms", *Journal of Machine Learning Research*, 2022b.
- Y. Jia and X. Zhou, "q-Learning in continuous time", *Journal of Machine Learning Research*, 2023.
- W. Tang and X. Zhou, "Regret of exploratory policy improvement and q-learning", working paper.

Problem Formulation

- $\left(\Omega, \mathcal{F}, \mathbb{P}; \{\mathcal{F}_t^W\}_{t\geq 0}\right)$, Brownian motion $W = \{W_t,\, t \geq 0\}$
- Action space $\mathcal{A}$: representing constraints on an agent's actions (or "controls")
- Admissible action or control $a = \{a_t,\, t \geq 0\}$: an $\{\mathcal{F}_t^W\}_{t\geq 0}$-adapted measurable process taking value in $\mathcal{A}$
- State (or "feature") dynamics governed by SDE in $\mathbb{R}^d$

$$\mathrm{d}X_t = b(t, X_t, a_t)\mathrm{d}t + \sigma(t, X_t, a_t)\mathrm{d}W_t,\ t > 0$$

- Objective: to achieve maximum expected total reward represented by *optimal value function*

$$w\,(t, x) := \sup \mathbb{E}\left[\int_t^T r\,(s, X_s, a_s)\,\mathrm{d}s + h(X_T)\,\middle|\, X_t = x\right],$$

where $(t, x) \in [0, T] \times \mathbb{R}^d$

Classical Model-Based Approach

- Dynamic programming (Fleming and Soner 1992, Yong and Z. 1998)
- HJB equation: optimal value function w satisfies

$$\frac{\partial v}{\partial t}(t,x) + \sup_{a\in\mathcal{A}} H(t,x,a,\frac{\partial v}{\partial x}(t,x),\frac{\partial^2 v}{\partial x^2}(t,x)) = 0; \quad v(T,x) = h(x)$$

- ... where (generalized) *Hamiltonian* (Yong and Z. 1998)

$$H(t,x,a,p,P) = \frac{1}{2}\text{tr}\left[\sigma(t,x,a)' P\sigma(t,x,a)\right] + p\cdot b(t,x,a) + r(t,x,a)$$

- Verification theorem: optimal (feedback) control *policy* is

$$\mathbf{a}(t,x) = \text{argmax}_{a\in\mathcal{A}} H\left(t,x,a,\frac{\partial v}{\partial x}(t,x),\frac{\partial^2 v}{\partial x^2}(t,x)\right)$$

- *Deterministic* policy, devised at $t=0$
- This approach requires oracle access of environment (functional forms of b,σ,r,h)

Exploratory Formulation (Wang et al. 2020, JMLR)

- *Exploratory* control $\pi = \{\pi_t(\cdot), t \geq 0\}$: a density-function-valued adaptive process
- Exploratory state dynamics, a controlled stochastic differential equation (SDE)

$$dX_t^\pi = \tilde{b}(t, X_t^\pi, \pi_t)dt + \tilde{\sigma}(t, X_t^\pi, \pi_t)dW_t,\ t > 0; \quad X_0^\pi = x, \quad (1)$$

where

$$\tilde{b}(t, X_t^\pi, \pi_t) := \int_{\mathcal{A}} b\left(t, X_t^\pi, a\right) \pi_t(a)da, \quad (2)$$

and

$$\tilde{\sigma}(t, X_t^\pi, \pi_t) := \sqrt{\int_{\mathcal{A}} \sigma^2\left(t, X_t^\pi, a\right) \pi_t(a)da} \quad (3)$$

- *Entropy-regularized* value function

$$J\left(t, x; \pi\right) = \mathbb{E}\Big[\int_0^T \left(\int_{\mathcal{A}} r\left(s, X_s^\pi, a\right) \pi_s\left(a\right) da - \gamma \int_{\mathcal{A}} \pi_s(a) \ln \pi_s(a) da \right) ds + h(X_T^\pi) \Big| X_t^\pi = x \Big] \quad (4)$$

where $\gamma > 0$ is an exogenous weighting parameter

Exploratory HJB Equation and Verification

- Optimal value function $V(t,x) = \sup_\pi J(t,x;\pi)$
- V satisfies *exploratory* HJB

$$v_t(t,x) + \sup_{\pi \in \mathcal{P}(\mathcal{A})} \int_{\mathcal{A}} \left[H(t,x,a,v_x(t,x),v_{xx}(t,x)) - \gamma \ln \pi(a) \right] \pi(a) da = 0,$$

  with $v(T,x) = h(x)$
- Optimal *feedback* control (a *stochastic* policy)

$$\boldsymbol{\pi}^*(a|t,x) = \frac{1}{Z(\gamma)} \exp\left(\frac{1}{\gamma} H(t,x,a,v_x(t,x),v_{xx}(t,x)) \right),$$

  where $a \in \mathcal{A}, \ (t,x) \in [0,T] \times \mathbb{R}^d$, and

$$\begin{aligned} Z(\gamma) &\equiv Z(\gamma,t,x,v_x(t,x),v_{xx}(t,x)) \\ &:= \int_{\mathcal{A}} \exp\left(\tfrac{1}{\gamma} H(t,x,a,v_x(t,x),v_{xx}(t,x)) \right) da \end{aligned}$$

  is the normalizing factor
- Gibbs measure

Extensions and Applications

- Gaussian exploration for LQ (Wang, Zariphopoulou and Z. 2020, JMLR)
- Mean–variance (Wang and Z. 2020, MF)
- Well-posedness of exploratory HJB equation (Tang, Zhang and Z. 2022, SICON)
- Simulated annealing (Gao, Xu and Z. 2022, SICON)
- Mean field games learning (Guo, Xu and Zariphopoulou 2022, MOR)
- Learning equilibrium mean-variance strategy (Dai, Dong and Jia 2023, MF)
- Non-entropy regularization (Han, Wang and Z. 2023, SICON)

Function Approximation

- Need to learn various functions (e.g. optimal value function and policy) in machine learning
- *Function approximation*: approximates the functions to be learned by parametric families of functions with finite-dimensional parameters
- Parametric forms may be inspired by problem structure or represented by neural networks

Policy Evaluation by Jia and Z. (2022a)

- To evaluate a given stochastic policy without knowing model parameters
- Martingale condition (by Feynman–Kac and BSDE)
- Martingality leads to a loss function and an orthogonality system of equations
- Solvable by stochastic gradient descent and stochastic approximation respectively

Policy Gradient by Jia and Z. (2022b)

- To compute gradient of the (parameterized) value function of a given policy
- Policy gradient turned into policy evaluation mathematically by considering an auxiliary running reward function
- This auxiliary reward function value along state is observable/accessible (i.e. data driven) by Ito's formula

Policy Improvement

Theorem (Wang and Z. 2020, Jia and Z. 2023)

Given $\boldsymbol{\pi} \in \mathbf{\Pi}$, *define*

$$\boldsymbol{\pi}'(\cdot|t,x) \propto \exp\left\{\frac{1}{\gamma} H\big(t, x, \cdot, \frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\right\}.$$

If $\boldsymbol{\pi}' \in \mathbf{\Pi}$, *then*

$$J(t,x;\boldsymbol{\pi}') \geq J(t,x;\boldsymbol{\pi}).$$

Moreover, if the following map

$$\mathcal{I}(\boldsymbol{\pi}) = \frac{\exp\{\frac{1}{\gamma} H\big(t, x, \cdot, \frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma} H\big(t, x, a, \frac{\partial J}{\partial x}(t,x;\boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t,x;\boldsymbol{\pi})\big)\} \mathrm{d}a}, \quad \boldsymbol{\pi} \in \mathbf{\Pi}$$

has a fixed point $\boldsymbol{\pi}^*$ *on* $\mathbf{\Pi}$, *then* $\boldsymbol{\pi}^*$ *is the optimal policy.*

Q-Learning

- The previous theorem is not implementable for learning because both H and J are unknown
- Recall classical stochastic control

$$w(t,x) = \sup \mathbb{E}\left[\int_t^T r(s, X_s, a_s)\,\mathrm{d}s + h(X_T) \,\middle|\, X_t = x \right]$$

- With fixed $\Delta t > 0$, Bellman's principle of optimality

$$w(t,x) = \sup \mathbb{E}\left[\int_t^{t+\Delta t} r(s, X_s, a_s)\,\mathrm{d}s + w(t+\Delta t, X_{t+\Delta t}) \,\middle|\, X_t = x \right]$$

- Q-function

$$Q_{\Delta t}(t,x,a) = \mathbb{E}\left[\int_t^{t+\Delta t} r(s, X_s, a)\,\mathrm{d}s + \sup_{a'} Q_{\Delta t}(t+\Delta t, X_{t+\Delta t}, a') \,\middle|\, X_t = x \right]$$

- $a^*(t,x) = \arg\max_a Q_{\Delta t}(t,x,a)$

No Q-Function in Continuous Time!

- Q-learning works inherently for discrete-time only: Δt is fixed
- Q-function collapses in continuous time when $\Delta t \to 0$ (Tallec et al. 2019)
- Impact of any action a is negligible on $[t, t + \Delta t]$ when $\Delta t \to 0$
- What should be a proper continuous-time counterpart of Q-function?

Continuous Time

- Given a policy $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, define

$$
\begin{aligned}
& Q_{\Delta t}(t, x, a; \boldsymbol{\pi}) \\
:=& \mathbb{E}^{\mathbb{P}}\Big[\int_t^{t+\Delta t} r(s, X_s^a, a)\mathrm{d}s \\
& \quad + \mathbb{E}^{\mathbb{P}}[\int_{t+\Delta t}^{T} [r(s, X_s^{\boldsymbol{\pi}}, a_s^{\boldsymbol{\pi}}) - \gamma \log \boldsymbol{\pi}(a_s^{\boldsymbol{\pi}} | s, X_s^{\boldsymbol{\pi}})]\mathrm{d}s + h(X_T^{\boldsymbol{\pi}}) | X_{t+\Delta t}^a] \Big| X_t^{\boldsymbol{\pi}} = x\Big] \\
=& J(t, x; \boldsymbol{\pi}) + \mathbb{E}^{\mathbb{P}}\Big[\int_t^{t+\Delta t} r(s, X_s^a, a)\mathrm{d}s + J(t+\Delta t, X_{t+\Delta t}^a; \boldsymbol{\pi}) - J(t, x; \boldsymbol{\pi})\Big] \\
=& J(t, x; \boldsymbol{\pi}) + \left[\frac{\partial J}{\partial t}(t, x; \boldsymbol{\pi}) + H\left(t, x, a, \frac{\partial J}{\partial x}(t, x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t, x; \boldsymbol{\pi})\right)\right]\Delta t + o(\Delta t)
\end{aligned}
$$

- Leading term J is independent of a, as expected
- Consider the first-order term instead!

q-Function

Definition (Jia and Z. 2022c)

The q-function associated with a given stochastic policy $\boldsymbol{\pi} \in \boldsymbol{\Pi}$ is defined as

$$q(t, x, a; \boldsymbol{\pi}) = \frac{\partial J}{\partial t}(t, x; \boldsymbol{\pi}) + H\left(t, x, a, \frac{\partial J}{\partial x}(t, x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t, x; \boldsymbol{\pi})\right).$$

Discussions

- q-Function is first-order *derivative* of conventional Q-function in time:

$$q(t, x, a; \boldsymbol{\pi}) = \lim_{\Delta t \to 0} \frac{Q_{\Delta t}(t, x, a; \boldsymbol{\pi}) - J(t, x; \boldsymbol{\pi})}{\Delta t}$$

- A continuous-time notion because *it does not depend on any time-discretization*
- Vital advantage for learning algorithm design as performance of RL algorithms is very sensitive wrt time discretization step (Tallec et al. 2019)
- Policy improvement theorem can now be expressed in terms of q-function:

$$\boldsymbol{\pi}'(\cdot|t, x) \propto \exp\left\{\frac{1}{\gamma} H\big(t, x, \cdot, \frac{\partial J}{\partial x}(t, x; \boldsymbol{\pi}), \frac{\partial^2 J}{\partial x^2}(t, x; \boldsymbol{\pi})\big)\right\} \propto \exp\left\{\frac{1}{\gamma} q(t, x, \cdot; \boldsymbol{\pi})\right\}$$

- Only need to learn q-function $q(\cdot, \cdot, \cdot; \boldsymbol{\pi})$ under any policy $\boldsymbol{\pi}$

Martingale Characterization

Theorem (Jia and Z. 2023)

Let a policy $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, a function $\hat{J} \in C^{1,2}\big([0,T) \times \mathbb{R}^d\big) \cap C\big([0,T] \times \mathbb{R}^d\big)$ and a continuous function $\hat{q} : [0,T] \times \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$ be given satisfying

$$\hat{J}(T,x) = h(x), \ \int_{\mathcal{A}} \big[\hat{q}(t,x,a) - \gamma \log \boldsymbol{\pi}(a|t,x)\big] \boldsymbol{\pi}(a|t,x)\mathrm{d}a = 0, \ \ \forall (t,x).$$

Then $\hat{J}$ and $\hat{q}$ are respectively the value function and the q-function associated with $\boldsymbol{\pi}$ if and only if for all $(t,x) \in [0,T] \times \mathbb{R}^d$, the following process

$$\hat{J}(s, X_s^{\boldsymbol{\pi}}; \boldsymbol{\pi}) + \int_t^s [r(t', X_{t'}^{\boldsymbol{\pi}}, a_{t'}^{\boldsymbol{\pi}}) - \hat{q}(t', X_{t'}^{\boldsymbol{\pi}}, a_{t'}^{\boldsymbol{\pi}})]\mathrm{d}t'$$

is an $(\{\mathcal{F}_s\}_{s \geq 0}, \mathbb{P})$-martingale, where $\{X_s^{\boldsymbol{\pi}}, t \leq s \leq T\}$ is the state process under $\boldsymbol{\pi}$ with $X_t^{\boldsymbol{\pi}} = x$. If it holds further that $\boldsymbol{\pi}(a|t,x) = \frac{\exp\{\frac{1}{\gamma}\hat{q}(t,x,a)\}}{\int_{\mathcal{A}} \exp\{\frac{1}{\gamma}\hat{q}(t,x,a)\}\mathrm{d}a}$, then $\boldsymbol{\pi}$ is the optimal policy and $\hat{J}$ is the optimal value function.

Function Approximation

- *Function approximation*: approximates J and q by parametric families of functions J^θ and q^ψ respectively, where $\theta \in \mathbb{R}^L$ and $\psi \in \mathbb{R}^N$
- Parametric forms may be inspired by problem structure or neural networks

Martingality: Loss Function and SGD Algorithms

- $M_t^{\theta,\psi} = J^\theta(t, X_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) + \int_0^t [r(t', X_{t'}^{\boldsymbol{\pi}}, a_{t'}^{\boldsymbol{\pi}}) - q^\psi(t', X_{t'}^{\boldsymbol{\pi}}, a_{t'}^{\boldsymbol{\pi}})] \mathrm{d}t'$ is martingale
- $M_t^{\theta,\psi} = \mathbb{E}[M_T^{\theta,\psi} | \mathcal{F}_t] = \arg\min_\xi$ is $\mathcal{F}_t$-measurable $\mathbb{E}|M_T^{\theta,\psi} - \xi|^2, \ t \in [0, T]$
- *Martingale loss function*:

$$ML(\theta, \psi) := \frac{1}{2}\mathbb{E}\int_0^T |M_T^{\theta,\psi} - M_t^{\theta,\psi}|^2 \mathrm{d}t \to \text{min.}$$

- However

$$ML(\theta,\psi) \approx \frac{1}{2}\mathbb{E}\Big[\sum_{i=0}^{K-1}\Big(h(X_{t_K}) + \sum_{j=0}^{K-1} r_j \Delta t - J^\theta(t_i, X_{t_i}) - \sum_{j=0}^{i-1}(r_j - q^\psi(t_j, X_{t_j})\Big)^2 \Delta t\Big]$$

- This function only depend on observed data, not functional forms of b, σ, r, h
- Stochastic gradient descent (SGD) algorithm to solve for (θ, ψ)

Martingality: Orthogonality Conditions and SA Algorithms

- In general, $M^{\theta,\psi}$ is a square integrable martingale if and only if $\mathbb{E}\int_0^T \xi_t \mathrm{d}M_t^{\theta,\psi} = 0$ for any $\xi \in L^2_{\mathcal{F}}([0,T]; M^{\theta,\psi})$
- *Martingale orthogonality conditions*
- For numerical approximation methods, we can choose finitely many test functions in special forms
- For example, we can take $\xi_t = (\frac{\partial J^\theta}{\partial \theta}(t, X_t), \frac{\partial q^\psi}{\partial \psi}(t, X_t)) \in \mathbb{R}^{L+N}$
- Use stochastic approximation (SA) algorithms to solve the resulting system of equations to get (θ, ψ)

Actor–Critic Algorithms

- Actor: actions (controls)
- Critic: value (objective) functions
- Actor–critic algorithms: learning and self-improving

$$\boldsymbol{\pi}^n \xrightarrow{\text{q-learning}} (J^n, q^n) \xrightarrow{\text{PI}} \boldsymbol{\pi}^{n+1} \xrightarrow{\text{q-learning}} (J^{n+1}, q^{n+1}) \cdots$$

A q-Learning Algorithm

- Parametrizing $(J(t,x;\boldsymbol{\pi}), q(t,x,a;\boldsymbol{\pi}))$ with $\{(J^\theta(t,x), q^\psi(t,x,a))\}_{\theta,\psi}$
- Initialize with some (θ_1, ψ_1) and a control policy $\boldsymbol{\pi}^1(\cdot \,|\, \cdot, \cdot)$
- For $n \geq 1$:
  1. Update

$$\theta_{n+1} = \theta_n + \alpha_{\theta,n} \int_0^T \frac{\partial J^\theta}{\partial \theta}_{|\theta=\theta_n} (t, X_t^{\boldsymbol{\pi}^n}) G_{t:T}^n dt,$$

$$\psi_{n+1} = \psi_n + \alpha_{\psi,n} \int_0^T \int_t^T e^{-\beta(s-t)} \frac{\partial q^\psi}{\partial \psi}_{|\psi=\psi_n} (s, X_s^{\boldsymbol{\pi}^n}, a_s^{\pi^n}) ds G_{t:T}^n dt$$

  where $G_{t:T}^n := e^{-\beta(T-t)} h(X_T^{\boldsymbol{\pi}^n}) - J^{\theta_n}(t, X_t^{\boldsymbol{\pi}^n}) + \int_t^T e^{-\beta(s-t)} [r(s, X_s^{\boldsymbol{\pi}^n}, a_s^{\boldsymbol{\pi}^n}) - q^{\psi_n}(s, X_s^{\boldsymbol{\pi}^n}, a_s^{\boldsymbol{\pi}^n})] ds$

  2. Sample

$$\boldsymbol{\pi}^{n+1}(\cdot \,|\, t, x) \propto \exp\left(\frac{1}{\gamma} q^{\phi_{n+1}}(t, x, \cdot)\right)$$

Regret Bound

Theorem (Tang and Z. 2024)

Assume $\sigma(t,x,a) = \sigma(t,x)$ *and some technical conditions. Set* $\alpha_{\theta,n}, \alpha_{\psi,n} = \frac{A}{n+B}$ *for some constants* $A > 0$ *and* $B > 0$*, and let* $\varepsilon > 0$*. Then there exists* $C > 0$ *(depending on* γ *but not on* n, ε*) such that with probability* $1 - \varepsilon$*, the regret is*

$$\sum_{k=1}^{n} |V(t,x) - J(t,x;\boldsymbol{\pi}^k)| \leq \frac{C}{\varepsilon^{1/2}} n^{\frac{3}{4}} (\ln n)^{\frac{1}{2}}.$$

Model-Based vs Model-Free

- Data used
  - Model-based: exogenous
  - Model-free: both exogenous and endogenous
- What to learn
  - Model-based: the model
  - Model-free: the optimal strategy
- How to achieve optimality
  - Model-based: compare with others
  - Model-free: compare with selves

What Do We Need To Learn About Environment?

- Classical model-based approach: separates "estimation" and "optimization"
- Model-free RL approach: skips estimating a model and learns optimizing policies directly via PG or Q/q-learning
- But RL still learns *something* about the environment: q-function or Hamiltonian
- It is the Hamiltonian, rather than each and every individual model coefficient, that needs to be learned/estimated for optimization
- From a pure computational standpoint, estimating a single function is much more efficient and robust than estimating multiple functions (b, σ, r, h) in terms of avoiding or reducing over-parameterization, sensitivity to errors and accumulation of errors

Why Is q-Function Learnable?

- Itô's formula

$$q(t, X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}}; \boldsymbol{\pi})\mathrm{d}t = \mathrm{d}J(t, X_t^{\boldsymbol{\pi}}; \boldsymbol{\pi}) + r(t, X_t^{\boldsymbol{\pi}}, a_t^{\boldsymbol{\pi}})\mathrm{d}t + \{\cdots\}\mathrm{d}W_t.$$

- So q-function can be learned through temporal differences of the value function; hence the task of learning and optimizing can be accomplished in a data-driven way
- This would not be the case if we chose to learn individual model coefficients separately

Finally ...

- There are fundamental theoretical questions in machine learning that beg for answers
- Answering them often calls for fundamentally different thinking out of our comfort zone
- The mathematical techniques employed may still well be within our comfort zone (stochastic analysis, stochastic control, differential equations, etc.)