Premium control with reinforcement learning

Lina Palmborg, Stockholm University

based on joint work with F. Lindskog

October 19, 2022

Premium control problem for a mutual insurer

- Premium control problem in discrete time
- Mutual non-life insurer
  - Claim costs not known when premium is set (delays)
  - Premium level affects whether the company attracts or loses customers (feedback)
- Aim: find a premium rule that generates
  - a low premium
  - a premium that does not fluctuate too much over time
  - a premium that leads to a low probability of default
- Inspired by Martin-Löf (1983), (1994).

Model of the insurance company

- Surplus fund

$$G_{t+1} = G_t + \mathsf{EP}_{t+1} + \mathsf{IE}_{t+1} - \mathsf{OE}_{t+1} - \mathsf{IC}_{t+1} + \mathsf{RP}_{t+1}$$

- Earned premium

$$\mathsf{EP}_{t+1} = \frac{1}{2}(P_t N_{t+1} + P_{t-1} N_t)$$

Policy/premium rule

- A policy (premium rule) π determines the premium charged in state S_t
  - Deterministic policy: $P_t = \pi(S_t)$
  - Stochastic policy: $\pi(p|s) = \mathrm{P}(P_t = p \mid S_t = s)$
- Define the state S_t so that the system (S_t) evolves in a Markovian manner given the policy π, e.g.

$$S_t = (G_t, P_{t-1}, N_t, \ldots).$$

Markov decision process (MDP)

- $\mathcal{S}$ set of (non-terminal) states
- $\mathcal{A}$ set of actions (premium levels)
- $f(a, s, s')$ cost when taking action a in state s and transitioning to state s'
- $p(s' \mid s, a)$ probability of transitioning from state s to state s' after taking action a

Given this MDP, we want to find an optimal policy (premium rule).

The control problem

$$\underset{\pi}{\text{minimise}}\ \mathbb{E}_\pi\Big[\sum_{t=0}^{T}\gamma^t f(P_t, S_t, S_{t+1}) \mid S_0 = s\Big],$$

where γ is the discount factor.

$$f(P_t, S_t, S_{t+1}) := \begin{cases} c(P_t), & \text{if } G_{t+1} \geq G_{\min}, \\ c(\max \mathcal{A})(1+\eta), & \text{if } G_{t+1} < G_{\min}, \end{cases}$$

- c an increasing, strictly convex function $\implies$ premiums (P_t) will be averaged
- $T := \min\{t : G_t < G_{\min}\}$ $\implies$ termination (default)
- $\eta > 0$ $\implies$ high cost in case of default

Value functions

- Value function

$$v_\pi(s) := \mathbb{E}_\pi\Big[\sum_{t=0}^{T} \gamma^t(-f(P_t, S_t, S_{t+1})) \mid S_0 = s\Big]$$

- Bellman equation

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi\big[-f(P_0, S_0, S_1) + \gamma v_\pi(S_1) \mid S_0 = s\big] \\ &= \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}} p(s'|s,a)\big(-f(a,s,s') + \gamma v_\pi(s')\big) \end{aligned}$$

- Action-value function

$$\begin{aligned} q_\pi(s,a) &:= \mathbb{E}_\pi\Big[\sum_{t=0}^{T} \gamma^t(-f(P_t, S_t, S_{t+1})) \mid S_0 = s, P_0 = a\Big] \\ &= \mathbb{E}\big[-f(P_0, S_0, S_1) + \gamma v_\pi(S_1) \mid S_0 = s, P_0 = a\big] \end{aligned}$$

"charge premium a in state s, then follow premium rule π"

Optimal value functions and optimal policy

- Optimal value function

$$v_*(s) = \max_\pi v_\pi(s)$$

We want to find an optimal policy π_*, i.e. $v_{\pi_*}(s) = v_*(s)$

- Optimal action-value function

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

"charge premium a in state s, then follow an optimal premium rule"

- Bellman optimality equation

$$v_*(s) = \max_{a\in\mathcal{A}} q_*(s,a) = \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} p(s'|s,a)\big(- f(a,s,s') + \gamma v_*(s')\big)$$

- Optimal policy

$$\pi_*(s) = \operatorname*{argmax}_{a\in\mathcal{A}} q_*(s,a)$$

Policy iteration

If state space is not too large, and the transition probabilities are explicitly known $\implies$ use Bellman equation (e.g. policy iteration)

Let $k = 0$, and choose some initial deterministic policy π_0.

(i) Determine $V_k(s)$ as the unique solution to the system of equations

$$V_k(s) = \sum_{s' \in \mathcal{S}} p(s'|s, \pi_k(s)) \Big(- f(\pi_k(s), s, s') + \gamma V_k(s') \Big).$$

(ii) Determine an improved policy $\pi_{k+1}(s)$ by computing

$$\pi_{k+1}(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \sum_{s' \in \mathcal{S}} p(s'|s, a) \Big(- f(a, s, s') + \gamma V_k(s') \Big).$$

(iii) If $\pi_{k+1}(s) \neq \pi_k(s)$ for some $s \in \mathcal{S}$, then increase k by 1 and return to step (i).

Temporal difference (TD) control algorithms

Transition probabilities not explicitly known $\implies$ TD control algorithm (e.g. SARSA, Q-learning)

- These algorithms directly estimate $q_*(s, a)$
- Remember: $\pi_*(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, q_*(s, a)$
- Based on real or simulated data (without needing explicit expressions for transition probabilities)
- Still need to store $q_*(s, a)$ for each (s, a) $\implies$ state space cannot be too large

SARSA

Given some policy π that generates actions we can sample $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$

- A_t action taken at time step t (here P_t)
- R_{t+1} reward after taking action A_t in state S_t and transitioning to state S_{t+1} (here $-f(P_t, S_t, S_{t+1})$)

The iterative update for the estimated action-value function

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha_t \big(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\big),$$

where α_t is a step size parameter.

- Intuition: $R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})$ is a slightly better estimate of $q_\pi(S_t, A_t)$ than $Q(S_t, A_t)$

Exploration/exploitation

For SARSA to converge to the optimal action-value function the policy π that generates actions needs to

- be exploratory, i.e. it needs to keep trying different actions that might not currently be optimal
- exploit what has been experienced previously, by progressively choosing the actions that appear to be optimal

Examples:

- ε-greedy policy
- Softmax policy

Exploration/exploitation

- Greedy policy (deterministic)

$$\pi(s) = \underset{a}{\operatorname{argmax}}\, Q(s, a)$$

- ε-greedy policy (stochastic)

$$\pi(a|s) = \begin{cases} 1 - \varepsilon, & \text{if } a = \underset{a}{\operatorname{argmax}}\, Q(s, a), \\ \dfrac{\varepsilon}{|\mathcal{A}| - 1}, & \text{otherwise.} \end{cases}$$

- Softmax policy (stochastic)

$$\pi(a|s) = \frac{\exp\{Q(s, a)/\tau\}}{\sum_{\bar{a} \in \mathcal{A}} \exp\{Q(s, \bar{a})/\tau\}}$$

SARSA with function approximation

State space large $\implies$ SARSA with function approximation

- Action-value function $q_\pi(s,a)$ is approximated by a parameterised function $\hat{q}(s,a;w)$
- Want to find weight vector w that minimises

$$\frac{1}{2}\sum_{s,a}\mu(s,a)\big(q_\pi(s,a)-\hat{q}(s,a;w)\big)^2,$$

  where $\mu(s,a)$ is the fraction of time spent in state-action pair (s,a)
- Stochastic gradient descent:

$$w_{t+1}=w_t+\alpha_t\big(q_\pi(S_t,A_t)-\hat{q}(S_t,A_t;w_t)\big)\nabla\hat{q}(S_t,A_t;w_t)$$

- Problem: q_π is unknown!

SARSA with function approximation

- Idea: As in standard SARSA, replace $q_\pi(S_t, A_t)$ with

$$R_{t+1} + \gamma \hat{q}(S_{t+1}, A_{t+1}; w_t)$$

(a slightly better estimate of $q_\pi(S_t, A_t)$ than $\hat{q}(S_t, A_t; w_t)$).

- Iterative update for the weight vector

$$w_{t+1} = w_t + \alpha_t \big(R_{t+1} + \gamma \hat{q}(S_{t+1}, A_{t+1}; w_t) - \hat{q}(S_t, A_t; w_t)\big) \nabla \hat{q}(S_t, A_t; w_t)$$

- We use linear function approximation,

$$\hat{q}(s, a; w) := w^\top x(s, a), \quad \nabla \hat{q}(s, a; w) = x(s, a)$$

where $x(s, a)$ are basis functions.

Solving the control problem

- Goal: find optimal premium rule (low, stable premium, low probability of default)
- Cannot fully specify the transition probabilities of the MDP in a realistic setting
  $\Longrightarrow$ need to use reinforcement learning (e.g. SARSA)
- State space too large in a realistic setting
  $\Longrightarrow$ need to use function approximation
- SARSA learns from real or simulated experience
  $\Longrightarrow$ need a lot of data!
  $\Longrightarrow$ simulate data from a suitable stochastic environment

Solving the control problem

How do we determine if the function approximation chosen is appropriate?

- Start with a simple model of the environment (MDP fully specified, state space not too large)
- Solve with classical methods
  $\Longrightarrow$ ”true” optimal premium rule
- Solve with SARSA with function approximation
  $\Longrightarrow$ approximate optimal premium rule
- If approximate optimal premium rule approximates the ”true” optimal premium rule well, then solve the problem using a more realistic model of the environment

Model of the insurance company

- Surplus fund

$$G_{t+1} = G_t + \mathsf{EP}_{t+1} + \mathsf{IE}_{t+1} - \mathsf{OE}_{t+1} - \mathsf{IC}_{t+1} + \mathsf{RP}_{t+1}$$

- Earned premium

$$\mathsf{EP}_{t+1} = \frac{1}{2}(P_t N_{t+1} + P_{t-1} N_t)$$

Simplified model

- Number of contracts

$$N_{t+1} := N \quad \text{for all } t \quad \text{(non-random)}.$$

- Operating expenses

$$\mathsf{OE}_{t+1} = \beta_0 + \beta_1 N.$$

- Investment earnings

  $\mathcal{L}(\mathsf{IE}_{t+1} + G_t \mid \mathcal{F}_t, G_t > 0)$ is negative binomial,

$$\mathbb{E}[\mathsf{IE}_{t+1} + G_t \mid \mathcal{F}_t, G_t > 0] = (1+\xi)G_t,$$

$$\mathrm{Var}(\mathsf{IE}_{t+1} + G_t \mid \mathcal{F}_t, G_t > 0) = \frac{1+\xi+\nu}{\nu}(1+\xi)G_t.$$

Simplified model

- Paid claims

$$\mathcal{L}(\mathsf{PC}_{t+1} \mid \mathcal{F}_t) = \mathsf{Pois}(\mu N).$$

- Incurred claims and runoff profit

$$\mathsf{IC}_{t+1} - \mathsf{RP}_{t+1} = \mathsf{PC}_{t+1}.$$

- $\implies$ we define the state as $S_t = (G_t, P_{t-1})$.

Results - simplified model, N_t non-random

Figure: Policy iteration

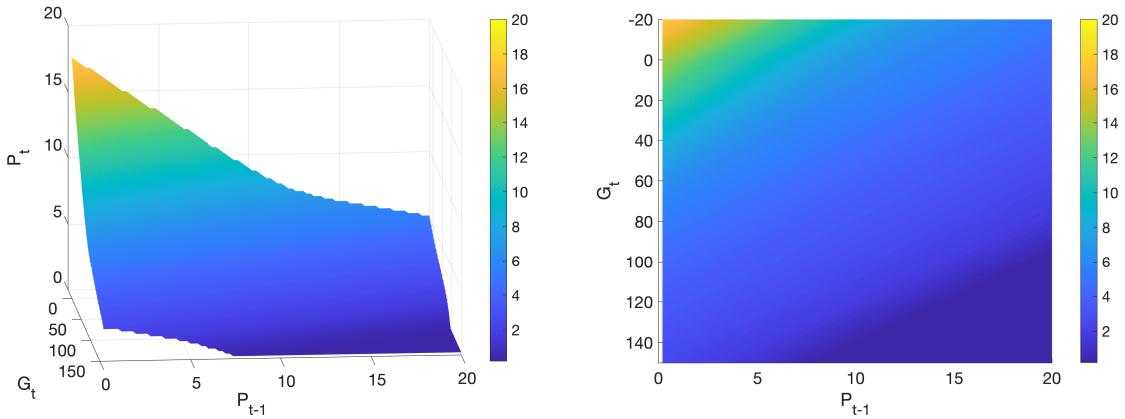
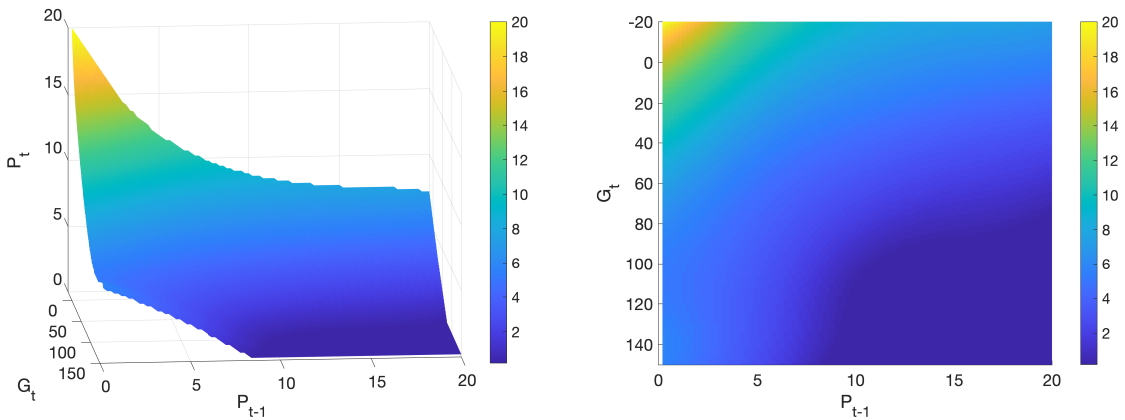


Figure: Function approximation



Results - simplified model, N_t non-random

	Expected reward
Policy iteration	**-85.91**
Q-learning	-86.50
Fourier 3 with softmax policy	**-86.11**
Fourier 2 with softmax policy	-86.30
Fourier 1 with softmax policy	-86.59
Fourier 3 with ε-greedy policy	-92.74
Best constant policy	-122.70
Myopic policy with terminal state, $p_{\min} = 0.2$	-97.06
Myopic policy with terminal state, $p_{\min} = 5.8$	-90.40
Myopic policy with constraint, $p_{\min} = 0.2$	-121.52
Myopic policy with constraint, $p_{\min} = 6.4$	-93.58

Table: Expected discounted total reward based on simulation (uniformly distributed starting states).

More realistic setting

- Number of contracts

$$\mathcal{L}(N_{t+1} \mid \mathcal{F}_t) = \text{Pois}(aP_t^b), \quad a > 0, b < 0.$$

- Operating expenses

$$\text{OE}_{t+1} = \beta_0 + \beta_1 \widetilde{N}_{t+1}, \quad \widetilde{N}_{t+1} = (N_{t+1} + N_t)/2.$$

- Investment earnings as before,
  $\mathcal{L}(\text{IE}_{t+1} + G_t \mid \mathcal{F}_t, G_t > 0)$ is negative binomial.

More realistic setting

- Paid claims

$$\mathcal{L}(\mathsf{PC}_{t+1} \mid \mathcal{F}_t, \widetilde{N}_{t+1}) = \mathsf{Pois}(\alpha_1 \mu \widetilde{N}_{t+1} + \alpha_2 \mu \widetilde{N}_t),$$

  where $\alpha_1, \alpha_2 \in [0, 1]$ with $\alpha_1 + \alpha_2 = 1$.
- Incurred claims and runoff profit

$$\mathsf{IC}_{t+1} - \mathsf{RP}_{t+1} = \mathsf{PC}_{t+1} + \alpha_2 \mu \widetilde{N}_{t+1} - \alpha_2 \mu \widetilde{N}_t.$$

- $\implies$ we define the state at time t as

$$S_t = (G_t, P_{t-1}, N_{t-1}, N_t)$$

Results - more realistic setting

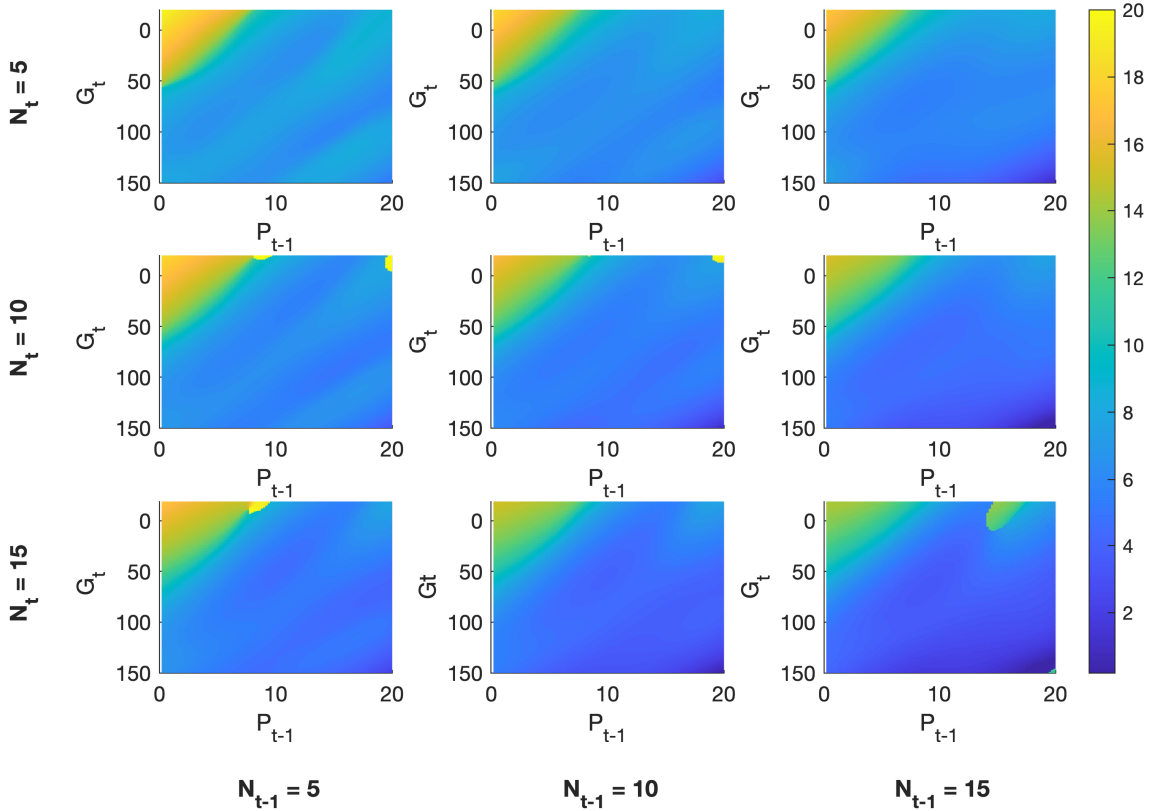


Figure: Optimal policy in more realistic setting with terminal state using linear function approximation with 3rd order Fourier basis, for $N_t, N_{t-1} \in \{5, 10, 15\}$.

Results - more realistic setting

	Expected reward
Fourier 3	**-97.17**
Fourier 2	-104.41
Fourier 1	-128.83
Policy from simplified model	-116.70
Myopic policy with constraint, $p_{\min} = 0.2$	-360.69
Myopic policy with constraint, $p_{\min} = 6.4$	-100.92
Best constant policy	-131.85

Table: Expected discounted total reward based on simulation (uniformly distributed starting states).

Results - more realistic setting

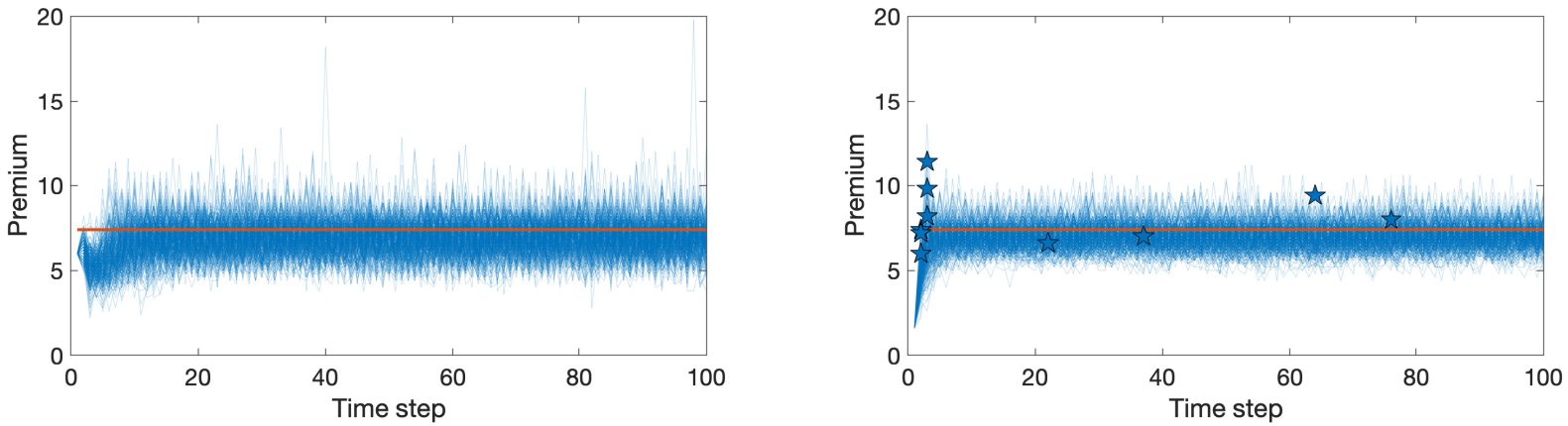


Figure: Simulated trajectories using policy with 3rd order Fourier basis (left) or policy from the simplified model (right). Starting state $S_0 = (100, 15, 5, 5)$.
The red line shows the best constant policy. Each star indicates at least one termination at that time step.

Conclusion

- Reinforcement learning techniques enable us to solve more realistic premium control problems
- ”Model free”, i.e. not specific to the stochastic model of the insurance company used here
- $\Longrightarrow$ optimal premium rules that can be used in practice

However...

- Reinforcement learning method needs to be carefully implemented
- Cost function (”reward signal”) needs to be designed to ensure that the objective of the specific insurer is met

Main references

- Preprint available at SSRN: `https://papers.ssrn.com/abstract=4156450`
- A. Martin-Löf (1983), Premium control in an insurance system, an approach using linear control theory. *Scandinavian Actuarial Journal*, 1983(1):1–27.
- A. Martin-Löf (1994), Lectures on the use of control theory in insurance. *Scandinavian Actuarial Journal*, 1994(1):1–25.
- R. S. Sutton and A. G. Barto (2018), *Reinforcement learning: An introduction*. MIT press.
- D. P. Bertsekas and J. N. Tsitsiklis (1996), *Neuro-dynamic programming*. Athena Scientific.