Simulation-based Approximate Policy Iteration for Generalized Semi-Markov Decision Processes

Emmanuel Rachelson [1]

Patrick Fabiani [1]
Frédérick Garcia [2]

[1]ONERA-DCSD

[2]INRA-BIA
Toulouse, France

ECAI08, July 23rd, 2008

Plan

Plan

Planning under uncertainty with time dependency.
$\longrightarrow$ planning to coordinate with an uncertain and unstationnary environment.

Planning under uncertainty with time dependency.
$\longrightarrow$ planning to coordinate with an uncertain and unstationnary environment.

Should we open more lines ?

Planning under uncertainty with time dependency.
$\longrightarrow$ planning to coordinate with an uncertain and unstationnary environment.

Airplanes taxiing management

Planning under uncertainty with time dependency.
$\longrightarrow$ planning to coordinate with an uncertain and unstationnary environment.

Onboard planning for coordination

Planning under uncertainty with time dependency.
$\longrightarrow$ planning to coordinate with an uncertain and unstationnary environment.

Adding or removing trains ?

Main idea

Why is writing an MDP for the previous problems such a difficult task ?

"Lots of things occur in parallel"

- concurrent phenomena
- partially controlable dynamics

Main idea

Why is writing an MDP for the previous problems such a difficult task ?

"Lots of things occur in parallel"

- concurrent phenomena
- partially controlable dynamics

Main idea

Why is writing an MDP for the previous problems such a difficult task ?

"Lots of things occur in parallel"

- concurrent phenomena
- partially controlable dynamics

Typical features

- Continuous time
- Hybrid state spaces
- Large state spaces
- Total reward criteria
- Long trajectories / long episodes

How do we model all this ?

GSMDP, (*Younes et al., 04*)

GSMDP, (*Younes et al., 04*)

GSMP, (*Glynn, 89*)
Several semi-Markov processes affecting the same state space

One process conditionned by the choice of the action undertaken

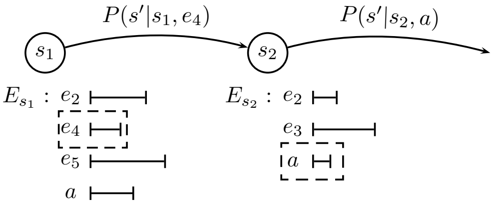
$$\rightarrow \langle S, E, A, P, F, r \rangle$$

GSMDP, (Younes et al., 04)

GSMP, (*Glynn, 89*)
Several semi-Markov processes affecting the same state space

One process conditionned by the choice of the action undertaken

$$\rightarrow \langle S, E, A, P, F, r \rangle$$



Controling GSMDP

non-Markov behaviour !

$\longrightarrow$ no guarantee of an optimal Markov policy

(*Younes et al., 04*): approximate your model with phase-type (exponential) distributions.

Supplementary variables technique (*Nilsen, 98*). Large dimension state spaces.

Our approach: no hypothesis, simulation-based API.

Controling GSMDP

non-Markov behaviour !

$\longrightarrow$ no guarantee of an optimal Markov policy

(*Younes et al., 04*): approximate your model with phase-type (exponential) distributions.

Supplementary variables technique (*Nilsen, 98*).
Large dimension state spaces.

Our approach: no hypothesis, simulation-based API.

Controling GSMDP

non-Markov behaviour !

$\longrightarrow$ no guarantee of an optimal Markov policy

(*Younes et al., 04*): approximate your model with phase-type (exponential) distributions.

Supplementary variables technique (*Nilsen, 98*).
Large dimension state spaces.

Our approach: no hypothesis, simulation-based API.

Plan

1. Time and MDP: motivation and modeling
   - Examples
   - Problem features
   - GSMDP
2. Focusing Policy search in Policy Iteration
   - Policy Iteration algorithms
   - Asynchronous Dynamic Programming
   - Real-Time Policy Iteration
3. Dealing with large dimension, continuous state spaces
   - RL, Monte-Carlo sampling and Statistical Learning
   - The ATPI algorithm (naive version)
   - The ATPI algorithm (complete version)

Policy Iteration

Policy evaluation: V^{π_n}

One-step improvement: π_{n+1}

Policy Iteration

- performs search in policy space
- converges in less iterations than VI
- takes longer than VI

Policy Iteration

Approximate evaluation: V^{π_n}

One-step improvement: π_{n+1}

Asynchronous Dynamic Programming

Bellman backups can be performed in any order, the algorithm eventually reaches the optimal policy.

Example

Asynchronous Value Iteration

$$V_{n+1}(s) \leftarrow \max_{a\in A} r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)V_n(s')$$

Some states can be updated several times before some others are updated for the first time.

Asynchronous Dynamic Programming

Bellman backups can be performed in any order, the algorithm eventually reaches the optimal policy.

Example

Asynchronous Policy Iteration

$$\pi_{n+1}(s) \leftarrow \arg\max_{a\in A} r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a) V^{\pi_n}(s')$$

We can choose to update only some states before entering a new evaluation of π step.

(*Barto et al., 95)* Learning to act using real-time dynamic programming.

RTDP

Asynchronous VI with heuristic guidance.
Updated states at step $n+1$ = states visited by the one-step lookahead greedy policy w.r.t V_n.

Is there an equivalent for policy iteration ?
We introduce:

RTPI

At iteration $n+1$, updated states are states visited by the one-step lookahead greedy policy w.r.t V^{π_n}. *ie.* states visited by the application of π_{n+1}.

Practical motivation for RTPI

Motivation: don't want / can't improve the policy everywhere

- too time/resource consuming
- not useful with regard to 'relevant' information gathered

Useful ? Interesting ? Relevant ?
$\longrightarrow$ "Improving the policy in the situations I am likely to encounter today"

In other words ...

Which subset of states for Asynchronous PI ?

The ones visited by policy simulation.

Practical motivation for RTPI

Motivation: don't want / can't improve the policy everywhere

- too time/resource consuming
- not useful with regard to 'relevant' information gathered

Useful ? Interesting ? Relevant ?
$\longrightarrow$ "Improving the policy in the situations I am likely to encounter today"

In other words . . .

Which subset of states for Asynchronous PI ?

The ones visited by policy simulation.

Practical motivation for RTPI

Motivation: don't want / can't improve the policy everywhere

- too time/resource consuming
- not useful with regard to 'relevant' information gathered

Useful ? Interesting ? Relevant ?
$\longrightarrow$ "Improving the policy in the situations I am likely to encounter today"

In other words . . .

Which subset of states for Asynchronous PI ?

The ones visited by policy simulation.

Practical motivation for RTPI

Motivation: don't want / can't improve the policy everywhere

- too time/resource consuming
- not useful with regard to 'relevant' information gathered

Useful ? Interesting ? Relevant ?
$\longrightarrow$ "Improving the policy in the situations I am likely to encounter today"

In other words . . .
Which subset of states for Asynchronous PI ?
The ones visited by policy simulation.

Plan

Simulation-based policy evaluation

Our hypothesis: we have a generative model of the process.
$\rightarrow$ (Monte-Carlo) simulation-based policy evaluation.

Statistical learning

Simulating the policy
$\Leftrightarrow$ Drawing a set of *trajectories*
$\Leftrightarrow$ Finite set of realisations of r.v. $R^{\pi}(s)$

We need to

- abstract (*generalize*) local information from samples
- *compactly* store previous knowledge of $V^{\pi}(s) = E(R^{\pi}(s))$.

Simulation-based policy evaluation

Our hypothesis: we have a generative model of the process.
$\longrightarrow$ (Monte-Carlo) simulation-based policy evaluation.

Statistical learning

Simulating the policy
$\Leftrightarrow$ Drawing a set of *trajectories*
$\Leftrightarrow$ Finite set of realisations of r.v. $R^{\pi}(s)$

We need to

- abstract (*generalize*) local information from samples
- *compactly* store previous knowledge of $V^{\pi}(s) = E(R^{\pi}(s))$.

Regression for RL

Reminder:

Approximate evaluation: V^{π_n}

One-step improvement: π_{n+1}

(nearest neighbours, SVR, kLASSO, LWPR)

ATPI

RTPI algorithm on continuous variables with simulation-based policy evaluation + regression.

main:
Input : π_0 or $\tilde{V}_0$, s_0
loop
  $TrainingSet \leftarrow \emptyset$
  for $i = 1$ to N_{sim} **do**
    $\{(s,v)\} \leftarrow \text{simulate}(\tilde{V}, s_0)$
    $TrainingSet \leftarrow TrainingSet \cup \{(s,v)\}$
  end for
  $\tilde{V} \leftarrow \text{TrainApproximator}(TrainingSet)$
end loop

ATPI

RTPI algorithm on continuous variables with simulation-based policy evaluation + regression.

simulate$(\tilde{V}, s_0)$**:**
$ExecutionPath \leftarrow \emptyset$
$s \leftarrow s_0$
while horizon not reached **do**
  $action \leftarrow \text{ComputePolicy}(s, \tilde{V})$
  $(s', r) \leftarrow \text{GSMDPstep}(s, action)$
  $ExecutionPath \leftarrow ExecutionPath \cup (s', r)$
end while
convert execution path to $\{(s, v)\}$
return $\{(s, v)\}$

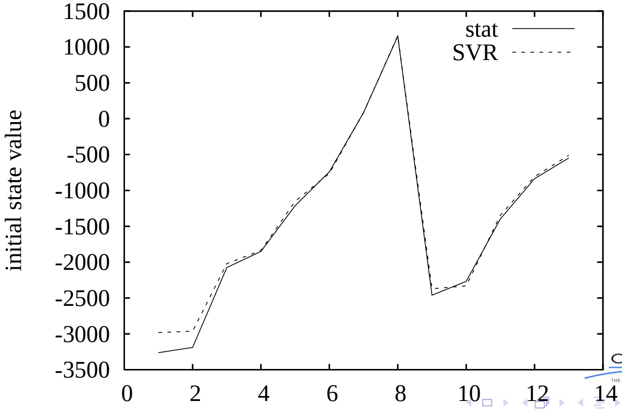
ATPI

RTPI algorithm on continuous variables with simulation-based policy evaluation + regression.

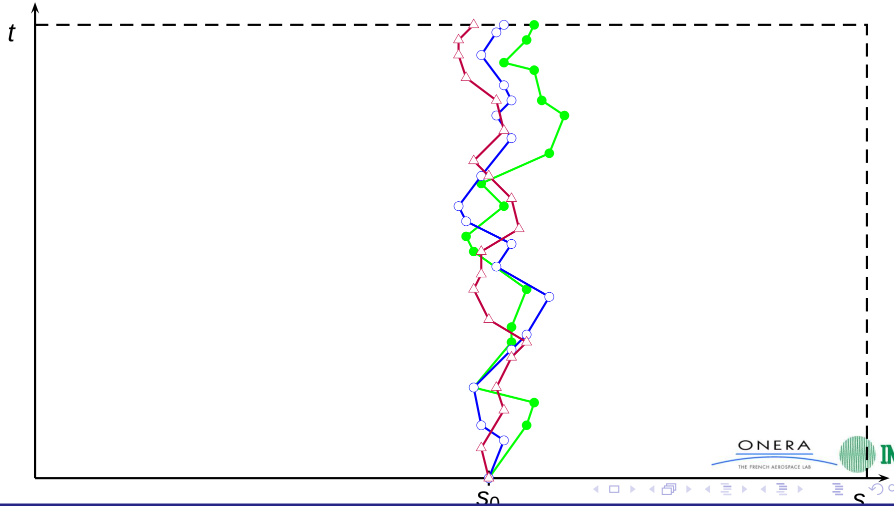
ComputePolicy$(s,\tilde{V})$**:**
for $a \in A$ **do**
  $\tilde{Q}(s,a) = 0$
  for $j = 1$ to $N_{samples}$ **do**
    $(s',r) \leftarrow \text{GSMDPstep}(s,a)$
    $\tilde{Q}(s,a) \leftarrow \tilde{Q}(s,a) + r + \gamma^{t'-t}\tilde{V}(s')$
  end for
  $\tilde{Q}(s,a) \leftarrow \frac{1}{N_{samples}}\tilde{Q}(s,a)$
end for
$action \leftarrow \arg\max_{a \in A} \tilde{Q}(s,a)$
return $action$

Subway problem results

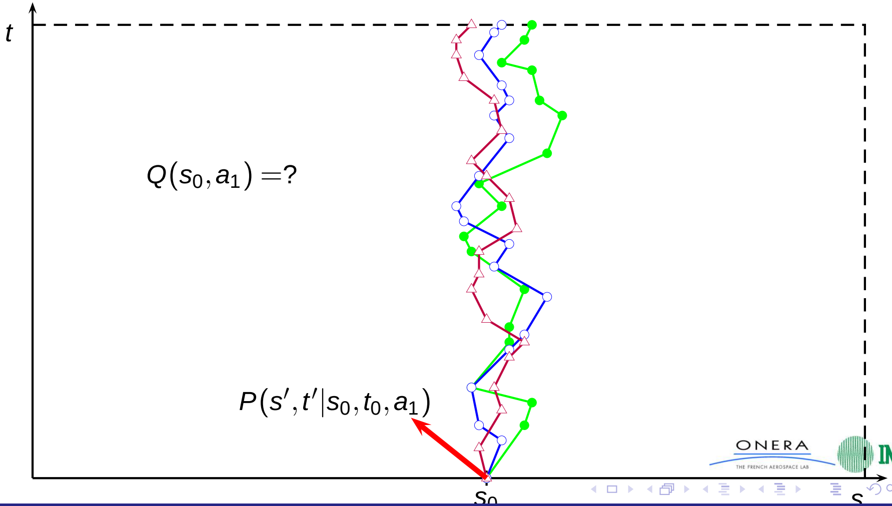
Initial version of online-ATPI with SVR.
Initial policy sets trains to run all day long.



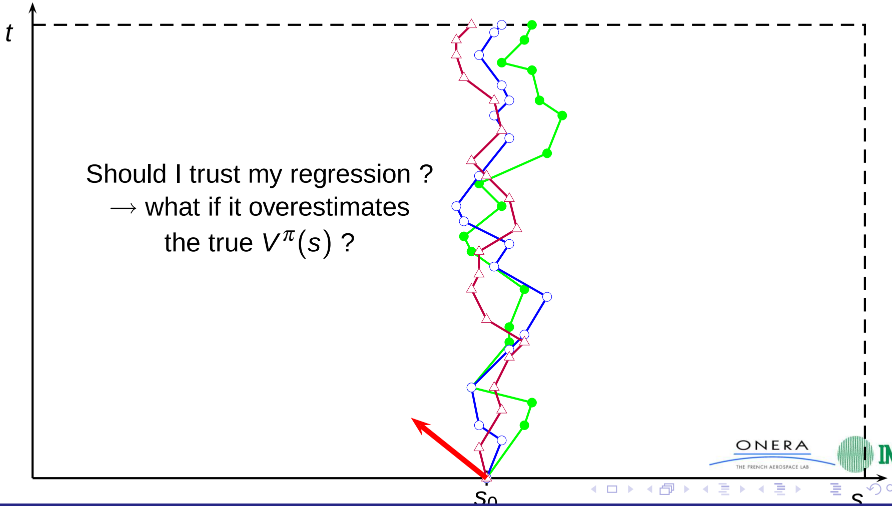
Is there anybody out there ?



Is there anybody out there ?

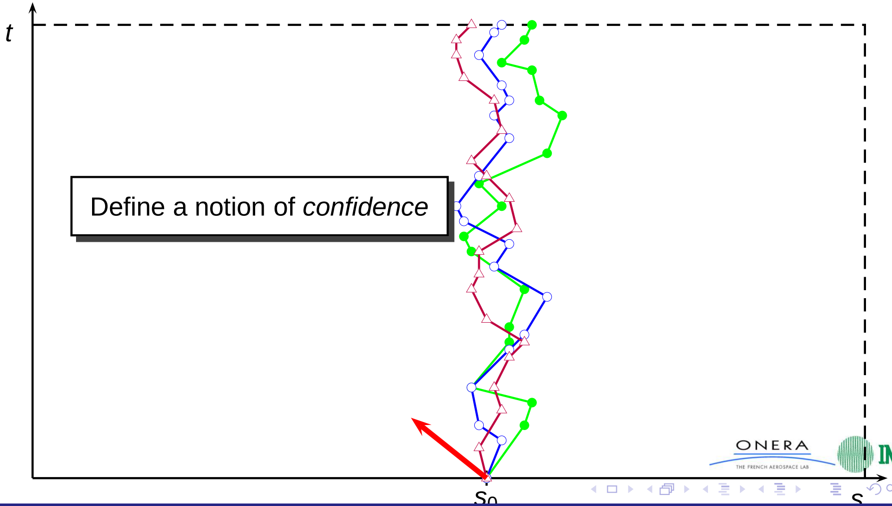


Is there anybody out there ?



Should I trust my regression ?
$\longrightarrow$ what if it overestimates
the true $V^\pi(s)$?

Is there anybody out there ?



Introducing confidence

- "confidence" $\Leftrightarrow$ having enough points around s
  $\Leftrightarrow$ approaching the sufficient statistics for $V^\pi(s)$
  $\longrightarrow$ approx. measure: pdf of the underlying process.
- What should we do if we are not confident ?
  $\longrightarrow$ generate data – increase the samples' density – simulate
- Storing the policy ?

Same problem for policy storage than for value function:
(*Lagoudakis et al., 03*) RL as Classification.

Full statistical learning problem:
(local incremental) regression (V^π), classification (π),
density estimation (*conf*)

Introducing confidence

- "confidence" $\Leftrightarrow$ having enough points around s
  $\Leftrightarrow$ approaching the sufficient statistics for $V^{\pi}(s)$
  $\rightarrow$ approx. measure: pdf of the underlying process.
- What should we do if we are not confident ?
  $\rightarrow$ generate data – increase the samples' density – simulate
- Storing the policy ?

Same problem for policy storage than for value function:
(*Lagoudakis et al., 03)* RL as Classification.

Full statistical learning problem:
(local incremental) regression (V^{π}), classification (π),
density estimation (*conf*)

Introducing confidence

- "confidence" $\Leftrightarrow$ having enough points around s
  $\Leftrightarrow$ approaching the sufficient statistics for $V^{\pi}(s)$
  $\longrightarrow$ approx. measure: pdf of the underlying process.
- What should we do if we are not confident ?
  $\longrightarrow$ generate data – increase the samples' density – simulate
- Storing the policy ?

Same problem for policy storage than for value function:
(*Lagoudakis et al., 03)* RL as Classification.

Full statistical learning problem:
(local incremental) regression (V^{π}), classification (π),
density estimation (*conf*)

Introducing confidence

- "confidence" $\Leftrightarrow$ having enough points around s
  $\Leftrightarrow$ approaching the sufficient statistics for $V^{\pi}(s)$
  $\longrightarrow$ approx. measure: pdf of the underlying process.
- What should we do if we are not confident ?
  $\longrightarrow$ generate data – increase the samples' density – simulate
- Storing the policy ?

Same problem for policy storage than for value function:
(*Lagoudakis et al., 03)* RL as Classification.

Full statistical learning problem:
(local incremental) regression (V^{π}), classification (π),
density estimation (*conf*)

ATPI - complete version

ATPI

samples $\leftarrow \emptyset$
for $i = 1$ to N_{sim} **do**
  while $t < horizon$ **do**
    estimate Q-values
    $s' \leftarrow$ apply best action
    store (s, a, r, s') in *samples*
  end while
end for
$train\tilde{V}^{\pi}(samples)$
$train\tilde{\pi}(samples)$

ATPI - complete version

estimate $Q(s,a)$

$\tilde{Q}(s,a) \leftarrow 0$

for $i = 1$ to N_a **do**

  $(r,s') \leftarrow$ pick next state

  if $confidence(s') = true$ **then**

    $\tilde{Q}(s,a) \leftarrow \tilde{Q}(s,a) + \frac{r+\tilde{V}^{\pi}(s')}{N_a}$

  else

    $data = simulate(\pi, s')$

    $retrain \tilde{V}^{\pi}(data)$

    $\tilde{Q}(s,a) \leftarrow \tilde{Q}(s,a) + \frac{r+\tilde{V}^{\pi}(s')}{N_a}$

  end if

end for

return $\tilde{Q}(s,a)$

Conclusion

GSMDP Modeling of large scale temporal problems of decision under uncertainty.

RTPI Introduction of a new asynchronous PI method performing partial and incremental state space exploration guided by simulation / local policy improvement.

ATPI Design of a RTPI algorithm for continuous, high dimensional state spaces, exploiting the properties of the time variable and bringing together results from:

- discrete events simulation
- simulation-based policy evaluation
- approximate asynchronous policy iteration
- statistical learning

GiSMoP C++ library

$\longrightarrow$ http://emmanuel.rachelson.free.fr/fr/gismop.html

Future work

RTPI Independent algorithm

- study convergence
- compare with RTDP

ATPI Algorithm improvement and testing

- Even non-parametric methods need some tuning ! (currently: LWPR / MC-SVM / OC-SVM)
- error bounds for API
- other benchmarks

Thank you for your attention !