Reinforcement learning with raw image pixels as input state

Damien Ernst[†], *Raphaël Marée*, Louis Wehenkel

Department of Electrical Engineering and Computer Science
University of Liège, Belgium
[†] Postdoctoral Researcher FNRS

IWICPAS - August, 2006

What is reinforcement learning ?

Reinforcement learning = learning what to do, how to map states to actions, from the information acquired from interaction with a system.

Classical setting for reinforcement learning:

- the reinforcement learning agent wants to minimize a long term cost signal
- the information the reinforcement learning agent has is a set of samples
- a sample = (state, action taken while being in this state, instantaneous cost, successor state)

Reinforcement learning is a promising approach for designing autonomous robots able to fulfill specific tasks (helping disabled persons, cleaning a house, playing soccer, ...)

Reinforcement learning and visual input

In many practical problems: the input state is made of visual percepts

A visual percept is composed of hundreds if not thousands of elements $\Rightarrow$ may be problematic if used as such as input space.

Up to now, people believed that it was not possible to use components describing the image as such in a reinforcement learning algorithm $\Rightarrow$ feature extraction techniques.

BUT, two new elements:

- Recent advances in image classification = possible to work directly with image pixels by relying on state-of-the art supervised learning methods [Marée et al., CVPR 2005]
- Recent advances in reinforcement learning = the newly introduced fitted Q iteration family of algorithms can exploit the generation capabilities of any supervised learning method.

  [Ernst et al., JMLR 2005]

Question ?

If using directly image pixels works in image classification and since we have now reinforcement learning algorithms that can exploit the generalization capabilities of any supervised learning method, then why not to use directly image pixels in reinforcement learning ?

Problem formulation

Deterministic version

Discrete-time dynamics: $x_{t+1} = f(x_t, u_t) \quad t = 0, 1, \ldots$ where $x_t \in X$ and $u_t \in U$.

Cost function: $c(x, u) : X \times U \to \mathbf{R}$. $c(x, u)$ bounded by B_c.
Instantaneous cost: $c_t = c(x_t, u_t)$

Discounted infinite horizon cost associated to stationary policy $\mu : X \to U$: $J^\mu(x) = \lim_{N \to \infty} \sum_{t=0}^{N-1} \gamma^t c(x_t, \mu(x_t))$ where $\gamma \in [0, 1[$.

Optimal stationary policy μ^* : Policy that minimizes J^μ for all x.

Objective: Find an optimal policy μ^*.

We do not know: The discrete-time dynamics and the cost function.

We know instead: A set of system transitions:
$\mathcal{F} = \{(x_t^l, u_t^l, c_t^l, x_{t+1}^l)\}_{l=1}^{\#\mathcal{F}}$.

Some dynamic programming results

Sequence of state-action value functions Q_N: $X \times U \to \mathbb{R}$

$$Q_N(x,u) = c(x,u) + \gamma \min_{u' \in U} Q_{N-1}(f(x,u),u'), \quad \forall N > 1$$

with $Q_1(x,u) \equiv c(x,u)$, converges to the Q-function, unique solution of the Bellman equation:

$$Q(x,u) = c(x,u) + \gamma \min_{u' \in U} Q(f(x,u),u').$$

Necessary and sufficient optimality condition:

$$\mu^*(x) \in \arg\min_{u \in U} Q(x,u)$$

Suboptimal stationary policy μ_N^*:

$$\mu_N^*(x) \in \arg\min_{u \in U} Q_N(x,u).$$

Bound on μ_N^*:

$$J^{\mu_N^*} - J^{\mu^*} \leq \frac{2\gamma^N B_c}{(1-\gamma)^2}.$$

Fitted Q iteration

Fitted Q iteration computes from $\mathcal{F}$ the functions $\hat{Q}_1$, $\hat{Q}_2$, ..., $\hat{Q}_N$, approximations of Q_1, Q_2, ..., Q_N.

Computation done iteratively by solving a sequence of standard supervised learning problems. Training sample for the k^{th} ($k \geq 1$) problem is $\left\{ \left((x_t^l, u_t^l),\ c_t^l + \gamma \min_{u \in U} \hat{Q}_{k-1}(x_{t+1}^l, u) \right) \right\}_{l=1}^{\#\mathcal{F}}$ with $\hat{Q}_0(x, u) \equiv 0$. From the k^{th} training sample, the supervised learning algorithm outputs $\hat{Q}_k$.

$\hat{\mu}_N^*(x) \in \arg\min_{u \in U} \hat{Q}_N(x, u)$ is taken as approximation of $\mu^*(x)$.

Fitted Q iteration: some remarks

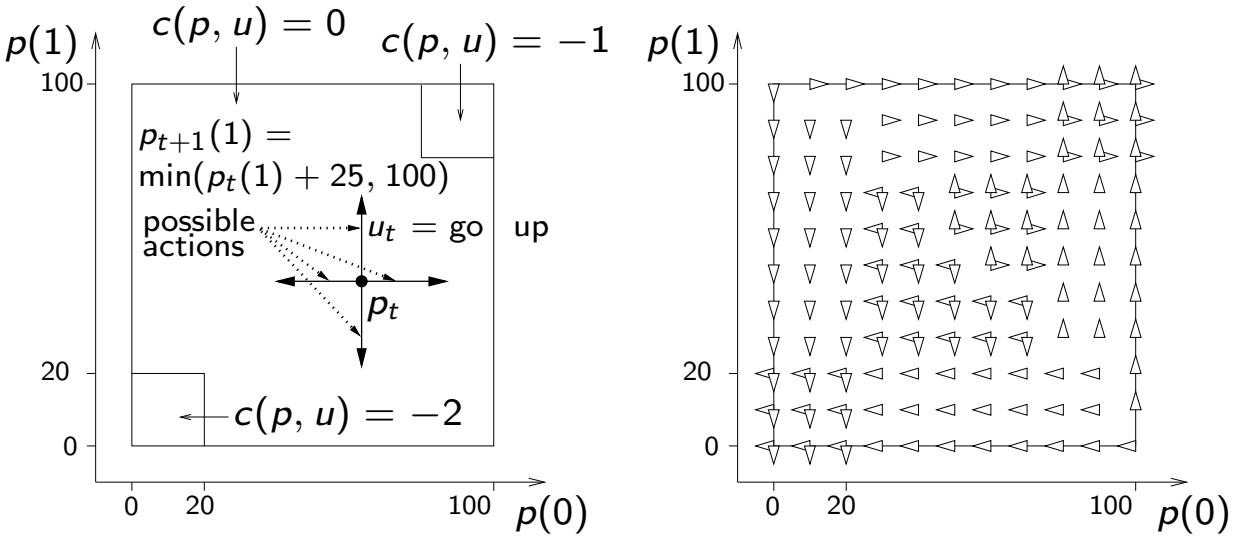
Performances of the algorithm depends on the supervised learning method chosen.

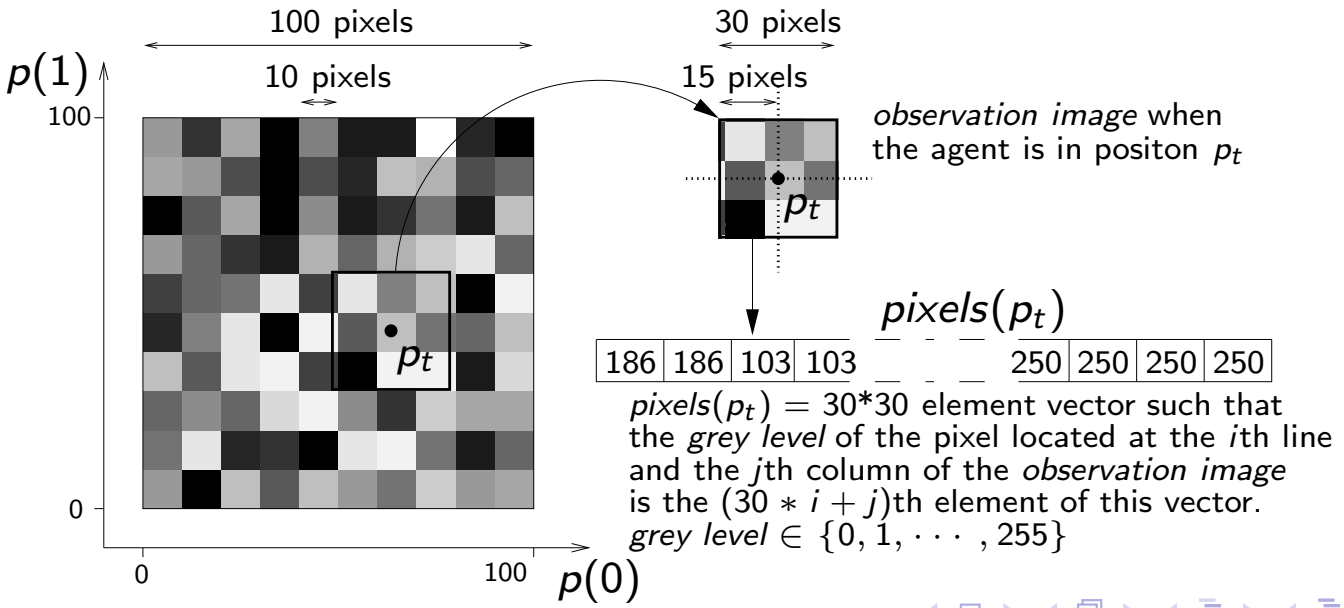
Excellent performances have been observed when combined with supervised learning methods based on ensemble of regression trees.

Works also for stochastic systems

Consistency can be ensured under appropriate assumptions on the supervised learning method, the sampling process, the system dynamics and the cost function.

Our experimental protocol: test problem





Framework parameters

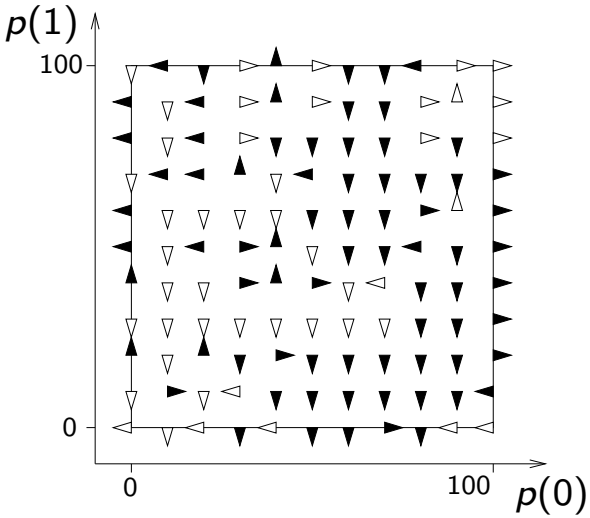
Four-tuples generation ($\#\mathcal{F} = \mathrm{n}$)

- We repeat n times the sequence of instructions:
  1. draw p_0 at random in P and u at random in U;
  2. observe r_0 and p_1;
  3. add $(pixels(p_0), u_0, r_0, pixels(p_1))$ to $\mathcal{F}$.

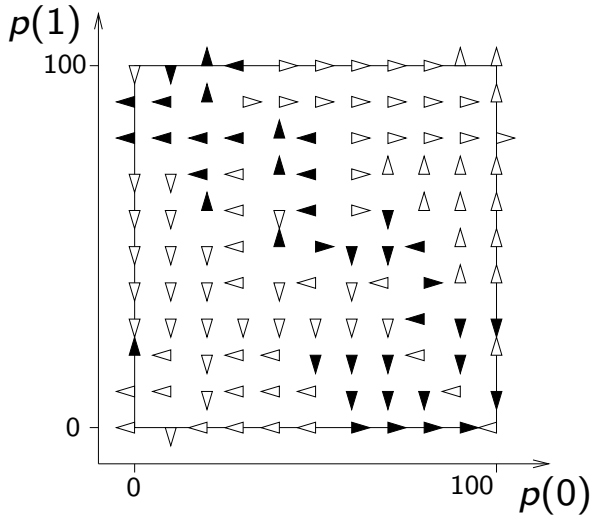
Fitted Q iteration algorithm

- $\hat{Q}_k$ computed with Extra-Trees [Geurts et al., Machine Learning 2006]
- Number of iterations $N = 10$
- Approximation of the optimal policy $\hat{\mu}^*_{10}(x) = \arg\max_u \hat{Q}_{10}(x, u)$

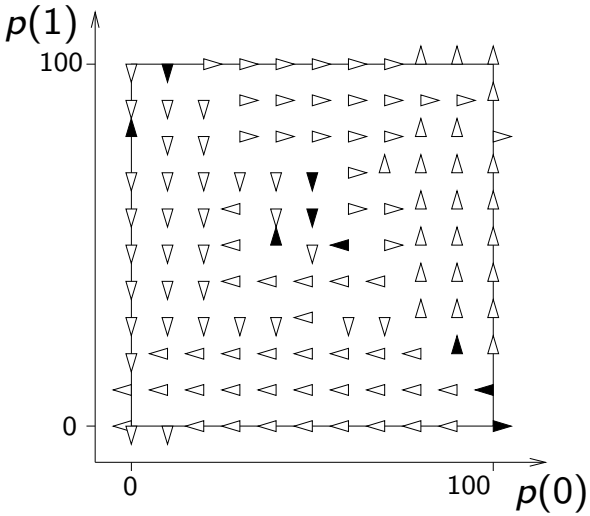
Results



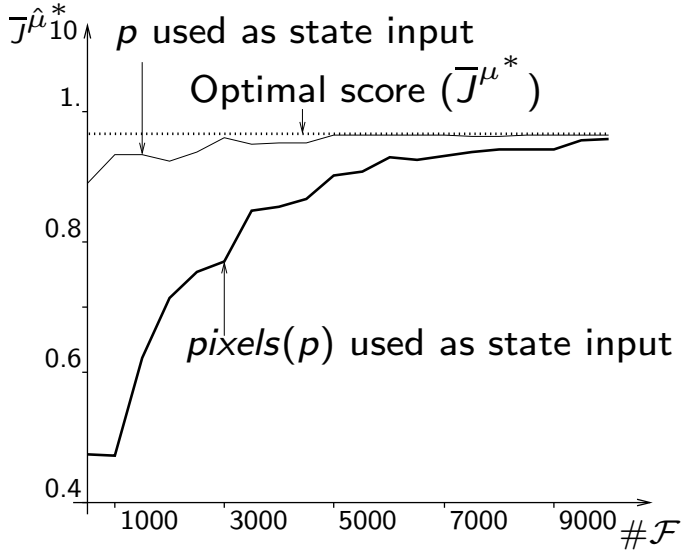
(a) $\hat{\mu}^*_{10}$, 500 system trans.



(b) $\hat{\mu}^*_{10}$, 2000 system trans.



(c) $\hat{\mu}^*_{10}$, 8000 system trans.



(d) score versus nb system trans.

Influence of the navigation image characteristics

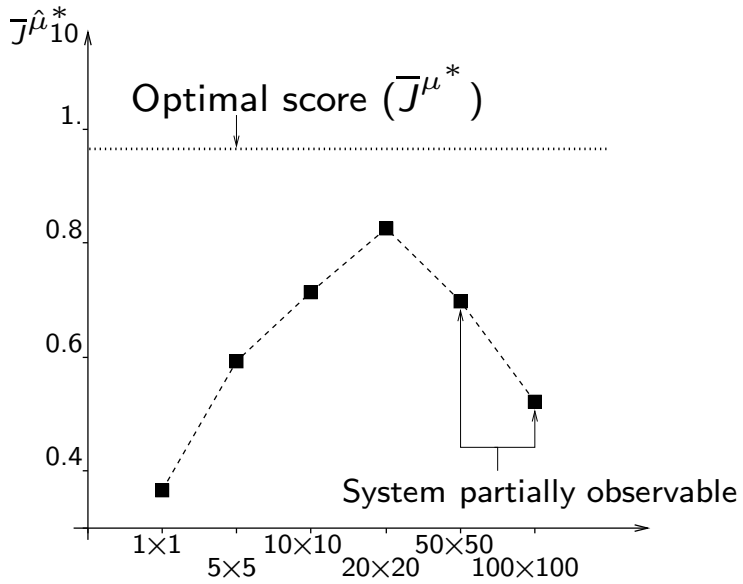


Figure: Evolution of the score with the size of the constant grey level tiles. 2000 system samples.

Conclusions

We have applied a new reinforcement algorithm known as fitted Q iteration to the problem of navigation from visual percepts. State inputs were the raw pixels.

Good results even if in such conditions information is spread over a large number of low-level input variables $\Rightarrow$ Question the need for still going through a feature extraction phase.

Strong influence of the learning quality on the characteristics of the images the agent gets as input states.

References

- "Tree-based batch mode reinforcement learning". D. Ernst, P. Geurts and L. Wehenkel. In Journal of Machine Learning Research. April 2005, Volume 6, pages 503-556.
- "Random subwindows for robust image classification". R. Marée, P. Geurts, J. Piater and L. Wehenkel. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, June 2005, Volume 1, pages 34-40.
- "Extremely Randomized Trees". P. Geurts, D. Ernst, and L. Wehenkel. Machine Learning, Volume 36, Number 1, page 3-42, 2006.
- "Reinforcement learning with raw pixels as state input". D. Ernst, R. Marée and L. Wehenkel. International Workshop on Intelligent Computing in Pattern Analysis/Synthesis (IWICPAS). Proceedings series: Lecture Notes in Computer Science, Volume 4153, pages 446-454, August 2006.