Where am I? Let me learn from the real world!

Luciano Serafini and Paolo Traverso
Fondazione Bruno Kessler
Trento, Italy

Introduction and Motivations

Most automated planning techniques are based on abstract representations of the real world, which are usually called *"planning domains"*. A planning domain is most often formalized as a deterministic, nondeterministic, or stochastic state transition system (Ghallab, Nau, and Traverso 2016). The agent perceives the environment through sensors that provide data in a continuous space. These perceptions are at a different level from the abstract discrete state space. Most of the works in planning and learning (see, e.g., (Sutton and Barto 1998; Geffner and Bonet 2013))[1] assume a fixed correspondence between sensor data and abstract states. After acting, the agent knows exactly the state it has reached. This is a rather strong and unrealistic assumption, since it supposes that the agents knows the environment in which it operates, and that such an environment is immutable.

We should instead provide a formal framework where the mapping between abstract states and the real world is part of the model of the agent, it is explicitly represented in the model, and it is learned and updated along the "life" of the agent.

Reference Model

A *deterministic planning domain*[2] is a triple $\mathcal{D} = \langle S, A, \gamma \rangle$, composed of a finite non empty set of states, a finite non empty set of actions and a state transition function. A *planning problem* is a triple $\mathcal{P} = \langle \mathcal{D}, s_0, S_g \rangle$ composed of a planning domain, an initial state and a set of goal states. A plan π for $\mathcal{D}$ is a partial function from S to A. A simple planning domain is shown in Figure 1.(a). domain. The way in which an agent perceives the world is modeled by a *perception function*, i.e., a function $f : \mathbb{R}^n \times S \to R^+$, defined as $f(\boldsymbol{x}, s) = p(\boldsymbol{x}|s) \cdot p(s)$, where $p(\boldsymbol{x}, s)$ is a join PDF on $\mathbb{R}^n \times S$. In other words $f(\boldsymbol{x}, s)$ is the likelihood of observing $\boldsymbol{x}$ being in a state s. Figure 1.(b) shows an example of perception function $f(\langle x, y \rangle, s_{ij})$ for the abstract planning domain 1.(a) in the real world 1.(c), where $p(\langle x, y \rangle | s_{ij}) = \mathcal{N}\left(\left\langle i - \frac{1}{2}, j - \frac{1}{2} \right\rangle, \boldsymbol{\Sigma}\right)$, for some constant covariance matrix $\boldsymbol{\Sigma}$ and $p(s_{ij}) = .25$.

Notice that the agent's abstract model of Figure 1 is not coherent with the real world for two main reasons: First, transition from and to s_{22} are not possible in the real world, due to the presence of walls; second there are two missing states, corresponding to the rightmost rooms in the building. Intuitively the most coherent model is shown in Figure 2. The planning, acting, and learning algorithm should be able to learn a more coherent planning domain and perception function as for instance the one shown in Figure 2.

Planning, Acting, and Learning Algorithm

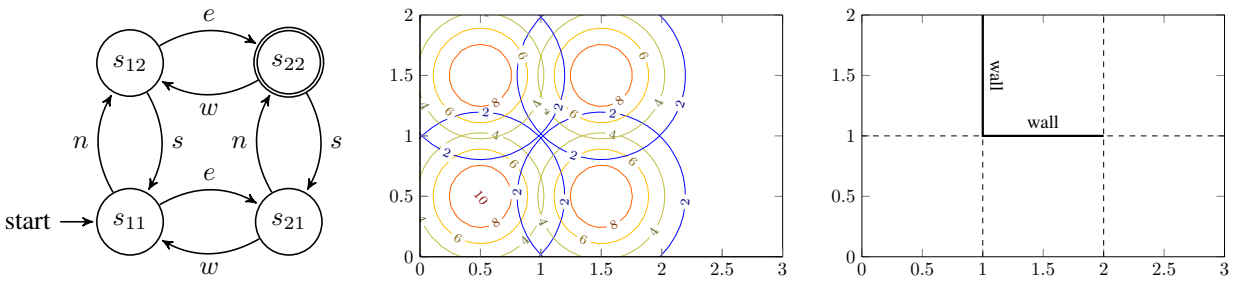
Algorithm 1 PlanActLearn

Require: $\mathcal{P} = \langle \langle S, A, \gamma \rangle, s_0, S_g \rangle$ {A planning problem}
Require: f {a perception function}
  $\mathcal{T} \leftarrow \langle\rangle$ {The empty history of transitions}
  $\mathcal{O} \leftarrow \langle\rangle$ {The empty history of observations}
  while $s_0 \notin S_g$ **do**
    $\pi \leftarrow plan(\mathcal{P})$
    while $\pi(s_0)$ is defined **do**
      $\boldsymbol{x} \leftarrow act(\pi(s_0))$
      $s_0' \leftarrow \text{argmax}_{s \in S} f(\boldsymbol{x}, s)$
      if $f(\boldsymbol{x}, s_0') < \epsilon$ **then**
        $s_0' \leftarrow s_{new}$
        $S \leftarrow S \cup \{s_{new}\}$
      end if
      $\mathcal{T} \leftarrow append(T, \langle s_0, \pi(s_0), s_0' \rangle)$ {extend the transition history with the last one}
      $\mathcal{O} \leftarrow append(\mathcal{O}, \langle s_0', \boldsymbol{x} \rangle)$ {extend the observation history with the last one}
      $\gamma \leftarrow \mathsf{update\_trans}(\gamma, \mathcal{T})$
      $f \leftarrow \mathsf{update\_perc}(f, \mathcal{O})$
      $s_0 \leftarrow s_0'$
    end while
  end while

Algorithm 1 interleaves planning, acting, and learning. We look for a plan ($plan(\mathcal{P})$), we execute the plan in the current initial state s_0, and perceive the data from the real world in the vector of real numbers $\boldsymbol{x}$. We get the state s_0' that maximizes likelihood, and if it is below a given threshold ϵ, we create a new state s_{new}. The functions update_trans and

[1] In some works (see, e.g., (Co-Reyes et al. 2018)) the two levels are collapsed, since planning is performed in a continuous space

[2] The model can be extended to stochastic domains.



(a) Planning problem (b) Perception function (c) Real world

Figure 1: (a) A planning problem on a domain composed of 4 states, corresponding to 4 rooms, no walls between them, and 4 actions n, s, w, and e (go north, south, west, and east). (b) A perception function associated to the planning domain. (c) The real world: the building has 6 (and no 4) rooms, and two walls

update_perc update the transition function γ and the perception function f, respectively, depending on the data available in $\mathcal{T}$ and $\mathcal{O}$. The update functions take into account what has been observed in the past, i.e., $\mathcal{O}$ and $\mathcal{T}$, and what has been just observed, i.e., $\langle s_0, \pi(s_0), s_0' \rangle$ and $\langle s_0', \boldsymbol{x} \rangle$. Update functions can be defined in several different ways, depending on whether we follow a cautious strategy, where changes are made only if there are a certain number of evidences from acting and perceiving the real world, or a more impulsive reaction to what the agent has just observed. The update of the perception function is based on the current preception function $f(\langle x, y \rangle, s)$ for $s \in S$ and the set of observations $\mathcal{O}$. We suppose that the perception function is parametric on $\boldsymbol{\theta} = \langle \theta_1, \ldots, \theta_k \rangle$. In our example $\boldsymbol{\theta}$ contains μ and Σ the mean and the covariance matrix. Given a new observation $\langle \boldsymbol{x}, s \rangle$ and a set of previous observations $\mathcal{O}$, we have to update the parameters $\boldsymbol{\theta}$ in order to maximize the likelihood of the entire set of observations extended with the new observation. A general procedure for sequential estimation is described in (Bishop 2006). Like in the case of the revision of the transition function, we should balance the update depending on whether we are cautious or impulsive.

Challenges for Future Work

In this abstract we have provided some intuitions towards a novel model for planning and learning. The formal framework should be defined in detail, as well as its implementation and an experimental evaluation. Finally, we should study the relation with Logic Tensor Flow (Donadello, Serafini, and d'Avila Garcez 2017), where reasoning in the model constrains the (deep) learning task.

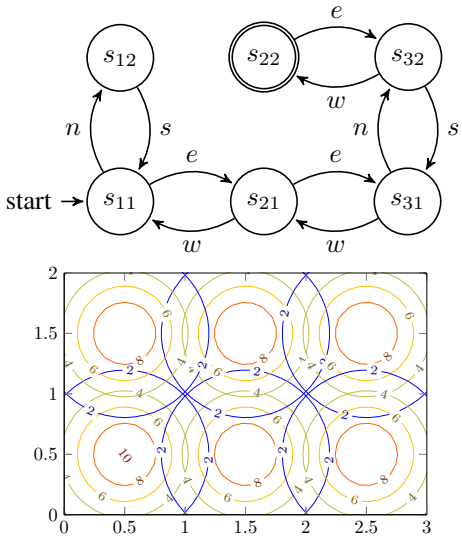


Figure 2: The new planning domain and perception function is obtained by extending the initial domain of Figure 1, with two new states, and the corresponding perception functions

References

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag.

Co-Reyes, J. D.; Liu, Y.; Gupta, A.; Eysenbach, B.; Abbeel, P.; and Levine, S. 2018. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. In *Proceedings of ICML 2018*, 1008–1017.

Donadello, I.; Serafini, L.; and d'Avila Garcez, A. S. 2017. Logic tensor networks for semantic image interpretation. In *Proceedings of IJCAI 2017*, 1596–1602.

Geffner, H., and Bonet, B. 2013. *A Concise Introduction to Models and Methods for Automated Planning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Ghallab, M.; Nau, D. S.; and Traverso, P. 2016. *Automated Planning and Acting*. Cambridge University Press.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press.