Practical Examples Using Semantic Simulation With Reinforcement Learning

D. Gueorguiev 12/4/2022

The Game Addition

Let us consider the game $Addition$ described in *Blackwell's Theory of Games and Statistical Decisions* (Blackwell & Girshik, 1978, p. 14):

I and II alternatively choose integers, each choice being one of the integers $1, \dots, k$ and each choice made with the knowledge of all preceding choices. As soon as the sum of the chosen integers exceeds N, the last player to choose pays his opponent one unit.
The situation at which player I finds himself at his rth move is described by a sequence $s_r = (i_1, i_2, \dots, i_{2r-2})$ with each i_j being one of the integers $1, \dots, k$ and

$\sum_{j=1}^{2r-2} i_j \le N$

Denote by S_r the set of possible sequences s_r where $r = 2, \dots, \left[\frac{N}{2}\right] + 1$ and $[z]$ denotes the closest integer which does not exceed z. A strategy x for I consists of a set of $\left[\frac{N}{2}\right] + 1$ functions $f_1, \dots, f_{\left[\frac{N}{2}\right]+1}$, where f_r is a function defined on S_r assuming only values $1, 2, \dots, k$: f_r specifies I's rth move when the previous history of the play is s_r. Similarly, a strategy y for II is a set of $\left[\frac{(N+1)}{2}\right]$ functions $g_1, \dots, g_{\left[\frac{(N+1)}{2}\right]}$, where g_r is defined for the set T_r of all sequences $t_r = (i_1, \dots, i_{2r-1})$ with each i_j being one of the integers 1,2, …, k and

$\sum_{j=1}^{2r-1} i_j \le N$

Define $i_1(x,y) = f_1$ and inductively for $j > 0$,
$i_{2j}(x,y) = g_j\left(i_1(x,y), \dots, i_{2j-1}(x,y)\right)$
$i_{2j+1}(x,y) = f_{j+1}\left(i_1(x,y), \dots, i_{2j}(x,y)\right)$
(this induction describes the manner in which a referee would carry out the instructions of the players) and let $j^*(x,y)$ be the largest j for which $i_j(x,y)$ is defined. Then

$$M(x,y) = \begin{cases} 1 \text{ if } j^*(x,y) \text{ is even} \\ -1 \text{ if } j^*(x,y) \text{ is odd} \end{cases}$$

Constructing semantic universe for the game $Addition$

Let us consider the following thought experiment – we have two players playing the $Addition$ game described earlier. Each player is represented by semantic simulation which has its own set of semantic structures and semantic template which recognizes the rules of the game. Let us start our experiment by looking in the semantic template which recognizes the rules of the game which we will name *semantic* *recognizer*. That is - we are interested in what the semantic recognizer might be taking as an input and producing as an output and how the semantic recognizer template would be interacting with the rest of the semantic structures running in the simulation.

Let us assume that the semantic simulation corresponding to each of the two players I and II is limited to the simply connected regions R_1 and R_2 in semantic space. Additionally, we introduce an Arbiter which will be assigned its own simply connected region R_3 in semantic space. Let $\dim(R_1) = \dim(R_2) = \dim(R_3) = L$. Let us assume that $R_1 \cap R_2 \cap R_3 = C$ where C is finite, closed and simply connected region of semantic space with the same number of dimensions L. We will denote C as the *common simulation region*.

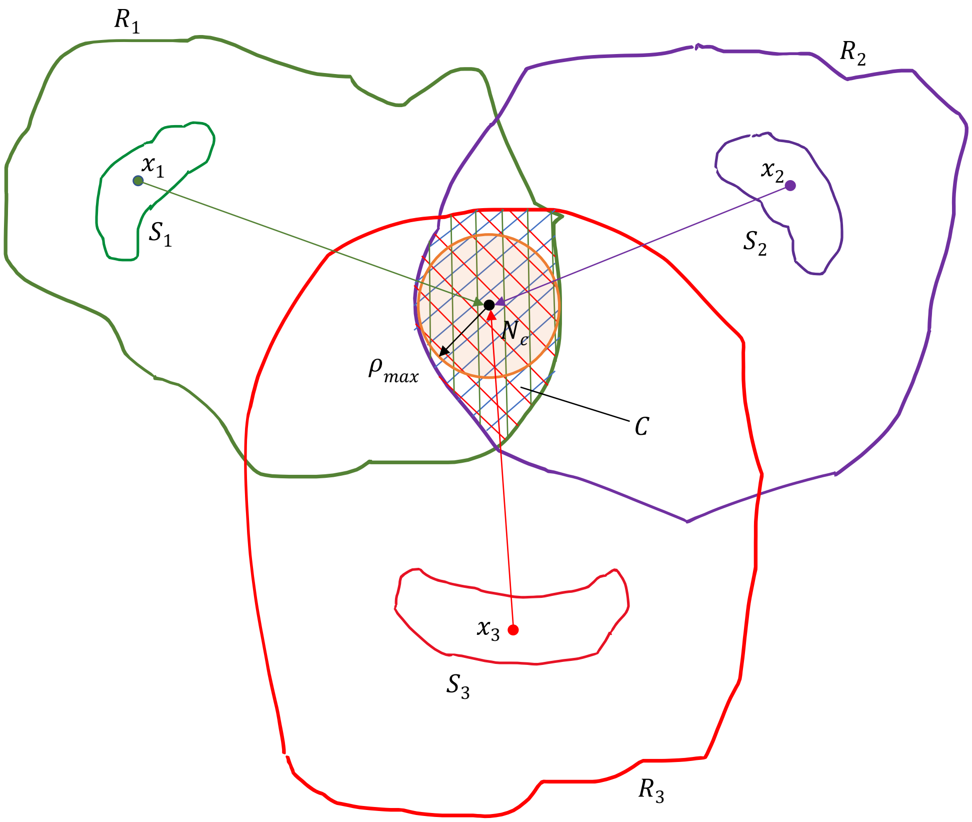

Figure 1: Layout of the simulation space in Blackwell's game *Addition*

Definition: *neutral point* of a simply connected region in metric space
Let C is a simply connected region in some L dimensional metric space. Then the point N_c is a neutral point *iff* it is the center of the largest L dimensional sphere which can fit entirely in the simply connected region C without including any points outside of C. Formally,
$\exists\ N_c \in C \therefore\ \rho_{max} = \max\limits_{\rho} |N_c - x| \le \rho\ \ \forall\ x \in C$

With N_c we denote *the neutral point* of the common simulation region C. The neutral point will be the attraction center for all outputs from player I and II's as well as the arbiter simulations. Both players I and II as well as the Arbiter will produce an output which will be a semantic particle starting its existence at a point inside their respective regions S_1, S_2, S_3 shown on Figure 1.

A couple notational conventions which will simplify the discussion:
Let us have a template T_s defined over the semantic region C.
In the future we will denote the region over which the template is defined with the appropriate symbol denoting the region in parentheses;
Thus $T_s(C)$ indicates that T_s is defined over C. A particle p_s having trajectory intersecting with specific region C will be denoted with the following notation $p_s \rightsquigarrow C$. We denote a template match, that is the template T_s has matched the input represented by p_s, with the following symbolic notation
$T_s(p_s \rightsquigarrow C) \upharpoonleft$.
Notation for chaining template actions:

Here is how the game simulation will proceed:
For simplicity let us assume that the game parameter N defined earlier is given and it is known by the two players and the Arbiter. Also, we will assume that the Arbiter will make decision who will be the first of the two players to play; for simplicity the decision-making process of the Arbiter will be omitted from the discussion. Let us represent this decision-making process of the Arbiter by the semantic template T_s (s for start of the game). The template T_s accepts an input indicating the start of the game.

The input indicating the start of the game will be supplied as particle with specific signature which we will denote with p_s. As soon as the arbiter template T_s detects that the signature of p_s is present in C it sends either a particle $p_{s,1}$ to region S_1 or $p_{s,2}$ to region S_2. In case the case of $p_{s,1} \rightsquigarrow S_1$ a template $T_{1,0}$ which belongs to Player I will recognize the signature of $p_{s,1}$ that is $T_{1,0}(p_{s,1} \rightsquigarrow S_1) \upharpoonleft$. On a match $T_{1,0}$ will send a messenger particle $m_{1,0}$ to another template of Player I - $T_{1,1}$.
In the case when $p_{s,2}$ is sent to S_2 a template T_2 which belongs to Player II will recognize the signature of $p_{s,2}$.

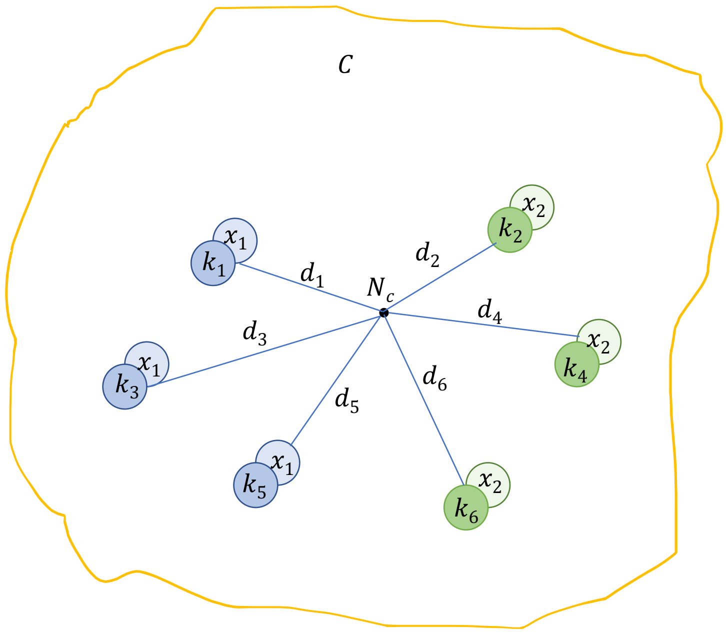

Figure 2: Possible final arrangement of the semantic particles produced by the two players at the end of a game of *Addition*

Figure 3: Semantic structure formed by the final arrangement of the output of the two players

A semantic particle is produced at S_0 in C by the arbiter announcing a proposed value of N.

Bibliography

Bang-Jensen, J., & Gutin, G. (2007). *Diagraphs: Theory, Algorithms and Applications.* Odense, Denmark, London, UK: Springer-Verlag.

Bellman, R. (1972). *Dynamic Programming.* Princeton, NJ: Princeton University Press, Sixth Printing.

Blackwell, D. A., & Girshik, M. A. (1978). *Theory of Games and Statistical Decisions.* New York: Dover Publications; Illustrated edition.

Denardo, E. V. (1982). *Dynamic Programming: Models and Applications.* Englewood Cliffs, NJ: Prentice-Hall.

Francois-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An Introduction To Deep Reinforcement Learning. *Foundations and Trends in Machine Learning: Vol. 11, No. 3-4*.

Gosavi, A. (2022). Reinforcement Learning: Tutorial and Recent Advances. *INFORMS Journal on Computing*.

Harmon, M. E., & Harmon, S. S. (Jan 1997). *Reinforcement Learning: A Tutorial.* Wright Patterson AFB OH 45433: Avionics DIrectorate, Wright Laboratory, Air Force Materiel Command.

Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020, Nov 1). *Offline Reinforcement Learning: Tutorial, Review, and Perespectives on Open Problems*. Retrieved from arxiv.org: https://arxiv.org/abs/2005.01643

Neapolitan, R. E. (2019). *Bayesian Networks.* Upper Saddle River, NJ: Prentice-Hall.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: Introduction, second edition.* Cambridge, Massachusetts: The MIT Press.