

# Introducción al Aprendizaje por Refuerzos (III)

June 21, 2017

## 1 Introducción al Aprendizaje por Refuerzos (III)

- Model Based Reinforcement Learning (Arquitectura Dyna). Integración de Planning, Acting, y Learning.
- Dyna-Q.
- Corrección de los Modelos.
- Dyna-Q+.
- Ejercicios.

### 1.1 5to año - Ingeniería en Sistemas de Información

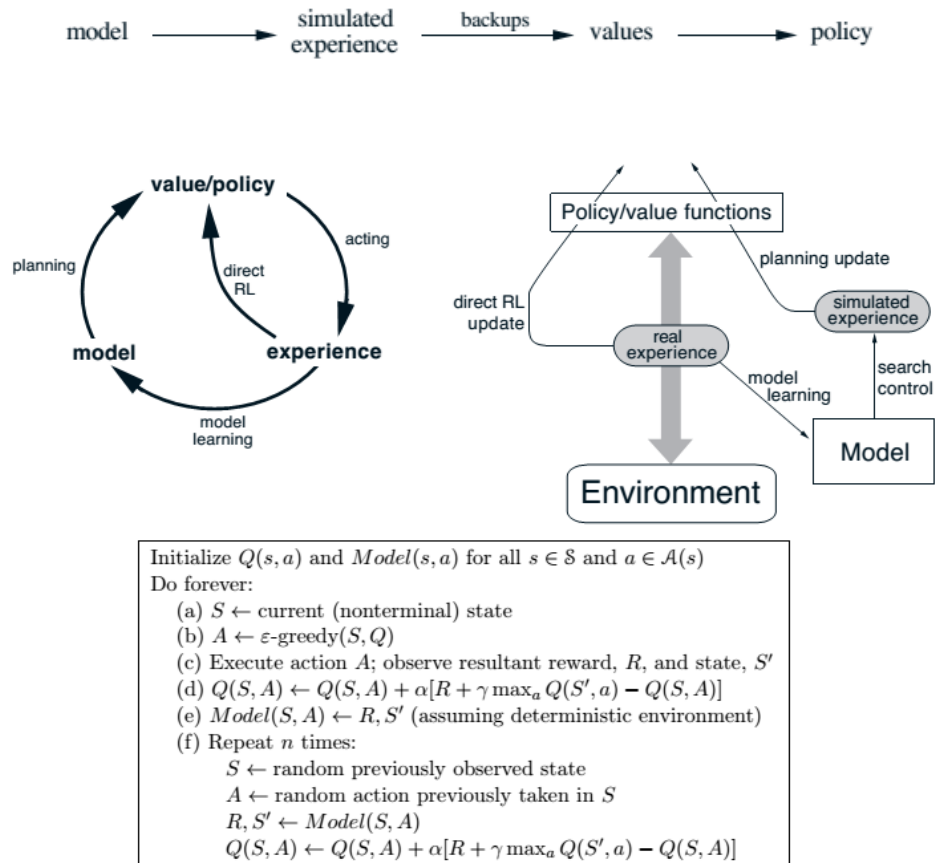
#### 1.1.1 Facultad Regional Villa María

### 1.2 Aprendizaje por Refuerzos basado en Modelos.

- Los modelos del entorno representan cualquier elemento que permita al agente predecir cómo el entorno responderá ante las acciones que ejecuta. Dado un estado y una acción, el modelo produce una predicción del siguiente estado y el reward asociado.
- Si el modelo es estocástico, entonces existen distintos posibles estados siguientes/rewards, cada uno con distinta probabilidad de ocurrir.
- Algunos modelos producen una descripción de todas las posibilidades con sus probabilidades asociadas: los mismos se denominan **modelos de distribución**. Otros modelos producen solo una de las posibilidades, sampleadas en función de su probabilidad (la cuál es desconocida a priori por el agente); estos se denominan **modelos basados en ejemplos**.
- Ambos tipos de modelos pueden ser empleados para replicar o generar experiencia. Dado un estado-acción inicial, un modelo basado en ejemplos produce una posible transición, y un modelo de distribución genera todas las posibles transiciones pesadas por su probabilidad de ocurrir. De esa manera, un modelo basado en ejemplos podría producir un episodio completo, y un modelo de distribución, todos los posibles episodios y sus posibilidades.
- En ambos casos, decimos que el modelo es utilizado para producir **experiencia simulada**.

### 1.3 Arquitectura Dyna: Integración de Planning, Acting, y Learning.

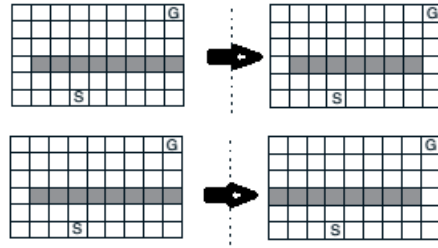
- El término *planning* en entornos RL se utiliza para referirse a cualquier proceso computacional que emplea el modelo como entrada y produce o mejora una política de interacción con un entorno determinado.



- Cuando el *planning* es realizado on-line, mientras se interactúa con un entorno, surgen cuestiones importantes que es necesario resolver. Por ejemplo, la nueva información proveniente de la interacción podría impactar en el modelo y de esa manera modificar el proceso de *planning*.
- Si la toma de decisiones y el aprendizaje de modelos (*decision-making* y *model-learning* respectivamente) son procesos basados en computación intensiva, entonces los recursos computacionales deberían ser divididos entre ambos.
- Dyna-Q es una arquitectura que integra las funciones más importantes para un agente que realiza planificación (*planning*) on-line. La misma, establece al menos dos roles para la experiencia real generada por el agente: el primero, establece que la misma puede ser empleada para mejorar el modelo (*model learning*, hacer que refleje la realidad lo mejor posible), o bien mejorar la función de valor y por ende la política, empleando los métodos basados en Aprendizaje por Refuerzo (*direct-RL*).

## 1.4 Corrección de los modelos

- En muchos entornos determinísticos, una vez aprendido el modelo, puede utilizarse de manera directa para propagar cambios producidos por las Diferencias Temporales.



- En entornos no determinísticos, o en aquellos en los cuales se da el hecho de que en algún momento de la interacción agente-entorno este último cambia, es necesario incorporar mecanismos de auto-corrección de modelos para evitar obtener políticas sub-óptimas.
- El problema general de corrección de modelos es otra versión del trade-off exploración-explotación.
- En el contexto del *planning* en RL, la exploración significa ejecutar acciones que mejoran el modelo, mientras que la explotación significa comportarse de manera óptima de acuerdo a lo establecido por el modelo. Es decir, se busca que el agente explore para detectar cambios en el entorno, pero sin degradar la performance del mismo de manera determinante.

#### 1.4.1 Dyna-Q+

- El método *Dyna-Q+* permite resolver el trade-off mencionado empleando una heurística basada en la cantidad de tiempo que un par estado-acción no es visitado de manera real por el agente.
- Mientras más tiempo haya pasado desde una interacción real, mayor es la probabilidad de que las dinámicas en determinados pares estado-acción hayan cambiado, y que por lo tanto, el modelo sea incorrecto.
- Para motivar el testeo de acciones que hace largo tiempo no se ejecutan en determinado estado, se otorga un *bonus especial* a las acciones simuladas que tienen en cuenta dichas acciones.
- En particular, si la recompensa para una transición determinada es  $R$ , y dicha transición no ha sido ejecutada realmente en  $\tau$  time steps, entonces las actualizaciones basadas en *planning* son llevadas a cabo como si la misma produjera una recompensa de  $R + \kappa\sqrt{\tau}$ , para un  $\kappa$  comprendido entre 0 y 1. El parámetro  $\kappa$  se denomina **curiosidad computacional**.

### 1.5 Ejercicios

Fecha de entrega: **27/06/2017**

Nota: la resolución de los ejercicios es **individual**; en el caso de que dos ejercicios enviados contengan un código igual o muy similar (sin considerar los comentarios), se los considerará a ambos como desaprobados. La reutilización del código de los notebooks está permitida (por ejemplo para confeccionar gráficos).

1.1 Modifique la implementación de Q-Learning realizada en el TP 8, para integrar aprendizaje basado en modelos (Dyna-Q).

1.2 Compare la convergencia de la implementación basada en  $Sarsa(\lambda)$ , respecto a la implementación de Dyna-Q empleando una gráfica de Steps Acumulados Promedio vs. Episodio.