Aprendizaje Reforzado

Maestría en Ciencia de Datos, DC - UBA

Julián Martínez
Javier Kreiner

Detalles administrativos

- Clases lunes y miércoles a de 19 a 22
- 6 de febrero al 18 de marzo
- Laboratorio Turing en el DC
- Las clases son teórico-prácticas y hay que tener una compu con el software instalado para poder acompañar
- Evaluación: Ejercicios semanales 50%, Trabajo/Presentación final 50%
- Emails:
  - julianfm7@gmail.com
  - javkrei@gmail.com

Estructura del curso - Overview

- Introducción a Aprendizaje Reforzado (Reinforcement Learning).
- Framework matemático. Framework computacional. Procesos de decisión Markovianos.
- Funciones de Valor. Cálculo y aproximación. Programación dinámica, ecuación de Bellman. Evaluación de políticas. Iteración de políticas.
- Método de Monte Carlo, predicción y control. On-policy y off-policy.
- Método de Diferencias Temporales.
- Métodos aproximados: Aproximación de funciones de valor.
- Métodos de Gradiente de Política.
- Exploración vs Explotación.
- Integración de Planificación y Aprendizaje.
- Introducción a Deep Reinforcement Learning.

Fuentes

Libros:

- **Reinforcement Learning: An Introduction.** Richard S. Sutton and Andrew G. Barto, Second Edition, MIT Press, Cambridge, MA, 2018 (http://incompleteideas.net/book/the-book-2nd.html)
- **Algorithms for Reinforcement Learning.** Csaba Szepesvári, Synthesis Lectures On Artificial Intelligence And Machine Learning #9, Morgan & Claypool publishers, 2010

Fuentes

Cursos online:

- UCL RL course: http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
- https://github.com/yandexdataschool/Practical_RL/tree/coursera
- https://yadi.sk/d/loPpY45J3EAYfU

Plan de la clase

- Qué es Aprendizaje Reforzado (Reinforcement Learning)
- Relación con otros campos
- Diferencias con otros paradigmas de ML
- Características propias de Aprendizaje Reforzado
- Definición del problema
- Ejemplos
- Review de matemática
- Ejercicio de programación
- (Software utilizado)
- Bonus: Lecturas recomendadas

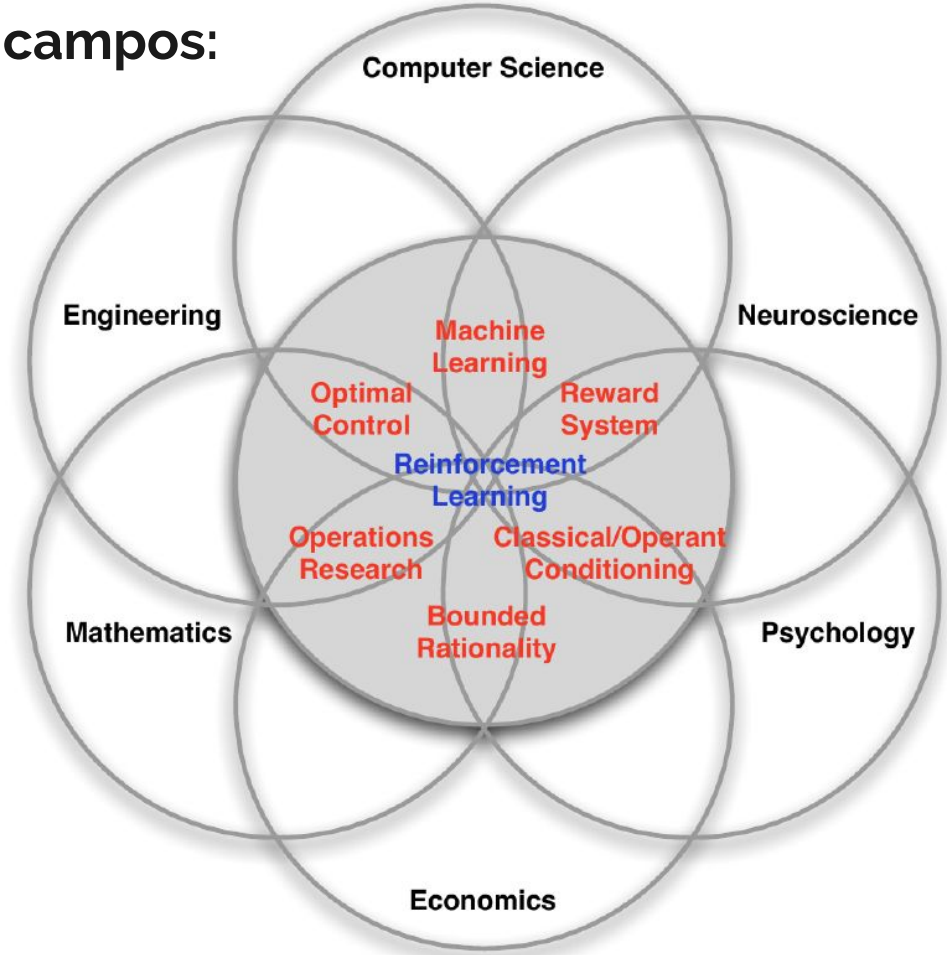
Qué es Aprendizaje Reforzado

- Es aprender a ‘qué hacer’ - cómo mapear situaciones a acciones - para maximizar una señal de recompensa
- Un problema, una clase de métodos de solución, el campo de estudio del problema
- Al agente no se le dice lo que debe hacer, debe aprenderlo interactuando con el ambiente y recibiendo recompensas
- El problema central: aprendizaje a través de la interacción con el ambiente por prueba y error
- El objetivo: la toma de decisiones secuenciales de manera óptima

Qué es Aprendizaje Reforzado

- El agente obtiene información o ‘sensa’ u obtiene información a través de sus ‘sentidos’ o aparatos de medición
- A través de esa interacción con el ambiente el agente obtiene información y aprende sobre el comportamiento del mundo
- Con esa información decide qué acciones realizar en cada momento
- El comportamiento es guiado por un objetivo que se modela en forma de una recompensa numérica
- De nuevo, el objetivo la toma de decisiones secuenciales de manera óptima, esto es: maximizar la recompensa acumulada

Relación con otros campos:



Diferencias con otros paradigmas de ML, RL vs Aprendizaje supervisado

- No viene dado un dataset con inputs y targets
- No hay un ‘supervisor’, o sea input y targets, hay una señal de recompensa
- El feedback se recibe con retraso, no es instantáneo
- Las decisiones son secuenciales, los datos no son i.i.d.
- Las acciones del agente modifican los datos que va recibiendo

Diferencias con otros paradigmas de ML, RL vs aprendizaje supervisado

Aprendizaje supervisado	**Aprendizaje reforzado**
Aprende a aproximar respuestas de referencia	Aprende la estrategia óptima a través de prueba y error
Necesita un dataset con las respuestas correctas	Necesita el feedback de las propias acciones del agente
El modelo no afecta los datos de entrada	El agente puede modificar sus propias observaciones (dependen de las acciones)

Diferencias con otros paradigmas de ML, RL vs Aprendizaje no supervisado

Aprendizaje no supervisado	**Aprendizaje reforzado**
Aprende la estructura subyacente de un conjunto de datos	Aprende la estrategia óptima a través de prueba y error
No hay feedback	Necesita el feedback de las propias acciones del agente
El modelo no afecta los datos de entrada	El agente puede modificar sus propias observaciones (dependen de las acciones)

Aprendizaje Reforzado

- Objetivo: tomar decisiones de tal manera de maximizar las recompensas futuras acumuladas
- Aspectos distintivos:
  - Las acciones pueden tener consecuencias de largo plazo
  - Las recompensas pueden llegar con retraso
  - Puede ser bueno sacrificar recompensa ahora para obtener más en el futuro

Aprendizaje Reforzado

- Uno de los elementos fundamentales es el dilema de exploración vs explotación
- Explotación significa tomar la acción que creemos dado nuestro modelo actual nos dará más recompensa
- Exploración es tomar acciones de tal manera de aprender más respecto al ambiente y las recompensas de cada acción
- La dificultad de equilibrar entre exploración y explotación es otro problema distintivo de aprendizaje reforzado que no aparece en otros campos de machine learning

Aprendizaje Reforzado

- Considera el problema integral de un agente con un objetivo interactuando con el ambiente y aprendiendo a tomar decisiones
- En ese sentido tanto aprendizaje supervisado como no supervisado son en general parte de sistemas mayores y resuelven un subproblema
- El campo de RL desarrolla técnicas y busca principios generales de aprendizaje que funcionen en este setup general de interacción con el entorno

Videos de algunos ejemplos

- arquero robotico: https://www.youtube.com/watch?v=CIF2SBVY-J0
- robot humanoide: https://www.youtube.com/watch?v=No-JwwPbSLA
- helicoptero: https://www.youtube.com/watch?v=0JL04JJjocc
- blockout: https://www.youtube.com/watch?v=eG1Ed8PTJ18
- space invaders: https://www.youtube.com/watch?v=W2CAghUiofY

Definición del problema

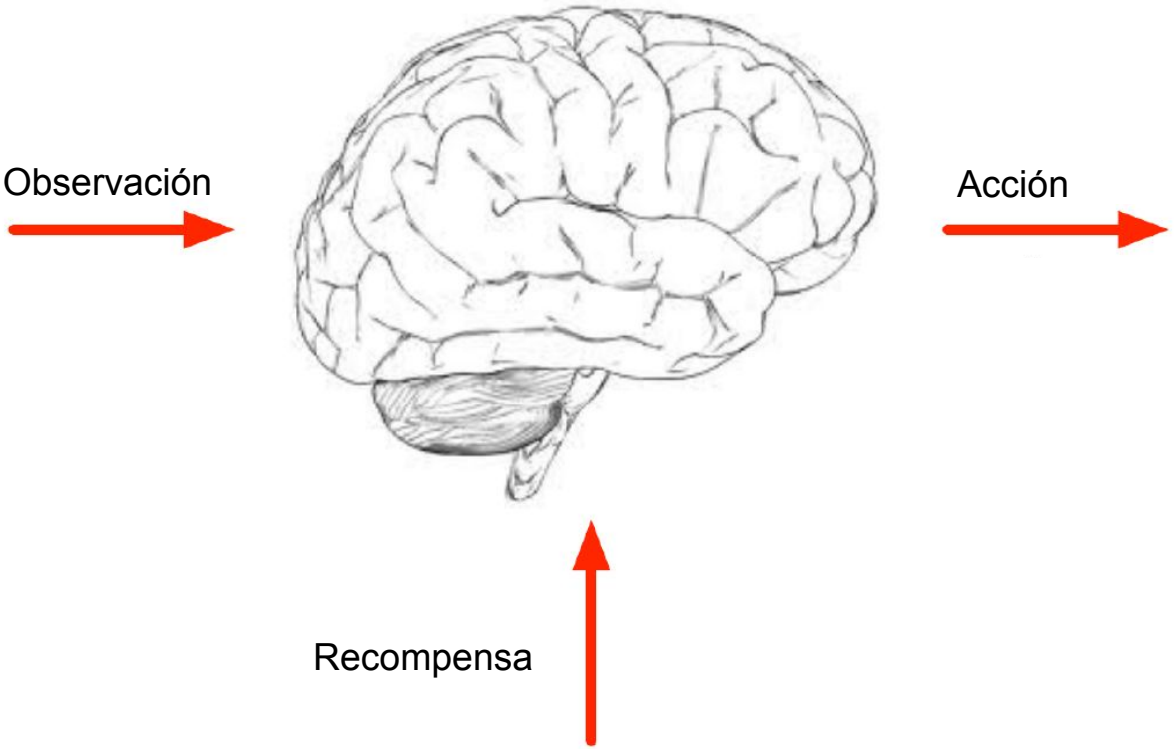
- Un agente interactúa con el ambiente
- El agente puede tomar diferentes acciones
- El agente recibe feedback (señal de recompensa) según las acciones que va tomando, asumimos que todos los posibles objetivos pueden representarse así (reward hypothesis)
- Al tomar acciones también modifica las partes del mundo que explora (“active learning”)
- El objetivo es aprender la política óptima, el mapa observación -> acción, de manera de maximizar la recompensa acumulada
- Puede haber incerteza/azar en cómo se comporta el mundo

Definición del problema - componentes

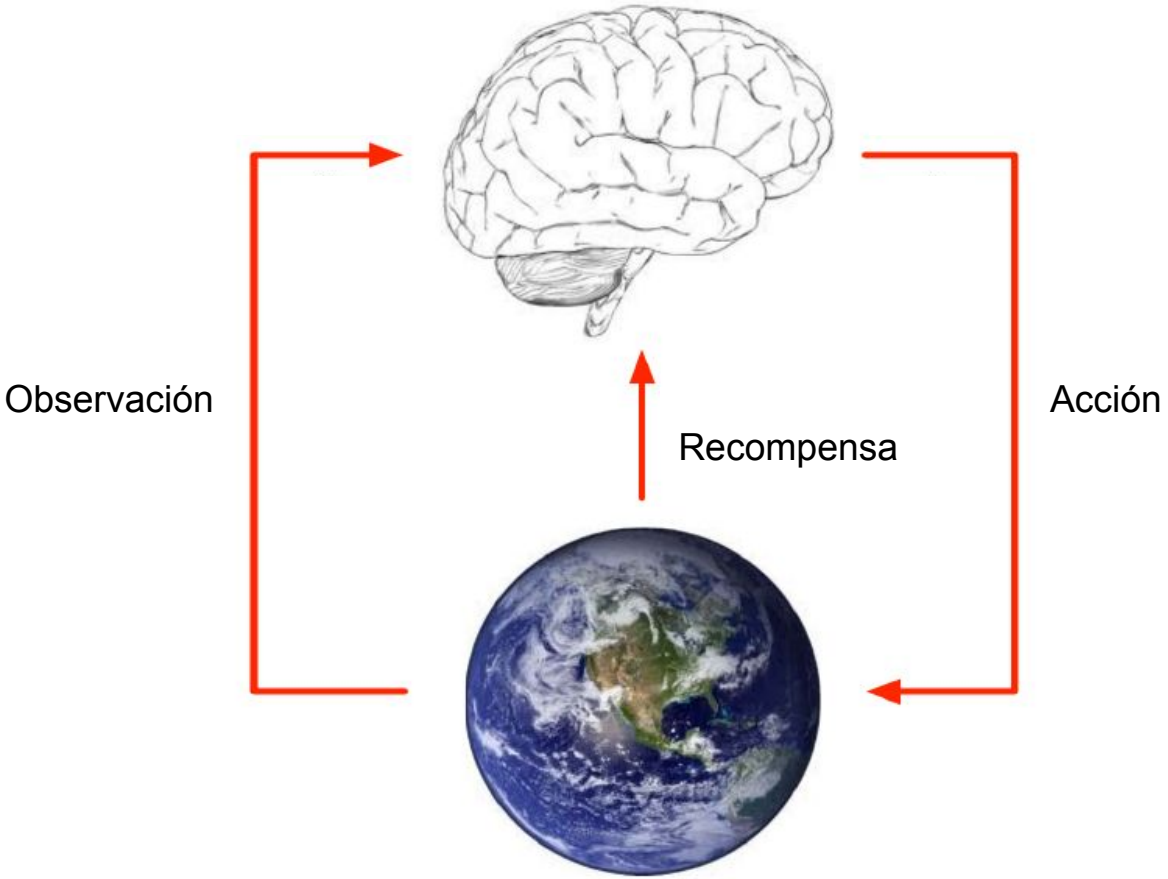
- El ambiente - estados, dinámica de transición (puede ser estocástico)
- Observaciones del ambiente (ídem)
- Acciones posibles en cada estado del ambiente
- Un agente
  - Una política (de acción)
  - Una señal de recompensa
  - (Una función de valor (qué es bueno en el largo plazo))
  - Un modelo del ambiente (opcional), para planificación

Objetivo: maximizar la esperanza recompensas acumuladas

Setup

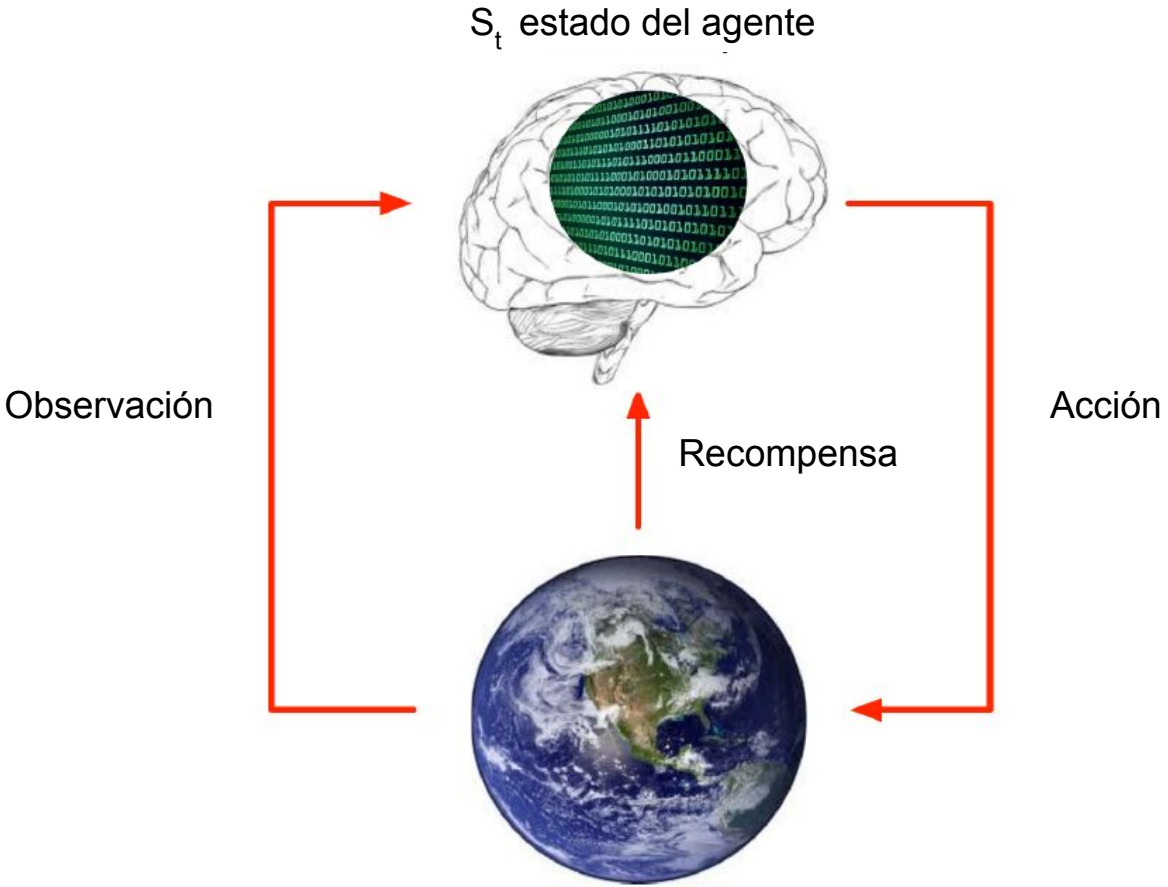


Setup



Setup

Setup



Ejemplos (en cada uno de estos, recompensa, acciones, observaciones?)

- Un jugador de ajedrez, Teg, go, Backgammon, etc.
- Un helicóptero debe realizar piruetas
- Diseñar landing page para maximizar retención
- Chatbots con diversos objetivos: psicólogos, servicio al cliente
- Tratamiento médico personalizado
- Administración de una cartera de acciones
- Robots
- Asistentes de navegación

Más ejemplos (en cada uno de estos, recompensa, acciones, observaciones?)

- Elegir ads para maximizar la ganancia
- Controlar una central de energía
- Jugar juegos de computadora mirando las imágenes
- Una gacela aprende a caminar/correr
- Preparar el desayuno
- El banco central toma decisiones en función de la economía
- Una bacteria se comporta en función del ambiente químico a su alrededor
- Vehículos autónomos

Compañías utilizando Aprendizaje Reforzado

- Deepmind: AlphaGo, AlphaZero, Atari Games, https://deepmind.com/
- Trading algorítmico: Hihedge, https://www.hihedge.com/, https://pit.ai/
- Ambientes de cultivo controlables: Optimal Labs: http://optimal.ag/
- Aprendizaje de robots/vehículos autónomos: http://covariant.ai/, https://www.latentlogic.com/, https://www.osaro.com/, http://prowler.io/, https://www.fanuc.com/
- Análisis de datos: http://intelligentlayer.com/
- Chatbots: https://rasa.com/

Review de Matemática:

- Probabilidad Condicional / Bayes

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Fórmula de Probabilidad Total

Referencias para la parte de probabilidad:

- Brémaud, Pierre. Markov chains: Gibbs fields, Monte Carlo simulation, and queues. Vol. 31. Springer Science & Business Media, 2013.
- Grimmett, Geoffrey, and David Stirzaker. Probability and random processes. Oxford university press, 2001.
- Notas de Sebastián Grynberg (FIUBA).

Review de Matemática:

- Esperanza Condicional

$$E[X] = E[E[X|Y]] = \sum_{y} E[X|Y = y]\, p_Y(y)$$

- Independencia

Esperanza “Recursiva”

Una rata está atrapada en un laberinto. Inicialmente puede elegir una de tres sendas. Si elige la primera se perderá en el laberinto y luego de 12 minutos volverá a su posición inicial; si elige la segunda volverá a su posición inicial luego de 14 minutos; si elige la tercera saldrá del laberinto luego de 9 minutos. En cada intento, la rata elige con igual probabilidad cualquiera de las tres sendas.
Calcular la esperanza del tiempo que demora en salir del laberinto.

Ej 5.13, PyE - FIUBA

Para entregar

Una rata está atrapada en un laberinto. Inicialmente elige al azar una de tres sendas. Cada vez que vuelve a su posición inicial **elige al azar entre las dos sendas que no eligió la vez anterior**. Por la primera senda, retorna a la posición inicial en 8 horas, por la segunda retorna a la posición inicial en 13 horas, por la tercera sale del laberinto en 5 horas.

Calcular la esperanza del tiempo que tardará en salir del laberinto.

Obs: Ya no vale que el experimento “se resetea” después de la primer decisión, pero… un contexto te lleva al otro

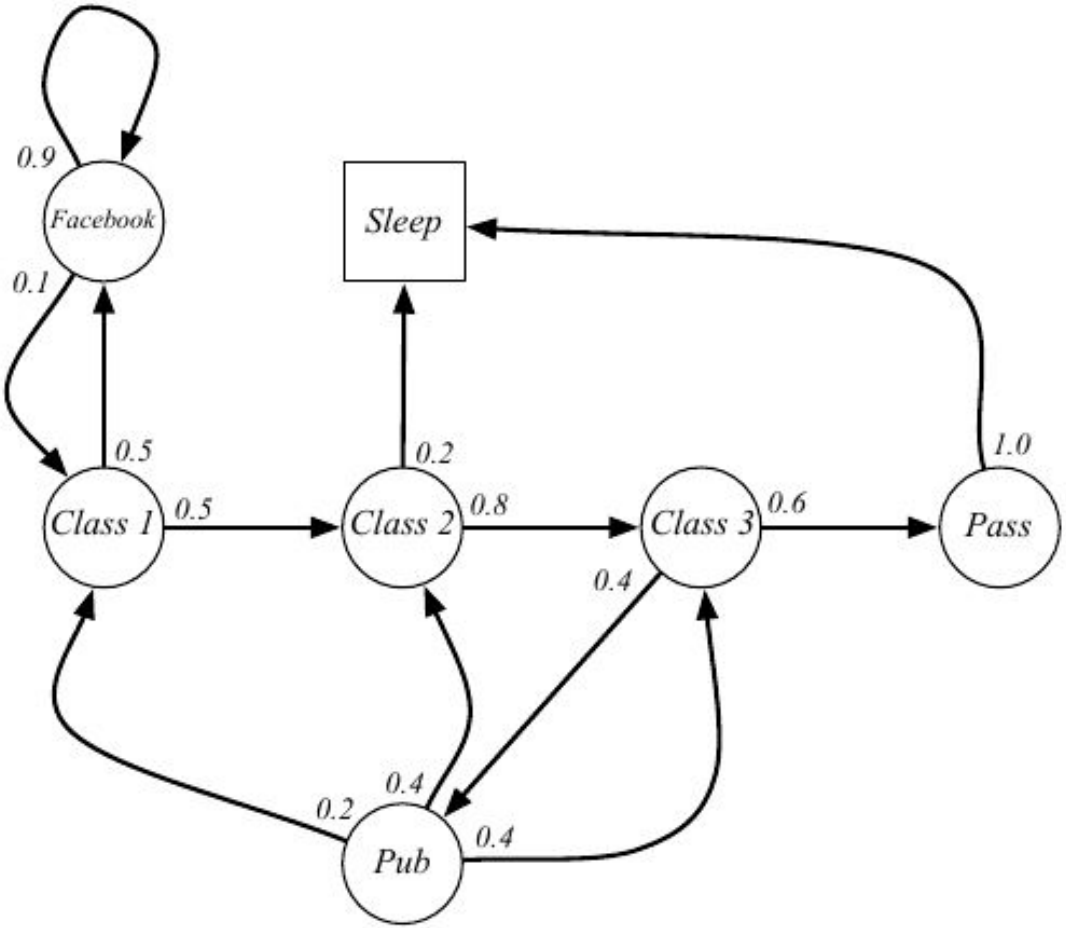
Por si el otro no los deja dormir

La corporación Cobani Products produjo 6 RoboCops, cada uno de los cuales está fallado con probabilidad 1/4. Cada RoboCop es sometido a una prueba tal que si el Robocop está fallado se detecta la falla con probabilidad 4/5. Sea Y la cantidad detectada de RoboCops fallados, calcular E[Y].

Review de Matemática:

- Cadenas de Markov

$$P(X_{n+1} = s' | X_n = s) = p_{ss'}$$


0.9
Facebook
0.1
Sleep
0.5
0.2
1.0
Class 1
0.5
Class 2
0.8
Class 3
0.6
Pass
0.4
0.4
0.2
Pub
0.4

Programación

- Cadena de Markov
- Ejercicio:
  - simular 100 tiradas
  - obtener el tiempo de visita medio de cada estado
  - longitud media de la cadena

Software utilizado

- ubuntu 16.04, Python, jupyter, open ai gym
- Cómo instalarlo?:
  - Linux:
    - sudo apt-get update
    - sudo apt-get install python3 python3-pip ipython3 python3-fontconfig
    - sudo apt-get install libglu1-mesa-dev freeglut3-dev mesa-common-dev python-opengl
    - pip3 install numpy pandas matplotlib jupyter gym
  - Windows: (inspirado en https://github.com/openai/gym/issues/11#issuecomment-242950165)
    - instalar Windows Subsystem for Linux (WSL): https://docs.microsoft.com/en-us/windows/wsl/install-win10
    - instalar ubuntu 16.04 LTS para WSL yendo a Microsoft Store (barra de búsqueda de Windows) y buscando Ubuntu 16.04
    - correr una consola WSL (buscar Ubuntu en la barra de búsqueda de Windows)
    - realizar los mismos pasos que en el instructivo de linux en esa consola
    - instalar vcXsrv/xming;
    - correr vcXsrv (elegir one large window); tipear en la consola de comandos de WSL: export DISPLAY=:0
  - Correr jupyter: jupyter notebook --no-browser
  - Tenerlo instalado para la próxima clase

Lectura recomendada:

- AlphaGo paper: https://ai.google/research/pubs/pub44806
- Brief Survey of Deep RL: https://arxiv.org/pdf/1708.05866.pdf