Resumen del Paper:

Curiosity-Driven Exploration by Self-supervised Prediction

Ignacio Monardes

29 de diciembre de 2025

Referencia

Autores: Deepak Pathak et al.
Título: *Curiosity-Driven Exploration by Self-supervised Prediction*
Conferencia/Journal: ICML 2017
Link: `https://arxiv.org/abs/1705.05363`

1. Motivación

Explica:

- ¿Qué problema aborda el paper? En muchos ambientes las rewards son sparse, por lo que una manera de incentivar al agente a que explore es dar una reward intrinseca por explorar. Se define curiosidad como la diferencia entre las consecuencias predichas y las consecuencias reales por las acciones.
- ¿Por qué es importante? Es importante porque la exploración permite aprender cosas a largo plazo. La exploración permite entender la relación entre las acciones que tomamos y como cambia el mundo.
- ¿Qué limitaciones tienen los enfoques previos? Las limitaciones que se tiene es que hay que definir algo como novedoso. Para eso, necesitamos entender la dinámica del ambiente ($s_{t+1} = f(s_t, a_t)$). Esta dinámica puede ser dificil de predecir en el mundo de las imagenes y sumando la estocacidad se vuelve complejo.

2. Idea Principal

Describe la contribución central del paper en alto nivel:

- ¿Qué proponen?

  Proponen obtener un latente vectorial a partir de la imagen actual. Luego, crear un modelo que dado el latente actual y la acción intente predecir la siguiente imagen. Se da el error de predicción latente como recompensa de curiosidad intrinseca.

- ¿Qué lo hace diferente a trabajos anteriores? Lo hace diferente en el sentido de poder predecir como cambia el mundo en base a las acciones del agente. Para ello, toma el vector latente como error para poder aprender información útil y no imagenes crudas.

3. Metodología

- Modelamiento: Se define la reward como:

$$r_t := r_t^i + r_t^e$$

donde r_t^i es la reward intrinseca por la 'curiosidad' y r_t^e es la reward real del ambiente.

Se define la función *inverse dynamics* g como:

$$\hat{a}_t = g\left(\varphi(s_t),\, \varphi(s_{t+1}) |\, \theta_I\right)$$

como la acción que debió usarse para hacer la transición. Se crea la Loss

$$\text{mín}\ \ L_I(\hat{a}_t,\, a_t)$$

También se crea el modelo *forward dynamics* f:

$$\hat{\varphi}(s_{t+1}) = f(\varphi(s_t), a_t |\, \theta_F)$$

Se entrena mediante:

$$L_F\left(\varphi(s_t),\, \hat{\varphi}(s_{t+1})\right) = \frac{1}{2}\|\hat{\varphi}(s_{t+1}) - \varphi(s_t)\|_2^2$$

Y finalmente se define

$$r_t^i := \frac{\eta}{2}\left\|\hat{\varphi}(s_{t+1}) - \varphi(s_{t+1})\right\|_2^2$$

donde η es un factor que controla la exploración intrinseca.

Finalmente, la función a minimizar es:

$$\min_{\theta_P, \theta_I, \theta_F}\left[-\lambda E_{\pi(s_t;\, \theta_P)}\left[\sum_t r_t\right] + (1-\beta)L_I + \beta L_F\right]$$

donde $\beta \in [0, 1]$ es un número arbitrario que se usa para decidir como escalarlo y λ asigna prioridad a la recompensa.

- Assumptions:

Se usarán las imágenes en blanco y negro y redimensionadas a 42x42.

- Arquitectura:

A3C: 4 Convolutional layers con 32 filtros, kernel size de 3x3, stride 2, padding 1, ELU después de cada convolutional layer. El output de la última layer alimenta un LSTM con 256

unidades. Se usan 2 fully layers para predecir: value function, action de la representación del LSTM.

Intrinsic Curiosity Model (ICM):

- latent $\varphi(s_t)$: 4 conv layers de 32 filtros, kernel size de 3x3, stride 2, padding 1. ELU entre cada conv. $\varphi(s_t) \in \mathbb{R}^{288}$
- inverse dynamic $g(\varphi(s_t), \varphi(s_{t+1}))$: se concatenan los φ's, se pasan por una fully connected de salida 256 y luego a una fully connected de salida 4.
- forward dynaimc $f(\varphi(s_t), a_t)$: Se concatena $\varphi(s_t)$, a_t se pasa por dos fully connected de salidas 256 y 288. $\beta = 0{,}2$, $\lambda = 0{,}1$.

El resultado final se le llama **ICM + A3C**.

4. Experimentos

- Entornos o datasets

  El primer ambiente en el que se entrenó fue en VizDoom que es básicamente Doom. Se juega en el ambiente de 'DoomMyWayHome-v0' donde se debe explorar un mapa 3D y llegar a un destino. Se terminan los episodios con 2100 steps o llegando al objetivo. Se tiene reward sparse de 0 si se mueve y 1 si se llega al objetivo.

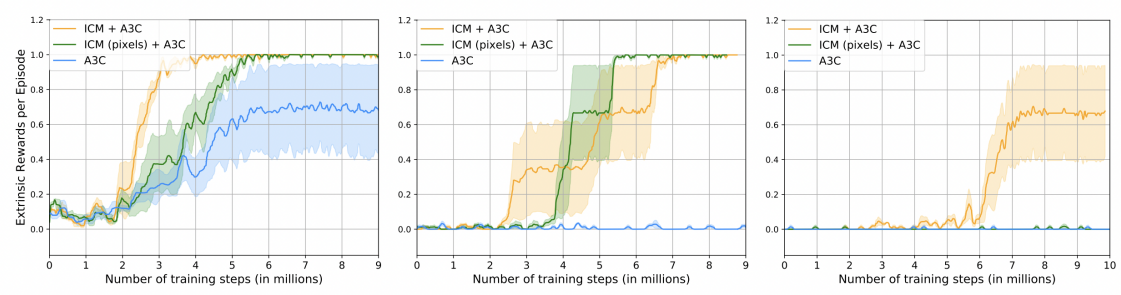
  El segundo ambiente en el que se entrena es en Super Mario Bros. Se tienen 14 acciones de dirección que puede ser ir hacia arriba y derecha al mismo tiempo, etc.

  Las imagenes son en blanco y negro y son redimensionadas a 42x42.

- Baselines

  - A3C.
  - ICM-pixels + A3C: sin inverse dynamics ni latente, solo pixel.
  - VIME/TRPO.

- Settings: Para VizDoom se usan 3 tipos de settings. Dense donde se spawnea en alguna de 17 opciones de spawn posibles. Sparse parte desde el cuarto 14 que está a 270 pasos. Very Sparse parte del cuarto 17 que está a 350 steps.



(a) "dense reward" setting (b) "sparse reward" setting (c) "very sparse reward" setting

El resultado del caso del ICM+A3C es claramente mejor y sobre todo en ambientes sparses. El 66 % de las runs logró aprender en el caso very sparse una buena política.

5. Resultados

Resume los resultados más importantes:

- ¿Supera a los baselines?
- ¿En qué escenarios funciona mejor / peor?

6. Discusión Crítica

Tu análisis:

- Fortalezas
- Debilidades
- Supuestos cuestionables
- Qué no queda claro

7. Conclusiones

Comentario Personal (opcional)

Tu opinión:

- ¿Te parece una buena contribución?
- ¿La usarías en tu investigación?