Resumen del Paper:

Curiosity-Driven Exploration by Self-supervised Prediction

Ignacio Monardes

2 de enero de 2026

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

Problema presente

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.

Relevancia del problema

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.

Agujero en la literatura / limitaciones previas

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

Idea concisa

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.

Innovación

Se diferencia de trabajos previos en 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

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

Ayuda como regualrizador del espacio latente para aprender información relevante a lo que puede controlar o que lo puede afectar. Nota: L_I es cross-entropy para acciones discretas y MSE para acciones continuas.

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.

Nuevos escenarios

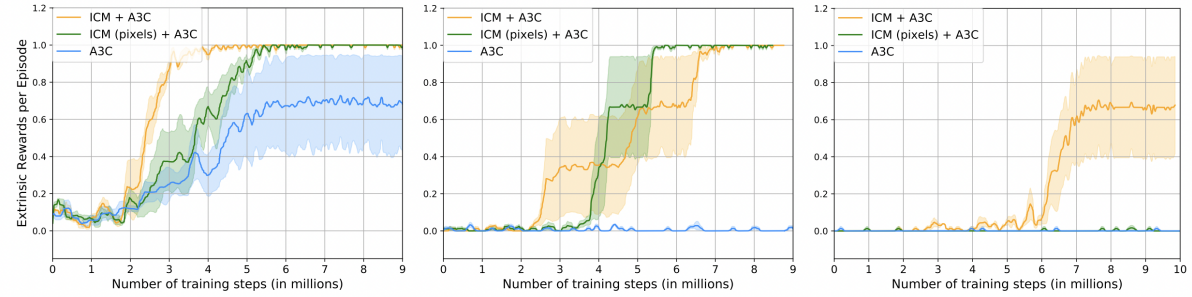
Se plantea si se puede transferir el conocimiento de exploración de un nivel a otro nivel el cual sea relativamente similar. Opciones:

- Aplicar la política sin ningún cambio.
- Fine-tuning con solo curiosidad (intrínseca).
- Adaptar la política a una reward del ambiente (extrínsica).

5. Resultados

Primero se comparó con VizDoom en llegar al lugar objetivo entre A3C, ICM-pixel + A3C y ICM + A3C.

Comparación con el estado del arte o baselines



(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 y aprendió una buena política.

La siguiente tabla muestra los resultados de la transferencia de aprendizaje con y sin fine tunning de la instrinsic reward en diferentes niveles.

Level Ids	Level-1	Level-2				Level-3			
Accuracy Iterations	Scratch 1.5M	Run as is 0	Fine-tuned 1.5M	Scratch 1.5M	Scratch 3.5M	Run as is 0	Fine-tuned 1.5M	Scratch 1.5M	Scratch 5.0M
Mean ± stderr	711 ± 59.3	31.9 ± 4.2	466 ± 37.9	399.7 ± 22.5	455.5 ± 33.4	319.3 ± 9.7	97.5 ± 17.4	11.8 ± 3.3	42.2 ± 6.4
% distance > 200	50.0 ± 0.0	0	64.2 ± 5.6	88.2 ± 3.3	69.6 ± 5.7	50.0 ± 0.0	1.5 ± 1.4	0	0
% distance > 400	35.0 ± 4.1	0	63.6 ± 6.6	33.2 ± 7.1	51.9 ± 5.7	8.4 ± 2.8	0	0	0
% distance > 600	35.8 ± 4.5	0	42.6 ± 6.1	14.9 ± 4.4	28.1 ± 5.4	0	0	0	0

Table 1. Quantitative evaluation of the agent trained to play Super Mario Bros. using only curiosity signal without any rewards from the game. Our agent was trained with no rewards in Level-1. We then evaluate the agent's policy both when it is run "as is", and further fine-tuned on subsequent levels. The results are compared to settings when Mario agent is train from scratch in Level-2,3 using only curiosity without any extrinsic rewards. Evaluation metric is based on the distance covered by the Mario agent.

Se puede apreciar que IMC+A3C tiene una buena capacidad de transferencia sobre nuevos ambientes. Para el caso de mario Level 3 el hacerle el fine tunning hace que le vaya peor por explorar partes inútiles.

El siguiente gráfico es un finetunning de un extrinsic reward. Se preentrena un agente para explorar y luego se finetunea para conseguir recompensa en un mapa diferente con texturas nuevas.

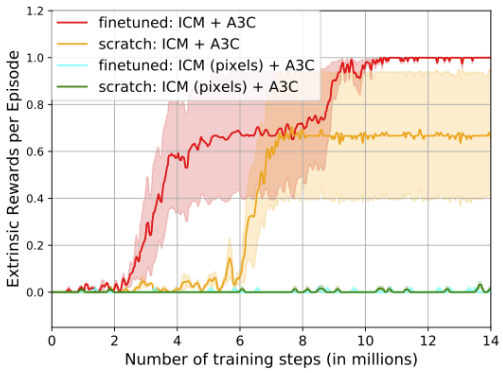


Figure 8. Performance of ICM + A3C agents on the test set of *VizDoom* in the "very sparse" reward case. Fine-tuned models learn the exploration policy without any external rewards on the training maps and are then fine-tuned on the test map. The scratch models are directly trained on the test map. The fine-tuned ICM + A3C significantly outperforms ICM + A3C indicating that our curiosity formulation is able to learn generalizable exploration policies. The pixel prediction based ICM agent completely fail. Note that textures are also different in train and test.

¿En qué escenarios funciona mejor / peor?

Funciona bien en ambientes de episodios largos de exploración de forma sencilla. Es bueno recreando imagenes.

Funciona mal cuando se requiere aprender una política de acciones para realizar cosas específicas como hacer 15 pasos para hacer un salto especializado. Se complementa bien con una recompensa intrinseca para poder avanzar.

6. Discusión Crítica

Fortalezas

Una señal intrinseca de exploración que no depende de heurísticas ni de conteo de estados visitados. El uso de un espacio latente aprendido permite que al recompensa de curiosidad sea robusta a cambios visuales irrelevantes y ruido estocástico. Compatible con métodos existentes de RL.

Buena transferencia de conocimiento.

Debilidades

La recompensa intrinseca no permite aprender una política de movimientos necesarios sino una exploración de como moverse más bien. No siempre lo novedoso es lo mejor, a veces es mejor conseguir la política buscada en vez de explorar.

Supuestos cuestionables

Capacidad del espacio latente aprendido. Uso de imagenes en gris y no en color. Podría variar el η de exploración durante el entrenamiento y no dejarlo fijo.

Qué no queda claro

Aporte en ambientes con recompensas densas. Impacto computacional.

7. Conclusiones

Buen aporte para incentivar la exploración en ambientes basados en imagenes con diferentes texturas. Buena transferencia de conocimientos en diferentes juegos/objetivos.

Falta mejorar el control fino de acciones.

Comentario Personal

¿Es una buena contribución?

Muy buena contribución. Logra objetivos más dificiles cambiando la recompensa y aprendiendo una versión comprimida del estado.

¿Lo usaría en mi investigación?

Sí, está asociado a intentar modelar el mundo de forma interna para poder predecir las imágenes mediante un espacio latente compacto.