Hierarchical Reinforcement Learning: Maze with Tasks

Rubén Cid Costa[1], **Aimar Nicuera Usandizaga**[2], **Daniel Obreo Sanz**[3]

[1, 2, 3]Universidad Carlos III de Madrid
[1]100538592@alumnos.uc3m.es, [2]100538592@alumnos.uc3m.es, [3]100538592@alumnos.uc3m.es

Abstract

La realización de objetivos secuenciales en un entorno es una tarea compleja de modelar y aprender. Para poder representar estos escenarios, una de las técnicas más usadas es el Aprendizaje por Refuerzo Jerárquico (*Hierarchical Reinforcement Learning (HRL)*). En este contexto, este trabajo se enfoca en Feudal Learning, una variante de HRL que organiza las tareas en una estructura jerárquica de niveles de abstracción. Este documento detallará las bases teóricas y su aplicación sobre un ambiente de desarrollo con tareas de navegación y obtención de subobjetivos.

Introducción

En muchos dominios, como la robótica o sistemas autónomos, las tareas implican la realización de objetivos secuenciales en entornos complejos. La capacidad de modelar y aprender estos escenarios es un desafío crucial para la inteligencia artificial.

El aprendizaje por refuerzo jerárquico (HRL) se presenta como un enfoque para la resolución de estos problemas. Mientras que otras técnicas previas enfrentan dificultades para escalar con el número de tareas y su complejidad, el Aprendizaje por Refuerzo Jerárquico (HRL) organiza el proceso en niveles de abstracción. Dentro de este marco, el Aprendizaje Feudal se presenta como un enfoque que permite modelar las tareas mediante un jerarquía de abstracción.

Este trabajo se centra en el estudio de Feudal Learning, una variante de HRL. Se presentan las bases teóricas de esta técnica como diferentes algoritmos o métodos de aprendizaje que se han desarrollado en este campo. También, se mostrará su aplicación sobre un entorno simulado diseñado para tareas de navegación en laberintos y obtención de subobjetivos.

Marco Teórico y Estado del Arte

El aprendizaje por refuerzo jerárquico (HRL por sus siglas en inglés) se ha consolidado como una estrategia para abordar problemas complejos que involucran secuencias de tareas. Este enfoque se originó como una extensión de los modelos de decisión de Markov para tareas de largo horizonte de decisión mediante el uso de modelos semi-MDP(Sutton, Precup, and Singh 1999). Con ellos, se establecen las bases teóricas para la abstración temporal.

Posteriormente, con los años, han surgido una serie de algoritmos que implementa y abordan las limitaciones de HRL. Dietterich (Dietterich 2000) propuso el modelo MaxQ, descompone las tareas en jerarquías de subtareas para la optimización y planificación de diferentes funciones de valor. Por otro lado, en (Barto and Mahadevan 2003) se exploran los avances en estructuras jerárquicas para diferentes dominios y la relevancia de HRL para la resolución de problemas de gran complejidad.

Dentro de este marco teórico, el Aprendizaje Feudal se presenta como un técnica que organiza las tareas en niveles jerárquicos, donde cada nivel es operado con distintos grados de abstracción tanto en el estado como en temporalidad. Este enfoque (Dayan and Hinton 1992) divide la jerarquía en managers, que operan con abstracciones del estado y dan ordenes de acción y en workers que realizan las acciones. Esta colaboración jerárquica permite aprender y ejecutar tareas de manera más eficiente, aprovechando la modularidad y la capacidad de escalar a problemas complejos.

Sistemas de Control Feudal

En 1992, P. Dayan y G. Hinton (Dayan and Hinton 1992) definen los sistemas de control feudal como un reflejo de las sociedades feudales. En ellos, se define una jerarquía de *managers*, *supermanagers* y *workers*. Cada uno opera con un grado de abstracción de estados y temporal distinto. Los managers tienen poder absoluto sobre sus subordinados. Pueden dar ordenes que deben ser seguidas, dan tareas, recompensas y castigos si las ordenes no son seguidas. Esta estructura de jerarquíca de mando permite aprender y comprender tareas complejas con el fin de maximizar el refuezo.

Este sistema de control se basa en dos principios:

- **Ocultamiento de la Recompensa (Reward Hiding)**. Los *managers* recompensan a los *submanagers* únicamente si operan en consonancia con las ordenes dadas. Los *submanagers* deben aprender a obedecer y aprender que debe hacer el siguiente agente para cumplir de manera eficiente. De la misma manera, el *submanager* es recompensado si su subordinado ejecuta las ordenes a pesar de no llegar a cumplir las ordenes propias del *submanager*.

- **Ocultamiento de la Información (Information Hiding)**. Los *managers* solamente deben conocer la información necesaria con un nivel de granularidad distinto a la de los *workers*. Las decisiones se toman en base a un espacio de estados menor pero más denso en información. También, las órdenes dadas a un *submanager* no son propagadas hacia el resto de niveles para no afectar la toma de decisiones. Tampoco, se propagan hacia arriba las acciones del *worker* hacia el resto de *managers*.

Estos principios siguen una dinámica de poder y gestión similar a la de sociedades feudales con diferente interés en cada estramento de la organización. No obstante, todos se alinean para la obtención y maximización de una recompensa (o refuerzo) dado por el problema.

Para el aprendizaje e implementación, se propusó el uso de una versión modificada del Aprendizaje Q-Learning. Cada *manager*, *submanager* y *worker* tiene asociada una función acción-valor (*Q Tabla*). La principal diferencia con respecto el algoritmo original es la modificación de la función de recompensa. En este caso, se modifica en función de las acciones tomadas por cada nivel de la jerarquía en consonancia con su nivel de obediencia.

FeUdal Networks: FuN

Las redes feudales se basan en la arquitectura de aprendiaje por refuerzo feudal, una arquitectura del aprendizaje por refuerzo jerárquico. Esta arquitectura emplea un sistema de control, conocido como *manager*, que asigna tareas a un subsistema conocido como *worker* que debe aprender a ejecutarlas de manera óptima.

La arquitectura del sistema se muestra en la imagen siguiente (Science 2024).

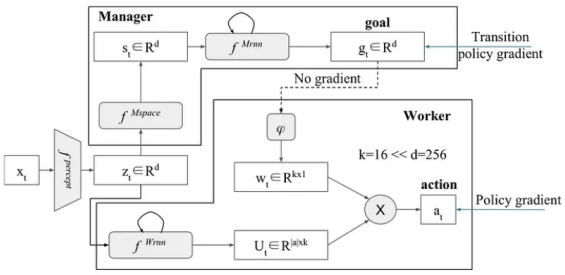

Figure 1: Arquitectura de una red feudal.

La entrada de esta red es procesada por una capa de percepción, que emplea capas convolucionales para extraer características de la imagen de entrada. A continuación, estas características son procesadas tanto por el worker como por el manager, cada uno de manera distinta; el manager extrae objetivos y el worker aprende a alcanzar esos objetivos.

El objetivo principal del manager es generar metas que el worker debe cumplir. Recibe la percepción del entorno, proporcionada por el módulo de percepción, ese estado es procesado por una red recurrente LSTM para mantener un estado interno y poder capturar información relevante en horizontes temporales largos. El manager emplea esta información para predecir un objetivo direccional en el espacio latente, este objetivo es un vector unitario, lo que asegura que el worker se enfoque en la dirección y no en la posición absoluta.

Para entrenar el manager, se emplea la recompensa obtenida, y emplea la similitud coseno entre la dirección en la que se movió el worker y la compara con el objetivo establecido, empleando la similitud coseno como función de pérdida. Esta pérdida incentiva al Manager a emitir objetivos que maximicen el progreso hacia estados ventajosos.

El vector de objetivos se envía al worker sin propagar gradientes, esto garantiza que los objetivos mantengan un significado semántico independiente, en lugar de ser simples variables latentes optimizadas de manera conjunta.

En el caso del worker, también se emplea una red LSTM para mantener un estado interno y poder capturar información relevante, pero en este caso, el worker recibe tanto la percepción del entorno como el objetivo del manager. El worker emplea esta información para predecir la acción que debe realizar para alcanzar el objetivo. La acción se predice en el espacio de acciones, y se emplea

Definición

Estado del arte

Evaluación práctica

Debido a la disipación de las paredes que se experimentaba al realizar las abstracciones del mapa de estados a medida que se disminuía el nivel del gerente, el aprendizaje resultaba altamente perjudicado llegando trasmitirse ordenes imposibles de realizar debida a la naturaleza cerrada del mapa del problema.

Para enfrentar dicha adversidad se probó a disminuir el número de gerentes/subgerentes, limitándose al uso del gerente de menor nivel de abstracción; aquel que veía el mapa de estados en la escala original. Por lo tanto, se realizó una disminución de agentes para quedarse con únicamente 2: un gerente encargado de determinar la siguiente zona destino y otro agente encargado en navegar hasta dicha posición de interés. Dicha estructura contaba con las mismas limitaciones y políticas de recompensa definidas para el proceso anterior.

Para facilitar el arranque y el uso del mánager desde el principio, se facilitaba al mánager una pista de la localización de las monedas que debían ser recogidas para que guiara al trabajador (o worker) hacía dichas zonas sin necesidad de un proceso inicial donde las directrices del worker podían ser "ignoradas" debida a su aleatoriedad.

El trabajador a su vez realizaba un proceso de exploración que se veía recompensado cuando alcanzaba el punto objetivo determinado por el mánager. Mediante estas exploraciones, iba creando sus caminos o matrices Q que iba actualizando y afinando con el fin de encontrar el camino óptimo a partir de las recompensas obtenidas por llegar a los destinos asignados por el manager.

En cuanto al proceso de aprendizaje, se incluyeron diferentes implementaciones para ver la influencia que estas variaciones o perturbaciones tenían en el proceso de aprendizaje del super-agente. Las perturbaciones más comunes consistían en limitar el conocimiento que tenía el teacher (dándole solo el conocimiento de la solución empezando

desde las coordenadas (1,1) o el conocimiento completo) o variando la posición de comienzo del agente.

En cuento a los resultados se observó como sin ayudas externas el proceso de aprendizaje era más lento al tener que explorar muchos estados y no contar con una ayuda externa en caso de caer en bucles debido a los estrechos caminos que componían el mapa. Cuando se le introducía la ayuda externa, teacher, dicho proceso de aprendizaje se veía acortado siendo capaz de definir la ruta optima con mayor rapidez. Sin embargo, la influencia de un teacher con conocimiento de un solo camino ante el que siempre tenía la solución optima suponía una gran diferencia también en los momentos en los que se partía de zonas inexploradas. Esto se debía a que el agente tenía que lograr llegar hasta le punto más cercano de conocimiento del teacher para poder obtener su ayuda, lo que lo ponía en similar situación que cuando no contaba con ayuda externa.

En conclusión, esta nueva estructura feudal, de un noble (mánager) y un subordinado (worker), supuso una mejora en comparación con el nivel anterior debida a la simplicidad y mayor énfasis en las limitaciones del mapa de estados que permitía definir. Mediante los experimentos que se realizaron se observó no solo una mejora en el proceso de aprendizaje si no que una capacidad de llegar a soluciones en mapas de 20x20 tras 9000 iteraciones en caso de no incluir la ayuda del teacher o 1000 iteraciones cuando se le facilitaba un teacher para poder preguntar (limitando el número de consultas a realizar en 3xP; siendo P los pasos del camino más corto para solucionar el problema).

Conclusiones

References

Barto, A. G.; and Mahadevan, S. 2003. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, 13(1-2): 41–77.

Dayan, P.; and Hinton, G. E. 1992. Feudal reinforcement learning. *Advances in neural information processing systems*, 5.

Dietterich, T. G. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13: 227–303.

Science, T. D. 2024. Hierarchical Reinforcement Learning: Feudal Networks.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2): 181–211.