Artículo de Investigación. Revista Ciencia e Ingeniería. Vol. 41, No. 1 pp. 67-74, diciembre-marzo, 2020.
ISSN 1316-7081. ISSN Elect. 2244-8780 Universidad de los Andes (ULA)

Uso de sistemas multiagentes para el aprendizaje automático

Using multi-agent systems for machine learning

González Pérez, Yuleisy*; Kholod, Ivan Ivanovich
Facultad de Tecnologías Computacionales e Informática, Universidad Electrónica, San Petersburgo, Rusia
* yuleisy2688@gmail.com

Resumen

La interacción entre los sistemas informáticos y el aprendizaje que algunos de ellos son capaces de lograr es vital, los cambios son visibles en la forma tradicional de analizarlos y desarrollarlos. La necesidad de interactuación entre los componentes del sistema es cada vez más importante para resolver tareas conjuntas que individualmente serían muy costosas o incluso imposibles. Los sistemas multiagentes ofrecen una amplia plataforma para realizar tareas distribuidas que cooperan entre sí, pero también permiten incluir en cada agente un comportamiento dotado de inteligencia que puede desarrollarse a través de técnicas de aprendizaje automático. El método aprendizaje por refuerzo insertado en el aprendizaje automático es muy útil para su uso con los agentes, permite a los agentes aprender a través de la interacción de muestras y errores en un entorno dinámico. Este documento aborda conceptos, características, relaciones y otros aspectos importantes del uso de sistemas multiagentes para el aprendizaje automático.

Palabras claves: agente, aprendizaje automático, sistemas multiagentes.

Abstract

The interaction between computer systems and the learning that some of them are able to achieve is vital, the changes are visible in the traditional way of analyzing and developing them. The need for interaction between system components is increasingly important in resolving joint tasks that would be individually very costly or even impossible. Multiagent systems offer a broad platform for performing distributed tasks that cooperate with each other, but also allow for the inclusion in each agent of intelligent behavior that can be developed through automatic learning techniques. The reinforcement learning method inserted in automatic learning is very useful for use with agents, allows agents to learn through the interaction of samples and errors in a dynamic environment. This document addresses concepts, characteristics, relationships and other important aspects of the use of multiagent systems for automatic learning.

Keywords: agent, machine learning, multiagent systems.

1 Introducción

El progreso en la tecnología y las comunicaciones ha llevado a la apertura de un modelado del pensamiento humano. En este esfuerzo, una de las ciencias que más destaca es la Inteligencia Artificial (IA). Una de las disciplinas de la IA es el Aprendizaje Automático para la creación de sistemas de aprendizaje automático, su nombre en inglés es "Machine Learning" (ML).

ML se refiere al estudio y la construcción de algoritmos para la investigación y la predicción. Estas previsiones se pueden considerar clasificaciones de entrada basadas en el reconocimiento de las plantillas existentes en ellas. Existen diferentes métodos de clasificación del ML que varían en el tipo objetivo de aprendizaje (por ejemplo, matemático, estadístico, biológico), eficiencia y complejidad. Por ejemplo: redes neuronales artificiales, reglas de asociación, máquinas de soporte de vectores, árboles de decisión, redes bayesianas y análisis de clústeres (Amigone y col., 2017).

Los agentes inteligentes actúan como parte del software que se puede ejecutar sin control humano para lograr los objetivos proporcionados por los usuarios. El agente gestiona, procesa y obtiene información sobre su entorno y puede interactuar con otros agentes. En este caso, los modelos y lenguajes formales proporcionan mecanismos de abstracción y de representación de dicha información (Amaolo y col., 2017). Del mismo modo, los lenguajes y métodos formales básicos del ML proporcionan formas mejoradas de gestionar la información y obtienen nuevos conocimientos, definiendo así un entorno adecuado para hacer frente a problemas reales y, por lo tanto, complejos.

Por su parte, los Sistemas Multiagentes (SMA) representan un área de crecimiento continuo para el desarrollo de aplicaciones comerciales e industriales a gran escala, ya que proporcionan una solución más natural a los problemas complejos. En este tipo de sistemas, cada agente tiene capacidades limitadas e información incompleta sobre su entorno (Agis y col., 2017). En la actualidad, estos sistemas se utilizan para resolver problemas complejos en IA y otras disciplinas. En los últimos años, su campo de aplicación ha aumentado, principalmente debido a los desarrollos teóricos y prácticos en el campo de la Inteligencia Artificial Distribuida (IAD).

2 Método

A continuación, se muestran los métodos utilizados para el estudio realizado:

- Métodos teóricos:

Histórico – lógico: en la realización del estudio del estado del arte para investigar la relación que existe y cómo se utilizan los SMA para el ML.

Análisis – síntesis: para el estudio de las fuentes bibliográficas existentes referente al tema, identificando los elementos más importantes y necesarios para dar solución al problema planteado.

Inductivo – deductivo: con el fin de detectar las principales iniciativas y tendencias en cuanto al uso de SMA para el ML, soluciones existentes y SMA que hacen uso de métodos del ML, con el objetivo de determinar cuáles son las alternativas viables a incorporar en la presente investigación.

- Métodos empíricos:

Análisis documental: en la consulta de la literatura especializada en las temáticas afines a la investigación.

3 Marco Teórico

3.1 Aprendizaje automático

Autores como Arthur Samuel concluyen que *"el ML brinda a los equipos la oportunidad de aprender sin ser programados explícitamente para crear algoritmos que puedan aprender y hacer proyecciones de datos"*. El objetivo del ML es, según Javier Di Deco Sampedro: crear un sistema capaz de dar una respuesta satisfactoria al ingresar información en él. Dependiendo de la retroalimentación recibida por el sistema, se destacan varios paradigmas (Sampedro 2012).

Estudiar en este contexto significa identificar patrones complejos en millones de datos. Una máquina que realmente aprende es un algoritmo que analiza los datos y puede predecir el comportamiento futuro. Por lo tanto, el ML es un proceso de conocimiento de inducción, es decir, un método que le permite obtener a través de la generalización, un operador general de las afirmaciones que describen casos específicos. El ML aprende sobre los datos detectando la estructura y los patrones subyacentes. El objetivo principal del ML es extraer la información contenida en el conjunto de datos para obtener conocimientos que permitan tomar decisiones sobre nuevos conjuntos de datos. Formalmente se define así (Castillo 2015):

"El sistema aprende de la experiencia E con respecto al conjunto de tareas T y el indicador de rendimiento R, si su rendimiento en T, medido de acuerdo con R, mejora de acuerdo con la experiencia E"

En muchos casos, el campo de acción del ML se superpone con el ámbito de la minería de datos, ya que las dos disciplinas se centran en el análisis de datos. La minería de datos detecta patrones previamente desconocidos, mientras que el ML se usa para reproducir patrones conocidos y generar proyecciones basadas en patrones. Por lo tanto, se concluye que el procesamiento de datos inteligentes tiene una función de investigación, mientras que el ML se centra en la predicción.

Los algoritmos de aprendizaje se basan en una serie de datos para aprender y luego aplicar la experiencia adquirida en otros conjuntos. Es necesario evaluar su rendimiento en otro conjunto al que el sistema ha sido entrenado para obtener una evaluación confiable de su capacidad de generalización antes de nuevos ejemplos. Un conjunto de datos disponible se divide en dos subconjuntos: un conjunto de entrenamiento y un conjunto de verificación. Por lo tanto, el modelo se crea a partir de los datos de aprendizaje y se evalúa en un conjunto de prueba que medirá la precisión del modelo. El resultado obtenido en este conjunto es una buena aproximación a lo que se espera obtener para los nuevos datos (Castillo 2015).

En términos matemáticos, hay dos tipos de problemas en el ML, dependiendo del tipo de objetos que desea predecir: problemas de clasificación y problemas de regresión.

3.1.1 Métodos de aprendizaje automático

La perspectiva del ML difiere del tipo de retroalimentación que el crítico proporciona al alumno. Sobre la base de este criterio, los principales métodos de aprendizaje son: aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semi supervisado, aprendizaje por refuerzo, transducción y aprendizaje multitarea (Khaled y col., 2015) (Godoy 2017):

- Aprendizaje supervisado

En el aprendizaje supervisado, el objetivo es investigar el cumplimiento de las respuestas correctas para las entradas que se le proporcionan; esto utiliza un conjunto de datos de aprendizaje, conexiones de pares que consisten en los patrones de entrada y salida correctos. Por lo tanto, el sistema le enseña a mostrar la salida correcta para cada patrón de entrada representado en él (Chapelle y col., 2009). Algunos ejemplos de técnicas de aprendizaje supervisado son (Torunoğlu y col., 2011): árboles de decisiones, máxima entropía, redes bayesianas, máquinas de soporte de vectores. Las técnicas de supervisión del ML son ampliamente utilizadas principalmente en tareas: clasificación, regresión, recomendación y aplicación.

- Aprendizaje no supervisado

En algoritmos de aprendizaje no supervisado, no se requiere intervención humana para desarrollar un conjunto de datos previamente clasificados para representar el algoritmo

de aprendizaje. El objetivo de la enseñanza no supervisada es encontrar modelos interesantes teniendo en cuenta la distribución y la compilación de los datos que se presentan (Chapelle y col., 2009). Ejemplos de técnicas de aprendizaje no supervisado son las técnicas de agrupación.

- Aprendizaje semi supervisado

Los algoritmos de aprendizaje semi supervisado son un término entre el aprendizaje supervisado y no supervisado. En este tipo de aprendizaje, los datos se dividen en dos partes: un grupo de datos clasificados y un grupo de datos no clasificados. Esto se llama aprendizaje semi supervisado estándar.

Para Olivier Chapelle, Bernhard Schölkopf y Alexander Zien, el aprendizaje semi supervisado es más útil cuando hay más datos no clasificados que los datos clasificados. Especialmente cuando se necesita mucho esfuerzo para obtener datos clasificados, se tarda mucho tiempo o es muy costoso; porque, además, obtener datos no clasificados generalmente es más barato (Chapelle y col., 2009). Ejemplos de técnicas de aprendizaje semi supervisado son: las técnicas de máquinas de vectores de soporte transductivo, maximización de las expectativas. Algunas aplicaciones de aprendizaje semi supervisado mencionadas por los autores son: el reconocimiento de voz, la clasificación de páginas web y secuencia de la proteína.

- Aprendizaje por refuerzo

En el aprendizaje por refuerzo, el algoritmo aprende observando el mundo que lo rodea. Su información de entrada es la retroalimentación que recibe del mundo exterior en respuesta a sus acciones. Por lo tanto, el sistema aprende sobre la base de pruebas y errores. En lugar de un instructor que le diga al agente qué hacer, el agente inteligente debe aprender cómo el medio ambiente se comporta a través de la recompensa (refuerzos) o castigos, como resultado del éxito o fracaso, respectivamente. El objetivo es aprender la función de valor que ayuda al agente inteligente para maximizar la señal de recompensa y, por lo tanto, optimizar sus políticas para comprender el comportamiento del entorno y tomar las decisiones correctas para lograr sus objetivos formales.

Los principales algoritmos de aprendizaje por refuerzo se desarrollan como parte de los métodos de resolución de problemas de decisión finitos de Markov, que incluyen ecuaciones Bellman y las funciones de valor. Los tres métodos principales son: Programación Dinámica, métodos de Monte Carlo y aprendizaje de Diferencias Temporales (Sutton y col., 2017).

Entre las implementaciones desarrolladas se encuentra AlphaGo, es un programa de IA desarrollado por Google DeepMind para jugar el juego de mesa Go. En marzo de 2016, AlphaGo ganó el partido ante el jugador profesional Lee Se-Dol, que tiene la categoría novena dan y 18 títulos mundiales. Entre los algoritmos utilizados se encuentran el árbol de búsqueda de Monte Carlo. También utiliza el aprendizaje profundo con redes neuronales.

- Transducción

Al igual que el aprendizaje supervisado, pero no constituye explícitamente una función, ya que los datos no tienen una etiqueta, es sin clasificar. Están diseñados para predecir las categorías de ejemplos futuros basados en los ejemplos de entrada, sus categorías y los nuevos ejemplos en el sistema (Mondeleón y col., 2017).

- Aprendizaje multitarea

Los métodos de aprendizaje que utilizan el conocimiento previamente aprendido por el sistema cuando se enfrentan a problemas similares a los ya vistos. Esto implica la solución simultánea de diferentes tareas; en particular, el aprendizaje de la tarea se mejora y se complementa con el aprendizaje común de otras tareas relacionadas con la primera (Mondeleón y col., 2017).

Las ventajas más importantes de usar el ML, en comparación con las alternativas más comunes de análisis manual, reglas comerciales codificadas y modelos estadísticos simples son las siguientes (Brink y col., 2017): precisión, automatización, rapidez, personalización, escalabilidad.

3.2 Inteligencia artificial distribuida

La IAD se ha definido como un subcampo de la IA que ha adquirido una importancia significativa debido a su capacidad para resolver problemas complejos del mundo real. Se centra en los comportamientos inteligentes colectivos que son el producto de la colaboración de diversas entidades llamadas agentes. La investigación principal en el campo de la IAD incluye tres direcciones diferentes: la IA paralela, la Solución Cooperativa de Problemas Distribuida (SCPD) y los SMA (Balaji y col., 2010).

La IA paralela se refiere principalmente a las metodologías utilizadas para facilitar los métodos clásicos de IA cuando se aplica a arquitecturas de hardware distribuidas, como cálculos multiprocesadores o en clúster. El objetivo principal de la IA paralela es aumentar la velocidad de trabajo y trabajar en hilos paralelos para encontrar una solución global a un problema específico.

La SCPD es similar a la IA paralela y considera cómo se puede resolver el problema mediante la asignación de recursos y conocimientos entre una gran cantidad de módulos interactivos conocidos como un objeto informático. Al resolver problemas distribuidos, la relación entre los objetos de cálculo, el número de información que se pasa se predefinen y se incrustan en el diseño del objeto de cálculo. La SCPD es difícil debido a las estrategias integradas y, por lo tanto, ofrece poca flexibilidad o falta de ella.

A diferencia de la SCPD, los SMA manejan el comportamiento de los objetos informáticos disponibles para resolver este problema. En los SMA, cada objeto de información se denomina agente.

3.3 Agente

Un agente puede ser un sistema reactivo o deliberativo que tiene un cierto grado de autonomía en el sentido de que una tarea puede ser delegada a él, y el mismo sistema determina la mejor manera de realizar esa tarea. Tales sistemas se denominan agentes, ya que se consideran seres activos, productores útiles de acciones: se envían a su entorno para alcanzar las metas y persiguen activamente esos objetivos, descubriendo por sí mismos la mejor manera de lograr estos objetivos (Mateus 2015).

El agente se define como una entidad computacional libre que resuelve un problema. Además, es capaz de realizar acciones flexibles en un entorno abierto, dinámico e impredecible. A menudo, se desarrollan en entornos en los que interactúan y cooperan con otros agentes, teniendo en ocasiones intereses contradictorios. Difieren de los objetos (en el sentido de la programación orientada a objetos) en que son entidades autónomas capaces de tomar decisiones sobre sus acciones e interacciones. Los agentes no pueden ser invocados directamente como objetos. Sin embargo, se pueden construir utilizando tecnologías orientadas a objetos (Herrero 2015).

Un agente es una unidad de computación que percibe su entorno a través de sensores y actúa en ese entorno utilizando efectores. Es autónomo porque controla su comportamiento y puede actuar sin la intervención de otros agentes o personas. Actualmente, los agentes tienen un alcance muy amplio y hay muchos tipos diferentes de agentes (por ejemplo: reactivos, deliberativos, inteligentes, interfaz, colaboración) que a su vez se centran en diferentes entornos de aplicación. Los agentes incluyen contribuciones de varias áreas de la IA, como SCPD y la AI paralela. Es por eso que los agentes heredan: modularidad, velocidad (a través del paralelismo), fiabilidad (a través de la redundancia), facilidad de mantenimiento, reutilización e independencia de la plataforma.

Como software, un agente tiene todas las funciones del sistema de información que puede ser programado, pero lo que le da la calidad de agente es una estructura que identifica cuáles son las frases determinadas para lograr sus objetivos y que forman parte de su composición como entidad. Independientemente de la función realizada por el agente, debe admitir tres fundamentos (Mouratidis y col., 2010): ubiquidad, autonomía y flexibilidad.

3.3.1 Arquitectura de los agentes

En el ámbito de los agentes, la arquitectura se ocupa de los aspectos relacionados con la construcción de sistemas informáticos (que describen la relación entre los módulos de software / hardware) que satisfacen las propiedades de la teoría de los agentes. En los agentes hay una gran variedad de arquitecturas. La primera clasificación de arquitecturas se puede realizar según si todas las capas tienen acceso al entorno, o solo el nivel más bajo tiene acceso. La primera ofrecerá la ventaja de la concurrencia entre las capas a través de un alto conocimiento de la gestión para coordinar las capas, y la segunda reducirá este control a través de una mayor complejidad de la capa que interactúa con el medio ambiente (Agüero 2014).

Estos estilos se extienden de agentes puramente reactivos que operan en una manera simple estímulo-respuesta como los basados en la arquitectura de los apartados y, en el otro extremo, los más "deliberativos" que razonan sus acciones, como la clase de los agentes Creencia-Deseo-Intención (Fasli y col., 2009), (Ribino y col., 2013) cuya prevalencia crece día a día (incluyendo los productos comerciales tales como JACK (Winikoff 2005) de Software Orientado a Agente).

Entre ambos extremos, se pueden encontrar combinaciones híbridas; con este enfoque, el agente se construye por medio de dos o más subsistemas. Uno de ellos es el de deliberación que contiene un modelo simbólico del mundo y el otro es reactivo. Estas arquitecturas combinan módulos reactivos con módulos de deliberación. Los módulos reactivos son responsables de procesar los estímulos que no necesitan deliberación, mientras que los módulos de deliberación determinan qué acciones se deben adoptar para cumplir con los objetivos locales y cooperativos de los agentes (Agüero 2014).

Las siguientes son algunas de las características que en la literatura generalmente se atribuyen a los agentes en mayor o menor medida para resolver problemas específicos (Franklin y col., 1996) (Iglesias 1998) (Julián y col., 2000) (Palma y col., 2008): continuidad temporal, autonomía, sociabilidad, racionalidad, reactividad, proactividad, adaptabilidad, movilidad, veracidad, benevolencia.

3.3.2 Clasificación de los agentes

Las combinaciones de estas características muestran diferentes tipos de agentes. Además, se mencionan otros tipos de agentes que dependen de las características del entorno en el que se encuentran o de las funciones que realizan. En resumen, estos agentes se pueden clasificar como (Agüero 2014): agentes cooperativos o colaborativos, agentes de interfaz, agentes móviles, agentes de información, agentes reactivos, agentes híbridos.

Por lo tanto, desde el punto de vista del usuario, el agente se puede utilizar para realizar las siguientes funciones (López 2005): ejecución de tareas, conocimiento de su entorno, capacidad de comunicación.

3.4 Sistemas distribuidos

Hay diferentes definiciones de sistemas distribuidos en la literatura especializada. Según los autores de Tanenbaum y Steen, estas definiciones no son satisfactorias y no tienen

nada que ver con el resto; lo definen brevemente como: una colección de computadoras independientes que dan al usuario la impresión de un único sistema armonizado (Tanenbaum y col., 2008).

Es un conjunto cooperativo y transparente de componentes de hardware y software remotos que se comunican entre sí a través de mensajería y tienen tres características fundamentales:

- Consta de varias computadoras.
- Hay una relación entre ellos.
- Tienen un estado común.

Las tecnologías de objetos distribuidos y agentes móviles han contribuido significativamente a la transparencia de la migración y, por lo tanto, a la transparencia del rendimiento. El concepto clave es la transparencia, el uso de diferentes procesadores debe ser invisible (transparente) para el usuario.

3.5 Sistemas multiagentes

Los agentes con sus habilidades sociales forman sistemas con más de un agente, es decir, SMA, que son el resultado de la necesidad de desarrollar aplicaciones complejas que tomaría a un agente individual demasiado tiempo para resolver. Los SMA están incluidos directamente en el área de la IAD. Se forman por un conjunto de objetos independientes o agentes que ayudan a resolver un determinado problema común o independiente, que puede conducir a un comportamiento cooperativo o competitivo (Mouratidis y col., 2010) (Manzoor y col., 2014). El objetivo se centra en la interacción entre los agentes que componen el sistema. Por lo tanto, este enfoque se basa en la combinación de capacidades de resolución de diferentes agentes, de modo que la interacción de todos ellos permite la resolución de problemas más complejos, la IAD está entre los diferentes agentes y la interacción entre ellos (Peula 2015).

La construcción de SMA combina tecnologías de diferentes áreas del conocimiento: técnicas de ingeniería de software para estructurar el proceso de desarrollo; técnicas de IA para equipar a los programas con la capacidad de responder a situaciones imprevistas y tomar decisiones, así como la programación paralela y distribuida para hacer frente a las tareas realizadas en diferentes máquinas de acuerdo con diferentes políticas de planificación. Debido a esta combinación de tecnologías, el desarrollo de SMA se complica (Gómez 2003).

La ventaja del aprendizaje multiagente es que el rendimiento del agente o el sistema completamente multiagente mejora gradualmente. El aprendizaje multiagente no es una cuestión de aprendizaje directo, sino que incluye modelos complejos de interacciones sociales. Esto conduce a funciones colectivas complejas (Parde y col., 2015). Muchos de los algoritmos desarrollados en el ML se pueden migrar a una configuración donde hay varios agentes de entrenamiento interdependientes e interactivos. Sin embargo, pueden requerir modificaciones para tener en cuenta otros agentes en el entorno (Yang y col., 2014, Qu y col., 2014).

La forma en que se lleva a cabo el aprendizaje del agente y la salida del proceso de aprendizaje depende de los algoritmos básicos. Estos algoritmos generalmente se refieren a uno de varios métodos, dependiendo de la tarea, la estructura del conocimiento y el resultado deseado. Estos métodos se pueden clasificar de acuerdo con diferentes aspectos. Las perspectivas de clasificación más comunes son: la perspectiva ML y la perspectiva funcional multiagente.

4 Relaciones entre sistemas multiagentes y el aprendizaje automático

Además de los beneficios de la naturaleza distribuida de los SMA, ya que el aumento de la eficiencia es posible a través del paralelismo utilizando múltiples agentes, la implementación de métodos de aprendizaje de amplificación puede aprovechar nuevos beneficios, mediante el intercambio de experiencias, la comunicación, el aprendizaje o la imitación de ellos.

El aprendizaje por refuerzo es un método del ML que se usa principalmente en entornos activos y dinámicos. El agente detecta el entorno y realiza acciones que lo modifican en el proceso iterativo de pruebas y errores. Cada vez que un agente realiza una acción en un entorno, informa sobre su nuevo estado y aprendizaje, que es una medida de la distancia al objetivo.

El objetivo del agente en el proceso del aprendizaje por refuerzo es encontrar la estrategia que lo hará elegir la mejor acción y obtener la recompensa esperada más alta en cualquier estado. Se cree que el agente ha aprendido una estrategia óptima, comúnmente llamada política óptima, cuando es capaz de acumular la mayor recompensa posible por una tarea asignada. Por lo general, para resolver el problema adecuadamente, es suficiente encontrar una política cercana a la óptima (García 2015).

Para tomar decisiones, el agente aprende a acumular experiencia para mejorar su rendimiento. Al igual que al principio, el agente no conoce cómo se comporta el entorno en el que funciona, no tiene otra opción que elegir acciones aleatorias y descubrir los resultados de sus acciones. Después de un cierto tiempo, tendrá la experiencia de las mejores acciones en cada situación. Esto hace que el proceso de aprendizaje sea lento porque la tarea debe repetirse varias veces.

El aprendizaje por refuerzo difiere del aprendizaje supervisado, un tipo de aprendizaje estudiado en la mayoría de los estudios actuales en el campo del ML. El aprendizaje supervisado es el aprendizaje de un conjunto de materiales de capacitación con etiquetas presentadas por un experto observador externo. Cada ejemplo es una descripción de la situación junto con la definición, la etiqueta, la acción correcta que el sistema debe tomar en esta situación, que a

menudo define la categoría a la que se refiere la situación. El objetivo de este tipo de aprendizaje es que el sistema se extrapole o resuma sus respuestas de manera que funcione correctamente en situaciones que no están presentes en el conjunto de entrenamiento. Este es un tipo importante de aprendizaje, pero no es lo suficientemente adecuado para el estudio de la interacción. En tareas interactivas, a menudo no es aconsejable obtener ejemplos de comportamientos deseados que sean correctos y representativos de todas las situaciones en las que el agente debe actuar. En un territorio desconocido donde la capacitación debe ser lo más ventajosa posible, el agente debe ser capaz de aprender de su propia experiencia.

El aprendizaje por refuerzo también es diferente de lo que los investigadores de ML llaman aprendizaje no supervisado, que generalmente se asocia con la búsqueda de una estructura oculta en colecciones de datos no marcados. Los términos, aprendizaje supervisado y aprendizaje no supervisado parecen clasificar exhaustivamente los paradigmas, pero no lo hacen. Si bien se cree que el aprendizaje por refuerzo es un tipo de aprendizaje no supervisado, ya que no se basa en ejemplos de comportamiento correcto, el aprendizaje por refuerzo intenta maximizar la señal de recompensa en lugar de buscar una estructura oculta. El descubrimiento de la estructura de la experiencia del agente puede ser útil para mejorar el aprendizaje, pero en sí mismo no afecta el problema de aprender mediante el refuerzo, con el fin de maximizar la señal de recompensa. Por lo tanto, el aprendizaje por refuerzo se considera el tercer paradigma del ML, junto con el aprendizaje supervisado y el aprendizaje no supervisado y tal vez otros paradigmas.

Uno de los desafíos que surge en el aprendizaje por refuerzo, pero no en otros tipos de aprendizaje, es el intercambio entre la exploración y la explotación. Para obtener una excelente recompensa, un agente de aprendizaje por refuerzo debe preferir las acciones que ha intentado en el pasado que se han reconocido como efectivas para obtener una recompensa. Sin embargo, con la finalidad de detectar tales acciones, debe seleccionar las acciones que no ha elegido anteriormente. El agente debe emplear lo que ya ha experimentado para obtener recompensas, pero también debe investigar para mejorar la gama de acciones en el futuro (Sutton y col., 2017).

4.1 Aprendizaje mediante el refuerzo en sistemas multiagentes

El aumento de la eficiencia se puede lograr mediante el paralelismo cuando los agentes utilizan estructuras descentralizadas de la tarea. La división adecuada de la tarea en subtareas y su asignación a varios agentes paralelos aumentará la eficiencia general del sistema. Compartir experiencias puede ayudar a los agentes con tareas similares a aprender más rápido y mejor, por ejemplo:

El aumento de la eficiencia se puede lograr a través del paralelismo cuando los agentes utilizan estructura descentralizada de la tarea, la separación adecuada de las tareas en subtareas. Las tareas y subtareas son asignadas a diferentes agentes en paralelo para mejorar la eficiencia global del sistema. El intercambio de experiencia podría ayudar a los agentes con objetivos similares a aprender más rápido y mejor, por ejemplo:

- Los agentes pueden compartir información a través de una comunicación directa para que el agente pueda aprender de su propia experiencia y además de otros agentes.
- Los agentes más calificados en sus tareas pueden enseñar cómo resolver sus tareas.
- Un agente que todavía está aprendiendo puede observar e imitar a los agentes más calificados, basándose en la experiencia de otros agentes para acelerar el proceso de aprendizaje.

Además, cuando uno o más agentes fallan en su tarea en un SMA, los demás agentes pueden encargarse de sus tareas, lo que hace que el SMA sean más confiable. La mayoría de las plataformas de agentes permiten agregar fácilmente nuevos agentes al sistema.

5 Conclusiones

En general, el ML se dedica a la construcción de programas que, utilizando la experiencia, son capaces de mejorar automáticamente su rendimiento. La combinación de dos áreas de cálculo en desarrollo continuo, como la IA y la IAD, permitió la implementación de agentes móviles. Los agentes generalmente se pueden considerar como una extensión del paradigma de un objeto distribuido, donde dos rasgos distintivos principales son la autonomía y la sociabilidad. En general, los agentes de diferentes tipos participan en SMA. Algunos son más simples de carácter reactivo que generalmente ofrecen servicios a otros, más complejos, híbridos o deliberativos. Esto depende de los objetivos del agente y de la necesidad de razonar sobre su contexto, que el diseñador decide aplicar una u otra arquitectura.

Referencias

Agis RA, Sebastian G, García AJ, 2017, Conocimiento Compartido y Razonamiento Argumentativo Colaborativo para Entornos de Múltiples Agentes en Ambientes Distribuidos. Memoria del evento XIX Workshop de Investigadores en Ciencias de la Computación, Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca.

Agüero MJ, 2014, Diseño de organizaciones virtuales ubícuas utilizando desarrollo dirigido por modelos. Tesis Doctorado, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia.

Amaolo M, Dolz D, Grosso G, Kogan P, 2017, Agentes In-

teligentes. Modelos Formales y Aplicaciones para la Educación. Memoria del evento XIX Workshop de Investigadores en Ciencias de la Computación, Departamento de Teoría de la Computación - Facultad de Informática, Universidad Nacional del Comanue, Buenos Aires.
Amigone F, Rodríguez J, Parra G, 2017, Hacia la Definición de un Agente Generador de Conocimiento de Valor Social para Poblaciones en Riesgo. Memoria del evento XIX Workshop de Investigadores en Ciencias de la Computación, Departamento de Teoría de la Computación - Facultad de Informática, Universidad Nacional del Comanie, Buenos Aires.
Anglada M, Alcalá J, Llanes L, Mateo A, Salán M, 2002, Fractura de Materiales. Barcelona: Edicions UPC.
Balaji PG, y Srinivasan D, 2010, An Introduction to Multi-Agent Systems. Springer-Verlag Berlin Heidelberg.
Brink H, Richards JW, Fetherolf M, 2017, Real-World Machine Learning. Shelter Island: Manning Publications Co.
Castillo GN, 2015, Técnicas de Machine Learning para el Post-Proceso de la predicción de la Irradiancia. Tesis de maestría, Departamento de Estadística e Investigación Operativa, Universidad de Granada, 2015.
Chapelle O, Schölkopf B, Zien A, 2009, Semi-Supervised Learning. IEEE Transactions on Neural Networks 20, nº 3 (Marzo 2009): 542.
Fasli M, Botond V, 2009, BDI Agents: Flexibility, Personalization, and Adaptation for Web-Based Support Systems. Intelligent Agents in the Evolution of Web and Applications.
Franklin S, y Graesser A, 1996, Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. Springer-Verlag.
García RO, 2015, Aprendizaje por refuerzo mediante transferencia de conocimiento cualitativo. Tesis Doctoral, Instituto Nacional de Astrofísica, Óptica y Eléctrica, Tonantzintla, Puebla.
Godoy VÁ, 2017, Técnicas de aprendizaje de máquina utilizadas para la minería de texto. Bibliotecológica 31, nº 71 (enero/abril 2017): 103-126.
Gómez SJ, 2003, Metodologías para el desarrollo de sistemas multi-agente. Revista Iberoamericana de Inteligencia Artificial, nº 18.
Herrero CC, 2015, Modelización de la concurrencia en sistemas distribuidos multiagente: Autómatas Cooperativos. Tesis Doctoral.
Iglesias FC, 1998, Definición de una metodología para el desarrollo de Sistemas Multiagente. Tesis Doctoral, Madrid.
Julián V, y Botti V, 2000, Agentes Inteligentes: el siguiente paso en la Inteligencia Artificial. ATI.
Khaled MK, Abdelaziz M, Nazmy TT, Salem AB, 2015, Machine Learning Algorithms for Multi-Agent Systems. ResearchGate.
López TB, 2005, Agentes Inteligentes. Instituto Tecnológico Nuevo Laredo.
Manzoor U, y Zafar B, 2014, Multi-Agent Modeling Toolkit – MAMT. Simulation Modelling Practice and Theory 49.
Mateus SSP, 2015, Modelo de un Entorno Virtual Inteligente basado en la percepción y el razonamiento de sus elementos con un personaje para la generación de realismo. Tesis Doctoral, Departamento de Ciencias de la Computación y la Decisión, Universidad Nacional de Colombia, Medellín.
Mondeleón A, Vegas E, y Reverter F, 2017, Big Data. Hacia la cuarta revolución industrial. Barcelona: Ediciones de la Universidad de Barcelona.
Mouratidis H, Kolp M, Giorgini P, y Faulkner S, 2010, An Architectural Description Language for Secure Multi-Agent Systems. Web Intelligence and Agent Systems.
Parde N, y Nielsen R, 2015, Design Challenges and Recommendations for Multi-Agent Learning Systems Featuring Teachable Agents. In Proceedings of the 2nd Annual GIFT Users Symposium. Pennsylvania. 12-13.
Peula PJM, 2015, Sistema de coordinación multiagente basado en comportamientos aprendidos. Tesis Doctoral, Málaga.
Qu S, Jian R, Chu T, Wang J, Tan T, 2014, Comuptional reasoning and learning for smart manufacturing under realistic conditions. 2014 International Conference on Behavioral, Economic, and Socio-Cultural Computing (BESC2014). Shanghai: IEEE, 2014. 1-8.
Ribino P, Cossentino M, Lodato C, Lopes S, Sabatucci L, Seidita V, 2013, Ontology and Goal Model in Designing BDI Multi-Agent Systems. Journal WOA@ AI* IA 1099 (2013).
Sampedro, Javier De Seco, 2012, Estudio y aplicación de técnicas de aprendizaje automático orientadas al ámbito médico: estimación y explicación de predicciones individuales. Trabajo de Fin de Máster presentado para la obtención del título Máster en Ingeniería Informática y de Telecomunicaciones, Ingeniería Informática, Madrid: Universidad Autónoma de Madrid.
Sutton RS, Barto AG, 2017, Reinforcement Learning: An Introduction. London, England.
Tanenbaum AS, Van SM, 2008, Sistemas Distribuidos principios y Paradigmas. 2. México: Pearson Educación de México, S.A. de C.V.
Torunoğlu D, Çakırman E, Ganiz MC, Akyokuş S, Gürbüz MZ, 2011, Analysis of Preprocessing Methods on Classification of Turkish Texts. IEEE, Julio.
Winikoff M, 2005, Jack™ Intelligent Agents: An Industrial Strength Platform. Springer 15.
Yang Z, y Shi X, 2014, An agent-based immune evolutionary learning algorithm and its application. Proceeding of the 11th World Congress on Intelligent Control and Automation. Shenyang: IEEE, 2014. 5008-5013.

Recibido: *10 de septiembte de 2019*

Aceptado: *25 de noviembre de 2019*

González Pérez, Yuleisy: *MSc en Informática Aplicada (UCI, Cuba). Aspirante PhD Computer Science and Computer Facility (LETI, Rusia).*

Kholod, Ivan Ivanovich: *PhD en Ciencias Técnicas. Decano en funciones de la Facultad de Tecnologías Computacionales e Informática (LETI, Rusia). Correo electrónico: iiholod@mail.ru*