Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 11, No. 5, Mes Mayo, 2018
ISSN: 2306-2495 | RNPS: 2343
Pág. 11-16
http://publicaciones.uci.cu

Tipo de artículo: Artículo original
Temática: Soluciones Informáticas
Recibido: 10/03/18 | Aceptado: 04/5/18 | Publicado: 25/05/18

Diseño paralelo de algoritmo de aprendizaje por refuerzo para entornos big data

Parallel design of reinforcement learning algorithm for big data environment

Jacinto Rivero Hernández[1], Jimmy Linares Lagarto[1], Antonio Lamazares Fernández[1], Lester Guerra Denis[1], Humberto Díaz Pando[1]

[1]Universidad Tecnológica de la Habana (CUJAE), lguerra@ceis.cujae.edu.cu

[*] Autor para correspondencia: lguerra@ceis.cujae.edu.cu

Resumen

El aprendizaje por refuerzo es una forma de aprendizaje basado en la prueba y error. Este tipo de aprendizaje se aplica a problemas complejos que requieren en la actualidad procesar grandes volúmenes de datos. Algunas de estos problemas sonadministración de recursos, problemas de planificación, control de tráfico, robótica, detección de intrusos, control de sistemas energéticos, detección de fraude en banca online, etc.Los algoritmos de aprendizaje por refuerzo tradicionales no están preparados para tratar conentornos big data, desde el enfoque de un único agente. Aprovechando las capacidades de computo de los sistemas multicomputadorse diseñó una variante del algoritmo Q-Learningenfocada a sistemas multiagentes. Luego con ambas variantes se realizó un experimento para validar que la solución acelera el proceso de aprendizaje basado en la cantidad de iteraciones que tardan las soluciones en completar un episodio. Finalmente se valida la solución a partir de la aplicación de una prueba estadística con los resultados obtenidos.

Palabras clave: aprendizaje por refuerzo; big data; sistemas multiagentes.

Abstract

Reinforcement learning is a kind of learning based on trial and error. This kind of learning is applied to complex problems that requires to process big data nowadays. Some of these problemsare resources management, scheduling problems, traffic control, robotics, intrusion detection systems, energy systems, fraud in online banking, etc. Traditional reinforcement learning algorithms are not prepared to interact with big data environments, from an only

agent approach. We designed a Q-Learning variant for multi-agent system that exploits the computation capabilities of multi-computer systems. After that, we carry out an experiment with both variantsto validatethat the solution improves the learning process speed, based on the iteration quantity that delaysto finish an episode. Finally, the solution is validated applying a statistic test with the obtained results.

Keywords: *big data; reinforcement learning; multi-agent system.*

Introducción

El aprendizaje por refuerzo es un área del aprendizaje automático, enfocada en como un agente debe actuar en su entorno para maximizar una recompensa acumulada(1). Visto desde otro punto de vista es una forma de aprendizaje basado en la prueba y error. Este tipo de aprendizaje generalmente se utiliza para dar solución a problemas donde la generación de un modelo a priori es muy costosa. Algunas áreas de aplicación de este tipo de aprendizaje son administración de recursos, problemas de planificación, control de tráfico, robótica, detección de intrusos, control de sistemas energéticos, detección de fraude en banca *online*, etc. Muchos de estos problemas en la actualidad requieren del procesamiento de grandes volúmenes de información.

El desarrollo de las tecnologías de la información y las telecomunicaciones presente en la actualidad ha conllevado a la acumulación de un conjunto de datos con tres características principales: gran volumen, alta velocidad de generación y variedad estructural. A los conjuntos de datos con estas características se les conoce por el término *big data*(2).

Las formas tradicionales de almacenar, transferir, procesar y gestionar son incapaces, al menos en un tiempo razonable, de tratar con estos datos, debido al límite en la capacidad de procesamiento actual. Haciendo énfasis en el procesamiento de los datos la mayor parte de las soluciones a este problema son a través del empleo de las capacidades de cómputo de los sistemas distribuidos.

Los algoritmos de aprendizaje por refuerzo tradicionales no están preparados para tratar con entornos *big data*(1)(3). Esto se debe a que se ejecutan de manera secuencial desaprovechando los recursos de cómputo de las arquitecturas paralelas actuales.Por tanto, el objetivo de la investigación es diseñar una variante de aprendizaje por refuerzo para sistemas multiagentes, que mejore el rendimiento de los algoritmos tradicionales con respecto a la convergencia hacia la política óptima.

Materiales y métodos o Metodología computacional

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 11, No. 5, Mes Mayo, 2018
ISSN: 2306-2495 | RNPS: 2343
Pág. 11-16
http://publicaciones.uci.cu

El aprendizaje por refuerzo se basa en la interacción entre un agente y un entorno. El entorno envía un estado (S_t) al agente, el cual basado en su experiencia realiza una acción (A_t) en respuesta. Esta acción conlleva al entorno a un nuevo estado (S_{t+1}) y asociada a este genera una recompensa (R_{t+1}), estos datos son enviados de nuevo al agente y repite el proceso. De esta forma el objetivo del agente es maximizar la noción de recompensa que recibe, de esto se encarga el algoritmo de aprendizaje(1).

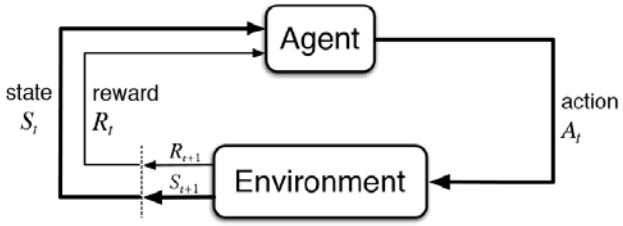

Fig 1: Interacción entre agente y entorno.

Una forma de clasificación de los algoritmos de aprendizaje por refuerzo se basa en su relación con un modelo del entorno. Los algoritmos basados en modelo tienen conocimiento de la dinámica del entorno, es decir la probabilidad de transición entre los estados del entorno. Sin embargo, este tipo de algoritmos no son prácticos para espacios de estados grandes, por tanto, no son de interés para esta investigación. Por otra parte, existe un grupo de algoritmos conocidos como libres de modelo ya que se basan solamente en la prueba y el error. Esto los hace especialmente útiles en problemas donde la generación de un modelo del entorno tiene un costo computacional excesivo, como un entorno *big data*.

Al generalizar un espacio de estados y acciones continuo estos se convierten en grandes conjuntos discretos.Esto plantea un reto para los algoritmos tradicionales por tanto surge la necesidad de emplear un algoritmo de aprendizaje por refuerzo para sistemasmultiagentes que permita un mayor aprovechamiento del conocimiento, el cual se logra con un conjunto de agentes aprendiendo en paralelo y socializando su conocimiento.

Esta investigación se enfoca en el estudio de Q-Learning(Fig 2)un algoritmo de aprendizaje por refuerzo libre de modelo basado en la bien conocida ecuación de Bellman. En él se asigna un valor llamado Q, también conocido como Q-valor, a cada par estado-acción. Este valor indica que tan buena es una acción en un estado dado con respecto al estado final. La meta de Q-Learning es maximizar los Q-valores obtenidos a largo plazo y de esta forma converger a la política óptima.

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 11, No. 5, Mes Mayo, 2018
ISSN: 2306-2495 | RNPS: 2343
Pág. 11-16
http://publicaciones.uci.cu

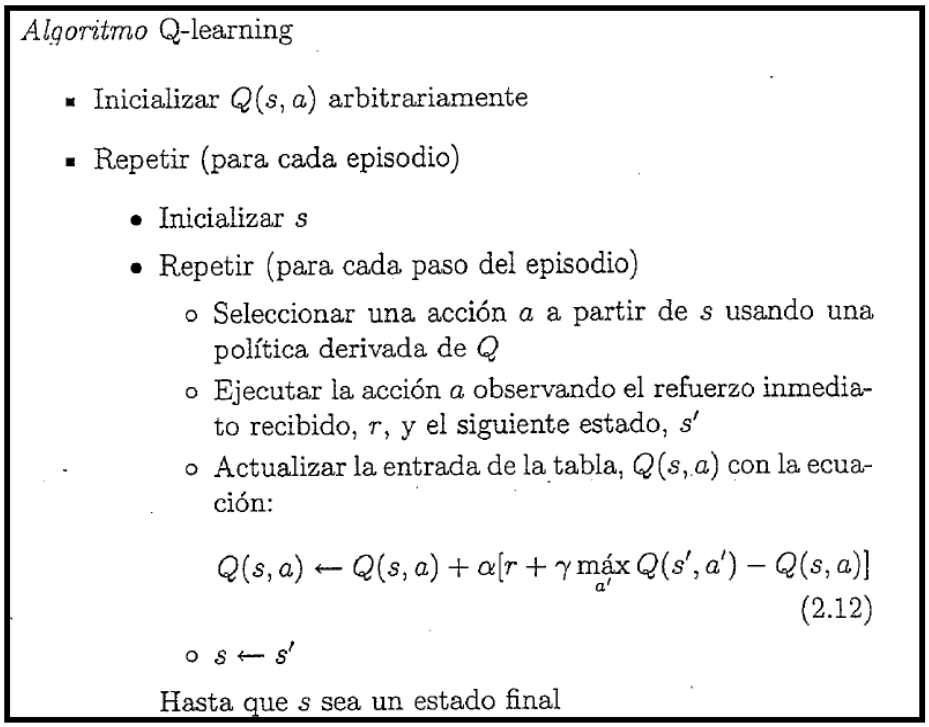

Fig 2: Pseudocódigo de Q-Learning.

Apache Spark es un *framework* de computación distribuida diseñado para ser rápido y de propósito general (4). Además, posee mecanismos de tolerancia a fallos, manejo de grandes volúmenes de datos y gran comunidad de desarrolladores lo que lo convierte en una buena herramienta para implementar la solución. Este *framework* tiene APIs para varios lenguajes de programación como R, Python, Scala y Java. Se seleccionó Scala como lenguaje de programación ya que el *framework* está desarrollado en este.

Resultados y discusión

La solución propuesta es el diseño de una variante de Q-Learning para sistemas multiagentes.Para validar la solución se realizó un experimento donde un agente tiene que encontrar un estado meta fijo en un espacio de estados continúo generalizado. El espacio de estados se modeló como una matriz cuadrada de orden 4. El espacio de acciones está formado por los posibles movimientos del agente (arriba, abajo, izquierda, derecha). El objetivo del experimento es comparar la cantidad de iteraciones que tarda un agente en completar un episodio (encontrar la meta) comparado con un sistema multiagente. La cantidad de iteraciones es una medida directa de la eficiencia en el proceso de aprendizaje.

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 11, No. 5, Mes Mayo, 2018
ISSN: 2306-2495 | RNPS: 2343
Pág. 11-16
http://publicaciones.uci.cu

Para llegar a conclusiones se realizó una prueba de hipótesis donde:

H_0: Cantidad de iteraciones de un agente = Cantidad de iteraciones del sistema multiagente.

H_1: Cantidad de iteraciones de un agente > Cantidad de iteraciones del sistema multiagente.

Para probar las hipótesis se utilizó la prueba estadística no paramétrica de Mann-Whitney(5). Se seleccionó esta porque no se cuenta con evidencias para afirmar que los datos sigan una distribución normal. El valor de máximo nivel de riesgo aceptable para rechazar una hipótesis nula verdadera(α) utilizado fue 0,05.

Para llevar a cabo el siguiente experimento se utilizó una unidad experimental con las siguientes características:

• Procesador Intel® Core™ i3-6100U @ 2,3 GHz con 4 hilos de ejecución.

• 4 GB de Memoria RAM.

Los factores controlables identificados en el experimento son el índice de aprendizaje (α), el factor de descuento (γ) y la política de selección de estados. Los factores no controlables identificados son los procesos ejecutados por el sistema operativo. El tratamiento dado a los factores controlables se muestra en la tabla 1.

Tabla 1: Tratamiento de los factores controlables.

Factor	Tratamiento
índice de aprendizaje (α)	0,1
factor de descuento (γ)	0,95
política de selección de estados	ε-greedy

Luego de obtener ambas versiones del algoritmo se pasó a analizar el comportamiento de las métricas de rendimiento. La prueba que se detalla a continuación, es resultado de 10 ejecuciones del algoritmo en cada una de las versiones que se comparan.

Tabla 2: Resumen de los resultados del experimento.

	Cantidad de iteraciones de un agente	**Cantidad de iteraciones del sistema multiagente**
Mínimo	55582	5337
Máximo	628369	49024
Media	192868,7	25989,6

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 11, No. 5, Mes Mayo, 2018
ISSN: 2306-2495 | RNPS: 2343
Pág. 11-16
http://publicaciones.uci.cu

Con estos resultados se pasó a comprobar si la versión paralela del mismo realiza menos iteraciones que la secuencial. Para ello se aplicó la prueba estadística descrita anteriormente utilizando el lenguaje R.

```
        Wilcoxon rank sum test

data:  secuencial and paralelo
W = 100, p-value = 5.413e-06
alternative hypothesis: true location shift is greater than 0
```

Fig 2: Resultados de prueba.

Como se muestra en la Fig 2, el valor $p=0,5 * 10^{-5}$ que es menor que 0,05 permite concluir que la versión paralela del algoritmo disminuye la cantidad de iteraciones y por tanto aprende más rápido (5).

Conclusiones

Los experimentos realizados y los resultados obtenidos con las pruebas de hipótesis no paramétricas de Mann-Whitney arrojan que la solución propuesta disminuye la cantidad de iteraciones en relación a la variante tradicional. Este resultado indica que el sistema multiagente converge más rápido a la política óptima.

Referencias

1. **Russell, Stuart y Norvig, Peter.***Artificial Intelligence, A Modern Approach.* New Jersey : Prentice Hall, 2010.

2. **Davis, Kord y Patterson, Doug.***Ethics of Big Data.* s.l. : O'Reilly Media, Inc., 2012.

3. **Google DeepMind.***Massively Parallel Methods for Deep Reinforcement Learning.* London : s.n., 2015.

4. **Karau, Holden, y otros.***Learning Spark.* s.l. : O'Reilly Media, Inc., 2015.

5. **Hernández, Victor, Ramos, Eduardo y Yanéz, Ildefonso.***Probabilidad y sus aplicaciones en Ingeniería Informática*. 2007.