Aprendizaje por Refuerzo aplicado a la resolución de problemas no triviales

Marcelo Errecalde, Alfredo Muchut, Guillermo Aguirre y Cecilia Montoya
{merreca, amuchut, gaguirre, cmontoya}@unsl.edu.ar
P338403 - Universidad Nacional de San Luis. Argentina

1. Introducción

El Aprendizaje por Refuerzo (en inglés Reinforcement Learning y de ahora en más AR) ataca el problema de aprender a controlar agentes autónomos, mediante interacciones por prueba y error con un ambiente dinámico, el cual le provee señales de refuerzo por cada acción que realiza.
Si los objetivos del agente están definidos por la señal de refuerzo inmediata, la tarea del agente se reduce a aprender una estrategia de control (o política) que permita maximizar la recompensa acumulada a lo largo del tiempo (ver [14] para una formalización de esta tarea)
Si bien en sus orígenes el AR sirvió como una herramienta teórica limitada a problemas con pequeños espacios de estados, en la actualidad sus aplicaciones han alcanzado áreas de considerable complejidad tales como robótica, manufacturación industrial, problemas de búsqueda combinatorial, etc.
La aplicación del AR a problemas del mundo real, trajo aparejado la necesidad de adaptar las técnicas existentes en el área para manejar características complejas propias de este tipo de ambientes (ambientes estocásticos no estacionarios con grandes espacios de estados y/o acciones).
En esta presentación, describimos el trabajo realizado por nuestro grupo de investigación en la aplicación del AR a problemas no triviales del mundo real. Para ello, describimos en las secciones 2 a 4, los 3 factores principales que a nuestro criterio deben ser tenidos en cuenta al trabajar con AR en este tipo de ambientes: balance entre exploración y explotación, aceleración del proceso de aprendizaje y generalización. La sección 5 por su parte, describe cuales son los avances y resultados que hemos logrado en relación a estos 3 ítems, y una breve descripción del plan de trabajo futuro

2. Balance entre exploración y explotación

Una de las principales características del AR está dada por el hecho de delegar en el agente que aprende la responsabilidad de determinar la estrategia para explorar el ambiente.
A diferencia del aprendizaje supervisado, en AR es el agente quien controla los ejemplos de entrenamiento mediante la secuencia de acciones que elige. Esto implica que el agente debe balancear la *exploración* de nuevos estados y acciones para obtener nueva información que le permita evitar óptimos locales, y la *explotación* de estados y acciones ya aprendidos y con un alto reward que le garantice un reward acumulado aceptable.
Dado que es imposible explorar y explotar en forma simultánea con una única selección de acción, esta situación es a menudo denominada como el "conflicto" entre exploración y explotación.
Existen varias propuestas para lograr el balance entre exploración y explotación (ver [19] para un survey) pudiéndose mencionar entre las más conocidas a la estrategia $\in$-greedy [6], valores iniciales optimistas [6], métodos de selección de acción basados en la distribución de Boltzmann [6, 10, 14, 19], el método de estimación de intervalo [7], el bono de exploración usado en Dyna [15, 16] y los mapas de competencia [18].
La necesidad de mantener un mínimo de exploración es un factor fundamental en problemas reales, si se toma en cuenta las características dinámicas de este tipo de ambientes.

3. Aceleración del aprendizaje

Uno de los principales problemas del AR es que el agente requiere un gran número de episodios de entrenamiento para aprender una función de valor aceptable.
Existen actualmente dos enfoques principales para la aceleración del proceso de aprendizaje: 1) permitir la incorporación de información provista por un observador externo [8, 9, 12] y 2) integrar learning con planning [8, 11, 12, 15, 16].
El primero consiste en posibilitar que un experto u observador externo pueda incorporar "consejos" que le sirvan al agente para aprender ciertos aspectos complejos del ambiente en forma más eficiente.
En el segundo caso, el agente aprende un modelo del ambiente en forma simultánea al aprendizaje de la política, lo que permite hacer un uso más intensivo de una cantidad limitada de experiencia.

4. Generalización

El AR se basa en la estimación de funciones de valor óptimas definidas sobre el conjunto de estados o el conjunto de acciones. Cuando el espacio de estados o acciones es pequeño, estas funciones son usualmente representadas explícitamente en forma tabular.

Si bien este tipo de representación trabaja relativamente bien en dominios pequeños del estilo de los laberintos artificiales, se torna inadecuado para la mayoría de los dominios complejos del mundo real, donde los espacios de estados o acciones suelen ser excesivamente grandes e incluso continuos. En estos casos, la representación tabular explícita de las funciones de valor no sólo implicará requerimientos de memoria inaceptables, sino además un uso ineficiente de la experiencia adquirida durante el aprendizaje. Cuando los espacios de estados o acciones incluyen variables continuas o sensaciones complejas tales como imágenes visuales, la mayoría de los estados encontrados nunca habrán sido experimentados antes. La única manera para aprender algo sobre estas tareas es generalizar desde estados previamente experimentados a aquellos que nunca han sido vistos antes. Esto se puede lograr a través de las técnicas de generalización clásicas en aprendizaje automático las cuales permiten un almacenamiento compacto de la información aprendida y la transferencia de conocimiento entre estados y acciones "similares".

En este sentido, la idea consiste en aproximar las funciones involucradas en las arquitecturas y algoritmos de AR usando cualquiera de la amplia variedad de técnicas de aproximación de funciones para aprendizaje supervisado que soportan ejemplos de entrenamiento con ruido, como por ejemplo métodos de backpropagation [8, 17], basados en memoria local [1], árboles de decisión [2], etc.

5. Estado de avance. Resultados obtenidos y trabajo futuro

Los trabajos iniciales del grupo, se concentraron en la aceleración del proceso de aprendizaje mediante la integración de planning y learning y las técnicas para balancear exploración y explotación [4].

Este trabajo, mostró la importancia que tiene la elección de una adecuada técnica de exploración y de que manera el aprendizaje simultáneo de un modelo del ambiente puede acelerar el proceso de aprendizaje. Si bien en este caso, el problema a resolver se restringió a un problema de los laberintos, sirvió para visualizar de que manera la incorporación en el ambiente de múltiples estados absorventes que constituyen óptimos locales, invalidan desde un punto de vista práctico a técnicas de exploración que garantizan en teoría la convergencia a una política óptima para el agente.

Actualmente este trabajo está siendo extendido a ambientes no estacionarios, con distintos tipos de modificaciones dinámicas (Blocking mazes [14] y Shortcut mazes[14]) a los fines de analizar las falencias de las distintas políticas de exploración para controlar este tipo de cambios ambientales. En este sentido, además de comprobar la ineficacia de las técnicas que privilegian la exploración en las etapas tempranas del aprendizaje, hemos comprobado que algunas soluciones a este problema descansan en la asistencia directa de un observador externo como en [12] y que sus propuestas originales en su forma pura se tornan inválidas para detectar ciertos cambios dinámicos del ambiente.

Dado que es nuestro interés preservar la máxima autonomía posible del agente, nuestra línea de investigación se ha orientado a analizar de que manera el agente puede detectar en forma automática el grado y tipo de variabilidad del ambiente, de manera tal de adaptar (aprender) los factores que afectan su política de exploración. En este sentido estamos trabajando sobre propuestas de soluciones a este problema obteniéndose resultados preliminares que sustentan la factibilidad de nuestro enfoque.

En lo referido a aplicaciones de AR a problemas reales, se han implementado agentes de AR que negocian en forma automática con otros agentes, en un escenario de negociación basado en ofertas y contraofertas similar al planteado en [20, 21]. A diferencia de estos trabajos (basados en aprendizaje bayesiano) el agente vendedor/comprador de AR no contó con conocimiento previo sobre la política de ofertas del otro agente, sino que debió aprender por negociaciones repetidas con el mismo agente. Los resultados obtenidos mostraron que los agentes de AR obtuvieron una mayor utilidad que agentes con políticas fijas, aprendiendo una política de ofertas óptima en relación al precio de reservación del otro agente que negocia (incluso con variaciones dinámicas de dicho precio de reservación). Este trabajo se realizó en el contexto de las ideas presentadas en [3, 5] para la integración de aprendizaje y sistemas multiagentes. Si bien la implementación actual del sistema fue realizada en C++, estamos trabajando en su migración a Java utilizando un soporte para la definición de agentes y su comunicación (JATLite).

En lo referido al aspecto de generalización hemos dejado el tratamiento de este aspecto como último

punto de investigación. Esta decisión está motivada en el convencimiento de que las representaciones tabulares de AR integradas con planning aplicadas a ambientes no estacionarios es aún un problema abierto y que la inclusión de técnicas de generalización sólo obscurecería más esta problemática.
No obstante esto, hemos comenzado con el estudio e implementación de redes neuronales de backpropagation para la aproximación de las funciones de valor. Dado que este trabajo está en sus etapas iniciales hemos tratado de subsanar el problema de espacios de estados grandes, utilizando representaciones de estados y acciones que pudieran ser almacenadas en forma tabular, al costo de restringir el problema en muchos casos.
A los fines de dar una solución transitoria a este punto, hemos implementado una versión paralela de Q-Learning utilizando las librerías de Parallel Virtual Machine (PVM) sobre una red de workstations Sun con dos o más procesos "esclavos" trabajando sobre diferentes piezas de la tabla Q. Si bien la política fue aprendida en forma correcta, esta implementación se realizó para estudiar la factibilidad del enfoque, no habiéndose realizado aún estudios comparativos con la versión secuencial de Q-Learning. En este sentido, estamos actualmente portando el sistema a una máquina paralela Parsytec Power PC de 32 procesadores para analizar su comportamiento en este tipo de arquitectura.

6. Referencias

[1] J.Boyan y A. Moore. "Generalization in reinforcement learning: Safely approximating the value function". In Tesauro, Toretzky y Leen (Eds), Advances in Neural Information Proccesing Systems: Proc. of the 1994 Conference, pags. 369 - 376. Cambridge, MA. The MIT Press, 1995.
[2] D.Chapman y L.P. Kaelbling. "Input generalization in delayed reinforcement learning: An algorithm and performance comparisons". In Proceedings of the Twelfth International Conference on Artificial Intelligence, pags. 726 - 731. Morgan Kaufmann, San Mateo, CA, 1991.
[3] M.Errecalde y G. Aguirre. "Extendiendo interfases de usuario: agentes, aprendizaje y distribución". Trabajos del WICC'99. Workshop de aspectos teóricos de la Inteligencia Artificial, 1999.
[4] M.Errecalde, M. Crespo y C. Montoya. "Aprendizaje por Refuerzo: Un estudio comparativo de de sus principales métodos". Proc. del II Encuentro Nacional de Computación (ENC'99). Sociedad Mexicana de Ciencia de la Computación. México, 1999.
[5] M.Errecalde, y G. Aguirre. "Una propuesta para integrar agentes de interfase, aprendizaje y sistemas multiagentes". Proc. del II Encuentro Nacional de Computación (ENC'99). Sociedad Mexicana de Ciencia de la Computación. México, 1999.
[6] L. P.Kaelbling, M. Littman y A. Moore. "Reinforcement Learning: A Survey". Journal of Artificial Intelligence Research 4 (1996) - 237-285 - Mayo 1996.
[7] L.P.Kaelbling. Learning in Embedded Systems. MIT Press. Cambridge, MA, 1993.
[8] L. - J.Lin. "Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching". Machine Learning - Volumen 8 - Número 3/4 - Mayo 1992.
[9] R.Maclin y J. W. Shavlik. "Creating Advice-Taking Reinforcement Learners". Machine Learning - Volumen 22 - Págs. 251 - 282. 1996.
[10] T.Mitchell. "Machine Learning". Capítulo 13. (Versión preliminar).
[11] A.Moore y C. Atkeson. "Prioritized Sweeping: Reinforcement Learning with Less Data and Less Time ". Machine Learning - Volumen 13 - Número 1 - Octubre 1993.
[12] J.Peng y R. J. Williams. "Efficient learning and planning within the Dyna framework". Adaptative Behavior, 1(4), Págs. 437-454, 1993.
[13] S.Russell y P. Norvig. "Artificial Intelligence. A modern Approach".Prentice –Hall –1995.
[14] R.Sutton y A. Barto. " Reinforcement Learning: an introduction". The MIT Press, 1998.
[15] R.Sutton. "Dyna. an Integrated Architecture for Learning, Planning, and Reacting" Working Notes of the AAAI Spring Symposium, pp.151-155, 1991.
[16] R.Sutton. "Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming". – Proceedings of the Seventh Int. Conf. On Machine Learning, pp. 216-224, Morgan Kaufmann, 1990.
[17] G.Tesauro. "Practical issues in temporal difference learning". Machine Learning - Volumen 8 - Número 3/4 - Pags. 257 - 277. Mayo 1992.
[18] S.Thrun y K. Moller. "Active exploration in dynamic environments". Advances in Neural Information Proccesing Systems, 4, pags. 531 - 538. San Mateo, CA, Morgan Kaufmann, 1992.
[19] S.Thrun. "The role of Exploracion in Learning Control". Handbook in Intelligent Control: Neural, Fuzzy, and Adaptative Approaches, White, D. A., & Sofge, D. A. (Eds.).
[20] D.Zeng y K. Sycara. "Bayesian learning in negotiation". International Journal of Human-Computer Studies, 48, 1998.
[21] D.Zeng y K. Sycara. "Benefits of learning in negotiation". In Proceedings of AAAI-97, 1997.