

PROYECTO DE CURSO INTELIGENCIA ARTIFICIAL 2020-1

Gemma Bel-Enguix
Helena Gómez-Adorno
Alejandro Pimentel

Antecedentes

Se propone una variante del Naming Game, juego clásico de inteligencia artificial que fue por primera vez imaginado por Wittgenstein (1953). Desde los años noventa del siglo XX se han llevado a cabo algunas implementaciones del modelo: Naming Games (Steels and McIntyre, 1999), Guessing Games (De Beule et al., 2006), Description Games (van Trijp, 2008).

En 2006, Baronchelli et al. publicaron una versión del Naming Game que puede describirse como sigue: existe una comunidad de agentes que tiene por objetivo ponerse de acuerdo sobre cómo nombrar un objeto (convergencia). Para ello en cada unidad de tiempo un agente interactúa con otro intercambiando su conocimiento sobre ese objeto. Las interacciones finalizan cuando todos los agentes tienen en su memoria una sola palabra, y la misma, para nombrar ese objeto. Este artículo estudia algunas propiedades como la transición del orden al desorden, la distribución de la memoria de los agentes y las dinámicas de optimización.

En este proyecto se propone un diseño basado en el artículo de Baronchelli, con algunas especificaciones que se van a explicar a continuación.

Descripción del problema

Existe una sociedad con un conjunto N de agentes y un conjunto C de comunidades, de manera que el número de agentes por comunidad es $N \geq 2$.

En este mundo, existen uno o más objetos O . Todos los agentes de la sociedad tienen que ponerse de acuerdo para dar un nombre diferente a cada uno de esos objetos. El juego termina cuando todos los agentes tienen una sola palabra y la misma para nombrar a cada uno de esos objetos.

El sistema es más eficiente cuanto menos tiempo toma la convergencia. Este tiempo (T_{conv}) se mide en interacciones peer-to-peer entre pares de agentes.

Comunidades

Los agentes se agrupan en comunidades. El programa debe permitir la definición libre del número de agentes y comunidades de acuerdo a las restricciones que se han establecido en la descripción del problema. Los agentes no pueden ser menos de 20.

Para la asignación de cada agente a una comunidad se sugiere mantener una distribución de Zipf sobre las poblaciones de las comunidades, pero se pueden implementar otras fórmulas de carácter aleatorio.

Agentes

Cada agente se define por:

- su memoria
- su protocolo de comunicación

Lo que cada agente tiene en su memoria varía en cada tiempo t_i , y corresponde a una lista de palabras para cada objeto. La lista puede ser vacía. En el estado inicial, t_0 , todos los agentes tienen la memoria vacía. Cuando un individuo tiene que comunicarse, y no tiene almacenada ninguna palabra, tiene que buscar una. Esto se puede hacer de dos maneras:

- El agente inventa la palabra aleatoriamente
- Se acude a una bolsa de palabras de tamaño $N \times O$.

Cada individuo tiene memoria para guardar M palabras para cada objeto. La actuación de cada memoria al recibir una palabra de otro agente puede funcionar por sustitución, como en el protocolo del artículo, o de forma probabilística: cada palabra de la lista tiene una probabilidad, que se incrementa o decrece con la interacción.

Protocolos de comunicación

El experimento de Baronchelli establece un sólo protocolo de comunicación para todos los agentes. El proyecto permite definir diferentes protocolos para diferentes clases de agentes o comunidades.

Cada individuo tiene una probabilidad P de tener comunicación con un individuo de cada una de las comunidades dependiendo de los protocolos de comunicación diseñados. Estos pueden tener diferentes distribuciones de probabilidad entre la propia comunidad y las otras, o bien se puede asignar “embajadores” específicos en cada comunidad, que sean los que interactúan con las demás.

En cada unidad de tiempo t tiene que haber comunicación entre un par de agentes. Se debe distribuir entre ellos el rol de hablante oyente y sobre qué objeto(s) se van a comunicar.

Parámetros de estudio y presentación

Los datos a estudiar son, con las diferentes configuraciones:

- T_{conv} : tiempo que tarda el sistema en ponerse de acuerdo.
- T_{max} : momento de la computación en que hay un mayor número de palabras totales en el sistema.
- W_{max} : número máximo de palabras totales.
- W_{dif} : número máximo de palabras diferentes.

Se pide analizar la influencia de los parámetros N, C, O, y los protocolos de comunicación en cada una de estas variables, especialmente en el tiempo de convergencia T_{conv} .

Restricciones

- Llegar al menos en algún experimento a 100 agentes,
- Usar al menos dos valores de O, siendo O el número de objetos,
- Implementar el programa con al menos dos protocolos de comunicación diferentes.

Entrega: Python, reporte en Latex formato lncs y presentación en clase.

Este es un trabajo de investigación experimental donde se pide que tomen decisiones, implementen, comparen, infieran, y saquen conclusiones. El reporte debe explicar su metodología, pero también, en forma de artículo científico, debe explicar a qué puntos han llegado.

Bibliografía

- Joachim De Beule, Bart De Vylder, and Tony Belpaeme. A cross-situational learning algorithm for damping homonymy in the guessing game. In Luis Mateus Rocha, Larry S. Yaeger, Mark A. Bedau, Dario Floreano, Robert L. Goldstone, and Alessandro Vespignani, editors, *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 466–472, Cambridge, MA, 2006. MIT Press.
- Luc Steels and Angus McIntyre. Spatially distributed naming games. *Advances in Complex Systems*, 1(4):301–323, January 1999. doi: 10.1142/S021952599800020X.
- Remi van Trijp. The emergence of semantic roles in Fluid Construction Grammar. In Andrew D.M. Smith, Kenny Smith, and Ramon Ferrer i Cancho, editors, *The Evolution of Language. Proceedings of the 7th International Conference (EVOLANG 7)*, pages 346–353, Singapore, 2008. World Scientific Press.
- Ludwig Wittgenstein. *Philosophische Untersuchungen*. Suhrkamp, Frankfurt, 1967.