

EXPERIMENTO ECONÓMICO BASADO EN AGENTES

Ian Derek Farren Gause

ABSTRACT

El estudio de la inteligencia de grupo (swarm intelligence) consiste en analizar el comportamiento global de una población que se forma a partir de agentes básicos. Estos agentes actúan individualmente de manera muy simple, pero el resultado colectivo es un actuar bastante complejo (hormigas, termitas, abejas, etc.). En este trabajo, se presenta un acercamiento al estudio de la inteligencia de grupo basado en agentes más complejos. Estos agentes tienen la capacidad de aprender y por lo tanto desarrollarse. Los agentes presentados son seres humanos, y el ambiente en el que se desenvuelven es una economía privada y cerrada representada por un mercado en el cual los agentes negocian el intercambio de un bien. Se presenta además, la utilidad de la computación en las ciencias sociales.

Keywords: Swarm Intelligence, Agent based simulation, Neural Networks, Computational Intelligence.

1. Introducción

La intención de este trabajo es mostrar la utilidad que tiene la computación en las ciencias sociales. Para esto se dará un primer paso en el estudio de la inteligencia de grupo (swarm intelligence) basado en agentes complejos. Estos agentes tienen, a diferencia de los agentes básicos usados tradicionalmente (hormigas, termitas, etc.), la capacidad de aprender de sus aciertos y errores históricos. Los agentes utilizados en este trabajo son seres humanos. Los seres humanos son los agentes más complejos que existen, y su comportamiento global es aún más complejo, razón por la cual es necesario partir por un estudio de los agentes humanos y así simular en un computador el comportamiento de la comunidad. La economía es la ciencia que se ve más beneficiada con este nuevo paradigma.

2. Las Ciencias Sociales como Ciencias “duras”

Herbert Simon (Premio Nobel de Economía en 1978) argumentaba que las ciencias sociales son en realidad las ciencias “duras”. Simon planteaba que muchos de los procesos sociales de crucial

importancia eran complejos y pueden ser cuidadosamente descompuestos en subprocesos de tipo económico, demográfico, cultural, espacial. Estos subprocesos pueden ser estudiados en forma agregada para analizar así el proceso social como un todo.

Además, las ciencias sociales son ciencias “duras” por que ciertos tipos de experimentación controlada son “duras”. En particular, es difícil testear la hipótesis sobre la relación existente entre el comportamiento del individuo y su entorno, hipótesis del tipo: si los individuos se comportan de tal forma (siguiendo reglas específicas) entonces la sociedad como un todo mostrará alguna propiedad particular. ¿Cómo logra generar el heterogéneo micro-mundo de los comportamientos individuales las regularidades globales macroscópicas de la sociedad?

Otra preocupación de importancia para cuentistas sociales es que el actor racional, esto es, una persona informada con infinita capacidad de computar información que maximiza una función fija de utilidad, tiene muy poca relación con un ser humano. Aún así, no hay ninguna metodología natural para relajar estos supuestos del individuo.

Es una práctica común suprimir la heterogeneidad del individuo real en la construcción de modelos. Esto se hace en forma explícita (como en modelos basados en agentes representativos en macroeconomía), o bien implícita, cuando modelos altamente agregados son usados para representar procesos sociales. Si bien de estos modelos se puede sacar información valiosa, ellos filtran todas las consecuencias dadas por la heterogeneidad del sistema. Aunque pocos científicos sociales negarían que estas consecuencias pueden ser crucialmente importantes, no ha habido ninguna metodología propuesta para estudiar poblaciones altamente heterogéneas.

Las ciencias sociales, especialmente la teoría de juegos y la teoría del equilibrio general, han sido formuladas en base a un supuesto equilibrio estático, y han ignorado la dinámica del tiempo. Nuevamente, no ha habido ninguna metodología propuesta para estudiar la dinámica en desequilibrio de sistemas sociales.

3. Computadores: ¿complementos o sustitutos?

Con el acelerado crecimiento que presenta el poder de los computadores, se presentan nuevas oportunidades y potencial a las ciencias económicas. Sin embargo, como sucede con cualquier herramienta nueva, existe la discusión acerca de cual es el rol que le corresponde a la computación dentro de la economía. Algunos roles son obvios, por lo que gran parte de los científicos están de acuerdo en que el análisis computacional es necesario y ofrece una guía en la discusión de políticas. Sin embargo, la real discusión se encuentra al momento de proponer a la computación, y “ejemplos” generados por computadoras, como sustituto a la clásica forma de análisis de teorías económicas basada en assumption-teorema-prueba. Dentro del ambiente económico, el rol de la computación está definido como complemento y como un ingrediente de gran ayuda en actividades de investigación. Sin embargo, la teoría computacional pareciera ser un sustituto a los análisis teóricos convencionales.

Ostrom (1988), propuso que existen tres medios formas de investigar para científicos sociales, estas son mediante argumentación verbal, matemáticas y simulación computacional. La simulación computacional, consiste en representar el modelo como un programa computacional. La programación computacional puede ser usada para estudiar tanto teorías cuantitativas como cualitativas. La lógica detrás del uso de programación computacional no es muy diferente a la lógica usada en los conocidos modelos estadísticos. En ambos caso hay un fenómeno que los investigadores quieren entender mejor. Este es el objetivo. Los investigadores entonces construyen un modelo del objetivo mediante un proceso de abstracción (este modelo puede ser una serie de ecuaciones matemáticas, una ecuación estadística, como lo sería una regresión, o un programa

computacional). Luego, ellos observan el comportamiento del modelo y lo comparan con observaciones del mundo social real. Si la información entregada por el modelo es lo suficientemente parecida a la del mundo social real, se usa esto como evidencia a favor de la validación del modelo (o se usa la diferencia entre la información entregada por el modelo y el mundo social real como evidencia para rechazar el modelo).

4. Modelos de Simulación basados en agentes

4.1.- Características de los modelos de Simulación basados en agentes

Los modelos basados en agentes se definen por estar constituidos por agentes que interactúan entre sí y con el entorno de una forma independiente. Estos agentes consisten en pequeños programas que son capaces de controlar sus propias acciones basándose en su percepción, completa o parcial, del entorno en que se mueven (Huhns y Singht, 1998). En la mayoría de los casos, los agentes buscan la consecución de algún tipo de objetivo, la supervivencia en el medio, la obtención de beneficios o el crecimiento, conceptos todos ellos aplicados más frecuentemente a las personas que a programas informáticos (Gilbert y Troitzsch, 1999).

Los agentes pueden ser entidades bastante básicas (como lo sería una hormiga) o bien seres con capacidad de razonar y aprender (como lo serían los seres humanos). La programación de agentes básicos es bastante simple, pero la programación de agentes avanzados tiene un grado de complejidad muy alto. Si bien las emociones de los seres humanos no pueden ser programadas, muchas de sus decisiones y aprendizajes sí. Para lograr esto se utilizan redes neuronales y algoritmos genéticos que proporcionan un aprendizaje y adaptación realista y válida. Las principales aplicaciones de los modelos de simulación basados en agentes son:

- Economía
- Ciencias Políticas
- Cultura/Antropología
- Ecología
- Biología
- Física
- Geografía
- Ciencias Computacionales
- Negocios/Industria
- Ciencias Militares

Aunque las características que pueden tener los agentes son muy amplias, Wooldridge y Jennings (1995) señalaron algunas que suelen mostrarse en la mayor parte de los casos:

- *Autonomía:* Los agentes actúan de forma independiente no estando controladas desde el exterior ni sus acciones ni su estado interno.
- *Habilidad social:* Los agentes interactúan entre sí mediante algún tipo de lenguaje.
- *Reactividad:* Los agentes son capaces de percibir su entorno y responder a los estímulos recibidos.
- *Pro actividad:* Los agentes no son sólo capaces de reaccionar frente al entorno sino que son capaces de llevar a cabo acciones por iniciativa propia para así alcanzar un objetivo.

Estas características coinciden en gran medida con las que podría esperarse del comportamiento de las empresas en los mercados.

Las empresas gozan de autonomía en sus decisiones, no estando controladas por sus competidores; tienen habilidad social, ya que son capaces de interactuar en el mercado dando y recibiendo información: son asimismo reactivas, pues observan el mercado en el que compiten y son capaces de responder a los estímulos que reciben; y por último, son pro activas pues pueden poner en práctica estrategias que les permitan alcanzar sus objetivos finales.

Gilbert y Troitzsch (1999) han propuesto una serie de etapas que han de seguirse en el desarrollo de cualquier modelo de simulación, incluidos los basados en agentes, si se desea que sean metodológicamente correctos.

En primer lugar, ha de partirse de la identificación de un problema o de un fenómeno que se desea estudiar; es decir, se ha de establecer un objetivo. En nuestro caso, el objetivo consistirá en determinar el efecto que las características de las empresas entrantes tienen sobre la estructura del mercado.

Una vez establecido, se hace necesario definirlo mediante la construcción de un modelo que interrelacione las distintas variables que intervienen. Generalmente, es necesario contar en esta etapa de algún tipo de observaciones que permita dotar al modelo de los parámetros iniciales necesarios.

A partir de este punto, se desarrolla el programa informático de simulación, se ejecuta y se almacenan los resultados obtenidos para su posterior análisis.

Junto a estas etapas principales, existen otras no menos importantes pero que a menudo no reciben la atención que merecen y que son, en gran medida, la causa de la escasa confianza que muchos economistas tienen en la simulación.

Estas etapas son las de verificación, validación y análisis de sensibilidad. En la fase de verificación ha de comprobarse el funcionamiento informático del modelo. Debe por lo tanto eliminarse en esta fase todos los posibles errores de programación, *bugs*, que hayan podido quedar ocultos en el código. Esta búsqueda de errores se ve dificultada gravemente por el hecho de que en la mayor parte de los modelos sociales las simulaciones dependen de números pseudo aleatorios que simulan los efectos de los efectos aleatorios y de las variables inobservables (Gilbert, 1996), por lo que al diferir en cada simulación los resultados obtenidos resulta complicado comprobar su validez.

En la fase de validación, se comprueba que la simulación se comporta de acuerdo con el modelo de partida y es una imagen fiel del mismo. Para ello, es necesario que el modelo sea compatible con las especificaciones establecidas en la etapa de desarrollo del modelo y con la evidencia disponible.

Por último, es necesario realizar un análisis de sensibilidad para apreciar el grado de robustez de la simulación frente a pequeños cambios en los parámetros y las condiciones iniciales.

4.2.- Construcción de un modelo económico basado en Agentes

La idea es construir un modelo computacional en el cual los agentes (piezas de software) operan (se comportan) (Gilbert y Terna, 2000).

Primero, hay que considerar a los agentes como objetos, esto es, como piezas de software capaces de contener datos y reglas para trabajar en los datos. Las reglas poseen los mecanismos necesarios para reaccionar a mensajes que vienen desde el exterior del objeto.

Segundo, tenemos que observar el comportamiento individual de cada agente analizando los estados de la variable interna, y al mismo tiempo, los resultados que nacen del comportamiento colectivo. Un factor importante que debe cumplir el software es la sincronización de los tiempos, esto quiere decir que tenemos que estar seguros de que las observaciones hechas a nivel agregado y el conocimiento que estamos recogiendo de los estados internos de los agentes sean consistentes en términos del programa del experimento.

Ahora, supongamos que una comunidad de agentes que actúan en base a información pública (común para todos los agentes) y privada (específica para cada agente) y a reglas locales simples muestra, como un todo, un comportamiento interesante, complejo o inesperado. Hay dos tipos de comportamientos inesperados: comportamientos imprevistos y comportamientos impredecibles. Un comportamiento imprevisto sería, por ejemplo, cuando buscamos un estado de equilibrio en el modelo, por aparece algún tipo de comportamiento cíclico. Un comportamiento impredecible ocurre cuando, por ejemplo, el experimento entrega datos caóticos.

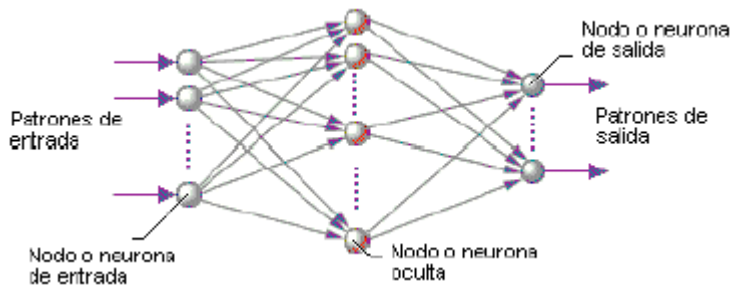
El modelo entregado en este trabajo representa una economía cerrada y privada, con tres agentes productores y tres agentes consumidores. Cada agente ofrece y demanda un producto dependiendo de su función. Los agentes tienen la capacidad de aprender (al igual que en la realidad) de forma de adaptarse en el proceso de negociación para así lograr el intercambio. El aprendizaje de los agentes viene dado por dos principios: consistencia interna y consistencia externa. La consistencia interna de los agentes hace referencia a su capacidad de evaluar en forma coherente lo que tiene que hacer para lograr cierto resultado y pronosticar el efecto de sus actos. De esta forma se produce un aprendizaje cruzado, el agente pronostica los efectos de sus actos por un lado (¿que pasaría si actuó de tal forma?) y los actos de sus efectos por otro (¿que tengo que hacer para lograr tal efecto?). Esta consistencia interna es la base de toda decisión para los seres humanos. La consistencia externa tiene relación con la capacidad de los agentes de adaptarse a los requerimientos de otro agente, esto es, entregar o quitar en función de mi objetivo. Ambas consistencias se entrenan y son, en definitiva, las que hacen la diferencia entre el agente exitoso y el que no lo es (tal como funciona en la vida real).

Las decisiones de los agentes de este trabajo fueron representadas por redes neuronales debido a su capacidad de aprendizaje.

4.2.1.- Redes Neuronales

La idea de incluir redes neuronales en un modelo basado en agentes se basa en la capacidad que tienen estas funciones vectoriales como estructuras muy cercanas a las funciones de regresión no lineal multidimensional.

Las redes neuronales son un sistema de conexiones que unen muchas unidades de procesamiento. Estas unidades de procesamiento pueden ser consideradas como una versión simplificada de una neurona, donde los parámetros w_{ij} representan el peso que se le da a cada



valor de entrada a_j . La entrada final a cada neurona es la suma ponderada según los pesos w_{ij} de los valores a_j . Es interesante mencionar que la misma operación se puede representar en lenguaje algebraico matricial como Ax , donde x es el vector que contiene los a_j valores de entrada y A es la matriz que contiene los pesos w_{ij} .

$$y = g(B \cdot f(Ax))$$

La red utilizada para representar el aprendizaje de los agentes en este trabajo, tiene tres capas: una capa de entrada, una capa oculta y una capa de salida. La red posee funciones de activación en cada una de las unidades de procesamiento. La función de activación es una función que transforma en forma no lineal la suma ponderada de los valores de entrada a_j . La función usada en la red de este trabajo es la función de activación logística:

$$f(z) = \frac{1}{1 + e^{-z}}$$

4.2.2.- Aprendizaje

Para encontrar los valores w_{ij} (pesos) de la red usamos el método "Backpropagation"

El algoritmo de aprendizaje Backpropagation proporciona una forma de entrenar una red multicapa con alimentación hacia adelante. Comienza alimentando los valores de la entrada de acuerdo a las siguientes ecuaciones:

$$net_i = \sum_{j \in A} O_j W_{ji} \quad \forall i : i \in B$$

donde A es el grupo de neuronas en una capa, y B es la otra. O_j es la activación para la neurona j , y W_{ji} son los pesos asignados a la conexión entre las neuronas j e i . En la ecuación anterior, se toman los valores de salida y se alimentan a la siguiente capa a través de los pesos. Esta operación se realiza para cada neurona en la siguiente capa, produciendo un valor de red para el. Este valor es la suma de todos los valores de activación en las neuronas de la capa anterior, y cada valor de red es aplicado ahora a la siguiente ecuación, conocida como función de activación, para producir la activación de esa neurona.

$$O_i = f(net_i) = \frac{1}{1 + e^{-net_i}}$$

Después de que todas las neuronas tienen un valor de activación asociado a un patrón de valores de entrada, el algoritmo sigue buscando errores en cada neurona que no es entrada.

Los errores encontrados para las neuronas de salidas, son propagados hacia atrás, a la capa anterior para que puedan ser asignados a neuronas de las capas escondidas, esto se calcula por:

$$\delta_i = f'(net_i) \sum_{j \in E} \delta_j W_{ij} \quad \forall i : i \in D$$

donde D es el grupo de neuronas en una capa que no es de entrada y E es el grupo de neuronas de la siguiente capa. Este cálculo se repite para cada capa escondida en la red.

Después de que se ha encontrado la activación y el error asociado a cada grupo de neuronas, los pesos se actualizan, primero encontrando el valor que cada peso debe modificarse, esto se logra calculando:

$$\Delta W_{ij} = CO_i \delta_j \quad \forall i, j : i \in A, j \in B$$

donde C, conocida como la razón de aprendizaje, es una constante que controla el valor del cambio de los pesos y W_{ij} es el cambio de los pesos entre la neurona i y j. El peso es cambiado evaluando:

$$W_{ij,t+1} = W_{ij,t} + \Delta W_{ij}$$

4.2.3.- Aprendizaje cruzado

Para desarrollar este modelo se estableció la siguiente hipótesis general: un agente, actuando en un ambiente económico, debe desarrollar y adaptar su capacidad de evaluar, en forma coherente:

1. Lo que tiene que hacer para obtener un resultado específico.
2. Como pronosticar las consecuencias de sus actos.

Esta hipótesis se mantiene para el caso en que el agente interactúa con otros agentes. Mas allá de esta consistencia interna, agentes pueden desarrollar otras características, por ejemplo la capacidad de actuar o evaluar los efectos de sus actos según sugerencias del ambiente (seguir reglas sociales) o de otros agentes (imitarlos). Esto es consistencia externa.

El comportamiento económico, ya sea simple o complejo, es producto de la consistencia interna y la consistencia externa del agente. Para un observador externo, los agentes aparentemente operan en base a objetivos y planes. Obviamente ellos no poseen la capacidad de producir tales entidades, las que son invenciones del observador. Es interesante destacar que observaciones de agentes económicos en el mundo real (seres humanos) por parte de un observador externo, puede producir el mismo error.

El nombre aprendizaje cruzado, representa la técnica usada para entrenar la red neuronal que representa las decisiones tomadas por cada agente manteniendo una consistencia interna. Dada la hipótesis general, el comportamiento de cada agente produce resultados que pueden ser separados en dos tipos:

1. Actos a realizar.
2. Pronóstico de los efectos de los actos a realizar.

Los objetivos necesarios para entrenar la red desde el punto de vista de los actos a realizar y los objetivos necesarios para entrenar la red desde el punto de vista del pronóstico de los actos están contruidos en forma cruzada. Los objetivos de los actos a realizar están contruidos en función de los pronósticos de los efectos de los actos a realizar, de forma de desarrollar la capacidad de

decidir acciones que produzcan efectos cercanos a los resultados esperados. A su vez, los objetivos de los efectos pronosticados por el agente están contruidos en función de los actos realizados por el agente, de forma de desarrollar la capacidad de estimar los efectos de los actos que el mismo agente está decidiendo. La base de este método de aprendizaje es aprender adivinando y haciendo, es decir, si bien el algoritmo de aprendizaje es Backpropagation, es a base de aprender y actuar: es necesaria la acción para producir la información necesaria para entrenar la red que simula al agente. Un entrenamiento tradicional no sería útil en este caso, ya que las reglas que relacionan las entradas y las salidas de la red tienen que ser "descubiertas" por medio de experimentos realizados por la misma red.

El aprendizaje es realizado mediante cuatro pasos por iteración. Si consideramos cada iteración como un día en la vida de nuestros agentes, cada agente tiene una experiencia de negociación diaria, de la cual aprende en cuatro pasos que estiman la salida de la red y propaga el error cuadrático medio a través de la misma, determinando los pesos de la red. Los pasos de aprendizaje son:

1. Salidas de la red: Acciones a realizar y efectos de estas acciones
2. Objetivos para los efectos: Aparecen de las acciones a realizar. De esta forma los efectos esperados de los actos realizados son cada vez mas parecidos a los efectos reales de los actos realizados.
3. Objetivos para los actos: El error encontrado entre los objetivos para los efectos y los efectos pronosticados por la red en (2) son el punto de partida para determinar los objetivos de los actos. Así, los actos evolucionan en el sentido de los efectos pronosticados. Este error es compartido en forma aleatoria por los diversos actos.
4. Backpropagation: los pesos de la red se corrigen de forma de minimizar los errores y así conseguir pronósticos de actos más cercanos a los efectos reales de los actos y actuar de forma más coherente con los pronósticos esperados de esos actos.

4.2.4.- Consistencia interna

Formalmente, el entrenamiento cruzado puede ser genéricamente explicado introduciendo el efecto E que aparece de realizar dos acciones A_1 y A_2 . E , A_1 y A_2 son salidas de la red. El objetivo para el efecto es:

$$E^o = f(A_1, A_2)$$

donde f es una función que representa el efecto real de las acciones A_1 y A_2 .

La intención es encontrar una salida E (efecto pronosticado) lo mas cercana posible a E^o , que es el efecto real. El error de E es:

$$e = E^o - E$$

Para minimizar este error, es propagado a través de toda la red.

La intención ahora es encontrar los actos a realizar que sean más consistentes con efectos pronosticados, es decir, encontrar los objetivos para los actos (A_1^o , A_2^o) que son coherentes con el efecto entregado por la red (E) para así corregir los valores A_1 y A_2 . Para esto definimos las funciones g_1 y g_2 que representan las funciones inversas de f para cada una de las variables A_1 y A_2 .

$$A_1 = g_1(E, A_2)$$

$$A_2 = g_2(E, A_1)$$

Definiendo la variable aleatoria $\phi_1 \sim U(0,1)$ (proveniente de una distribución uniforme en el intervalo $[0,1]$), y $\phi_2 = 1 - \phi_1$, tenemos que:

$$A_1^o = g_1(E^o - e \cdot \phi_1, A_2)$$

$$A_2^o = g_2(E^o - e \cdot \phi_2, A_1)$$

Los errores de A_1 y A_2 son:

$$a_1 = A_1^o - A_1$$

$$a_2 = A_2^o - A_2$$

La idea es minimizar los errores e , a_1 , a_2 mediante backpropagation y representar así a un agente internamente coherente.

4.2.5.- Consistencia externa

Cada agente buscará realizar actos que vayan en línea con sus intereses dentro de la sociedad. Para esto debe tener en cuenta el efecto que estos actos hacen en los actos de otros agentes. Recordemos que la intención de cada agente es maximizar su beneficio, y muchas veces un aumento en el beneficio de un agente significa una disminución de beneficio en otro. Para lograr un equilibrio, cada agente regula sus actos en función del actuar de otros agentes. Este experimento utilizó el objetivo de entrenamiento de estos actos como el promedio entre los actos de los agentes.

5.- Estructura del experimento

5.1.- Entradas y salidas de las redes neuronales

En particular para el experimento, las redes neuronales que representan a los consumidores y productores tienen tres entradas y tres salidas compuestas por:

Consumidores:

Las redes neuronales que representan a los agentes consumidores tienen las siguientes variables de entrada: Gasto $_{t-1}$, el precio x cantidad que el agente pagó en el periodo anterior; Precio demandado $_{t-1}$, el precio del bien propuesto por el consumidor en el periodo anterior; Cantidad demandada $_{t-1}$, la cantidad propuesta por el consumidor en el periodo anterior.

Las variables de salida son: Gasto $_t$, el pronóstico del gasto para este periodo, lo que el agente cree que gastará. Esta salida tiene relación únicamente con la consistencia interna del agente. Precio demandado $_t$, el precio propuesto por el agente consumidor para intercambiar. Esta salida representa las características tanto internas como externas del comportamiento del agente, ya que el precio a demandar depende de la consistencia interna que éste tiene con el gasto pronosticado por un lado, y de la consistencia externa que tiene con el proceso de negociación con los agentes productores por otro. Así, el agente demanda precios de acuerdo con lo que él espera gastar, considerando además que el precio sea el mínimo dentro de los parámetros de precios razonables que lograrían un potencial intercambio con el productor. Esta negociación la hacen (conciente o inconscientemente) todos los seres humanos.

Productores:

Las redes neuronales que representan a los agentes productores tienen las siguientes variables de entrada: Ingreso_{t-1}, el ingreso del periodo anterior; Precio ofrecido_{t-1}, el precio propuesto por el productor en el periodo anterior; Cantidad ofrecida_{t-1}, la cantidad puesta en el mercado por el productor para vender en el periodo anterior.

Las variables de salida son: Ingreso_t, el pronóstico del ingreso para este periodo, el dinero que el agente cree que recibirá por ventas en este periodo. Esta salida tiene relación únicamente con la consistencia interna del agente. Precio ofrecido_t, el precio propuesto por el agente productor para intercambiar. Esta salida representa las características tanto internas como externas del comportamiento del agente, ya que, al igual que el caso anterior, el precio a ofrecer depende de la consistencia interna que éste tiene con el ingreso pronosticado por un lado, y de la consistencia externa que tiene con el proceso de negociación con los agentes consumidores por otro. Así, el agente ofrece precios de acuerdo con lo que él espera recibir como ingreso, considerando además que el precio sea el mínimo dentro de los parámetros de precios razonables que lograrían un potencial intercambio con el vendedor. Aquí estamos viendo justamente el otro lado de la moneda en la negociación.

5.2.- Objetivos de entrenamiento

Consumidores:

Para la red que representa el aprendizaje de un consumidor, el entrenamiento cruzado está representado por los siguientes objetivos de entrenamiento:

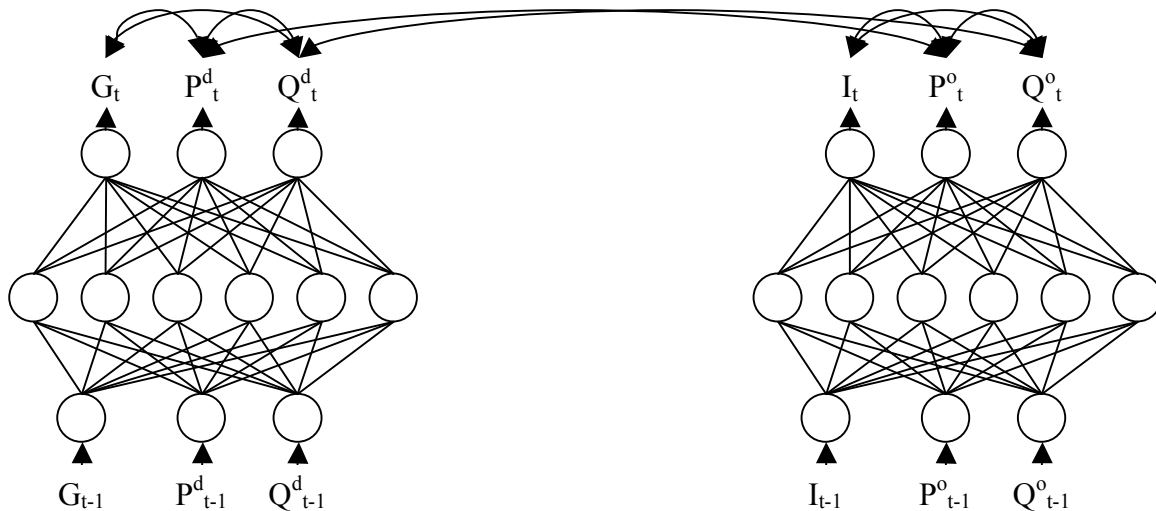
- Objetivo para el gasto: su objetivo es Precio demandado_t x Cantidad demandada_t. Así el agente logra una consistencia interna.
- Precio demandado. Su objetivo es una combinación entre $\frac{\text{gasto} - e \cdot n}{\text{cantidad demandada}}$ (ver consistencia interna) y $\frac{\text{precio oferta} + \text{precio demanda}}{2}$ (ver consistencia externa).
- Cantidad demandada: Su objetivo es una combinación entre $\frac{\text{gasto} - e \cdot n}{\text{precio demandado}}$ (ver consistencia interna) y $\frac{\text{cantidad oferta} + \text{cantidad demanda}}{2}$ (ver consistencia externa).

Productores:

- Objetivo para el ingreso: su objetivo es Precio ofrecido_t x Cantidad ofrecida_t. Así el agente logra una consistencia interna.
- Precio ofrecido. Su objetivo es una combinación entre $\frac{\text{ingreso} - e \cdot n}{\text{cantidad ofrecida}}$ (ver consistencia interna) y $\frac{\text{precio oferta} + \text{precio demanda}}{2}$ (ver consistencia externa).

- Cantidad demandada: Su objetivo es una combinación entre $\frac{\text{ingreso} - e \cdot n}{\text{precio ofrecido}}$ (ver consistencia interna) y $\frac{\text{cantidad oferta} + \text{cantidad demanda}}{2}$ (ver consistencia externa).

El siguiente dibujo muestra las relaciones de aprendizaje entre una red que representa el aprendizaje de un consumidor y un red que representa el aprendizaje de un productor.



5.3.- Detalles técnicos

En el experimento se creó un ambiente de relación entre tres agentes consumidores y tres agentes productores representado por un programa escrito en lenguaje matlab. Cada "día" se encuentran todos los consumidores con todos los productores y si el precio de algún producto es menor o igual al precio dispuesto a pagar por el consumidor, se realiza la transacción. La cantidad a transar será la mínima cantidad entre la demandada por el consumidor y la ofrecida por el productor. Un "día" para el programa es una iteración. El programa corrió durante 20 días o 20 iteraciones.

Las redes neuronales utilizadas tienen una capa oculta, ya que más capas no mejoraban el pronóstico. Experimentalmente seis neuronas en la capa oculta fueron las que dieron resultados más estables. Todas las neuronas poseen funciones de activación logísticas excepto en la capa de salida donde las funciones de activación son identidad. El algoritmo de entrenamiento utilizado es el de Levenberg-Marquardt. Este algoritmo es una modificación del método de Newton, el que fue diseñado para minimizar funciones que sean la suma de los cuadrados de otras funciones no lineales; es por ello que el algoritmo de Levenberg - Marquardt, tiene un excelente desempeño en el entrenamiento de redes neuronales donde el rendimiento de la red está determinado por el error medio cuadrático.

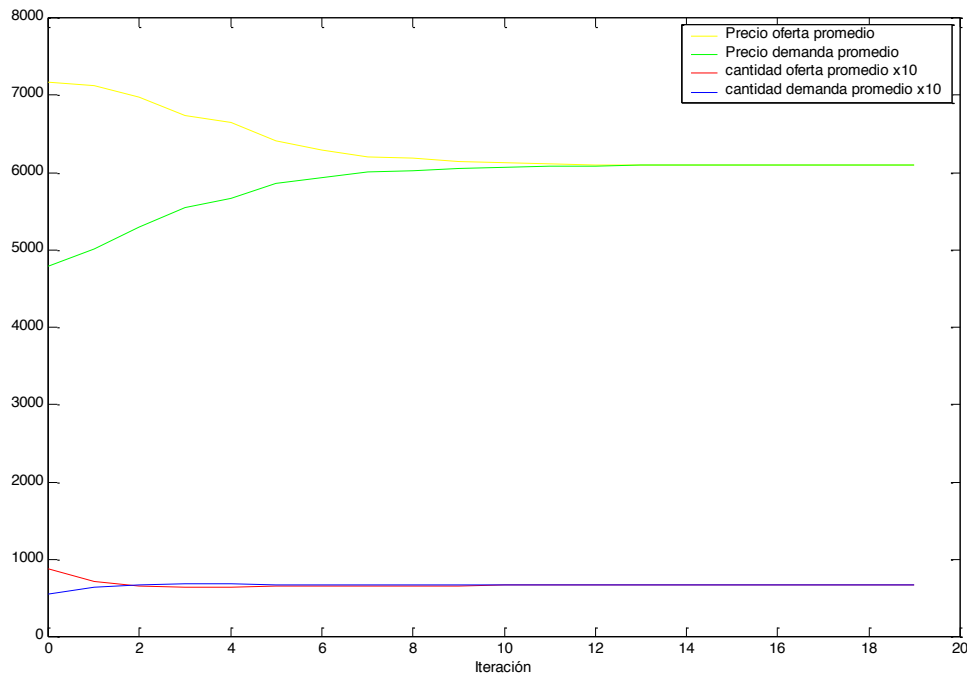
Los actos de los agentes que representan la cantidad o el precio a ofrecer o demandar tienen que cumplir con consistencias tanto internas como externas. En el entrenamiento de estos actos se utilizó un objetivo ponderado entre el objetivo que representa la consistencia interna y el objetivo que representa la consistencia externa. La ponderación de estos objetivos varía en la vida real

dependiendo de factores psicológicos del individuo. En este experimento los objetivos de consistencia interna y externa fueron ponderados igualmente cada uno (factor = 0,5). Las variables precio y cantidad iniciales del programa son variables aleatorias. El precio inicial propuesto por los productores tiene una distribución $N(9.000,1000)$ y el de los consumidores $N(3.000,1000)$. Las cantidades iniciales propuestas a transar tiene una distribución $N(100,10)$ para los productores y $N(50,10)$ para los consumidores. Las variables iniciales de ingreso es 900.000 y de gasto es 300.000.

Si bien el programa fue escrito en matlab y no en un lenguaje caracterizado por la orientación al objeto (como lo es objC, por ejemplo), la intención fue que este código tuviera las principales características de un programa orientado al objeto (NeXT, 1993). Por esta razón el código se divide en tres sub-programas: Un programa que representa el actuar de los agentes consumidores, un programa que representa el actuar de los agentes productores y un programa que representa las reglas del ambiente (el mercado). Estos programas están disponibles en www.2mstudios.cl/dfarren/ bajo los nombres mercado.m, consumidores.m y vendedores.m. Además estos programas se pueden ver: en el anexo 1 para el mercado o ambiente, anexo 2 para los agentes consumidores y anexo 3 para los agentes productores.

5.4.- Resultados

El siguiente gráfico muestra la evolución del actuar de los agentes en el tiempo. La información graficada son los promedios de los precios y cantidades x 10 de los dos grupos de agentes, consumidores y productores:



Debido al rápido aprendizaje de los agentes, un equilibrio espontáneo de mercado emerge a contar del décimo día de negociación, ya estaban ofreciendo y demandando aproximadamente el mismo precio.

Esta velocidad de aprendizaje puede ser modificada cambiando el algoritmo de entrenamiento con la intención de validar el modelo.

La cantidad a ofrecer y demandar tuvo una velocidad de aprendizaje similar, pero dada su menor diferencia inicial, a los dos días los agentes estaban ofreciendo y demandando las mismas cantidades.

La consistencia de los agentes queda mostrada en la capacidad de llegar al acuerdo de negociación sostenido.

En el largo plazo cada agente transa aproximadamente $6.000 \times 75 = 450.000$ U.M./día, lo que para este experimento significa un producto de $450.000 \times 3 = 1.350.000$ UM/día y un producto per cápita de $1.350.000/6 = 225.000$ U.M.

6. Conclusiones

Debido al gran avance computacional de la actualidad, es posible utilizar estas ventajas para la resolución de problemas reales difíciles o imposibles de manipular por la vía de la modelación matemática. Por esta razón la simulación mediante ordenador es hoy en día una práctica común en todas las disciplinas de la ingeniería e investigación. Además, el desarrollo computacional ha contribuido notablemente en el estudio de la inteligencia computacional creando computadores y modelos capaces de tomar decisiones y aprender.

Estos avances han sido justamente el pilar de este trabajo. El modelo de simulación basado en agentes mostrado es un primer paso en un estudio mas profundo del comportamiento de grupos sociales mediante la computación. La simulación de agentes inteligentes tiene ventajas significativas en diversas disciplinas, en especial en economía. El estudio de este trabajo mostró la validez de la simulación basada en agentes y su potencial en el desarrollo de las ciencias sociales.

Se presentó un modelo representativo de agentes consistentes, capaces de interactuar entre si. El resultado de la simulación mostró como el aprendizaje de los agentes lleva a un proceso de negociación. La negociación consideró una consistencia interna y externa del agente. La consistencia interna fue representada por la capacidad del agente de predecir su beneficio, y así tomar decisiones de negociación según esta predicción. La consistencia externa fue representada por la capacidad del agente de tomar decisiones de negociación en función de los otros agentes. Ambos tipos de consistencia son la principal característica en la tome de decisiones de los seres humanos.

Anexo 1

```
%Programa que representa al ambiente en el cual se desenvuelven los
agentes. El ambiente es un mercado, y los agentes son
%consumidores y productores

clear all

global vendedor1
global vendedor2
global vendedor3
vendedor1=newff([0 12000; 0 12000; 0 12000],[6 3],{'logsig'
'purelin'},'trainlm'); %definición de redes vendedores
vendedor2=newff([0 12000; 0 12000; 0 12000],[6 3],{'logsig'
'purelin'},'trainlm');
vendedor3=newff([0 12000; 0 12000; 0 12000],[6 3],{'logsig'
'purelin'},'trainlm');

global consumidor1
global consumidor2
global consumidor3
consumidor1 = newff([0 12000; 0 12000; 0 12000],[6 3],{'logsig'
'purelin'},'trainlm'); %definición de redes consumidores
consumidor2 = newff([0 12000; 0 12000; 0 12000],[6 3],{'logsig'
'purelin'},'trainlm');
consumidor3 = newff([0 12000; 0 12000; 0 12000],[6 3],{'logsig'
'purelin'},'trainlm');

global precio_oo
global cantidad_oo
global precio_dda
global cantidad_dda
precio_oo=normrnd(9000, 1000, 1, 3); %precios de oferta
iniciales
cantidad_oo=normrnd(100, 10, 1, 3); %cantidades de oferta
iniciales
precio_dda=normrnd(3000, 1000, 1, 3); %precios de demanda
iniciales
cantidad_dda=normrnd(50, 10, 1, 3); %cantidades de demanda
iniciales

global ingreso
global ingreso_objetivo
global precio_oo_objetivo
global cantidad_oo_objetivo

global gasto
global gasto_objetivo
global precio_dda_objetivo
global cantidad_dda_objetivo

global n
global e

ingreso=[900000 900000 900000]; %ingresos iniciales
```

```

gasto=[300000 300000 300000];           %gastos iniciales
ingreso_real=zeros(1,3);
gasto_real=zeros(1,3);
k=0;
while k<20
    i=1;
    j=1;

    n=rand;
    vendedores
    consumidores
        while i<3                       %todos los consumidores se encuentran
con todos los productores en un día
            while j<3
                if precio_oo(i)<precio_dda(j)      %la transacción se
realiza solo si el precio ofrecido es menor al precio demandado

ingreso_real(i)=precio_oo(i)*min(cantidad_oo(i),cantidad_dda(j));    %la
cantidad a transar es la menor entre
                gasto_real(j)=ingreso_real(i);
%la cantidad ofrecida y la cantidad demandada
                end

                j=j+1;
            end
            i=i+1;
        end
    k=k+1;
    ingreso_vendedor_1(k)=ingreso_real(1);
    ingreso_vendedor_2(k)=ingreso_real(2);
    ingreso_vendedor_3(k)=ingreso_real(3);
    ingreso_promedio(k)=mean(ingreso_real);
    gasto_consumidor_1(k)=gasto_real(1);
    gasto_consumidor_2(k)=gasto_real(2);
    gasto_consumidor_3(k)=gasto_real(3);
    gasto_promedio(k)=mean(gasto_real);
    prom_precio_oo(k)=mean(precio_oo);
    prom_precio_dda(k)=mean(precio_dda);
    prom_cantidad_oo(k)=mean(cantidad_oo);
    prom_cantidad_dda(k)=mean(cantidad_dda);
    ingreso_real
    gasto_real
end

v=0:1:k-1;
plot(v,prom_precio_oo,'y',v,prom_precio_dda,'g')
hold
plot(v,prom_cantidad_oo.*10,'r',v,prom_cantidad_dda.*10,'b')
xlabel('Iteración')
legend('Precio oferta promedio','Precio demanda promedio','cantidad
oferta promedio x10','cantidad demanda promedio x10')

```

Anexo 2

```
%Programa que representa el actuar de los tres consumidores del mercado
function x=consumidores
```

```
global consumidor1
global consumidor2
global consumidor3
global gasto
global precio_dda
global cantidad_dda
global precio_oo
global cantidad_oo
global gasto_objetivo
global precio_dda_objetivo
global cantidad_dda_objetivo
global n
global e
```

```
%consistencia interna
```

```
gasto_objetivo=precio_dda.*cantidad_dda;
```

```
%consistencia interna+consistencia externa
```

```
e(2,1:3)=gasto_objetivo-gasto;
precio_dda_objetivo=0.5*(gasto_objetivo-
e(2).*n)./cantidad_dda+0.5*(precio_oo+precio_dda)/2 ;
cantidad_dda_objetivo=0.5*(gasto_objetivo-e(2).*(n-
1))./precio_dda+0.5*(cantidad_oo+cantidad_dda)/2;
```

```
consumidor1.trainParam.epochs = 10;
consumidor1 = train(consumidor1, {[gasto(1) precio_dda(1)
cantidad_dda(1)]'} , {[gasto_objetivo(1) precio_dda_objetivo(1)
cantidad_dda_objetivo(1)]'});
```

```
consumidor2.trainParam.epochs = 10;
consumidor2 = train(consumidor2, {[gasto(2) precio_dda(2)
cantidad_dda(2)]'} , {[gasto_objetivo(2) precio_dda_objetivo(2)
cantidad_dda_objetivo(2)]'});
```

```
consumidor3.trainParam.epochs = 10;
consumidor3 = train(consumidor3, {[gasto(3) precio_dda(3)
cantidad_dda(3)]'} , {[gasto_objetivo(3) precio_dda_objetivo(3)
cantidad_dda_objetivo(3)]'});
```

```
evolucion_consumidores(1:3,1)= sim(consumidor1, [gasto(1) precio_dda(1)
cantidad_dda(1)]');
evolucion_consumidores(1:3,2)= sim(consumidor2, [gasto(2) precio_dda(2)
cantidad_dda(2)]');
evolucion_consumidores(1:3,3)= sim(consumidor3, [gasto(3) precio_dda(3)
cantidad_dda(3)]');
```

```
gasto=evolucion_consumidores(1,:);          %actualización del gasto
pronosticado
```



```

precio_dda=evolucion_consumidores(2,:); %actualización del precio a
demandar
cantidad_dda=evolucion_consumidores(3,:); %actualización de la cantidad a
demandar

```

Anexo3

%Programa que representa el actuar de los tres productores del mercado

```
function x=vendedores
```

```

global vendedor1
global vendedor2
global vendedor3
global ingreso
global precio_oo
global cantidad_oo
global precio_dda
global cantidad_dda
global ingreso_objetivo
global precio_oo_objetivo
global cantidad_oo_objetivo
global n
global e

%objetivos cruzados
%consistencia interna
ingreso_objetivo=precio_oo.*cantidad_oo;

%consistencia interna+consistencia externa
e(1,1:3)=ingreso_objetivo-ingreso;
precio_oo_objetivo=0.5*(ingreso_objetivo-
e(1).*n)./cantidad_oo+0.5*(precio_oo+precio_dda)/2;
cantidad_oo_objetivo=0.5*(ingreso_objetivo-e(1).*(n-
1))./precio_oo+0.5*(cantidad_oo+cantidad_dda)/2;

vendedor1.trainParam.epochs = 10;
vendedor1 = train(vendedor1, {[ingreso(1) precio_oo(1) cantidad_oo(1)]'}
, {[ingreso_objetivo(1) precio_oo_objetivo(1) cantidad_oo_objetivo(1)]'});

vendedor2.trainParam.epochs = 10;
vendedor2 = train(vendedor2, {[ingreso(2) precio_oo(2) cantidad_oo(2)]'}
, {[ingreso_objetivo(2) precio_oo_objetivo(2) cantidad_oo_objetivo(2)]'});

vendedor3.trainParam.epochs = 10;
vendedor3 = train(vendedor3, {[ingreso(3) precio_oo(3) cantidad_oo(3)]'}
, {[ingreso_objetivo(3) precio_oo_objetivo(3) cantidad_oo_objetivo(3)]'});

evolucion_vendedores(1:3,1)= sim(vendedor1, [ingreso(1) precio_oo(1)
cantidad_oo(1)]');

```

```
evolucion_vendedores(1:3,2)= sim(vendedor2,[ingreso(2) precio_oo(2)
cantidad_oo(2)]');
evolucion_vendedores(1:3,3)= sim(vendedor3,[ingreso(3) precio_oo(3)
cantidad_oo(3)]');

ingreso=evolucion_vendedores(1,:);      %actualización del ingreso
pronosticado
precio_oo=evolucion_vendedores(2,:);    %actualización del precio a
ofrecer
cantidad_oo=evolucion_vendedores(3,:);  %actualización de la cantidad a
producir
```

Referencias

Herbert Simon, *Nobel Lectures, Economics 1969-1980*, World Scientific Publishing Co., Singapore (<http://www.nobel.se/economics/laureates/1978/simon-autobio.html>).

T. Ostrom, (1998): *Computer Simulation: the Third Symbol System*, Journal of Experimental Social Psychology, 24, 1998.

Huhns, M. and Singh, M.P. (1998): *Readings in Agents*; Morgan Kaufmann, San Mateo, CA.

Gilbert y Troitzsch, (1999): *Simulation for the Social Scientist*, Open University Press.

Wooldridge M. and Jennings N.R., (1995): *Intelligent Agents: Theory and Practice*.

Wooldridge M. and Jennings N.R. (1995): *Agent theories, architectures and languages: a survey*. In: Wooldridge, M. and Jennings, N.R. (eds) *Intelligent Agents*, Lecture Notes in AI, Vol. 890, pp. 1-39, Springer-Verlag.

Troitzsch, K. G., U. Mueller, N. Gilbert and J. E. Doran, Eds. (1996) *Social science microsimulation* Berlin: Springer.

Object-Oriented Programming and the Objective-C Language, NeXT Computer, Inc., 1993.

N. Gilbert and P. Terna, *How to build and use agent-based models in social science*, Mind&Society, 1, 2000.

Günter Haag, Ulrich Mueller, and Klaus G. Troitzsch, eds.: *Economic Evolution and Demographic Change* (<http://www.uni-koblenz.de/~kgt/EcoEvo.html>).

Rainer Hegselmann, Ulrich Mueller, and Klaus G. Troitzsch, eds.: *Modelling and Simulation in the Social Sciences from a Philosophy of Science Point of View* (<http://www.uni-koblenz.de/~kgt/Bremen.html>).

Klaus G. Troitzsch, Ulrich Mueller, Nigel Gilbert, Jim Doran, eds: *Social Science Microsimulation* (<http://www.uni-koblenz.de/~kgt/DagProc.html>).

Nigel Gilbert and Jim Doran, eds.: *Simulating Societies* (<http://www.uni-koblenz.de/~kgt/SimSoc.html>).

Nigel Gilbert and Rosaria Conte, eds.: *Artificial Societies* (<http://www.uni-koblenz.de/~kgt/ArtSoc.html>).

Rosaria Conte and Cristiano Castelfranchi, eds.: *Cognitive and social action* (<http://www.uni-koblenz.de/~kgt/ConteCastelfranchi.html>)

Dirk Helbing: *Quantitative Sociodynamics* (<http://www.uni-stuttgart.de/UNIuser/thphys/helbing/kluwer.html>).

Dirk Helbing: *Verkehrsdynamik: Traffic Dynamics* (<http://www.uni-stuttgart.de/UNIuser/thphys/helbing/springer.html>).

Andreas Flache: *The Double Edge of Networks. An Analysis of the Effect of Informal Networks on Cooperation in Social Dilemmas*, Thesis Publishers, Amsterdam 1996, ISBN 90-5170-400-3 (<http://www.uni-bayreuth.de/departments/philosophie/andreas.htm>).

Rosaria Conte, Rainer Hegselmann, Pietro Terna, eds.: *Simulating Social Phenomena*, Berlin etc. (Springer) 1997 (LNEMS 456).

Wim B G Liebrand, Andrzej Nowak, Rainer Hegselmann, eds.: *Computer Modeling of Social Processes*, London: Sage 1998 (<http://www.sagepub.co.uk/books/details/b006003.html>).

Nigel Gilbert and Klaus G. Troitzsch: *Simulation for the Social Scientist*, Open University Press 1999 (<http://www.uni-koblenz.de/~kgt/SimForSocSci.html>).