

75.68 - Sist. de Soporte para Celdas de Prod. Flexible

**Sistemas de autómatas y aprendizaje automático**

**(“SimCity”)**

79373 Pablo Cristian Viera Morales

84985 Emmanuel Espina

88240 Leandro Romero

87249 Nicolas Morales

**Jefe de Cátedra**

Hernán Merlino

InDICE

Introducción 3

Aprendizaje automático 3

Conceptos Generales 3

Experiencia práctica 5

Objetivo 5

Alcance 5

Escenario 5

Modelo del simulador 5

Implementación del simulador 6

Agentes 7

Problemas 7

Estrategias de resolución 7

Conclusiones 10

ANEXO I – Código fuente 11

# Introducción

## Aprendizaje automático

**Aprendizaje automático** o **Aprendizaje de máquinas** (Machine learning), es una rama de la Inteligencia Artificial que tiene por objetivo el desarrollo de técnicas que permiten a sistemas a sistemas de procesamiento de datos, ‘aprender’. Entendiendo este ***aprendizaje*** como el proceso a través del cual se adquieren o modifican habilidades, destrezas o conductas como resultado del análisis, experiencia, observación o instrucción.

### Conceptos Generales

* **Sistema autónomo**: Entidad capaz de decidir por sí misma las acciones que llevará a cabo para alcanzar sus metas, basándose en su conocimiento y experiencias acumuladas.
* **Autómata / Agente** : Paradigma del modelo, el cual percibe y actúa según el entorno en el que está situado y las condiciones o características del mismo. El agente basa sus decisiones en base a los principios de optimización y consistencia con el fin de lograr un objetivo o finalidad.

Los agentes se caracterizan según el tipo de proceso utilizado para la obtención de resultados válidos racionales, los principales son :

Respuesta predeterminada: ante cada tipo de cuestión presentada al sistema

Algoritmos genéticos: Basados en la evolución de las cadenas de ADN de los seres vivos.

Redes neuronales: Análoga al funcionamiento físico del cerebro de seres vivos

Razonamiento : Símil del razonamiento humano, en base a una lógica formal.

El agente se caracteriza por tener un objetivo definido o motivación que lo llevan a intentar comprender su entorno y modificar su comportamiento interactivamente como resultado de cambios en las condiciones del mismo.

* **Inferencia**: Evaluación realizada sobre expresiones bien formadas de un leguaje, que al ser relacionadas intelectualmente o mediante algoritmos de implementación del proceso deductivo, permiten trazar una línea lógica de implicación entre las diferentes expresiones.
* **Conocimiento:** Hechos o información adquirida a través de la experiencia, educación o comprensión. El conocimiento es adquirido a través de técnicas de **aprendizaje** (proceso por medio del cual se adquieren o modifican habilidades, destrezas, conocimientos, conductas o valores)

# Experiencia práctica

## Objetivo

Estudio y análisis de las técnicas de creación de programas capaces de generar comportamientos a partir de información no estructurada suministrada en forma de ejemplos o ‘situaciones iniciales’. A partir de dichas experiencias, el agente realiza un proceso de inducción del conocimiento por medio de la representación de datos y métodos de manipulación y caracterización de la información del entorno.

La experiencia se basa en la definición de agentes con preferencias particulares sobre ciertas características de las distintas zonas que componen una ciudad que habitan y el objetivo final de los mismos es encontrar la zona que mejor se ajuste a la definición de sus preferencias.

En base a lo descripto, el éxito de una zona de la ciudad se percibe por el grado de población de la misma, ya que a mayor densidad, mayor satisfacción de las necesidades de sus habitantes. Por el contrario, una zona poco poblada indica la necesidad de mejora de sus características a fin de que resulte más atractiva para los potenciales pobladores.

## Alcance

El presente trabajo se basa en el estudio de la simulación de un entorno compuesto por ciudades con características definidas bajo diferentes grupos de aspectos y una población de individuos autómatas (agentes) que recorren libremente las ciudades y deciden en qué zonas se establecerán finalmente

## Escenario

Simulación en base a diversas zonas con características comunes asociadas en un determinado grado a cada una de ellas. En dicho entorno se alojan diferentes ‘agentes’ que recorren las mismas en búsqueda de aquella que mejor satisfaga sus preferencias.

## Modelo del simulador

Las ciudades se implementan como un conjunto de **Zonas**, las cuales a su vez, poseen asociados parámetros ponderados que las caracterizan al individuo ideal que puede vivir en esa zona. Las caracteristicas consideradas para esta simulación son:

CULTURA

DIVERSION

SEGURIDAD

EDUCACION

NATURALEZA

POPULOSO

FAMILIAR

TRANQUILO

TRANSPORTE

EXCLUSIVIDAD

COSTO

De acuerdo al valor del atributo característico, la zona de la ciudad se identificará en mayor o menor medida con el agente y consecuentemente le resultará más o menos atractiva. En base a qué tan atractiva le parezca una sona el agente podrá elegirla o no.

El agente por lo tanto realiza una primera evaluación de todas las zonas de las ciudad ponderando cada una de ellas y asignandoles un puntaje. Este es un procedimiento que suele ser realizado en la vida real por las personas que construyen una tabla de desición con distintos atributos deseables de un departamento a alquilar y obtienen un puntaje final para cada uno de ellos.

Se agrega un parámetro de indesición a esta selección. Un agente completamente determinista elegiría la zona con el puntaje más alto obtenido. Para simular comportamiento emocional humano se consideran los puntajes mas altos (dentro de un rango configurable) y se elige uno aleatoriamente.

Se generaron un conjunto de personalidades de agente estereotípicas. Para obtener mayor variabilidad en la fase de entrenamiento, se utilizo un algoritmo genetico para que, a partir de esas personalidades características, se obtengan “cruzas” que combinen 2 o 3 personalidades. En este proceso se descartan las personalidades donde la desviación típica de los atributos de la personalidad es baja, a fin de evitar “personalidades chatas” que no aportarian al entrenamiento del sistema.

## Implementación del simulador

Ciudad

El simulador es escencialmente una red neuronal tipo SOM. Cada zona de la ciudad es un nodo de la red que contiene un vector de atributos identicos a la de personalidad del agente.

Esta red se inicializa con todos valores de cada nodo en 0 excepto en aquellos donde hay algun elemento de ciudad. El valor inicial de estos nodos dependerá del elemento que se encuentre ahí. La definición de estos atributos esta en el archivo HeuristicaSimple.java, contendio en el anexo.

Estos nodos pre-cargados sirven como semilla para entrenar la red.

## Agentes

Los agentes están representados por un set de preferencias ([ver figura-1](#Figura1)) ponderadas que representan el grado de importancia de cada característica para el agente.

Como ejemplo, el *cultura=1,0* representa un alto grado de afinidad del agente con la cultura. Una zona de la ciudad que tenga un valor elevado en dicho rubro, será más afín para este agente. Este valor representa la evolución en términos culturales como podrían ser museos, obras de teatro o cine, espacios culturales o artísticos, etc.

### Problemas

**Satisfacción:** Por su característica de “individuo único” un agente posee preferencias innatas, las cuales difícilmente puedan ser satisfechas en forma exacta y permanente por una determinada zona en las que podría vivir. Esto deberá resolverse por grado de afinidad

**Aprendizaje automático:** La conducta de evaluación y adaptabilidad del agente con a las distintas zonas, debe resolverse a fin de permitir que éste elija el sitio en el que desea vivir de acuerdo al mayor grado de satisfacción posible, debiendo en ocasiones, adaptar su preferencias en forma temporal.

### Estrategias de resolución

**Satisfacción:**

La implementación se realiza mediante un vector de preferencias y atributos descriptivos del agente. Trabajamos con dos vectores, uno representa los niveles nativos del agente, los cuales siempre serán una referencia y otro vector representa la zona. Diferentes métricas pueden utilizarse para comparar estos dos vectores. En este trabajo se ha optado por la solución más sencilla que es utilizar la norma Euclideana. No obstante podrían implementarse estrategias más sofiticadas y que no dependan de una operación matemática.

Al visitar una zona de la ciudad, una heurística de afinidad calcula el grado de cumplimiento de la misma y otorga un valor que representa cuán adecuada es la zona para el agente (score).

Habiendo calculado todos los scores de la ciudad, el agente elige uno entre los scores más altos.

Para la generación de habitantes en forma realista, se implemento un *algoritmos genético* que permite una generación rápida de agentes con características individuales diferentes a partir de personalidades estereotipicas.

**Aprendizaje automático:** El aprendizaje o evaluación de la zona, se realiza mediante Heurística de comparación entre los valores deseados y los propios de la zona a evaluar. Como resultado de la aplicación de la misma, se obtiene una ponderación de la zona evaluada. Esta ponderación es propia de la red SOM. El funcionamiento del algoritmo implementado puede resumirse en los siguientes pasos:

Para cada agente que se agrega:

1. Calcular la métrica de afinidad del agente con todos los nodos de la red (ie ciudad)
2. Seleccionar un el subconjunto de los puntajes mas altos (los puntajes más altos son los numericamente menores, debido a que una norma pequeña indica similitud de los vectores). Se aceptan todos los nodos que tengan un puntaje p, que es superior al puntaje minimo obtenido, pero considerablemente inferior al máximo. Este parametro se modela con el valor “indesición”. Una indesición de 0 toma solamente el nodo de mayor puntaje. Una indesición de 1 toma la totalidad de los nodos como candidatos.
3. Seleccionar aleatoriamente uno de los nodos de entre los candidatos pre-seleccionados en el paso 2, ese será el nodo elegido por el agente.
4. Afectar las características de los nodos vecinos. La forma de afectarlos es calcular la influencia que tiene este agente en ese nodo sobre los demas nodos de la red. Para ello se consideran los demas nodos que estan a una distancia r del nodo elegido (el radio de influencia). Para cada nodo se calcula un factor de influencia que depende de la proximidad al agente y de la indesición con que se eligió.

Asimismo el radio de influencia va disminuyendo con cada iteración. Se demarca de esa forma la fase de aprendizaje ya que despues de varias iteraciones los nuevos agentes ya no van a modificar la red. Tambien se considera un learningRate que es un factor que disminuye con el tiempo hasta volverse, eventualmente 0, y que hace que el factor de influencia del punto 4 sea cada vez menor.

Al introducir una gran cantidad de agentes en la red se logra entrenarla y se forman zonas o clusteres de agentes en diferentes sectores de la ciudad, correlacionados estos con los valores iniciales de la red que a su vez fueron definido por los elementos de la ciudad en esas posiciones.

Ejemplo de agentes:

  

  

  

  

## Conclusiones

Durante el desarrollo del presente trabajo, tuvimos la oportunidad de analizar la problemática del desarrollo de autómatas y la heurística que los gobierna para lograr que tomen decisiones coherentes con características predefinidas. Asimismo se pudo ver como una red neuronal SOM con un estado inicial forzado y entrenada con agentes generados automaticamente puede formar clusteres que mantienen una relación con los valores iniciales de la misma.

La red SOM representa una solucion sumamente sencilla al problema de asignar agentes a una zona optima de un escenario. No obstante es fundamental los agentes utilizados para el entrenamiento. El uso de agentes con personalidades completamente aleatorias no proporcionó resultados satisfactorios.

En cuanto a performance la red utilizada puede agregar agentes en O(n) donde n es el tamaño de la red.

Modificando los parámetros de la red por ensayo y error se logró tener en la mayoría de los casos comportamientos que se aproximan al los que el sentido común dicta, logrando asi agentes con un nivel de inteligencia primitivo pero notable por la sencillez del algoritmo.

## ANEXO I – Fragmentos de código fuente

**public** Personalidad habitanteInicial(ElementoCiudad zona) {

Personalidad personalidad = **new** Personalidad();

**double** cultura = 0;

cultura += zona.itemsCiudad.contains(ItemCiudad.*TEATRO*) ? 1 : 0;

cultura += zona.itemsCiudad.contains(ItemCiudad.*UNIVERSIDAD*) ? 1 : 0;

personalidad.put(AtributoPersonalidad.*CULTURA*, cultura);

**double** diversion = 0;

diversion += zona.itemsCiudad.contains(ItemCiudad.*TEATRO*) ? 1 : 0;

diversion += zona.itemsCiudad.contains(ItemCiudad.*BAR*) ? 1 : 0;

diversion += zona.itemsCiudad.contains(ItemCiudad.*PARQUE*) ? 1 : 0;

personalidad.put(AtributoPersonalidad.*DIVERSION*, diversion);

**double** seguridad = 0;

seguridad += zona.itemsCiudad.contains(ItemCiudad.*COMISARIA*) ? 1 : 0;

seguridad += zona.itemsCiudad.contains(ItemCiudad.*BOMBEROS*) ? 1 : 0;

seguridad += zona.itemsCiudad.contains(ItemCiudad.*HOSPITAL*) ? 1 : 0;

seguridad += zona.itemsCiudad.contains(ItemCiudad.*PELIGRO*) ? -1 : 0;

personalidad.put(AtributoPersonalidad.*SEGURIDAD*, seguridad);

**double** educacion = 0;

educacion += zona.itemsCiudad.contains(ItemCiudad.*UNIVERSIDAD*) ? 1 : 0;

personalidad.put(AtributoPersonalidad.*EDUCACION*, educacion);

**double** naturaleza = 0;

naturaleza += zona.itemsCiudad.contains(ItemCiudad.*PARQUE*) ? 1 : 0;

personalidad.put(AtributoPersonalidad.*NATURALEZA*, naturaleza);

**double** familiar = 0;

familiar += zona.itemsCiudad.contains(ItemCiudad.*PARQUE*) ? 1 : 0;

familiar += zona.itemsCiudad.contains(ItemCiudad.*UNIVERSIDAD*) ? 1 : 0;

familiar += zona.itemsCiudad.contains(ItemCiudad.*BAR*) ? -1 : 0;

familiar += zona.itemsCiudad.contains(ItemCiudad.*HOSPITAL*) ? 1 : 0;

familiar += zona.itemsCiudad.contains(ItemCiudad.*SUCIO*) ? 1 : 0;

familiar += zona.itemsCiudad.contains(ItemCiudad.*IGLESIA*) ? 1 : 0;

personalidad.put(AtributoPersonalidad.*FAMILIAR*, familiar);

**double** tranquilo = 0;

tranquilo += zona.itemsCiudad.contains(ItemCiudad.*BAR*) ? -1 : 0;

tranquilo += zona.itemsCiudad.contains(ItemCiudad.*EXCLUSIVA*) ? 1 : 0;

tranquilo += zona.itemsCiudad.contains(ItemCiudad.*PARQUE*) ? 1 : 0;

tranquilo += zona.itemsCiudad.contains(ItemCiudad.*TRANSITO*) ? -1 : 0;

personalidad.put(AtributoPersonalidad.*TRANQUILO*, tranquilo);

**double** transporte = 0;

transporte += zona.itemsCiudad.contains(ItemCiudad.*COLECTIVO*) ? 1 : 0;

transporte += zona.itemsCiudad.contains(ItemCiudad.*SUBTE*) ? 1 : 0;

transporte += zona.itemsCiudad.contains(ItemCiudad.*TRANSITO*) ? -1 : 0;

personalidad.put(AtributoPersonalidad.*TRANSPORTE*, transporte);

**double** barrioExclusivo = 0;

barrioExclusivo += zona.itemsCiudad.contains(ItemCiudad.*EXCLUSIVA*) ? 1 : 0;

barrioExclusivo += zona.itemsCiudad.contains(ItemCiudad.*CARA*) ? 1 : 0;

personalidad.put(AtributoPersonalidad.*EXCLUSIVIDAD*, barrioExclusivo);

**double** costo = 0;

costo += zona.itemsCiudad.contains(ItemCiudad.*CARA*) ? -1 : 0;

personalidad.put(AtributoPersonalidad.*COSTO*, costo);

**return** personalidad;

}