Prólogo

Aprendizaje
[Machine Learning]

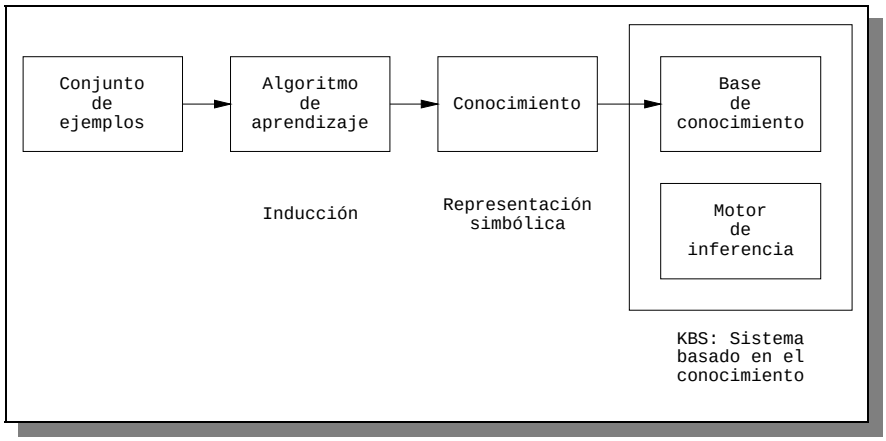
Hasta hace no demasiado tiempo se utilizaba el término “procesamiento de datos” para describir la utilización de los ordenadores en distintos ámbitos. Hoy se utiliza otro término, *IT* [*Information Technology*] que se refiere a lo mismo pero implica un cambio de enfoque. Se hace énfasis no únicamente en el procesamiento de grandes cantidades de datos, sino en la extracción de información significativa de esos datos.

Los datos son información cruda, colecciones de hechos que deben ser procesados para que sean significativos. La información se obtiene asociando hechos (en un contexto determinado). El conocimiento utiliza la información obtenida en un contexto concreto y la asocia con más información obtenida en un contexto diferente. Finalmente, la sabiduría (término que nadie utiliza en IA) aparece cuando se obtienen principios generales de fragmentos de conocimiento.

Hasta ahora, la mayor parte del software ha sido desarrollada para procesar datos, información a lo sumo. En el futuro se trabajará con sistemas que procesen conocimiento. La clave reside en asociar elementos información provenientes de distintas fuentes y sin conexión obvia de tal forma que la combinación nos proporcione beneficios. Este es uno de los desafíos más importantes de la actualidad: la construcción de sistemas que extraigan conocimiento de los datos de forma que sea práctico y beneficioso.

En Inteligencia Artificial, el aprendizaje se entiende como un proceso por el cual un ordenador acrecienta su conocimiento y mejora su habilidad. En él se resaltan dos aspectos complementarios: el refinamiento de la habilidad y la adquisición de conocimiento.

Tal como lo definió Simon, el aprendizaje denota cambios en el sistema que son adaptativos en el sentido de que permiten al sistema hacer la misma tarea o tareas a partir de la misma posición más eficiente y/o efectivamente la siguiente vez.



El proceso de adquisición automática de conocimiento

Muchas de las técnicas de aprendizaje usadas en IA están basadas en el aprendizaje realizado por los seres vivos. Para ellos la experiencia es muy importante, ya que les permite no volver a cometer los mismos errores una y otra vez. Además, la capacidad de adaptarse a nuevas situaciones y resolver nuevos problemas es una característica fundamental de los seres inteligentes. Por lo tanto, podemos aducir varias razones de peso para estudiar el aprendizaje: en primer lugar, como método de comprensión del proceso de aprendizaje (desde el punto de vista de la Psicología) y, en segundo término, aunque no por ello sea menos interesante, para conseguir programas que aprendan (desde una perspectiva más propia de la Inteligencia Artificial).

Clasificación de las técnicas de aprendizaje

Una primera clasificación de las técnicas de aprendizaje existentes se puede realizar atendiendo a la filosofía seguida en el proceso de adquisición del conocimiento

- En el **aprendizaje supervisado** (o aprendizaje a partir de ejemplos, con profesor), los ejemplos de entrada van acompañados de una clase o salida correcta.

- En el **aprendizaje no supervisado** (aprendizaje por observación, sin profesor) se construyen descripciones, hipótesis o teorías a partir de un conjunto de hechos u observaciones sin que exista una clasificación a priori de los ejemplos. El ejemplo más significativo de este tipo es el de los métodos de agrupamiento o *clustering*.

La familia más importante de técnicas de aprendizaje es, sin duda, la formada por algoritmos de aprendizaje supervisado. Dentro de ella se engloban:

- Técnicas de ***aprendizaje memorístico*** [*rote learning*], usadas ya por Samuel en 1963 en un programa que jugaba a las damas.

- Modelos de ***aprendizaje por ajuste de parámetros***, consistentes en adaptar un procedimiento de evaluación que combina información de varias fuentes en un único estadístico global. Este tipo de modelos es muy utilizo en juegos (p.ej. ajedrez).

- Algoritmos de ***aprendizaje inductivo***, mediante los cuales se infieren leyes a partir de ejemplos particulares. Ejemplos de este grupo los encontramos en la construcción de árboles de decisión (ID3, CART, SLIQ...), en algoritmos basados en la metodología STAR de Michalski (como INDUCE o AQ) y también en los espacios de versiones de Mitchell.

Las fases del aprendizaje: entrenamiento y prueba

En todo proceso de aprendizaje se han de distinguir dos etapas: entrenamiento y prueba. Durante el entrenamiento se han de extraer las conclusiones apropiadas a partir de un conjunto de casos de entrenamiento y se debe obtener un modelo capaz de mostrar lo aprendido. Posteriormente, durante la fase de prueba se tiene que estudiar la calidad del modelo obtenido en la fase de entrenamiento usando un conjunto de datos de prueba (distintos a los usados en el entrenamiento).

En la etapa de entrenamiento, durante la construcción de un modelo adecuado a los ejemplos conocidos, se han de perseguir los siguientes objetivos:

✯ *Precisión* (proporción de ejemplos aprendidos correctamente lo mayor posible).

✯ *Velocidad* (tiempo de aprendizaje razonable).

✯ *Comprensibilidad* (el experto humano debe ser capaz de entender el porqué de las cosas).

✯ *Funcionamiento en tiempo real* (adaptabilidad del sistema ante nuevas situaciones).

En la fase de prueba, se tiene que realizar una estimación del error cometido. Para ello se suele utilizar una matriz (denominada matriz de confusión) en la que se muestran los errores que se han cometido al clasificar los casos de prueba para cada categoría o clase. Hay que tener en cuenta que en algunas aplicaciones un falso positivo no es igual de importante que un falso negativo (vg: Medicina). En esos casos se suele utilizar una matriz de costos en la que se fija la penalización por clasificar un objeto de la clase *i* como de la clase *j*. Las técnicas más comunes para realizar la estimación del error en la fase de prueba son:

- Un conjunto de N **ejemplos de prueba**, a partir de los cuales se contabilizan los errores E cometidos por el modelo aprendido durante la etapa de entrenamiento. El estimador del error es simplemente E/N. Dicho estimador es bueno si el conjunto de ejemplos usados para la prueba es independiente del conjunto de casos de entrenamiento. En la práctica, el conjunto de prueba se suele crear seleccionando aleatoriamente una parte del conjunto disponible de ejemplos (del 20% al 30%).

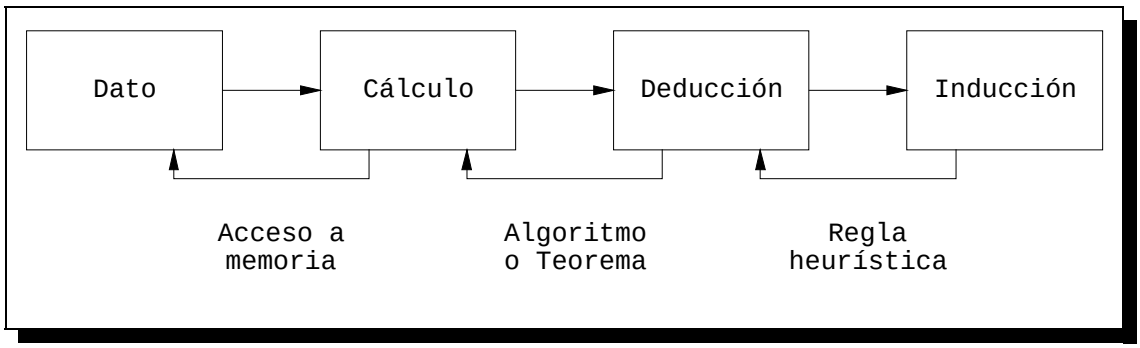
- Si se emplea el **error de resubstitución**, el conjunto de ejemplos usados en la prueba es el mismo que se usó en la fase de entrenamiento, lo que tiende a subestimar el error real.

- La **validación cruzada** [*CV: Cross Validation*] garantiza una buena estimación del error. El conjunto total de ejemplos se particiona en un número determinado de subconjuntos. Cada uno de ellos es utilizado en la evaluación de un modelo construido por los restantes subconjuntos. Cuando se usan N subconjuntos se denomina *N-fold cross validation*. Un caso particular de esta técnica es dejar un caso fuera (*leave one out*): $|S|$*-fold cross validation*, donde S es el conjunto de ejemplos disponibles.

Un ejemplo sencillo: El aprendizaje memorístico

En el denominado aprendizaje memorístico [rote learning], se almacena el conocimiento obtenido para utilizarlo cuando vuelva a ser necesario (tal como se hace en Programación Dinámica). Los dos problema principales a los que hay que enfrentarse serán qué almacenar y cómo almacenarlo para poder recuperarlo eficientemente (para lo cual se puede hacer uso de técnicas de indexación).

De hecho, todos los sistemas de aprendizaje se construyen sobre un proceso de aprendizaje memorístico que almacena, mantiene y recupera conocimiento en una base de conocimiento.



Reducción de datos (Lenat, 1979)

El aprendizaje memorístico no es útil en entornos que cambian constantemente (sistemas no monótonos) porque el aprendizaje memorístico presupone que la información almacenada seguirá siendo válida más tarde.

Una posibilidad consiste en almacenar toda la información obtenida y eliminar, cuando se agote la memoria, la información que lleve más tiempo sin utilizarse (siguiendo una estrategia LRU [*Least Recently Used*] como en las memorias caché).

Otra alternativa es, obviamente, evaluar mediante algún tipo de heurística si merece la pena almacenar la información obtenida o recalcularla cuando se vuelva a necesitar (realizando un análisis de costos y beneficios).

Bibliografía

Hubert L. Dreyfus
"What Computers Still Can't Do. A Critique of Artificial Reason"
The MIT Press, 1994 [Fourth printing]

Un libro muy recomendable lleno de críticas a la Inteligencia Artificial (y a los que investigan en el tema).

Yves Kodratoff
"Introduction to Machine Learning"
Pitman Publishing, 1988

Un libro extraño sobre ML [*Machine Learning*]. Es destacable su tercer apéndice, titulado "*ML in context*", en el que se exponen algunas ideas bastante discutibles y se realiza una curiosa analogía entre la educación y ML.

Roger S. Pressman
"Ingeniería del Software. Un enfoque práctico"
McGraw-Hill, 1993 [3ª edición]

La biblia de la Ingeniería del Software, en la cual aparece el razonamiento con el que se abre este capítulo.

Elaine Rich & Kevin Knight
"Inteligencia Artificial" [2ª edición]
McGraw-Hill, 1994

Uno de los libros clásicos de Inteligencia Artificial. Incluye un capítulo dedicado íntegramente a temas de *Machine Learning*.

Sabrina Sestito & Tharam S. Dillon
"Automated Knowledge Acquisition"
Australia: Prentice Hall, 1994

Un libro sobre adquisición automática de conocimiento (es decir, ML) en el que se tratan, entre otros modelos de aprendizaje, los árboles de decisión (CLS, ID3 y CART), la generación progresiva de reglas (STAR) y las redes neuronales. También se citan algunos sistemas curiosos basados en marcos (como AM o EURISKO, ambos de Lenat).

Larry Wos
"Programs that offer fast, flawless, logical reasoning"
Communications of the ACM, June 1998

Un artículo dedicado a un programa de razonamiento automático [*automated reasoning*] denominado OTTER [*Organized Theorem-proving Techniques for Effective Research*, "nutria" en castellano], un interesante programa escrito en C cuyo código está disponible en Internet.