Learning and Expert Systems

Unit 5

Brandon Syiem

Learning

- Most of the AI programs/solutions we have seen perform task “intelligently” but do not enhance or “learn” any new problem solving techniques/methods.
- As such, some people argue whether the AI solutions discussed so far can truly be considered “intelligent”.
- There are many ways in which an AI program can learn, but to understand this we must first understand what is “Learning”.
- One way to define learning, as proposed by Simon[1983]

> ...changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time.

Cont ..

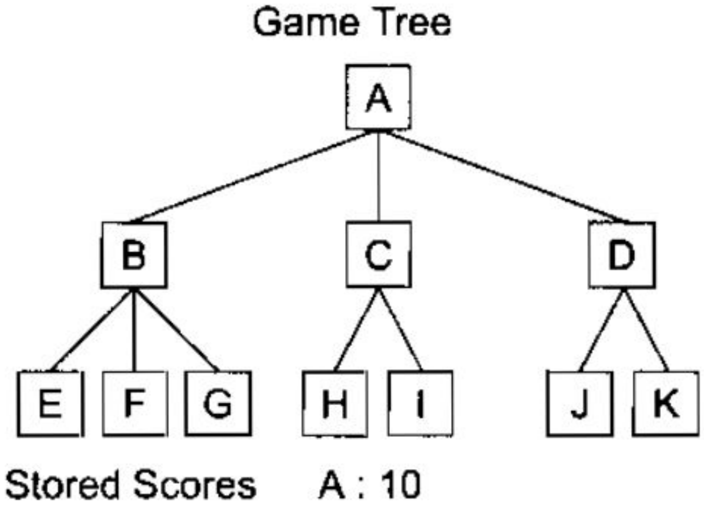
- Given the above definition, Learning can cover a wide range of phenomenon.
  - Skill refinement
  - Knowledge acquisition
- AI solutions are heavily dependent on knowledge about the problem world and as such “Knowledge acquisition” will be the focus of our discussions.
- Knowledge acquisition can include different activities.
  - Rote Learning
  - Learning by taking advice
  - Learning from problem-solving experience
  - Learning from examples

Rote Learning

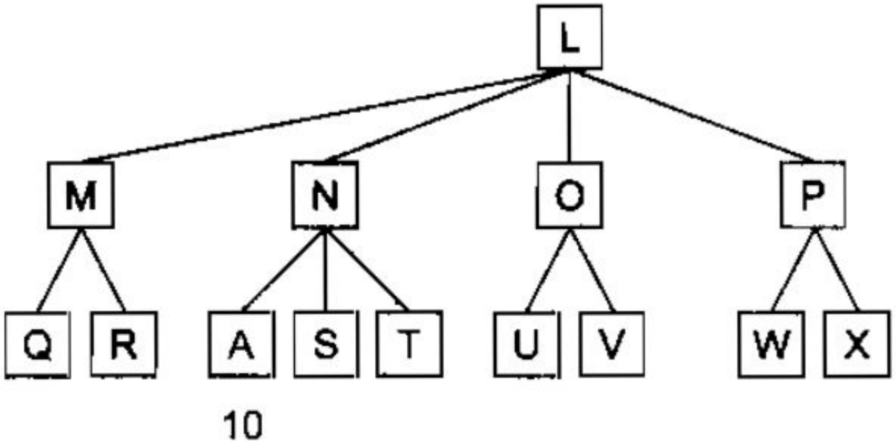
- Is the simplest form of learning
- Storing of data/information is a form of learning
- Data caching is a form of storage that is used to store computed values so that the value does not need to be recomputed
- Caching that is performed by AI programs to increase performance is called ***rote learning.***
- Let's look at the example of the first checkers AI by Samuel [1963]

Example

- Samuel's checkers AI used rote learning along with the Minimax algorithm.
- Let's say fig (a) is from the first game and fig (b) is from another game in the future



(a)



(b)

Cont ..

- In the first game, we have calculated the value at node A to be 10 (along with a best path which is not shown in the figure).
- This value (10) along with the best path and board configuration / State (A) is stored.
- When this board configuration is encountered again in some future game, the program does not need to reapply the minimax algorithm when it reaches state A and can instead just look up the store value, best path based on the state.

Rote learning cont ..

- Two important capabilities
  - **Organized Storage of Information** - For the above solution of using a stored value to be more efficient than recomputing it, it is necessary to have a quick way of accessing the stored information quickly. (indexing, etc.)
  - **Generalization** - The number of objects/values that need to be stored may be very large and as such some generalization is required.

Learning by taking advice

- One of the most common things that come to mind when “learning” is mentioned is a teacher and a student.
- A computer by itself cannot do anything until a programmer writes a program instructing the computer what to do.
- This is essentially a student - teacher relationship where the computer is the student, the teacher is the programmer and the program is the advice.
- However, the program may need a compiler/interpreter to actually be of any use.

Example

- Say a line of advice is given to an AI program that plays chess.
- The advice goes:

IF POSSIBLE, TAKE THE OPPONENT'S QUEEN FIRST

- This means little to the AI program if it cannot turn it into a series of moves or logic.
- For instance, the computer can adjust its static evaluation function to favour moves that lead to board configurations where the opponent's queen is taken.
- The process of converting "advice" into a form that is useful to the program is called ***operationalization***

Learning by Induction

- One of the problems in AI is **classification**.
- Classification relates to assigning a specific **predefined class** to a data point.
- It is a form of **Supervised Learning.**

Height (feet)	**Weight (kg)**	**Age (years)**	**Gender**
5’4	65	26	Male
6’2	84	29	Male
5’4	63	27	Female

Example 1

Cont ..

Classes can be defined in two ways:

1. Isolate a set of features relevant to the task. Define each class by a weighted sum of these features (classes can then be defined using a scoring function).

$$w1*f1 + w2*f2 + w3*f3 \ldots\ldots + wn*fn$$

- The **f** in the given equation is the value of the **feature** and the **w** is the **weight** attached to it.
- **w** can be negative to signify negative evidence

Cont ..

2. Isolate a set of features relevant to the task. Define each class as a structure composed of these features.
    a. say if we wanted to classify animals, we can store the body of each animal can be stored as a structure

Body

Colour	Feathers	Quadrupedal/Bipedal

- The first scheme, using a statistical approach is more efficient.
    - Examples for classifiers that use the first scheme: Neural Networks, Regression models
- The second scheme, using a structure is more flexible.
    - Examples for classifiers that use the first scheme: KNN

Cont ..

- How to select attributes/features?
- Or, How do you define a class?
- This problem of constructing a class is called **concept learning or Induction.**
- If classes are defined using a scoring function
  - We can use something called **coefficient adjustment** for concept learning
- Otherwise, there are a number of ways in which coefficient adjustment can be achieved given that the classes are defined using a structure.
  - Winston’s Learning Program (pg. 356)
  - Version Spaces (pg. 358)
  - Decision Trees (pg. 364)

Explanation Based learning

Expert Systems

Expert systems

- Solve problems usually solved by human experts.
    - Medical Diagnosis
    - Proving mathematical theorems
- Requires large amounts of domain knowledge
- Uses more than one reasoning mechanism
- But how do these systems work? How do they represent and use the knowledge the have?

Representing and Using Domain Knowledge

- Most widely used way of representing domain knowledge is by a **set of production rules.**
- Requires a **frame system** that defines the objects that occur in the rules.
- Different systems use slightly different ways of representing production rules based on the specifics of the domain

Examples

- All Examples are re-written in english as opposed to the actual rules in the systems (which may be written in some specific language like LISP.)
- **MYCIN (Domain - Diagnosis and Treatment)**

```
If:     (1) the stain of the organism is gram-positive, and
        (2) the morphology of the organism is coccus, and
        (3) the growth conformation of the organism is clumps,
            then there is suggestive evidence (0.7) that
            the identity of the organism is staphylococcus.
```

- MYCIN
    - Is used for diagnosis and as such cannot be absolutely certain of results and therefore uses **certainty factors**

Examples Cont ..

- **R1 (Domain - Configure DEC VAX systems)**

```
If: the most current active context is distributing
        massbus devices, and
    there is a single-port disk drive that has not been
        assigned to a massbus, and
    there are no unassigned dual-port disk drives, and
        the number of devices that each massbus should
        support is known, and
    there is a massbus that has been assigned at least
        one disk drive and that should support additional
        disk drives,
    and the type of cable needed to connect the disk drive
        to the previous device on the massbus is known
then: assign the disk drive to the massbus.
```

- R1
    - Deals with configuring systems and as such is always sure of what to do at a certain point of time

More Examples in the book. Pg - 423. (PROSPECTOR, DESIGN ADVISOR)

Reasoning with Knowledge

- The mentioned systems may use
  - Forward Chaining
  - Backward Chaining
  - Or even a combination of the two
- Example:
  - MYCIN uses **backward chaining to find which organism** is present and **forward chaining to suggest treatment.**

Expert System Shells

- Initially, all expert systems were written from scratch
  - Usually in LISP.
- However, since most expert systems consist of similar components, namely
  - Production rules
  - Interpreter for the rules
- It was a logical step to separate the rules from the interpreter to allow for more flexibility.
- The Interpreters are called **shells.**

Cont ..

- Shells provide the following support to expert systems:
  - Rules
  - Frame system
  - Reasoning mechanisms
- They also provide support for knowledge acquisition.
- Another important responsibility for shells is to make it easy to integrate the expert system with other programs
  - UI
  - Programs that acquire data from other sources (such as databases).

Explanation

- Expert systems should aim to be easy to interact with.
- There are two additional requirements of expert systems:
    - **Explain their reasoning:**
        - People do not accept results if they do not understand how to result was achieved
        - The reasoning process should proceed in understandable steps.
    - **Knowledge Acquisition and modification of new Knowledge:**
        - Interacting with human experts.
        - Derive knowledge from raw data.
- First system to support both explanation and knowledge acquisition was TEIRESIAS (pg. 425), which was a front-end for MYCIN.

Is the patient's illness with ORGANISM-1 a hospital-acquired infection?
** WHY
[i.e., WHY is it important to determine whether or not the infection with ORGANISM-1 was acquired while the patient was hospitalized?]
[2.0] This will aid in determining the category of ORGANISM-1. It has already been established that
[2.1] the gram stain of ORGANISM-1 is gramneg, and
[2.2] the morphology of ORGANISM-1 is rod, and
[2.3] the aerobicity of ORGANISM-1 is facultative
Therefore, if
[2.4] the infection with ORGANISM-1 was not acquired while the patient was hospitalized, then
there is strongly suggestive evidence (.8) that the category of ORGANISM-1 is enterobacteriaceae
[ALSO: there is weakly suggestive evidence (.1) that the identity of ORGANISM-1 is pseudomonas-aeruginosa]
[RULE037]

*TEIRESIAS interacts with a doctor to continue its reasoning.

** HOW [2.3]
[i.e., HOW was it established that the aerobicity of ORGANISM-1 is facultative?]
The following were used:

[3.1] RULE027 indicated there is strongly suggestive evidence (.8) that the aerobicity of ORGANISM-1 is facultative.

[3.2] RULE033 indicated there is weakly suggestive evidence (.2) that the aerobicity of ORGANISM-1 is not facultative.

Since this gave a cumulative CF of (.6) for facultative, the evidence indicates that the aerobicity of ORGANISM-1 is facultative.

Knowledge Acquisition

- How are expert systems built in the first place??
    - Knowledge experts interview a domain expert to extract knowledge
    - Acquired knowledge about the domain is then translated into rules.
    - These rules along with an interpreter form the initial system and is then refined until it exhibits expert-level performance.
- The above process is time consuming and thus a more **automatic** approach is desirable.
- Note that no process till date (atleast to my knowledge) have fully automated the above process.

Cont ..

- Semi-automatic ways to create knowledge bases are used.
- Some expert systems allow interaction with human experts in the relevant domain to support the following:
  - Entering Knowledge
  - Maintaining Knowledge base consistency [example: conflicts]
  - Ensuring knowledge base completeness [example: gaps in knowledge base]
- Example
  - MOLE (pg. 427)