Inductive Programming: Tutorial 3
One-shot induction and Bias reformulation

Stephen Muggleton

The aim of this tutorial is to help you understand concepts in Lecture 3, involving One-shot induction and Bias reformulation.

Question 1

Using a table describe three differences between the properties of Human Learning and Statistical Machine Learning.

Solution

Characteristic	**Human**	**Statistical ML**
Examples per concept	Few (≈ 1)	Many ($\geq 10K$)
Concepts	Many ($\geq 10K$)	Few (≈ 1)
Background knowledge	Large	Small

Question 2

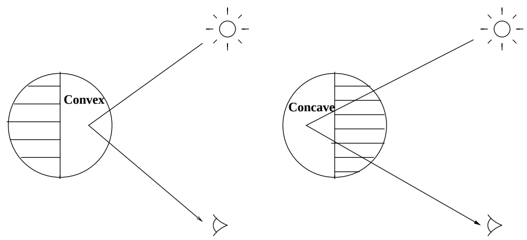
This question compares human visual perception bias with Logic-Based Inductive Programming in the identification of areas of concavity and convexity in images.

1. Give an example of human bias in interpreting concave and convex areas of an image.
2. Draw a line diagram to show how shadows in images can be used to identify convex and concave areas in an image.
3. Provide a piece of recursive Prolog background knowledge which indicates the path of a reflecting light ray.
4. What effect does the use of relevant background knowledge have on the number of images required for IP versus Statistical Machine Learning to identify objects accurately?
5. Name an aspect of an IP system which is critical for dealing with images.

Solution

1. Images of Craters and Mountains have been repeatedly confused by NASA scientists analysing the surface of Mars. By inverting an image of a crater it can appear to humans to be a mountain. This visual illusion is based on the human visual system's innate assumption that the light source is above the scene.

2. The line diagram is as follows.



3. The recursive Prolog background knowledge is as follows.

$$light(X, X).$$
$$light(X, Y) :- reflect(X, Z), light(Z, Y).$$

4. Because of the use of a strong bias IP requires far fewer images to induce a recognition model than Statistical Machine Learning.

5. IP systems require noise-tolerant mechanisms to deal with images.

Question 3

Imagine the following textual analogy question was set in an IQ test.

bob	BOB
alice	?

Explain why, among the answers below, humans would be expected to mark Answer 1 as wrong and Answer 2 as right.

Answer 1

bob	BOB
alice	BOICE

Answer 2

bob	BOB
alice	ALICE

Solution Both could be considered as correct since there exist feasible programs for each answer. However, people tend to strongly prefer the second answer since it seems much simpler given human textual bias.

Question 4

Imagine a multi-task Metagol problem involving a set of k predicates $\langle p_1, .., p_k \rangle$.

1. Explain circumstances in which it might be faster for Metagol to induce both p_i and p_j where $i \neq j$ than it is to learn p_j alone.

2. Explain the difference between *Dependent* and *Independent* multi-task IP.

3. Explain the circumstances in which you would expect a significant difference in the running time and accuracy of Dependent and Independent IP.

4. In the circumstances you described in the last part of the question, what differences would you expect in running time and accuracy of Dependent and Independent IP?

5. What differences would you expect in the programs induced from Dependent and Independent IP?

Solution

1. This can happen if inducing p_i generates sub-predicates which allow p_j to have fewer clauses than it would otherwise.

2. Independent multi-task IP predicates are not allowed to re-use subpredicates introduced by other tasks. By contrast, dependent multi-task IP allows sub-predicates introduced for one task to be re-used in subsequent tasks.

3. You would expect a significant difference in the running time and accuracy of Dependent and Independent IP in the case that tasks are sufficiently related to allow re-use of sub-tasks.

4. Assuming tasks are sufficiently related to allow re-use of sub-tasks, dependent learning should a) reduce running time and b) increase accuracy. Effect a) happens because the hypothesis space for Metagol increases exponentially with n the number of clauses. Effect b) occurs because Error increases linearly with n.

5. Again assuming tasks are sufficiently related to allow re-use of sub-predicates Dependent IP will lead to a more compact program with more dependencies between predicates in the call graph.