ARTIFICIAL INTELLIGENCE

LECTURE 10 (draft)

Ph. D. Lect. Horia Popa Andreescu

2018-2019 3rd year, semester 5

INTRODUCTION

- The content of this lecture is based on chapter 18 from [RN].

Learning

- Based on the type of feedback available, learning is:
    - Supervised
    - Unsupervised
    - Reinforcement learning
- Supervised learning involves learning a function from examples of it's inputs and outputs. E.g. interpolation
- Unsupervised learning involves learning patterns in the input when no specific output values are supplied.
- Reinforcement learning in which the learning agent learns from reinforcements (rewards)

Knowledge representation (KR)

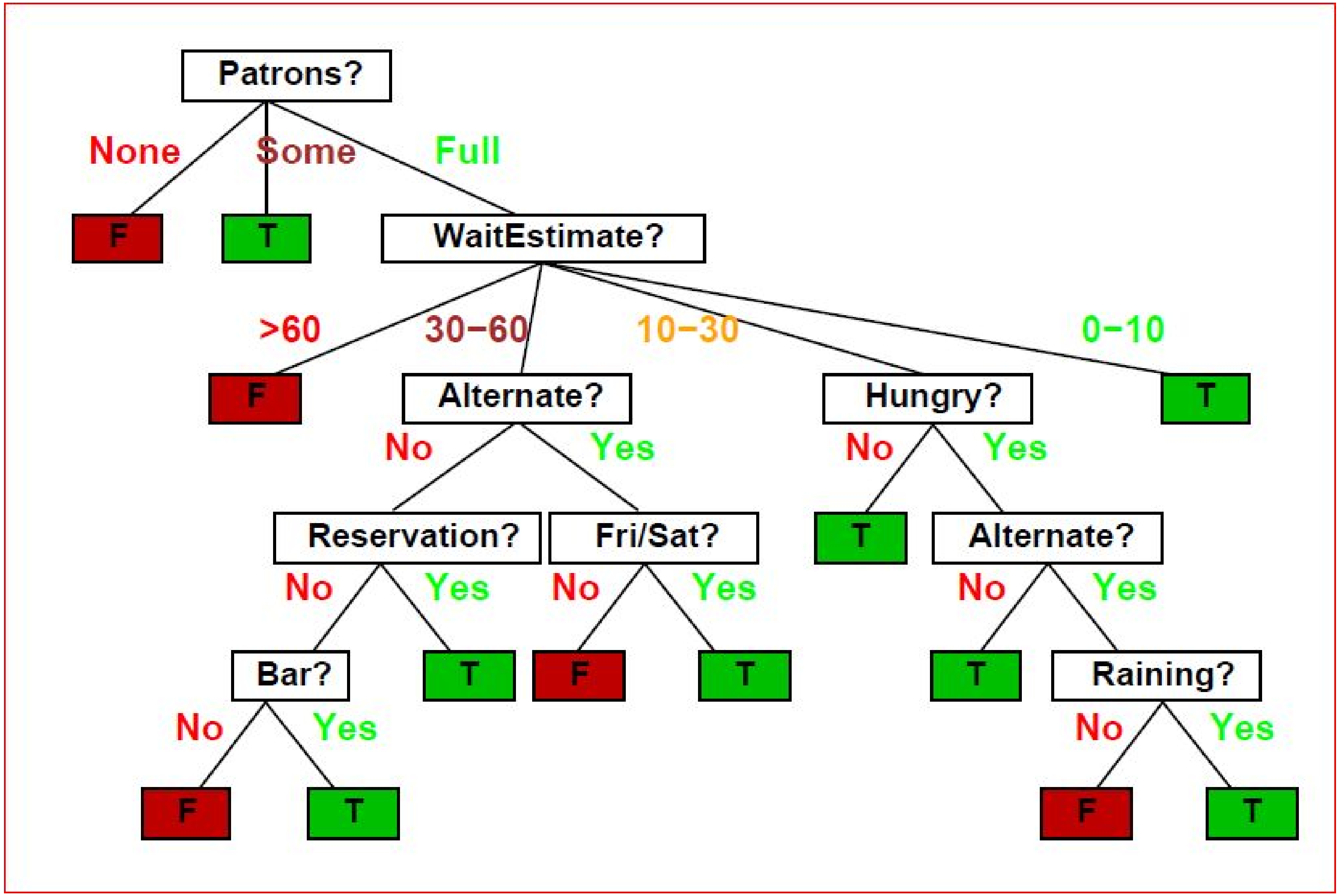
- KR plays an important role in determining how the learning algorithm works[RN651]:
  - linear weighted polynomials for utility functions in game-playing programs;
  - propositional and first-order logical sentences for all of the cornponents in a logical agent;
  - and probabilistic descriptions such as Bayesian networks for the inferential components of a decision-theoretic agent.
- Availability of prior knowledge:
  - No prior knowledge and the agent learns from his own experience
  - Existence of background knowledge

- Inductive learning:
  - given a collection of examples *f*, return a function *h* that approximates *f*
  - *h* is called a hypothesis. A good hypothesis will predict unseen examples correctly.
  - *H* is called a **hypothesis space** if it contains all the hypotheses we will consider. E.g. the set of polynomials of degree at most *k*
  - A **consistent hypothesis** is one that agrees with all the data
  - We say that a learning problem is **realizable** if the hypothesis space contains the true function; otherwise, it is **unrealizable**. E.g. if the correct function is a sinusoidal, then the polynomials cannot represent it correctly
  - Usually we cannot tell if a problem is realizable because we do not know the function. A solution is to use prior knowledge to derive a hypothesis space in which we know the true function must lie

LEARNING DECISION TREES

- A decision tree takes as input **objects** described by a **set of attributes** and gives as an output a **decision,** the predicted output value for the input.
- Attributes and output values can be:
  - discrete
  - continuous
- The learning of:
  - a discrete-valued function is called **classification learning**
  - a continuous function is called **regression**.
- A decision reaches a conclusion by performing a set of tests.
- Each internal node in the tree corresponds to a test of the value of one of the properties, and the branches from the node are labeled with the possible values of the test.

DECISION TREE [R18-14]



- More commands for debugging [GR]

EXAMPLE EXPERT SYSTEM

- https://web.archive.org/web/20130130102710/http://www.cs.trinity.edu/~yzhang/teaching/spring2011/CSCI3344/projects/instruction/cat.clp
- The original location of the file is: http://www.cs.trinity.edu/~yzhang/teaching/spring2011/CSCI3344/projects/instruction/cat.clp

BIBLIOGRAPHY

- [GR] Giarratano J., Riley G. – Expert Systems Principles and Programming, 3rd edition, PWS Publishing, 2002
- [CLIPS] CLIPS online documentation (visited nov. 2012) http://clipsrules.sourceforge.net/OnlineDocs.html
- [6] Barnum P.- Basic JESS Tutorial (visited nov. 2012) http://www.cs.rochester.edu/~brown/242/docs/JessTutorial.html
- [Ra] Rahimi S. -- Advanced Expert Systems, (online lecture, visited december 2012) http://www2.cs.siu.edu/~rahimi/cs537/slides/chapter02.pdf
- [RN] Russel S., Norvig P. – Artificial Intelligence – A Modern Approach, 2nd ed. Prentice Hall, 2003 (1112 pages)
- [R] Stuart Russel – Course slides (visited oct. 2012 at http://aima.cs.berkeley.edu/instructors.html#homework)
- [W1] Mark Watson – Practical Artificial Intelligence Programming With Java AI 3rd ed., 2008