Artificial Intelligence
A Modern Approach
(Reading Notes)

Shale Craig
shalecraig@gmail.com

May 18, 2013

Contents

Part I

Artificial Intelligence

Chapter 1

Introduction

We've tried to understand how we think (perceive, understand, predict & manipulate). AI goes further and also tries to build these intelligent entities.

1.1 What is AI?

AI can be split into four sought after qualities in computers:

- Thinking Humanly [Subsection 1.1.2]
- Thinking Rationally [Subsection 1.1.3]
- Acting Humanly [Subsection 1.1.1]
- Acting Rationally [Subsection 1.1.4]

1.1.1 Acting Humanly (The Turing Test Approach)

The Turing Test provides an operational definition of intelligence. A given program passes if a human interrogator cannot tell if the responses are from a human or computer. For now, a computer that passes needs the following capabilities:

1. NLP - to communicate in English.

2. Knowledge Representation - to store what it knows and hears.

3. Automated Reasoning - reason on stored information to new conclusions.

4. Machine learning - to detect extrapolate and react to patterns.

Turing avoids physical interaction because it is unrelated to intelligence. A similar Total Turing Test uses a video input as well. Passing the Total Turing Test requires these extra skills:

5. Computer Vision - to perceive objects.

6. Robotics - to manipulate objects.

These six disciplines compose most of AI.

1.1.2 Thinking Humanly (The Cognitive Modeling Approach)

To write a program that thinks like a human, we need to understand human minds. We can understand through introspection, psychological experiments, and brain imaging. Once we have a theory, we can implement it as a computer program. Cognitive Science brings together Computer Models (AI) and experimental psychology to understand how the human mind operates. It is important to note that an algorithm behaving accurately doesn't imply it uses a good model.

1.1.3 Thinking Rationally (The "Laws of Thought" Approach)

Generally, rational thought is synonymous with logical thought. We can create programs to compute logical correctness, but these programs have bad problems (decidability being one of them). It's hard to turn informal knowledge and state it in logical notation. There is a difference between solving problems "in principle" and in practice.

1.1.4 Acting Rationally (The Rational Agent Approach)

<u>Computer agents</u> are expected to autonomously perceive their environment, persist over time, create and set goals. <u>Rational agents</u> act to achieve the best outcome. In the

“laws of thought” approach, emphasis is on inference, but that’s only part of being a rational agent. Acting rationally doesn’t always involve inference (i.e. reflexively recoiling from a hot stove), and sometimes acting rationally occurs because something just “must be done”. Learning is not just for knowledge, but it also allows us to more effectively behave rationally. This approach has advantages over the other approaches. It is more general than the “laws of thought” approach because it’s more than just correct inference. Scientific development helps this approach more than the other developments. Sometimes we can’t always react 100% rationally, as that is too computationally expensive. We assume that perfection is a good starting point for our analysis.

1.2 Foundations Of Artificial Intelligence

This is a brief history of ideas that contribute to AI. It glosses over many parts.

1.2.1 Philosophy

Reasoning is like numerical computation, we can use formal rules to form conclusions.

If the mind is governed by physical laws, then it has no more free will than a rock “deciding” to fall toward the center of the earth. Free will is the perception of choices appears to the choosing entity. Nothing can be trusted from senses, because they can be lies. General rules are acquired by exposure to repeated associations between their elements. Logical positivism states that all knowledge is logical theories of connected observation sentences.

How does knowledge lead to action?
Aristotle defines that actions are justified by expected outcome and goals. He defines an iterative algorithm that builds dependencies to achieve a goal.

1.2.2 Mathematics

We need to develop logic, computation, and probability to implement philosophy. Boolean logic is logic on propositions. We can use first-order logic to include objects and relations. These give us the formal rules to draw conclusions.

We have algorithms to prove statements. There are true but undecidable statements. Church-Turing thesis is Turing machines can compute all computable functions. Intractable problems are ones in EXPTIME. (Solvable only in exponential time with respect to input).

Probability allows us to reason with uncertain information. Bayes rule underlies most modern approaches to reasoning in AI systems.

1.2.3 Economics

Economies can be thought of as a individual agents maximizing their success. People make choices that lead to preferred outcomes, or "utility". Decision theory provides a framework for decisions made under uncertainty. Some decisions have order associated with them - uses Marko Decision Processes.

1.2.4 Neuroscience

Neuroscience is the study of brains. Localized areas of the brain are responsible for specific cognitive functions. A collection of simple cells can lead to thought, action, and consciousness. With computers of unlimited capacity, we still don't know how to make a brain.

1.2.5 Psychology

Cognitive psychology views the brain as an information processing device. Stimulus is translated into an internal model. Internal models are manipulated by cognitive processes to form new models. New models are translated into action.

1.2.6 Computer Engineering

Computer performance doubled every 18 months until 2005. Power dissipation problems mean an increase cores instead of clock speed. Work in AI owes much to CS and the converse is also true.

1.2.7 Control Theory and Cybernetics

Controlling something is acting to minimize “error” (state vs goal). Modern control theory designs systems to maximize a function over time. Control theory is similar to AI because AI tries to maximize a function. The two are different because controls are continuous, AI is otherwise.

1.2.8 Linguistics

Understanding language requires an understanding of subject and context, not just sentence structure. It’s hard.

1.3 The History of Artificial Intelligence

1.3.1 The Gestation of Artificial Intelligence (1943-1955)

Initially knowledge of physiology and propositional logic formed a basis for AI. Artificial neurons were supposed to be “on” or “off”. Systems of neurons are Turing Complete [exciting!]. The <u>Hebbian Rule</u> dictates connection strengths between neurons. Alan Turing introduced the Turing Test.

1.3.2 The Birth of Artificial Intelligence (1956)

Logic Theorist (program) was written to think non-numerically
AI embraces creativity, self improvement, and language,
it formed its own field in cs.

1.3.3 Early Enthusiasm, Great expectations (1952 - 1969)

Early AI research was successful because challenges were "AI can't do X". GPS was written to be the first program to establish sub-goals - it "thinks humanly". Using physical symbols as objects is necessary and sufficient for general AI. Geometry Theorem Provider is made to prove tricky geometry problems. A checkers agent was better than it's creator, disproving a theory. Perceptron convergence theorem says that learning algorithms can learn for a given input.

1.3.4 A Dose of Reality (1966-1973)

Initially, AI researchers weren't shy about being optimistic. Later on, they found they were running into three main problems: First, the programs had no contextual basis for making decisions. Second, AI solved by brute-forcing "microworlds", which is bad at scaling. Genetic Algorithms were built because they seemed more minor, but they didn't build good programs. Third, the basic structures imposed limitations; simple perceptrons are limited.

1.3.5 Knowledge Based Systems: The Key to Power? (1969 - 1979)

Weak methods try to solve general problems by stringing together elementary reasoning. They fail for large problem spaces. A better approach is to use domain-specific knowledge to help solve a problem. Cook-book recipes helped solve hard problems from the basis of easy problems. Heuristic programming was for investigating the capabilities of expert systems. Expert systems are expensive, as they require a ton of information gathering.

1.3.6 AI Becomes an Industry (1980-Present)

In 1989 nearly every US corp had an AI group and was using expert systems. Later on, many companies failed on their promises.

1.3.7 The Return of Neural Networks (1986-Preser

Back propagation for neural nets was invented by at least four different groups. Some think that humans manipulate symbols, others don't think connections, not symbols are involved. Current view is that they are complementary, but this question remains open. Modern Neural Network is two fields: One is for effective network architectures. The other is concerned about empirical properties of actual neurons.

1.3.8 AI Adopts the Scientific Method (1987-Prese

AI initially rebelled against Control Theory and Statistics, but is returning to scientific rigor. Speech recognition was tried with limited success in the past. Hidden Markov Models now dominate speech recognition. Hidden Markov Models are based on rigorous mathematical theory, and they can use real-world data. Bayesian networks allow fast representation of uncertain knowledge.

1.3.9 The Emergence of Intelligent Agents (1995-present)

Intelligent agents have been developed more recently to solve specific issues. Some AI Giants (McCarthy, ++) believe that AI should go back to making “machines that think”. General Intelligence AI is ruled by a universal algorithm for learning and acting. General Intelligence AI is the golden goal.

1.3.10 The Availability Of Very Large Data Sets (2001-Present)

Some recent work says if we analyze more data, we don’t need to focus on the algorithm as much. The idea is that you can learn context from large enough corpuses. Small corpuses are pretty much expert systems. Once AI passes the “data bottleneck”, then it becomes scarily accurate.

1.4 The State of the Art

AI can do tons today. It can do driving, speech recognition, autonomous planning/scheduling, game playing, spam fighting, logistics planning, robotics, machine translation, and much more. We will go into these in a much higher level of detail in future chapters.

1.5 Summary

Important points:

- Intelligence is concerned with rational action. Agents try to make good choices.
- Economists formalized decision making.
- Neuroscientists discovered how the way brain works in contrast to computers.
- Computer Engineers provide powerful machines for AI.
- Control theory was the initial basis for AI. Initially, the two were very different, but they grow more similar.
- Recent progress on understanding intelligence has grown at pace with computer capabilities.

- The field of AI has grown substantially; sub-fields of AI are integral to other fields.

Chapter 2

Intelligent Agents

Chapter 1 referred to rational agents, here we define them.

2.1 Agents and Environments

Agents perceive their environment through sensors and act on it with actuators. A "percept" is the agent's perceptual inputs at any given instant. "Percept sequences" are time-series of percept states. "Agent functions" describe the percept sequence to action mapping of an agent. Agent functions are implemented as programs. Agents are meant to be a tool for analyzing systems.

2.2 Good Behaviour - The Concept of Rationality

Rational agents do the right thing, but what is the right thing? Consequences of behaviour is a big measure. Sometimes there is a performance measure to evaluate the environment state. Some measures can be implemented dumbly, they need to be carefully designed.

2.2.1 Rationality

Rationality depends on four tenets:

1. The performance that defines the criterion of success.

2. The agent's prior knowledge of environment.

3. The actions performable by the agent.

4. The agent's percept sequence to date.

Rational agents can be defined as follows:

For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

It is necessary and sufficient that agents that fulfill all four tenets to be rational.

2.2.2 Omniscience, Learning, and Autonomy

Omniscient agents are not possible within reality, because they know the outcome of their actions. Rational agents draw conclusions about everything they can know, not everything knowable. Rational agents then do information gathering to be better at predicting consequences. Rational agents should learn what they should expect to have to compensate for. Applying learned knowledge is a prerequisite to success in many environments.

2.3 The Nature of Environments

We know about rationality, let's talk about environment.

2.3.1 Specifying the Task Environment

Task Environments are tuples of performance, environment, actuators, and sensors. (PEAS) Some task environments like the Roomba are very simple to define while others like taxicab drivers are very hard.

2.3.2 Properties of Task Environments

We can categorize task environments into a few broad categories:

- Fully v.s. Partially observable environments - if the sensors can see everything, then the task is fully observable.
- Single Agent v.s. Multiagent - if the agent is alone, or if there are multiple agents playing against it.
- Deterministic v.s. Stochaistic - if the next step is completely determined by the current state and action, or if there is some sort of randomness involved.
- Episodic v.s. Sequential - Episodic task environments are one-off episodes with consequences of actions. Sequential task environments are where short-term actions have consequences.
- Static v.s. Dynamic - Static environments do not change while computation occurs. Dynamic environments are where waiting for computation is choosing to "do nothing".
- Discrete v.s. Continuous - State, time, and percepts can either be continuously occurring, or distinctly different states, times, and percepts.
- Known v.s. Unknown - In known environments, outcomes are known. In unknown environments, agents will need to understand how it works to make good decisions.

Expectedly, the hardest environment is Partially Observable, Multiagent, Stochaistic, Sequential, Dynamic, Continuous and Unknown. Just like driving a car in an unfamiliar country, with unknown driving laws.

2.4 The Structure of Agents

The job of AI is to design agent programs to map precepts to actions. Agents are compromised of a physical sensor/actuator architecture and a program.

2.4.1 Agent Programs

All agents in this book take current precept as input and control actuators. Agents use current percept as input, agent functions use the entire history. Table-Driven agents implement the desired agent function, but are impossibly hard to populate tables for. Tables of sqrts have been replaced by a five-line newtonian regression method on electronic calculators. The question is: can AI do for general intelligent behavior what Newton did for square roots? We think yes.

2.4.2 Simple Reflex Agents

Simple reflex agents select actions on the basis of current precept only. These are always very simple (e.g. if car-ahead-is-braking then start-braking). They are simple, but they have very limited intelligence; partial observability kills these agents.

2.4.3 Model-Based Reflex Agents

The most effective way to handle partial observability is to keep an internal model of what the agent can't see. These agents need to understand the wold evolving around them, and how their actions affect the world.

2.4.4 Goal-Based Agents

Agents need a goal to base decisions towards. They need to understand some situations as desirable, and others as less. Goal-based action is sometimes straightforward, but is often not. This involves decision about the future (e.g. What will make me happy?). Goal-based agents appear less efficient, but knowledge makes them more flexible.

2.4.5 Utility-Based Agents

Goals are not enough to generate high-quality behavior, as they are only binary. Utility is a measure of "how good" a state is, computed by a utility function. This is not the only way to be rational, but it is much more flexible. *Rational utility-based agents choose the action that maximize the expected utility the agent expects to derive, on average, given the probabilities and utilities of each outcome.* Utility functions allow tradeoffs between different (possibly uncertain) goals.

2.4.6 Learning Agents

Turing proposes that we can build learning machines by building then teaching. Learning elements are responsible for making improvements; Performance elements are responsible for selecting external actions. A Critic compares of the learning element to an external standard. Problem Generators disrupt the monotony of learned things with actions that lead to informative, new experiences.

2.4.7 How the Components of Agent Programs Work

There are a few ways to represent the states of agent programs:

- Atomic Representation - each state has no internal structure.
- Factored Representation - splits up states into a set of attributes.
- Structured Representation - world is things that are related, not just variables with value.

The more expressive a representation, the more complex learning and reasoning become.

2.5 Summary

This chapter tours AI really quickly, but here are some key points:

- Agents perceive and act in an environment.
- An agent function for an agent specifies the action taken in response to any percept sequence.
- The performance measure evaluates behaviour of the agent.
- A rational agent acts to maximize the expected value of an action given the precept sequence so far.
- Task environments include the performance measure, external environment, the actuators, and the sensors.
- To design agents, we should specify task environments first.
- Task environments vary along dimensions. (agents, determinsitic, etc.)
- The agent program implements the agent function.
- There are a bunch of basic agent-program designs reflecting info used in the decision process.

- Simple reflex agents respond to percepts, model-based reflex agents maintain a model. Goal-based agents act to achieve goals, utility-based agents act to maximize utility (“happiness”.)
- All agents can improve through learning.

Chapter 3

Solving Problems by Searching

Remark 1. *At this point I only started writing summaries, instead of breaking down notes at every section.*

3.1 Summary

We can use graph (or tree) searches to make decisions to achieve goals. These work well in environments that are deterministic, observable, static, and completely known.

- Goals need to be identified before searching.
- Problems are 5-tuples:
  - Initial state

- A set of actions
- A transition model (state x action $\rightarrow$ state)
- A goal test function
- A path cost function

- Search algorithms treat states and actions as atomic.
- Tree-searches consider all possible paths to find a solution, graph-searches avoid redundant paths.
- Search algorithms are judged on the basis of completeless, optimality, time complexity, and space complexity.
- Uninformed search methods only access the problem definition. They include:
  - BFS/DFS
  - Uniform-cost search expands the node with the lowest path cost.
  - Iterative deepening DFS iteratively increases the depth limit of DFS until a goal is found. It boasts $O(n)$ space, it is optimal for unit step cost and has time complexity comparable to BFS.
  - Bidirectional search techniques are possible sometimes, and greatly reduce time complexity.

- Informed search methods may use heuristic functions $h(n)$ to estimate the cost of a solution from n.
  - Best-first search solution is like BFS, and uses $h(n)$ to pick the node.
  - Greedy best-first solutions expands nodes with minimal $h(n)$. It is not optimal, but is fast.
  - A* expands nodes with minimal $f(n) = g(n) + h(n)$ ($g(n)$ is the cost so far). We say $h(n)$ is admissible when it never over-estimates the cost to n. The best heuristic is the closest to the actual cost, so multiple heuristics can be combined by $\max(h_1(x), h_2(x), \ldots)$.
  - RBFS (Recursive best-first search) and SMA* (simplified memory bounded A*) are robust and optimal, and only use limited memory.
- Heuristic search performance depends on the accuracy of $h(n)$.