Introduction to Machine Learning

Introduction

Varun Chandola

January 31, 2017

Outline

Contents

1 Making Intelligent Machines

Artificial Intelligence or AI has been getting immense publicity in recent years, mostly due to advances in deep learning methods and their applications to areas such as self driving cars, robotic assistants, and language translation systems. The area of AI is more than half a century old, when computer scientists and mathematicians embarked on the quest for making intelligent machines. Now that idea pervades or threatens to pervade every aspect of our daily life. Read an interesting article on AI in "The Great A.I. Awakening" in New York Times, December 2016.

2 Turing's Test

Alan Turing posed the following question in his seminar paper [1] - "Can Machines Think?" One of the biggest drawbacks of the Turing Test was the fact that it does not directly test the intelligence of a machine, but the fact that can a machine be created which can imitate human behavior, which is different from intelligence.

- Can machines think?
- Simple version - Can machines imitate humans?
- E.g., ELIZA
  - `http://www.manifestation.com/neurotoys/eliza.php3`
- Difference between "human behavior" and "intelligent behavior".

3 What Makes an Artificial Machine Intelligent

1. Talk. See. Hear.
   - Natural Language Processing, Computer Vision, Speech Recognition
2. Store. Access. Represent. (Knowledge)
   - Ontologies. Semantic Networks. Information Retrieval.
3. Reason.
   - Mathematical Logic. Bayesian Inference.

4. **Learn.**
    - Improve with Experience
        - Machine Learning

4 Human Learning

- What do we learn?
    - Concepts (this is a chair, that is not a chair)
    - Distinguishing concepts (this is a chair, that is a table)
    - Other things (language, juggling, using a remote)
- How do we learn?
    1. Teaching (Passive).
    2. Experience (Active).
        (a) Examples.
        (b) Queries.
        (c) Experimentation.

5 Definition of Machine Learning

- Computers learn without being **explicitly programmed**.
    - Arthur Samuel (1959)
- A computer program learns from experience E with respect to some task T, if its performance P while performing task T improves over E.
    - Tom Mitchell (1989)

5.1 Why Machine Learning?

One could argue as to why one would want a machine to "learn" over experience instead of programming a machine to "know" everything. But such approach is infeasible. First, because knowing everything will require a significant amount of storage. Second, the success of such a system assumes that the creator already knows everything about the problem, which often is an optimistic and false assumption.

Machine learning is implicitly based on the notion that the new experiences that a machine will encounter will be similar to the past experiences. Hence, instead of "pre-loading" the machine with knowledge about all possible experiences, it is efficient to selectively learn from these experiences. Moreoever, as long as the new experience has some structural relationship with the past experiences, the machine can perform well in unseen situations as well.

- Machines that know everything from the beginning?
    - Too bulky. Creator already knows everything. Fails with new experiences.
- Machines that learn?
    - Compact. Learn what is necessary.
    - Adapt.
    - Assumption: Future experiences are not too different from past experiences.
        - Have (structural) relationship.

5.2 Example

- Our future overlords (Kiva)
    - https://youtube.googleapis.com/v/6KRjuuEVEZs

Current warehouse robots like Kiva do not learn over time. They are programmed to behave in a certain manner. We consider the scenario of making the robots even smarter by allowing them to stack objects into appropriate locations for efficient retrieval.

How would go about designing such system? The key task to be executed is: given an input object, can the robot tell if it is a chair or a table. There are some obvious engineering issues here. How does the robot take the object as input? How does it communicate (tell) its decision? We will ignore those for this course and assume that the robot's decision making module is a software which takes a digital representation of the object as input and outputs a binary signal, 0 for table and 1 for chair.

We still need to figure out what the digital representation is going to be. Here is the first key difference between humans and machines, in the context of learning. Humans automatically perceive a sensory input. Machines see a machine translated input.

Using raw data directly to distinguish between tables and chairs might not work. Different objects of the same type (e.g., tables) might "appear" different in terms of the raw input (It should be noted that in some situations, e.g., in handwritten digit recognition, using raw input directly might still work as long as the different objects belonging to same class are roughly similar, such as different ways in which the number '9' can be written.).

Thus instead of using raw digital data, it is better to use higher order descriptions or **features**. This leads to the question – what features? Easy answer is, features that characterize the aspect of the objects that distinguishes one type from another. For tables and chairs, one feature might be – "has back rest". This feature, which takes only two values (yes or no) is appropriate. Obviously, the best would be the binary feature – "is a chair"!! But getting such a feature is expensive (in fact it is the knowledge that is being learnt). This is the first lesson: **features should be easy to compute**.

What about the feature – "has four legs"? While this feature is true for chairs (mostly), it is also true for tables. So using this feature will not help. Second lesson: **features should be distinguishing**.

The feature – "is white" maybe too restrictive, it only describes white chairs and will be false for chairs of other colors. Third lesson: **features should be representative**.

6 Learning from (Past) Data

Deductive Logic

- All birds can fly
- Dodo is a bird
- ⇒ Dodo can fly

Inductive Logic

- A stingray can swim
- Stingray is a fish
- ⇒ All fish can swim

Core Tenet

- ~~Deduce~~ Induce from past
- Generalize for future

7 Simple ML Example

`www.r2d3.us/visual-intro-to-machine-learning-part-1/`

References

References

[1] A. M. Turing. Computing machinery and intelligence. Mind, 59:433–460, 1950.