Introduction to Machine Learning

CSE474/574: Lecture 2

Varun Chandola <chandola@buffalo.edu>

28 Jan 2015

Outline

Contents

1 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".

2 Machine Intelligence

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

3 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)

3.1 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.

3.2 Running 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**.

4 Concept Learning

The basic form of learning is *concept learning.* The idea is to learn a general description of a category (or a concept) from specific examples. A *concept* is essentially a way of describing certain phenomenon; an object such as a table, an idea such as *steps that will make me successful in life*.

The need to go from specific to general is the core philosophy of machine learning.

In the context of machine learning, concepts are typically learnt from a set of examples belonging to a super-category *containing* the target category; furniture. Thus the learnt concept needs to be able to distinguish between the target and everything else. The goal of concept learning is:

- Infer a boolean-valued function $c : x \rightarrow$ {`true`,`false`}
- *Input*: Attributes for input x
- *Output*: `true` if input belongs to concept, else `false`
- Go from specific to general (*Inductive Learning*).

4.1 Example – Finding Malignant Tumors

Attributes

1. **Shape** `circular,oval`
2. **Size** `large,small`
3. **Color** `light,dark`
4. **Surface** `smooth,irregular`
5. **Thickness** `thin,thick`

Concept
Malignant tumor.

For simplicity, we assume that the attributes that describe the objects are *boolean.* One can even think of these attributes as constraints over the actual attributes of the objects.

4.2 Notation

- X - Set of all possible instances.
  - What is $|X|$?
- Example: {`circular,small,dark,smooth,thin`}
- D - Training data set.
  - $D = \{\langle x, c(x)\rangle : x \in X, c(x) \in \{0, 1\}\}$
- Typically, $|D| \ll |X|$

The number of all possible instances will be 2^d, where d is the number of attributes. If attributes can take more than 2 possible values, the number of all possible instances will be $\prod_{i=1}^{d} n_i$, where n_i is the number of possible values taken by the i^{th} attribute.

4.3 Representing a Possible Concept - Hypothesis

As mentioned earlier, a concept can be thought of as a function over the attributes of an object which produces either *true* or *false*. For simplicity, we assume that the concept is:

- A conjunction over a subset of attributes
  - A malignant tumor is: *circular* **and** *dark* **and** *thick*
  - {`circular,?,dark,?,thick`}

In practical setting this is too simplistic, and you will rarely find a setting where a simple conjunctive concept will be sufficient.

The target concept is *unknown.* We have made strong assumptions about the *form* of the target concept (conjunctive), but it still needs to be learnt from the training examples.

- Target concept c is unknown
  - Value of c over the training examples is known

4.4 Hypothesis Space

- **Hypothesis**: a potential concept
- Example: {`circular,?,?,?,?`}
- **Hypothesis Space** ($\mathcal{H}$): Set of *all hypotheses*
  - What is $|\mathcal{H}|$?
- Special hypotheses:
  - Accept *everything*, {`?,?,?,?,?`}
  - Accept *nothing*, $\{\emptyset, \emptyset, \emptyset, \emptyset, \emptyset\}$

As in most *inference* type of tasks, concept learning boils down to searching for the best hypothesis that approximates the target concept from the entire hypothesis space.

As shown above, in a hypothesis, each binary attribute can have 3 possibilities, '?' being the third one. There can be total 3^d possible hypotheses. There is one more hypothesis which *rejects* (gives a negative output) for any example. Thus there can be $3^d + 1$ possibilities. In general, the size of the hypothesis space, $\mathcal{H} = \prod_{i=1}^{d} n_i + 1$.

Note that we represent the *accept nothing* or *reject everything* hypothesis with $\{\emptyset, \emptyset, \emptyset, \emptyset, \emptyset\}$ or just $\emptyset$ for simplicity. One can think of the value '$\emptyset$' to signify that the input be rejected no matter what the value for the corresponding attribute.

References

References

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