**CSE 513T Theory of Artificial Intelligence & Machine Learning** August 30, 2016

# Lecture 1

*Lecturer: Brendan Juba* *Scribe: Hao Pan, Editor: Alexander Cruise*

## 1 Introductory Query

Class interaction: Pick 10 people in the class randomly (with replacement) to ask why they are interested in taking this course. These answers bring up two following questions:

1. What does it achieve to ask 10 people out of 50 why they are interested in taking this class?
2. How can I infer anything about the 40 that I didn't ask?
3. What can/can't I learn by asking a few selected at random?
4. What can I actually do with this knowledge?

These questions are representative of the main themes of this course.

## 2 Knowledge Representations

### 2.1 Knowledge(Classical)

We consider a 'domain' of 'space of instances': some set X typically defined by some (finite) list of attributes: e.g. $X \equiv$ all possible students.

- $x_1 \equiv$ wants to go deeper into the machine learning.
- $x_2 \equiv$ wants to learn the theory of machine learning.
- $x_3 \equiv$ nice professor.
- ...

We can abstractly represent the set by a number of vectors of Booleans. Each $x \in X$ is a vector of $n$ booleans, each boolean representing a reason the student $x$ is interested in the class. $X$ is the space of all possible students, and the cardinality of this set is $|X| = 2^n$. $D$ is the space of actual students, has a cardinality of $|D| =$(number of students), and is a sub-space of X. Any $C \subseteq X$ specifies a property. We'll say that $C$ is valid or true if $D \subseteq C$.

e.g. $C = \{x \in X$:'expect to understand Brexit'$= 0\}$. ($C$ is the set of students that were not here for understanding Brexit). Presumably, $C \subseteq X$ is valid here. Let $C \subset X$('nontrivial property'). Then until we examine every last $x \in D$, we can't conclude that C is valid for D.

### 2.2 PAC Semantics

Suppose that instead of a set of possible actual worlds D ($D \subseteq X$), we have a probability distribution D over X. e.g. each student has probability of 1/50 of being queried. As before, we can't establish anything about 100% of D without observing 100%. But, we can make a good guess about 90% without having to ask anywhere close to 90%. Suppose 10% of students did enter the class with the hope of gaining an

understanding of brexit. How likely was were we to miss this 10% when we choose 10 at random (with replacement)? How about I choose another 10 people?

Since we're asking 10 independently.

Pr[all 10 miss the Brexit students]=Pr[miss the Brexit student at a single selection]$^{10}$

Pr[all 10 miss the Brexit students] = $(0.9)^{10} \approx 0.34$

If I asked another 10, just square it:

Pr[all 20 miss the Brexit students] = $(0.9)^{20} \approx 0.11$

So the conclusion 'no one is interested in improving their Brexit game' is Probably(0.89) Approximately(0.90) Correct('PAC').

Note that this conclusion does not depend on the size of the class.

Even if there were 100 or 10,000 in the class, by examining 20% we gain 89% confidence about 90%.

## 2.3 Representing Properties

Helpful to think of our possible worlds as a database (Property can be shown as a database):

| | play a mean guitar | gives you up | lets you down |
|---|---|---|---|
| Rick Astley | 0 | 0 | 0 |
| Fraddie | 0 | 1 | 0 |
| Jimmy Hendrix | 1 | 1 | 0 |

Properties (e.g. 'C') are defined by the sets of attributes that make them true. Instead of writing out the database defining C, we'd prefer some more compact Knowledge Representation.

**Conjunctions:** one example of a compact knowledge representation is the conjunction, shown below for Rick Astley.

$$\text{Rick Astley} = \neg\text{gives you up} \wedge \neg\text{lets you down} = \varphi(x)$$

- Where $\neg$ and $\wedge$ represent Boolean operators 'NOT' and 'AND', respectively.
- A Boolean variable or its negation(e.g. $\neg$gives you up) is referred to as a literal.
- The class of conjunctions consists of all possible sets of literals joined by the AND connector.

**Decision Trees:** another example of a compact knowledge representation is the decision tree(rooted binary tree). An example decision tree is shown below.

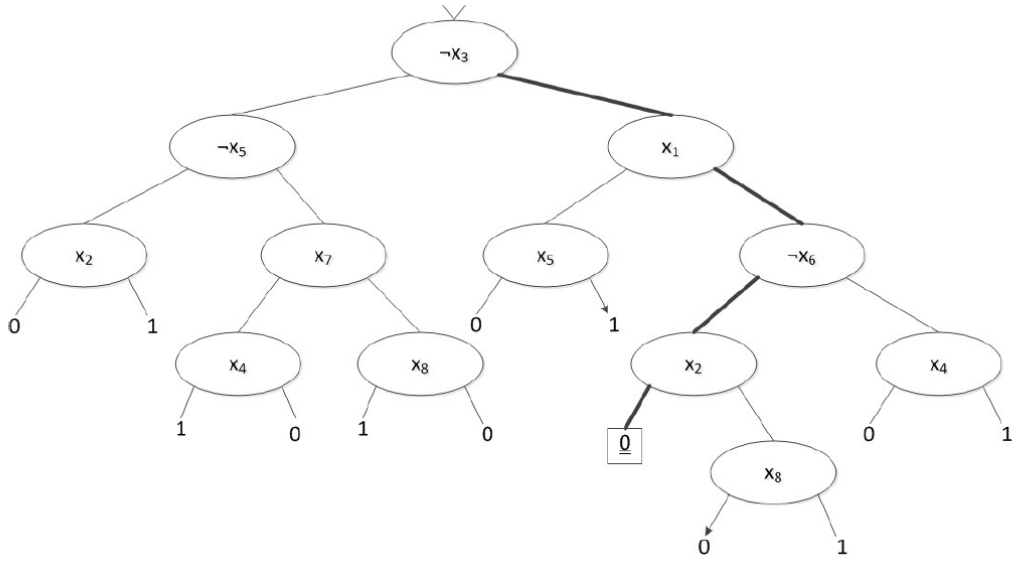


**Figure 1**: Decision Tree

- Internal nodes labeled by literals.
- Leaves labeled by Boolean values.
- Right when literal is true
- Left when literal is false

e.g. $\overrightarrow{X} = (10010110)$ is mapped to '0': does not have the represented property. The darkened path segments represent the path followed by $\overrightarrow{X} = (10010110)$. The conjunction $\neg x_3 \wedge x_1 \wedge x_6 \wedge \neg x_2$ defines the 'bucket' that collects x. Formally, a "Knowledge Representation" identifies a subset of X with a string over an alphabet (e.g. Boolean encoding) that captures one of these structures. We want to be able to evaluate whether or not the property holds on a given instance efficiently. So we need the representations to be 'small' so that the algorithms would be fast at evaluating membership. Knowledge representations differ in their expressive power. Every conjunction can be written as a decision tree, but there are decision trees that can't be written as conjunctions.