CSE 513T Theory of Artificial Intelligence & Machine Learning Jan. 16, 2018

Lecture 0

Lecturer: Brendan Juba *Scribe: Jizhou Huang & Anthony Dulay*

1 Survey

We asked 10/40 at random: what do you hope to get out of the course. Part of the answers are listed below.

- x_1: "learn machine learning theory - analysis of algorithms"
- x_2: "get background in theory of ML, understand applications"
- x_3: "ML/AI currently black - box, hope to understand it"
- x_4: "want to know whether to pursue deeper in ML"
- .....

Each of these answers will be treat as an attribute in the following sections.

2 A Classical Definition of Knowledge

We consider a "domain" or "space of possibilities" $\mathcal{X}$ typically described by a list of *attributes*. We abstractly represent these attributes by some number ("n") of boolean attributes (later, we may consider more general attributes). So, we think of $\mathcal{X}$ as $\{0,1\}^n$ and each $x \in \mathcal{X}$ is a boolean vector, where $x_i \in \{0,1\}$ indicates if the ith attribute of x is true.

3 PAC Semantic

Any subset $\mathcal{C} \subseteq \mathcal{X}$ defines a property. Likewise, the actual space of possibilities $\mathcal{D}$ is another subset of $\mathcal{X}$. As an instance in this lecture, we will treat $\mathcal{D}$ as the set of students actually in this class. If $\mathcal{D} \subseteq \mathcal{C}$ then every $x \in \mathcal{D}$ satisfies $\mathcal{C}$, therefore, we would say "$\mathcal{C}$ is valid for $\mathcal{D}$". For example, $\mathcal{C} = \{x \in \mathcal{X} | \text{wanted to understand spectre bug} = 0\}$ is valid for $\mathcal{D}$ if, presumably, no one came here to learn about computer security. If $\mathcal{C}$ is a proper subset of $\mathcal{X}$ ($\mathcal{C} \subset \mathcal{X}$), we cannot conclude $\mathcal{C}$ is valid for $\mathcal{D}$ unless we ask every single student.

Suppose instead of a set of possibilities $\mathcal{D}$, we had a probability distribution of $\mathcal{D}$ over $\mathcal{X}$. e.g. All 40 students have a probability of $\frac{1}{40}$ of being chosen. So, while we are not able to know everything 100% without asking all the students, we can make a good guess on what 90% of us are expecting by asking far less than 90 percent.

So lets suppose $\geq 10\%$ wanted to understand spectre bug. How likely were that 10% of the students missed by the survey? What is the probability that none of 4 students were chosen in these 10 random selections? Each time we randomly choose one, the probability that we missed all of them would be 0.9. Since every choice was made independently, the probability we missed in all 10 times is $0.9^10 \approx 0.34$. If we were to ask another 10 people, the probability that I miss in all 20 times is ≈ 0.12. So, this is saying that we learn about approximately 90% of the class with a 88% probability. Therefore, this claim is Probably Approximately Correct (PAC Semantic).

The same analysis hold true even if the population is arbitrary large. So, suppose we only queried 20 people out of 1,000,000, we still can get the 88% confident about 90% of the population.

4 Knowledge Representation

Properties are defined by the sets of attributes values that satisfy them. We can always think of this as a database

	gives you up	lets you down	plays guitar
Jimi Brexit	1	0	1
Coldplay	0	1	1
Rick Astley	0	0	0

Table 4.1: Database representation

Talk about defining the property of Rick Astley

gives you up	lets you down	plays guitar
0	0	0
0	0	1

Table 4.2: Property of Rick Astley

4.1 Conjunction

However, we prefer a more compact representation, which is *Conjunction*. For example, we may express the properties of Rick Astley in conjunction as "¬gives you up ∧ ¬lets you down".

Conjunction: AND of "literals", such as $l_1 \wedge \neg l_2 \wedge l_3$.
Literal: A boolean attribute or its negation, such as "¬gives you up" and "lets you down".

Figure 4.1: Definition of conjunction

4.2 Decision Tree

Another example is decision tree as shown in figure 4.2.

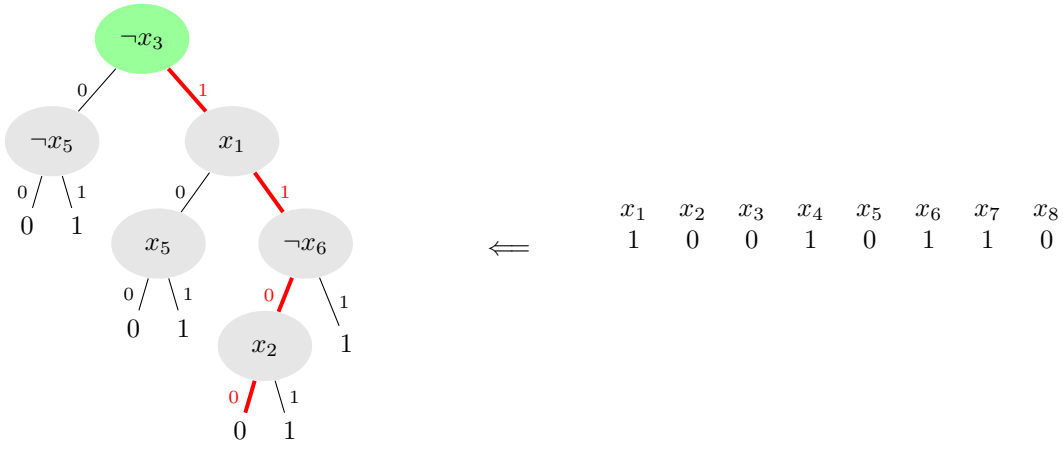


Figure 4.2: Decision tree (left) represents the same knowledge in database (right)

4.3 Expressive Abilities

Formally, a knowledge representation just assigns subsets of $\mathcal{X}$ to strings of symbols over alphabet. We also wish to be able to evaluate the representation on points $x \in \mathcal{X}$ efficiently (conjunctions and decision trees generally do have ties quality).

Representations differ in expressive abilities. One can express conjunctions as decision trees. An example is given in figure 4.3.

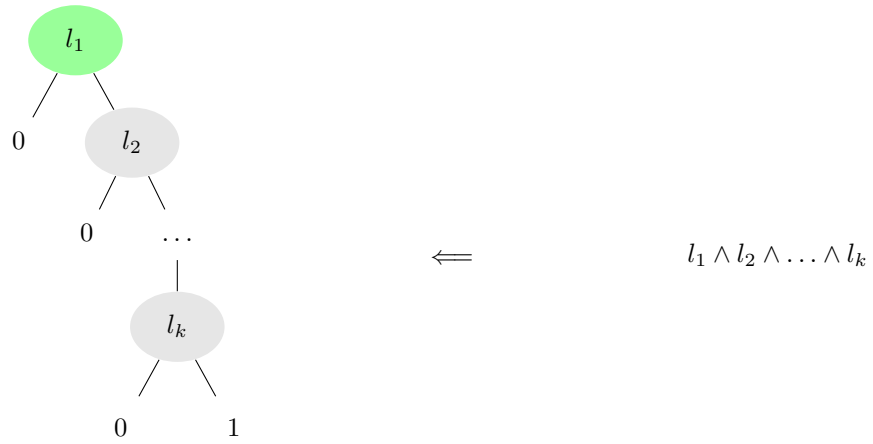


Figure 4.3: An example of how a conjunction (right) is converted to a decision tree (left)

However, there are decision trees which cannot be written as conjunctions.

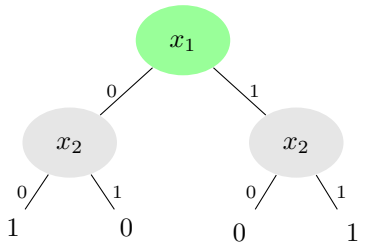


Figure 4.4: XNOR logic expressed by decision tree

To convert the decision tree shown in figure 4.4 to an equivalent conjunction, we cannot include either "x_1 or x_2" or "$\neg x_1$ or $\neg x_2$". That is to say, we can't have any literals always be 1, but it's not.

So, why do we even talk about conjunctions at all? Well, one reason is that we will see a fast algorithm for acquiring a conjunctive definition of a property from examples. No similarly fast algorithm is currently known to exist for decision trees. In general, more expressive power always comes at some cost, understanding costs and trade-offs involved is one of the main themes to the course.

Roughly, we'll see given a list a of examples drawn from $\mathcal{D}$ labeled by whether or not those examples satisfy the unknown property $\mathcal{C}$. Our task would be to form a conjunction $\mathcal{C}$ that agrees with the unknown property with high probability over $\mathcal{D}$. So the rule for this would be that this column "$\mathcal{C} = c(x)$" is true for most of $\mathcal{D}$ with high probability.