CSE 513T Theory of Artificial Intelligence & Machine Learning Oct 11, 2016

Lecture 13

Lecturer: Brendan Juba *Scribe: Wei Wang*

Today's Topics

- Learning halfspaces
- Declarative knowledge & Reasoning
- Proof systems

1 Learning halfspaces

Theorem 1 *Let c be a Boolean property, $\mathcal{H}$ be any representation class of VC-dimension d. Then any solution h to $Consis_{\mathcal{H}}$ on a data set containing $O\left(\frac{1}{\varepsilon}(d\log\frac{1}{\varepsilon}+\log\frac{1}{\delta})\right)$ examples drawn from an arbitrary distribution $\mathcal{D}$ and labeled by c, satisfies w.p. $\geq 1-\delta$, $P_{x\in\mathcal{D}}[c(x)=h(x)]\geq 1-\varepsilon$.*

Recall in HW2, we see VC dimension of halfspaces in $\mathcal{R}^n$ is $n+1$, so $m=O\left(\frac{1}{\varepsilon}((n+1)\log\frac{1}{\varepsilon}+\log\frac{1}{\delta})\right)$ examples suffice to fit a halfspace in $\mathcal{R}^n$. Notice $Consis_{\text{Halfspace}}[(x^{(1)},b^{(1)}),\cdots,(x^{(m)},b^{(m)})]$ is the problem that finds $w_1,\cdots,w_n,t\in\mathcal{R}$ satisfying the following m constraints:

$$w_1x_1^{(i)}+\cdots+w_nx_n^{(i)}\geq t\ , \qquad i\in\{i=1,\cdots,m:b^{(i)}=1\}$$

$$w_1x_1^{(i)}+\cdots+w_nx_n^{(i)}< t\ , \qquad i\in\{i=1,\cdots,m:b^{(i)}=0\}$$

This is an instance of linear programming. Data points give constraints for LP.

Rewrite it in the standard form: find $w_1,\cdots,w_n,t\in\mathcal{R}$ s.t.

$$-x_1^{(i)}w_1-\cdots-x_n^{(i)}w_n+(1)\cdot t\leq 0, \qquad i\in\{i=1,\cdots,m:b^{(i)}=1\}$$

$$x_1^{(i)}w_1+\cdots+x_n^{(i)}w_n-(1)\cdot t< 0, \qquad i\in\{i=1,\cdots,m:b^{(i)}=0\}$$

which is of the form $A(\mathbf{w},t)\leq\mathbf{0}$ ('homogenous' LP).

Notice, for any A, we can use an algorithm for solving $Consis_{\text{Halfspace}}$ to find $\mathbf{y}$ s.t. $A\mathbf{y}\leq\mathbf{0}$, i.e. to find a feasible point for the homogenous LP $A\mathbf{y}\leq\mathbf{0}$.

Recall, Lemma (informal) proper PAC learning of halfspaces enables us to solve $Consis_{\text{Halfspace}}$ $\Longrightarrow$ solve homogenous LP feasibility.

Theorem 2 *(Kachiyan'79, Karmarkar'84) There are efficient algorithms to solve linear programming.*

Corollary 1 *Halfspaces are PAC learnable.*

Figure 1 summarizes the learnability of representations.

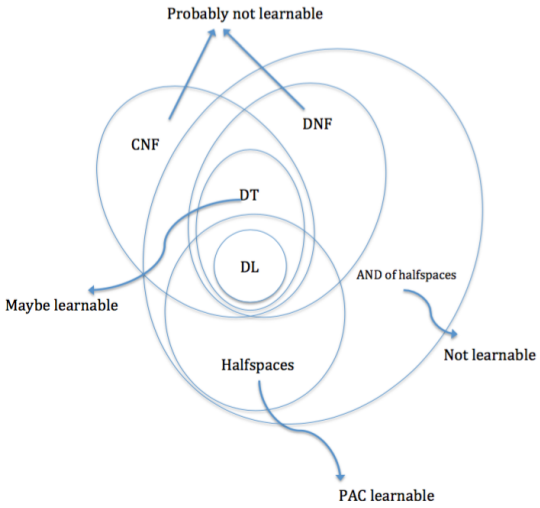

Figure 1: Learnability of Representations

2 Declarative knowledge & Reasoning

2.1 Declarative knowledge

Recall we had a set $\mathcal{X}$ of possible values of our attributes, encoding possible situations/scenes/worlds, given by the settings of n Boolean (propositional) attributes describing one aspect of these scenes. For example, we might have an attribute $x_1 \equiv$ "there is a dog", and $x_2 \equiv$ "there is a furry animal". For convenience, we might denote these as literals on some $x \in \mathcal{X}$ as dog(x) $(= x_1)$, furry_animal(x) $(= x_2)$. $\mathcal{X}$ contains all possible combinations of $x_1 + x_2$. The set of actual possibilities (that we wish to reason about) D might only contain scenes in which whenever dog(x) is true, furry_animal(x) is also true. That is, the set $C = \{x : \text{dog}(x) \Rightarrow \text{ furry_animal}(x)\}$ contains D ($C \supseteq D$). We say C is "valid for D". We write $D \models C$. We associate the set C with the representation of the condition used to define it here, dog $\Rightarrow$ furry_animal. Such representations are called declarative knowledge.

We've talked about how to acquire representations from examples in PAC learning. Declarative knowledge captures assumptions we make about the world or knowledge we acquire from book etc. We refer to the collection of such representations defining the constraints that a system possesses (on the space of actual possibilities) as its knowledge base. If the knowledge base contains representation $\psi_1, \cdots, \psi_k$, then the knowledge base (KB) corresponds to the representation $\psi_1 \wedge \cdots \wedge \psi_k$, i.e. $\bigcap_{i=1}^{k}\{x : \psi_i(x)\}$.

Suppose that the KB contains the representation dog $\Rightarrow$ furry_animal, and the representation dog. In such scenes, furry_animal will also be valid. We say that furry_animal is entailed by the KB dog, dog $\Rightarrow$ furry_animal. Following the notation from earlier, we write {dog, dog $\Rightarrow$ furry_animal } $\models$ furry_animal, or more typically, just KB $\models$ furry_animal for the above KB.

Notice, a small KB may entail a large collection of facts. It's generally infeasible to list out all of these possible facts explicitly.

2.2 Reasoning

Reasoning (or deductive inference) is the task of deciding whether or not a representation is entailed by a given KB.

Formally, for classes of representations $\mathcal{F}$ and $\mathcal{H}$, a reasoning task has the form: given a KB $\subseteq \mathcal{H}$ and a query $\varphi \in \mathcal{F}$, output "accept" or "reject".

1. If the algorithm only accepts when KB $\models \varphi$, we say that it is sound.
2. If the algorithm accepts φ whenever KB $\models \varphi$, we say that it is complete.

Notice, for a KB$= \{\psi_1, \cdots, \psi_k\}$ and a query φ, KB does not entail φ iff there is a $x^* \in \mathcal{X}$ s.t. $x^* \in \{x : \psi_1(x) \wedge \cdots \wedge \psi_k(x)\}$ and $x^* \notin \{x : \varphi(x)\}$. So, it is equivalent to deciding if $\psi_1 \wedge \cdots \wedge \psi_k \wedge \neg\varphi$ is satisfiable.

In particular, if $\mathcal{H}$ contains all possible ORs on three literals (and $\mathcal{F}$ is almost anything), we see that sound and complete reasoning would allow us to solve 3SAT. So, it is NP-hard. Since finding counterexamples is sufficient, it is an NP-searching problem. Thus, it's NP-complete.

3 Proof system

Instead of restricting $\mathcal{H}$, we relax completeness. We consider a limited family of proofs, and we will only require the algorithm to accept when there is a proof of φ from KB$= \{\psi_1, \cdots, \psi_k\}$ in the family. This limited family of conclusions that we consider is a "proof system".

Proof systems will vary in their expressive power, and we will ask for which proof system is efficient reasoning possible?