Learning Theory for Nontrivial Inference

Xiyu Zhai

Contents

Abstract

We study learning theory in the setting where inference is nontrivial computationally. We show that semisupervised learning algorithms based on select-verify has a nontrivial advantage over supervised learning, including close to human sample complexity. We specialize our theories for computer visions and did experiments to verify our claims. This shall shed light upon the future evolution of AI, leading to sample efficient deep learning or even better.

1 Introduction

Deep learning has been very successful in the passing decade, however, it's still far from human in certain aspects, say, sample complexity for few shot learning. It takes human very few examples to recognize objects, play games well, but it takes machines thousands times more to perform at the same level. It's still unclear how to overcome these shortcomings.

We shed light on these issues through rethinking machine learning setup. Typically, machine learning theory assumes no computational difficulty in inference, which in our

views potentially misses important structures. We make a new PAC learning setup that takes into account nontrivial computational structure in inference, such that there is a tradeoff between sample complexity and computational complexity leading to nontrivial algorithms.

Our discussion is domain specific in nature. We specialise our theories for computer vision to study the case of shape classification, MNIST for example. We did experiments to verify the assumptions aligh perfectly with the dataset. This relates to Hinton's work on deformable models.

2   Related Work

Energy Model

3   Setup

Input space $\mathcal{X}$, output space $\mathcal{Y}$, a realizable hypothesis class $\mathcal{H}_0$, i.e. an a priori set containing functions from X to Y.

Note that we use $\mathcal{H}_0$ to denote that it's special. It's the minimal hypothesis class that is realizable based on a priori assumptions. We shall define more hypothesis classes because it's probably hard to work in the original $\mathcal{H}_0$.

We assume that functions in $\mathcal{H}$ is computationally nontrivial. In this paper, we assume that there are sets $\mathcal{W}$ (weight space) and $\mathcal{C}$ (configuration space) and a score function $s : \mathcal{X} \times \mathcal{W} \times \mathcal{C} \times \mathcal{Y} \to \mathbb{R}$, then

$$h_w(x) := \operatorname*{argmin}_{y \in \mathcal{Y}} \sup_{c \in \mathcal{C}} s(x, w, c, y). \tag{1}$$

if argmin gives more than one element, pick the one according to some predefined order.

We assume s is easy to compute, at least in P.

Remark. *This can lead to NP problem.*

Suppose $\mathcal{Y} = \{true, false\}$, and that $s(x, w, c, false) \equiv 0$, then $h_w(x) = true$ when

$$\sup_{c \in \mathcal{C}} s(x, w, c, true) > 0 \tag{2}$$

which is equivalent to

$$\exists c \in \mathcal{C}, s(x, w, c, true) > 0, \tag{3}$$

which is in NP.

Pick a nice s, we can make h_w NP-hard.

However, we wouldn't necessarily make it this hard, but harder than there could be a simple "analytical" solution for this.