

Sample-Efficient Pattern Recognition: Model-Based Reinforcement Learning is All You Need

Xiyu Zhai

Contents

1	Introduction	1
2	Related Work	2
3	Setup	2
4	Relaxation of Hypothesis Class	3
5	Learning Complexity	3
6	Experiments on MNIST	3
7	Conclusion	5
8	Future Work	5

Abstract

We study pattern recognition by viewing it as a model-based reinforcement learning problem to gain minimize sample complexity. In fact, model-based reinforcement learning is actually a vehicle converting the inherent NP problem of pattern recognition to ML problem. We verify our theories on the MNIST dataset.

1 Introduction

Deep learning has been very successful in the passing decade, especially for pattern recognition tasks like image classification. However, it's still far from human in terms of sample complexity. Hence we seek to understand what it takes to have minimal sample complexity for pattern recognition tasks, which can lead to better deep learning or more advanced form of AI.

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 align perfectly with the dataset. This relates to Hinton’s work on deformable models.

2 Related Work

Bayesian Model

Energy Model

Theoretical Computer Science

Reinforcement Learning

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} (world stone space) and \mathcal{A} (action space) and a reward function $r : \mathcal{X} \times \mathcal{W} \times \mathcal{A} \times \mathcal{Y} \rightarrow \mathbb{R}$, then

$$h_w(x) := \operatorname{argmax}_{y \in \mathcal{Y}} \sup_{a \in \mathcal{A}} r(x, w, a, y). \quad (1)$$

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 (as commonly known by RL).*

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

$$\sup_{a \in \mathcal{A}} r(x, w, a, true) > 0 \quad (2)$$

which is equivalent to

$$\exists a \in \mathcal{A}, r(x, w, a, true) > 0, \quad (3)$$

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 so that learning is needed.

4 Relaxation of Hypothesis Class

We can approximate h_w by

$$\tilde{h}_{w,\{c_1,\dots,c_n\}} := \operatorname{argmax}_{y \in \mathcal{Y}} \max_{i \in [n]} r(x, w, a_i(x), y). \quad (4)$$

where a_1, \dots, a_n are policies $\mathcal{X} \rightarrow \mathcal{A}$, called action suggestors.

Basically, we break a ML problem into learning a verifier as world model and then learning action suggestors. Learning verifier needs labeled samples, but learning action suggestors needs only unlabeled samples.

5 Learning Complexity

Suppose

6 Experiments on MNIST

We claim our theories characterize exactly the MNIST dataset.

Example (MNIST). Here we describe briefly a function in mathematical terms which we believe is the ground truth for the MNIST dataset. Details can be seen in appendix.

We take the convention that the fill of the digits is white and the background is black.

The image is represented by a $[0, 1]$ -valued 28×28 matrix $I = (I_{ij})_{0 \leq i \leq 27, 0 \leq j \leq 27}$.

- **digit one of the simplest kind**, which constitutes 95% of all images of digit one.

A typical image looks like this:

[an image here]

Think about how it's drawn. The person when writing down a digit one like this has an ideal version in mind, a straight line that is almost vertical. So take Γ_1 to be the set of straight lines with slopes satisfying some easy constraint. Then we should define $\mathfrak{s}(x; \gamma)$ for $\gamma \in \Gamma_1$ such that

- for "most" points over γ , it's surrounded by white pixels;
- for "most" non-white pixels, it's away from γ .

One choice could be

$$\mathfrak{s}(x; \gamma) = a_1 \int_0^1 \max_{(i,j) \in [27] \times [27]} 1_{\|\gamma(t) - (i,j)\|_2 < \epsilon} I_{ij} dt - a_2 \sum_{(i,j) \in [27] \times [27]} 1_{I_{ij} < 0.5} \operatorname{dist}((i,j), \gamma) \quad (5)$$

where ϵ is an appropriate small number and $a_1, a_2 > 0$ are appropriate coefficients. In fact the function applies when γ is any path, not necessarily a straight line. Formally it is defined over the path space (without basepoint) $\Gamma = M([0, 1], [0, 1]^2)$. The dimensionality of the configuration space is 6, which can possibly be reduced to 5.

- **digit seven of the simplest kind.**

A typical image looks like this:

[an image here]

Here we consider all continuous $\gamma : [0, 2] \rightarrow [0, 1]^2$ such that $\gamma|_{[0,1]}, \gamma|_{[1,2]}$ are straight line segments. Additionally, there should be some constraint on the positions of $\gamma(0), \gamma(1), \gamma(2)$ such that $\overline{\gamma(0)\gamma(1)}$ is very close to being horizontal, and $\overline{\gamma(1)\gamma(2)}$ should be roughly vertical downward.

The score function \mathfrak{s} is the same.

The dimensionality of the configuration space is 6, which can actually be reduced to 5.

- **digit zero.**

We consider smooth curves $\gamma : [0, L] \rightarrow [0, 1]^2$ with arc length parametrization such that the mean curvature is always nonnegative, i.e.

$$\|\gamma'(t)\| \equiv 1 \tag{6}$$

and

$$\gamma''(t) \times \gamma'(t) \geq 0 \tag{7}$$

Additionally, we require that $\gamma(0)$ is very close to $\gamma(L)$.

We should also require that γ is nondegenerate, which can be characterized by isoperimetric inequality.

All these γ form Γ_0 .

And we still use the same score function \mathfrak{s} .

- **general case.** Fix a graph $G = (V, E)$. Give it a natural topological structure and identify each e with $[0, 1]$. For each $e \in E$, we give assign a σ_e which is one of the following classes of curves:

- nonconvex but not straight
- nonconcave but not straight
- straight.

We define the total space as

$$\Omega_G = \{\gamma \in M(G, [0, 1]^2) : \forall e \in E, \gamma|_e \in \sigma_e\}. \tag{8}$$

Then the configuration space Γ_G is a subset of Ω_G such that it is given by a boolean function s in the sense that

$$1_{\Gamma_G}(\gamma) = s((\gamma(v))_{v \in V}, ((\gamma|_e'(0), \gamma|_e'(1), \text{dist}(\gamma|_e, \overline{\gamma|_e(0)\gamma|_e(1)})))_{e \in E}) \tag{9}$$

7 Conclusion

8 Future Work