Personal Coverage Notes on Machine Learning

Authored by Dun-Ming Huang; Department of Computer Science, UC Berkeley

Preface(s)

0.1 Preface

In this personal coverage note, I try to record some thought processes and learning results I obtained from watching lectures and videos about numerous machine learning literature or applications. For now, the personal coverage note covers the following sections:

0.1.1 COMPSCI 189: Introduction to Machine Learning

This part covers the basic concepts of machine learning, including supervised learning, unsupervised learning, and focuses specifically on statistical learning techniques.

0.1.2 COMPSCI 285: Deep Reinforcement Learning

This part covers fundamental knowledge regarding deep reinforcement learning, sampled from Sergey Levine's Deep Reinforcement Learning lecture.

0.1.3 COMPSCI 294-158: Deep Unsupervised Learning

This part covers fundamental knowledge regarding deep unsupervised learning, sampled from Pieter Abbeel's Deep Unsupervised Learning lecture.

0.1.4 Mu Li's Videos on YouTube

This part covers the notes I took from watching Mu Li's videos on YouTube, which covers a wide range of topics in machine learning.

Contents

Part I

COMPSCI 189: Introduction to Machine Learning

Chapter 1

Fundamentals of Machine Learning

In this chapter, we cover the introductory lecture of COMPSCI 189.

Learning Goals:

- Understand the fundamental workings of machine learning.
- Learn about classification as an example machine learning task.

Machine learning is a popular topic over the recent centuries. It is a subset of artificial intelligence that focuses on the development of algorithms that allow computers to learn from and make predictions based on data. The study of machine learning per se is a long journey, even disregarding the participation of research activities. In this section of the note, we concentrate on statistical learning, which involve fundamental techniques of machine learning that are popularized before the current pulvinar of deep learning. Deep learning techniques will be addressed in later sections of the entire note.

1.1 The Framework of Machine Learning

Machine learning is the use and development of computer systems that can learn without explicit instructions; that is, they learn a specific pattern of the provided data via statistical measures, in an autonomous and algorithmic manner. The learning process is largely valuable on the ability of machine learning algorithms to draw insights, or **inferences**, upon the provided training data. Fundamentally, statistical learning is all about finding patterns in data, and using them to make predictions. An abstraction of this will be issued in Lecture 5 of the section.

Machine learning is a data-driven approach. As mentioned before, all that a machine learning can learn from is what the distribution of a provided training data provides. This is an important insight in the future. Just as how humans cannot learn what a cat is if they have never learned anything about a cat, a machine cannot learn about cat if the data we provide to its algorithm never describes what a cat is. In summary, what an algorithm learns is largely dependent on the data we provide to it.

1.2 Classification as Example Machine Learning Task

Classification, as you may have learned in highschool biology, is the process of categorizing things based on their properties. In machine learning, classification is a task that involves predicting the category of a given data point. For example, provided a picture that may entail a cat or a dog, a machine learning algorithm would be asked to classify it as either a cat or a dog. This is convenient in that humans do not have to process this judgment manually, and can instead automate this task with a fairly accurate algorithm.

How do we really decide if a given data point is a cat or a dog? For example, suppose the datapoint I am provided is an image, how do I transform this image into a decision's label (a cat, versus a dog)? In classification, we usually use numbers to denote the label of a category (hereby we call it a "class"). A classifier h, therefore, is a function that is provided a datapoint $\vec{x}$ and outputs a numeric label for the representing class:

$$h(\vec{x}) = \begin{cases} 1 & \text{if the algorithm considers } \vec{x} \text{ is a cat} \\ 0 & \text{if the algorithm considers } \vec{x} \text{ is a dog} \end{cases}$$

The question now comes down to:

1. $\vec{x}$: How is the colorful image we see in human eyes represented as a vector?

2. "if the algorithm considers $\vec{x}$ as a something": how is this rule implemented programatically?

For question (1), the image's pixels can be converted into color-representing numeric values, then flattened into a vector based on the spatial ordering of pixels. For question (2), the algorithm learns a function h that outputs the above mapping. The takeaways of above questions are as follows:

1. The machine learning algorithm receives data in numeric form, such as a list of numbers (vectors), but not qualitatively.

2. The machine learning algorithm learns a function that is tailored to our need.

1.3 The Train-Validate-Test Framework

In machine learning, an algorithm usually follows the framework of train, validate, test. These aspects of the paradigm are summarized as follows.

1.3.1 Aspects of the T V T Framework

Training a Model. Recall that any machine learning algorithm produces a function h, which we also call a model, by having the algorithm detect patterns in our datapoints $\vec{x}$'s. The act of learning an appropriate function h that behaves well on our given datapoints is called **training** a model. That is, we are training a machine learning model on a provided dataset, and the resulting model should learn a function h that accurately predicts the class label (dog vs. cat) of a given datapoint $\vec{x}$ (an image). In this phase, models are provided a labeled dataset; that is, a set of images that are labeled either as a cat or a dog. Such dataset we use to train the model is otherwise known as a **training set**. Usually, we continue with the training phase until the algorithm's model has reached a satisfying accuracy for the training set.

Validating a Model. We have trained a model with images of cats and dogs, and now it's time to evaluate the model. More precisely, it's time to evaluate the model on datapoints it has not seen yet. After all, when a model is deployed into the real world, it is expected for the model to be able to classify cats and dogs from pictures immediately before us, mostly unseen to anyone, rather than just known images that are already labeled in a dataset. The dataset that we use to validate the model, entirely unseen during the training phase, is known as the **validation set**. We will stay in this phase until our model has reached a satisfying accuracy for the validation set.

Testing a Model. At last, we evaluate our model again using another unseen dataset, called the **testing set**. The testing set is used to evaluate the model's performance on a dataset that is entirely unseen during the training and validation phases. This is the final phase of the train-validate-test framework, and the model's performance on the testing set is the final metric of the model's performance.

1.3.2 Justification: Overfitting and Underfitting

Hi

1.3.3 A Summary of Questions up to This Point

Hi

Chapter 2

Linear Classifiers

Chapter Description.

2.1 section title

Section.

Theorem 2.1.1. Tested Theorem

I am the bone of my sword.

Definition 2.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 2.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 3

Gradient Descent

Chapter Description.

3.1 section title

Section.

Theorem 3.1.1. Tested Theorem

I am the bone of my sword.

Definition 3.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 3.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 4

Support Vector Machine

Chapter Description.

4.1 section title

Section.

Theorem 4.1.1. Tested Theorem

I am the bone of my sword.

Definition 4.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 4.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 5

It Is All About The Layers of Abstraction

Chapter Description.

5.1 section title

Section.

Theorem 5.1.1. Tested Theorem

I am the bone of my sword.

Definition 5.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 5.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 6

Decision Theory, Bayesian Decision Rule

Chapter Description.

6.1 section title

Section.

Theorem 6.1.1. Tested Theorem

I am the bone of my sword.

Definition 6.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 6.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 7

Gaussian Discriminant Analysis and Maximum Likelihood Estimation

Chapter Description.

7.1 section title

Section.

Theorem 7.1.1. Tested Theorem

I am the bone of my sword.

Definition 7.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 7.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 8

Eigendecomposition of Symmetric Matrices

Chapter Description.

8.1 section title

Section.

Theorem 8.1.1. Tested Theorem

I am the bone of my sword.

Definition 8.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 8.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 9

Abstractions of a Regression Problem

Chapter Description.

9.1 section title

Section.

Theorem 9.1.1. Tested Theorem

I am the bone of my sword.

Definition 9.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 9.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 10

Newton's Method and Logistic Regression

Chapter Description.

10.1 section title

Section.

Theorem 10.1.1. Tested Theorem

I am the bone of my sword.

Definition 10.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 10.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 11

Statistical Justifications for Regressions

Chapter Description.

11.1 section title

Section.

Theorem 11.1.1. Tested Theorem

I am the bone of my sword.

Definition 11.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 11.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 12

Ridge Regression and Regularization

Chapter Description.

12.1 section title

Section.

Theorem 12.1.1. Tested Theorem

I am the bone of my sword.

Definition 12.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 12.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 13

Decision Trees

Chapter Description.

13.1 section title

Section.

Theorem 13.1.1. Tested Theorem

I am the bone of my sword.

Definition 13.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 13.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 14

The Kernel Trick

Chapter Description.

14.1 section title

Section.

Theorem 14.1.1. Tested Theorem

I am the bone of my sword.

Definition 14.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 14.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 15

Introduction to Neural Networks

Chapter Description.

15.1 section title

Section.

Theorem 15.1.1. Tested Theorem

I am the bone of my sword.

Definition 15.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 15.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 16

Tricks and Heuristics for Neural Networks

Chapter Description.

16.1 section title

Section.

Theorem 16.1.1. Tested Theorem

I am the bone of my sword.

Definition 16.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 16.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 17

Convolutional Neural Network

Chapter Description.

17.1 section title

Section.

Theorem 17.1.1. Tested Theorem

I am the bone of my sword.

Definition 17.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 17.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 18

Unsupervised Learning: PCA

Chapter Description.

18.1 section title

Section.

Theorem 18.1.1. Tested Theorem

I am the bone of my sword.

Definition 18.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 18.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 19

Unsupervised Learning: Clustering Algorithms

Chapter Description.

19.1 section title

Section.

Theorem 19.1.1. Tested Theorem

I am the bone of my sword.

Definition 19.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 19.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 20

Unsupervised Learning: Clustering Algorithms

Chapter Description.

20.1 section title

Section.

Theorem 20.1.1. Tested Theorem

I am the bone of my sword.

Definition 20.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 20.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 21

Geometry of High-Dimensional Space

Chapter Description.

21.1 section title

Section.

Theorem 21.1.1. Tested Theorem

I am the bone of my sword.

Definition 21.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 21.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 22

Learning Theory

Chapter Description.

22.1 section title

Section.

Theorem 22.1.1. Tested Theorem

I am the bone of my sword.

Definition 22.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 22.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 23

AdaBoost

Chapter Description.

23.1 section title

Section.

Theorem 23.1.1. Tested Theorem

I am the bone of my sword.

Definition 23.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 23.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 24

k-Nearest Neighbor Approaches

Chapter Description.

24.1 section title

Section.

Theorem 24.1.1. Tested Theorem

I am the bone of my sword.

Definition 24.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 24.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Part II

COMPSCI 285: Deep Reinforcement Learning

Chapter 25

Introduction to Deep Reinforcement Learning

Chapter Description.

25.1 Motivation to Reinforcement Learning

Let us begin a motivating problem: how can we let a robot hand pick up something?

In classical robotics, the problemsolving process is to: (1) Define the problem in modeling perspectives, (2) Model the problem using mathematical equations, and (3) Solve the problem via a designed algorithm. However, as we accumulate technological knowledge, now we have a second option, which is to set it up as a machine learning problem. With the knowledge we have currently learned in statistical learning, we are inclined to use supervised learning; that is, provided some data of the robot and the environment, we train some model that can provide the robot an action to comply with. However, this approach is not well-informed from human experience, and the crafting of such data is still difficult. So, instead, we follow the line of thought of letting robots earn their own experiences via trial and error. This approach develops into a **reinforcement learning** setting.

In a reinforcement learning seting, robots collect examples of their own behavior, and label its success (significance as well) based on a state-derived reward signal (function). The robot then learns to maximize the reward signal by adjusting its behavior. Eventually, we obtain a **policy** (a function $\pi : \mathcal{S} \to \mathcal{A}$ that provides an action in response to a seen state) that the robot can follow to pick an object up.

So, reinforcement learning is really an experience-collecting framework: it allows for a freedom of trial, a disregard for a pre-existing dataset (although in most situations we will still have one), and inherits from machine learning approaches the waive of need to manually design solutions for each specific problem. Like statistical learning, reinforcement learning is also a massively scaled process of density estimations for underlying distributions (say, $p_\theta(x)$, or $p_\theta(y|x)$) of the training data. However, reinforcement learning is different in that it enables learning via agent-environment interactions, which provides a new source of information; and, it allows for new applications like evolutionary algorithms, controls, and optimizations. Reinforcement learning is mainly an approach for a design of behavior that does not require human intervention. These behavior are impressive because it is unthought of, as well as because of its delicate mimicry for human results (say, artistic outcomes).

25.2 Introduction to Reinforcement Learning

Reinforcement learning is both a mathematical ofmralism for learning-based decision making, and an approach for learning decision making and control from experience.

In supervised learning, the framework follows as a provided dataset $\mathcal{D} = \{(x_i, y_i)\}$ that contributes to the learning of a function $f : \mathcal{X} \to \mathcal{Y}$, which itself is the ultimate fruit of a supervised learning paradigm. Recall that supervised learning makes several assumptions regarding its paradigm. One, the data provided to us are i.i.d. samples from an underlying distribution. Two, we have known ground truth outputs in training.

In reinforcement learning, however, we ignore both of these assumptions. One, data is not i.i.d., becuase previous outputs influence future inputs (in a Markovian fashion). Two, ground truth answer is not known, and we are only provided a reward signal that notifies us whetehr a demonstration from the agent is successful or not. Reinforcement learning is therefore ran on reward-labeled data, rather than ground-truth-labeled data.

To summarize the paradigm of reinforcement learning, it is comprised of the following aspects:

1. An **agent** that interacts with the world to achieve a specific task.
2. An **environment** that the agent interacts with. This can be a Minecraft flat world for an agent that tries to learn walking on. This environment can be both in-real-life and simulated. In most occassions, it is simulated.
3. The input of learning is a **state** s_t that represents the current situation of the agent.
4. The output of learning, generally, is a **action** a_t that the agent takes in response to the state.
5. A **reward** r_t that the agent receives after taking an action.
6. A **policy** π that the agent follows to take an action.

The data we receive for reinforcement learning is therefore a sequence of (s_t, a_t, r_t) tuples that the agent collects from the environment as it interacts with it.

25.3 Motivation Towards Deep Reinforcement Learning

The fusion of data-driven AI and reinforcement learning provides us a complementary approach. In data-driven AI (deep learning), while we extract valuable inferences about the real world from data, we don't actively attempt to perform better than the data. Meanwhile, in reinforcement learning, while we extract emergent behavior to do better than existing data, we are not prepared with a way to extract inferences regarding the environment, and are not provided a means of using data at scale. That is, Data-Driven AI is about using data, while reinforcement learning is about using optimizations. Therefore, deep reinforcement learning is expected to excel at both learning and searching: learning from data and searching for (discovering) better ways to interact with the environment provided the data.

Noted, we have deep neural network architectures that extracts inference well, and RL algorithms that are compatible with these approaches. However, at the current stage, learning-based control in truly real-world settings remains a major open problem. We will discuss these topics at lengths with later sections of the note.

In the current state, we face the following open challenges:

1. We don't yet have amazing methods that both use data and Reinforcement Learning
2. Humans can learn incredibly quickly, but deep RL methods are usually slow, even in simulators.
3. Humans can reuse past knowledge, but domain transfer is a problem to deep reinforcement learning.
4. The role of prediction and design of reward functions are still not very clear in reinforecment learning.

Chapter 26

Supervised Learning of Behaviors

Chapter Description.

26.1 Terminology and Notation in DRL

In this section, we will cover several temrinologies and notations that the community uses regarding learning situations.

In our paradigm, we concern an input, a system that processes the input, and the output. Decision-making problems consider these as respectively observations o_t, policy $\pi_\theta(a_t|o_t)$, and actions a_t. These symbols are subscripted by time because the context of a decision-making problem is usually a chronological sequence of events. Note that, the production of next observation, o_{t+1}, is based on the impacted state o_t upon the transpiration of a_t. Note that the format of a_t, for example, is not limited to a vector; it can also be a continuous distribution, as we would sometimes like to sample actions to take rather than using an almost deterministic policy. The policy π_θ is parameterized by θ, which is a vector of parameters that the policy uses to make decisions (that is, to provide action provided observation). They, therefore, assign the probability to all possible actions provided a specific state.

We also introduce the notion of a state, s_t, which is a partial observation. This notion is introduced by the fact that the entirety of an environment is not always observable (and in most situations, we do not observe the entirety of an environment). Therefore, realistically, the policy we learn is $\pi_\theta(a_t|s_t)$, which we call a partially observed policy. A very appropriate analogy is perhaps our visual-neural system: our eyes guide our decisions based on what objects are posed in our environment, particularly what is in front of our eyes. But, we do not gain information from our eyes regarding what is behind us, simply because the environment we are situated in is partially observable: the sensors we have (eyes) simply do not detect what is behind. Therefore, in reinforcement learning paradigms, we work with state-action pairs rather than observation-action pairs.

We develop this into what we call a Markov Decision Paradigm. In this paradigm, we assume that the state s_t is sufficient to make decisions, and that the future state s_{t+1} is independent of the past states $s_{t-1}, s_{t-2}, \ldots$ given the current state s_t and the action a_t. The paradigm per se contains Markov-ness; that is, s_t only depends on s_{t-1}, and the connection of states is only nonzero for $p(s_{t+1}|s_t, a_t)$. The involvement of only states and actions is a reflection of the partial observability we suffer in environments, which produces a Partially Observable Markov Decision Process (POMDP).

Note: in some literature, states will be issued as x_t, and action u_t instead, due to the involvement of background in contorl theory from some influential figures.

26.2 Imitation Learning

Imitation learning concerns the learning of a policy π_θ from a dataset of expert demonstrations. **Behavioral cloning**, an approach of imitation learning, is the idea that, via supervised learning, we learn a policy that regresses an underlying function $f : \mathcal{S} \to \mathcal{A}$ that maps the expert-induced states to experr-performed actions. These expert demonstrations are, as assumed, provided to the algorithm. However, behavioral cloning is usually not a good solution to general problems. Therefore, we expect the agent to practically clone the behavior (state-action reaction) of the expert.

The reason why is because, although expert demonstrations provide us many trajectories, once we receive a state outside of the provided trajectories, the actions our agent provides are not well-defined in nature. It is moreso that, once the agent is exposed to an unseen state, the action it provides via its policy does not guarantee a continued cloning of the expert-demonstrated trajectory. This is because the nature of a trajectory destructs the i.i.d. assumption of supervised learning; that is, because of temporal dependencies, we encounter problems in behavioral cloning that does not appear in conventional supervised learning problems. (A spam classifier generalizes well to unseen emails, but a behavioral cloning agent does not generalize well to unseen states). We can propose ad-hoc solutions, such as data augmentation that increases familarity of the agent to unseen states, but these solutions are not always guaranteed to work.

To make behavioral cloning work, we must make modifications to the existing paradigm. Let us begin with lessons of the story:

1. Different from conventional supervised studying problems, imitation learning via behavioral cloning is not guaranteed to work.
2. To work against it, we can use data augmentation as well as modify data collection methods.
3. An exotic solution, like a multi-task learning formulation, may help to generalize to unseen states and perform good imitation learning.
4. The most intuitive approach is perhaps modifying the algorithm along which we do imitation learning.

26.3 Theoretical Analysis of Failure in Behavioral Cloning

The main culprit of behavioral cloning's failure is the dsitributional shift problem. Suppose that we have some policy $\pi_\theta(a_t|o_t)$ trained on a dataset of expert demonstrations. Here, let us say that the distribution provided by expert dataset is $p_{data}(o(t))$, but the distribution of observation the policy faces is $p_{\pi_\theta}(o_t)$. Note that, since our policy is trained under $p_{data}o(t)$, the objective of that training is $\max_\theta \mathbb{E}_{o_t \sim p_{data}(o(t))} \log \pi_\theta(a_t|o_t)$. However, the policy is evaluated under $p_{\pi_\theta}(o_t)$, which is not the same as the training distribution. This problem is otherwise known as **distributional shift**, and this occurs due to the policy's own deviation from the training distribution.

The lesson of such problem is that we should perhaps define more precisely what we want to define as "well-learned". A policy that is good would perhaps not be a point-estimate for actions, since it can easily lead to deviations in policy behavior. Perhaps we may define a cost otherwise. Suppose that π^* is the optimal policy for us to clone:

$$c(s_t, a_t) = \begin{cases} 0 & \text{if } a_t = \pi^*(s_t) \\ 1 & \text{otherwise} \end{cases}$$

Now, then, our training objective becomes:

$$\min \mathbb{E}_{s_t \sim p_{\pi_\theta}(s_t)}[c(s_t, \pi_\theta(s_t))]$$

Assume that we have an upper bound of the policy mistake, $\pi_\theta(a \neq \pi * (s)|s) \leq \epsilon$. Then, we observe an incurred cost of:

$$\begin{aligned} &\mathbb{E}[\sum_t c(s_t, \pi_\theta(s_t))] \\ &\leq \epsilon T + (1-\epsilon)(\epsilon(T-1) + (1-\epsilon)(\dots)) \in 0(\epsilon T^2) \end{aligned}$$

Therefore, the error of behavioral cloning is quadratic to the length of its trajectory (T).

It turns out that an analysis composed by Ross et al. (2011) shows that the error of behavioral cloning is quadratic to the length of the trajectory. That is, imitation learning is prone to failure at any timestep, and lacks a means of recovery from a policy's small failures.

26.4 Addressing Problem of Imitation Learning

To address the problem of imitation learning, we can consider the following approaches:

1. Data Augmentation and collection
2. Powerful models that make very few mistakes
3. Multi-task learning formulation of imitation learning
4. Changing the algorithm of use (where we discuss DAgger)

Data Augmentation and Collection. Behavioral cloning is difficult because the model doesn't generalize well to mistakes. What is we involve data regarding mistakes instead? If the dataset involves mistakes, due to additional steps in data collection and augmentation process, although the training set will be diluted, we can now access corrections in our BC paradigm.

Powerful Models. Failures from fitting the expert can stem from several reasons. First, the expert's behavior is non-Markovian. While the policy we train is Markovian-ly conditioned, the expert's behavior are not formulated on a Markovian approach. That is, provided exposure to states $s_t = s_{t'}$, it is likely that the action elicited from these states are different. Therefore, human demonstrators post a very unnatural circumstance for a Markovian policy. Perhaps one remedy is to use the entire history of a trajectory, and a sequence model is capable of processing it as a temporal sequence of frames. However, the exploitation of entire history may still work poorly, simply because including the history may still be harmed by incomplete information, which can lead to causal confusion. Second, the expert's behavior may be multimodal. That is, the expert may have multiple ways of solving a problem, and the policy we train may not be able to capture all of these modes. This is specifically unhelpful for a continuous distribution of actions, which rely on the use of mean and variance that is difficultly characterizing for bimodal distributions. Although we can instead choose expressive continuous distributions for actions (namely, use other classes of distributions), or use an autoregressive discretization with high-dimensional action spaces, they do pose higher computational costs to this procedure.

Multi-task Learning Formulation. Training a policy that has one sole destination of trajectory can be difficult, but perhaps a multi-task learning formulation that attempts to let policies reach multiple different destinations. This approach can be summarized as "goal-conditioned behavioral cloning", which can provide more opportunities to learn corrections despite distributional shift; that is, to maximize $\log \pi_\theta(a_t^i | s_t^i, g = s_T^i)$. However, this approach actually introduces a secondary source for distributional shift, making the approach theoretically worse.

Changing the Algorithm of Use: DAgger. The idea of DAgger, Dataset Aggregation, is to make $p_{data}(o_t) = p_{\pi_\theta}(o_t)$. That is, we collect training data from $p_{\pi_\theta}(o_t)$ by running the policy $\pi_\theta(a_t|o_t)$ in the environment. Then, we aggregate the dataset of expert demonstrations and the dataset of the policy's own data, and train the policy on the aggregated dataset. The algorithm is as follows:

1. Collect a dataset of expert demonstrations $\mathcal{D} = \{(o_t^i, a_t^i)\}_{i=1}^N$.
2. For $k = 1, 2, \dots, K$:
   (a) Train the policy π_θ on $\mathcal{D}$.
   (b) Collect a dataset of the policy's own data $\mathcal{D}_{\pi_\theta} = \{(o_t^i, \pi_\theta(o_t^i))\}_{i=1}^N$.
   (c) Aggregate the datasets $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\pi_\theta}$.

26.5 A Hint at the Need of Reward and Cost Signals

Deep learning works best when data is plentiful, but humans are finite sources of data (that is, humans cannot provide all data and label all data). Therefore, to resolve the disruptive demand of large ground truths, we expect our reinforcement learning algorithm to learn autonomously. To enable algorithms to evaluate their experiences, and exceeding their own performances, we offer a reward signal called **reward function**, $r(s, a)$, which provides a numeric evaluation for an observed pair of state and action. For example, for imitation learning, we can propose the reward function $r(s, a) = \log p(a = \pi^*(s)|s)$.

Chapter 27

Introduction to Reinforcement Learning

Chapter Description.

27.1 section title

Section.

Theorem 27.1.1. Tested Theorem

I am the bone of my sword.

Definition 27.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 27.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 28

Policy Gradients

Chapter Description.

28.1 section title

Section.

Theorem 28.1.1. Tested Theorem

I am the bone of my sword.

Definition 28.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 28.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 29

Actor-Critic Algorithms

Chapter Description.

29.1 section title

Section.

Theorem 29.1.1. Tested Theorem

I am the bone of my sword.

Definition 29.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 29.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 30

Value Function Methods

Chapter Description.

30.1 section title

Section.

Theorem 30.1.1. Tested Theorem

I am the bone of my sword.

Definition 30.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 30.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 31

Deep RL with Q-Functions

Chapter Description.

31.1 section title

Section.

Theorem 31.1.1. Tested Theorem

I am the bone of my sword.

Definition 31.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 31.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 32

Advanced Policy Gradients

Chapter Description.

32.1 section title

Section.

Theorem 32.1.1. Tested Theorem

I am the bone of my sword.

Definition 32.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 32.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 33

Optimal Control and Planning

Chapter Description.

33.1 section title

Section.

Theorem 33.1.1. Tested Theorem

I am the bone of my sword.

Definition 33.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 33.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 34

Model-Based Reinforcement Learning

Chapter Description.

34.1 section title

Section.

Theorem 34.1.1. Tested Theorem

I am the bone of my sword.

Definition 34.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 34.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 35

Model-Based Policy Learning

Chapter Description.

35.1 section title

Section.

Theorem 35.1.1. Tested Theorem

I am the bone of my sword.

Definition 35.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 35.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 36

Exploration (Part 1)

Chapter Description.

36.1 section title

Section.

Theorem 36.1.1. Tested Theorem

I am the bone of my sword.

Definition 36.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 36.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 37

Exploration (Part 2)

Chapter Description.

37.1 section title

Section.

Theorem 37.1.1. Tested Theorem

I am the bone of my sword.

Definition 37.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 37.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 38

Offline Reinforcement Learning (Part 1)

Chapter Description.

38.1 section title

Section.

Theorem 38.1.1. Tested Theorem

I am the bone of my sword.

Definition 38.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 38.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 39

Offline Reinforcement Learning (Part 2)

Chapter Description.

39.1 section title

Section.

Theorem 39.1.1. Tested Theorem

I am the bone of my sword.

Definition 39.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 39.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 40

Reinforcement Learning Theory

Chapter Description.

40.1 section title

Section.

Theorem 40.1.1. Tested Theorem

I am the bone of my sword.

Definition 40.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 40.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 41

Variational Inference and Generative Models

Chapter Description.

41.1 section title

Section.

Theorem 41.1.1. Tested Theorem

I am the bone of my sword.

Definition 41.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 41.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 42

Connection between Inference and Control

Chapter Description.

42.1 section title

Section.

Theorem 42.1.1. Tested Theorem

I am the bone of my sword.

Definition 42.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 42.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 43

Inverse Reinforcement Learning

Chapter Description.

43.1 section title

Section.

Theorem 43.1.1. Tested Theorem

I am the bone of my sword.

Definition 43.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 43.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 44

Reinforcement Learning with Sequence Models

Chapter Description.

44.1 section title

Section.

Theorem 44.1.1. Tested Theorem

I am the bone of my sword.

Definition 44.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 44.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 45

Meta-Learning and Transfer Learning

Chapter Description.

45.1 section title

Section.

Theorem 45.1.1. Tested Theorem

I am the bone of my sword.

Definition 45.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 45.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 46

Challenges and Open Problems

Chapter Description.

46.1 section title

Section.

Theorem 46.1.1. Tested Theorem

I am the bone of my sword.

Definition 46.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 46.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Part III

COMPSCI 294-158: Deep Unsupervised Learning

Chapter 47

Introduction to Deep Unsupervised Learning

Chapter Description.

47.1 section title

Section.

Theorem 47.1.1. Tested Theorem

I am the bone of my sword.

Definition 47.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 47.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 48

Autoregressive Models

Chapter Description.

48.1 section title

Section.

Theorem 48.1.1. Tested Theorem

I am the bone of my sword.

Definition 48.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 48.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 49

Flow Models

Chapter Description.

49.1 section title

Section.

Theorem 49.1.1. Tested Theorem

I am the bone of my sword.

Definition 49.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 49.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 50

Latent Variable Models

Chapter Description.

50.1 section title

Section.

Theorem 50.1.1. Tested Theorem

I am the bone of my sword.

Definition 50.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 50.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 51

GAN and Implicit Models

Chapter Description.

51.1 section title

Section.

Theorem 51.1.1. Tested Theorem

I am the bone of my sword.

Definition 51.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 51.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 52

Diffusion Models

Chapter Description.

52.1 section title

Section.

Theorem 52.1.1. Tested Theorem

I am the bone of my sword.

Definition 52.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 52.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 53

Self-Supervised Learning, Non-Generative Representation Learning

Chapter Description.

53.1 section title

Section.

Theorem 53.1.1. Tested Theorem

I am the bone of my sword.

Definition 53.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 53.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 54

Leage Language Models

Chapter Description.

54.1 section title

Section.

Theorem 54.1.1. Tested Theorem

I am the bone of my sword.

Definition 54.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 54.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 55

Video Generation

Chapter Description.

55.1 section title

Section.

Theorem 55.1.1. Tested Theorem

I am the bone of my sword.

Definition 55.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 55.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 56

Semi-Supervised Learning and Unsupervised Distribution Alignment

Chapter Description.

56.1 section title

Section.

Theorem 56.1.1. Tested Theorem

I am the bone of my sword.

Definition 56.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 56.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 57

Compression

Chapter Description.

57.1 section title

Section.

Theorem 57.1.1. Tested Theorem

I am the bone of my sword.

Definition 57.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 57.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 58

Multimodal Models

Chapter Description.

58.1 section title

Section.

Theorem 58.1.1. Tested Theorem

I am the bone of my sword.

Definition 58.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 58.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 59

Parallelization

Chapter Description.

59.1 section title

Section.

Theorem 59.1.1. Tested Theorem

I am the bone of my sword.

Definition 59.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 59.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 60

AI for Science (Gues Instructor)

Chapter Description.

60.1 section title

Section.

Theorem 60.1.1. Tested Theorem

I am the bone of my sword.

Definition 60.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 60.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Chapter 61

Neural Radiance Fields (Guest Instructor)

Chapter Description.

61.1 section title

Section.

Theorem 61.1.1. Tested Theorem

I am the bone of my sword.

Definition 61.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 61.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Part IV

Mu Li's Videos on YouTube

Chapter 62

Note Name

Chapter Description.

62.1 section title

Section.

Theorem 62.1.1. Tested Theorem

I am the bone of my sword.

Definition 62.1.1. Tested Theorem

Steel is my body and fire is my blood.

Example Question 62.1.1: Rules?

I have created over a thousand blades.

Unknown to death, nor known to life.

Revision Log

- August 22nd: Note is created