Machine Learning **Spring 2021**

Lecture 1: Introduction to Machine Learning

Lecturer: Liwei Wang *Scribe: Junyi Guo, Zimai Guo, Kexing Zhou, Hongzhe Li*

1.1 Machine Learning

The notation "Machine Learning" is usually assisociated with Deep Learning and Big Data, but it is of much longer history. Machine Learning can be traced back to the 1960s, and it has a formal and clear declearation.

Definition 1.1 (Machine Learning) *Machine Learning means learning from experience or learning from data.*

1.1.1 History of Machine Learning

1. **Vapnic and Cherononkis** proposed VC theory from 1969 to 1971.
2. **Leslie Valient** developed the PAC (probabilistic approximately correct) theory, which won him the 2010 Turing Award.
3. **R.Schapire and Y.Freund** proposed boosting algorithm in 1990s, and received the Godel Price.
4. **Vapnik** brought out the SVM (support Vector Machine) algorithm.
5. **Pearl** put forward Graphic Models in 1980s and won the 2011 Turing Award.
6. **Yoshua Bengio, Geoffrey Hinton and Yann LeCun** Opened up a new field of deep learning.

1.1.2 Tasks and Brunches in Machine Learning

With decades of development, machine learning become a huge topic nowadays. Main brunches are listed below.

Supervised Learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs [1]. Many common tasks are included in supervised learning:

- *classification*: Assign a category to each item.
- *regression*: Predict a real value for each item.
- *ranking*: Order items according to some criterion.

Unsupervised Learning is a type of algorithm that learns patterns from untagged data. The philosophy of unsupervised learning is to discover patterns in the data itself. Famous unsupervised learning models include Boltzmann Machine, Autoencoder and VAE.

Reinforcement Learning is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward[1]. Reinforcement learning can be called "learning to control" and is an important approach in robotic systems and autonomous driving.

Online Machine Learning is a method of machine learning in which data becomes available in a sequential order and is used to update the best predictor for future data at each step. Online machine learning can form a closed-loop system that is common in commercial applications such as recommendation systems.

1.2 Recommended Books

1.2.1 Learning Theory

1. *Foundations of Machine Learning*
2. *Understanding Machine Learning from Theory to Algorithms*

1.2.2 Reinforcement Learning

1. *Reinforcement Learning Online Course*
2. **Not recommend to read** *Reinforcement Learning An Introduction*

1.2.3 Graphical Models

1. Koller's Online Course

1.2.4 Optimization

1. *Convex Optimization Algorithm and Complexity*

1.3 Formulation of Learning

The example below are based on a classification problem.

1. Collect Training data
   Symbols: $(x_1, y_1), \cdots, (x_n, y_n), x_i \in \mathcal{X}, y_i \in \mathcal{Y}$
   $\mathcal{X}$: instance space
   $\mathcal{Y}$: label space

2. Learn a Classifier
   Use algorithm $\mathcal{A}, \mathcal{A} : S \mapsto f, f : \mathcal{X} \to \mathcal{Y}$
3. Using f on test data
   $f(x_{n+1}), f(x_{n+2}), \cdots$

How can we ensure that f behaves well in test data?

1. iid: Independent homogeneous distribution, $(x_i, y_i) \sim \mathcal{D}_{\mathcal{X}\mathcal{Y}}$

How to evaluate the performance of f

1. Training Error: $\frac{1}{n}\sum_{i=1}^{n} I[y_i = f(x_i)]$
2. Test Error(Expected Error): $\mathsf{P}_{(x,y)\sim\mathcal{D}_{(\mathcal{X}\mathcal{Y})}}(y \neq f(x))$

Suppose $\mathcal{D}_{\mathcal{X}\mathcal{Y}}$ is known, the theoretical **optimal classifier** is the Bayes classifier defined as

$$f^*(x) = \underset{y\in\mathcal{Y}}{\operatorname{argmax}} \mathbb{P}(y|x)$$

1.4 Generation

Is it the best to make training error 0%?

Consider a fitting problem. For a training set $(x_1, y_1), \cdots, (x_n, y_n)$, we can choose to use either a linear model or a n-order polynomial model to fit. Obviously, the training error for the n-order polynomial model is much smaller than that for the linear model. However, small training error does not necessarily lead to small test error, which is known as the **overfitting** phenomenon. As the parameters of the model increase, the model becomes better fitted, but generation may decrease, which is the result we do not want.

How can we choose a better model?

There is no such thing as the best model for any given problem. It is possible for any model to perform well for a particular problem. Therefore, we should choose models carefully to achieve both good problem solving and good interpretability.

1.5 Basic Inequalities

1.5.1 Markov Inequality

For random value $X \geq 0$, if $EX < +\infty$, then:

$$\mathbb{P}(X \geq k) \leq \frac{EX}{k}, \quad \forall k > 0$$

1.5.2 Chebyshev Inequality

For random value X, if EX exists, and $EX^2 < +\infty$(which also means $\text{var}(X) < +\infty$), then:

$$\mathbb{P}(|X - EX| \geq t) \leq \frac{\text{var}(X)}{t^2}, \quad \forall t > 0$$

For random value X, if $EX, EX^2, ..., EX^r$ is all known, then

$$\mathbb{P}(X \geq k) \leq \min_j \frac{EX^j}{t^j}, \quad \forall j = 1, 2, \ldots, r$$

1.5.3 Moment Generating Function

Definition 1.2 (Moment Generating Function) *For random value X, if forall $n \in \mathbb{N}$, EX^n exists, we can define Moment Generating Function of X:*

$$M(t) := E(e^{tX}) = 1 + tEX + t^2\frac{EX^2}{2!} + t^3\frac{EX^3}{3!} + ...$$

We can use moment generating function to caculate a upperbound of $\mathbb{P}(X \geq k)$

Lemma 1.3 (upperbound of tail destribution) *$\forall t > 0$, we apply Markov Inequality to $\mathbb{P}(X \geq k)$:*

$$\begin{aligned}\mathbb{P}(X \geq k) &= \mathbb{P}(e^{tX} \geq e^{tk}) \\ &\leq e^{-tk}E(e^{tX}) \\ &= e^{-tk}M(t), \ (\forall t > 0)\end{aligned}$$

then we get the upperbound:

$$\mathbb{P}(X \geq k) \leq \inf_{t>0}\{e^{-tk}M(t)\}$$

References

[1] Stuart J. Russell, Peter Norvig (2010) Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall ISBN 9780136042594.

[2] https://en.wikipedia.org/wiki/Supervised_learning

[3] https://en.wikipedia.org/wiki/Unsupervised_learning

[4] https://en.wikipedia.org/wiki/Reinforcement_learning

[5] https://en.wikipedia.org/wiki/Online_machine_learning

[6] https://www.wikiwand.com/en/Markov's_inequality

[7] https://www.wikiwand.com/en/Chebyshev's_inequality

[8] https://www.wikiwand.com/en/Moment-generating_function