Fundamentals of Learning and Information Processing

Session 11: Statistical Machine Learning (I)

Gang Li

Deakin University, Australia

2021-07-31

Table of Content

Whom am I?

- Associate Professor **Gang Li**
  - Deakin University
    - *University Thesis Examination* Committee
    - *Cyber Analytics & AI* leader, Deakin CSRI
    - Deputy Director, Deakin D2I
    - Reviewers
      - IJCAI, AAAI several times as *Area Chair*
      - PAKDD, ACML as *Senior PC*
      - KSEM, etc. several times as *PC/General Chairs*
      - DSS, JTR as *Associate Editor*
  - IEEE Technical Committee
    - Task Force on *Educational Data Mining*
      - Chair 2020-
    - *Data Mining and Big Data Analytics*
      - Vice Chair 2017-2019
    - *Enterprise Information Systems*
    - *Enterprise Architecture and Engineering*

Deakin University

Table 1: Major Rankings of Deakin University

	Benchmark	2014-2015	2015-2016	2016-2017	2017-2018
ARWU	World	301-400	201-300	201-300	201-300
	National	12-19	9-14	11-14	10-15
CWTS Leiden	World	284	325	314	226
	National	11	17	17	11
QS#	World	360	324	355	293
	National	19	17	19	17
THE	World	301-350	301-350	251-300	301-350
	National	13-15	18-19	12-18	17-21
THE<50	World	45	50	43	50
	National	6	8	8	8

https://www.australianuniversities.com.au/rankings/

Web Resources

Social Media

- : Twitter
- : Reddit
- : Linkedin

Official Websites

- : Google Scholar
- : http://www.tulip.org.au
- : https://github.com/tulip-lab

What is *Machine Learning*?

☞ *Machine Learning* is a field of study that gives computers the ability to learn without being explicitly programmed.

- using experience to gain expertise
- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E

☞ *Statistics* evolves to an independent science, which tries to make sense of observations in the real world:

- deals with analysis of the frequency of past events.
- is seeing a footprint, and guessing the animal.

Common Draw conclusions that hold for environment (or the data distribution) from which those samples are picked.

Machine Learning vs Statistics

Machine Learning

- Data are collected randomly.
- ML aims to come up with the hypothesis automatically.
- ML assumes as little as possible on the distributions: "distribution-free"
- ML pays more attention to time and space complexity
- ML focuses on "finite sample bounds": given the size of available samples, it aims to figure out the degree of accuracy that a learner can expect based on such data.

Statistics

- Data collected on *purpose*.
- Human experts come up with the hypothesis, and statisticians view the samples and check the validity of the hypothesis
- Statistics works under assumption of certain prescribed models.
- Statistics pays less attention to algorithmic issues
- Statistics is interested in asymptotic behavior

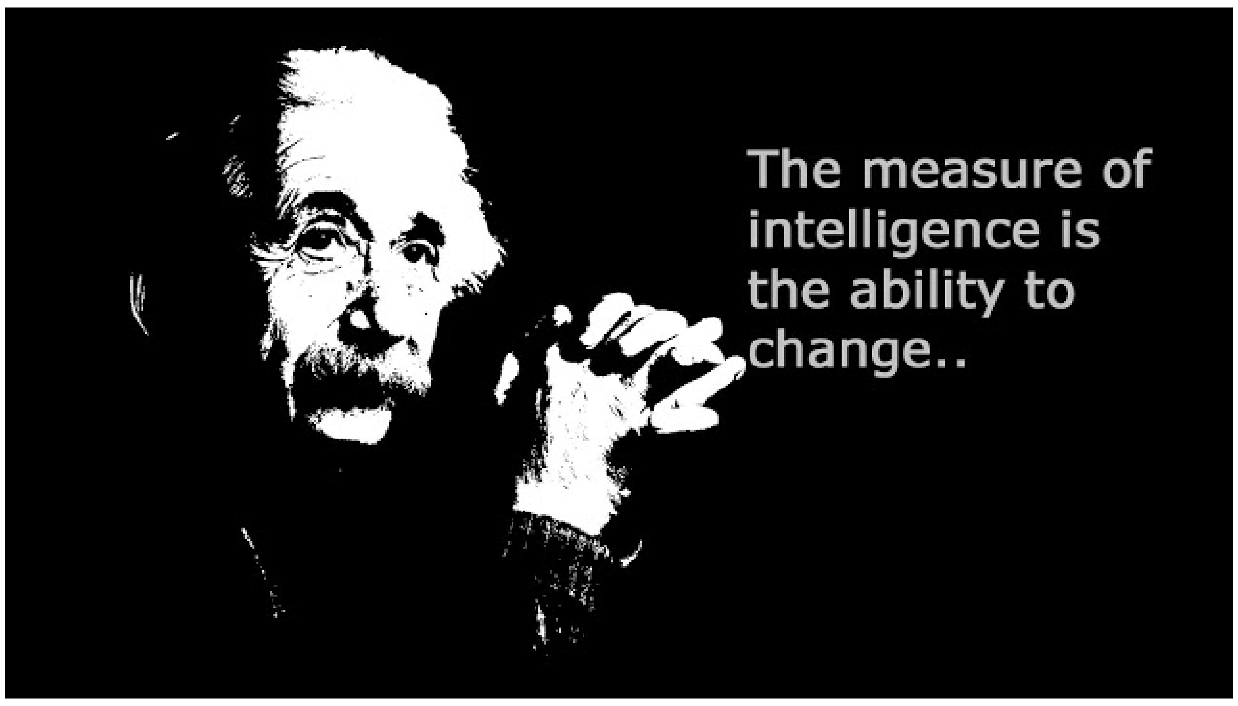
Why do we need Machine Learning?

Complexity There are tasks that are too complex to program

- Tasks performed by Humans or Animals
- Tasks beyond Human capabilities

Adaptivity Traditional program stays unchanged once written down and deployed.

- ML Programs' behavior adapts to their input data
- ML Programs are adaptive to changes in the environment they interact with



How to Learn?

There are 4 types of learners

A Sponge

- which absorbs everything
- needs a big memory and efficient retrieval mechanism

A Funnel

- which lets in at one end, and discharges at the other
- not even intelligent

A Sieve

- which forgets the essentials but retains the unimportant
- how to tell what is important?

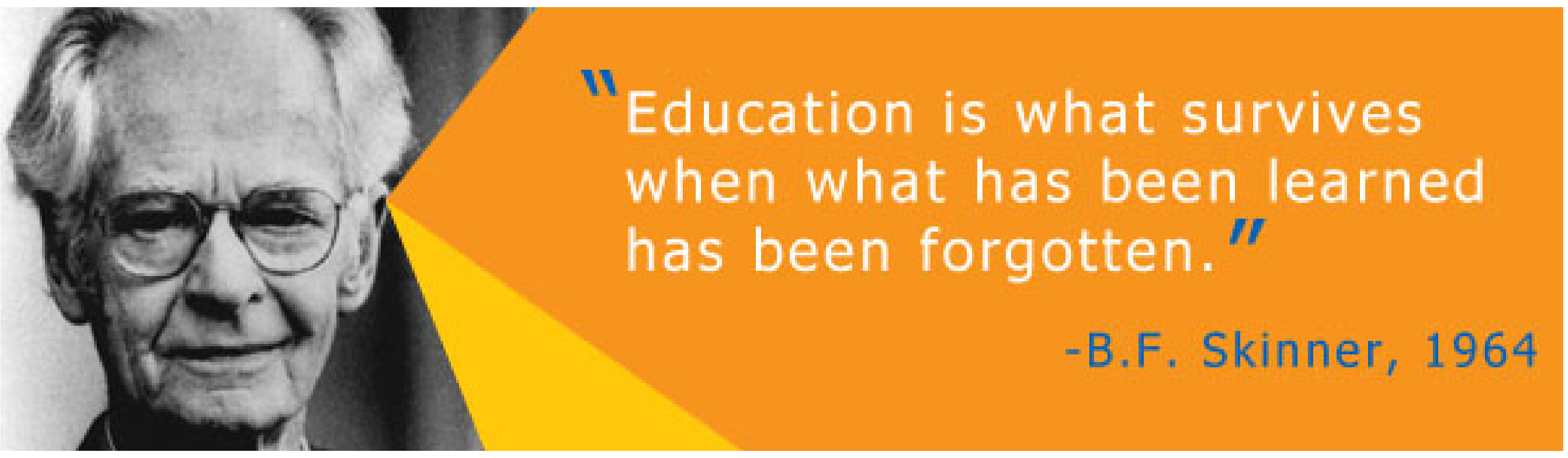
A Strainer

- which memorizes the good and rejects the worthless
- how to tell what is good and what is bad?

What is the good to memorize?

Generalization, also known as inductive reasoning or inductive inference, is an important aspect of learning system, the ability to progress from individual examples to predict unseen new examples.

- What is learnable?
- How to learn?
- How can we know that what we learned is true?



Pigeon Training

- Pigeons Turn
- Pigeons Superstition

Behaviorism and Superstitious

- Superstitious Pigeons
- Behaviorism and Your Superstitious Beliefs

Black-box Approach

- James Randi tests crystal power and applied Kinesiology

Machine Learning Types

Aim Training data $\overset{learning}{\Longrightarrow}$ Model (Hypothesis, etc.) $\overset{prediction}{\Longrightarrow}$ Unknown data
Generalization Model should fit unknown data well, not training data!

There are four common parameters along which learning paradigms can be classified

Supervised versus Unsupervised

- based on training data are labelled or not
- semisupervised learning, reinforcement learning etc.

Active versus Passive

- Active learner interacts with the environment at training time, by posing queries or performing experiments.
- Passive learner only observes the information provided by the environment without influencing or directing it.

Helpfulness of the Teacher

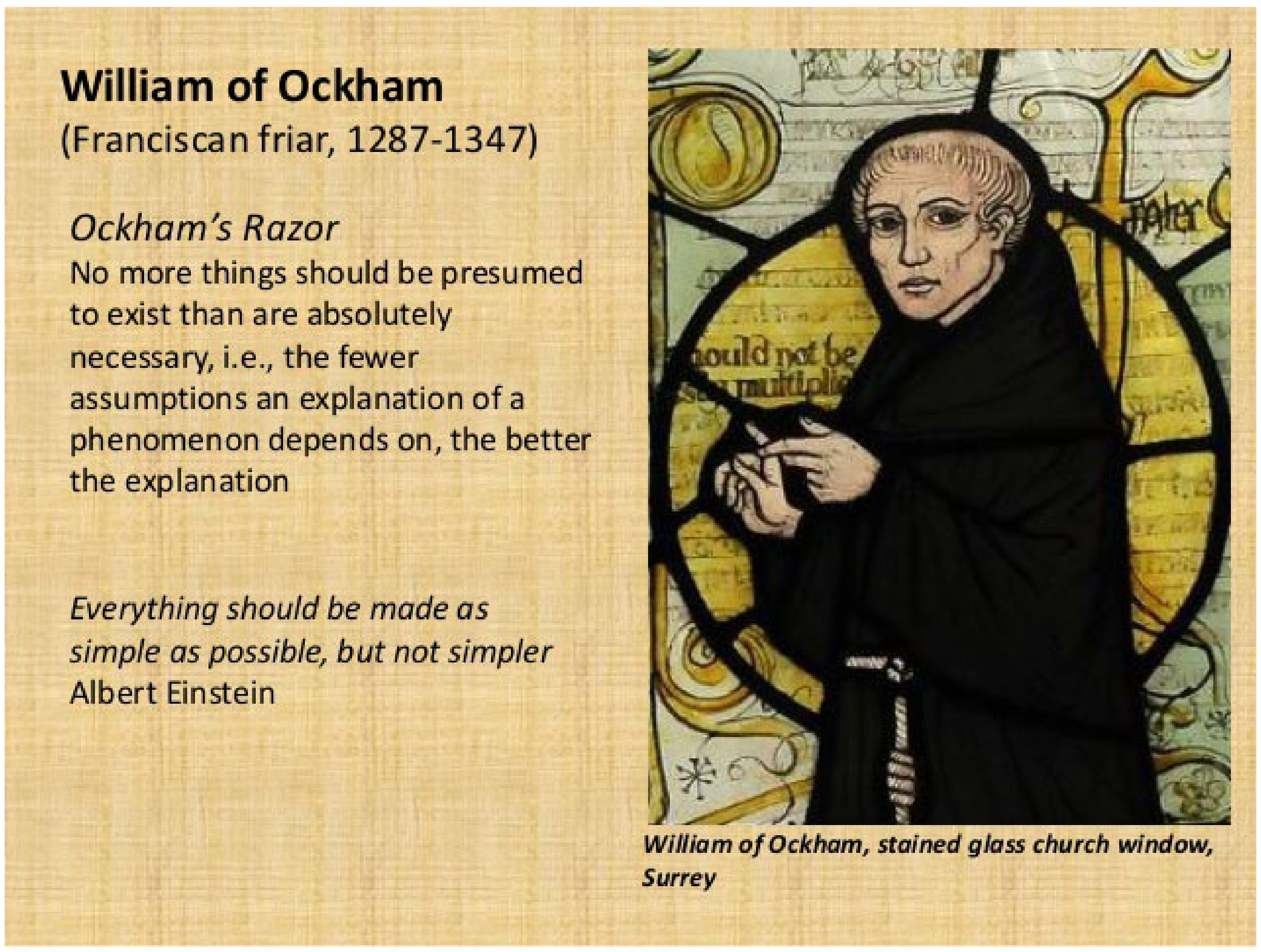
- Trainer, indifferent teacher, or adversarial teacher

Online versus Batch

- The training data comes continuously or as a batch

Machine Learning Principles

Occam's Razor: A short explanation tends to be more valid than a long explanation

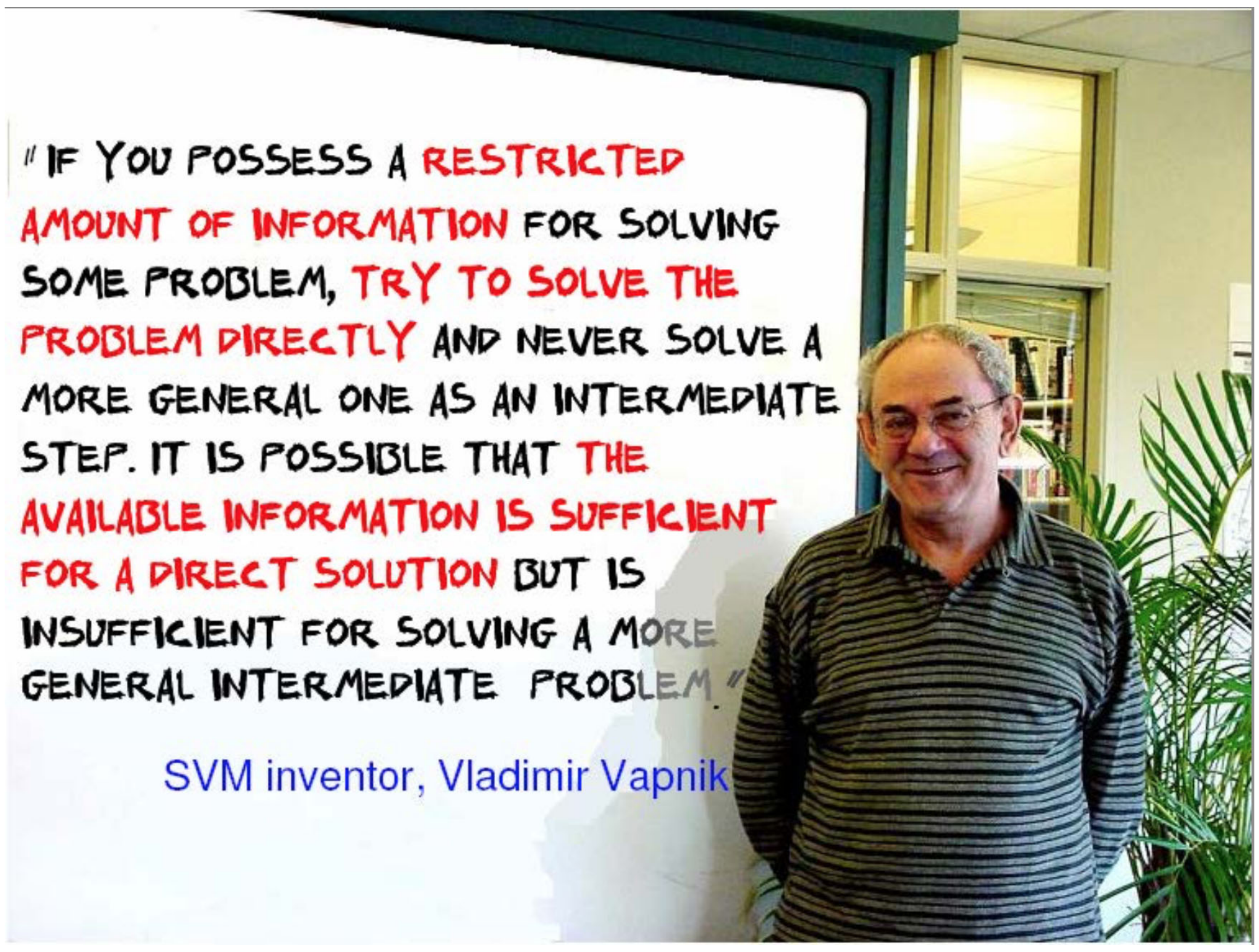


Machine Learning Principles

No Free Lunch Theorem: No learning is possible without some prior knowledge

Machine Learning Principles

Vapnik's Principle: When solving a problem of interest, do not solve a more general problem as an intermediate step



The Statistical Learning Framework 17 / 30

The Statistical Learning Framework

Domain Set $\mathcal{X}$: the set of objects that we wish to label
Label set $\mathcal{Y}$: the set of possible labels.

The learner's task is to:

Input: Training Data, with m examples, where $x_i \in \mathcal{X}$ obeys some fixed but unknown distribution $\mathcal{D}$, and the corresponding $y_i = f(x_i)$ from a target hypothesis f or some random procedure.

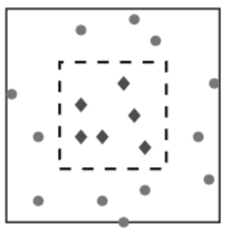
$$S = \{(x_1, y_1), \cdots, (x_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$$

Output: Prediction Rule, also known as a *predictor*, a *hypothesis* or a *classifier*.

$$h : \mathcal{X} \to \mathcal{Y}$$

Example.

- $\mathcal{X} = \mathcal{R}^2$ representing sound and weight of melon
- $\mathcal{Y} = \{\pm 1\}$ representing "tasty" or "non-tasty"
- $h(x) = 1$ if x within the inner rectangle.



□

- What should be the goal of the learner?
- How to measure success?

The Generalization Risk

The error, also known as generalization error, the risk or the true error, of a prediction rule is the probability that it does not predict the correct label on a random data point generated by the underlying distribution.

- Let f be the correct labelling function, so we will try to find h s.t. $h \approx f$
- Let $\mathscr{D}$ be the probability distribution over $\mathscr{X}$
- Given a domain subset, $A \subset \mathscr{X}$, the value of $\mathscr{D}(A)$ is the probability to see a point $x \in A$.
- The error of a prediction rule $h : \mathscr{X} \to \mathscr{Y}$ is defined as:

$$L_{(\mathscr{D},f)}(h) \stackrel{def}{=} \mathscr{P}_{x\sim\mathscr{D}}[h(x) \neq f(x)] \stackrel{def}{=} \mathscr{D}(\{x : h(x) \neq f(x)\})$$

- Can we find h s.t. $L_{(\mathscr{D},f)}(h)$ is small?

The Empirical Risk

Since neither $\mathscr{D}$ nor f is available for the learner, the true error has to be approximated, and one useful notion is through the training error, also known as the empirical error and the empirical risk, the error the classifier incurs over the training sample S:

$$L_S(h) \stackrel{def}{=} \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

- with a fixed h, we have $E[L_S(h)] = L_{(\mathscr{D},f)}(h)$

ERM(S)

This learning paradigm, which comes up with a predictor h that minimizes $L_S(h)$, is called Empirical Risk Minimization (ERM):

Input $S = \{(x_1, y_1), \cdots, (x_m, y_m)\}$
Output any h that minimizes $L_S(h)$

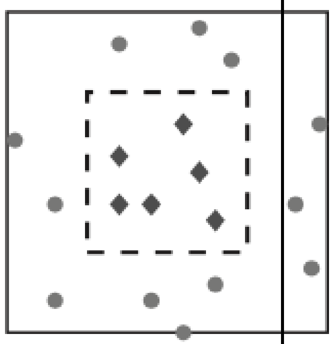
- Is this a good one?

Overfitting

When a predictor performs excellently on the training set, but it performs very poorly on the true "world" (unseen test set), this phenomenon is called overfitting.

Example.

Although ERM seems very natural, it may fail miserably.



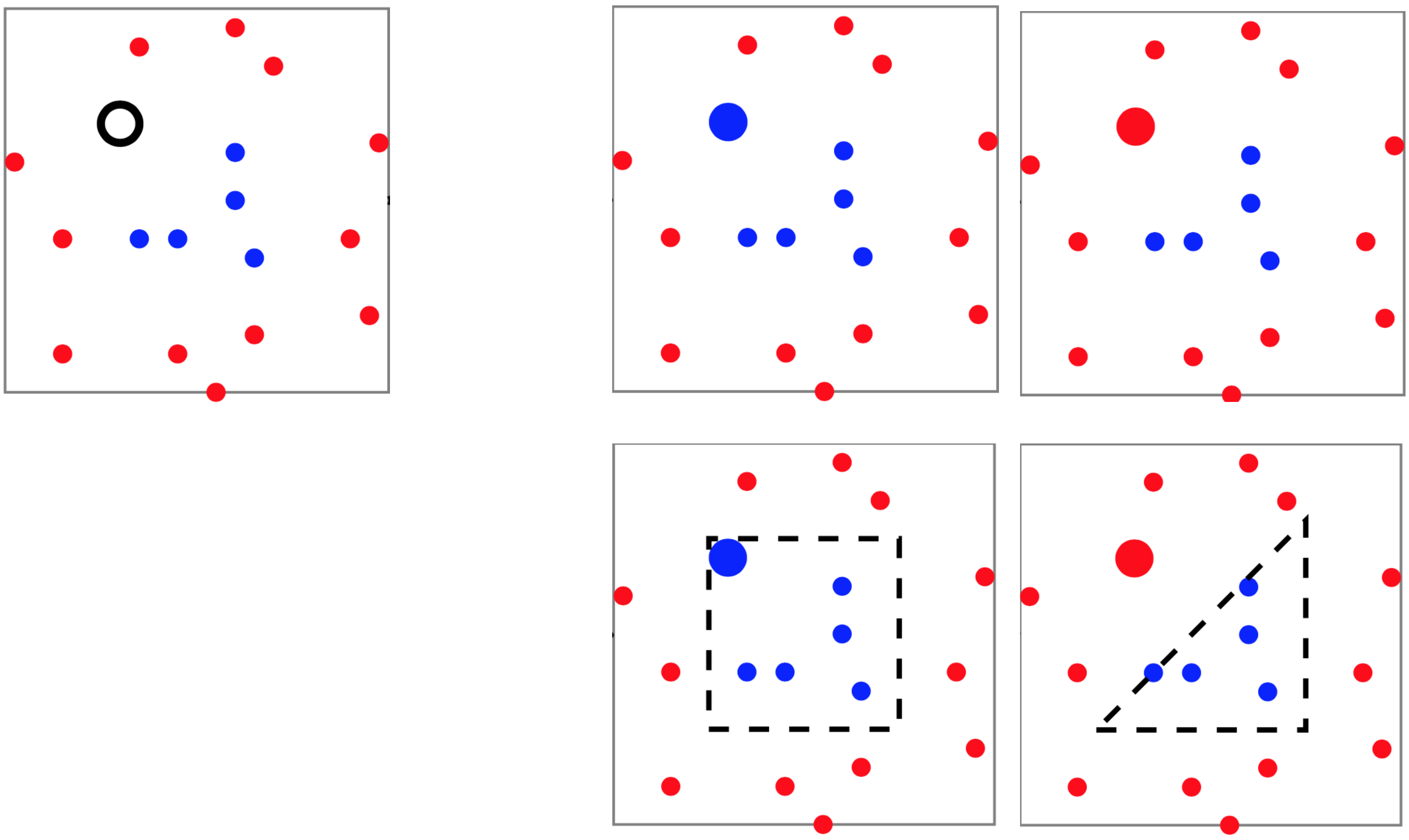
- Assume that
  - the probability distribution $\mathscr{D}$ is such that samples are distributed uniformly within the gray square
  - f determines 1 if the sample is within the inner square.
- Consider a predictor: $h_S(x) = \begin{cases} y_i & \text{if } \exists i \in [m] \text{ s.t. } x_i = x \\ 0 & \text{otherwise} \end{cases}$
  - $L_S(h_S) = 0$: it is one of the empirical minimum cost hypothesis.
  - $L_{\mathscr{D}}(h_S) = \frac{1}{2}$: the true error of any classifier that predicts 1 only on a finite number of instances is $\frac{1}{2}$

□

What is Learnable and *How to Learn*?

Mission Impossible?

- If $\mathscr{X} = \infty$ and on each day we see a new x_t, then the learner can not know its label and might always be wrong
- If $\mathscr{X} < \infty$, the learner can memorize all labels, and there is no learning involved.



Empirical Risk Minimization with Inductive Bias

- A common rectification to overfitting is to incorporate prior knowledge:
  - before seeing the data, the learner should choose in advance a hypothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ from which the target f comes from.
- ERM with Inductive Bias learner chooses a predictor $h \in \mathcal{H}$, with the lowest $L_S(h)$ over $\mathcal{H}$:

$$ERM_{\mathcal{H}}(S) \in \underset{h \in \mathcal{H}}{\operatorname{argmin}} L_S(h)$$

A fundamental question in learning theory:

- Over which hypothesis classes $ERM_{\mathcal{H}}$ learning will not result in overfitting?

Finite Hypothesis Classes

- Let $\mathcal{H}$ be a finite hypothesis class, namely the number of h in $\mathcal{H}$ has an upper bound. We make two simplified assumptions:
  - Realizability: there exist $h^* \in \mathcal{H}$ s.t. $L_{(\mathcal{D},f)}(h^*) = 0$.
  - Independently Identically Distributed (I.I.D.): every x_i is independently sampled according to $\mathcal{D}$.
- Let h_S denote the result of $ERM_{\mathcal{H}}(S)$: $h_S \in \operatorname{argmin}_{h \in \mathcal{H}} L_S(h)$

Issues.

- Since $L_{(\mathcal{D},f)}(h_S)$ depends on the randomly sampled training set S, consequently, there is randomness in the choice of h_S and in the risk $L_{(\mathcal{D},f)}(h_S)$.
- We will therefore address the *probability* to sample a training set for which $L_{(\mathcal{D},f)}(h_S)$ is not too large.
  - We use the accuracy parameter ϵ for the quality of the prediction: $L_{(\mathcal{D},f)}(h_S) > \epsilon$ as a failure of the learner, while $L_{(\mathcal{D},f)}(h_S) \leq \epsilon$ as *approximately correct*.
  - We denote the *probability* of getting a non-representative sample by δ, and call $(1-\delta)$ the confidence parameter of the prediction

□

Learning Finite Hypothesis Classes

Fix ϵ and δ, if $m \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} = \frac{1}{\epsilon}[\log(|\mathcal{H}|)+\log(\frac{1}{\delta})]$, then for every $\mathcal{D}$ and f, with probability of at most δ over the choice of S of size m, we have

$$L_{(\mathcal{D},f)}(ERM_{\mathcal{H}}(S)) > \epsilon$$

Proof.

1. Let $S|_x = (x_1, \cdots, x_m)$ be the instances of the training set.
2. We would like to prove: $\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(ERM_{\mathcal{H}}(S)) > \epsilon\}) \leq \delta$
3. Let $\mathcal{H}_B$ be the set of "bad" hypotheses:

   $$\mathcal{H}_B = \{h \in \mathcal{H} : L_{(\mathcal{D},f)}(h) > \epsilon\}$$

4. Let M be the set of "misleading" samples:

   $$M = \{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$$

5. Observe:

   $$\{S|_x : L_{(\mathcal{D},f)}(ERM_{\mathcal{H}}(S)) > \epsilon\} \subseteq M = \bigcup_{h \in \mathcal{H}_B} \{S|_x : L_S(h) = 0\}$$

□

Proof.

6. We have shown: $\{S|_x : L_{(\mathcal{D},f)}(ERM_{\mathcal{H}}(S)) > \epsilon\} \subseteq \bigcup_{h \in \mathcal{H}_B} \{S|_x : L_S(h) = 0\}$
7. Using the union bound, we have

   $$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(ERM_{\mathcal{H}}(S)) > \epsilon\}) \leq \sum_{h \in \mathcal{H}_B} \mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \leq |\mathcal{H}_B| \max_{h \in \mathcal{H}_B} \mathcal{D}^m(\{S|_x : L_S(h) = 0\})$$

8. If $h \in \mathcal{H}_B$ then $L_{(\mathcal{D},f)}(h) > \epsilon$, so $\mathcal{D}^m(\{S|_x : L_S(h) = 0\} = (1 - L_{(\mathcal{D},f)}(h))^m < (1-\epsilon)^m$.
9. We have shown: $\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(ERM_{\mathcal{H}}(S)) > \epsilon\}) < |\mathcal{H}_B|(1-\epsilon)^m$
10. Using $1 - \epsilon \leq e^{-\epsilon}$ and $|\mathcal{H}_B| \leq |\mathcal{H}|$, we conclude:

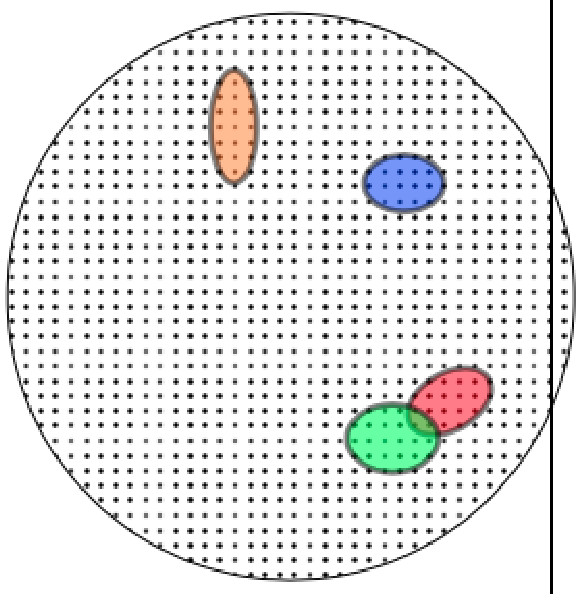
    $$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(ERM_{\mathcal{H}}(S)) > \epsilon\}) < |\mathcal{H}|e^{-\epsilon m}$$

11. If $m \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$, the right hand side would be at most δ.

□

Illustration.

- Each point is a possible sample $S|_x$.
- Each colored oval represents misleading samples for some $h \in \mathcal{H}_B$
- The probability mass of each such oval is at most $(1-\epsilon)^m$.
- But the algorithm might err if it samples $S|_x$ from any of these ovals.



□

ERM Learning

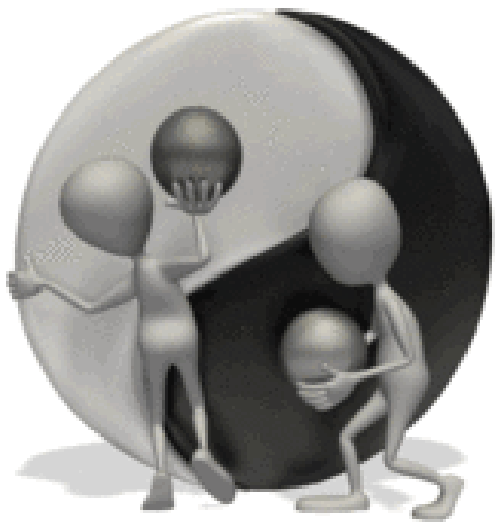
Theoretical analysis:

1. Let $\mathscr{H}$ be a class of binary classifiers over a domain $\mathscr{X}$. Let $\mathscr{D}$ be an unknown distribution over $\mathscr{X}$, and let f be the target hypothesis in $\mathscr{H}$. Fix some $h \in \mathscr{H}$. Show that the expected value of $L_S(h)$ over the choice of $S|_x$ equals $L_{(\mathscr{D},\{\})}(h)$, namely,

   $\mathscr{E}_{S|_x \sim \mathscr{D}^m}[L_S(h)] = L_{(\mathscr{D},f)}(h)$

2. We have shown that the predictor $h_S(x) = \begin{cases} y_i & \text{if } \exists i \in [m] \text{ s.t. } x_i = x \\ 0 & \text{otherwise} \end{cases}$ leads to overfitting. While this predictor seems to be very unnatural, the goal of this exercise is to show that it can be described as a thresholded polynomial. That is, how that given a training set $S = \{(x_i, f(x_i))\}_{i=1}^m \subseteq (\mathscr{R}^d \times \{0,1\})^m$, there exists a polynomial p_S such that $h_S(x) = 1$ if and only if $p_S(x) \geq 0$, where h_S is as defined above. It follows that learning the class of all thresholded polynomials using the ERM rule may lead to overfitting.

Questions?



Contact Information

Associate Professor **Gang Li**
School of Information Technology
Deakin University
Geelong, Victoria 3216, Australia

GANGLI@TULIP.ORG.AU
Open Resources of TULIP-Lab
Team for Universal Learning and Intelligent Processing