Master of Technology in Knowledge Engineering

Unit 1

Intelligent Systems & Techniques for Business Analytics

Machine Learning Fundamentals

Inductive Reasoning & Learning

Sam GU Zhan 顾 瞻

zhan.gu@nus.edu.sg

Objective

- To understand the position of machine learning
- To introduce theoretical foundation of inductive reasoning
- To learn the basic concepts of learning from data
- To understand the role of linearity assumption in correlation analysis for variable assessment

Outline

- Machine learning, Data mining, and Knowledge based system
- Methods of reasoning
  - » Deductive, inductive, and others
  - » Learning and reasoning

All Disney characters are cute. → Since Donald Duck is a Disney character, so he is cute.

Mickey Mouse is a cute Disney character.
Donald Duck is a cute Disney character.
→
All Disney characters are cute.

- Statistical induction
  - » Learning from data, Perceptron learning
  - » Inductive learning problem
  - » Reliability of induction
- Data and information for learning
  - » Variable assessment, correlation coefficient, probability distribution, linearity assumption

What is Machine Learning (ML)

- A branch of ***artificial intelligence***
  - » concerns the systematic construction of systems and study of algorithms that can *learn* from data to improve their *knowledge* or performance
- A marriage of *statistics* and *knowledge representation*
  [P. Flach, 2012]
- What we are interested in:
  - » How can a machine learn? *Learning algorithm; Classification, Regression, Clustering*
  - » How do we quantify the resources needed to learn a given concept? *Data*
  - » Is our target achievable? *Target function f ; Knowledge Engineering*
  - » Can we know whether the learning process succeeded or failed? *ML experiment / evaluation*

The Core of ML

- Deals with
  - » ***Representation*** of
    - ♦ data instances (*feature space*), and
    - ♦ hypothesis (*knowledge, intelligence learned/stored inside model*) evaluated on these instances

  are components of all machine learning systems.

  - » ***Generalization***
    - ♦ The property that the system will perform well on unseen data instances
      - the conditions under which this can be guaranteed are a key object of study in the subfield of *computational learning theory* (not to further discuss)

Machine Learning and Data Mining

- ***Machine Learning***
  - » ML is the study of learnability, and focuses on
    - ♦ the *learning* ***model*** to achieve prediction that is *probably*, *approximately* correct

most of time

mostly right

  - » performance is usually evaluated with respect to the ***ability*** to reproduce/apply *known knowledge* (*generalization*) on new data inputs

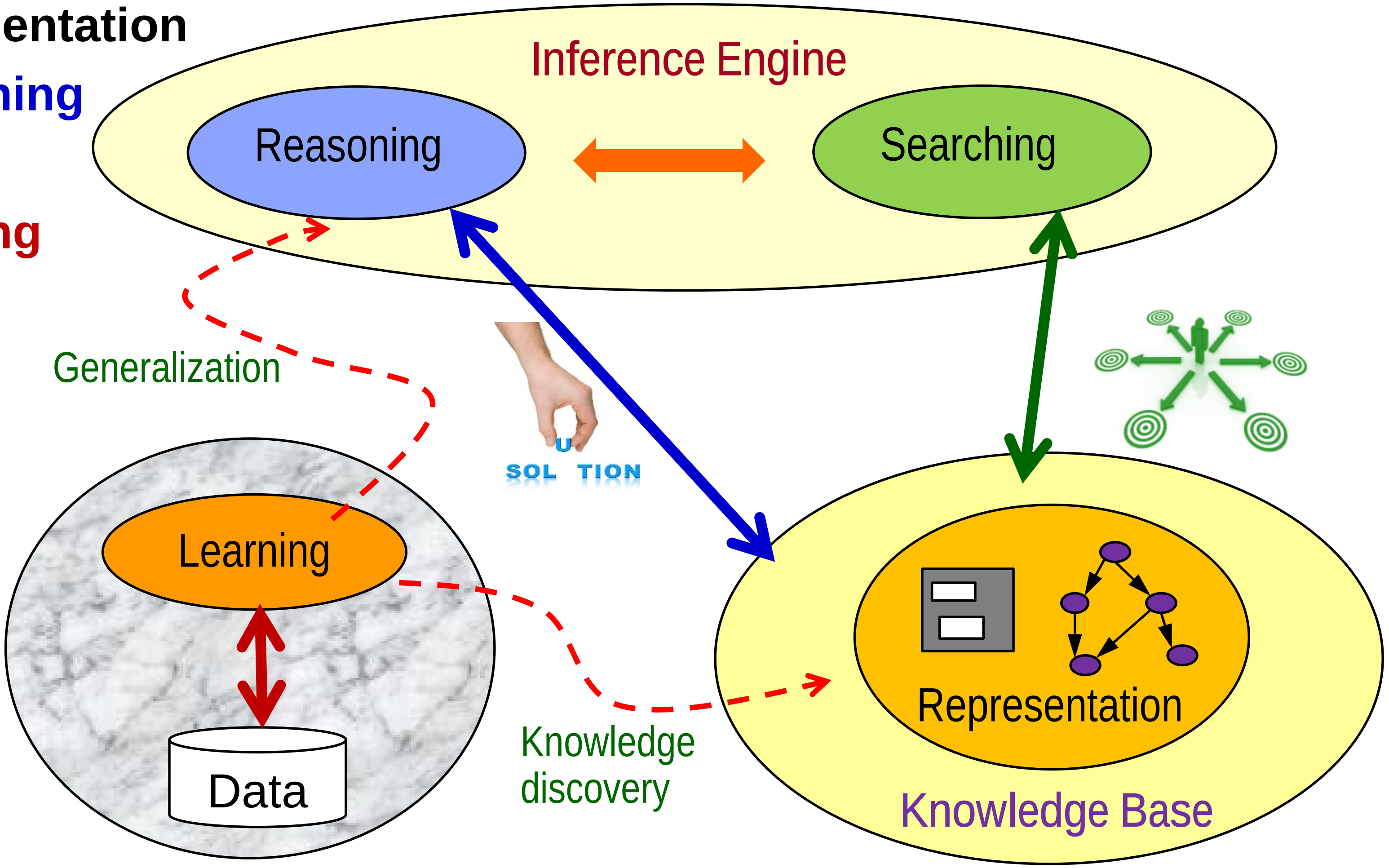
- ***Data Science / Mining***
  - » uses many ML methods, and focuses on
    - ♦ the ***process*** of *discovery* of (previously) unknown properties (*knowledge*) from the data, to solve practical problems, and
    - ♦ the application in business valued prediction and decision making

ML Working for Data Science

- Specialty groups of ML community working on data science / mining:
  - » Neural networks, Support vector machines, Fuzzy logic, Genetic algorithms and programming, Information retrieval, Knowledge acquisition, Text processing, Inductive logic programming, Expert systems, Dynamic programming
    - ♦ All areas have the same objective in mind but accomplish it with their own tools and techniques
- Data science / mining is applied machine learning
  - » NLP, Computer Vision, Robotics, Unsupervised learning, Reinforcement learning, Voice Recognition, Credit risk management, and many more industrial applications

ML & Knowledge Bases System

- Representation
- Reasoning
- Search
- Learning



Machine Learning versus Statistical Learning

- Think of SL as one of the methods/approaches of realizing ML
  - » ML has computer science roots (e.g. rule-based learning)
  - » SL roots in mathematics, which is based on probability theory and models

  * Robot learning is a kind of ML (reinforcement), but not SL

- Statistics versus Statistical Learning
  - » Statistics is broader than statistical learning, covers $\underbrace{\text{descriptive statistics}}_{\textit{Visualizing and summarizing data}}$ + $\underbrace{\text{inferential statistics}}_{\textit{Making inferences, drawing conclusions}}$

What is Deep Learning

- ***Deep learning*** is part of a broader family of ML methods based on learning ***representations*** of data
  - » has been characterized as a buzzword, or a ***rebranding of neural networks***

- Deep Learning = Deep Machine Learning
  - » Deep learning is also known as ***deep structured learning***, hierarchical learning or ***deep machine learning***,
  - » is a branch of machine learning based on a set of algorithms that attempt to model ***high level (and sometimes hidden)*** abstractions in data
    - ♦ by using a ***deep*** graph with multiple processing layers, composed of multiple linear / nonlinear transformations

Kind of Artificial Neural Network

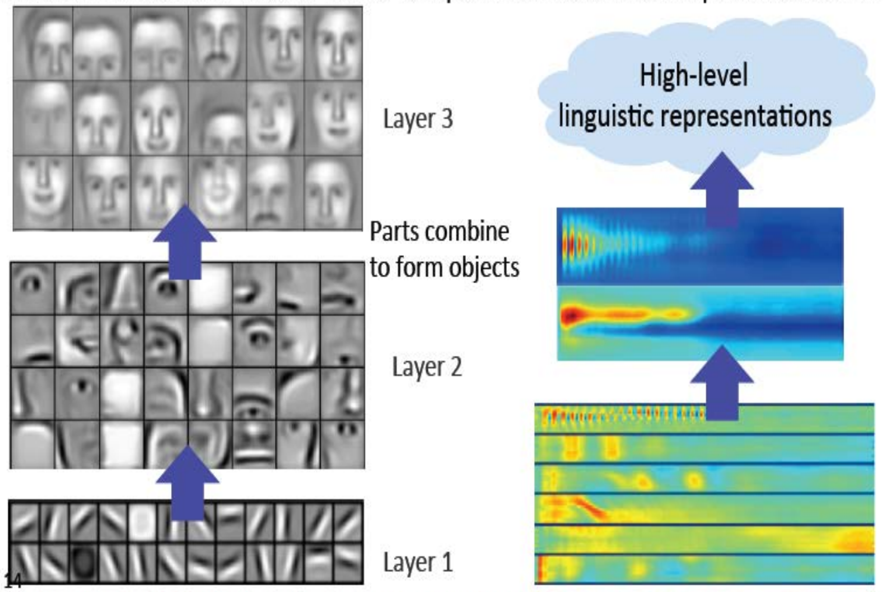
The Power of Deep ML: representation

- An observation (e.g., an image) can be represented in many ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of particular shape, etc.
  - » Research in this area attempts to make ***better*** representations and create models to learn these representations from largescale unlabelled data.
  - » Some representations are better than others at simplifying the learning task (e.g., face recognition or facial expression recognition)

The Power of Deep ML: representation (cont.)

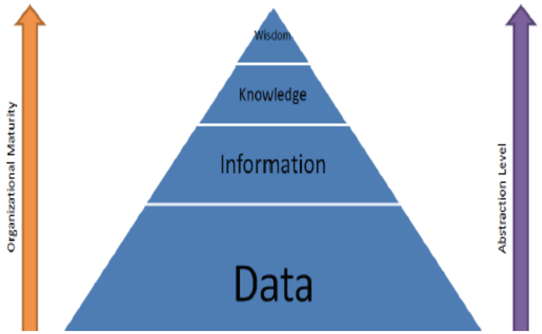
» Deep learning is about learning multiple levels of feature representation and abstraction that help to make sense of data such as images, sound, and text.

Successive model layers learn deeper intermediate representations


Layer 3
Parts combine to form objects
Layer 2
Layer 1
High-level linguistic representations
14

Prior: underlying factors & concepts compactly expressed w/ multiple levels of abstraction

- The big payoff of deep learning is to allow learning higher levels of abstraction
- Higher-level abstractions disentangle the factors of variation, which allows much easier generalization and transfer



(source: Yoshua Bengio at http://www.iro.umontreal.ca/~bengioy/talks/mlss-austin.pdf

Deep Architectures & Applications

- One of the promises of deep learning is
  - » replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised *feature learning* and hierarchical feature extraction

- Various deep learning architectures such as
  - ♦ *deep* neural networks, convolutional neural networks, *deep* belief networks, recurrent neural networks, long short time memory
  - » have been applied to fields like computer vision, object recognition, automatic speech recognition, natural language processing, audio recognition and bioinformatics where they have been shown to produce state-of-the-art results on various tasks.

Methods of Reasoning

Inference Paradigms

- Besides forward chaining (FC) and backward chaining (BC) inference, another basic inference paradigm of rules in classical AI is *generation & test*
  - » Key idea: *a combination of FC and BC*
    - ♦ Generate a likely solution using FC, then test if the solution meets requirements using BC by carrying out the processing simultaneously
    - ♦ When there are many likely candidates of solution, seek for more evidences by priority to narrow down the solution set
  - » Application examples
    - ♦ in early expert systems: DENTRAL, MYCIN
    - ♦ in Singapore: CES

Methods of Knowledge-based Reasoning

- ***Reasoning*** (in broad sense) is a process of how humans
  - » draw conclusions (or make decisions) from a body of knowledge (the Knowledge Base); or
  - » create new knowledge by combining piecemeal knowledge

- There are several important ***reasoning*** methods
  - » Deductive, Inductive, Abductive, Analogical

can be supported by ***inference*** with

- » forward chaining
- » backward chaining
- » or a combination of them

Deduction

- Given A and A → B, we conclude B
  - » Key idea:
    - ♦ ***deduce*** new fact from logically related known fact
    - ♦ if premise is true, the conclusion is guaranteed to be true

rule:	A → B
fact:	A
conclude:	B

Example 6.1:

Rule:	If it is raining (A), then the street is wet (B)
Fact:	it is raining (A)
Conclusion:	the street is wet (B)

Deduction (cont.)

- In Modus ponens (discussed previously)

$$\frac{\begin{array}{l} P \\ P \rightarrow Q \end{array}}{\therefore\ Q}$$

  - » If both the rule (implication) and the known fact have full truth, then the conclusion has full truth as well
  - » The truth of conclusion is established on a firm logical basis
    - ☝ deduction is often applied for “proof”

Abduction

- Given B and A → B, we conclude A
  - » Key idea:
    - ♦ ***explain*** effects in terms of their causes
    - ♦ the conclusion follows from the observed evidence

rule:	A → B
fact:	B
conclude:	A

Example 6.2:

Rule:	If it is raining (A), then the street is wet (B)
Fact:	the street is wet (B)
Conclusion:	*it is raining (A)*

Abduction (cont.)

- Often relates to probabilistic reasoning based on conditional probability
  - » Previous example may be interpreted in conditional probability   p(B|A)
    - the probability of “street is wet” given “it is raining”, say p(B|A) = 1 (100% certain)
  - » If given B (wet street seen), what is the probability of “raining”, p(A|B)?
    - ♦ by Bayes theorem   p(A|B) = P(A)*p(B|A) / P(B) is usually < 1
    - ♦ the raining days are usually not more than the days of “wet street” which may be caused by other reasons

Analogy

- Key idea:
  - use similarity to ***retrieve*** past cases to provide explanation for new case
  - Analyze case differences, then adapt to new situation and draw conclusion

Example 6.3:

Case: Tigers are big cats, carnivorous, & live in India

New case: *Lions are similar to tigers*

Conclusion: *Lions are big cats, carnivorous, & live in India*

Analogy (cont.)

- This type of reasoning is used in case-based reasoning systems
- Key challenges
  - » Similarity measure defined on a sufficiently complete vocabulary
  - » Case adaptation has no systematic approaches, but is more a knowledge-based processing

    (the “tigers” example has no adaptation)

Induction

- Given $P = \{a, b, c, \ldots\}$ and $a \to Q$, $b \to Q$, $c \to Q$,
  then $P \to Q$
  - » Key idea:
    - ♦ use observations to draw “premises and conclusion”
    - ♦ form a ***generalization*** from a body of knowledge (facts)

Example 6.4:

Observation: All cars seen entering ISS staff carpark are Japanese models

Conclusion: *all ISS staff owned cars are Japanese cars?*

Induction (cont.)

- This is basic idea of ***data driven machine learning*** in neural networks, rule-induction and decision trees
  - » Inductive rules are of heuristic in nature
  - » It captures knowledge from data
- With increasing amount of data available, inductive reasoning and learning attracts more attention
- *Reliability is an issue*
  - » Learnt model cannot produce reliable results for those new inputs which are outside training data’s know value range (unseen data).

Methods of Reasoning: Summary

- Deductive reasoning
  - » is based on Modus ponens, with logical truth guaranteed
- All the other methods
  - » have no guaranteed true conclusion even with true premises given, but are very useful in KBS where there is no deductive rules available
- Handling of uncertainty and imprecision in knowledge and reasoning is important

Reasoning and Proof

- In traditional way, reasoning can be modeled by a formal proof of argument
  - » One starts by accepting certain premises, and then accepts intermediate conclusions that follow from the premises or earlier intermediate conclusions in accordance with rules of inference (e.g.: Modus ponens)
  - » One ends by accepting new conclusions that one has inferred directly or indirectly from the original premises

More than Proof

- Reasoning is ***more than a proof***, also a matter of connecting ‘knowledge dots’; and rejecting things/beliefs from what one starts out believing
  - » Reasoning often involves abandoning things one starts out believing, e.g.:
    - ♦ One discovers an inconsistency in one’s early beliefs and so reasons about which belief to give up
    - ♦ One starts by accepting a particular datum/belief that one later rejects as an “outlier”
- More generally, one regularly *modifies* previous opinions in light of new data/information/evidence.

The Need of Induction

- In the traditional view, a deductive logic is a theory of reasoning and concerned with deductive rules of inference , such as
  - ♦ All apples have seeds

    So, the next apple will have seeds

knowledge

- It has been suggested that we need an inductive reasoning that specifies inductive inference rule, like
  - » Many apples have been found to have seeds. Until now, no apples have been found not to have seeds.

    So, the next apple will have seeds.

data

Deduction

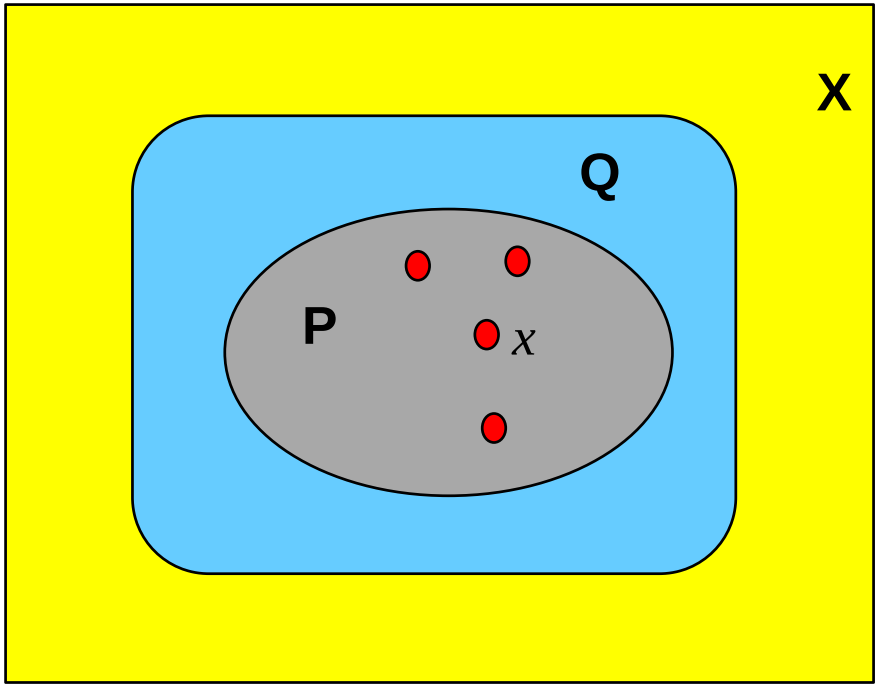
- Example 6.5:
  - » Express the rule in example 6.1 (raining and wet street) in predicate logic

$$\forall x\ p(x) \rightarrow q(x)$$

- ♦ “for everyday, if that is a raining day then the street on that day is wet”
- ♦ $x \in X$, X is the universal set of all the days under discussion
- ♦ $p(x)$ indicates those days of raining
- ♦ $q(x)$ indicates those days with wet street

Deduction (cont.)

- Relationship between the predicates p (raining) and q (wet street) in sets
  - » For all x∈X, if x∈P then x∈Q
    - ♦ Subsets P ⊆ Q
  - » Important assumption based on *knowledge*
    - ♦ If it is raining then the street ***must*** be wet
      - *There is no such a case that it is a raining day but the street is not wet on that day*



Induction

- Example 6.6:
  - » Examine Example 6.4 for the relationship (ISS staff drive Japanese cars) using predicates *c* (cars seen) and *j* (Japanese cars)
  - » x ∈ X
    - ♦ X is the universal set of all the cars under discussion
  - » c(x)
    - ♦ indicates those cars seen entering ISS staff carpark
  - » j(x)
    - ♦ indicates Japanese cars
  - » s(x)
    - ♦ indicates cars owned by ISS staff

Induction (cont.)

- What from observations:

  $x_1 , x_2 , \ldots, x_n \in C \cap J \neq \varnothing$

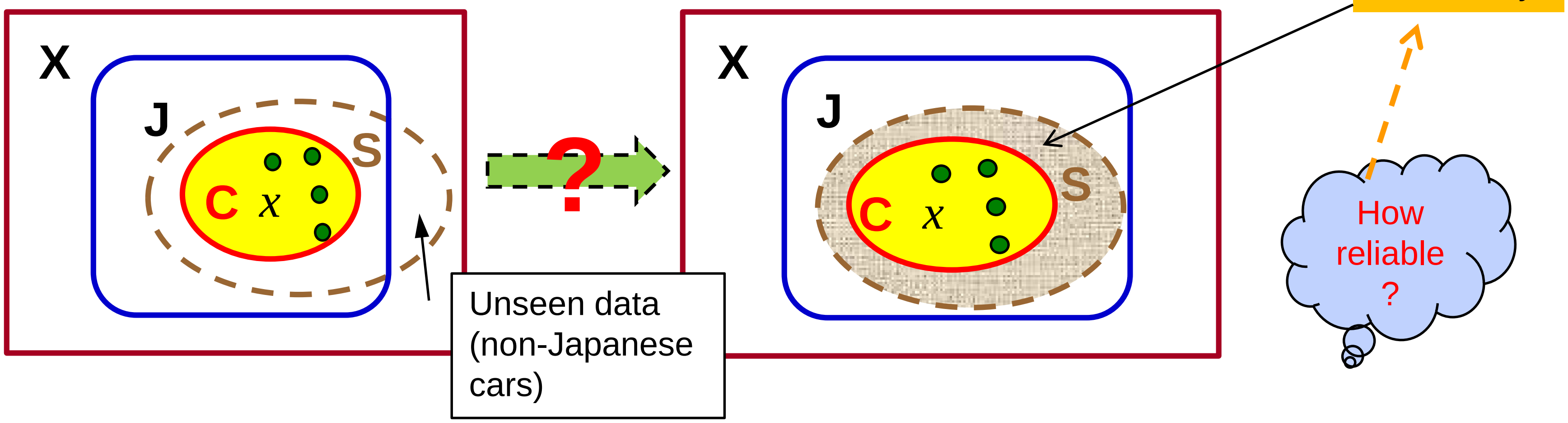
$\forall x \; c(x) \rightarrow j(x)$ ← Observation (for sure)

$\Downarrow$ $\quad C \subseteq S$

$\exists x \; s(x) \wedge j(x)$

- What we try to conclude:

  $\exists x \; s(x) \wedge j(x) \xrightarrow{?} \forall x \; s(x) \rightarrow j(x)$ → Prediction with uncertainty



Deduction and Induction: theoretical foundation

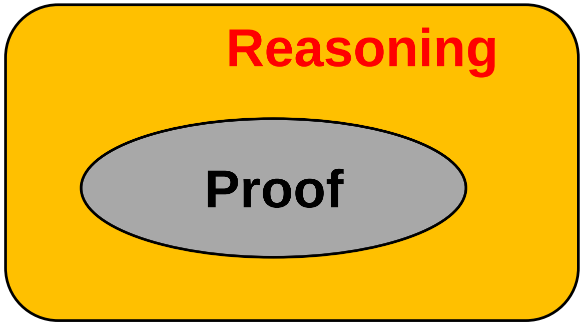
- Deduction and induction rely on different theoretical foundations
  - » Deduction: based on *logic* (Modus Ponens)
    - ♦ To deduce particular conclusions from universally true knowledge base (*knowledge driven*)
      - *from* universal *to* particular
  - » Induction: based on *statistics*
    - ♦ To generalize many particular affirmative observations (facts) to universal affirmative propositions (*data driven*)
      - *from* particular *to* universal

Deduction and Induction: theoretical foundation (cont.)

- The traditional picture of induction and deduction, conflates two quite different things
- Deduction
  - » **What follows from what**
    - ♦ with perfect conditional reliability
- Induction
  - » **What can be inferred from what**
    - ♦ with imperfect conditional reliability

Deduction and Induction: Proof or Reasoning

- A logical proof (deductive argument)
  - » is an abstract structure of propositions
    - ♦ premises → conclusion (implication)
- Reasoning ≠
  - » is a process of activity, involving argumentations, assembling evidence to support a viewpoint.
    - ♦ "premises": ≠
      - starting position of belief, evidence, …
    - ♦ "conclusion":
      - explanation, result, …



Induction: learning & reasoning

- The concept of induction actually crosses over two important human activities
  - Learning (*generalization*) (ISS staff drive Japanese cars)
    - ♦ $\exists x\ s(x) \wedge j(x) \Longrightarrow \forall x\ s(x) \rightarrow j(x)$
  - Reasoning (*prediction*)
    - ♦ For next car $x_{n+1} \in X$, if s(x') is true then j(x') is true ?
- We are interested in two things
  - how to get a good inductive rule generalized from observations (*learning*)
  - how reliable we can do the inference, if only given the inductive rule (*reliability of reasoning*)

What is (Machine) Learning?

- For a specific *task*, learning is any process by which a system improves *performance* from *experience*
  - » the specific “ task” is a piece of work
    - ♦ E.g.: recognize objects in picture
  - » the “performance” is comparable measurements
    - ♦ E.g.: Number of objects detected; Accuracy
  - » the “experience” **can be** expressed by data/features
    - ♦ ***Machine learning from data → Data science?***

Some of Typical Tasks: Classification

- Assign object/event to one of a given finite set of categories.

  Credit card applications,
  Financial investments,
  Fraud detection in e-commerce,
  Worm detection in network packets,
  Spam filtering in email,
  Recommended articles in a newspaper
  Recommended books, movies, music, or jokes,
  Medical diagnosis, DNA sequences,
  Spoken words, Handwritten letters, Astronomical images
  … …

Some of Typical Tasks: Problem Solving

- Performing actions in an environment in order to achieve a goal
  - » Playing checkers, chess, go, sudoku, or backgammon
  - » Planning a project
  - » Balancing a pole
  - » Driving a car or a jeep
  - » Flying a plane, helicopter, or rocket
  - » Controlling a mobile robot
  - » Pass KE exams
  - » … …

Learning from Data

- Learning from data is used in situations where we don't have an analytic solution, but we do have data that we can use to construct an *empirical solution/model*

Learning from Data: an example

- Application example as our metaphor:
  - » Suppose that a bank receives thousands of credit card applications every day, and it wants to automate the process of evaluating them.
  - » The bank knows of no magical formula that can pinpoint when credit should be approved, but it has a lot of data. The bank uses historical records of previous customers to figure out a good formula for credit approval.
  - » Each customer record has personal information related to credit, such as annual salary, years in residence, outstanding loans, etc. the record also keeps track of whether approving credit for that customer was a good idea.
  - » This data guides the learning and construction of a successful model for credit approval that can be used on future application.

Learning Problem: main components

- A ***data set*** D of input-output sample cases
  - » $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $y_i = f(\mathbf{x}_i)$ for $i = 1, \ldots, N$
- The ***unknown target function***
  - » $f: X \rightarrow Y$ where X is the ***input space*** (set of all possible inputs **x**), and Y is the ***output space*** (set of all possible outputs, in credit card example: just a yes/no decision)
- The ***hypothesis set*** *H* of candidate formulas under consideration, and ***final hypothesis*** h from *H*
- The ***learning algorithm*** that uses the data set to choose a formula $h: X \rightarrow Y$ that approximates f

Learning Problem: solution

- Given the input **x**, when a new customer applies for credit,
  - » the bank will base its decision on h (the hypothesis produced by the learning algorithm), but not on f (the ideal target function which remains **<u>unknown</u>**)
- To achieve a good solution, the algorithm chooses h that best matches f on the historical training data of previous customers
  - » with the **<u>hope</u>** that it will continue to approximate f on new customers (i.e.: the justification remains to be seen)

No guarantee

Learning Model

- Given a specific learning problem
  - » The target function and training data are determined by the problem we want to solve
  - » The learning algorithm and hypothesis set are determined by the ***learning model*** we adopt

the hypothesis set

+

the learning algorithm

} the *leaning model*

  - » The final hypothesis selected by the leaning algorithm based on training data is sometimes called the ***model learned*** (or simply, *model*)

Learning Model (cont.)

- Usually we specify the ***hypothesis set*** H through a functional form $h(x)$ that all the hypothesis $h \in H$ share
  - » This functional **form** can be explicitly describable
    - ♦ e.g.: a linear function

There is no explicitly given list of all the candidates of hypothesis

  - » Or described using a more complex topological structure
    - ♦ e.g.: a tree structure or a neural network structure
- The ***learning algorithm*** changes adjustable parameters of function h to choose a good hypothesis from H

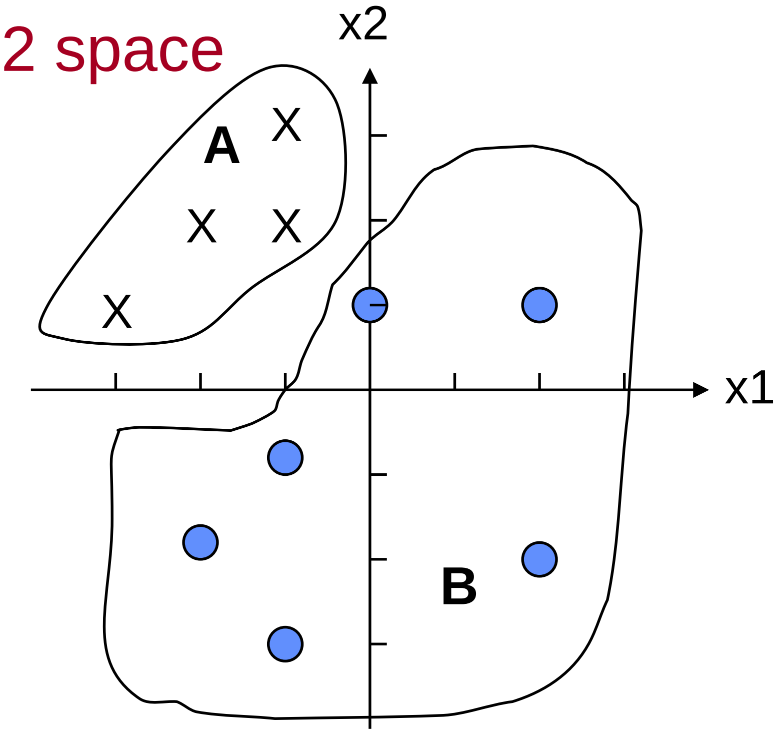
Classification with Single Perceptron

- ***Perceptron*** is one simple learning model

Inventor: Frank Rosenblatt, 1957
simplest artificial neuron with Heaviside step activation function

- Example 6.7:
  - » Given ten sample points on X1-X2 space
  - » Class A with four data points
    - ♦ (-3, 1), (-2, 2), (-1, 2), (-1, 3)
  - » Class B with six data points
    - ♦ (-1, -1), (-2, -2), (-1, -3)

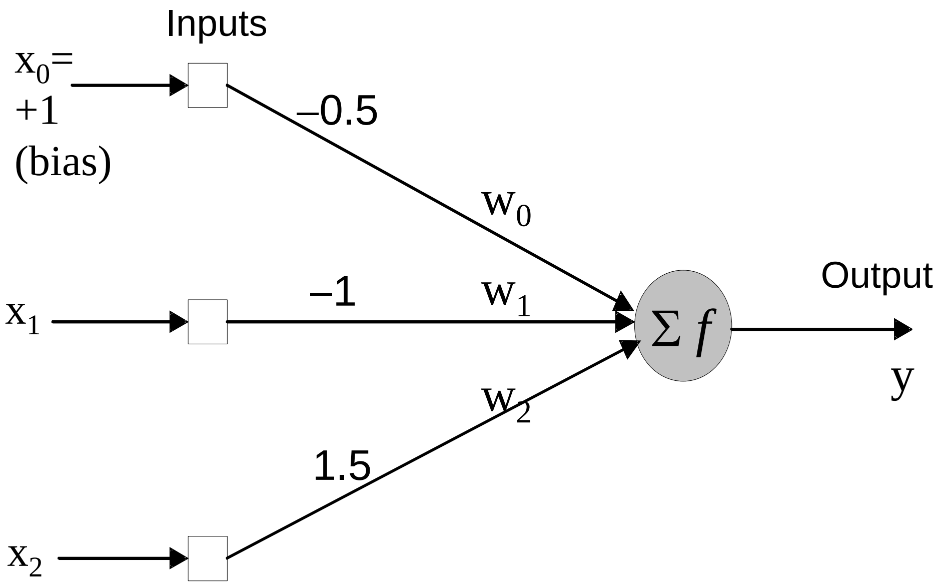
      (2, -2), (2, 1), (0, 1)



Classification with Single Perceptron (cont.)

- Single perceptron
  - ♦ $I = \Sigma_i x_i w_i$
  - ♦ Activation function

$$y = f(I) = \begin{cases} 1 & \text{if } I > 0 \\ 0 & \text{otherwise} \end{cases}$$



» y = 1 indicate class A

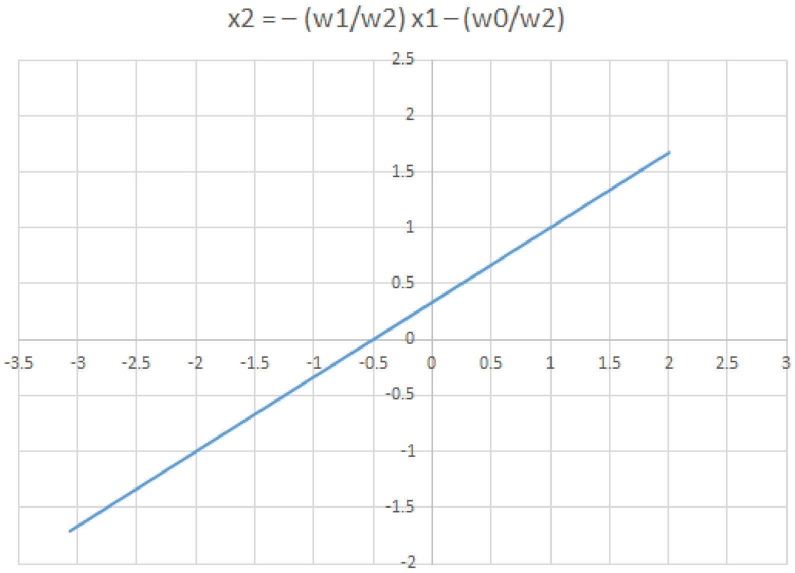
» y = 0 indicate class B

» Let initial weights be $w_0 = -0.5$, $w_1 = -1$, $w_2 = 1.5$

Note: a hard-limiting function is used here for activation function (for easy discussion), other kinds of continuous functions are possible

Classification with Single Perceptron (cont.)

Example 6.7 (cont.)



- class A:
  - » No error for all the four data points
    - ♦ (-3, 1), (-2, 2), (-1, 2), (-1, 3)
- class B:
  - » No error for first five data points
    - ♦ (-1,-1), (-2,-2), (-1,-3), (2,-2), (2,1)
  - » However, there is an error for (0, 1)

Perceptron: formulation

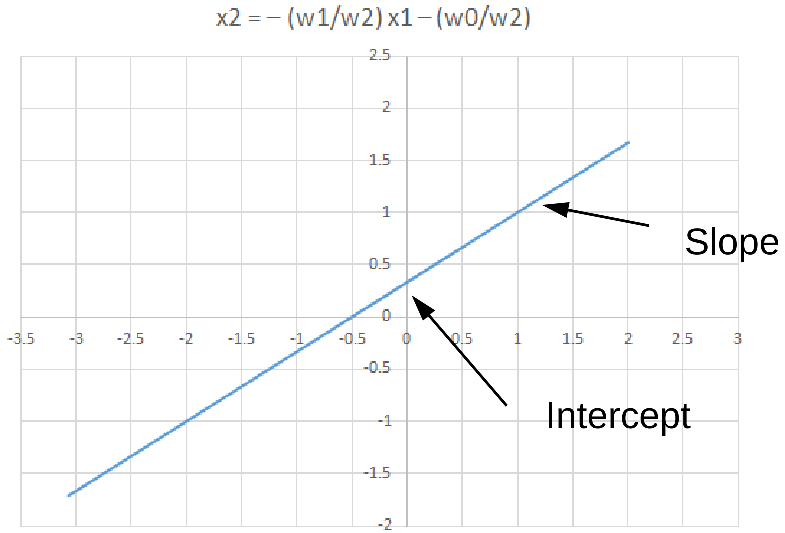
- Consider 2-dimensional space
  - » Equation for straight line is usually represented as

$$x_2 = a \times x_1 + b$$

- Weighted sum for single perceptron

$$w_0 + w_1 \times x_1 + w_2 \times x_2 = 0$$

$$x_2 = -\,(w_1/w_2)\; x_1 - (w_0/w_2)$$



$-\,(w_0/w_2)$ — ***intercept*** 0.5/1.5 = 0.333333333

$-\,(w_1/w_2)$ — ***slope*** 1/1.5 = 0.666666667

Perceptron: formulation (cont.)

- The activation function

$$\sum_{i=1}^{n} W_i X_i + b = 0$$

the functional form of hypothesis

» forms a hyperplane in the n-dimensional space, dividing the space into two halves.

♦ using θ* = –b as a threshold to produce output value (1 or 0) means classifying the instances to two classes.

♦ when n = 2, the hyperplane becomes a line.

» It can also be represented as

$$\sum_{i=0}^{n} W_i X_i = 0$$

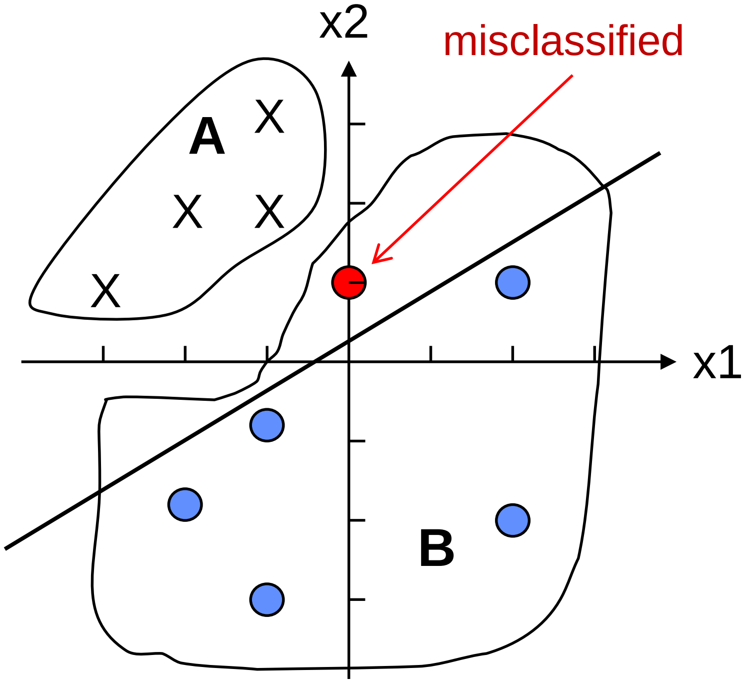
☝ Letting $\theta = -W_0$ makes it adjustable during learning

Perceptron: Hyperplane

- In Example 6.7:
  - » the initial weights
    - ♦ $w_0 = -0.5$, $w_1 = -1$, $w_2 = 1.5$

    form up the hyperplane:
    - ♦ $x_2 = (2/3)\, x_1 + 1/3$

    slope = 2/3, intercept = 1/3



- How can we find a right hyperplane that can classify all the data correctly?

  — ***Perceptron learning***

Perceptron Learning Algorithm

- The idea of perceptron learning
  - » If the output is ONE and should be ONE or if the output is ZERO and should be ZERO (*no error*), do nothing (no change to weights).
  - » If the output is ZERO (inactive) and should be ONE (active), **increase** the weight values on all active input links.
  - » If the output is ONE (active) and should be ZERO (inactive), **decrease** the weight values on all active input links.

Perceptron Learning Algorithm (cont.)

- Perceptron learning algorithm as a formula
  - » $\Delta W_i = \alpha\ (T - O)\ X_i = \alpha \times \text{error} \times \text{input-}i$
  - » $W_i\ (t+1) = W_i\ (t) + \Delta W_i$

where

$W_i(t)$ — the weight at time *t* (or *t*-th iteration)
ΔW_i — the change made to weight W_i
X_i — the *i*-th input
α — learning step ($0 < \alpha < 1$)
T — target output
O — actual output

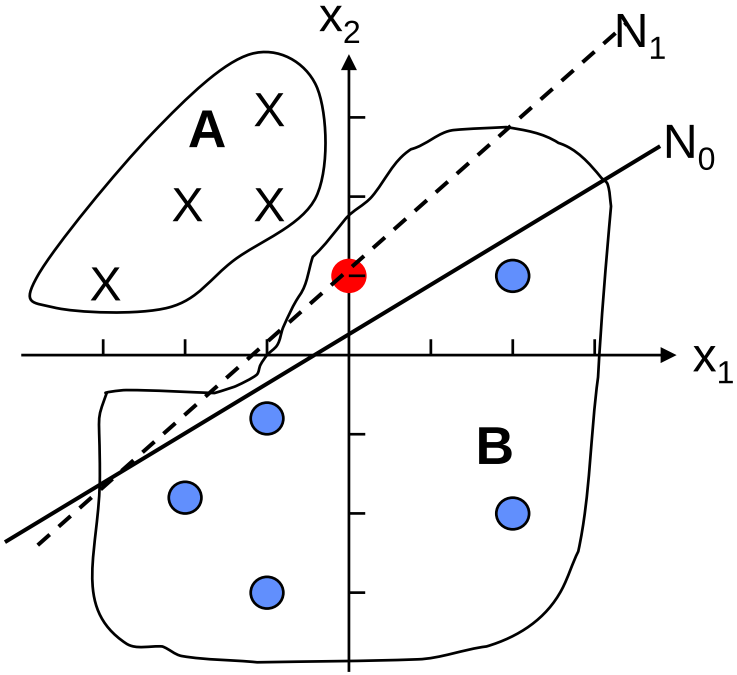
Perceptron: Learning Hyperplane

- Continue the example
  - initial weight $w_0 = –0.5$, $w_1 = –1$, $w_2 = 1.5$
- Assume a learning step $\alpha = 0.5$
  - using the 10 given sample data for training (assuming the sequence as appearance)
- There is an error for (0, 1) of class B
  - Learning

$$\Delta W_0 = \alpha\ (T – O)\ X_0 = 0.5 \times (0 – 1) \times 1 = -0.5$$
$$\Delta W_1 = \alpha\ (T – O)\ X_1 = 0.5 \times (0 – 1) \times 0 = 0$$
$$\Delta W_2 = \alpha\ (T – O)\ X_2 = 0.5 \times (0 – 1) \times 1 = -0.5$$
$$W'_0 = –0.5 + (–0.5) = –1$$
$$W'_1 = –1$$
$$W'_2 = 1.5 + (–0.5) = 1$$

Perceptron: Learning Hyperplane (cont.)

- N_0, the perceptron with the initial weights
  - » $w_0 = -0.5$, $w_1 = -1$, $w_2 = 1.5$
    - ♦ slope = 2/3, intercept = 1/3
- N_1, the perceptron after learning
  - » $w_0 = -1$, $w_1 = -1$, $w_2 = 1$
    - ♦ slope = 1, intercept = 1

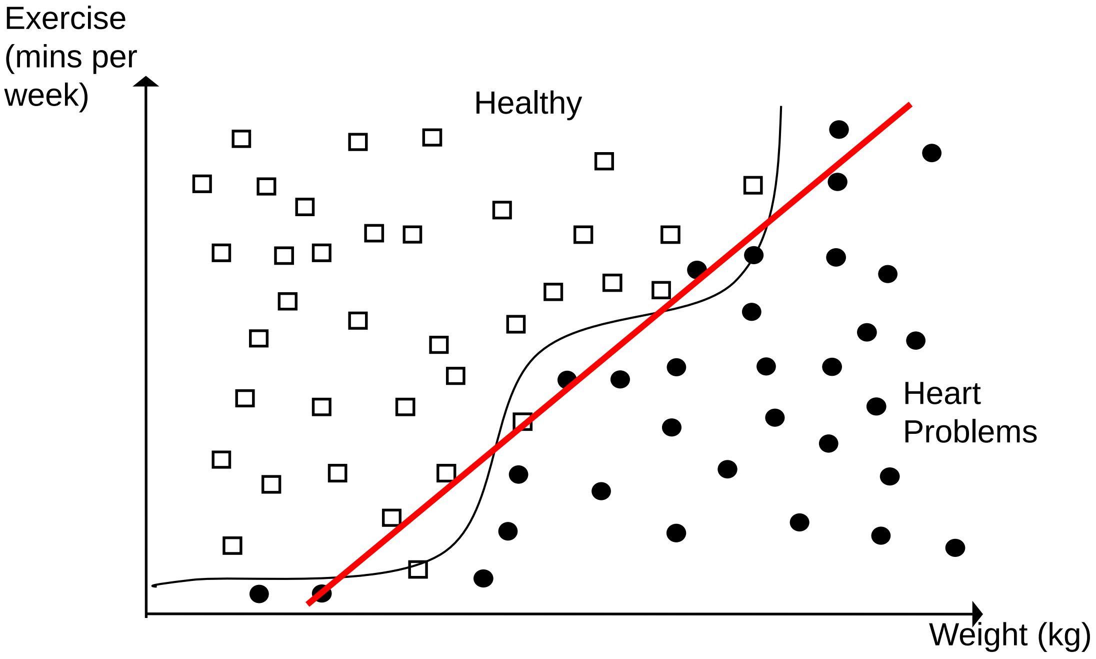


Previously misclassified point (0, 1) can be handled correctly now, but the decision hyperplane is not optimal.

Single Perceptron and Linear Separability

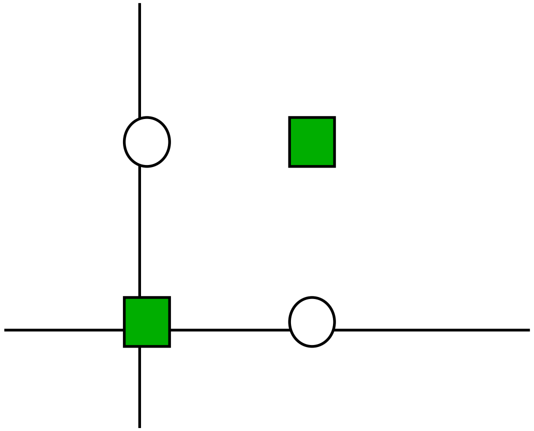
- ***Linear separability***
  - » When a linear hyperplane exists to place all instances of one class on one side and all the other class on the opposite side.

- E.g.: *exercise-health* problem is not linearly separable
  - » single perceptron cannot handle it correctly



XOR Problem

- Four objects
  - » (0, 0) -> 0 (false)
  - » (0, 1) -> 1 (true)
  - » (1, 0) -> 1 (true)
  - » (1, 1) -> 0 (false)
- Initial weights
  - » $w_0 = 1, w_1 = 1, w_2 = 1$
  - » learning step $\alpha = 0.5$



- How can we solve this problem (separate 🟩 and ○)?

  (to be further discussed later)

Discussion: setup for learning from data

- Express each of the following tasks in a) and b), in the framework of learning from data by specifying

the input space X

the output space Y

the target function f: $X \rightarrow Y$

and the specifics of the data set that we will learn from

a) Medical diagnosis: a patient walks in with a medical history and some symptoms, and you want to identify the cause

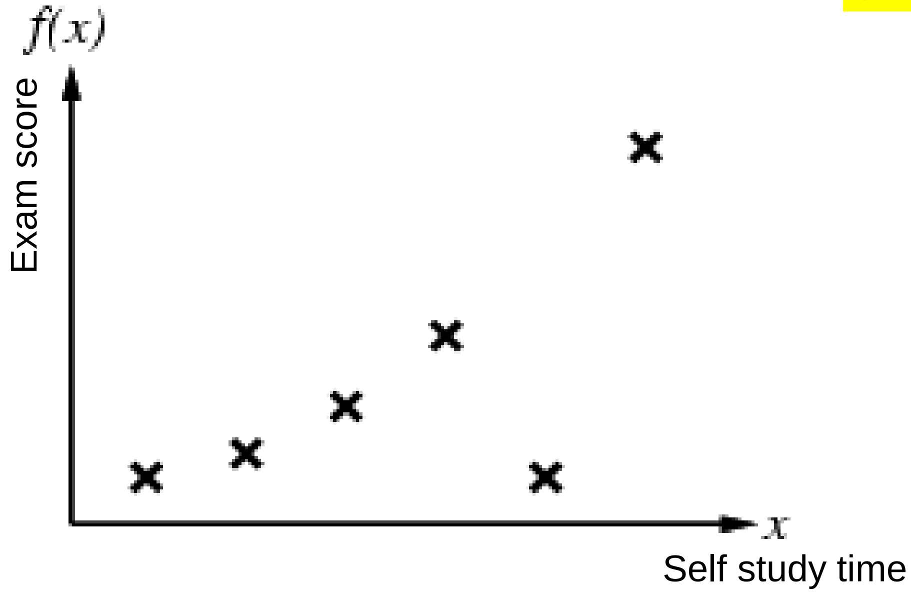
b) Determining if an email is a spam or not

Inductive Learning

Inductive Learning Method

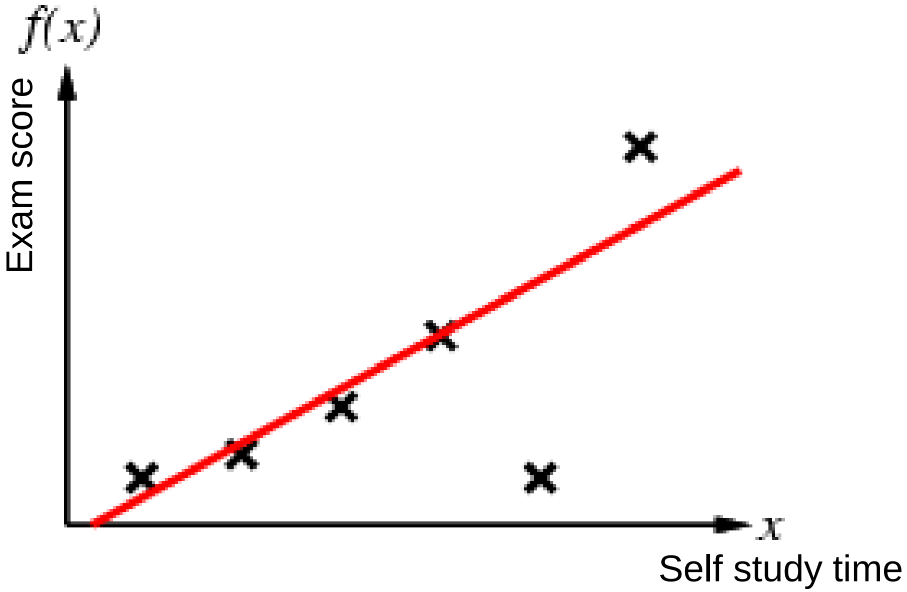
- Construct/adjust h to agree with f on training set
  - » h is *consistent* if it agrees with f on all training data
  - » e.g., curve fitting

f is unknown



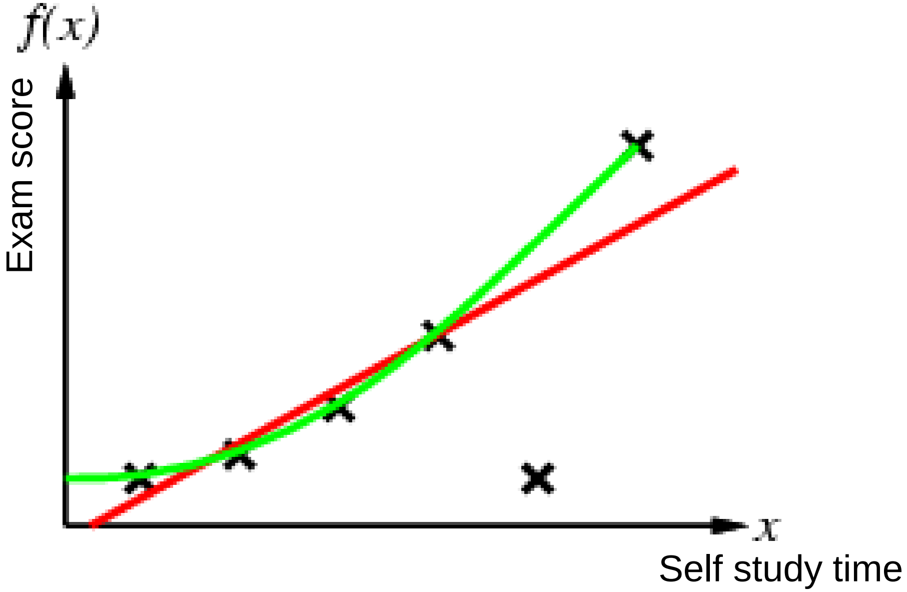
Inductive Learning Method

- Construct/adjust *h* to agree with *f* on training set
  - » *h* is *consistent* if it agrees with *f* on all training data
  - » e.g., curve fitting



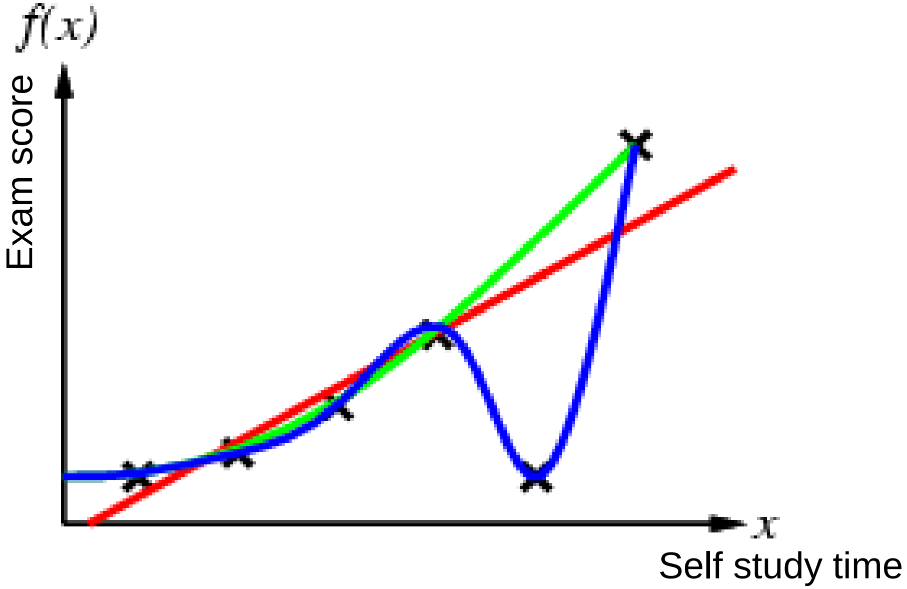
Inductive Learning Method

- Construct/adjust *h* to agree with *f* on training set
  - » *h* is *consistent* if it agrees with *f* on all training data
  - » e.g., curve fitting



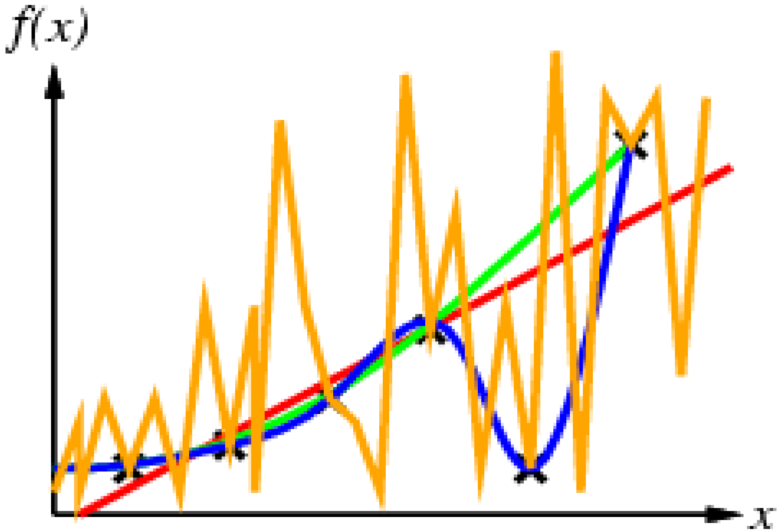
Inductive Learning Method

- Construct/adjust h to agree with f on training set
  - » h is *consistent* if it agrees with f on all training data
  - » e.g., curve fitting



Inductive Learning Method

- Construct/adjust h to agree with f on training set
  - » h is *consistent* if it agrees with f on all training data
  - » e.g., curve fitting



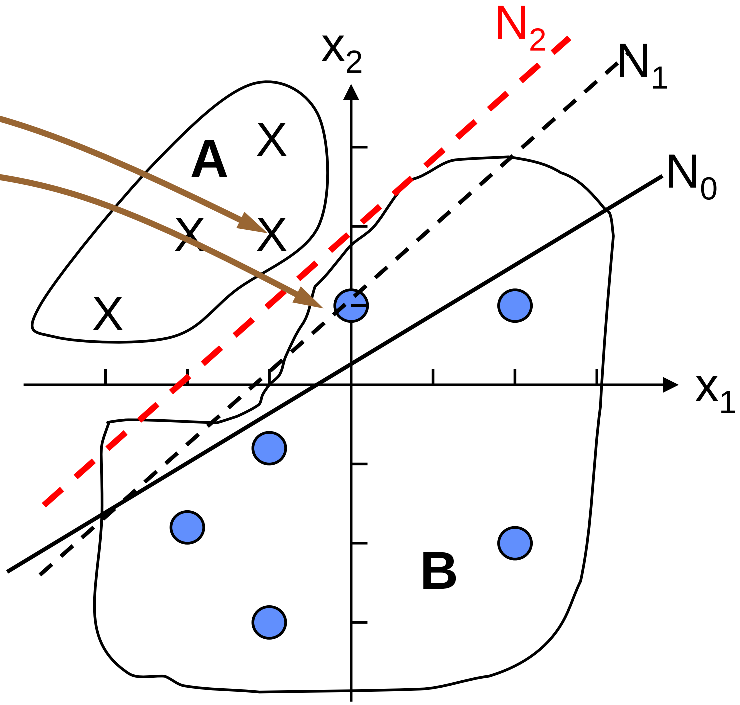
- Ockham’s razor: prefer the simplest hypothesis consistent with data

Generalization

- Hypotheses (a machine learnt model) must generalize to correctly classify new instances, which are not in/seen the training data
- Simply memorizing training samples results a 100% consistent hypothesis, which may not generalize well
- *Occam's razor*:
  - » Finding a *simple* hypothesis helps generalization in new instance prediction.

Perceptron: which hyperplane ?

- Recall the previous example
  - » The separation b/w the hyperplane and the closest data point is called the ***margin of separation***, denoted by ρ
  - » N_2 offers a larger margin of separation than N_1



- ***Optimal hyperplane***
  - » The particular hyperplane with the margin of separation ρ maximized for ***better generalization***

(Note: in some literature, use M to denote margin and M = 2ρ)

Inductive Learning: performance

- When we aim to find a simple hypothesis *approximately* consistent with training data
  - » We are under the ***supervised learning*** paradigm which requires ***labelled*** training data (with correct output given)

    (unsupervised learning will be briefly introduced in Day-7)

- Learning performance = prediction accuracy measured on *test set* (to ensure generalization)

- Training data set vs. Test data set

Reliability of Induction

Reliability of Reasoning

- Deduction
  - » using given knowledge to deduce what conclusion follows, with ***perfect conditional reliability***
    - ♦ It never leads from true premises to a false conclusion
- Inductive rules are rules about what can be inferred from what
  - » There are instances with true “premises” but false “conclusions”
    - ♦ E.g.: You observed all cars entering ISS staff carpark are Japanese model. Mr. A is an ISS staff, owing a Ford (American). He was just on leave on the day.

Reliability of Induction: carpark example

- It might be suggested that we measure the reliability of a rule by percentage of instances with premises that have true conclusion
  - » for the carpark example:
    - ♦ Among *N* ISS staff owned cars, *k* are Japanese cars
      - Therefore “Mr. A owns a Japanese car” may be considered to have reliability of *k*/*N*

Reliability of Induction: how to measure

- If considering only inductive arguments of the form that people have actually made or will make, presumably a finite number
  - » reliability might be measured as the percentage of actual inference of this sort with true premises that also have true conclusions
- Through the actual application of the inductive rule to *estimate* the reliability of rule (i.e. through testing on sample cases)

Reliability of Induction: difficulty

- In general cases, the rule has *infinitely* many instances with true premises
  - ♦ *Infinitely* many of which have false conclusions
  - ♦ *Infinitely* many of which have true conclusions

- Given infinitely many cases of each sort
  - » the percentage of instances with true conclusions is ***not clearly defined***

Example of Induction: poisoned bread

- You are hungry and about to bite into a hot crusty bread.
  - » However, a ‘friend’ stops you and says "Don’t do it. That piece of bread will poison you."
- What do you say in reply?
  - » You would probably reply that "Bread that smells this good has not poisoned me in the past, so it will not poison me now."
- We tend to count such arguments as ***inductively strong***. Moreover, the premise is known to be true.

Example of Induction: poisoned bread (cont.)

- However, no inductively strong argument *guarantees* the truth of its conclusion.
  - » It is *possible* that this piece will poison you even though bread of this kind has not poisoned you in the past.
- So, you weaken your conclusion, and say
  - » "It is *highly probable* that this piece of bread will not poison me."

    — is it rationally justifiable in any objective sense?

- A form of inductive argument is reliable if it yields *approximately true* conclusions *most of the time*.
  - » Simple ***enumerative induction*** provides such a form.

A Look Ahead

- We are interested in finding
  - » an inductive method that will use data to select a rule(*) from a certain set of rules *C*, for classifying new cases on the basis of their observed characteristics.
- Suppose that each rule in *C* has a certain "***expected error***" on new cases.
  - » We want a method for finding the rule in C with the least expected error, given enough data.

(*) The "rule" here is understood in a broad sense as a *hypothesis*, a *model* or a *classifier*

How to Choose a Good Rule ?

- One obvious idea is to select a rule (i.e.: a decision rule, or a model) from *C* with the least error on the ***data***, and then use that rule in order to classify new data.
- This is basically the method of ***enumerative induction*** or simply ***induction***
  - » Offers an unrestricted generalization and reasons *from particular instances to all instances*
- However, it has no guarantee of truth
  - » As the premises in this form of reasoning, even if true, *does not entail* the conclusion's truth

Expected Error of a Rule

- The best of all possible rules is standardly called ***Bayes Rule***, with the least ***expected error***

- To estimate the expected error of a rule from *C*
  - » We need identify the (possibly unknown) frequency of actual errors we will make using the rule
    - ♦ to consider that the expected error of a rule is the (possibly unknown) ***probability*** of error using that rule

- Background probability distribution
- Rule set C

Background Probability

- Claims about actual reliability presuppose
  - » a possibly unknown *objective* ***background statistical probability***

- We want to find a rule from *C*
  - » whose expected error measured by that background probability distribution is as low as possible

☝ However, with unknown background probability, we are actually unable to measure expected error but estimate ***empirical error*** on an available data set

*Restricting Rules

- To select a rule from *C* with the least error on the data
  - There has to be a restriction on what rules are in *C*, but cannot include all possible rules (infinitely many) in *C*
    - must have some sort of ***inductive bias***
      - i.e.: prefer some rules over others
- However, restricting the rules in *C*
  - *runs the risk of not including the Bayes rule*, i.e.: the rule with the least expected error on new cases

***Restricting Rules** (cont.)

- The Vapnik-Chervonenkis dimension, ***VC dimension***, is one of the great discoveries of ***statistical learning theory***
  - » Provides a measure of the "*richness*" of the set of rules
- The higher the VC dimension of rule set (i.e.: the richer)
  - » the more powerful of the rules for complex classification

and

  - » the more difficult to choose a rule from *C* whose *empirical error* on the data is the least, through enumerative induction

(further discussion about statistical learning theory and *VC dimension* is found in "*Support Vector Machines*" in basic elective "*Computational Intelligence I*")

Summary: inductive reasoning & learning

- Need of inductive reasoning
  - » Deduction is about what follows what: premises
  - » Induction is about what can be inferred from what: observations/big-data
- Reliability of induction
  - » Induction has no guaranteed truth
  - » Expected error of a rule/model
  - » Background probability (Often unknown)
  - » Empirical error on available data (estimated error)
- Machine learning using inductive reasoning
  - » Enumerative induction
  - » Reliability of enumerative induction

Data and Information for Learning

Data & Information for Learning

- The reliability of induction is measured in terms of its convergence to the expected error rate of Bayes rule, *given more and more data*

- In this session, we further examine
  - » How to assess variables and evaluate the correlation between variables using data?
    - ♦ Is the correlation coefficient always meaningful?
  - » Is having more data always better than less?

Classification and Correlation

- In general no perfect classification is possible, because there is no perfect correlation (correlation analysis will be further discussed later)
- Some reasons in reality
  - » noise in data, and/or
  - » (human) errors in measurement in the observed features
  - » the relation b/w features and classification may be at best merely probabilistic even apart from issues of noise (think of quantum physics theory)
  - » in estimation, the possibility that the variable depends on other factors than those we use to make our estimate (i.e. Assume conditional independence to simplify modeling)

Variable Assessment

- An essential task in the model-building process is
  - » Assessing the relationship between a ***predictor*** variable and a ***dependent*** variable
    - ♦ E.g.: recognize fruit using color, shape and taste
      - Predictors: Color, Shape, Taste, Weight…
      - Dependent variable: a fruit type: apple, banana…
- If the relationship is identified and tractable, then
  - » the predictor variable is re-expressed to reflect the uncovered relationship, mapping, influence, and
  - » Consequently tested for inclusion into the model

Correlation Coefficient

- Most methods of variable assessment are based on the ***correlation coefficient***

- The correlation coefficient, denoted by r,
  - » is a measure of the strength of the straight-line or *linear relationship* between two variables,
    - ♦ $r \in [-1, +1]$
  - » E.g.: how strong the correlation between the price and the quality of a fruit

Correlation Coefficient: calculation

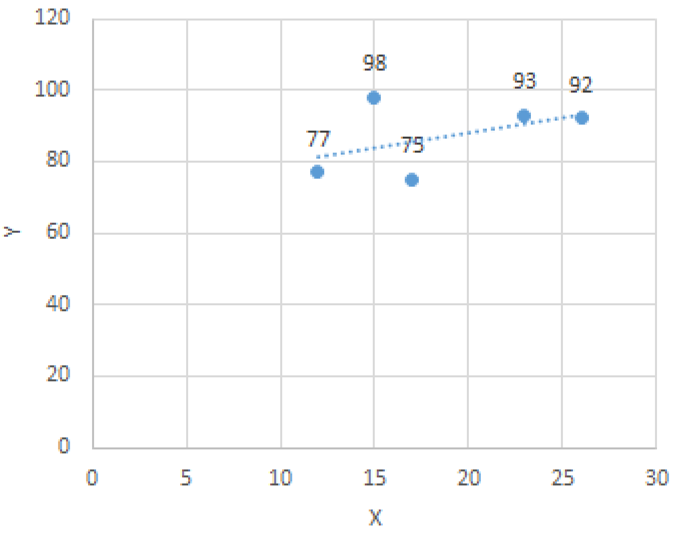
- The calculation of Pearson correlation coefficient for two variables, say X and Y, with sample size *n*
  - ♦ $r_{X,Y} = \sum_i [zX_i * zY_i]/(n-1)$
  - » ***Standardized scores*** of X and Y
    - ♦ $zX_i = [X_i – \text{mean}(X)]/\text{std}(X)$
    - ♦ $zY_i = [Y_i – \text{mean}(Y)]/\text{std}(Y)$
      - std — standard deviation (sample standard deviation)
      - zX and zY are both normalized to have means equal to zero, and standard deviations equal to one.
  - » ***Sample standard deviation***,

$$s = \sqrt{\frac{\sum (X - \overline{X})^2}{n-1}}$$

with sample mean $\overline{X}$

Correlation Coefficient: calculation (cont.)

- Example 6.8:



Obs	X	Y	zX	zY	zX*zY
1	12	77	-1.14	-0.96	1.11
2	15	98	-0.62	1.07	-0.66
3	17	75	-0.27	-1.16	0.32
4	23	93	-0.76	0.58	0.44
5	26	92	1.28	0.48	0.62
mean = 18.6		87			sum = 1.83
std = 5.77		10.32			
n = 5					$r_{x,y}$ = 0.46

$$r_{X,Y} = \Sigma_i[zX_i * zY_i]/(n-1)$$

Correlation Coefficient: interpretation

- The guidelines for interpreting r
  - ♦ 0 — no linear relationship
  - ♦ +1 (-1) — a perfect positive (negative) linear relationship via an exact linear linear/rule/model
  - ♦ b/w 0 and 0.3 (0 and -0.3) — a weak positive (negative) linear relationship via a shaky linear rule
  - ♦ b/w 0.3 and 0.7 (-0.3 and -0.7) — a moderate positive (negative) linear relationship via a fuzzy-firm linear rule
  - ♦ b/w 0.7 and 1.0 (-0.7 and -1.0) — a strong positive (negative) linear relationship via a firm linear rule

Exercise:
Variable assessment

Exercise: data sets

- Given four datasets
- Using excel to find

(a) mean, std, r for each dataset

(b) Get scatterplot for each dataset

obs	X1	Y1	X2	Y2	X3	Y3	X4	Y4
1	10	8.04	10	9.14	10	7.46	8	6.58
2	8	6.95	8	8.14	8	6.77	8	5.76
3	13	7.58	13	8.74	13	12.74	8	7.71
4	9	8.81	9	8.77	9	7.11	8	8.84
5	11	8.33	11	9.26	11	7.81	8	8.47
6	14	9.96	14	8.1	14	8.84	8	7.04
7	6	7.24	6	6.13	6	6.08	8	5.25
8	4	4.26	4	3.1	4	5.39	19	12.5
9	12	10.84	12	9.13	12	8.15	8	5.56
10	7	4.82	7	7.26	7	6.42	8	7.91
11	5	5.68	5	4.75	5	5.73	8	6.89
mean								
std								
r								

Linearity Assumption

- Is correlation coefficient always meaningful?
  - ☝ only when the ***linearity assumption*** is valid
- If the relationship between two variables under discussion is known to be linear, or the observed pattern between the variables appears to be linear,
  - » Then the correlation coefficient provides a reliable measure of the strength of the linear relationship
- Otherwise
  - » The correlation coefficient is **not useful or questionable.**
  - » Not all relationships are linear; there are non-linear ones.

Scatterplots

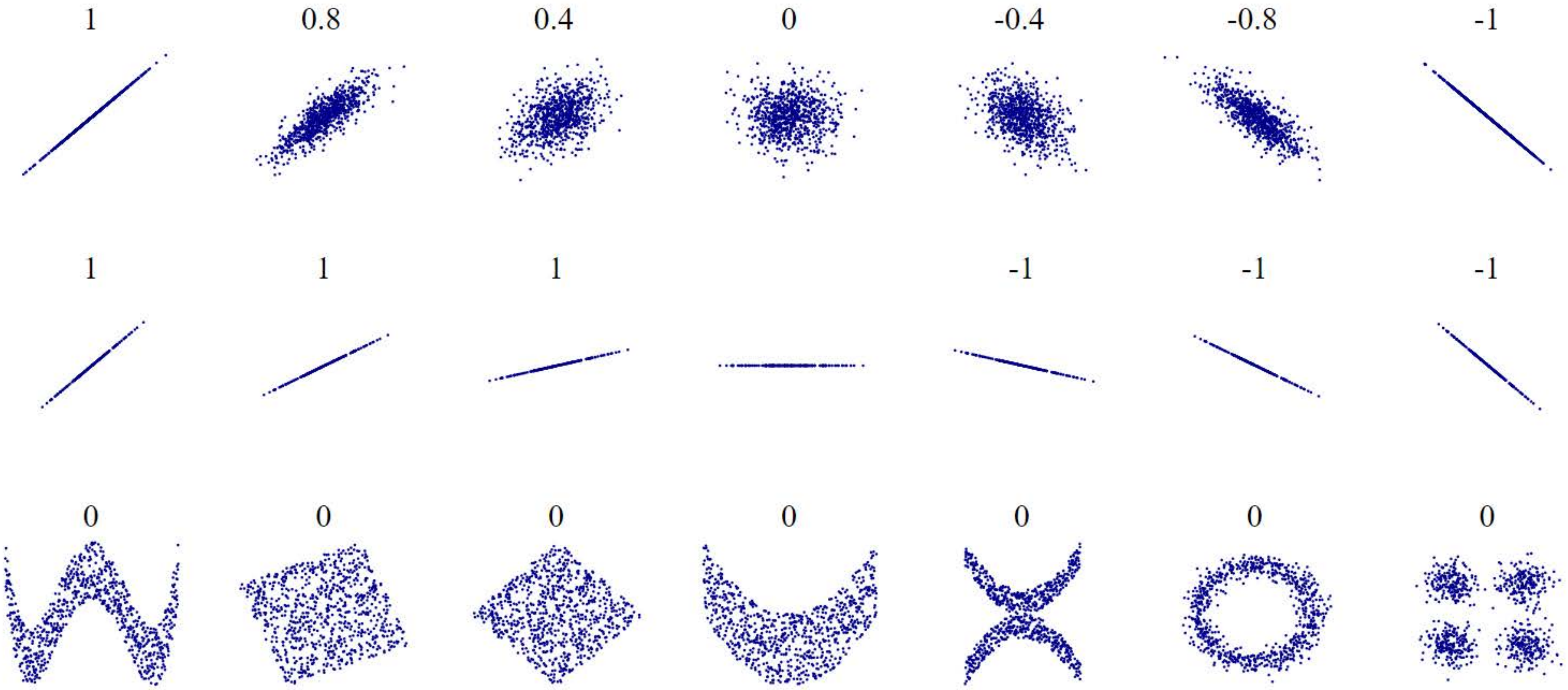
- The linearity assumption of the correlation coefficient can easily be tested with a ***scatterplot***
  - » a mapping of the paired points (X_i, Y_i) in a graph with two orthogonal axes
    - ♦ X and Y are typically assigned as the predictor and dependent variables, respectively
    - ♦ Index i = 1, …, n, where n is the sample size

Expletory Data Analysis EDA

- If the scatter of points in the scatterplot **appear** to overlay a straight-line, the assumption has been satisfied, then $r_{X,Y}$ provides a meaningful measure of the linear relationship between X and Y

Scatterplots & Correlation Coefficient

https://en.wikipedia.org/wiki/Pearson_correlation_coefficient



The More, the Better?

- Data mining is the process of revealing unexpected relationships in data and help unmask the underlying relationships in scatterplots filled with ***Big data***
  - ♦ Big data have rendered the scatterplot overloaded with data points, or information

! Paradoxically, scatterplots based on more information are actually less informative

Scatterplots with Big Data

- With a quantitative target variable
  - » the scatterplot typically becomes a cloud of points with sample-specific variation, called ***rough***, which masks the underlying relationship
- With a qualitative target variable
  - » there is discrete rough, which masks the underlying relationships.

- In either case, if the rough can be removed from the big data scatterplot, then the underlying relationship can be revealed.

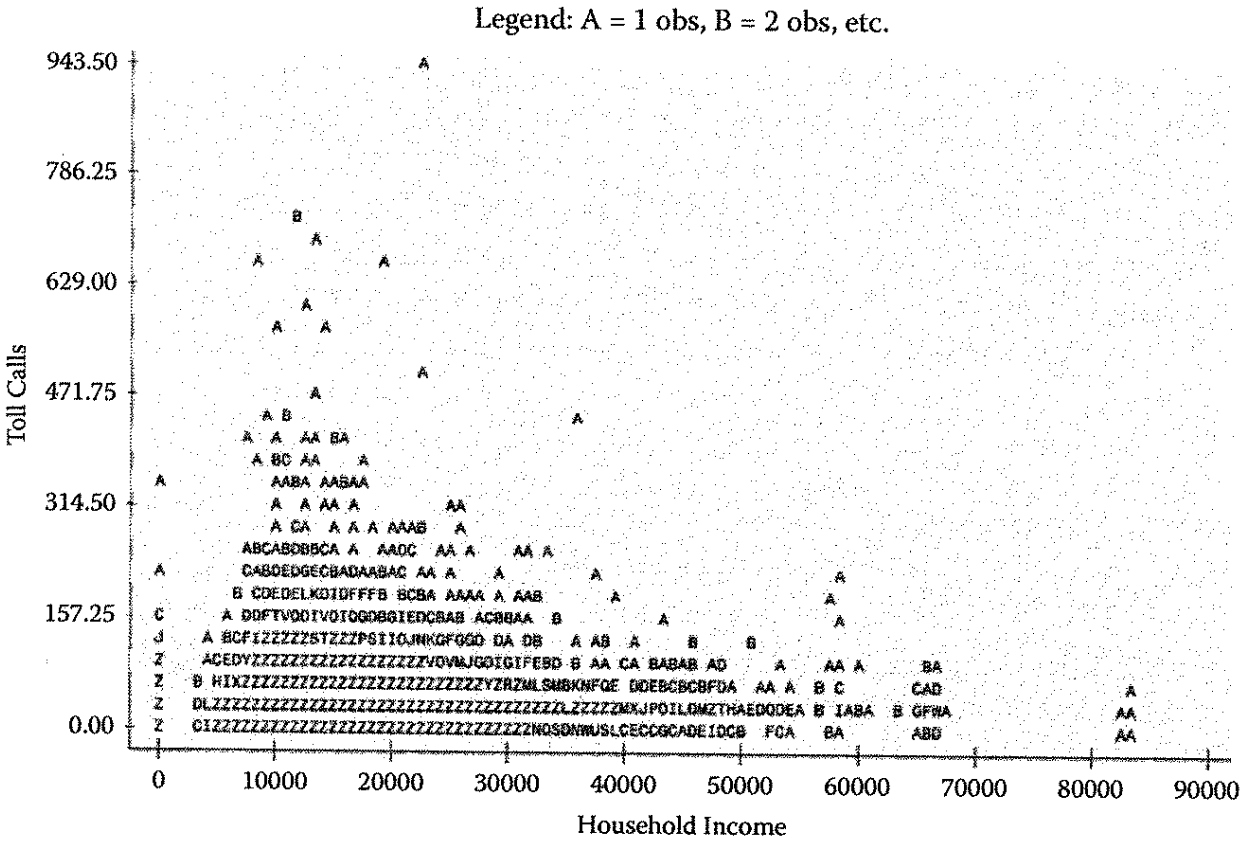
Scatterplots with Big Data (cont.)

- Example 6.9:
  - » Consider the quantitative target variable Toll (phone) Calls (TC) in dollars and the predictor variable Household Income (HI) in dollars from a sample of size 102,000. The calculated $r_{TC,HI}$ is 0.09.
  - » The TC-HI scatterplot shows a cloud of points obscuring the underlying relationship within the data (assuming a relationship exists), and is uninformative regarding an indication for the reliable use of the calculated $r_{TC,HI}$

(from: Bruce Ratner, *Statistical and Machine-Learning Data Mining*)

Scatterplots with Big Data (cont.)

Example 6.9 (cont.)



(Bruce Ratner, *Statistical and Machine-Learning Data Mining,* Figure 2.2)

Smoothed Scatterplot

- A rough-free scatterplot that reveals the underlying relationship in big data.
- The basic steps:
  1. Plot the (X_i, Y_i) data points in an X-Y graph
  2. Divide the X-axis into distinct and non-overlapping neighborhoods (slices, binning)
  3. Take the average of X within each slice
  4. Take the average of Y within each slice
  5. Plot the smooth points (smooth X, smooth Y), constructing a ***smooth scatterplot***
  6. Connect the smooth points, the resultant smooth trace line reveals the underlying relationship b/w X and Y

Smoothed Scatterplot

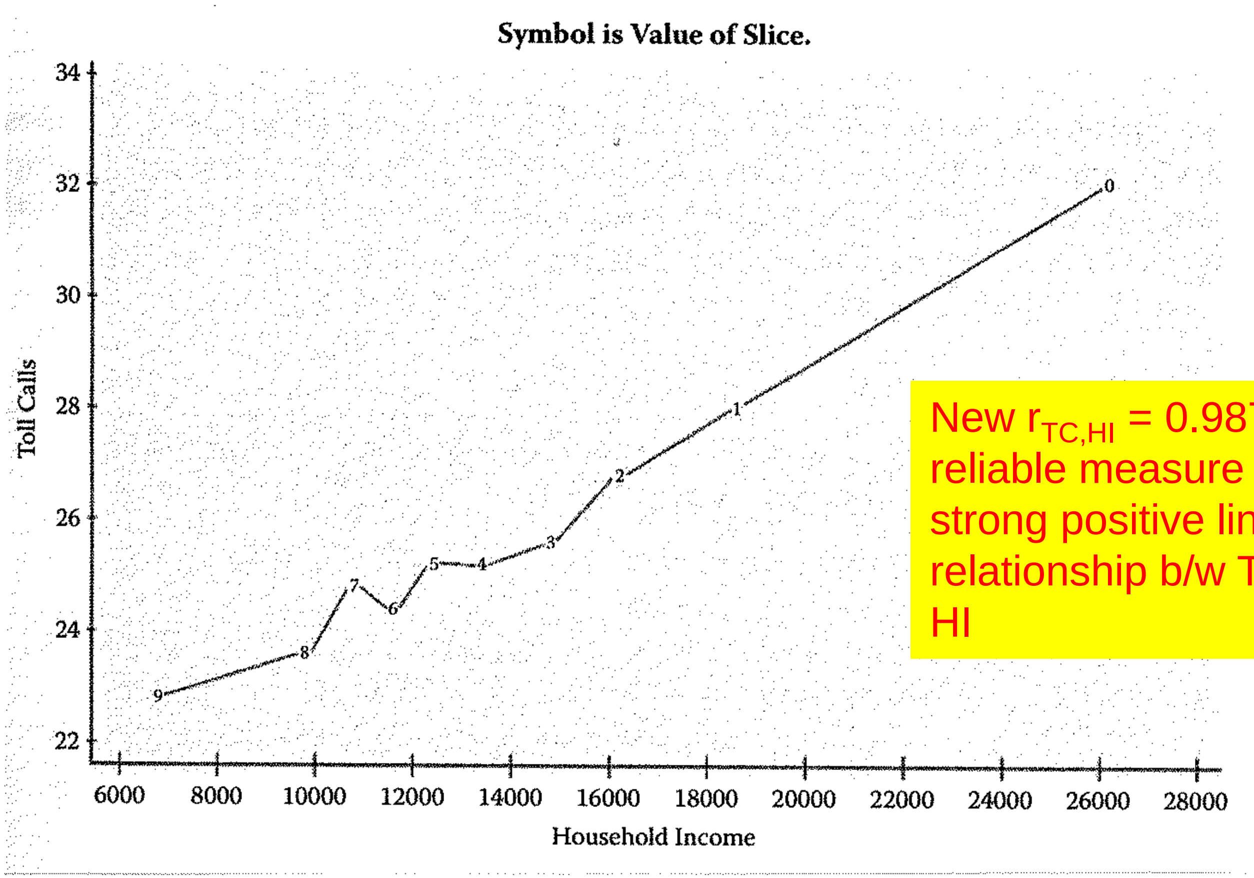
Example 6.9 (cont.)

- » The HI data are grouped into 10equal-size slices, each consisting of 10,200 observations
- » The table shows the averages (smooth points) for HI with TC within the slices (0 ~ 9)
- » The smooth points are then plotted and connected

Slice	Average Toll Calls ($)	Average Household Income ($)
0	31.98	26,157
1	27.95	18,697
2	26.94	16,271
3	25.47	14,712
4	25.04	13,493
5	25.30	12,474
6	24.43	11,644
7	24.84	10,803
8	23.79	9,796
9	22.86	6,748

Smoothed Scatterplot (cont.)

Example 6.9 (cont.)



New $r_{TC,HI}$ = 0.987 is a reliable measure of a very strong positive linear relationship b/w TC and HI

(Bruce Ratner, *Statistical and Machine-Learning Data Mining,* Figure 2.4)

Correlation Coefficient: R-squared

- The correlation coefficient, denoted by r,
  - » is a measure of the strength of the straight-line or linear relationship between two variables
  - » assumes theoretically any value in the closed interval [-1, +1]

- The value of r^2, called the *coefficient of determination*, denoted by ***R-squared***
  - » is typically taken as the percent of variation in one variable explained by the other variable or percent of variation shared between the two variables

Correlation Coefficient: rematching

- The correlation coefficient theoretically assumes values in the closed interval [-1, +1].
  - » However, due to the shapes of data, +1 or -1 may never be possible to reach, so a restricted interval can be better in indicating the correlation
  - » The length of the ***realized correlation coefficient closed interval*** is usually shorter than the theoretical closed interval, and is determined by the process of ***rematching***

Correlation Coefficient: rematching (cont.)

1. Take the original (X, Y) paired data to create new (X, Y) "rematched-paired" data such that
   - » the rematched-paired data produce the strongest positive and strongest negative relationship
     - ♦ To get the strongest positive relationship
       - the highest X value is paired with the highest Y value, the 2nd highest X with the 2nd highest Y value, ...
     - ♦ To get the strongest negative relationship
       - the highest X value is paired with the lowest Y value, the 2nd highest X with the 2nd lowest Y value, …

2. Using rematched-paired data to calculate
   - » $r_{X,Y}$(positive rematch) and $r_{X,Y}$(negative rematch)

Correlation Coefficient: adjusted r

- The realized correlation coefficient closed interval
  - » $[r_{X,Y}(\text{negative rematch}) , r_{X,Y}(\text{positive rematch})]$
- The ***adjusted correlation coefficient***
  - » When original $r_{X,Y}$ is positive
    - ♦ $r_{X,Y}(\text{adjusted}) = r_{X,Y}(\text{original}) / r_{X,Y}(\text{positive rematch})$
  - » When original $r_{X,Y}$ is negative
    - ♦ $r_{X,Y}(\text{adjusted}) = |r_{X,Y}(\text{original})| / r_{X,Y}(\text{negative rematch})$

☝ the adjusted correlate coefficient remains the same sign as the original correlation coefficient

Correlation Coefficient: adjusted r (cont.)

- Example 6.10:
  - » Carrying out rematching on the same data from Example 6.8
  - » The restricted, realized correlation coefficient [-0.99, +0.90]
  - » $r_{X,Y}$(adjusted) = = 0.46/0.90 = 0.51

	Original (X,Y)		Positive Rematch		Negative Rematch	
Obs	X	Y	X	Y	X	Y
1	12	77	26	98	26	75
2	15	98	23	93	23	77
3	17	75	17	92	17	92
4	23	93	15	77	15	93
5	26	92	12	75	12	98
r	0.46		+0.90		-0.99	

Correlation Coefficient: adjusted R-squared

- ***Adjusted R-squared***
  - » Is obtained from adjusted correlation coefficient.
  - » Has the same explanation as for R-squared,
    - ♦ but penalizes the statistic when unnecessary variables are included in the model

(no further discussion)

Implication of Rematching

- The correlation coefficient is ***restricted*** by observed shapes of the X and Y data.
  - » Regardless of the shape of either variable, symmetric or otherwise, if one variable’s shape is different from the other variable’s shape, the correlation coefficient is restricted
- A necessary condition to obtain a perfect correlation is that the shapes must be the same

Correlation: summary

- The *correlation coefficient* *r* is a measure of the strength of the linear relationship between two variables. It is only meaningful when the *linearity assumption* is valid
- The testing of linearity assumption is made easy by *scatterplot* or smoothed scatterplot
  - » When a smoothed scatterplot for big data does not reveal a linear relationship, its scatter can be tested for randomness model or for a general association model
    (not to be covered in this course)

Correlation: summary (cont.)

- Theoretically $r \in [-1, +1]$. However, due to the shapes of data, a *restricted interval* can be better in indicating the correlation.
  - » The restricted interval can be determined by *rematching*
  - » *Adjusted correlation coefficient* is obtained through rematching.
- *R*-squared is often misused as the measure to assess which model produces better prediction
  - » But MSE (mean squared error) is the measure for determining the better model, as well as the minimizing objective for many gradient descent based algorithms (further discussion in other courses)

References

- Harman, G. & S. Kulkarni, “Reliable Reasoning, Inductive and Statistical Learning Theory”, MIT, 2007
- Ratner, B., “Statistical and Machine-Learning Data Mining, Techniques for better Predictive Modeling and Analysis of Big Data”, CRC Press, 2012 (2nd Ed.)
- Kecman, V., “Learning and Soft Computing, Support Vector Machines, Neural Networks, and Fuzzy Logic Models”, MIT, 2001
- Ian Goodfellow, Yoshua Bengio, & Aaron Courville, “Deep Learning”, MIT 2016, **773** pages: http://www.deeplearningbook.org