Data mining, machine learning, and uncertainty reasoning

林偉川

1

課程進行

- 期中考(20%)、期末考(30%)
- 出席率(10%)
- 報告(40%)

2

Application

- Support the DSS (Decision Supporting System) for reasoning ***business intelligent (企業智慧)***
  - Call center data analysis and assessment
    - Inbound call analysis ➔ for customer service
    - Outbound call analysis ➔ for promotion sale or investigation
  - Shopping center data collection and assessment
    - Shopping DM guide → product location reference
    - Shopping habit ➔ customization shopping eDM
  - Support the ERP system for constructing better control of enterprise's resource ➔ purchase reference

Background Knowledge

- Database
- Data warehouse
- Conditional Probability ➔ Bayesian Theorem

$$\mathrm{P}(H \mid E, c) = \frac{\mathrm{P}(H \mid c)\ \mathrm{P}(E \mid H, c)}{\mathrm{P}(E \mid c)}$$

Relative Topic

- AI or Neural Network
- Database or DataWarehouse
- Data Mining or data reasoning
- Machine learning

5

Concept

Program = algorithm + data

+ **domain knowledge**

Applied domain knowledge in suitable data structure is fundamental for solving problems.

Uncertainty propagation will happen in many expert Systems.

Chess playing game uses brute-force approach has led more powerful than AI method because encoding abstract rule into computer is very difficult.

6

Data Mining

Different perspectives of data mining

- Statistics
- Pattern recognition (mining data as patterns)
- Database management system
- Artificial intelligent (encode domain knowledge into abstract rules)

7

Relation

- Data mining is first to mine data from a set of databases (data warehouse)
- Data that are mined is just number which have no meaning to some users ➔ decision making?
- Data can be filtered by machine learning to eliminate some unused data by some machine learning methods
- The data left by the machine learning is needed by users which should be supported by some experts' rules encoding ➔ uncertainty reasoning

8

Uncertainty reasoning

- Vagueness
- Incompleteness (imperfect data dealing)
- Missing value
- Inconsistency

9

What is Learning?

- "Learning denotes changes in a system that ... enable a system to do the same task more efficiently the next time." --Herbert Simon
- "Learning is constructing or modifying representations of what is being experienced." --Ryszard Michalski
- "Learning is making useful changes in our minds." --Marvin Minsky

10

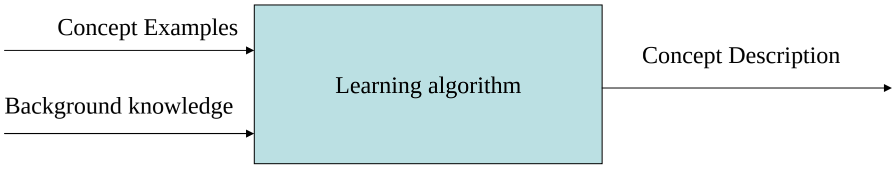
Why do Machine Learning?

- Understand and improve efficiency of human learning For example, use to improve methods for teaching and tutoring people, as done in Computer-aided instruction (CAI)
- Discover new things or structure that is unknown to humans. Example: Data mining
- Fill in skeletal or incomplete specifications about a domain
- Complex AI systems cannot be completely derived by hand and require dynamic updating to incorporate new information. Learning new characteristics expands the expertise and lessens the weak-point of the system

11

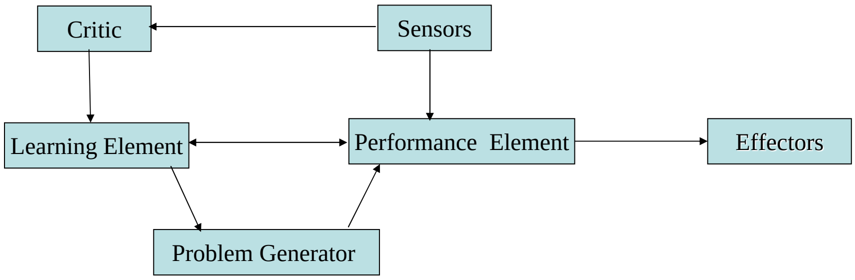
Machine learning

- Application of Machine learning can be achieved by grouping those are acquainted with existing machine learning methods and with the expertise in the given application domain to provide training data



12

Components of a Learning System



- Learning Element makes changes to the system based on how it's doing
- Performance Element is the agent itself that acts in the world
- Critic tells the Learning Element how it is doing (e.g., success or failure) by comparing with a fixed standard of performance
- Problem Generator suggests "problems" or actions that will generate new examples or experiences that will aid in training the system

13

Evaluating learning algorithm **Performance**

Several possible criteria for evaluating a learning algorithm:

- Predictive accuracy of classifier
- Speed of learner
- Speed of classifier

14

Problem solving by computer

- In expert system, programming has been replaced by knowledge-encoding
- Machine learning knowledge encoding is supposed to be replaced by induction from examples (training data)
- The learning system aims at determining a description of a given concept from a set of concept examples provided by the teacher and from the background knowledge

15

Problem solving by computer

- Background knowledge contains the information about the language used to describe the examples and concepts
- A task is to be solved by a computer is how to translate the problem into computational items
- In machine learning, this means how to represent concepts, examples, and the background knowledge ➔by some representation languages

16

Problem solving by computer

- The learning system should be able to deal with imperfections of data ➔ examples will often contain a certain amount of noise – errors in the descriptions or in the classifications
- Examples can be incomplete in the sense that some attribute values are missing ➔ background knowledge need not be perfect

17

Learning algorithm

Classified into two major categories:

- Black-box method such as neural network or mathematical statistical
  - Develop their own concept representation that is to be used for concept recognition purpose
  - Typically involves numerical calculation of coefficients, distances or weights
- Knowledge-oriented method aims at creating symbolic knowledge structures that satisfy the principle of comprehensibility.

18

Distinction criteria of concept learning

- Weak criterion ➔ the system uses sample data to generate an updated basis for improved performance on subsequence data
- Strong criterion ➔ weak criterion is satisfied and the system can communicate its internal updates in explicit symbolic form
- Ultra-strong ➔ both weak and strong criteria are satisfied and the system can communicate its internal updates in an operationally effective symbolic form

19

Distinction criteria of concept learning

- Artificial neural networks and statistical methods satisfies the weak criterion
- Artificial intelligence researcher have been concerned with the strong criterion
- This book is concerned with the knowledge-oriented algorithms capable of developing descriptions understandable to the user.
- Most of these methods are based on manipulating symbolic structures for machine learning to form a concept

20

Cognitive perspective

- Even when concepts can be defined precisely, a correct classification of an object based on the available data may present a difficult problem
- By a cluster, statisticians mean a group of objects that are relatively close to each other according to a chosen numerical distance
- Group of related concepts can often be organized into a generalization hierarchy represented by a tree or graph

21

Cognitive perspective

- Three important notions are suitable to the mutual relations among concept brief exposition: basic-level effect, typicality, and contextual dependency
- In an ordered hierarchy of concepts, one level can be understood as basic.
- The basic means that the concept on this level shares with sub-concepts a large number of features that can be described in recognizable terms

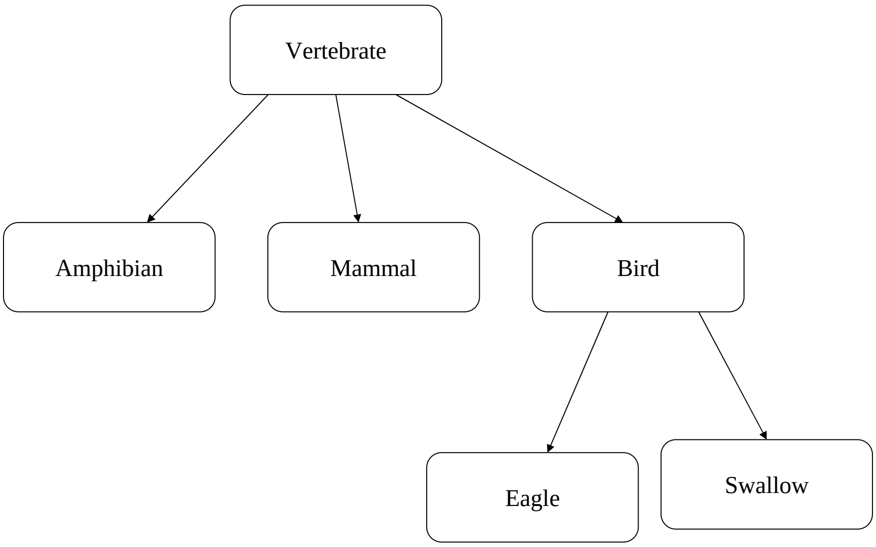
22

Cognitive perspective

- Eagle → bird (basic-level concept:wings,beak) → animal ?→ living being (not basic)
- BMW → car (basic-level concept) → transportation
- The concepts on lower levels can be understood as specializations of the basic-level concepts, whereas concepts on higher levels are often defined as groups of basic-level concepts sharing some important feature

23

Symbolic form



24

Cognitive perspective

- Typicality of distance plays a crucial role for learning a given concept
- Measure typicality by the number of features shared with other sub-concepts and by the number of features inherited from super-concepts (the greater number of inherited features can be found in the instance, the more typical the instance is)
- Real-world concepts are learnable only in an appropriate context (context dependency measurement)

25

Cognitive perspective

- A task to be solved by computer is how to translate the problem into computation terms
- How to represent concepts, examples, and the background knowledge by representation language
- Representation languages include Zero-order logic, attribute-value logic, Horn clauses, and second-order logic

26

Zero order logic

- Propositional calculus
- Examples and concepts are described by conjunction of Boolean constants that stand for individual features (attribute values)

$c \Leftarrow x \wedge y \wedge z$ (an object is an instance of the concept c whenever the conditions x, y, z hold simultaneously)

can_marry_jane $\Leftarrow$ male ^ grown_up ^ single

27

Zero order logic

- It describes only simple concepts and is difficult to capture complex concepts encountered in daily life
- It has low descriptive power and excluded widespread application of zero order logic in machine learning
- Can only be used to illustrate simple algorithms

28

Attributional Logic

- It improves over the zero-order logic is that the attributes are variables that can take on various values.
- * stands for any one, the or linking two or more attribute values is called internal disjunction
- Examples are presented in a table, each row represents an example and each column stands for an attribute ($c \Leftarrow x \wedge y \wedge z$)
- How a car can attract for a young entrepreneur?

Can table analysis

Object	Make	Size	Price	Classification
Car1	European	Big	Affordable	⊕ (positive)
Car2	Japanese	Big	Affordable	⊕
Car3	European	Medium	Affordable	⊗ (negative)
Car4	European	Small	Affordable	⊗
Car5	European	Medium	Expensive	⊕
Car6	Japanese	Medium	Affordable	⊗
Car7	Japanese	Medium	Expensive	⊕
Car8	European	Big	Expensive	⊕

Attributional Logic

- Boolean, numeric, symbolic, or mixed-value attributes can be considered, the scope of their value is often constrained by background knowledge
- Attributional logic (variable-value logic) is more practical than zero-order logic and provided the basis for machine learning algorithms (TDIDT '86 or AQ '83)
- A formal basis for such a description language was defined in variable-valued logic

31

First Order predicate Logic definition

FOL (Horn clause) supplies these primitives:

- **Variable symbols**. E.g., x, y
- **Connectives**. Same as in PL: not (~), and (^), or (v), implies (=>), if and only if (<=>)

32

FOL definition

- **Quantifiers**: Universal (∀) and Existential (∃)
  - Universal quantification corresponds to conjunction in that (∀ x)P(x) means that P holds for all values of x in the domain associated with that variable. E.g., (∀ x) dolphin(x) => mammal(x)
  - Universal quantifiers usually used with "implies" to form "if-then rules." E.g., (∀ x) IE-student(x) => smart(x) means "All IE students are smart." You can use universal quantification to make blanket statements about every individual in the world: (∀ x)IE-student(x) ^ smart(x) meaning that everyone in the world is a IE student and is smart.

33

FOL definition

- Existential quantifiers usually used with "and" to specify a list of properties or facts about an individual.
  - E.g., (∃ x) IE-student(x) ^ smart(x) means "there is a IE student who is smart." A common mistake is to represent this English sentence as the FOL sentence: (∃ x) IE-student(x) => smart(x). But consider what happens when there is a person who is NOT a IE-student.
  - Existential quantification corresponds to disjunction in that (∃ x)P(x) means that P holds for some value of x in the domain associated with that variable. E.g., (∃ x) mammal(x) ^ lays-eggs(x)

34

FOL definition

- Switching the order of universal quantifiers does not change the meaning: $(\forall\ x)(\forall\ y)\ P(x,y)$ is logically equivalent to $(\forall\ y)(\forall\ x)\ P(x,y)$. Similarly, you can switch the order of existential quantifiers.
- Switching the order of universals and existentials *does* change meaning:
  - Everyone likes someone: $(\forall\ x)(\exists\ y)$ likes(x,y)
  - Someone is liked by everyone: $(\exists\ y)(\forall\ x)$ likes(x,y)

35

Sentences are built up from terms and atoms

- A **term** (denoting a real-world individual) is a constant symbol, a variable symbol, or an n-place function of n terms. For example, x and $f(x_1, ..., x_n)$ are terms, where each x_i is a term.
- An **atom** (which has value true or false) is either an n-place predicate of n terms, or, if P and Q are atoms, then ~P, P V Q, P ^ Q, P => Q, P <=> Q are atoms

36

Sentences are built up from terms and atoms

- A **sentence** is an atom, or, if P is a sentence and x is a variable, then (∀ x)P and (∃ x)P are sentences
- A **well-formed formula (wff)** is a sentence containing no "free" variables. i.e., all variables are "bound" by universal or existential quantifiers. E.g., (∀ x)P(x,y) has x bound as a universally quantified variable, but y is free.

37

Translating English to FOL

- Every gardener likes the sun.
  (∀ x) gardener(x) => likes(x,Sun)
- You can fool some of the people all of the time.
  (∃ x)(∀ t) (person(x) ^ time(t)) => can-fool(x,t)
- You can fool all of the people some of the time.
  (∀ x)(∃ t) (person(x) ^ time(t) => can-fool(x,t)
- All purple mushrooms are poisonous.
  (∀ x) (mushroom(x) ^ purple(x)) => poisonous(x)

38

Translating English to FOL

- No purple mushroom is poisonous.
  ~(∃ x) purple(x) ^ mushroom(x) ^ poisonous(x)
  or, equivalently,
  (∀ x) (mushroom(x) ^ purple(x)) => ~poisonous(x)
- There are exactly two purple mushrooms.
  (∃ x)(∃ y) mushroom(x) ^ purple(x) ^ mushroom(y) ^ purple(y) ^ ~(x=y) ^ (∀ z) (mushroom(z) ^ purple(z)) => ((x=z) v (y=z))

39

Translating English to FOL

- Deb is not tall.
  ~tall(Deb)
- X is above Y if X is on directly on top of Y or else there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.
  (∀ x)(∀ y) above(x,y) <=> (on(x,y) v (∃ z) (on(x,z) ^ above(z,y)))

40

First Order predicate Logic – Horn Clause

- Inference Rule -- A Horn clause (FOL) consists of a head and a body as illustrated by the following definition:

  grandparent(x,y) :- parent(x,z), parent(z,y)
- The left part of :- is called the head, and the right part of :- is the body, the commas stands for conjunctions of universal quantified
- Grandparent and parent are called predicate and the variables in the parentheses are called arguments

41

First Order predicate Logic – Horn Clause

- If all predicates have precisely one argument, the language reduces into attribute-value logic
- If all predicates have precisely zero argument, the language reduces into zero-order logic
- Horn clauses constitute an advanced representation language that facilitates very complex description
- This form is the basis of a programming language → Prolog

42

Second-Order Logic(SOL)

- Built on the idea of the predicate name can be considered as variables

$p(x,y) :- q(x,xw) \wedge q(y,yw) \wedge r(xw,yw)$

for examples

1. brothers(x,y) :- son(x,xw) ^ son(y,yw) ^ equal(xw,yw)

Θ = {p=brothers, q=son, r=equal}

2. lighter(x,y) :- weight(x,xw) ^ weight(y,yw) ^ less(xw,yw)

Θ = {p=lighter, q=weight, r=less}

➔ groups of concepts share the same structure of admissible descriptions

➔ SOL is successfully used such structures to assist the search for the concept

43

Explicit constrained languages

- Representation languages based on logic are so rich and flexible that their use for machine learning is computational intractable
- Some constraints such as a limited number of predicates in the clause, a limited number of predicate arguments, excluded recursive definitions

44

Explicit constrained languages

- The number of variables in the body of the clause is not allowed to exceed a predefined threshold (only those variables can appear in the body that have already appeared in the head of the clause)

ancestor(x,y) :- parent(x,z), parent(z,y)   in FOL

ancestor(x,y) :- parent(x,y) should be added to avoid recursion

ancestor(x,y) :- parent(x,z), ancestor(z,y)

45