Overview

Jihoon Yang

Machine Learning Research Laboratory
Department of Computer Science & Engineering
Sogang University

Data Mining

- *We are drowning in information and starving for knowledge* – John Naisbitt
- We are entering the era of **big data**
  - Gartner's definition of 3Vs: Volume, Velocity, Variety
  - Big Data initiative of Obama in 2012, $200M
- Refers to *extracting* or *mining* knowledge from large amounts of data; So large or complex for traditional data processing applications are inadequate
- Technological Driving Factors
  - Larger, cheaper memory
  - Faster, cheaper processors
  - Success of DBs and the Web
  - New ideas in machine learning/statistics

Statistics vs. Data Mining vs. Machine Learning

- Traditional statistics
  - First hypothesize, then collect data, then analyse
  - Often model-oriented (strong parametric models)
- Data mining
  - Few if any a priori hypotheses
  - Data is usually already collected a priori
  - Analysis is typically data-driven not hypothesis-driven
  - Often algorithm-oriented rather than model-oriented
  - Statistical ideas are very useful in data mining (e.g. in validating whether discovered knowledge is useful)
  - Data mining relies heavily on ideas from machine learning
  - More emphasis on scalability (e.g. algorithms that can work on outside main memory data, or streaming data)
  - Somewhat more application-oriented: higher visibility in industry and in public, while ML is somewhat more theoretical, research oriented (i.e. emphasis on automating the discovery of regularities from data, characterizing what can be learned and under what conditions, obtaining guarantees regarding quality of learned models)

Machine Learning

- Algorithms or computation or information processing provide for study of cognition and life what calculus provided for physics
- We have a theory of intelligent behavior when we have precise information processing models (computer programs) that produce such behaviour
- We will have a theory of **learning** when we have precise information processing models of learning (computer programs that learn from experience)

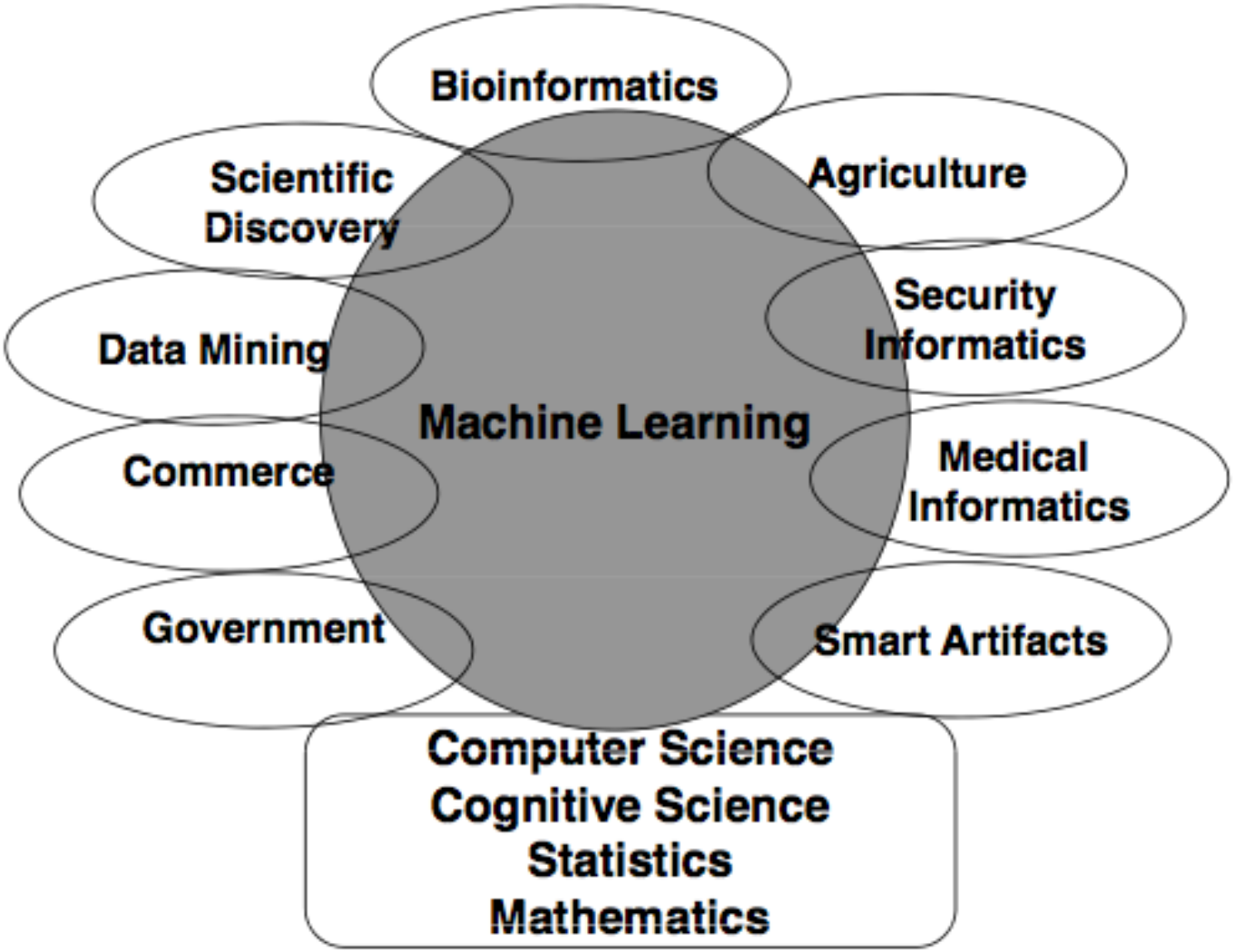
Why should machines learn?

- Intelligent behavior requires knowledge
- Explicitly specifying the knowledge needed for specific tasks is hard, and often infeasible
- Some tasks are best specified by examples (e.g. medical diagnosis, credit risk assessment)
- Buried in large volumes of data are useful predictive relationships (data mining)
- Machine learning is most useful when
  - The structure of the task is not well understood but a representative dataset is available
  - Task (or its parameters) change dynamically
- If we can program computers to learn from experience, we can
  - Dramatically enhance the usability of software (e.g. personalised information assistants)
  - Dramatically reduce the cost of software development (e.g. for medical diagnosis)
  - Automate data driven discovery (e.g. bioinformatics, social informatics)

ML Applications

- Medical diagnosis/image analysis (e.g. pneumonia)
- Spam filtering, fraud detection (e.g. credit cards, phone calls)
- Search and recommendation (e.g. google, amazon)
- Automatic speech recognition & speaker verification
- Locating/tracking/identifying objects in images & videos (e.g. faces)
- Printed and handwritten text parsing
- Driving computer players in games
- Computational molecular biology (e.g. gene expression analysis)
- Autonomous driving
- ...

ML in Context



What is ML?

- A program M is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance as measured by P on tasks in T in an environment Z with experience E

Examples

1. T: cancer diagnosis
   E: a set of diagnosed cases
   P: accuracy of diagnosis on new cases
   Z: noisy measurements, occasionally misdiagnosed training cases
   M: a program that runs on a general purpose computer

What is ML?

2. T: annotating protein sequences with function labels
   E: a data set of annotated protein sequences
   P: score on a test set not seen during training (e.g. accuracy of annotations)

3. T: driving on the interstate
   E: a sequence of sensor measurements and driving actions recorded while observing an expert driver
   P: mean distance traveled before an error as judged by a human expert

Canonical learning problems

- *Supervised learning*: given examples of inputs and corresponding desired outputs, predict outputs on future inputs
  - Classification
  - Regression
  - Time series prediction
- *Unsupervised learning*: given only inputs, automatically discover representations, features, structures, etc.
  - Clustering
  - Outlier detection
  - Compression
- *Reinforcement learning*: given sequences of inputs, actions from a fixed set, and scalar rewards/punishments, learn to select actions in a way that maximises expected reward

Machine Learning

- Learning involves synthesis or adaption of computational structures:
  - Classifiers
  - Functions
  - Logic Programs
  - Rules
  - Grammars
  - Probability distributions
  - Action policies

ML = Inference + Data Structures + Algorithms

Learning input-output functions

- *Target function f*: unknown to the learner – $f \in F$
- Learner's *hypothesis* about what f might be – $h \in H$, *Hypothesis space*
- *Instance space X*: domain of f, h
- *Output space Y*: range of f, h
- *Example*: an ordered pair (x, y) where $x \in X$ and $f(x) = y \in Y$
- F and H *may* or *may not* be the same!
- *Training set E*: a multi-set of examples
- *Learning algorithm L*: a procedure which given some E, outputs an $h \in H$

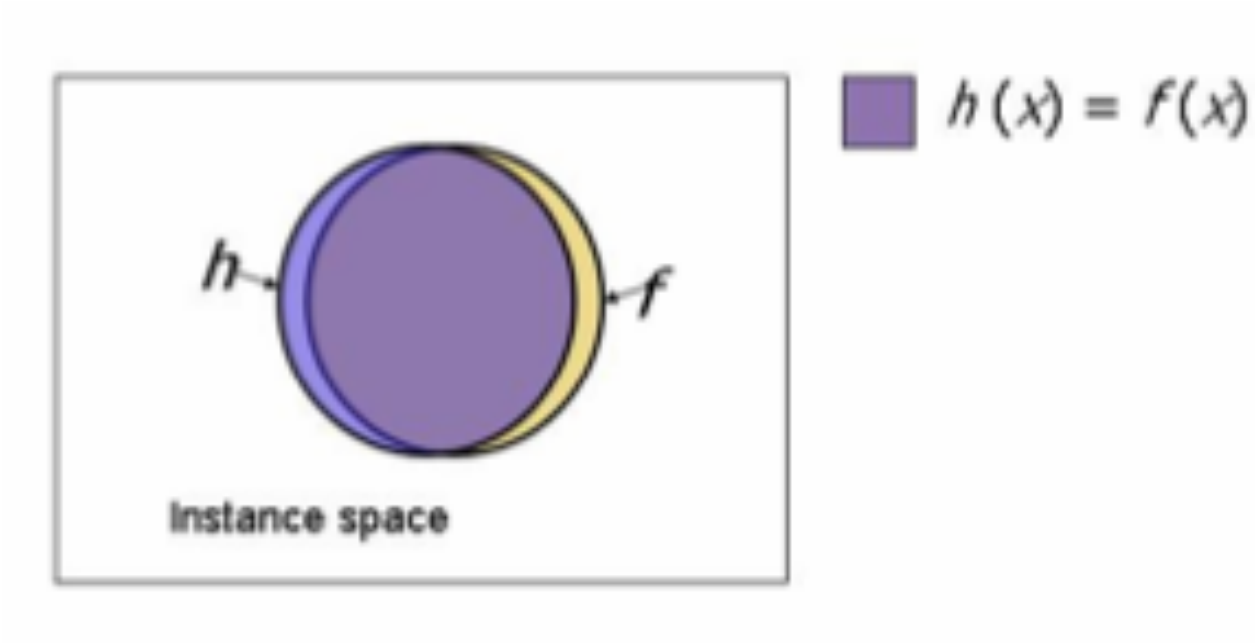
Learning input-output functions

- Training regime
  - Batch
  - Online
  - Distributed
    - Vertical fragmentation
    - Horizontal fragmentation
- Noise
  - Attribute noise
  - Classification noise
  - Both

Inductive learning

- **Premise**: A hypothesis (e.g. a classifier) that is consistent with a sufficiently large number of representative training examples is likely to accurately classify novel instances drawn from the same universe
- We can prove this is an optimal approach (under appropriate assumptions)
- When the number of examples is limited, the learner needs to be smarter (e.g. find a concise hypothesis that is consistent with the data)

Measuring classifier performance



Measuring classifier performance

N: Total number of instances in the data set
$TP(c)$: True Positives for class c, $FP(c)$: False Positives for class c
$TN(c)$: True Negatives for class c, $FN(c)$: False Negatives for class c
TP: True Positives over all classes

$$Accuracy = TP/N$$
$$Precision/Specificity(c) = \frac{TP(c)}{TP(c)+FP(c)}$$
$$Recall/Sensitivity(c) = \frac{TP(c)}{TP(c)+FN(c)}$$
$$False\ Alarm(c) = \frac{FP(c)}{TP(c)+FP(c)} = 1 - Precision(c)$$

Inductive bias

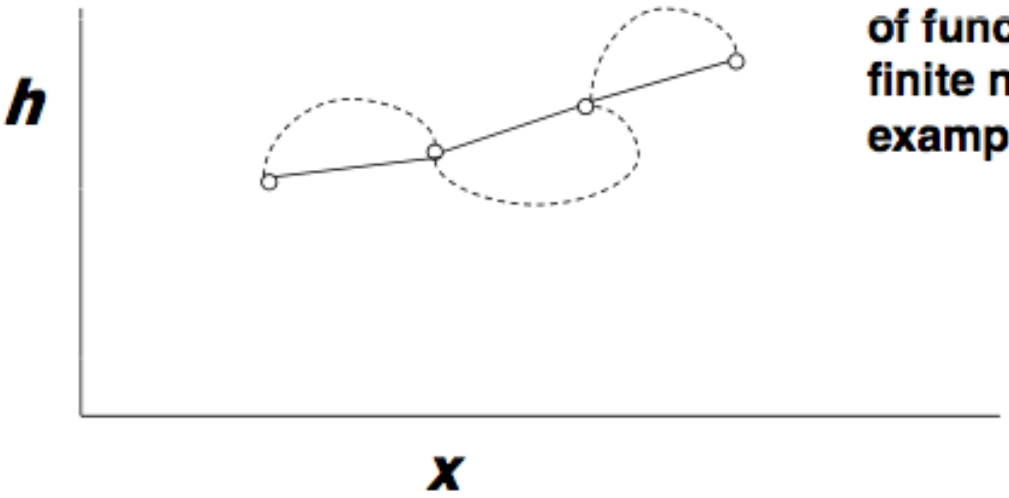
- Consider a concept learning algorithm L for the set of instances X. Let c be an arbitrary concept defined over X, and let $D_c = \{\langle x, c(x)\rangle\}$ be an arbitrary set of training examples of c. Let $L(x_i, D_c)$ denote the classification assigned to the instance x_i by L after training on the data D_c.
- The *inductive bias* of L is any minimal set of assertions B such that for any target concept c and corresponding training examples D_c

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)]$$

- In other words, the set of assumptions, that together with the training data, deductively justify the classifications assigned by the learner to future instances

Function learning and bias

Example



There is an infinite number of functions that match any finite number of training examples!

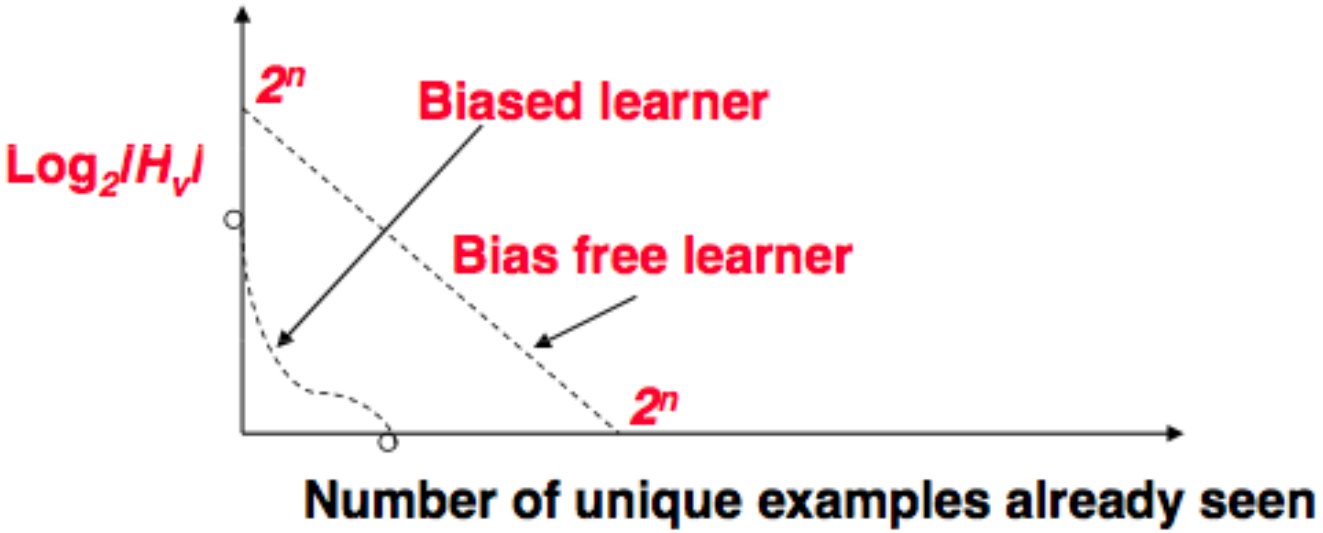
Bias free function learning is impossible!

Learning and bias

Suppose H = set of all n-input Boolean functions.
Suppose the learner is unbiassed. Then

$$|H| = 2^{2^n}$$

H_V = *version space* – the subset of H not yet ruled out by the learner



Learning and bias

- Weaker bias
  $\rightarrow$ more open to experience, flexible
  $\rightarrow$ more expressive hypothesis representation

- *Occam's razor*
  - Simple hypotheses preferred
  - Linear fit preferred to quadratic fit assuming both yield relatively good fit over the training examples

- *Learning in practice requires a trade-off between complexity of hypothesis and goodness of fit*; How this trade-off is done affects the learner's ability to generalise

Course Objectives

- Understand, implement, and use ML algorithms to solve practical problems
- Make intelligent choices among learning algorithms for specific applications
- Formulate and solve new ML problems combining or adapting elements of existing algorithms
- Analyze learning algorithms (e.g. performance guarantees) and distinguish between easy and hard learning problems
- Gain adequate background to understand current literature
- Gain an understanding of the current state of the art in ML
- Learn to conduct original research in ML