HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

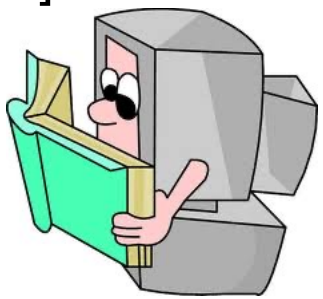
Introduction to
Data Science
(IT4142E)

Contents

- Lecture 1: Overview of Data Science
- Lecture 2: Data crawling and preprocessing
- Lecture 3: Data cleaning and integration
- Lecture 4: Exploratory data analysis
- Lecture 5: Data visualization
- Lecture 6: Multivariate data visualization
- **Lecture 7: Machine learning**
- Lecture 8: Big data analysis
- Lecture 9: Capstone Project guidance
- Lecture 10+11: Text, image, graph analysis
- Lecture 12: Evaluation of analysis results

What is Machine Learning?

- Machine Learning (ML) is an active subfield of Artificial Intelligence.
- ML seeks to answer the question [Mitchell, 2006]
  - *How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?*
- Some other views on ML:
  - Build systems that automatically improve their performance [Simon, 1983].
  - Program computers to optimize a performance objective at some task, based on data and past experience [Alpaydin, 2020]

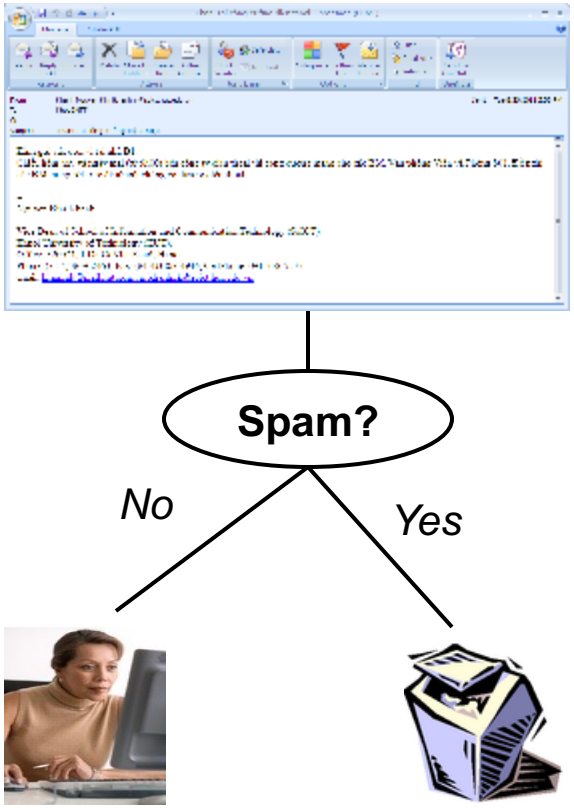


A learning machine

- We say that a machine *learns* if the system reliably improves its performance **P** at task **T**, following experience **E**.
- A *learning problem* can be described as a triple (**P, T, E**).
- ML is close to and intersects with many areas.
  - Computer Science,
  - Statistics, Probability,
  - Optimization,
  - Psychology, Neuroscience,
  - Computer Vision,
  - Economics, Biology, Bioinformatics, …

Some real examples (1)

- Spam filtering for emails
  - **T**: filter/predict all emails that are spam.
  - **P**: the accuracy of prediction, that is the percentage of emails that are correctly classified into normal/spam.
  - **E**: set of old emails, each with a label of spam/normal.



Some real examples (2)

- Image tagging
  - **T**: give some words that describe the meaning of a picture.
  - **P**: ?
  - **E**: set of pictures, each has been labelled with a set of words.

FISH WATER OCEAN
TREE CORAL

PEOPLE MARKET PATTERN
TEXTILE DISPLAY

BIRDS NEST TREE
BRANCH LEAVES

What does a machine learn?

- A mapping (function):

$$y^*: x \longmapsto y$$

  - x: observations (data), past experience
  - y: prediction, new knowledge, new experience,…

Where does a machine learn from?

- Learn from a set of training examples (**training set,** tập học, tập huấn luyện) $\{x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_M\}$
  - x_i is an observation (quan sát, mẫu, điểm dữ liệu) of x in the past.
  - y_j is an observation of y in the past, often called *label (nhãn)* or *response (phản hồi)* or *output (đầu ra)*.
- After learning:
  - We obtain a model, new knowledge, or new experience (f).
  - We can use that model/function to do **prediction** or **inference** for future observations, e.g.,

$$y = f(x)$$

Two basic learning problems

- There is an *unknown* function y^* that maps each x to a number $y^*(x)$
  - In practice, we can collect some pairs: (x_i, y_i), where $y_i = y^*(x_i)$
- **Supervised learning (học có giám sát):** find the true function y^* from a given training set {x_1, x_2, …, x_N, y_1, y_2,…, y_N}.
  - ***Classification*** (categorization, phân loại, phân lớp): if y only belongs to a discrete set, for example {spam, normal}
  - ***Regression*** (hồi quy): if y is a real number

Supervised learning: Regression

- Prediction of stock indices

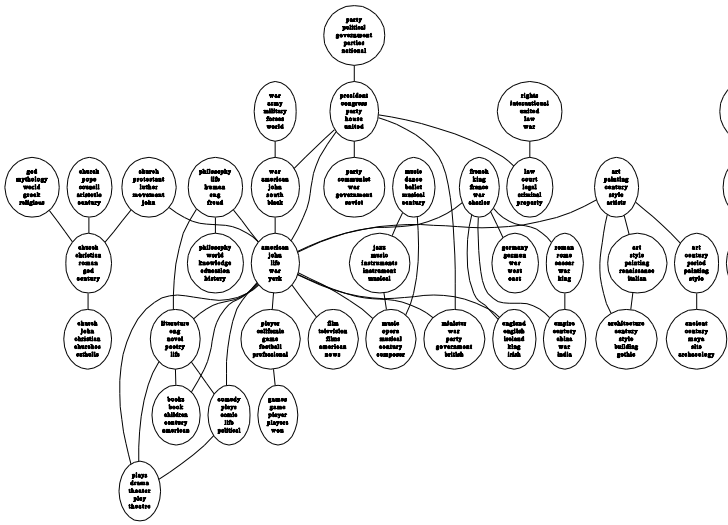
Supervised learning: classification

- ***Multiclass*** *classification (phân loại nhiều lớp):* when the output y is one of the pre-defined labels $\{c_1, c_2, \ldots, c_L\}$ (mỗi đầu ra chỉ thuộc 1 lớp, mỗi quan sát x chỉ có 1 nhãn)
  - Spam filtering: y in {spam, normal}
  - Financial risk estimation: y in {high, normal, no}
  - Discovery of network attacks: ?
- ***Multilabel*** *classification (phân loại đa nhãn):* when the output y is a subset of labels (mỗi đầu ra là một tập nhỏ các lớp; mỗi quan sát x có thể có nhiều nhãn)
  - Image tagging: y = {birds, nest, tree}
  - sentiment analysis

BIRDS NEST TREE

Two basic learning problems

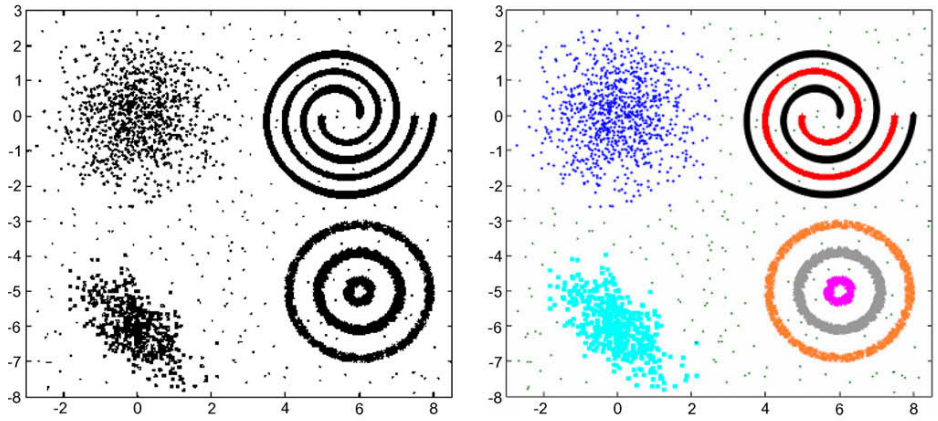
- **Unsupervised learning (học không giám sát):** find the true function y^* from a given training set $\{x_1, x_2, \ldots, x_N\}$.
  - y^* can be a data cluster
  - y^* can be a hidden structure
  - y^* can be a trend, …

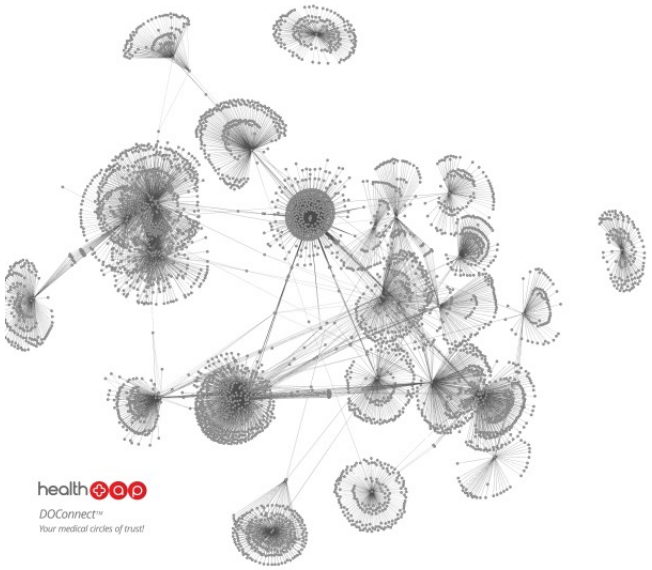


- Other learning problems:
  - semi-supervised learning,
  - reinforcement learning,
  - …

Unsupervised learning: examples (1)

- Clustering data into clusters
  - Discover the data groups/clusters

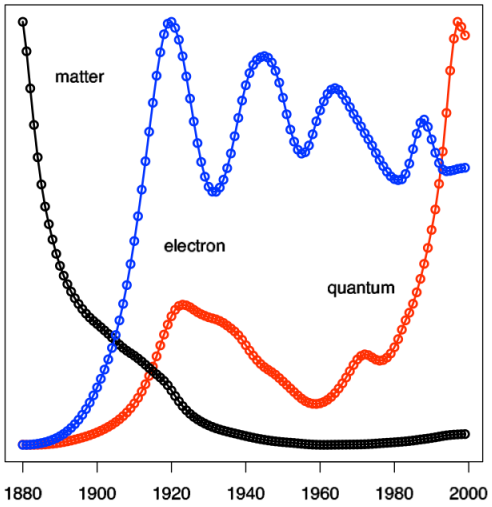




- Community detection
  - Detect communities in online social networks

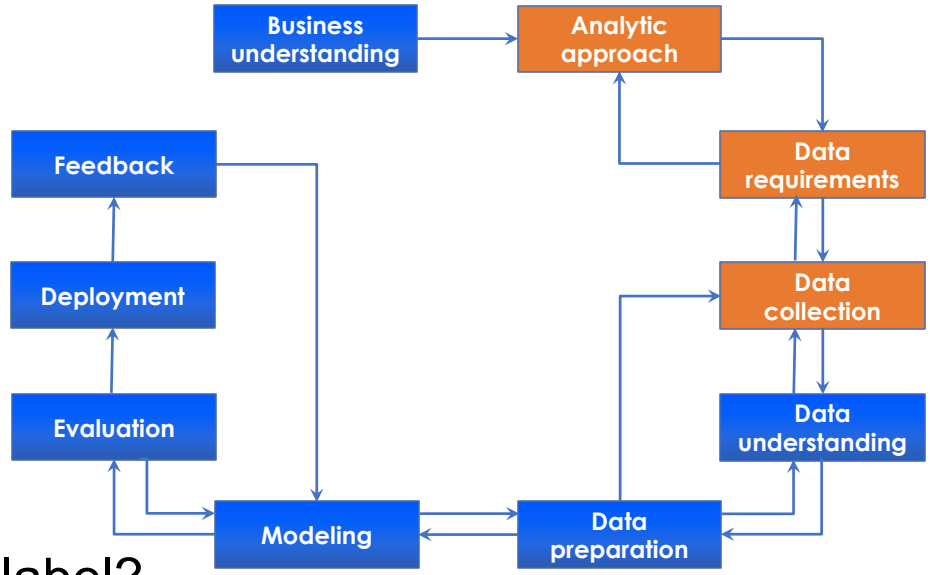
Unsupervised learning: examples (2)

- Trends detection
  - Discover the trends, demands, future needs of online users



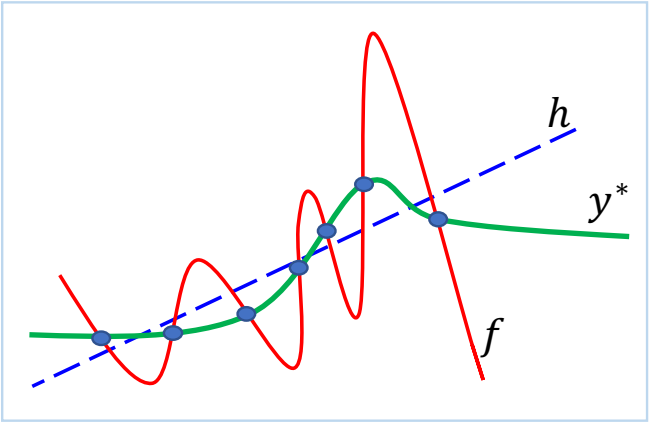
Design a learning system (1)

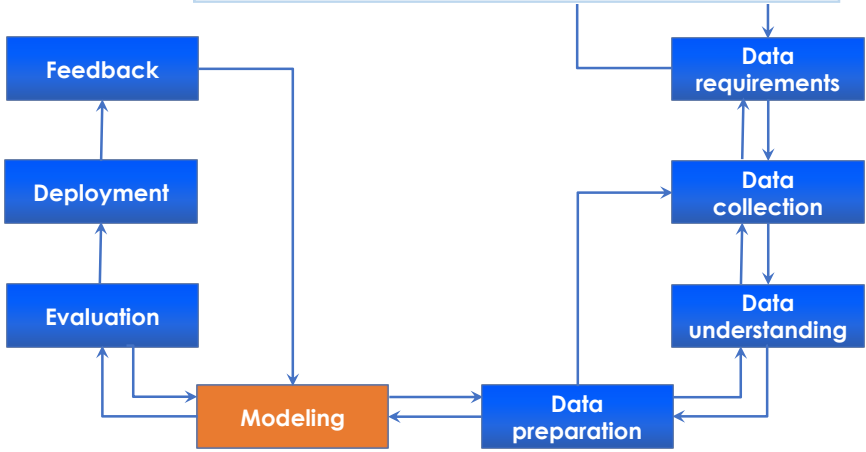
- Some issues should be carefully considered when designing a learning system.
- Determine the type of the function to be learned
  (Xác định **dạng bài toán học**)
  - $y^*: X \rightarrow \{0,1\}$
  - y^*: X → set of labels/tags
- Collect a **training set**:
  - Do the observations have any label?
  - The training set plays the key role in the effectiveness of the system.
  - The training observations should characterize the whole data space →good for future predictions.



Design a learning system (2)

- Select a representation or approximation (**model**) *f* for the unknown function y* (Chọn dạng hàm *f* để xấp xỉ hàm *y** chưa biết)
  - Linear model?
  - A neural network?
  - A decision tree? …
- Select a **learning algorithm** to find *f*:
  - Ordinary least square? Ridge?
  - Backpropagation?
  - ID3? …





ML: some issues (1)

- **Learning algorithm**
  - Often an iterative algorithm
  - Under what conditions the chosen algorithm will (asymtotically) converge?
  - For a given application/domain and a given objective function, what algorithm performs best?
- *No-free-lunch theorem* [Wolpert and Macready, 1997]:
  if an algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems.
  - *No algorithm can beat another on all domains.*
    (không có thuật toán nào luôn hiệu quả nhất trên mọi miền ứng dụng)

ML: some issues (2)

- **Training data**
  - *How many observations* are enough for learning?
  - Whether or not does the *size of the training set* affect performance of an ML system?
  - What is the effect of the *corrupted* or *noisy* observations?

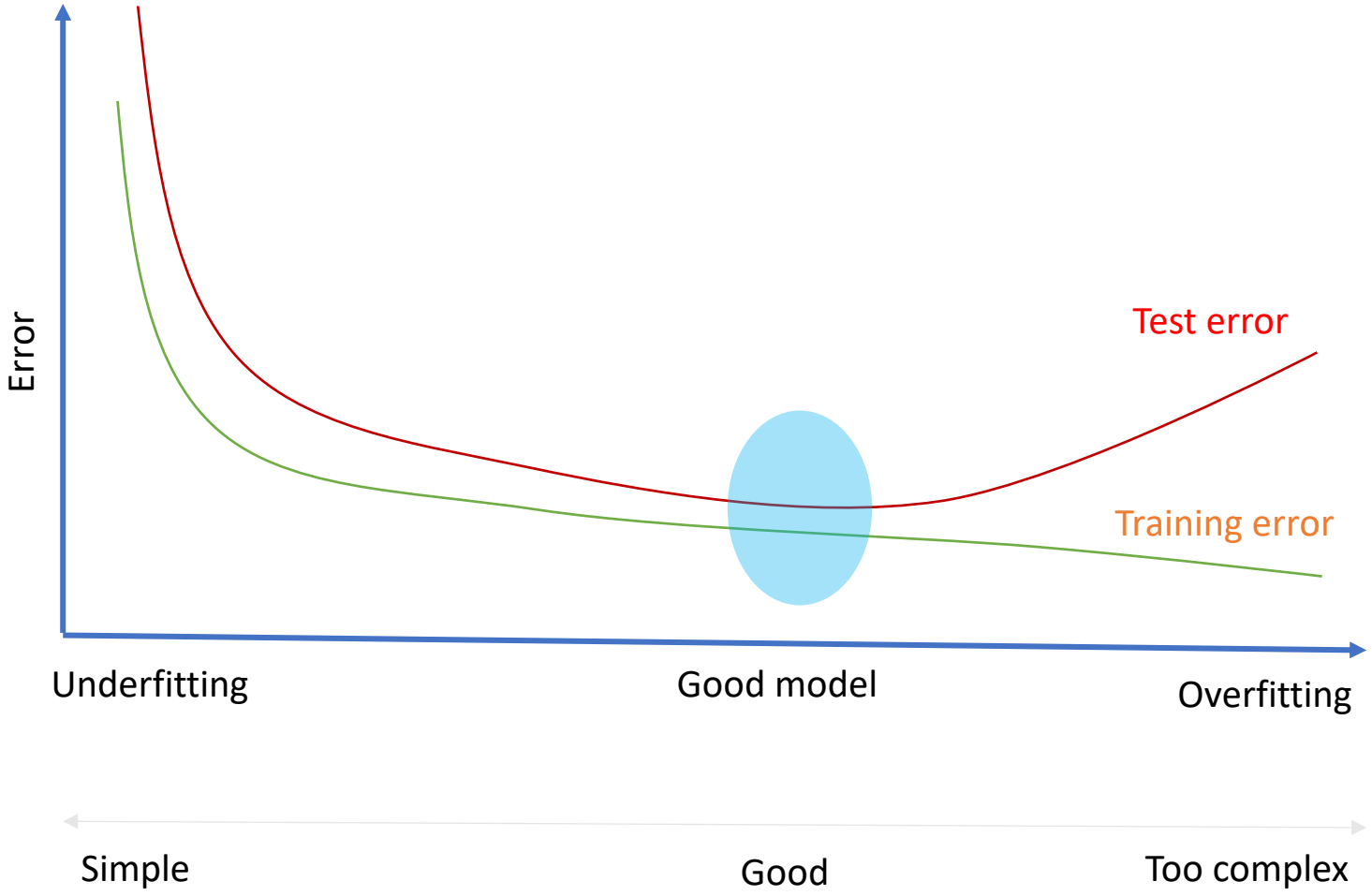
ML: some issues (3)

- **Learnability**:
  - The goodness/limit of the learning algorithm?
  - What is the generalization (tổng quát hoá) of the system?
    - Predict well new observations, not only the training data.
    - Avoid overfitting.

Overfitting (quá khớp, quá khít)

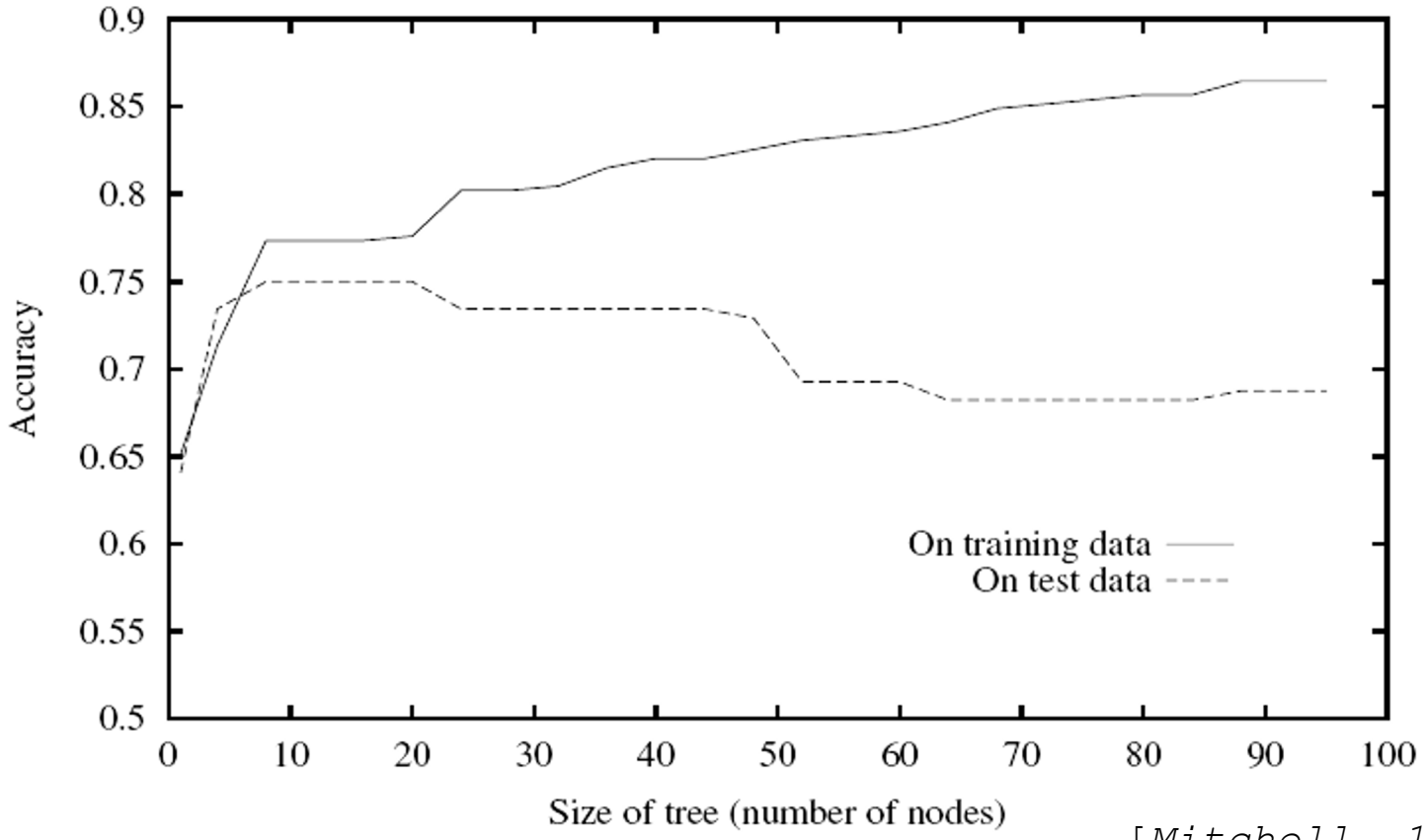
- Function h is called *overfitting* [Mitchell, 1997] if there exists another function g such that:
  - g might be worse than h for the training data, but
  - g is better than h for future data.
- A learning algorithm is said to overfit relative to another one if it is *more accurate in fitting* known data, but *less accurate in predicting* unseen data.
- Overfitting is caused by many factors:
  - The function/model is too complex or have too much parameters.
  - Noises or errors are present in the training data.
  - The training size is too small, not characterizing the whole space.

Overfitting



Overfitting: example

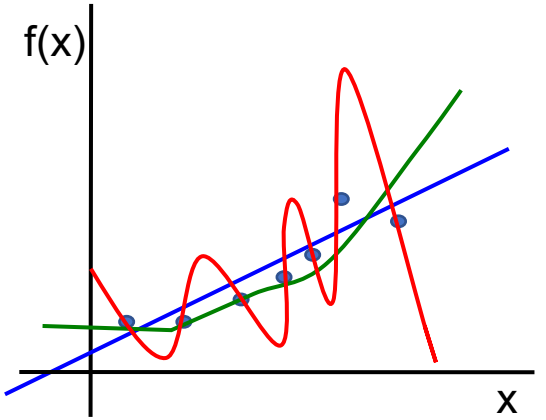
- Increasing the size of a decision tree can degrade prediction on unseen data, even though increasing the accuracy for the training data.

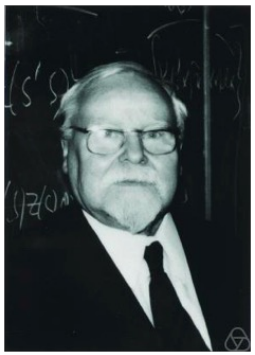


[Mitchell, 1997]

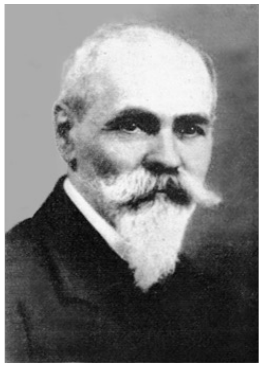
Overfitting: Regularization

- Among many functions, which one can generalize best from the given training data?
  - *Generalization is the main target of ML.*
  - Predict unseen data well.
- **Regularization:** a popular choice





Tikhonov, smoothing an ill-posed problem



Zaremba, model complexity minimization

Bayes: priors over parameters

Andrew Ng: need no maths, but it prevents overfitting!

(Picture from http://towardsdatascience.com/multitask-learning-teach-your-ai-more-to-make-it-better-dde116c2cd40)

References

- Alpaydin E. (2020). Introduction to Machine Learning. The MIT Press.
- Mitchell, T. M. (1997). Machine learning. *McGraw Hill*.
- Mitchell, T. M. (2006). *The discipline of machine learning*. Carnegie Mellon University, School of Computer Science, Machine Learning Department.
- Simon H.A. (1983). Why Should Machines Learn? In R. S. Michalski, J. Carbonell, and T. M. Mitchell (Eds.): Machine learning: An artificial intelligence approach, chapter 2, pp. 25-38. Morgan Kaufmann.
- Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*(11), 1134-1142.
- Wolpert, D.H., Macready, W.G. (1997), "No Free Lunch Theorems for Optimization", *IEEE Transactions on Evolutionary Computation* **1**, 67.

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

soict.hust.edu.vn/ fb.com/groups/soict