Posts and Telecommunication Institute of Technology
Faculty of Information Technology

Introduction to Artificial Intelligence

Introduction to Machine Learning

Dao Thi Thuy Quynh

Contents

References

- N. Nilsson. Introduction to machine learning http://ai.stanford.edu/people/nilsson/mlbook.html
- T. Mitchell. Machine learning. McGraw-Hill, 1997.
- E. Alpaydin. Introduction to machine learning. MIT Press, 2004.
- M. Mohri, A. Rostamizadeh, A. Talwalkar. Foundations of Machine Learning. MIT Press, 2012.

Tools and data

- Weka Toolkit
  - http://www.cs.waikato.ac.nz/~ml/weka
- UC Irvine dataset
  - http://www.ics.uci.edu/~mlearn/ML/Repository.html

Some applications of machine learning (1/3)

- Applications that are difficult to develop in the usual way because they do not exist or are difficult to explain human experiences and skills
  - Handwriting, sound, image recognition
  - Self-driving car, Mars exploration

- Computer program are adaptable: solutions change over time or according to specific situations
  - Personal help program
  - Network routing

Some applications of machine learning (2/3)

- Mining (analyzing) data
  - Medical records ➔ medical knowledge
  - Sales data ➔ business rules

Some applications of machine learning (3/3)

- Most of today's artificial intelligence applications use machine learning

...	...
Web search	Product recommendation
Speech recognition	Advertisement placement
Handwriting recognition	Smart-grid energy optimization
Machine translation	Household robotics
Information extraction	Robotic surgery
Document summarization	Robot exploration
Question answering	Spam filtering
Spelling correction	Fraud detection
Image recognition	Fault diagnostics
3D scene reconstruction	AI for video games
Human activity recognition	Character animation
Autonomous driving	Financial trading
Music information retrieval	Protein folding
Automatic composition	Medical diagnosis
Social network analysis	Medical imaging
...	...

What is Machine Learning?

- Learning:
  - …acquire knowledge or skills…
  - "*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.*" Tom Mitchell (1997)

- Machine learning:
  - Solve problem from experience
  - …is carried out by computer program that has abilities:
    - Do work (công việc) T better
    - Follow by criteria (tiêu chí) P
    - Using sample data or experience E

Example

- Learning to play chess
  - T: play chess
  - P: number of games won
  - E: self-play experience
- Learning to recognize letters
  - T: recognize letters from pictures
  - P: percentage of correct recognition
  - E: digital image of letters and corresponding label
- Machine translation
  - T: translate a English sentence to Vietnamese
  - P: level of translation (for example: number of correct sentence, number of correct clause,…)
  - E: pairs of English sentence and corresponding Vietnamese sentence

Problems with concern(1/2)

- What is the specific experience?
  - Direct and indirect experience
    - Direct: specific status + corresponding correct move
    - Indirect: entire game and result
  - Supervised and Unsupervised
    - Supervised
    - Unsupervised
    - Semi-supervised
- What needs to be learned? How to demonstrate knowledge?
  - Knowledge to be learned is represented as a target function, a specific target function needs to be selected.
  - Example of playing chess:
    - Select move: status ($trạng\ thái$) → move ($nước\ đi$)
    - Point: status ($trạng\ thái$) → point ($điểm\ số$)

Problems with concern (2/2)

- Which algorithm is used to learn?
  - Using function
  - Example: $\text{point} = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + \cdots$
  - Using laws
  - Using neural network
  - Using decision trees
  - Using probability models …

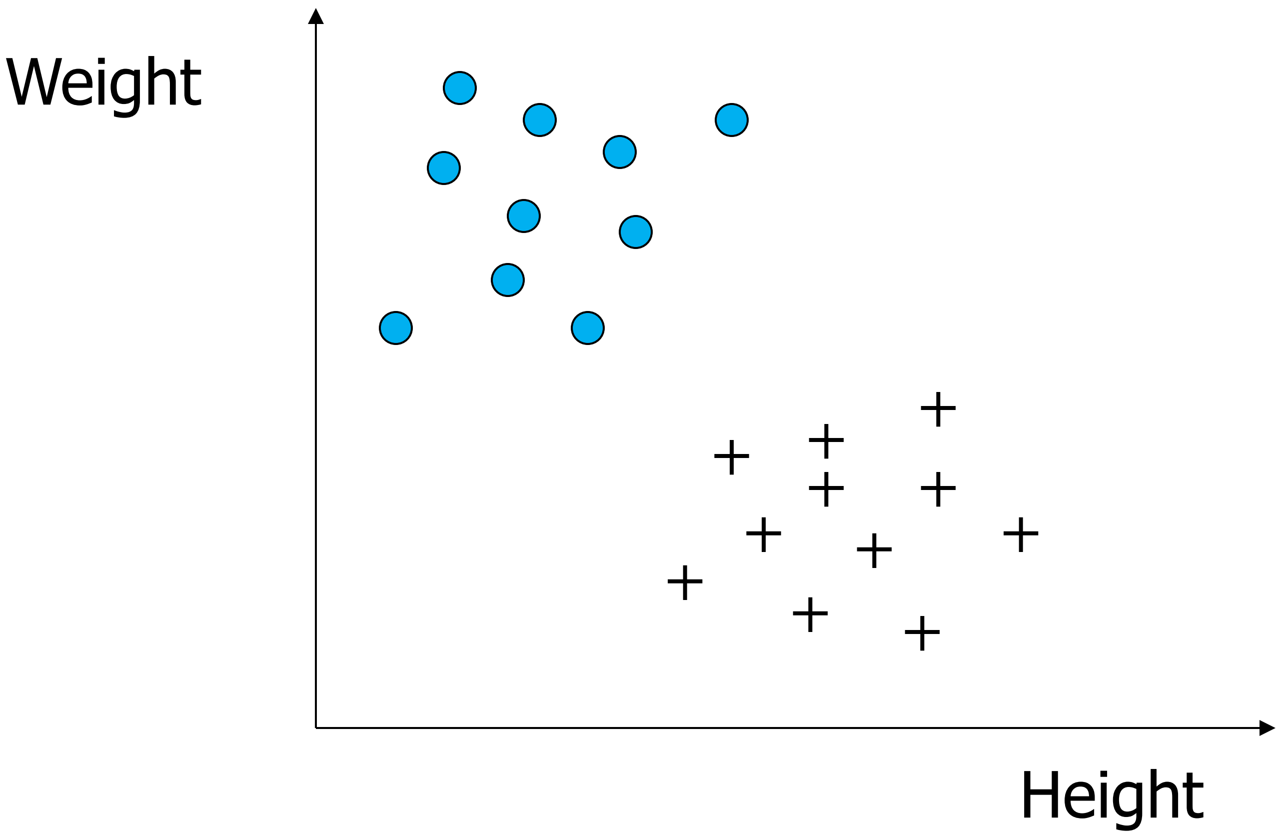
Some definitions

- Samples: is the object to be processed (example: classifier)
  - For example: when filtering spam email, each email is a sample
- Sample is usually described by a set of features:
  - For example: in disease diagnose, the features are symptoms of patients and other attribute like height, weight,…
- Label: describe the type of object we need to predict
  - For example: classification label of an email can be "Spam" or "Normal".

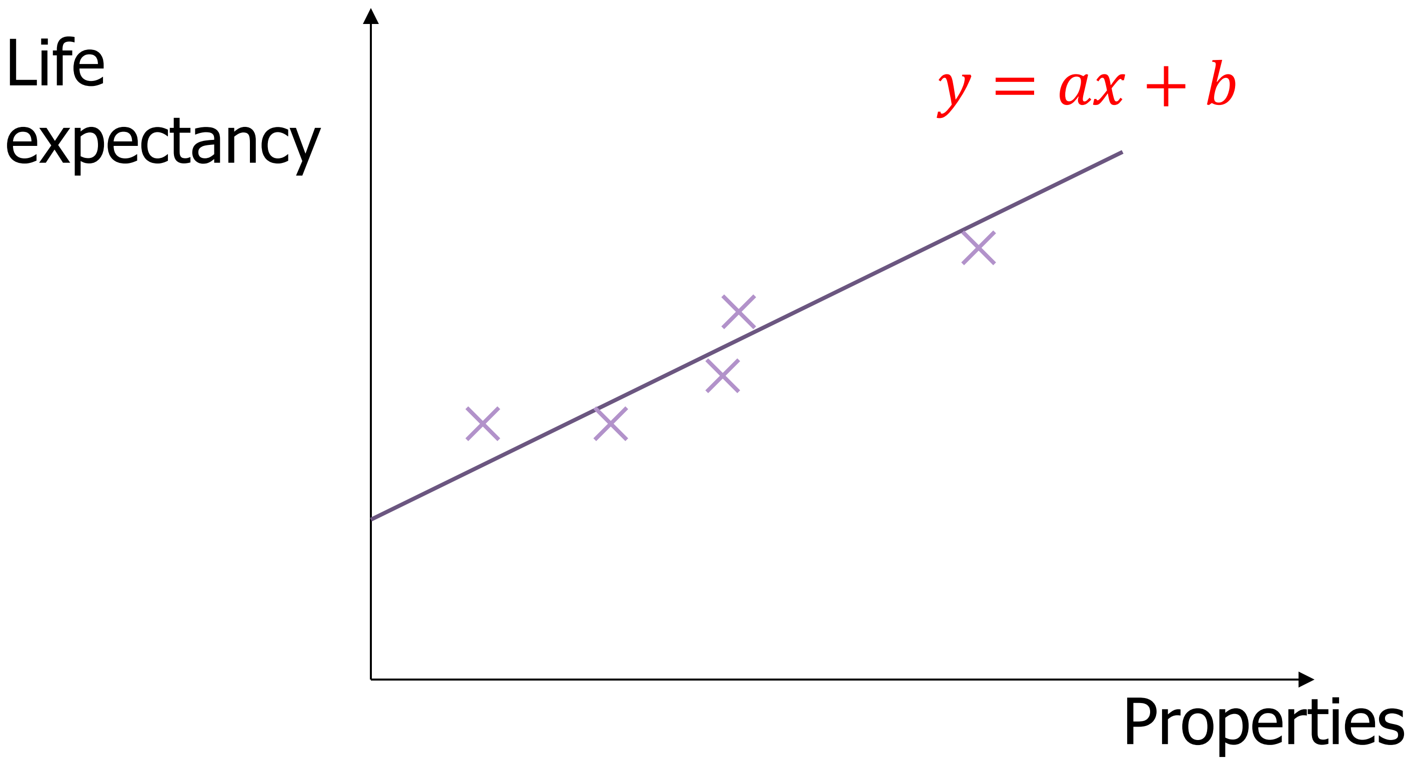
Some popular forms of machine learning

- Supervised learning
  - Classification
  - Regression
- Unsupervised learning
  - Association
  - Clustering
- Semi-supervised learning
- Reinforcement learning

Classification



Regression



Applications: predict market price, …

Association

- Example
  - Transaction analysis, sale (invoice)
- $P(Y|X)$
  - Probability a person who bought X also buy Y.
- Example of association
  - People who buy bread often but milk
  - People who buy beer often buy peanuts

Clustering

- Group similar samples
- No output value
- Applications
  - Customer clustering, student clustering
  - Image segmentation
  - Design microchips

Reinforcement learning

- Experience is not given directly as input/ output
- System receives a reward as a result of a certain sequence of actions
- Algorithms need to learn how to act to maximize reward value
- Example: learning to play chess
  - System do not know directly which move is suitable for each situation
  - Just know the result of game after a series of moves

Contents

- Introduction
- Decision tree
- Naïve Bayes classification
- Instance-based learning

Training data

Ngày	Trời	Nhiệt độ	Độ ẩm	Gió	Chơi tennis
D1	nắng	nóng	cao	yếu	không
D2	nắng	nóng	cao	mạnh	không
D3	u ám	nóng	cao	yếu	có
D4	mưa	trung bình	cao	yếu	có
D5	mưa	lạnh	bình thường	yếu	có
D6	mưa	lạnh	bình thường	mạnh	không
D7	u ám	lạnh	bình thường	mạnh	có
D8	nắng	trung bình	cao	yếu	không
D9	nắng	lạnh	bình thường	yếu	có
D10	mưa	trung bình	bình thường	yếu	có
D11	nắng	trung bình	bình thường	mạnh	có
D12	u ám	trung bình	cao	mạnh	có
D13	u ám	nóng	bình thường	yếu	có
D14	mưa	trung bình	cao	mạnh	không

features

label

sample

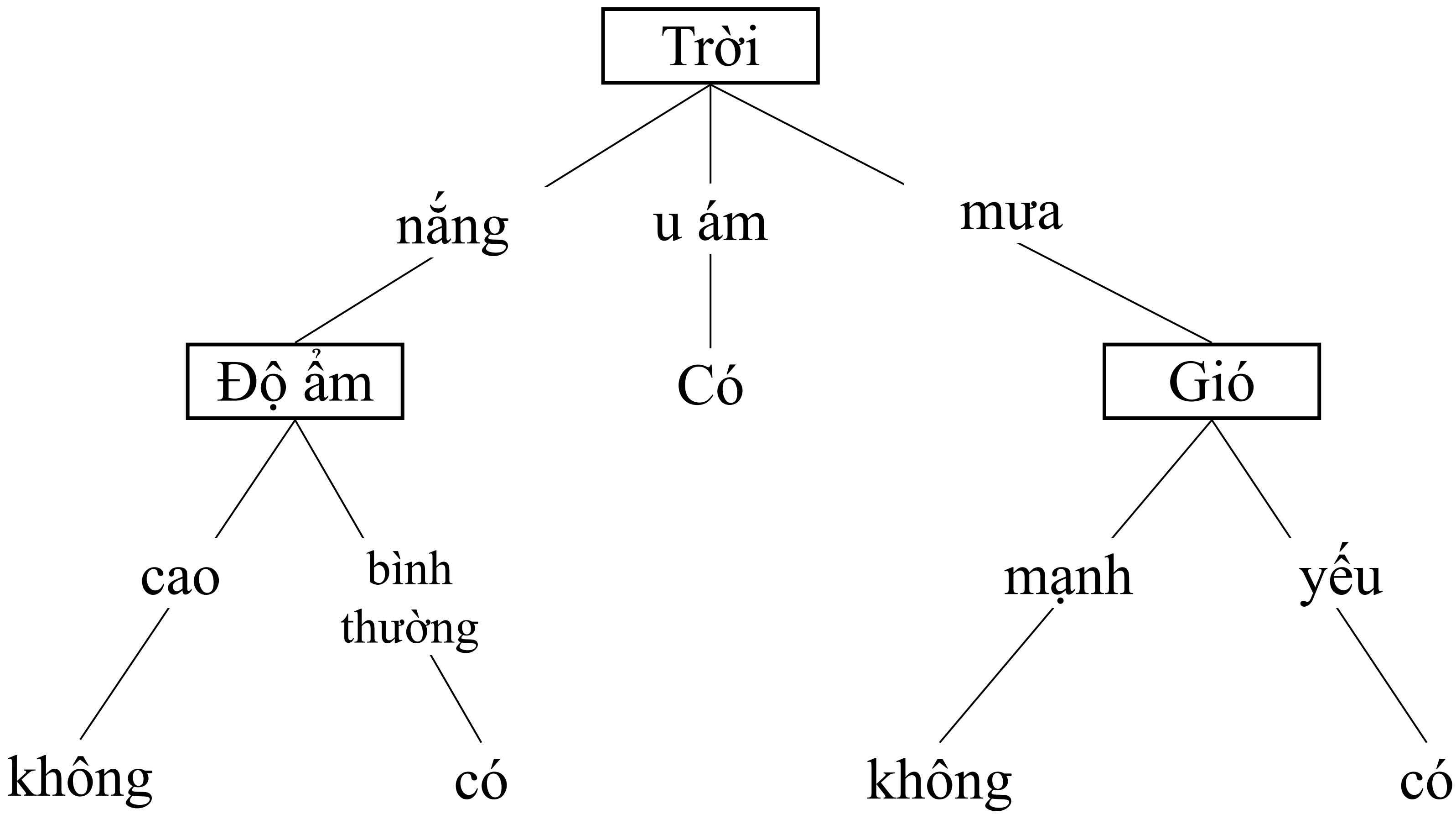
Ngày	Trời	Nhiệt độ	Độ ẩm	Gió	Chơi tennis
D1	nắng	nóng	cao	yếu	không
D2	nắng	nóng	cao	mạnh	không
D3	u ám	nóng	cao	yếu	có
D4	mưa	trung bình	cao	yếu	có
D5	mưa	lạnh	bình thường	yếu	có
D6	mưa	lạnh	bình thường	mạnh	không
D7	u ám	lạnh	bình thường	mạnh	có
D8	nắng	trung bình	cao	yếu	không
D9	nắng	lạnh	bình thường	yếu	có
D10	mưa	trung bình	bình thường	yếu	có
D11	nắng	trung bình	bình thường	mạnh	có
D12	u ám	trung bình	cao	mạnh	có
D13	u ám	nóng	bình thường	yếu	có
D14	mưa	trung bình	cao	mạnh	không

Data

- n training sample, each sample is a pair $< \boldsymbol{x}, y >$
  - $\boldsymbol{x}$ is a features vector
  - y is label, $y \in C$ (a set of labels)

- Example for sample D4
  - $\boldsymbol{x} = (mưa,\ trung\ bình, cao, yếu)$
  - $y = có$

Example of Decision tree



What is Decision Tree?

- Is a tree-like classification model
  - Each mid-node (not a leaf node) corresponds to an features testing, each branch of node corresponding to a feature value at that node
  - Each leaf node corresponds to a label

- Classification process:
  - Sample goes from the root down to the bottom.
  - At each mid-node, features vector of that node is tested. Depending on feature value, sample moves down the corresponding branch
  - When reaching the leaf nod, sample is given a classification label

Representation as a principle

- Decision tree can be represented in terms of logical principle
- Each tree is a disjunction of principles, each principle includes conjunctions.
- Example:

(Trời = nắng ∧ Độ ẩm = bình_thường)

∨ (Trời = u_ám)

∨ (Trời = mưa ∧ Gió = yếu)

Decision tree

- Decision tree is learned (built) from training data
- For each data set, can we build many decision trees?
  - How to choose tree? Which tree?

- The learning process is the process of finding a decision tree that is suitable for the training data
  - Allows to exactly classify training data

ID3 Algorithm

- Build tree's nodes from the root

- Algorithm
  - **Init**: the current node is the root-node containing all of training data set
  - At the current node n, select features:
    - Unused at ancestor-node (previous node)
    - Allows to divide training data set into subsets **in the best way**
    - For each feature value selected, add a child-node below
    - Divide samples of current node into child-node by selected feature value
  - **Repeat** (recursively) until:
    - All features were used at above nodes, or
    - All samples of current node have the same label
    - Label of node is takend by the majority of labels of samples of current node

How to choose feature at each node?

Criteria for feature selection of ID3

- At each node n
  - The set (subset) of data corresponding to that node
  - Need to select the feature that allows the best splitting of the data set
- Criteria:
  - Data after being divided is as large as possible
  - Measure Information Gain - IG
  - **Select the feature with the largest IG**
  - IG is calculated based on entropy of set (subset) of data

Entropy

- The case that data set has 2 types of labels: true (+) or false (-)

$$\boldsymbol{Entropy(S) = -p_+ log_2 p_+ - p_- log_2 p_-}$$

p_+: % number of true samples, p_-: % number of false sample

- General case: has C types of label

$$\boldsymbol{Entropy(S) = \sum_{i=1}^{C} -p_i \, log_2 \, p_i}$$

p_i : % sample of S belongs to type i

- Example

$$\boldsymbol{Entropy([9^+, 5^-]) = -(9/14)log_2(9/14) - (5/14)log_2(5/14) = 0.94}$$

Information Gain

With set (subset) of samples S and features A

$$IG(S, A) = Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Where:

$values(A)$: set of values of feature A

S_v is the subset of S including samples having values of A is v

$|S|$ number of elements of S

Example for calculating IG

- Calculate $IG(S, Gió)$

$$values(Gió) = \{yếu, mạnh\}$$

$$S = [9+, 5-], H(S) = -\frac{9}{14}log_2\frac{9}{14} - \frac{5}{14}log_2\frac{5}{14} = 0.94$$

$$S_{yếu} = [6+, 2-], \mathrm{H}(S_{yếu}) = -\frac{6}{8}log_2\frac{6}{8} - \frac{2}{8}log_2\frac{2}{8} = 0.811$$

$$S_{mạnh} = [3+, 3-], \mathrm{H}(S_{mạnh}) = -\frac{3}{6}log_2\frac{3}{6} - \frac{3}{6}log_2\frac{3}{6} = 1$$

$$\begin{aligned} IG(S, Gió) &= H(S) - \frac{8}{14}\mathrm{H}(S_{yếu}) - \frac{6}{14}\mathrm{H}(S_{mạnh}) \\ &= 0.94 - \tfrac{8}{14}0.811 - \tfrac{6}{14}1 \\ &= 0.048 \end{aligned}$$

Attributes of ID3

- ID3 is an algorithm finding a decision tree that is suitable for training data
- Searching in the greedy way, starting from an empty tree
- Evaluation function is Information Gain
- ID3 has bias to select simple tree
  - Number of nodes is small
  - Features with large IG are located nearby

Training error and Test error (1/2)

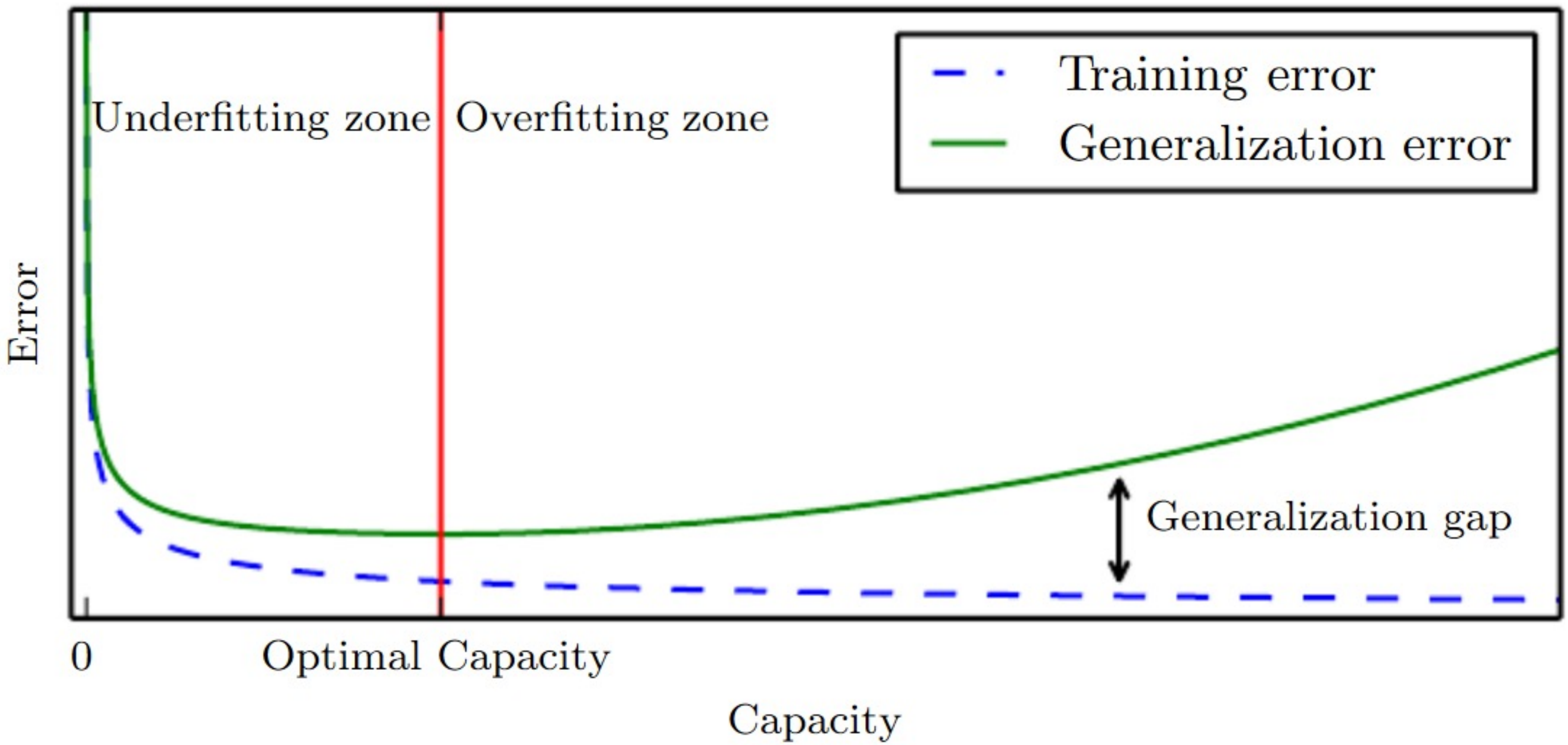
- Training error
  - Is error measured on training data
  - Be usually measured by the difference between predicted value of model and truth value of training data
  - In learning process, we try to minimize training error
- Test error
  - Is error measured on testing data
  - This is what we really care about!

How can we affect the performance of the model on test dataset when we only observe the training dataset?

Training error and Test error (2/2)

- i.i.d assumptions (independent, identically distributed)
  - Assume that samples (on both training and testing data) are independent, and training data and testing data have the same distribution.
  - If we fix parameters of model, then training error and test error will be equal
    - In training, parameters will be optimized by training error, so test error is usually bigger than training error.
- 2 factors to evaluate the performance of a machine learning model:
  - Ability to reduce training error
  - Ability to reduce the gap between training error and test error

Underfitting and Overfitting



Underfitting; Overfitting
Generalization error = test error
Capacity

Address overfitting by pruning tree

- Split data into 2 parts
  - Training
  - Testing
- Create enough big tree on training data
- Calculate accuracy of tree on testing data
- Remove subtree so that result on testing data is improved
- Repeat until there is no improvements for result

Address overfitting by pruning laws (C4.5)

- Convert tree into laws
- Pruning each law independent of the others
  - Remove some part in the left side part of law
- Arrange laws after pruning by accuracy of laws

Use feature having constant value

- Create new discrete features
- For example, with the continuous feature A, create discrete feature Ac as follows:
  - $Ac = true$ if $\mathrm{A} > c$
  - $Ac = false$ if $\mathrm{A} \leq c$
- How to determine threshold c?
  - Usually choose so that Ac gives the greatest information gain
- Can be divided into ranges with multiple thresholds.

Other measuring methods

- Information Gain (IG) prioritize features that have multiple values, for example, the date feature will have the highest IG.

- Split information:

$$SplitInformation(S, A) = -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

- Gain ratio:

$$GainRatio = \frac{InformationGain(S, A)}{SplitInformation(S, A)}$$

Contents

Naïve Bayes classification (1/2)

- In training phase, we a set of samples, each sample is a pair $<\boldsymbol{x}_i, y_i>$, where
  - $\boldsymbol{x}_i$ features vector
  - y_i is label, $y_i \in C$ (C is set of labels)
- After training, classifier need predict label y for new sample $\boldsymbol{x} =< x_1, x_2, \dots, x_n >$

$$y = argmax_{c_j \in C} P(c_j | x_1, x_2, \dots, x_n)$$

- Using Bayes principle:

$$y = argmax_{c_j \in C} \frac{P(x_1, x_2, \dots, x_n | c_j) P(c_j)}{P(x_1, x_2, \dots, x_n)}$$

$$= argmax_{c_j \in C} P(x_1, x_2, \dots, x_n | c_j) P(c_j)$$

Naïve Bayes classification (2/2)

Frequency of observing the label c_j on dataset D: $\frac{count(c_j)}{|D|}$

$$y = argmax_{c_j \in C} P(x_1, x_2, \ldots, x_n | c_j) P(c_j)$$

Using theory about independence probability (**Đơn giản!!!**)

$$P(x_1, x_2, \ldots, x_n | c_j) = P(x_1 | c_j) P(x_2 | c_j) \ldots P(x_n | c_j)$$

Number of occurrence x_i with c_j divided by number of occurence c_j: $\frac{count(x_i, c_j)}{count(c_j)}$

Example

- Decide classification label for following sample:

$$< Trời = nắng, Nhiệt\ độ = trung\ bình, Độ\ ẩm = cao, Gió = mạnh >$$

$$y = argmax_{c \in \{có, không\}} P(Trời = nắng|c)P(Nhiệt\ độ = trung\ bình|c)$$
$$P(Độ\ ẩm = cao|c)P(Gió = mạnh|c)P(c)$$

Contents

- Introduction
- Decision tree
- Naïve Bayes classification
- Instance-based learning

General principles

- No building model
- Only save training samples
- Define a label for a new sample based on samples in data set that are similar to the new sample.
- Called as "lazy learning"

K-nearest neighbors algorithm (KNN)

- k-Select k samples that are most similar to new sample., called as k neighbors
- Labeling for sample by using only information of k this neighbors
  - Label is decided based on the majority of k neighbors
- How to choose neighbors?

Distance measuring

- Assume that sample x has feature values
  $< a_1(x), a_2(x), \ldots, an(x) >$, where $a_i(x)$ is real number.
- Distance between 2 samples x_i and x_j is the Euclidean distance:

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{n} (a_l(x_i) - a_l(x_j))^2}$$

k-NN algorithm

Learning phase (training)

Save training samples of the form <x, f(x)> into the database

Classification phase

Input: parameter k

For sample x to be classified:

1. Calculate the distance d(x, x_i) from x to all samples xi in the database
2. Find k samples with the smallest d(x, x_i), assuming those k samples are x_1, x_2, ..., x_k.
3. Determine the classification label f'(x) is the label that occupies the majority in the set $\{x_1, x_2, \ldots, x_k\}$