

Khoa Toán, Đại học Khoa học Huế

HỌC MÁY

Machine Learning

TS. Nguyễn Hoàng Sơn

Nội dung môn học

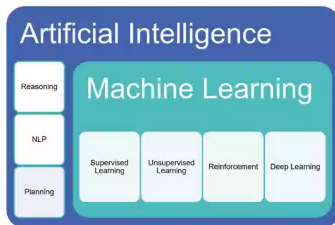
- ❶ Tổng quan về học máy
- ❷ Toán, Python và Weka
- ❸ Học có giám sát
 - Hàm mất mát
 - k -láng giềng gần nhất
 - Hồi quy tuyến tính
 - Cây quyết định
 - Bộ phân loại Naïve Bayes
 - Máy véc-tơ hỗ trợ
- ❹ Học không giám sát
 - Phân cụm k -mean
 - Giảm chiều bằng tập thô
- ❺ Học kết hợp
- ❻ Tài liệu tham khảo

Nội dung môn học

- 1 Tổng quan về học máy
- 2 Toán, Python và Weka
- 3 Học có giám sát
 - Hàm mất mát
 - k -láng giềng gần nhất
 - Hồi quy tuyến tính
 - Cây quyết định
 - Bộ phân loại Naïve Bayes
 - Máy véc-tơ hỗ trợ
- 4 Học không giám sát
 - Phân cụm k -mean
 - Giảm chiều bằng tập thô
- 5 Học kết hợp
- 6 Tài liệu tham khảo

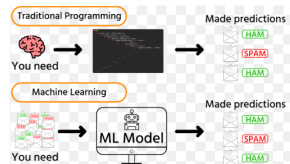
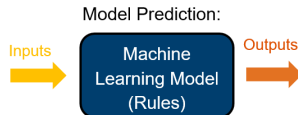
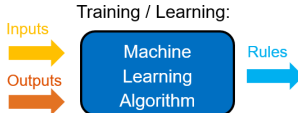
Chương 1. Tổng quan về học máy

- 1 Khái niệm học máy
- 2 Kiến trúc học máy
- 3 Dữ liệu trong học máy
- 4 Phân loại các mô hình học máy
- 5 Vấn đề quá khớp, chưa khớp
- 6 Ứng dụng của học máy



Khái niệm học máy: Định nghĩa

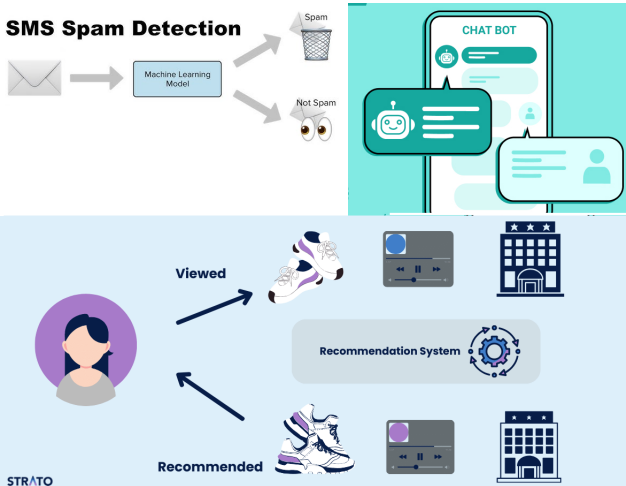
- **Học máy** (Machine Learning): 1 nhánh của TTNT (AI) giúp MT học tự động từ DL mà không cần lập trình một cách tường minh.



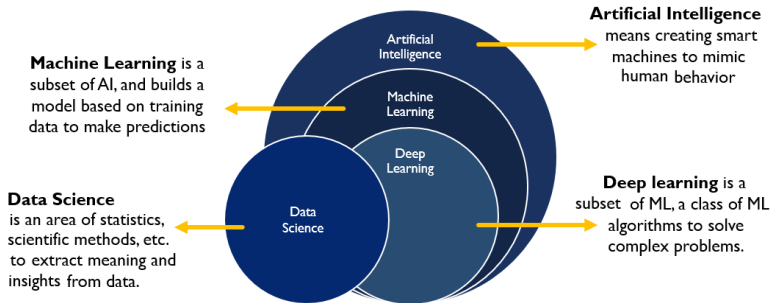
Khái niệm học máy: Mục tiêu

- Tạo ra các MH, TT có khả năng
 - ① **Học từ DL (DLHL):**
 - Tìm ra (ẩn) các *mẫu*, *xu hướng*, *quy luật* (MHHM).
 - ② **Dự đoán, phân loại DL (mới):**
 - MHHM $\Rightarrow$ *DD* hoặc *PL*.
 - ③ **Tinh chỉnh, cải thiện hiệu suất:**
 - Điều chỉnh các tham số của MHHM để cải thiện hiệu suất.

Khái niệm học máy: Ví dụ: một số ứng dụng HM

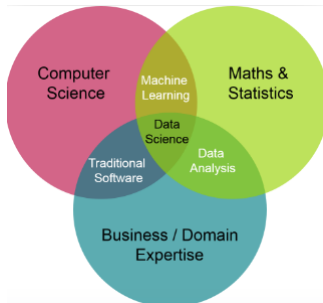


Khái niệm học máy: Mối quan hệ - (1)



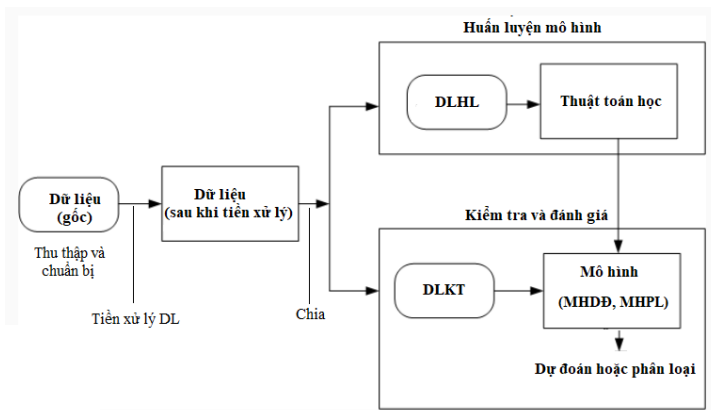
HM với TTNT, KHDL

Khái niệm học máy: Mối quan hệ - (2)



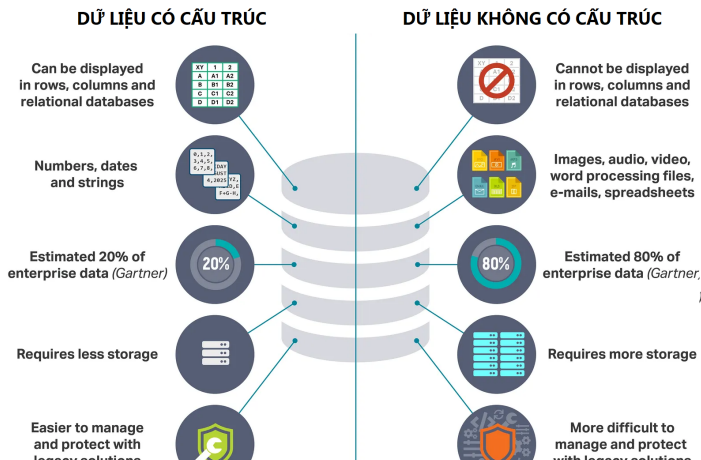
HM với Toán học, KHMT

Kiến trúc học máy



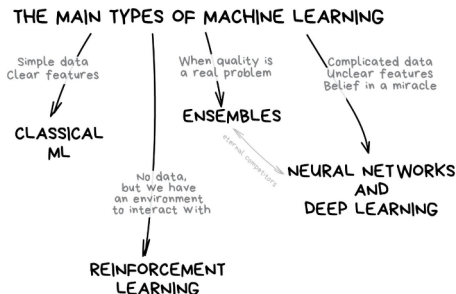
ML Architecture

Dữ liệu trong học máy



Phân loại các MHHM

- HM thường được chia thành các nhánh:
 - ① **HM cổ điển**
 - ① Học có giám sát (Supervised learning)
 - ② Học không giám sát (Unsupervised learning)
 - ② Học tăng cường (Reinforcement learning)
 - ③ Học kết hợp (Ensemble learning)
 - ④ Mạng Neural và học sâu (Deep learning)



Phân loại các MHHM: HCGS-Định nghĩa

- DLHL có nhãn, $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, MH học (tìm) từ D ra hàm

$$f: X \longrightarrow Y$$

- sao cho $f(x_i) \simeq y_i$ (nhất có thể) $\Rightarrow$ nó có thể DD chính xác nhãn cho những điểm DL mới chưa từng nhìn thấy.

$\leadsto$

- $X = \{x_1, \dots, x_n\}$: *điểm DL đầu vào*, với $x_i \in \mathbb{R}^d$ - véc-tơ đặc trưng
- $Y = \{y_1, \dots, y_n\}$: *nhãn (đầu ra) tương ứng*

Phân loại các MHHM: HCGS-Các bài toán

① Bài toán phân loại (Classification):

$$y_i \in \{C_1, \dots, C_k\}$$

$\Leftrightarrow Y$ là tập hữu hạn các lớp (rời rạc).

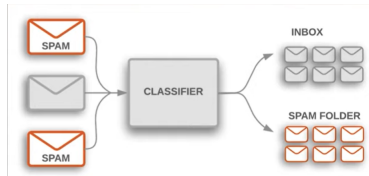
② Bài toán hồi quy (Regression):

$$y_i \in \mathbb{R}$$

$\Leftrightarrow Y$ là tập các giá trị thực liên tục.

Phân loại các MHHM: HCGS-Ví dụ

1 Bài toán phân loại email:

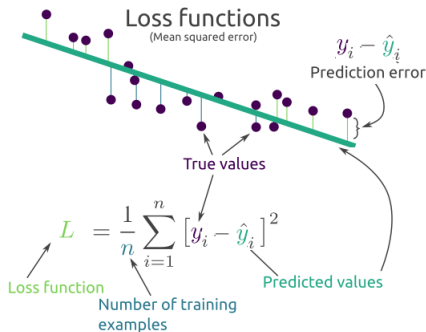


2 Bài toán DĐ giá nhà:

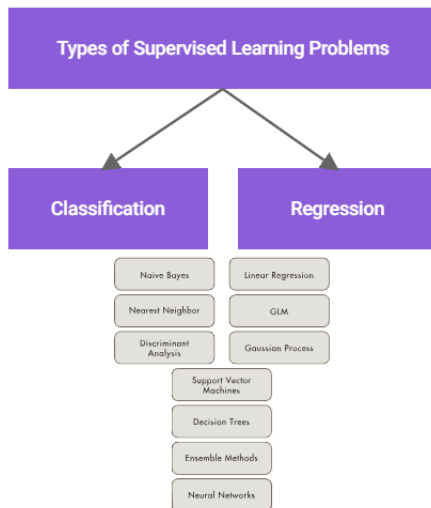


Phân loại các MHHM: HCGS-HMM và tối ưu hoá

- Quá trình HCGS thường liên quan đến việc tối ưu hóa 1 **hàm mất mát $\mathcal{L}$** (Loss function).
- $\mathcal{L}(f(x), y)$: đo sự sai khác giữa giá trị ĐĐ của MH $\hat{y} = f(x)$ và nhãn thực tế y .



Phân loại các MHHM: HCGS-Các TT



Phân loại các MHHM: HKGS-Định nghĩa

- DLHL không có nhãn, MH phải tự khám phá cấu trúc ẩn trong DL.
- $\leadsto$
 - DL $X = \{x_1, \dots, x_n\}, x_i \in \mathbb{R}^d$ - véc-tơ đặc trưng.
 - MH tìm 1 ánh xạ f hoặc cấu trúc C sao cho nó biểu diễn hoặc nhóm DL theo 1 tiêu chí nào đó (mà không cần nhãn y).

Phân loại các MHHM: HKGS-Bài toán (1)

- **Bài toán phân cụm (Clustering):**
 Tìm ánh xạ f từ DL vào các cụm:

$$f: X \longrightarrow C = \{C_1, \dots, C_k\}$$

sao cho

- ① Các điểm trong cùng 1 cụm có KC gần nhau:
 $\forall x_i, x_j \in C_k, d(x_i, x_j)$ nhỏ
- ② Các điểm thuộc cụm khác nhau có KC xa:
 $\forall x_i \in C_l, x_j \in C_t (t \neq l), d(x_i, x_j)$ lớn

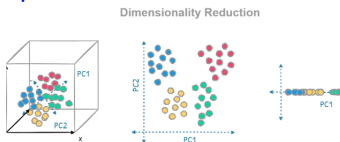
$\leadsto d$ là hàm KC.

Phân loại các MHHM: HKGS-Bài toán (2)

- **Bài toán giảm chiều DL** (Dimensionality Reduction): có 2 nhóm chính
 - ① **Trích xuất đặc trưng (Feature Extraction):** Biến đổi DL sang KG mới (như PCA).
 - Tìm ánh xạ

$$f: \mathbb{R}^d \longrightarrow \mathbb{R}^k$$

với $k < d$ sao cho thông tin quan trọng của DL vẫn được giữ lại.



- ② **Lựa chọn đặc trưng (Feature Selection):** Giữ nguyên các đặc trưng gốc của TDL nhưng loại bỏ những đặc trưng không quan trọng.
 - Sử dụng Tập thô (Rough Set) để tìm ra 1 *tập rút gọn* (Reduct) - là tập con tối thiểu các đặc trưng nhưng vẫn đảm bảo khả năng phân lớp chính xác như TDL ban đầu.

Phân loại các MHHM: HKGS-Bài toán (3)

- **Bài toán luật kết hợp** (Association rule): Tìm các LKH trên D có dạng:

$$X \Rightarrow Y$$

thỏa:

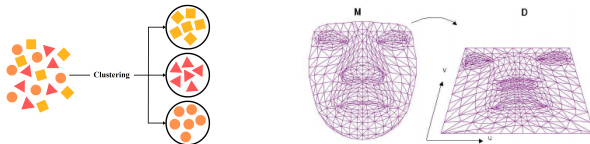
$$\textcircled{1} \text{ support}(X \Rightarrow Y) = \frac{|\{T \in D : X \cup Y \subseteq T\}|}{|D|} \geq \text{min_sup}$$

$$\textcircled{2} \text{ confidence}(X \Rightarrow Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X)} \geq \text{min_conf}$$

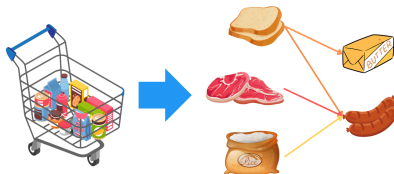
- $\leadsto$
 - $D = \{T_i : T_i \subseteq I, i = 1, 2, \dots, m\}$: CSDL giao dịch trên tập mục I .
 - $X, Y \subseteq I$ và $X \cap Y = \emptyset$.
 - $\text{min_sup}, \text{min_conf}$: các ngưỡng tối thiểu cho trước.

Phân loại các MHMM: HKGS-Ví dụ

① Bài toán phân loại hình, nhận dạng mặt người:

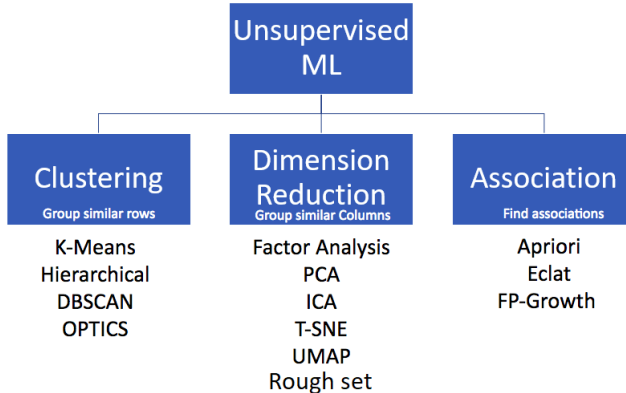


② Bài toán khai phá các mặt hàng:



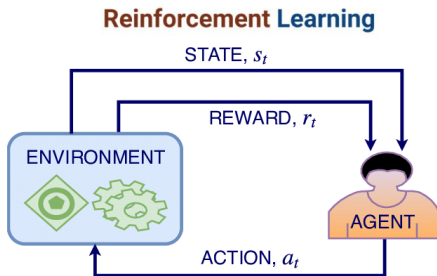
Purchase item A $\Rightarrow$ also purchase item B

Phân loại các MHHM: HKGS-Các TT



Phân loại các MHHM: HTC-Định nghĩa - (1)

- Một *tác nhân* (agent) học cách *hành động* (action) trong 1 *môi trường* (environment) để tối đa hóa *phần thưởng* (reward) theo thời gian.
- ~→
 - ① Hoạt động theo nguyên tắc thử và sai, sử dụng phản hồi từ môi trường để điều chỉnh chiến lược hành động.
 - ② Không yêu cầu DLHL, không có đánh giá các hành động là đúng hay sai.



Phân loại các MHHM: HTC-Định nghĩa - (2)

- Mô tả bằng toán học thông qua MH quá trình quyết định Markov (Markov Decision Process - MDP):
 - ① Ở thời điểm t , tác tử A có trạng thái $s_t \in S$, thực hiện hành động $a \in A$ (có XS $P_{sa}(s_{t+1})$) để nhận được 1 phần thưởng tức thời (cục bộ) $r_t \in R$.
 - ② $\Rightarrow$ Mục tiêu: tối đa phần thưởng chung R .
 - $\leadsto$
 - $P_{sa}(s_{t+1})$: 1 phân phối biểu diễn TTTT s_{t+1} của tác tử khi thực hiện hành động $a \in A$ tại trạng thái $s \in S$.

Phân loại các MHHM: HTC-Bài toán, TT

① Bài toán dự báo giá trị (Prediction):

- ① *Đệ quy Bellman*
- ② *Monte Carlo*

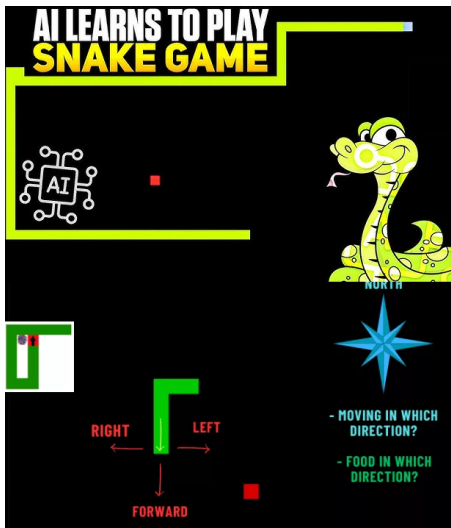
② Bài toán tối ưu hoá chính sách (Policy optimization):

- ① *Value Iteration, Policy Iteration*
- ② *Policy Gradient, PPO, TRPO*

③ Bài toán học trực tiếp (Control):

- ① *Q-learning*
- ② *Deep Q-Network (DQN)*

Phân loại các MHHM: HTC-Ví dụ



Phân loại các MHHM: HKH-Định nghĩa

- Kết hợp nhiều MHHM để cải thiện độ chính xác, tính ổn định của DD hơn (so với chỉ sử dụng 1 MH đơn lẻ).
- **Có các PP HMKH:**
 - ① *Phương pháp Bagging*
 - ② *Phương pháp Boosting*
 - ③ *Phương pháp Voting*
 - ④ *Phương pháp MultiClassClassifier*

Vấn đề quá khớp, chưa khớp: Giới thiệu

- 2 vấn đề thường gặp khi xây dựng MHHM:
 - ① **Quá khớp** (Overfitting)
 - ② **Chưa khớp** (Underfitting)

⇒ MHHM có ĐCX thấp.
- **Mục tiêu HM: vừa khớp** (Good Fitting $\in$ [Overfitting, Underfitting]).

Vấn đề quá khớp: Khái niệm, nguyên nhân

- MH học quá chi tiết từ DLHL (ghi nhớ DL nhiều, thay vì học quy luật chung) $\Rightarrow$
 - THL: (MH) độ chính xác cao
 - TKT: độ chính xác thấp
- **Nguyên nhân:**
 - ① DLHL chứa nhiều nhiễu.
 - ② DLHL và DLDB có “phân phối khác nhau nhiều” $\Rightarrow$ các qui luật học được ở DLHL không còn đúng trên DLDB.
 - ③ MH quá phức tạp so với DL (mới).
 - ④ MH quá nhiều tham số $\Rightarrow$ khả năng biểu diễn DL của nó không mang tính đại diện.

Vấn đề quá khớp: Cách khắc phục

- ① *Cross validation (k-fold): Kỹ thuật lấy lại mẫu* (Resampling), chia TDL thành k TDL con, cho phép HL trên các TDL khác nhau k lần.
- ② *Thêm DLHL: $\Rightarrow$ MH học tổng quát hơn.*
- ③ *Giảm ĐPT của MH (giảm số tầng mạng, số cây trong Random Forest ...).*
- ④ *Chính quy hoá (Regularization).*

Vấn đề chưa khớp: Khái niệm, nguyên nhân

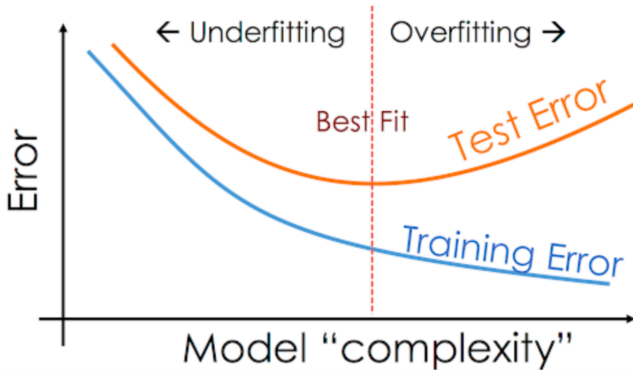
- MH không học được quy luật quan trọng từ DLHL (MH quá đơn giản so với bài toán) $\Rightarrow$
 - THL: DĐ kém
 - TKT: DĐ cũng kém.
 - $\leadsto$ thường ít xảy ra (so với quá khớp).
- Nguyên nhân:
 - ① MH quá đơn giản (ex. dùng MH HQT cho DLPT).
 - ② Không đủ DLHL, hay DL không chứa đủ thông tin.
 - ③ Sử dụng kỹ thuật regularization quá mạnh $\Rightarrow$ mất đi thông tin quan trọng.

Vấn đề chưa khớp: Cách khắc phục

- ❶ Kiểm tra loss trên THL và TKT.
- ❷ Thử các MH phức tạp hơn (ex. dùng CQĐ thay vì HQT).
- ❸ Thêm các đặc trưng phù hợp.
- ❹ Điều chỉnh regularization.

Vấn đề quá khớp, chưa khớp: Mối quan hệ

- Mối quan hệ giữa ĐPT MH và lỗi DD



Ứng dụng của học máy

- ① **KPDL:** ỨD kỹ thuật HM vào (*DL thương mại*)
 - ① CSDL
 - ② TDL lớn
- ② **NDM:** ỨD các kỹ thuật HM vào (*DL âm thanh, hình ảnh*)
 - Phát hiện các *mẫu có tính quy luật* trong DL

Nội dung môn học

- ① Tổng quan về học máy
- ② Toán, Python và Weka
- ③ Học có giám sát
 - Hàm mất mát
 - k -láng giềng gần nhất
 - Hồi quy tuyến tính
 - Cây quyết định
 - Bộ phân loại Naïve Bayes
 - Máy véc-tơ hỗ trợ
- ④ Học không giám sát
 - Phân cụm k -mean
 - Giảm chiều bằng tập thô
- ⑤ Học kết hợp
- ⑥ Tài liệu tham khảo

Chương 2. Toán, Python và Weka

- ① Kiến thức toán cơ bản dùng cho HM
- ② Các thư viện Python quan trọng cho HM
- ③ Phần mềm Weka



Kiến thức toán cho HM - (1)

(1) **XS và TK:** 2 lĩnh vực toán học cốt lõi cho MH

- XS: mô tả, đo lường sự không chắc chắn trong DL.
- TK: PP phân tích DL, làm rõ các mẫu và đặc trưng trong DL, đánh giá hiệu suất các MHHM.

(2) **Lý thuyết thông tin:** đo lường, phân tích thông tin trong DL để xác định các đặc trưng quan trọng và giảm thiểu sự lặp lại trong DL.

(3) **Đại số tuyến tính:** biểu diễn, XLDL

- Ma trận, véc-tơ: biểu diễn các đặc trưng, tham số trong MHHM.
- Các phép toán ĐSTT: xử lý các đặc trưng, các phép biến đổi DL phức tạp, giải các hệ PT tối ưu ...

Kiến thức toán cho HM - (2)

- (4) **Giải tích, tối ưu:** tìm kiếm GTTU cho các HMM, HMT. Các TTTU thường được dùng điều chỉnh tham số của MHHM:
- Tích phân, đạo hàm: xác định đạo hàm của các hàm mục tiêu.
 - TT hạ Gradient, các biến thể của TT hạ Gradient.
- (5) **Toán rời rạc:** phân tích TT, các MH liên quan đến xử lý DL không liên tục
- Đồ thị
 - Tập hợp ...

Các thư viện Python quan trọng - (1)

- **XLDL:**
 - ① *Pandas*: DL dạng bảng (*DataFrame*)
 - ② *Numpy*: mảng số học hiệu suất cao
 - ③ *Scipy*: khoa học, toán học, TK
- **Tiền XLDL, trực quan hóa DL:**
 - ① *Sklearn*: chuẩn hóa, mã hóa DL
 - ② *Matplotlib*: vẽ biểu đồ cơ bản
 - ③ *Seaborn*: trực quan hóa DL nâng cao
- **HM truyền thống:**
 - *Sklearn*:
 - ① TTHM cơ bản: *kNN*, *HQTT*, *CQĐ*, *SVM* ...
 - ② Đánh giá MH: *Accuracy*, *Precision*, *Recall*, *MSE*...
 - ③ Chia TDL

Các thư viện Python quan trọng - (2)

- **Học sâu:**
 - ① *Tensorflow & keras*: xây dựng MH học sâu
 - ② *PyTorch*: phổ biến, linh hoạt hơn *tensorflow*
- **XLDL nâng cao:**
 - ① *OpenCV*: XL ảnh
 - ② *NLTK & SpaCy*: XL NNTN (NLP)
 - ③ *Librosa*: XL âm thanh
- **Triển khai MHHM**: đưa MH vào thực tế thông qua API hoặc Web App
 - ① *Flask & FastAPI*: tạo API cho MHHM
 - ② *Streamlit*: xây dựng giao diện web đơn giản cho AI
 - ③ *ONNX & Tensorflow lite*: tối ưu hóa MH cho thiết bị di động hoặc IoT
- ~> **Kiến thức toán, thư viện Python**: HV tự đọc.

Phần mềm Weka - (1)

1 Tải và cài đặt: *

(https://waikato.github.io/weka-wiki/downloading_weka/#windows_1)

2 Hướng dẫn sử dụng: 1 2



Phần mềm Weka - (2)

- **Tập có đuôi ARFF (Attribute-Relation File Format):**
 - ① Tập tin văn bản theo bảng mã ASCII.
 - ② Mô tả các DT (có chung tập thuộc tính) trong KG n chiều, được sử dụng làm định dạng chuẩn cho DL dùng trong các MH (HM, KPDL) của Weka.
 - ③ Cấu trúc:
 - ① **Header:** *@relation* <tên tập> *@attribute* <thuộc tính> <kiểu DL>
 - ② **Data:** *@data* <nội dung tập>
 - ④ Kiểu DL: *Numeric* (real, integer), *Nominal*, *String*, *Date*.

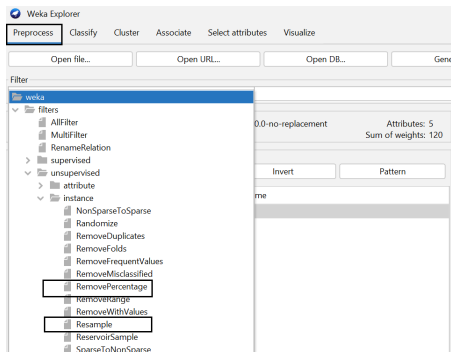
```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
```

Phần mềm Weka - (3)

- Chuẩn bị DL để HL và kiểm thử MH:
 - Tạo 2 tập train set và test set bằng 1 trong 2 bộ lọc:
 - ① *Resample*
 - ② *RemovePercentage*



Phần mềm Weka - (4)

- Chuẩn bị DL để HLMH:
 - Từ tập **train set**, chọn 1 trong 2 PP:
 - ① PP Holdout 1
 - ② PP Cross-Validation 2

Weka Explorer

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier

Choose **ZeroR**

Test options

☐ Use training set

☐ Supplied test set Set...

☐ Cross-validation Folds 10

☒ Percentage split % 70

More options...

Classifier output

Holdout

Cross-Validation : 5

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Complete Data	Test	Training	Training	Training	Training	Prediction Statistics
	Training	Test	Training	Training	Training	
	Training	Training	Test	Training	Training	
	Training	Training	Training	Test	Training	
	Training	Training	Training	Training	Test	
						Test
						Test
						Test
						Test
						Test

(Nom) class

Start Stop

Result list (right-click for options)

Nội dung môn học

- 1 Tổng quan về học máy
- 2 Toán, Python và Weka
- 3 Học có giám sát
 - Hàm mất mát
 - k -láng giềng gần nhất
 - Hồi quy tuyến tính
 - Cây quyết định
 - Bộ phân loại Naïve Bayes
 - Máy véc-tơ hỗ trợ
- 4 Học không giám sát
 - Phân cụm k -mean
 - Giảm chiều bằng tập thô
- 5 Học kết hợp
- 6 Tài liệu tham khảo

Chương 3. Học có giám sát

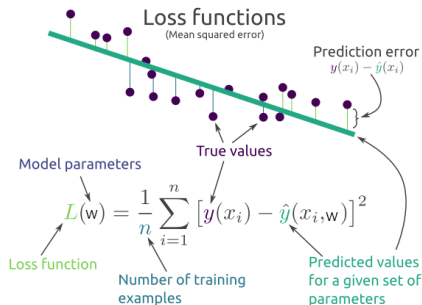
- 1 Hàm mất mát
- 2 k -láng giềng gần nhất
- 3 Hồi quy tuyến tính
- 4 Cây quyết định
- 5 Bộ phân loại Naive Bayes
- 6 Máy véc-tơ hỗ trợ



Hàm mất mát: Định nghĩa

- Hàm số đo độ sai lệch giữa GTĐĐ (predicted value) $\hat{y}$ của MHHM và GTTT (ground truth) y của DLHL:

$$\mathcal{L}(\hat{y}, y) = \hat{y} - y (= \mathcal{L}(w, b))$$



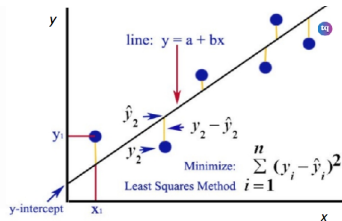
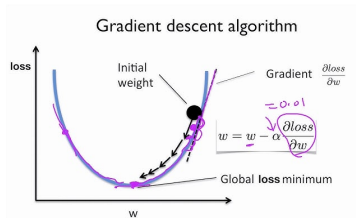
Hàm mất mát: Tối ưu HMM

- Mục tiêu HM:

$$\mathcal{L}(w, b) \rightarrow \min$$

⇔ điều chỉnh các tham số của MHHM (w, b) sao cho nó có khả năng ĐĐ chính xác hơn trên DL mới:

- 1 PP bình phương tối thiểu (OLS - Ordinary Least Squares)
- 2 PP hạ Gradient (Gradient Descent)
- 3 Các biến thể của PP hạ Gradient.



Hàm mất mát: Một số HMM - (1)

- $\leadsto (\hat{y}_i, y_i), i = 1, 2, \dots, m$
- **HQTT:**
 - ① *Mean Squared Error (MSE):* phổ biến

$$\mathcal{L}(w, b) = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$$

- ② *Mean Absolute Error (MAE):*

$$\mathcal{L}(w, b) = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|$$

- ③ *Huber:* kết hợp giữa MSE và MAE.

Hàm mất mát: Một số HMM - (2)

- **Phân loại:**

- ① *Cross-Entropy Loss (Binary):*

$$-\frac{1}{m} \sum_{i=1}^m [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

- ② *Cross Entropy Loss (Multiclass):*

$$-\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K y_{i,k} \log(\hat{y}_{i,k})$$

- ③ *Hinge Loss (SVM):*

$$\max(0, 1 - y_i \hat{y}_i)$$

- **Phân cụm:**

- *HMM k-mean:* Tổng bình phương KC từ các điểm DL đến trung tâm cụm của chúng.

Hàm mất mát: PPTU HMM-TT Gradient Descent-(1)

- Đầu vào: HMM $\mathcal{L}(w, b)$.
- Đầu ra: GTTU của tham số w, b .
- Phương pháp:
- Bước 1. (Khởi tạo w, b) Chọn NN hoặc theo giá trị ban đầu.
- Bước 2. (Lặp, điều chỉnh tham số w, b)
 - Tính Gradient:
 - ① $\frac{\partial \mathcal{L}(w, b)}{\partial w}$
 - ② $\frac{\partial \mathcal{L}(w, b)}{\partial b}$
 - Cập nhật tham số:
 - ① $w = w - \alpha \frac{\partial \mathcal{L}(w, b)}{\partial w}$
 - ② $b = b - \alpha \frac{\partial \mathcal{L}(w, b)}{\partial b}$
 - $\leadsto \alpha$ tốc độ học (constant > 0 , nhỏ)

Hàm mất mát: PPTU HMM-TT Gradient Descent-(2)

- **Bước 2.** (tiếp theo)
 - **Kiểm tra điều kiện dừng:** thoả 1 trong các điều kiện
 - ❶ Số lần lặp đã đủ (chọn).
 - ❷ Sự thay đổi của $\mathcal{L}(w)$ không đáng kể ($< \varepsilon$ -ngưỡng).
 - ❸ Gradient gần như bằng 0.
- **Bước 3.** (Kết luận) GTTU của w, b .

Hàm mất mát: PPTU HMM-TTGD-Ví dụ - (1)

- DLHL: $X = [1, 2, 3, 4]$, $\mathbf{y} = [2, 2.8, 3.6, 4.5]$
- Hàm DĐ (HQTТ): $\hat{\mathbf{y}} = \mathbf{w}X + b$
- $\leadsto$ HMM MSE: $\mathcal{L}(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$
- $\Rightarrow$ Gradient:
 - $\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = -\frac{2}{m} \sum_{i=1}^m x_i (y_i - \hat{y}_i)$
 - $\frac{\partial \mathcal{L}}{\partial b} = -\frac{2}{m} \sum_{i=1}^m (y_i - \hat{y}_i)$
- $\leadsto$
 - $\alpha := 0.01$, $epoch = 10$
 - $\mathbf{w} := 0.5$, $b := 0.5$

Hàm mất mát: PPTU HMM-TTGD-Ví dụ - (2)

X	y	w, b	$\hat{y} = wX + b$	$y - \hat{y}$	$\frac{\partial \mathcal{L}}{\partial w}$	$\frac{\partial \mathcal{L}}{\partial b}$
1	2	0.5	$0.5(1) + 0.5 = 1$	1	-8.2	-2.95
2	2.8		$0.5(2) + 0.5 = 1.5$	1.3		
3	3.6	0.5	$0.5(3) + 0.5 = 2$	1.6		
4	4.5		$0.5(4) + 0.5 = 2.5$	2		

⇒

$w = w - \alpha \frac{\partial \mathcal{L}}{\partial w}$	$b = b - \alpha \frac{\partial \mathcal{L}}{\partial b}$
0.582	0.5295

Hàm mất mát: PPTU HMM-TTGD-Ví dụ - (3)

Epoch	w	b	$\frac{\partial \mathcal{L}}{\partial w}$	$\frac{\partial \mathcal{L}}{\partial b}$
0	0.5	0.5	-8.2	-2.95
1	0.5820	0.5295	-7.2181	-2.5977
2	0.6542	0.5555	-6.3513	-2.2866
...				
9	0.9683	0.6690	-2.5412	-0.9235

- Python:

Epoch	w	b	dJ/dw	dJ/db
1	0.5820	0.5295	-8.2000	-2.9500
2	0.6502	0.5543	-6.8225	-2.4810
...				
9	0.8942	0.6484	-1.8643	-0.7913

Hàm mất mát: PPTU HMM-TTGD-Ví dụ - (4)

- Thực nghiệm:

Epoch	w	b	dJ/dw	dJ/db
1	0.5820	0.5295	-8.2000	-2.9500
2	0.6502	0.5543	-6.8225	-2.4810
3	0.7070	0.5752	-5.6751	-2.0903
...				
39999	0.8300	1.1500	0.0000	-0.0000
40000	0.8300	1.1500	0.0000	-0.0000

Sau epoch = 39999: w = 0.8300, b = 1.1500

Python

```

=== Classifier model (full
Linear Regression Model
y =
    0.83 * x +
    1.15

```

weka

	epoch	w	b
Weka	TU	0.83	1.15
Python	10000 (40000)	0.83	1.15

Hàm mất mát: PPTU HMM-TT OLS - (1)

- **Đầu vào:** 1 BĐV có m mẫu $\mathbf{x} = [x_1, x_2, \dots, x_m]$, và 1 BDR $\mathbf{y} = [y_1, y_2, \dots, y_m]$
- **Đầu ra:** GTTU của w, b (dùng trong MH HQT)
- **Phương pháp:**
- **Bước 1.** (Tính GTTB)

$$\textcircled{1} \bar{x} = \frac{1}{m} \sum_{i=1}^m x_i$$

$$\textcircled{2} \bar{y} = \frac{1}{m} \sum_{i=1}^m y_i$$

- **Bước 2.** (Tính w, b)

$$\textcircled{1} w = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

$$\textcircled{2} b = \bar{y} - w\bar{x}$$

Hàm mất mát: PPTU HMM-TT OLS - (2)

- $\leadsto$ **Đầu vào** (n BDV): ma trận

$$\mathbf{X} = (x_{ij})_{m \times n} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$$

$\mathbf{x}_j = [x_{1j}, x_{2j}, \dots, x_{mj}]^T$ véc-tơ cột, và véc-tơ cột giá trị

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m]$$

$\Rightarrow$

$$\textcircled{1} \quad W' = (\mathbf{X}'^T \mathbf{X}')^{-1} \mathbf{X}'^T \mathbf{Y}$$

$$\textcircled{2} \quad b = \bar{\mathbf{Y}} - w_1 \bar{x}_1 - \dots - w_n \bar{x}_n$$

$\leadsto$

- $\mathbf{X}'$: mở rộng của $\mathbf{X}$ bằng cách thêm véc-tơ cột 1
- $\mathbf{X}'^T$: ma trận chuyển vị của $\mathbf{X}'$
- $(\mathbf{X}'^T \mathbf{X}')^{-1}$: ma trận nghịch đảo của $\mathbf{X}'^T \mathbf{X}'$ (nếu tồn tại)
- W' : mở rộng thêm 1 thành phần của W , trong đó thành phần đầu tiên là b .

Hàm mất mát: PPTU HMM-TT OLS- Ví dụ - (1)

- 1 BĐV:

$$X = [1, 2, 3, 4], \quad y = [2, 2.8, 3.6, 4.5]$$

Bước 1: Tính giá trị trung bình

$$\bar{X} = \frac{1 + 2 + 3 + 4}{4} = 2.5,$$

$$\bar{y} = \frac{2 + 2.8 + 3.6 + 4.5}{4} = 3.225$$

Bước 2: Tính w

$$w = \frac{\sum (X_i - \bar{X})(y_i - \bar{y})}{\sum (X_i - \bar{X})^2}$$

X_i	y_i	$X_i - \bar{X}$	$y_i - \bar{y}$	$(X_i - \bar{X})(y_i - \bar{y})$	$(X_i - \bar{X})^2$
1	2	-1.5	-1.225	1.8375	2.25
2	2.8	-0.5	-0.425	0.2125	0.25
3	3.6	0.5	0.375	0.1875	0.25
4	4.5	1.5	1.275	1.9125	2.25

$$w = \frac{1.8375 + 0.2125 + 0.1875 + 1.9125}{2.25 + 0.25 + 0.25 + 2.25}$$

$$= \frac{4.15}{5} = 0.83$$

Bước 3: Tính b

$$b = \bar{y} - w\bar{X} = 3.225 - (0.83 \times 2.5)$$

$$= 3.225 - 2.075 = 1.15$$

Kết quả cuối cùng

$$w = 0.83, \quad b = 1.15$$

(PT HQT đơn $\hat{y} = 0.83X + 1.15$)

Hàm mất mát: PPTU HMM-TT OLS- Ví dụ - (2)

- 2 BDV:

X_1	X_2	Y
1	2	2.2
2	3	2.8
3	4	3.6
4	5	4.5

$$\begin{aligned}
 X &= \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \end{bmatrix} & \downarrow & X' = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 1 & 3 & 4 \\ 1 & 4 & 5 \end{bmatrix} & \Rightarrow & X'^T X' = \begin{bmatrix} 4 & 10 & 14 \\ 10 & 30 & 40 \\ 14 & 40 & 54 \end{bmatrix} \\
 Y &= \begin{bmatrix} 2.2 \\ 2.8 \\ 3.6 \\ 4.5 \end{bmatrix} & \Rightarrow & X'^T Y = \begin{bmatrix} 13.1 \\ 37.5 \\ 50.5 \end{bmatrix}
 \end{aligned}$$

$$\Rightarrow W' = (X'^T X')^{-1} X'^T Y. \Rightarrow W = \begin{bmatrix} 1.01 \\ 0.31 \\ 0.67 \end{bmatrix} b$$

$$(PT\ HQT\ đ\hat{a} \quad \hat{y} = 1.01 + 0.31X_1 + 0.67X_2)$$

k-láng giềng gần nhất: Giới thiệu

- MH *k*NN được xây dựng:
 - Lưu trữ DLHL, “gần như không học gì” từ DLHL
 - Khi cần ĐĐ DL mới, thì MH mới tính toán.
- PP HCGS đơn giản nhất của HM.
- **Ý tưởng:**
 - ① Dựa trên giả định những thứ *tương tự* hay *có tính chất gần giống nhau* sẽ nằm ở vị trí gần nhau (KG)
 - ② Sử dụng các *hàm tính KC* để xác định *k* điểm DL gần với điểm DL cần kiểm tra (mới) nhất
 - ③ Đưa ra *dự đoán* điểm DL cần kiểm tra dựa vào *k* điểm DL vừa xác định được.
- PP này có ít nhiều hạn chế, nhưng là giải pháp đầu tiên nên nghĩ đến khi giải quyết 1 bài toán HM.

k -láng giềng gần nhất: Bài toán

- k NN: phân loại, hồi quy

① Bài toán phân loại:

- **Nhãn** của DL mới x được suy ra từ k DL gần x nhất theo PP *Majority Voting* (bầu chọn theo đa số) hay *Mode* (TK).

② Bài toán hồi quy:

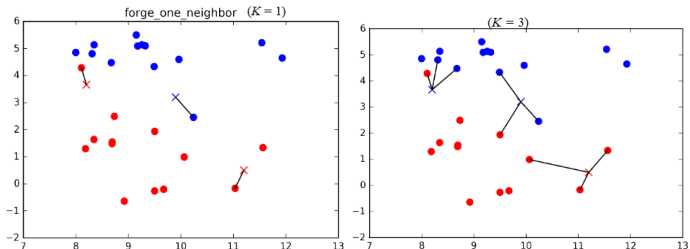
- **Giá trị** của DL mới x được tính:
 - ① $k = 1$: giá trị của DL gần x nhất
 - ② $k > 1$: GTTB hoặc TBTS của k DL gần x nhất.

k -láng giềng gần nhất: ví dụ

- **(Bài toán phân loại điểm)** GS 1 lớp học có 45 bạn và 1 bạn x , mà
 - x đã có điểm của 1 môn học nào đó là 7, nhưng không biết xếp loại môn học đó (*Giỏi, khá, trung bình, yếu*).
 - x không biết quy tắc nào để phân loại môn học đó.
- **Có 1 cách giải quyết đơn giản:**
 - Khảo sát k bạn ($k = 5$) có điểm môn học đó gần điểm x nhất và GS đã biết được xếp loại:
 - ① 6.4 - *Trung bình*
 - ② 6.6 - *Khá*
 - ③ 6.7 - *Khá*
 - ④ 7.1 - *Khá*
 - ⑤ 7.2 - *Khá*
 - $\Rightarrow$ Dự đoán: x xếp loại *Khá* (Majority Voting).

k -láng giềng gần nhất: Lưu ý

- Hình ảnh minh họa k NN với $k = 1, 3$:



- $\leadsto$ **Hàm KC trong HM (KPD L):**
 - Dùng để đo sự tương đồng giữa các điểm DL (KG)
 - Có nhiều, tùy bài toán lựa chọn KC cho phù hợp để có thể tăng hiệu suất của MH.

k -láng giếng gần nhất: Các hàm KC - (1)

- Xét 2 điểm DL (số) $x = (x_1, \dots, x_k), y = (y_1, \dots, y_k)$.

① KC Euclide:

$$d(x, y) = \left(\sum_{i=1}^k (x_i - y_i)^2 \right)^{1/2}$$

② KC Manhattan (City block, grid-based):

$$d(x, y) = \sum_{i=1}^k |x_i - y_i|$$

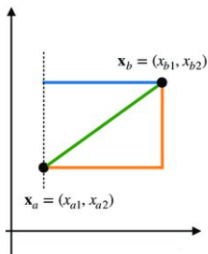
③ KC Minkowski:

$$d(x, y) = \left(\sum_{i=1}^k |x_i - y_i|^q \right)^{1/q}$$

④ KC Chebysev:

$$d(x, y) = \max_{i=1, \dots, k} |x_i - y_i|$$

k-láng giếng gần nhất: Các hàm KC - (2)



- $p = 2$ **Euclidean distance** - *L2-norm*

$$\|\mathbf{x}_a - \mathbf{x}_b\|_2 = (|x_{a1} - x_{b1}|^2 + |x_{a2} - x_{b2}|^2)^{\frac{1}{2}}$$

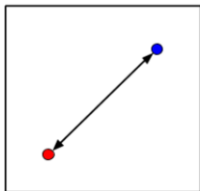
- $p = 1$ **Manhattan distance** - *L1-norm*

$$\|\mathbf{x}_a - \mathbf{x}_b\|_M = |x_{a1} - x_{b1}| + |x_{a2} - x_{b2}|$$

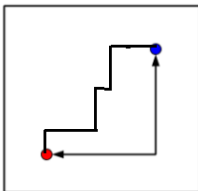
- $p = \infty$ **Chebyshev distance**

$$\|\mathbf{x}_a - \mathbf{x}_b\|_{\infty} = \max\{|x_{a1} - x_{b1}|, |x_{a2} - x_{b2}|\}$$

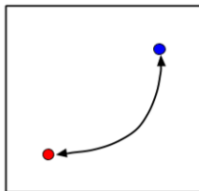
Euclidean



Manhattan



Minkowski



k -láng giềng gần nhất: Thuật toán k NN- (1)

- **Đầu vào:** Tập DLHL D và điểm DL mới x .
- **Đầu ra:** Nhãn lớp hoặc giá trị của x .
- **Phương pháp:**
- **Bước 1.** Chọn số láng giềng gần nhất $k \in \mathbb{N}$.
- **Bước 2.** Tính KC từ x đến các DL trong tập D (*Euclidean, Manhattan ...*).
- **Bước 3.**
 - ① Sắp các KC theo thứ tự không giảm
 - ② Xác định k điểm DL láng giềng gần với x nhất.
- **Bước 4. (Kết luận)**
 - ① **Bài toán phân loại:** ĐD lớp của x theo PP *Majority Voting* trong k láng giềng của x vừa tìm được.
 - ② **Bài toán hồi quy:** ĐD giá trị của x dựa trên *GTTB (TBTS)* trong k láng giềng của x vừa tìm được.

k-láng giềng gần nhất: kNN-Ví dụ - (1)

- (Tín dụng ngân hàng) Xét tập DL D về các người vay tiền:

Age	Loan	Default
25	\$40,000	N
35	\$60,000	N
45	\$80,000	N
20	\$20,000	N
35	\$120,000	N
52	\$18,000	N
23	\$95,000	Y
40	\$62,000	Y
60	\$100,000	Y
48	\$220,000	Y
33	\$150,000	Y
48	\$142,000	?

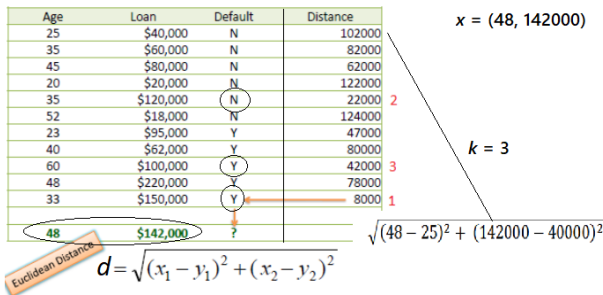
- Dự đoán người vay tiền mới

$$x = (\text{age} = 48, \text{loan} = 142000\$)$$

có khả năng vỡ nợ (*default*)?

k-láng giềng gần nhất: kNN-Ví dụ - (2)

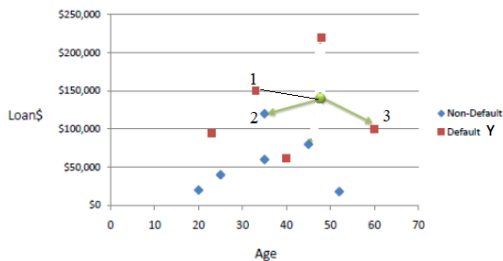
- Xét $k = 3$, KC Euclide:
- 3 láng giềng gần x nhất: (33, \$150000, Y), (60, \$100000, Y), (35, \$120000, N).



- Majority Voting: x có nhãn Y (có khả năng vỡ nợ).

k -láng giềng gần nhất: k NN-Ví dụ - (3)

- Biểu diễn D (DL 2 chiều) trong hệ tọa độ Descartes:



k-láng giềng gần nhất: *k*NN-Ví dụ - (4)

- (**Phân loại phim**) Xét DLHL D chứa một số phim với số lần đánh giá, số lần hôn trong phim và thể loại phim đó:

Tên phim	SLĐĐ	SLH	TL phim
<i>California Man</i>	3	104	Tình cảm
<i>He isn't really into dudes</i>	2	100	Tình cảm
<i>Beautiful Woman</i>	1	81	Tình cảm
<i>Kevin Longblade</i>	101	10	Hành động
<i>Robo Slayer 3000</i>	99	5	Hành động
<i>Amped II</i>	98	2	Hành động

- Dự đoán phim $x = (A, 18, 90)$ thuộc thể loại nào?

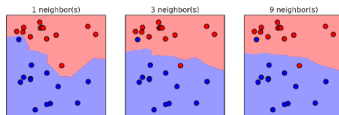
k -láng giềng gần nhất: kNN -Ví dụ - (5)

- Sử dụng KC Euclide, tính được:

Tên phim	KC
He isn't really into dudes	18.7
Beautiful Woman	19.2
California Man	20.5
Kevin Longblade	115.3
Robo Slayer 3000	117.4
Amped II	118.9

- Xét $k = 3$:
 - 3 láng giềng gần x nhất: *California Man*, *He isn't really into dudes*, *Beautiful Woman*.
 - Majority Voting: x thuộc thể loại *Tình cảm*.

k -láng giềng gần nhất: k NN-Lưu ý - (1)



① k nhỏ $\Rightarrow$ ranh giới rất phức tạp:

- Độ chính xác của MH tăng lên
- MH sẽ phức tạp hơn, tỉ lệ lỗi cao, nhạy với các outlier
 $\Rightarrow$ Dễ bị overfitting.

② k lớn $\Rightarrow$ ranh giới thoải hơn:

- Độ chính xác của MH giảm xuống
- MH đơn giản hơn, kết quả dự báo ổn định hơn do có “sự bình chọn” giữa nhiều quan sát, giảm ảnh hưởng của các outlier. Tuy nhiên, giảm đi tính chất láng giềng vì lấy TB quá nhiều $\Rightarrow$ ĐPT MH tăng lên.

k -láng giềng gần nhất: k NN-Lưu ý - (2)

- 1 Lựa chọn KC phù hợp.
- 2 Cần chuẩn hóa DL: để tránh sự chênh lệch giữa các đặc trưng.
- 3 Loại bỏ các outlier bằng các PP TXLDL.
- 4 Thường MH đạt hiệu suất tốt: $5 \leq k \leq 7$.

k-láng giềng gần nhất: Ứng dụng

- **Y tế:**

- ① Xác định bệnh lý của người bệnh.
- ② Xác định loại thuốc phù hợp.

- **Kinh tế:**

- ① Dự báo các sự kiện kinh tế trong tương lai.
- ② Dự báo tình hình thời tiết trong nông nghiệp
- ③ Xác định xu hướng thị trường chứng khoán.

- **Ngân hàng:**

- ① Xác định khả năng khách hàng chậm trả các khoản vay hoặc rủi ro tín dụng do nợ xấu.
- ② Xác định xem các giao dịch có hành vi phạm tội, lừa đảo hay không.

- **Thương mại điện tử:**

- ① Phân loại khách hàng theo sở thích cụ thể.
- ② Xây dựng hệ thống khuyến nghị dựa trên DL từ website, social media.

- **Giáo dục:**

- ① Phân loại các học sinh theo hoàn cảnh, học lực.

k -láng giềng gần nhất: Ưu nhược điểm

- **Ưu điểm:**

- ① **Triển khai, kết quả:** trực quan, dễ hiểu, dễ sử dụng. Kết quả ĐĐ DL mới rất đơn giản, đặc biệt dễ dàng cập nhật DLHL mới.
- ② **Tính linh hoạt:** không cần giả định về phân phối của DL. Có thể sử dụng cho cả bài toán phân loại, hồi quy.
- ③ **Hiệu suất:** quá trình xây dựng MH rất nhanh. Rất hiệu quả với tập DL nhỏ.
- ④ **Tham số:** không cần dựa vào các tham số, chỉ dựa trên các KC.

- **Nhược điểm:**

- ① **Cần xác định k :** Việc chọn giá trị k có ảnh hưởng lớn đến kết quả của TT
- ② **Không hiệu quả với DL lớn:** tính KC quá nhiều.
- ③ **DL phân loại (categorical):** cần phải được xử lý một cách cẩn thận vì k NN dựa trên các KC.

k -láng giềng gần nhất: Thực hành Weka - (1)

- Các bước tiến hành:

- ➊ Tải vào bộ nhớ **train set**
- ➋ Lựa chọn MH: **IBk**
- ➌ Thiết lập:
 - ➊ Cross-validation folds: 10
 - ➋ Lựa chọn các giá trị k (lẻ) khác nhau
- ➍ Kiểm thử MH với **test set**
- ➎ Lưu MH.

k-láng giềng gần nhất: Thực hành Weka - (2)

- Thực hành trên tập DL hoa *Iris.arff*:

① *Iris_train.arff* (80%)

② *Iris_test.arff* (20%)

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier is 'LinearNearestNeighbour' with the following configuration:

- Classifier: `weka.classifiers.lazy.LazyK`
- Test options: ☒ Use training set
- Cross-validation: ☒ Folds: 10
- Start view: ☒ Confusion matrix

The classifier output shows the following results:

```

Classifier output
using 1 nearest neighbour(s) for classification

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances 143
Incorrectly Classified Instances 7
Kappa statistic 0.93
Mean absolute error 0.0399
Root mean squared error 0.1747
Relative absolute error 0.9743 %
Root relative squared error 37.0495 %
Total Number of Instances 150

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  M
-----
1.000  0.000  1.000  1.000  1.000  1
0.940  0.040  0.922  0.940  0.931  0
0.920  0.080  0.939  0.920  0.929  0
Weighted Avg = 0.943  0.023  0.953  0.953  0.951  0

=== Confusion Matrix ===
 a b c <-- classified as
50 0 1 a = Iris-setosa
0 41 1 b = Iris-versicolour
0 4 46 c = Iris-virginica
  
```

Two dialog boxes are open over the main window:

- weka.gui.GenericObjectEditor** for `weka.classifiers.lazy.LazyK`: Shows 'About' and 'K-nearest neighbours classifier'.
- weka.gui.GenericObjectEditor** for `weka.classifiers.neighboursearch.LinearNearestNeighbour`: Shows 'About' and 'Class implementing the brute force search algorithm'. The 'distanceFunction' is set to 'EuclideanDistance'.

k-láng giềng gần nhất: Thực hành Python

- 1 Thực hành trên Jupyter Notebook.
- 2 HV tìm hiểu thêm.



Hồi quy tuyến tính: Giới thiệu

- **DD** giá trị của 1 biến y dựa trên giá trị của các biến khác x :
 - ① **Biến đầu ra** (y , biến phụ thuộc): muốn DD
 - ② **Biến đầu vào** (x , biến độc lập): dùng để DD
 - ③ y : hàm số **tuyến tính** của x
- Có 2 loại HQT:
 - ① **HQT đơn** (Simple): 1 BDV
 - ② **HQT đa** (Multiple): n BDV ($n > 1$).
- PP HCGS phổ biến nhất, được ứng dụng rộng rãi trong nhiều lĩnh vực.



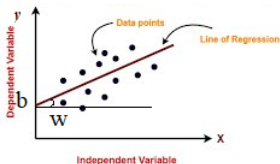
Hồi quy tuyến tính: Phân tích toán học - (1)

- **HQTT đơn**: PT có dạng

$$\hat{y} = wx + b \quad (1)$$



- ① $\hat{y}$: giá trị cần ĐĐ
- ② x : BĐV
- ③ w (weight): hệ số hồi quy (độ dốc-slope)
- ④ b (bias): hệ số tự do (hệ số chặn-intercept)



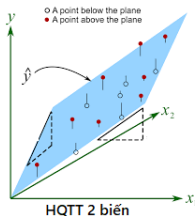
Hồi quy tuyến tính: Phân tích toán học - (2)

- **HQTT đa biến (tổng quát):**

$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b \quad (2)$$



- ① $x_1, x_2, \dots, x_n$: các BDV
- ② $w_1, w_2, \dots, w_n$: các hệ số hồi quy tương ứng



- **Mục tiêu:** ước lượng các hệ số của w, b trong MH sao cho $\hat{y} \simeq y(GT)$ nhất có thể.

Hồi quy tuyến tính: Phân tích toán học - (3)

- PT (2) biểu diễn dưới dạng ma trận:

$$\hat{\mathbf{y}} = \bar{\mathbf{X}} \mathbf{w}^T \quad (3)$$

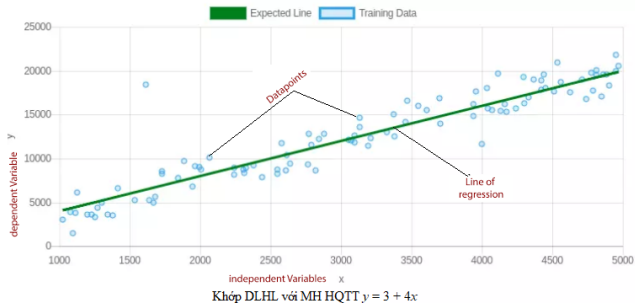


- 1 $\mathbf{X} = (x_{ij})_{m \times n}$: ma trận DL đầu vào, với mỗi cột $\mathbf{x}_j = [x_{1j}, x_{2j}, \dots, x_{mj}]^T$ là 1 biến, mỗi dòng là 1 quan sát
- 2 $\bar{\mathbf{X}}$: mở rộng của $\mathbf{X}$ bằng cách thêm véc-tơ cột 1
- 3 $\mathbf{w} = [w_1, \dots, w_n, b]$

$$\hat{\mathbf{y}} = \begin{pmatrix} x_{11} & \dots & x_{1n} & 1 \\ x_{21} & \dots & x_{2n} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{m1} & \dots & x_{mn} & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_n \\ b \end{pmatrix}$$

Hồi quy tuyến tính: Ví dụ - (1)

- MH HQT:



- DLHL $\Rightarrow$ MH HQT: $y = 3 + 4x$
- $\Rightarrow$ ĐĐ được y , với các DL mới x .

Hồi quy tuyến tính: Ví dụ - (2)

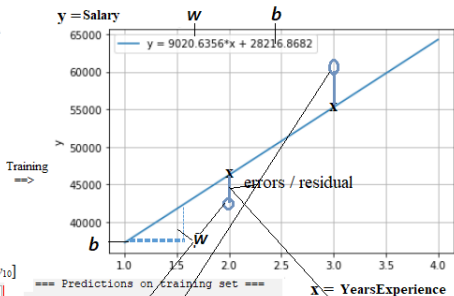
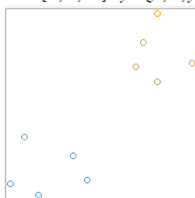
- ① Xét D_1 (tập DL về *chi tiêu* và *thu nhập* của 1 người trong năm ngoái):
 - GS HQTТ phân tích D_1 , và xác định: chi tiêu của người đó năm ngoái bằng 1 nửa thu nhập.
 - $\Rightarrow$ HQTТ sẽ DĐ: chi tiêu trong tương lai của người đó bằng cách giảm 1 nửa thu nhập trong tương lai.
- ② Xét tập DL về *tuổi* D_2 dựa vào các thông tin: $x = (x_1, x_2)$ với x_1 -*chiều cao*, x_2 -*trọng lượng* của 1 người.
 - GS QHTT giữa tuổi y và véc-tơ x : $y = 5 + 10x_1 + 13x_2$
 - $\Rightarrow$ HQTТ sẽ DĐ: tuổi của 1 người dựa vào các thông tin chiều cao và trọng lượng của người đó.

Hồi quy tuyến tính: Ví dụ - (3)

- HQTT đơn:

x		y
Relation: Salary_Data_10		DLHL
No.	1: YearsExperience Numeric	2: Salary Numeric
1	1.1	39343.0
2	1.3	46205.0
3	1.5	37731.0
4	2.0	43525.0
5	2.2	39891.0
6	2.9	56642.0
7	3.0	60150.0
8	3.2	54445.0
9	3.2	64445.0
10	3.7	57189.0

$x = [x_1, \dots, x_{10}]$ $y = [y_1, \dots, y_{10}]$



```

=== Predictions on training set ===
inst#  actual  predicted  error
1  39343  38139.567  -1203.433
2  46205  39843.694  -6261.306
3  37731  41747.822  4016.822
4  43525  46258.139  2733.139
5  39891  48062.267  8171.267
6  56642  54376.711  -2265.289
7  60150  55278.775  -4871.225
8  54445  57082.902  2637.902
9  64445  57082.902  -7362.098
10 57189  61593.22   4404.22
  
```

(Weka)

Hồi quy tuyến tính: Ví dụ - (4)

- HQTT đa:

	x_1	x_2	x_3	x_4	y
Relation: Price house-					
No.	condition Numeric	bedrooms Numeric	bathrooms Numeric	view Numeric	price Numeric
Quan sát thứ nhất — 1	3.0	0.0	0.0	2.0	1095000.0
2	3.0	0.0	0.0	0.0	1295648.0
3	3.0	1.0	1.0	0.0	0.0
4	2.0	1.0	0.75	0.0	80000.0
5	4.0	1.0	2.0	0.0	115000.0
6	4.0	1.0	1.0	0.0	129000.0
7	3.0	1.0	1.0	0.0	135000.0
8	3.0	1.0	1.0	0.0	148000.0
9	2.0	1.0	1.0	0.0	165000.0
Quan sát thứ 10 — 10	3.0	1.0	1.0	0.0	167500.0

$x = [x_{11}, x_{12}, \dots, x_{110}]$
 $y = [y_1, \dots, y_{10}]$

Hồi quy tuyến tính: Thuật toán

- **HĐĐ:**

- ① $x: \hat{y} = wx + b$

- ② $x_1, x_2, \dots, x_n: \hat{y} = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$

- **HMM (thường dùng): MSE**

- **Tối ưu HMM:**

$$MSE \rightarrow \min$$

$\Leftrightarrow$ tìm w, b trong MH sao cho $\hat{y} \simeq y$ nhất có thể:

- ① **OLS**

- Phù hợp với DL nhỏ, mặc định trong Weka

- ② **Gradient Descent:**

- Phù hợp với TDL lớn
- Gói sklearn trong Python dùng mặc định OLS, tuy nhiên khi DL lớn thì nó chuyển sang SGD (Stochastic Gradient Descent).

Hồi quy tuyến tính: Ứng dụng

- **Y tế:**
 - Dự thời gian phục hồi sau phẫu thuật, tỉ lệ sống sót của bệnh nhân hoặc tiến triển của 1 căn bệnh.
- **Chứng khoán và tài chính:**
 - Dự biến động chứng khoán và các chỉ số tài chính.
- **Kinh doanh:**
 - Dự giá nhà, giá sản phẩm, giá cổ phiếu, doanh số bán hàng, lợi nhuận kinh doanh ...
- **Web:**
 - Dự lượng truy cập trang web.
- **Dự báo thời tiết**
 - Dự báo nhiệt độ, độ ẩm và lượng mưa dự kiến.

Hồi quy tuyến tính: Ưu nhược điểm

- **Ưu điểm:**

- ① **Triển khai, kết quả:** dễ hiểu và dễ áp dụng. Không đòi hỏi nhiều kiến thức chuyên môn sâu, và rất có giá trị cho các quyết định kinh doanh.
- ② **Tính linh hoạt:** áp dụng được cho nhiều loại DL.
- ③ **Hiệu suất:** thời gian HL thường rất nhanh, đặc biệt là với DL lớn.
- ④ **Tham số:** thường ít cần các tham số tinh chỉnh, giúp giảm thời gian và công sức khi xây dựng MH.

- **Nhược điểm:**

- ① **Giả định tuyến tính:** không phải lúc nào mối quan hệ giữa các biến cũng là tuyến tính
- ② **Nhạy cảm bởi nhiễu:** các giá trị ngoại lai hoặc DL nhiễu có thể ảnh hưởng đến tính chính xác của MH
- ③ **Biến:** không thể ĐĐ cho biến phân loại, chỉ áp dụng cho các biến liên tục
- ④ **Dễ bị quá khớp**

Hồi quy tuyến tính: Thực hành Weka - (1)

- Các bước tiến hành: HQT

- ➊ Tải vào bộ nhớ **train set**
- ➋ Lựa chọn MH:
 - *SimpleLinearRegression* (đơn)
 - *LinearRegression* (đa)
- ➌ Thiết lập:
 - Cross validation folds: 10.
 - Các lựa chọn khác.
- ➍ Kiểm thử MH với **test set**
- ➎ Lưu MH.

Hồi quy tuyến tính: Thực hành Weka - (2)

- Thực hành trên tập DL: DĐ giá nhà
 - Đơn: *Price_house_new.csv*
 - Bộ: *Price_house_new_3.csv*

Relation: Price_house_new

No.	1: area Numeric	2: price Numeric
1	73.5	1.49
2	75.0	1.5
3	76.5	1.51
4	79.0	1.54
5	81.5	1.58
6	82.5	1.59
7	84.0	1.6
8	85.0	1.62
9	86.5	1.63
10	87.5	1.64
11	89.0	1.66
12	90.0	1.67
13	91.5	1.68

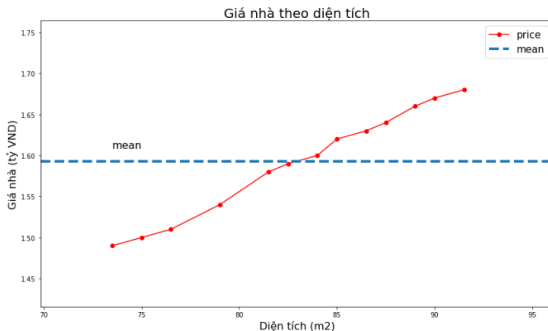
Relation: Price_house_new_3

No.	1: area Numeric	2: distance Numeric	3: price Numeric
1	73.5	20.0	1.49
2	75.0	18.0	1.5
3	76.5	17.5	1.51
4	79.0	16.0	1.54
5	81.5	15.5	1.58
6	82.5	14.5	1.59
7	84.0	12.0	1.6
8	85.0	10.0	1.62
9	86.5	8.5	1.63
10	87.5	7.5	1.64
11	89.0	5.0	1.66
12	90.0	2.0	1.67
13	91.5	1.5	1.68

Hồi quy tuyến tính: Thực hành Weka - (3)

Relation: Price_house_new

No.	1: area Numeric	2: price Numeric
1	73.5	1.49
2	75.0	1.5
3	76.5	1.51
4	79.0	1.54
5	81.5	1.58
6	82.5	1.59
7	84.0	1.6
8	85.0	1.62
9	86.5	1.63
10	87.5	1.64
11	89.0	1.66
12	90.0	1.67
13	91.5	1.68

Visualize
=>

HQTĐ đơn

Hồi quy tuyến tính: Thực hành Weka - (4)

```

Relation:      Price_house_new
Instances:     13
Attributes:    2
               area
               price

Test mode:     evaluate on training data
=== Classifier model (full training set) ===
Linear Regression Model
:
price =
|
|      0.0112 * area +
|      0.6626
|
(weka)

Regression Analysis:
Variable      Coefficient      SE of Coef      t-Stat
area          0.0112          0.0003      44.54
const         0.6626          0.0209     31.6491

Degrees of freedom = 11
R^2 value = 0.9945
Adjusted R^2 = 0.99398
F-statistic = 1983.8107

Time taken to build model: 0.19 seconds

```

MHĐĐ HQTĐ đơn

- $\leadsto b = 0.6626, w = 0.112$

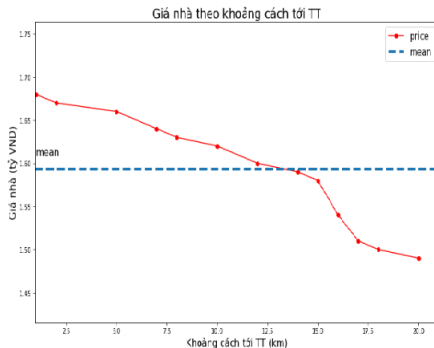
Hồi quy tuyến tính: Thực hành Weka - (5)

- Giả sử thêm biến *distance* (thông tin) về KC tới TT:

Relation: Price_house_new_2

No.	1: distance Numeric	2: price Numeric
1	20.0	1.49
2	18.0	1.5
3	17.5	1.51
4	16.0	1.54
5	15.5	1.58
6	14.5	1.59
7	12.0	1.6
8	10.0	1.62
9	8.5	1.63
10	7.5	1.64
11	5.0	1.66
12	2.0	1.67
13	1.5	1.68

Visualize
=>



- ⇒ KC tới trung tâm càng giảm thì giá nhà càng tăng ⇒ việc thêm biến *BDL distance* là rất có ý nghĩa.

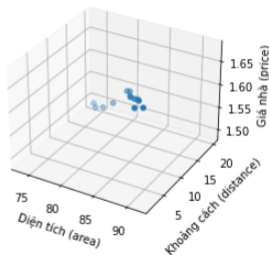
Hồi quy tuyến tính: Thực hành Weka - (6)

- ⇒ Tập DLHL *Price_house_new_3.csv* gồm 2 BĐV *area, distance* và 1 BĐR *price*:

Relation: Price_house_new_3			
No.	1: area Numeric	2: distance Numeric	3: price Numeric
1	73.5	20.0	1.49
2	75.0	18.0	1.5
3	76.5	17.5	1.51
4	79.0	16.0	1.54
5	81.5	15.5	1.58
6	82.5	14.5	1.59
7	84.0	12.0	1.6
8	85.0	10.0	1.62
9	86.5	8.5	1.63
10	87.5	7.5	1.64
11	89.0	5.0	1.66
12	90.0	2.0	1.67
13	91.5	1.5	1.68

Visualize
⇒

Biểu đồ 3D với diện tích, khoảng cách và giá nhà



HQTT bội

Hồi quy tuyến tính: Thực hành Weka - (7)

=== Classifier model (full training set) ===		Regression Analysis:			
		Variable	Coefficient	SE of Coef	t-Stat
Linear Regression Model		area	0.0123	0.001	12.2081
		distance	0.0011	0.001	1.125
		const	0.5592	0.0943	5.9327
price =		Degrees of freedom = 10			
0.0123 * area +		R^2 value = 0.9951			
0.0011 * distance +		Adjusted R^2 = 0.99413			
0.5592		F-statistic = 1016.4959			
		Time taken to build model: 0.24 seconds			
		(weka)			

MHĐĐ HQTĐ đơn

$$\bullet \leadsto b = 0.5592, w_1 = 0.0123, w_2 = 0.0011$$

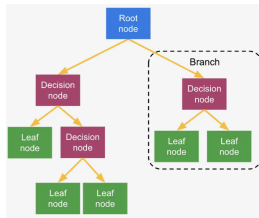
Hồi quy tuyến tính: Thực hành Python

- 1 Thực hành trên Jupyter Notebook.
- 2 HV tìm hiểu thêm.



Cây quyết định: Giới thiệu - (1)

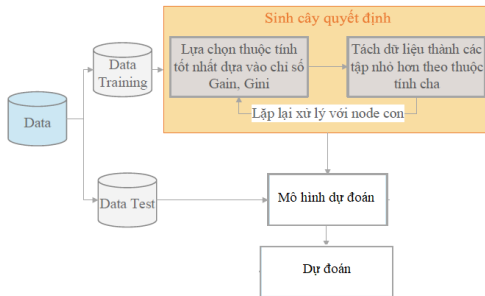
- **Cây quyết định** (Decision tree) đưa ra quyết định hoặc phân loại DL dựa trên cấu trúc dạng cây:
 - ① **Nút gốc** (Root): điểm bắt đầu cây, chứa toàn bộ DL
 - ② **Nút quyết định** (Decision): các nút trung gian đại diện cho 1 điều kiện hoặc thuộc tính cần kiểm tra
 - ③ **Cành** (Branch): các đường nối giữa các nút, thể hiện kết quả của điều kiện tại nút quyết định
 - ④ **Nút lá** (Leaf): các điểm cuối của cây, chứa quyết định cuối cùng hoặc phân loại DL.



- PP HCGS mạnh, phổ biến nhất.

Cây quyết định: Giới thiệu - (2)

- Ý tưởng:
 - ① Bắt đầu với toàn bộ DL tại nút gốc
 - ② Chọn thuộc tính tốt nhất để chia DL (*Entropy ...*)
 - ③ Lặp lại quá trình này trên từng tập DL con để tạo các nút quyết định
 - ④ Dừng lại khi DL đã được phân loại tốt hoặc đạt điều kiện dừng.



Cây quyết định: Giới thiệu - (3)

- **Các TT phổ biến:**

- ① **ID3** (Iterative Dichotomiser):

- *Entropy*
 - *Information Gain*

- ② **C4.5** (Cải tiến ID3):

- *Gain Ratio*
 - Hỗ trợ cho DL thiếu, nhiễu, liên tục.

- ③ **CART** (Classification and Regression Trees):

- Phân loại: *Gini Index*
 - Hồi quy: *Phương sai*.

Cây quyết định: Các đo-Entropy - (1)

- **Entropy** của tập DL D :

$$H(D) = - \sum_{i=1}^m p_i \log_2 p_i$$

~>

- m : số lớp (nhãn) trong D
- p_i : XS (tần suất) của lớp thứ i trong D .
- **Entropy của thuộc tính** A trong D :

$$H(D, A) = \sum_{v \in \text{dom}(A)} \frac{|D_v|}{|D|} H(D_v)$$

~>

- $\text{dom}(A)$: miền giá trị của A .
- $D_v \subseteq D$: chứa các mẫu có giá trị $v \in \text{dom}(A)$.

Cây quyết định: Các đo đo-Entropy - (2)

- **Độ lợi thông tin** (*Information Gain*) của A trong D :

$$IG(D, A) = H(D) - H(D, A).$$

- $\leadsto$

- Đo mức độ giảm độ hỗn tạp của TDL sau khi phân chia theo A .
- Chọn thuộc tính để phân nhánh:

$$A^* = \arg \max_A IG(D, A)$$

Cây quyết định: Các đo đo-Hệ số Gain

- **Hệ số độ lợi** (Gain Ratio) của A trong D :

$$GR(D, A) = \frac{IG(D, A)}{SI(D, A)}$$

↪ *Thông tin phân tách* (Split Information) của A trong D :

$$SI(D, A) = \sum_{v \in \text{dom}(A)} \frac{|D_v|}{|D|} \log_2 \frac{|D_v|}{|D|}$$

• ↪

- GR : cải tiến của IG .
- Chọn A^* để phân nhánh: $GR(D, A^*) \rightarrow \max$

Cây quyết định: Các đo đo-Chỉ số Gini

- **Chỉ số Gini** (Gini index) của D :

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2$$

- $\leadsto$ Đo độ hỗn tạp của 1 TDL.
- **Chỉ số Gini** của A trong D :

$$Gini(D, A) = \sum_{v \in \text{dom}(A)} \frac{|D_v|}{|D|} Gini(D_v)$$

- **Mức độ giảm tạp chất** (Gini Reduction) của A :

$$\Delta Gini(A) = Gini(D) - Gini(D, A)$$

- Chọn thuộc tính để phân nhánh:

$$A^* = \Delta Gini(A^*) \rightarrow \max$$

Cây quyết định: Thuật toán ID3

- **Đầu vào:** Tập DL D , tập thuộc tính $A = \{A_1, \dots, A_m\}$.
- **Đầu ra:** CQĐ.
- **Phương pháp:**
- **Bước 1.**
 - ① Tính $IG(D, A_i) = H(D) - H(D, A_i) \forall i = 1, 2, \dots, m$.
 - ② Chọn A^* : $IG(D, A^*) \rightarrow \max$.
- **Bước 2.**
 - ① Tạo nút mới ứng với A^* .
 - ② Tách DL thành các tập DL con dựa trên giá trị của A^* .
 - ③ Tạo các nhánh con ứng với mỗi giá trị của A^* , và thực hiện **ID3** đệ quy trên tập DL con tương ứng.
- **Bước 3.**
 - Nếu tất cả các mẫu thuộc về cùng 1 lớp, tạo nút lá và gán lớp đó cho nút lá.
 - Nếu không còn thuộc tính nào để chia nhánh, chọn lớp phổ biến nhất và tạo nút lá.
- **Bước 4.** (*Kết luận*) Trả về CQĐ.

Cây quyết định: ID3-Ví dụ - (1)

#	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Wind</i>	<i>Play</i>
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Cây quyết định: ID3-Ví dụ - (2)

1. Chọn thuộc tính phân nhánh tốt nhất

- **Tính $H(D)$:**

- $\hookrightarrow$

- ① $p_{Yes} = 9/14$

- ② $p_{No} = 5/14$

$\Rightarrow$

$$H(D) = -(9/14) \log_2(9/14) - (5/14) \log_2(5/14) \\ \simeq 0.94$$

- Ký hiệu $[9_{Yes}, 5_{No}] = [9+, 5-]$

- **Tính $H(D, A)$ cho mỗi A :**

- $A = Outlook$: có 3 giá trị, mỗi giá trị tương ứng 1 nút con và tập con $D_r, D_o, D_s \subseteq D$.

Cây quyết định: ID3-Ví dụ - (3)

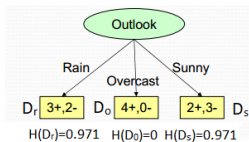
$D_r =$	#	Outlook	Temperature	Humidity	Wind	Play
	4	Rain	Mild	High	Weak	Yes
	5	Rain	Cool	Normal	Weak	Yes
	6	Rain	Cool	Normal	Strong	No
	10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No	

$D_o =$	#	Outlook	Temperature	Humidity	Wind	Play
	3	Overcast	Hot	High	Weak	Yes
	7	Overcast	Cool	Normal	Strong	Yes
	12	Overcast	Mild	High	Strong	Yes
	13	Overcast	Hot	Normal	Weak	Yes

$D_s =$	#	Outlook	Temperature	Humidity	Wind	Play
	1	Sunny	Hot	High	Weak	No
	2	Sunny	Hot	High	Strong	No
	8	Sunny	Mild	High	Weak	No
	9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes	

Cây quyết định: ID3-Ví dụ - (4)

- $\Rightarrow H(D, \text{Outlook}):$



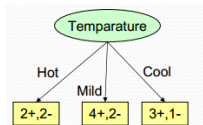
$$H(D_r) = -3/5 \cdot \log_2 3/5 - 2/5 \cdot \log_2 2/5 = 0.442 + 0.529 = 0.971$$

$$H(D_o) = -4/4 \cdot \log_2 4/4 - 0/4 \cdot \log_2 0/4 = 0 + 0 = 0$$

$$H(D_s) = -2/5 \cdot \log_2 2/5 - 3/5 \cdot \log_2 3/5 = 0.529 + 0.442 = 0.971$$

$$\Rightarrow H(D, \text{Outlook}) = 5/14 \cdot 0.971 + 4/14 \cdot 0 + 5/14 \cdot 0.971 = 0.694$$

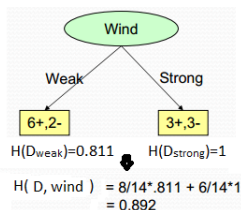
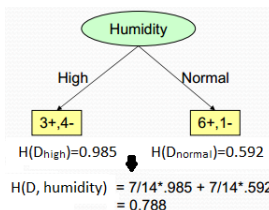
- Tương tự, $H(D, \text{Temperature}):$



$$\Rightarrow H(D, \text{temperature}) = 4/14 \cdot 1 + 6/14 \cdot 0.918 + 4/14 \cdot 0.811 = 0.911$$

Cây quyết định: ID3-Ví dụ - (5)

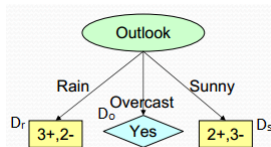
- $H(D, \text{Humidity}), H(D, \text{Wind})$:



- $\Rightarrow IG(D, \text{Outlook}) \rightarrow \max$. Chọn *Outlook*.

2. Tạo nút, chia nhánh, kiểm tra điều kiện dừng:

- Tạo nút *Outlook* với các nhánh và tập DL con:



Cây quyết định: ID3-Ví dụ - (6)

- Với tập D_r , tính tương tự:

① $H(D_r, \text{temperature}) = 2/5 * 1 + 3/5 * .918 = 0.951$

② $H(D_r, \text{humidity}) = 2/5 * 1 + 3/5 * .918 = 0.951$

③ $H(D_r, \text{wind}) = 2/5 * 0 + 3/5 * 0 = 0$

$\Rightarrow IG(D_r, \text{wind}) \rightarrow \max$. Chọn *wind* làm nút phân nhánh, với các nhánh con *weak, strong* và tập DL con tương ứng D_{rw}, D_{rs} :

#	Outlook	Temperature	Humidity	Wind	Play
4	Rain	Mild	High	Weak	Yes
$D_{rw} = 5$	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
$D_{rs} = 6$	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

- Các mẫu trên D_{rw}, D_{rs} đều thuộc cùng 1 lớp $\Rightarrow$ tạo các nút lá với nhãn là các lớp này cho các nhánh này.

Cây quyết định: ID3-Ví dụ - (7)

- Tương tự cho D_s :

① $H(D_s, \text{temperature}) = 2/5 * 0 + 2/5 * 1 + 1/5 * 0 = 0.4$

② $H(D_s, \text{humidity}) = 2/5 * 0 + 3/5 * 0 = 0$

③ $H(D_s, \text{wind}) = 2/5 * 1 + 3/5 * .918 = 0.951$

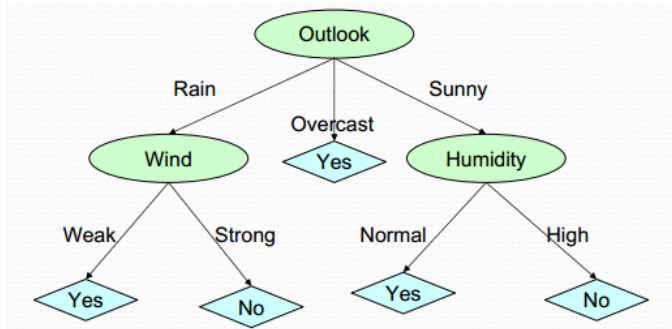
$\Rightarrow IG(D_s, \text{humidity}) \rightarrow \max$. Chọn *humidity* làm nút phân nhánh:

#	Outlook	Temperature	Humidity	Wind	Play	
D _{sh} =	1	Sunny	Hot	High	Weak	No
	2	Sunny	Hot	High	Strong	No
	8	Sunny	Mild	High	Weak	No
D _{sn} =	9	Sunny	Cool	Normal	Weak	Yes
	11	Sunny	Mild	Normal	Strong	Yes

- Các mẫu trên D_{sh} , D_{sn} đều thuộc cùng 1 lớp $\Rightarrow$ tạo các nút lá. TT dừng.
- $\Rightarrow$ ID3 dừng.

Cây quyết định: ID3-Ví dụ - (8)

3. Kết luận:



Cây quyết định: Ứng dụng

- **Y tế:**
 - ① Hỗ trợ quyết định trong chẩn đoán bệnh.
 - ② Phân loại bệnh dự đoán kết quả điều trị.
- **Tài chính và Ngân hàng:**
 - ① Đưa ra quyết định về việc cấp tín dụng cho cá nhân hoặc doanh nghiệp.
 - ② Dự đoán xu hướng thị trường tài chính và chứng khoán.
- **Kinh doanh:**
 - ① Phân loại sản phẩm và dịch vụ cho khách hàng.
 - ② Dự đoán hành vi mua hàng online của khách hàng.
 - ③ Dự đoán giá nhà.
 - ④ Tối ưu hoá chiến lược quảng cáo.
- **Thị giác máy tính:**
 - ① Phân loại hình ảnh.
 - ② Nhận dạng đối tượng.

Cây quyết định: Ưu nhược điểm

- **Ưu điểm:**
 - ① **Triển khai, kết quả:** tạo ra các quy tắc quyết định rõ ràng, dễ hiểu và dễ diễn giải kết quả của MH.
 - ② **Tính linh hoạt:** có thể xử lý tốt DL số (rời rạc) và DL phân loại. Không yêu cầu chuẩn hóa DL.
 - ③ **Hiệu suất:** cần ít DL để HL. Xây dựng nhanh và phân loại nhanh.
 - ④ **Tham số:** có thể tùy chỉnh các tham số của CQĐ để phù hợp với nhu cầu cụ thể của bài toán.
- **Nhược điểm:**
 - ① **Tương quan giữa các biến:** không biểu diễn được mối quan hệ phức tạp.
 - ② **Nhạy cảm bởi nhiễu**
 - ③ **Biến:** thường không hoạt động tốt khi xử lý các biến liên tục. Không phù hợp khi số lượng thuộc tính bùng nổ. Thường ưu tiên thuộc tính có nhiều giá trị.
 - ④ **Không ổn định:** nhạy cảm với sự biến đổi nhỏ trong DL
 - ⑤ **Dễ bị quá khớp.**

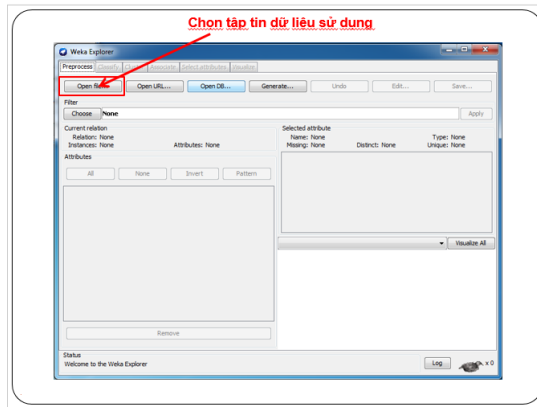
Cây quyết định: Thực hành Weka - (1)

- Các bước tiến hành:

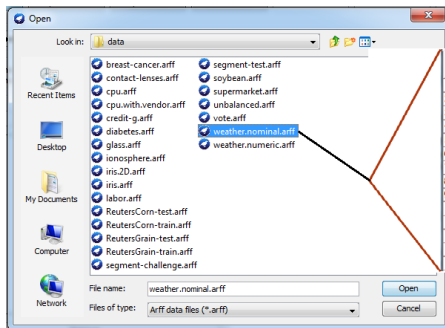
- ➊ Tải vào bộ nhớ **train set**
- ➋ Lựa chọn MH: **J48**
- ➌ Thiết lập:
 - Use training set.
 - Các lựa chọn khác: so sánh *kết quả dự đoán* với *giá trị thực* (*More option/ Output predictions: PlainText/Start*)
- ➍ Kiểm thử MH với **test set**
- ➎ Visualize CQĐ.
- ➏ Lưu MH.

Cây quyết định: Thực hành Weka - (2)

- Thực hành trên tập DL *weather.arff*:
 - weather_train.arff*
 - weather_test_1.arff*, *weather_test_2.arff*



Cây quyết định: Thực hành Weka - (3)



No.	outlook Nominal	temperature Nominal	humidity Nominal	windy Nominal	play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Cây quyết định: Thực hành Weka - (4)

Sử dụng
luôn tập
dữ liệu
học đưa
vào để
kiểm tra
mô hình

Sử dụng
tập dữ
liệu khác
để kiểm
tra mô
hình

Sử dụng
một số
% dữ
liệu đưa
vào để
học và
sử dụng
phần
còn lại
để kiểm
tra mô
hình

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose **Id3**

Test options

☒ Use training set

☐ Supplied test set Set...

☐ Cross-validation Folds 10

☐ Percentage split % 66

More options...

(Nom) play

Start Stop

Result list (right-click for options)

14:56:30 - trees.id3

Classifier output

Lựa chọn Thuật toán phân lớp được sử dụng (chọn thuật toán ID3)

Cây quyết định xây dựng được

```

outlook = sunny
| humidity = high: no (3.0)
| humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
| wind = strong: no (1.0)
| wind = weak: yes (3.0)
  
```

Time taken to build model: 0 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	12	85.7143 %
Incorrectly Classified Instances	2	14.2857 %
Kappa statistic	0.6889	
Mean absolute error	0.1429	
Root mean squared error	0.378	
Relative absolute error	30 %	
Root relative squared error	76.6097 %	
Total Number of Instances	14	

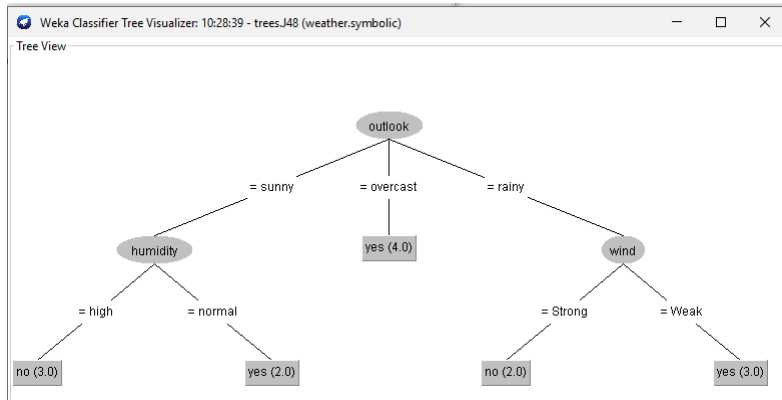
Status OK

Log x 0

Chỉ định thuộc tính (nhân) phân lớp

Cây quyết định: Thực hành Weka - (5)

- Visualize CQĐ:

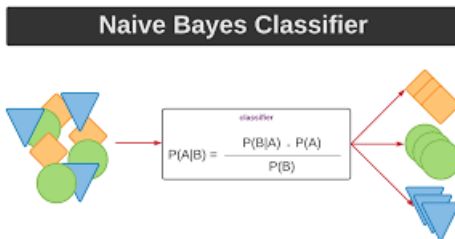


Cây quyết định: Thực hành Python

- 1 Thực hành trên Jupyter Notebook.
- 2 HV tìm hiểu thêm.

Bộ phân loại Naive Bayes: Giới thiệu

- **Bộ phân loại Naive Bayes** (NB Classifier - NBC) dựa trên *định lý Bayes* với giả định các đặc trưng đầu vào là độc lập với nhau khi biết nhãn lớp (giả định Naive - “ngây thơ”).



- PP HCGS phổ biến trong HM, được sử dụng rộng rãi trong nhiều lĩnh vực. Đặc biệt, khi gặp những bài toán cần 1 giải pháp nhanh và đơn giản.

Bộ phân loại Naive Bayes: Phân tích toán học-(1)

- Xét TDL D gồm các **mẫu đầu vào** $X = (x_1, x_2, \dots, x_n)$ và **nhãn lớp** $C \in \{C_1, C_2, \dots, C_m\}$.
- **Định lý Bayes:**

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \quad (4)$$

- **Giả định Naive:** các đặc trưng $x_1, x_2, \dots, x_n$ độc lập với nhau khi biết C :

$$P(X|C) = \prod_{i=1}^n P(x_i|C)$$

Bộ phân loại Naive Bayes: Phân tích toán học-(2)

- (4) $\Rightarrow$

$$P(C|X) \propto P(C) \prod_{i=1}^n P(x_i|C)$$

- **NBC** chọn nhãn lớp C^* có XS lớn nhất:

$$P(C^*) \prod_{i=1}^n P(x_i|C^*) \rightarrow \max$$

- **Các biến thể của Naïve Bayes và MHXS:** có 3 biến thể chính
 - ➊ Multinomial Naive Bayes
 - ➋ Gaussian Naïve Bayes
 - ➌ Bernoulli Naïve Bayes

Bộ phân loại Naive Bayes: Phân tích toán học-(3)

- **Multinomial NB:** DL rời rạc (*văn bản*) - x_i rời rạc

$$P(x_i|C) = \frac{N_{i,C} + \alpha}{N_C + \alpha d}$$



- ① $N_{i,C}$: số lần x_i xuất hiện trong các mẫu thuộc lớp C .
- ② N_C : tổng số đặc trưng trong các mẫu thuộc lớp C .
- ③ $\alpha \in \mathbb{N}$: hệ số *Laplace smoothing*, thường chọn 1.
- ④ d : số lượng giá trị có thể có của x_i .

Bộ phân loại Naive Bayes: Phân tích toán học-(4)

- **Gaussian NB:** DL liên tục- x_i số thực, và tuân theo phân phối chuẩn (Gaussian)- *đường cong hình chuông*

$$P(x_i|C) = \frac{1}{\sqrt{2\pi\sigma_C^2}} \exp\left(-\frac{(x_i - \mu_C)^2}{2\sigma_C^2}\right)$$

~>

- ① μ_C : GTTB của x_i trong lớp C .
- ② σ_C^2 : PS của x_i trong lớp C .
- **Bernoulli NB:** DL nhị phân- x_i giá trị *binary* (0, 1)

$$P(x_i|C) = \begin{cases} p_i & \text{nếu } x_i = 1 \\ 1 - p_i & \text{nếu } x_i = 0 \end{cases}$$

~> p_i : XS x_i xuất hiện trong các mẫu thuộc lớp C .

Bộ phân loại Naive Bayes: Phân tích toán học-(5)

Biến thể	Loại DL phù hợp	VD điển hình
<i>Multinomial</i>	DL đếm (rời rạc)	<i>Tần suất xuất hiện của từ trong văn bản</i>
<i>Gaussian</i>	DL số thực (liên tục)	<i>Dự đoán bệnh dựa trên chỉ số huyết áp, nhịp tim</i>
<i>Bernoulli</i>	DL nhị phân (0/1)	<i>Kiểm tra sự hiện diện của các từ khóa</i>

Bộ phân loại Naive Bayes: Thuật toán

- **Đầu vào:** Tập TDL D các mẫu với n đặc trưng, m lớp $C_1, \dots, C_m$ và mẫu X cần DĐ nhãn lớp.
- **Đầu ra:** Nhãn lớp của X .
- **Phương pháp:**
- **Bước 1.** Tính $P(C_j)$, $\forall j = 1, \dots, m$.
- **Bước 2.** Tính $P(X|C_j) = \prod_{i=1}^n P(x_i|C_j)$, $\forall j = 1, \dots, m$.
 - ① **DL liên tục:** dùng Gaussian
 - ② **DL rời rạc:** dùng Multinomial hoặc Bernoulli
- **Bước 3.** Tính $P(C_j|X) \propto P(C_j) \prod_{i=1}^n P(x_i|C_j)$, $\forall j = 1, \dots, m$.
- **Bước 4.** Chọn lớp C^* có XS lớn nhất:

$$P(X|C^*)P(C^*) \rightarrow \max.$$

Bộ phân loại Naive Bayes: ví dụ - (1)

(1) Xét TDL D về khách hàng:

<i>RID</i>	<i>age</i>	<i>income</i>	<i>student</i>	<i>credit_rating</i>	<i>Class: buys_computer</i>
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

- DĐ nhãn lớp cho khách hàng
 $X = (\text{youth}, \text{medium}, \text{yes}, \text{fair})$.

Bộ phân loại Naive Bayes: ví dụ - (2)

- $\leadsto C_1 = \text{yes}, C_2 = \text{no}$:
 - $P(C_1) = 9/14 = 0.643$
 - $P(C_2) = 5/14 = 0.357$
- $\leadsto$ **Multinomial NB**:
 - $X = (\text{youth}, \text{medium}, \text{yes}, \text{fair}), \alpha = 1.$
 - **age**: $d = 3$
 - $P(\text{youth}|C_1) = \frac{2+1}{9+3} = 0.25$
 - $P(\text{youth}|C_2) = \frac{3+1}{5+3} = 0.5$
 - **income**: $d = 3$
 - $P(\text{medium}|C_1) = \frac{5}{12} = 0.417$
 - $P(\text{medium}|C_2) = \frac{3}{8} = 0.375$
 - **student**: $d = 2$
 - $P(\text{yes}|C_1) = \frac{7}{11} = 0.636$
 - $P(\text{yes}|C_2) = \frac{2}{7} = 0.286$

Bộ phân loại Naive Bayes: ví dụ - (3)

- **credit_rating**: $d = 2$
 - $P(\text{fair}|C_1) = \frac{7}{11} = 0.636$
 - $P(\text{fair}|C_2) = \frac{3}{7} = 0.429$
- $\leadsto j = 1, 2$:

$$\begin{aligned}
 P(X|C_j) &= \prod_{i=1}^4 P(x_i|C_j) \\
 &= P(\text{youth}|C_j)P(\text{medium}|C_j)P(\text{yes}|C_j)P(\text{fair}|C_j)
 \end{aligned}$$

- $\Rightarrow$
 - $P(X|C_1) = 0.25 \times 0.417 \times 0.636 \times 0.636 = 0.042$
 - $P(X|C_2) = 0.5 \times 0.375 \times 0.286 \times 0.429 = 0.023$

Bộ phân loại Naive Bayes: ví dụ - (4)

- $\Rightarrow$
 - $P(C_1|X) \propto P(C_1)P(X|C_1) = 0.643 \times 0.042 = 0.027$
 - $P(C_2|X) \propto P(C_2)P(X|C_2) = 0.357 \times 0.023 = 0.008$
 $\Rightarrow P(C_1|X) > P(C_2|X).$
- $\Rightarrow X$ có nhãn lớp C_1 .
- $\Rightarrow$ ĐĐ khách hàng mua (yes) máy tính.

Bộ phân loại Naive Bayes: ví dụ - (5)

(2) Xét TDL về **phân loại học sinh** đậu (Pass) hoặc không (Fail) dựa trên 1 đặc trưng *điểm kiểm tra*:

ID	Điểm kiểm tra (x)	Kết quả (C)
1	85	Pass
2	90	Pass
3	78	Fail
4	65	Fail
5	95	Pass
6	70	Fail
7	88	Pass
8	76	Fail

- DĐ 1 học sinh mới có điểm kiểm tra $x = 80$.

Bộ phân loại Naive Bayes: ví dụ - (6)

- $\leadsto$

- $P(Pass) = 4/8 = 0.5$

- $P(Fail) = 4/8 = 0.5$

- $\leadsto$ *Pass*:

- x_i : 85, 90, 95, 88

- $\mu_{Pass} = \frac{85 + 90 + 95 + 88}{4} = 89.5$

- $\sigma_{Pass}^2 = \frac{(85 - 89.5)^2 + \dots + (88 - 89.5)^2}{4} = 13.25$

- $\leadsto$ *Fail*:

- x_i : 78, 65, 70, 76

- $\mu_{Fail} = \frac{78 + 65 + 70 + 76}{4} = 72.25$

- $\sigma_{Fail}^2 = \frac{(78 - 72.25)^2 + \dots + (76 - 72.25)^2}{4} = 27.19$

Bộ phân loại Naive Bayes: ví dụ - (7)

- $\leadsto$ **Gaussian NB:**
 - $x = 80$

$$P(80|Pass) = \frac{1}{\sqrt{2\pi \times 13.25}} \exp\left(-\frac{(80 - 72.25)^2}{2 \times 13.25}\right)$$

$$\simeq 0.00059$$

$$P(80|Fail) = \frac{1}{\sqrt{2\pi \times 27.19}} \exp\left(-\frac{(80 - 89.5)^2}{2 \times 27.19}\right)$$

$$\simeq 0.021$$

Bộ phân loại Naive Bayes: ví dụ - (8)

- $\Rightarrow$
 - $P(\text{Pass}|80) \propto P(\text{Pass})P(80|\text{Pass}) = 0.5 \times 0.00059 = 0.000295$
 - $P(\text{Fail}|80) \propto P(\text{Fail})P(80|\text{Fail}) = 0.5 \times 0.021 = 0.0105$
 $\Rightarrow P(\text{Fail}|80) > P(\text{Pass}|80).$
- $\Rightarrow$ Học sinh có điểm kiểm tra $x = 80$ có nhãn lớp *Fail*.
- $\Rightarrow$ ĐĐ học sinh không qua môn.

Bộ phân loại Naive Bayes: Ứng dụng

- **Y tế:**
 - ① Dự đoán tình trạng sức khỏe, nguy cơ bệnh lý.
 - ② Dự đoán kết quả điều trị từ các biểu hiện lâm sàng.
- **Phân loại email:**
 - Spam/Non-spam.
- **Phân loại Văn bản:**
 - ① Chủ đề (*tin tức, bài báo, đánh giá sản phẩm*).
 - ② Ngôn ngữ.
- **Tài chính:**
 - Dự đoán xu hướng thị trường tài chính.
- **Thị giác máy tính:**
 - Phân loại đối tượng trong ảnh (*xe hơi, người đi bộ, đèn giao thông ...*)
- **Hệ thống gợi ý/ tư vấn**

Bộ phân loại Naive Bayes: Ưu nhược điểm

- **Ưu điểm:**
 - ① **Triển khai, kết quả:** đơn giản, dễ triển khai, độ chính xác và tốc độ HL, kiểm thử rất cao.
 - ② **Tính linh hoạt:** áp dụng được cho nhiều loại DL (có cấu trúc, không có cấu trúc).
 - ③ **Hiệu suất:** xử lý tốt với các biến độc lập, đặc biệt hiệu quả với DL lớn, nhiều chiều.
- **Nhược điểm:**
 - ① **Giả định Naive:** ít gặp trong thực tế.
 - ② **Biến:**
 - ① Dễ bị ảnh hưởng bởi biến không quan trọng $\Rightarrow$ có thể dẫn đến kết quả không chính xác.
 - ② Không hoạt động tốt khi áp dụng cho các biến liên tục hoặc các biến có PP không chuẩn (cần rời rạc hoá phù hợp-biến phân loại).
 - ③ Hiệu suất MH giảm mạnh khi các thuộc tính có sự tương quan cao.

Bộ phân loại Naive Bayes: Thực hành Weka - (1)

- Các bước tiến hành:
 - ➊ Tải vào bộ nhớ **train set**
 - ➋ Lựa chọn MH: *Naive Bayes*
 - ➌ Thiết lập:
 - Cross validation folds: 10.
 - Các lựa chọn HL khác.
 - ➍ Kiểm thử MH với **test set**
 - ➎ Lưu MH.

Bộ phân loại Naive Bayes: Thực hành Weka - (2)

- Thực hành trên tập DL *labor.arff*:
 - labor_train.arff*
 - labor_test.arff*

The screenshot shows the Weka Explorer interface with the Naive Bayes classifier selected. The 'Classify' tab is active, and the 'Test options' section shows 'Cross-validation' with 'Folds' set to 10. The 'Classifier output' section displays the following results:

	2.0	6.0
full	5.0	11.0
[total]	13.0	18.0

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
 === Summary ===

Correctly Classified Instances	37	94.8718 %
Incorrectly Classified Instances	2	5.1282 %
Kappa statistic	0.8734	
Mean absolute error	0.0762	
Root mean squared error	0.2359	
Relative absolute error	19.0062 %	
Root relative squared error	52.236 %	
Total Number of Instances	39	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PR
0.909	0.036	0.909	0.909	0.909	0.873	0.938	0.7	
0.964	0.091	0.964	0.964	0.964	0.873	0.938	0.1	
Weighted Avg.	0.949	0.075	0.949	0.949	0.949	0.873	0.938	0.1

=== Confusion Matrix ===

a b <-- classified as

10	1	a = bad
1	27	b = good

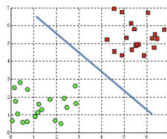
Bộ phân loại Naive Bayes: Thực hành Python

- 1 Thực hành trên Jupyter Notebook.
- 2 HV tìm hiểu thêm.

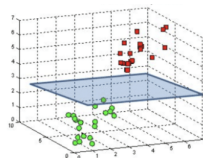


Máy véc-tơ hỗ trợ: Giới thiệu

- **Máy véc-tơ hỗ trợ** (Support Vector Machines - SVM): tìm siêu phẳng tối ưu để phân tách các lớp DL trong KG đặc trưng nhiều chiều.



R^2 - SP là đường thẳng



R^3 - SP là mặt phẳng

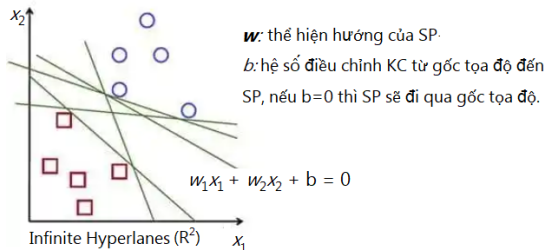
- PP có nền tảng toán học khá phức tạp, được đề xuất 1963 (*Vladimir N. Vapnik*, Nga).
- PP HCGS quan trọng, phổ biến và mạnh nhất của HM. Dùng cho cả **PL** (chủ yếu) và **HQ (SVR)**.

Máy véc-tơ hỗ trợ: Phân tích toán học - (1)

- **Siêu phẳng** (Hyperlane) trong KG $\mathbb{R}^n$:

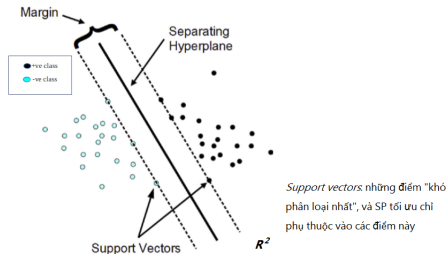
$$\mathbf{w} \cdot \mathbf{x} + b = 0 \text{ hay } \mathbf{w}^T \mathbf{x} + b = 0$$

- $\leadsto$ tích vô hướng $\mathbf{w} \cdot \mathbf{x}$:
 - $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$: véc-tơ đặc trưng đầu vào.
 - $\mathbf{w} = (w_1, \dots, w_n)$: véc-tơ pháp tuyến (trọng số).
 - b : hệ số điều chỉnh (bias).
 - $\leadsto$ 1 mặt phẳng với số chiều $n - 1$.



Máy véc-tơ hỗ trợ: Phân tích toán học - (2)

- **Lề (Margin)** của SP: KC từ SP đến các điểm DL gần nhất của mỗi lớp (Support vectors).



- **SP tối ưu:** SP có lề lớn nhất (max).

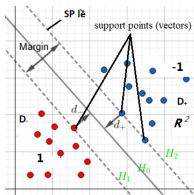
Máy véc-tơ hỗ trợ: PTTH-Tìm SP tối ưu - (3)

(1) DL phân tách tuyến tính

- Xét tập m điểm DL:

$$D = \{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathbb{R}^n; y_i \in \{-1, 1\}; i = 1, \dots, m\}$$

- $\leadsto$
 - $\mathbf{x}_i = (x_1, \dots, x_n)$: véc-tơ đặc trưng DL
 - y_i : nhãn phân lớp
- GS D được phân tách tuyến tính hoàn toàn $\Leftrightarrow \exists$ SP $H_0 : \mathbf{w}^T \mathbf{x} + b = 0$ tách D thành:
 - D_- : lớp gồm các DL có nhãn -1
 - D_+ : lớp các DL có nhãn 1 .



Máy véc-tơ hỗ trợ: PTTH-Tìm SP tối ưu - (4)



- Điều kiện phân lớp: H_0 phải thoả mãn

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \forall i$$

- KC từ $\mathbf{x}_i$ đến H_0 :

$$d_i(\mathbf{x}_i, H) = \frac{|\mathbf{w}^T \mathbf{x}_i + b|}{\|\mathbf{w}\|}$$

$\rightsquigarrow \|\mathbf{w}\| = \sqrt{w_1^2 + \dots + w_n^2}$: độ dài véc-tơ pháp tuyến.

- $\Rightarrow$ Với các support vectors ($y_i(\mathbf{w}^T \mathbf{x}_i + b) = 1$):

$$d_i = \frac{1}{\|\mathbf{w}\|}$$

Margin của H_0 :

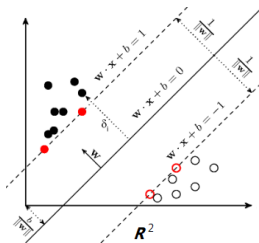
$$m = \frac{2}{\|\mathbf{w}\|}.$$

Máy véc-tơ hỗ trợ: PTTH-Tìm SP tối ưu - (5)

- $\hookrightarrow H_0$ là **SP tối ưu** $\Leftrightarrow \max \frac{2}{\|\mathbf{w}\|}$, hay $\min \frac{1}{2} \|\mathbf{w}\|^2$.
- $\equiv$ **Bài toán Hard-margin SVM** (bài toán tối ưu trong SVM):

$$\min_{\mathbf{w}, b} \left(\frac{1}{2} \|\mathbf{w}\|^2 \right)$$

$\leadsto$ ràng buộc tuyến tính: $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \forall i$.



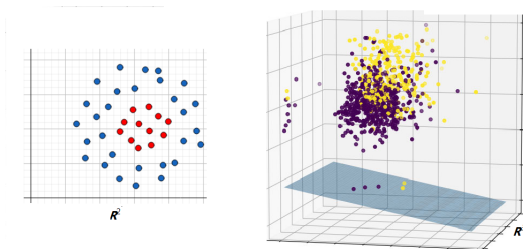
Máy véc-tơ hỗ trợ: PTTH-Tìm SP tối ưu - (6)

- Các PP giải bài toán Hard-margin SVM:

PP	Ưu điểm	Nhược điểm	Phù hợp cho
<i>QP</i>	Chính xác, TU toàn cục	Chậm với n lớn, $O(n^3)$	DL cỡ vừa
<i>SMO</i>	Hiệu quả, ít bộ nhớ	Phụ thuộc heuristic	Weka, LibSVM
<i>GD</i>	Đơn giản, dễ cài đặt	Chậm hơn SMO/QP, cần điều chỉnh α	Big data
<i>Nhân tử Lagrange</i>	Tận dụng đối ngẫu	Hội tụ chậm	LT đối ngẫu

Máy véc-tơ hỗ trợ: PTTH-Tìm SP tối ưu - (7)

(2) DL không phân tách tuyến tính

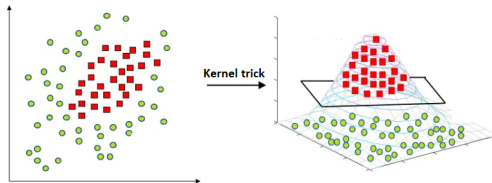
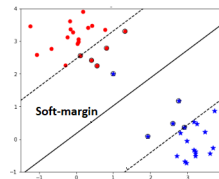
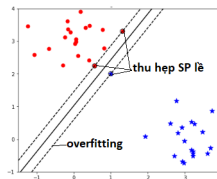


- ① Ranh giới giữa các lớp phức tạp
- ② Các lớp chồng lấn lên nhau
- ③ Thiếu đặc trưng quan trọng
- ④ KGĐT không phù hợp
- ⑤ DL phi tuyến, DL không cân bằng

Máy véc-tơ hỗ trợ: PTTH-Tìm SP tối ưu - (8)

- Một số PP giải quyết:

- 1 Thu hẹp các SP lề.
- 2 Soft-Margin SVM.
- 3 Kỹ thuật hàm nhân (Kernel trick).
- 4 Kết hợp Kernel trick và Soft-Margin SVM.



Máy véc-tơ hỗ trợ: Soft-Margin SVM - (9)

- PP nới lỏng các điều kiện lề, cho phép một số điểm DL vi phạm lề để đạt được sự cân bằng.
- **Bài toán Soft-Margin SVM:**

$$\min_{\mathbf{w}, b, \xi_i} \left(\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \right)$$

~>

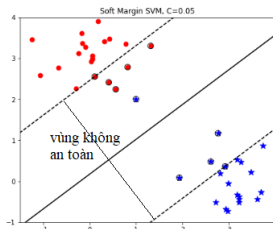
- ràng buộc tuyến tính:

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \forall i$$

- ξ_i : biến slack, đo mức độ sai lệch của $\mathbf{x}_i$ (vi phạm lề).
- C : tham số phạt (penalty).

Máy véc-tơ hỗ trợ: Soft-Margin SVM - (10)

- ⇒ Điều chỉnh C để cân bằng giữa việc *phân loại sai* và *tối đa hóa lề*.



- Cách chọn C :
 - Dùng *cross-validation* để tìm C tối ưu.
 - DL nhiều nhiễu $\Rightarrow$ chọn C nhỏ.
 - DL sạch $\Rightarrow$ chọn C lớn.

Máy véc-tơ hỗ trợ: Kỹ thuật hàm nhân - (1)

- **Kỹ thuật hàm nhân** (Kernel trick): ánh xạ DL từ KGĐT ban đầu sang 1 KGĐT mới có số chiều cao hơn:

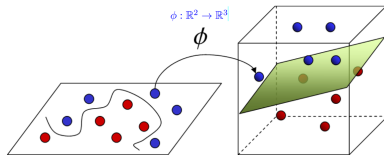
$$\phi : \mathbb{R}^n \rightarrow \mathbb{R}^d$$

- $\leadsto$

- ① $d \gg n$.
- ② $\mathbb{R}^d$: DL phân tách tuyến tính được.
- ③ Sử dụng **hàm nhân** $K(\mathbf{x}, \mathbf{y})$:

$$K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y})$$

- $\leadsto$ Không cần tính toán trực tiếp phép biến đổi $\phi(\mathbf{x})$ (tốn kém, do số chiều cao).



Máy véc-tơ hỗ trợ: Kỹ thuật hàm nhân - (2)

- Một số hàm nhân phổ biến:

1 Tuyến tính

- $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}.$

2 Đa thức

- $K(\mathbf{x}, \mathbf{y}) = (\gamma \mathbf{x}^T \mathbf{y} + r)^d, \gamma > 0, r \geq 0.$
- γ quyết định độ dốc của hàm nhân, r điều chỉnh sự dịch chuyển của SP, d là bậc của đa thức.

3 RBF (Radial basis function) hay Gaussian

- $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2), \gamma > 0.$

4 Sigmoidal

- $K(\mathbf{x}, \mathbf{y}) = \tanh(\gamma \mathbf{x}^T \mathbf{y} + r), \gamma > 0, r \geq 0.$

Máy véc-tơ hỗ trợ: Kỹ thuật hàm nhân-Ví dụ - (1)

Điểm	x_1	x_2	Label (y)
A	0	0	0
B	0	1	1
C	1	0	1
D	1	1	0

• **Kernel Polynomial bậc 2:**

$$K(x, y) = (x^T y + 1)^2$$

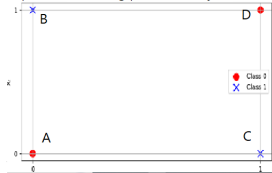
• **Tính kernel :**

◦ Với $x = (x_1, x_2)$ và $y = (y_1, y_2)$, ta có:

$$K(x, y) = (x_1 y_1 + x_2 y_2 + 1)^2$$

$$\rightarrow K(x, y) = x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 x_2 y_1 y_2 + 2x_1 y_1 + 2x_2 y_2 + 1$$

Tập DL XOR, không phân tách tuyến tính được)



◦ Nếu tính $\Phi(x)$ trực tiếp, nó sẽ tương đương với:

$$\Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

→ **6 chiều** thay vì 2 chiều ban đầu

Kết Quả Sau Khi Ánh Xạ

- Dữ liệu trong không gian mới (**6D**) có thể phân tách bằng siêu phẳng.
- **Không cần tính $\Phi(x)$** , chỉ cần dùng $K(x, y)$ trong SVM.

Máy véc-tơ hỗ trợ: Kỹ thuật hàm nhân-Ví dụ - (2)

Kiểm tra:

Xét 2 điểm $a = (0, 1)$ và $b = (1, 0)$:

Tính Trực Tiếp $\Phi(a)$ và $\Phi(b)$

$$\Phi(a) = (0^2, 1^2, \sqrt{2} \cdot 0 \cdot 1, \sqrt{2} \cdot 0, \sqrt{2} \cdot 1, 1) = (0, 1, 0, 0, \sqrt{2}, 1)$$

$$\Phi(b) = (1^2, 0^2, \sqrt{2} \cdot 1 \cdot 0, \sqrt{2} \cdot 1, \sqrt{2} \cdot 0, 1) = (1, 0, 0, \sqrt{2}, 0, 1)$$

→ Tích vô hướng:

$$\langle \Phi(a), \Phi(b) \rangle = 0 \cdot 1 + 1 \cdot 0 + 0 \cdot 0 + 0 \cdot \sqrt{2} + \sqrt{2} \cdot 0 + 1 \cdot 1 = \boxed{1}$$

Dùng Kernel

$$K(a, b) = (a^T b + 1)^2 = (0 \cdot 1 + 1 \cdot 0 + 1)^2 = \boxed{1}$$

Máy véc-tơ hỗ trợ: Kỹ thuật hàm nhân-Ví dụ - (3)

Sử dụng ánh xạ:

$$\Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

Tính toán cho từng điểm:

- **Điểm A (0,0):** $\Phi(A) = (0, 0, 0, 0, 0, 1)$
- **Điểm B (0,1):** $\Phi(B) = (0, 1, 0, 0, \sqrt{2}, 1)$
- **Điểm C (1,0):** $\Phi(C) = (1, 0, 0, \sqrt{2}, 0, 1)$
- **Điểm D (1,1):** $\Phi(D) = (1, 1, \sqrt{2}, \sqrt{2}, \sqrt{2}, 1)$

tìm **siêu phẳng** trong không gian 6D có dạng:

$$f(\Phi(x)) = w_1x_1^2 + w_2x_2^2 + w_3\sqrt{2}x_1x_2 +$$

$$w_4\sqrt{2}x_1 + w_5\sqrt{2}x_2 + w_6 \cdot 1 + b$$

sao cho:

- $f(\Phi(x)) \geq 0$ với lớp 1 (B, C).
- $f(\Phi(x)) < 0$ với lớp 0 (A, D).

Giải pháp:

- Chọn trọng số $\mathbf{w} = (1, 1, -2, 0, 0, -1)$ và $b = 0$:

$$f(\Phi(x)) = x_1^2 + x_2^2 - 2x_1x_2 - 1$$

Kiểm tra:

- **Điểm A (0,0):** $f(\Phi(A)) = 0 + 0 - 0 - 1 = -1$ (Class 0)
- **Điểm B (0,1):** $f(\Phi(B)) = 0 + 1 - 0 - 1 = 0$ (Class 1)
- **Điểm C (1,0):** $f(\Phi(C)) = 1 + 0 - 0 - 1 = 0$ (Class 1)
- **Điểm D (1,1):** $f(\Phi(D)) = 1 + 1 - 2 - 1 = -1$ (Class 0)

→ **Kết quả:** Dữ liệu đã được **phân tách tuyến tính** trong không gian 6D!

- **Minh Họa Trong Không Gian 3D (Rút Gọn)**

Để dễ hình dung, ta chỉ xét **3 thành phần chính** của $\Phi(x)$:

$$\Phi_{3D}(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

Trong không gian 3D, siêu phẳng có dạng:

$$w_1x_1^2 + w_2x_2^2 + w_3\sqrt{2}x_1x_2 + b = 0$$

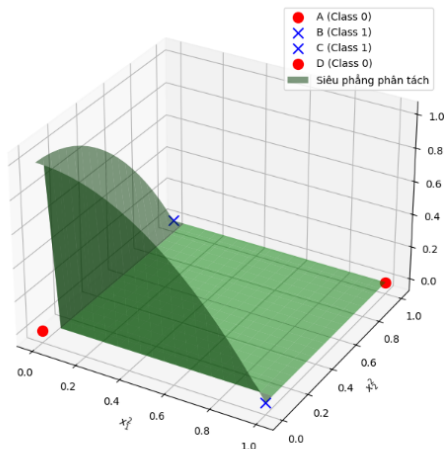
Chọn **trọng số** để phân tách:

- $w_1 = 1, w_2 = 1, w_3 = -2, b = -1$:

$$x_1^2 + x_2^2 - 2\sqrt{2}x_1x_2 - 1 = 0$$

Máy véc-tơ hỗ trợ: Kỹ thuật hàm nhân-Ví dụ - (4)

Siêu Phẳng Phân Tách Trong Không Gian 3D Sau Ánh Xạ



$$x_1^2 + x_2^2 - 2\sqrt{2}x_1x_2 - 1 = 0$$

Máy véc-tơ hỗ trợ: Ứng dụng

- **Y tế:**
 - Dự đoán các bệnh lý như tiểu đường, ung thư, tim mạch.
- **Tài chính:**
 - Dự đoán xu hướng thị trường tài chính, dự đoán giá cổ phiếu.
- **Phân loại ảnh và nhận dạng vật thể:**
 - Áp dụng trong hệ thống nhận dạng như khuôn mặt, chữ viết tay, biển số xe ...
- **Phân loại văn bản và phân loại email:**
 - Phân loại văn bản vào các nhóm như phân loại tin nhắn spam và không spam, phân loại văn bản theo chủ đề, phân loại tài liệu.
- **Nhận dạng giọng nói:**
 - Nhận dạng người nói, nhận dạng từ hoặc dịch giọng nói.

Máy véc-tơ hỗ trợ: Ưu nhược điểm

- **Ưu điểm:**
 - ① **Hiệu suất tốt:**
 - Trong KG chiều cao, DL lớn.
 - ② **Tính linh hoạt trong việc chọn hàm nhân:**
 - Có thể sử dụng nhiều hàm nhân khác nhau để phù hợp với đặc điểm của DL.
 - ③ **Tính tổng quát cao:**
 - Thường cho kết quả tốt trên DL mới, không nhất thiết là DL được sử dụng trong quá trình HL.
- **Nhược điểm:**
 - ① **Diễn giải kết quả kém:**
 - SVM tạo ra các MH phân loại phức tạp, làm cho việc diễn giải kết quả trở nên khó khăn.
 - ② **ĐPT tính toán cao:**
 - Tìm SP tối ưu tốn thời gian khi số lượng mẫu lớn.
 - ③ **Lựa chọn hàm nhân kernel:**
 - Lựa chọn hàm nhân phù hợp là 1 thách thức và có thể ảnh hưởng đến hiệu suất của SVM.
 - ④ **Không xử lý tốt với DL có nhiều nhiễu.**

Máy véc-tơ hỗ trợ: Thực hành với Weka

- Các bước tiến hành:
 - ➊ Tải vào bộ nhớ **train set**
 - ➋ Lựa chọn MH: **SMO**
 - ➌ Thiết lập:
 - 10-fold cross validation.
 - Các lựa chọn khác.
 - ➍ Kiểm thử MH với **test set**
 - ➎ Lưu MH.
- Thực hành trên tập DL *labor*:
 - ➊ *labor_train.arff*
 - ➋ *labor_test.arff*.

Máy véc-tơ hỗ trợ: Thực hành với Python

- 1 Thực hành trên Jupyter Notebook.
- 2 HV tìm hiểu thêm.



Nội dung môn học

- ① Tổng quan về học máy
- ② Toán, Python và Weka
- ③ Học có giám sát
 - Hàm mất mát
 - k -láng giềng gần nhất
 - Hồi quy tuyến tính
 - Cây quyết định
 - Bộ phân loại Naïve Bayes
 - Máy véc-tơ hỗ trợ
- ④ Học không giám sát
 - Phân cụm k -mean
 - Giảm chiều bằng tập thô
- ⑤ Học kết hợp
- ⑥ Tài liệu tham khảo

Chương 4. Học không giám sát

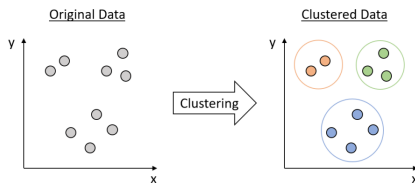
- 1 Phân cụm *k-mean*
- 2 Giảm chiều *Rough set*
- 3 Luật kết hợp *Apriori*



- $\leadsto$ Tự đọc: *Apriori*

Phân cụm k -mean: Giới thiệu - (1)

- **Phân cụm** (Clustering): 1 kỹ thuật trong HM dùng để nhóm các điểm DL có đặc điểm tương đồng thành các **cụm** (clusters) sao cho:
 - ① **Tối đa hóa** sự tương đồng trong mỗi cụm
 - ② **Tối thiểu hóa** sự tương đồng giữa các cụm



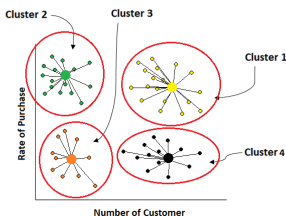
- **Mục tiêu chính của PC:**
 - ① Khám phá cấu trúc ẩn trong DL
 - ② Giảm kích thước DL
 - ③ Phân đoạn DL
 - ④ TXL cho các bài toán HM khác

Phân cụm k -mean: Giới thiệu - (2)

- Các PPC và TT:
 - ① **PC dựa trên KC:** k -mean, k -medoids
 - ② **PC phân cấp:** Agglomerative, Divisive
 - ③ **PC dựa trên mật độ:** DBSCAN, OPTICS
 - ④ **PC dựa trên MHXS:** GMM
 - ⑤ **PC theo đồ thị:** Spectral Clustering
- Một số loại cụm phổ biến:
 - ① **Cụm hình cầu:** các điểm nằm xung quanh 1 tâm trung bình
 - ② **Cụm có mật độ cao:** các điểm có mật độ cao được nhóm lại, các điểm thưa thớt bị coi là nhiễu
 - ③ **Cụm có cấu trúc phân cấp:** tạo ra 1 cây phân cấp từ các cụm nhỏ đến cụm lớn
 - ④ **Cụm không đồng nhất:** không có hình dạng cụ thể, dùng trong DL phức tạp.

Phân cụm k -mean: Giới thiệu - (3)

- **TT k -mean** (*Mac-Queen*, 1967): nhóm DL thành k cụm dựa trên KC giữa các điểm DL.
- **Ý tưởng của k -mean:**
 - ① Chia DL thành k cụm sao cho các điểm trong cùng cụm có đặc điểm tương đồng nhất
 - ② Dùng TBC để xác định trung tâm (centroid) mỗi cụm
 - ③ Lặp lại việc gán cụm, và cập nhật trung tâm cho đến khi hội tụ.



- $\leadsto$ PPC phổ biến trong HKGS.

Phân cụm k -mean: Các hàm KC - (1)

- Các KC đã biết: **Euclide, Manhattan, Minkowski, Chebysev**
- **KC Hamming**: đo số vị trí khác nhau giữa 2 chuỗi có cùng độ dài

$$d_H(x, y) = \sum_{i=1}^n \delta(x_i, y_i)$$

~>

- $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n)$: 2 chuỗi có độ dài n
- Hàm *delta Kronecker*:

$$\delta(x_i, y_i) = \begin{cases} 1 & \text{nếu } x_i \neq y_i \\ 0 & \text{ngược lại} \end{cases}$$

Phân cụm k -mean: Các hàm KC - (2)

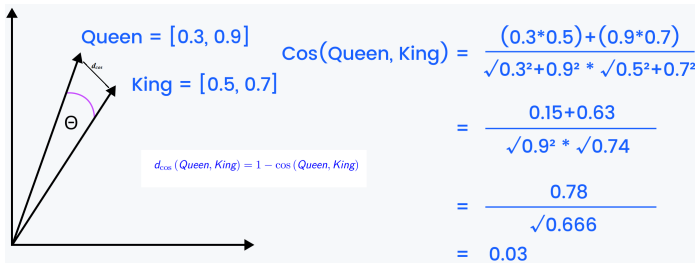
- **KC cosin**: đo độ tương đồng của 2 véc-tơ dựa vào góc giữa chúng trong KG nhiều chiều

$$d_{\cos}(a, b) = 1 - \cos(a, b) = \frac{a \cdot b}{\|a\| \|b\|}$$



- $a \cdot b = \sum a_i b_i$: tích vô hướng
- $\|a\| = \sqrt{\sum a_i^2}$, $\|b\| = \sqrt{\sum b_i^2}$: độ dài (chuẩn Euclide)
- $\cos(a, b)$: độ tương đồng *cosin*
- $d_{\cos}(a, b) \in [0, 2]$:
 - 0: 2 véc-tơ trùng nhau
 - 1: 2 véc-tơ trực giao
 - 2: 2 véc-tơ đối nhau.

Phân cụm k -mean: Các hàm KC - (3)



Phân cụm k -mean: Thuật toán - (1)

- **Vào:** Tập DL $X = \{x_1, x_2, \dots, x_n\}$.
- **Ra:** Các cụm của X .
- **Phương pháp:**
- **Bước 1.** Chọn k và các tâm $c_1, \dots, c_k$ cho k cụm $C_1, \dots, C_k$ từ X .
- **Bước 2.** Lặp lại quá trình sau cho đến khi **hội tụ**:
 - ① **Phân bổ mỗi ĐT vào cụm gần nhất:**
 - ① Tính các KC $d(x_j, c_i)$, $j = 1, \dots, n$; $i = 1, \dots, k$.
 - ② Nhóm các x_j vào cụm có tâm c_i : $d(x_j, c_i) \rightarrow \min$.
 - ② **Cập nhật lại tâm:** $\forall i = 1, \dots, k$

$$c_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$$

Phân cụm k -mean: TT-Điều kiện hội tụ

① Tâm cụm không thay đổi:

$$c_k^{(t+1)} = c_k^{(t)}$$

$\leadsto c_k^{(t)}$: tâm của cụm k tại vòng lặp t

② Không có sự thay đổi trong việc gán cụm:

$$S_k^{(t+1)} = S_k^{(t)}$$

$\leadsto S_k^{(t)}$: tập hợp các điểm DL cụm k tại vòng lặp t

③ Sai số hội tụ nhỏ hơn 1 ngưỡng hội tụ ϵ :

$$\max ||c_k^{(t+1)} - c_k^{(t)}|| < \epsilon$$

$\leadsto ||.||$: chuẩn Euclide, thường $\epsilon := 10^{-6}$

④ Số vòng lặp đạt tới tối đa:

$$t > T_{\max}$$

$\leadsto T_{\max}$: giá trị giới hạn, thường $T_{\max} \in [100 - 500]$.

Phân cụm k -mean: TT-Số cụm, tâm cụm

① Số cụm k :

- Chọn NN, hoặc theo kinh nghiệm
- Phương pháp Elbow

② Tâm cụm:

- Khởi tạo NN
- k -mean++

Phân cụm k -mean: TT-Tiền XLDL

① DL noise, outlier:

- *Noise*: dùng k -medoid
- *Outlier*: loại bỏ trước khi thực hiện TT

② Có nhiều đơn vị đo: chuẩn hóa DL

Phân cụm k -mean: TT-Ví dụ - (1)

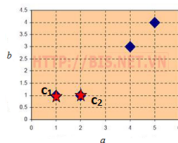
- Xét tập DL gồm 4 loại thuốc $X = \{u, v, x, y\}$ và 2 đặc trưng a (weight index), b (pH):

X	a	b
u	1	1
v	2	1
x	4	3
y	5	4

- Phân cụm tập DL trên bằng k -mean với số cụm là 2.

Phân cụm k -mean: TT-Ví dụ - (2)

- Ký hiệu:
 - $u = (1, 1) = (u_{11}, u_{12})$
 - $v = (2, 1) = (v_{11}, v_{12})$
 - $x = (4, 3) = (x_{11}, x_{12})$
 - $y = (5, 4) = (y_{11}, y_{12})$
- Chọn NN các tâm ban đầu:
 - $c_1 = u = (c_{11}, c_{12})$: tâm cụm C_1
 - $c_2 = v = (c_{21}, c_{22})$: tâm cụm C_2



Phân cụm k -mean: TT-Ví dụ - (3)

- **Phân bổ mỗi ĐT vào cụm gần nhất (lần 1):**

- KC Euclide mỗi ĐT đến tâm các cụm:

- $d(u, c_1) = 0$

- $d(v, c_1) = \sqrt{(v_{11} - c_{11})^2 + (v_{12} - c_{12})^2} = 1$

- $d(x, c_1) = \sqrt{(x_{11} - c_{11})^2 + (x_{12} - c_{12})^2} = 3.61$

- $d(y, c_1) = \sqrt{(y_{11} - c_{11})^2 + (y_{12} - c_{12})^2} = 5$

- $d(u, c_2) = \sqrt{(u_{11} - c_{21})^2 + (u_{12} - c_{22})^2} = 1$

- $d(v, c_2) = 0$

- $d(x, c_2) = \sqrt{(x_{11} - c_{21})^2 + (x_{12} - c_{22})^2} = 2.83$

- $d(y, c_2) = \sqrt{(y_{11} - c_{21})^2 + (y_{12} - c_{22})^2} = 4.24$

⇒ Ma trận KC:

$$M_1 = \begin{pmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{pmatrix}$$

- ⇒ So sánh các KC và nhóm vào cụm:

- u thuộc cụm C_1

- v, x, y thuộc cụm C_2 .

Phân cụm k -mean: TT-Ví dụ - (4)

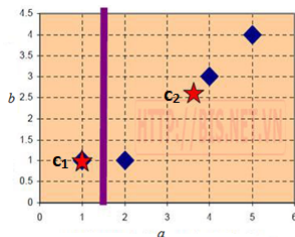
- **Cập nhật lại tâm các cụm (lần 1):**

- C_1 : chỉ có 1 ĐT $\Rightarrow c_1$ không đổi.

- C_2 :

- $c_{21} = \frac{2+4+5}{3} = \frac{11}{3}, c_{22} = \frac{1+3+4}{3} = \frac{8}{3}$

- $\Rightarrow c_2 = (\frac{11}{3}, \frac{8}{3})$



Phân cụm k -mean: TT-Ví dụ - (5)

- **Phân bổ lại các ĐT vào cụm (lần 2):**

- Tính lại các KC:

$$M_2 = \begin{pmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{pmatrix}$$

- So sánh các KC và nhóm vào cụm:

- u, v thuộc C_1
- x, y thuộc C_2

Phân cụm k -mean: TT-Ví dụ - (6)

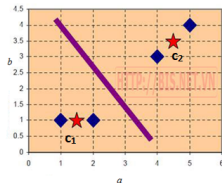
- Cập nhật lại tâm các cụm (lần 2):

- C_1 : $c_{11} = \frac{1+2}{2} = \frac{3}{2}$, $c_{12} = \frac{1+1}{2} = 1$

$$\Rightarrow c_1 = \left(\frac{3}{2}, 1\right)$$

- C_2 : $c_{21} = \frac{4+5}{2} = \frac{9}{2}$, $c_{22} = \frac{3+4}{2} = \frac{7}{2}$

$$\Rightarrow c_2 = \left(\frac{9}{2}, \frac{7}{2}\right)$$



Phân cụm k -mean: TT-Ví dụ - (7)

- **Phân bổ lại các ĐT vào cụm (lần 3):**

- Tính lại các KC:

$$M_3 = \begin{pmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{pmatrix}$$

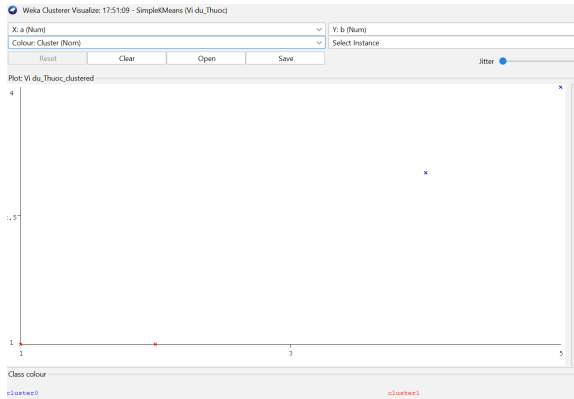
- So sánh các KC và nhóm vào cụm: các cụm không có sự thay đổi $\Rightarrow k$ -means dừng.

- **Kết quả phân cụm:**

X	a	b	Cụm
u	1	1	C_1
v	2	1	C_1
x	4	3	C_2
y	5	4	C_2

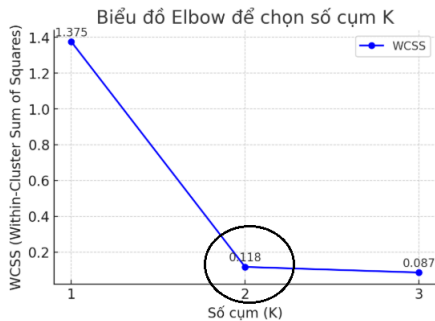
Phân cụm k -mean: TT-Ví dụ - (8)

- Visualize kết quả phân cụm bằng Weka:



Phân cụm k -mean: TT-Ví dụ - (9)

- Đồ thị WCSS để chọn k tối ưu:



Phân cụm k -mean: Ứng dụng

- **Y tế:**
 - Phân loại bệnh lý, dự đoán kết quả điều trị.
- **Tài chính:**
 - Phân loại các khoản chi tiêu, giao dịch tài chính.
- **Kinh doanh:**
 - Phân loại các nhóm khách hàng, đánh giá rủi ro.
- **Thị giác máy tính:**
 - Phân loại hình ảnh, đối tượng trong ảnh.
- **Phân loại văn bản và DL Web:**
 - Phân loại tài liệu, phân loại người dùng Web.
- **Nhận dạng tiếng nói:**
 - Phân loại, nhận dạng các mẫu âm thanh.
- **Sinh học:**
 - Phân loại thực vật, động vật, các mẫu gen.

Phân cụm k -mean: Ưu nhược điểm

- **Ưu điểm:**

- ① **Triển khai, kết quả:** dễ triển khai, đặc biệt đối với DL có kích thước lớn. Cho kết quả dễ hiểu.
- ② **Tính linh hoạt:** áp dụng được cho nhiều loại DL (số, văn bản, hình ảnh).
- ③ **Hiệu suất:** cao khi các cụm có kích thước và hình dạng tương đối đồng đều (*dạng cầu*), hoặc số cụm nhỏ, hoặc DL có tính chất liên tục.
- ④ **Tính hợp nhất:** có thể cập nhật DL dễ dàng khi có DL mới thêm vào, mà không cần phải HL lại toàn bộ MH.

- **Nhược điểm:**

- ① **Phụ thuộc vào giá trị khởi tạo k**
- ② **Nhạy cảm với trọng tâm ban đầu**
- ③ **Nhạy cảm với nhiễu**
- ④ **Không phù hợp với tập DL:** có các cụm với hình dạng phi tuyến tính, có hình dạng không đồng nhất, kích thước khác nhau.

Phân cụm *k*-mean: Thực hành Weka

- Các bước tiến hành:
 - ➊ Tải vào bộ nhớ **train set**
 - ➋ Lựa chọn MH: **SimpleKMeans** (nhóm **Cluster**)
 - ➌ Thiết lập:
 - 10-fold cross validation.
 - Các lựa chọn khác.
 - ➍ Kiểm thử MH với **test set**
 - ➎ Lưu MH.
- Thực hành trên:
 - ➊ *diabetes.arff*
 - ➋ *credit-g.arff*.

Phân cụm k -mean: Thực hành Python

- 1 Thực hành trên Jupyter Notebook.
- 2 HV tìm hiểu thêm.



Giảm chiều bằng tập thô: Giới thiệu

- **Tập thô** (Rough set) **cổ điển** (Pawlak, 1982): công cụ toán học cực kỳ mạnh để đo lường mức độ quan trọng của các biến mà không cần thông tin bổ sung (như *PPXS*, hay *hàm thuộc tính mờ*):

Bệnh nhân	Sốt (S)	Ho (H)	Bệnh (B)
u_1	cao	có	<i>có</i>
u_2	thấp	không	<i>không</i>
u_3	cao	không	<i>có</i>

- Thông thường, người ta sử dụng RS như 1 bước TXL:
 - ① DL thô (*nhieu nhieu, nhieu cot*).
 - ② Áp dụng thuật toán RS để tìm tập rút gọn (*giảm chiều*).
 - ③ DL sau khi giảm chiều $\Rightarrow$ đưa vào các MHHM (*SVM, Neural Networks ...*).

Giảm chiều bằng tập thô: Các khái niệm-1

- **Hệ thống tin** (Information System):

$$IS = (U, A, \{V_a\}_{a \in A})$$

- ① U : tập các *đối tượng* (vũ trụ), hữu hạn và $\neq \emptyset$
- ② A : tập các *thuộc tính*, hữu hạn và $\neq \emptyset$.
- ③ $\{V_a\}_{a \in A}$: họ các *tập giá trị* tương ứng với từng $a \in A$.
 Mỗi $a \in A$ xác định 1 hàm số $a : U \rightarrow V_a$, gán mỗi ĐT u 1 giá trị $a(u) \in V_a$.

- Một HTT đơn giản:

Bệnh nhân	Sốt (S)	Ho (H)	Bệnh (B)
u_1	cao	có	có
u_2	thấp	không	không
u_3	cao	không	có

$\leadsto U = \{u_1, u_2, u_3\}$, $A = \{S, H, B\}$, $S(u_1) = \text{cao}$,
 $H(u_3) = \text{không} \dots$

Giảm chiều bằng tập thô: Các khái niệm-2

- **Bảng quyết định** (Decision Table):

$$DT = (U, A = C \cup \{d\}, \{V_a\}_{a \in A}) \stackrel{\text{hay}}{=} (U, C \cup \{d\})$$

- C : các thuộc tính điều kiện (Input)
- d : thuộc tính quyết định (Output/Label)
- Một BQĐ đơn giản:

Bệnh nhân	Sốt (S)	Ho (H)	Bệnh (B)
u_1	cao	có	có
u_2	thấp	không	không
u_3	cao	không	có

$$\leadsto U = \{u_1, u_2, u_3\}, C = \{S, H\}, d = B.$$

- **Bài toán giảm chiều DL** (BQĐ): Tìm tập con $R \subseteq C$ sao cho khả năng phân lớp của R tương đương với C .

Giảm chiều bằng tập thô: Các khái niệm-3

- *Quan hệ không phân biệt: $B \subseteq C$*

$$IND(B) = \{(u, v) \in U \times U \mid \forall a \in B, a(u) = a(v)\}.$$

- 2 ĐT có cùng giá trị trên $B \Rightarrow$ chúng được gọi thuộc cùng 1 ***hạt tri thức*** (Knowledge Granules):

$$U/IND(B) = \{[u]_B : u \in U\} \stackrel{hay}{=} U/B$$

Giảm chiều bằng tập thô: Ví dụ-1

Bệnh nhân	Sốt (S)	Ho (H)	Bệnh (B)
u_1	cao	có	có
u_2	thấp	không	không
u_3	cao	không	có

- ↪ Các hạt được tính qua *IND*:

- $U / \{S, H\} = \{\{u_1\}, \{u_2\}, \{u_3\}\}$
- $U / \{S\} = \{\{u_1, u_3\}, \{u_2\}\}$
- $U / \{H\} = \{\{u_2, u_3\}, \{u_1\}\}$

Tập thuộc tính	Các hạt trị thức (Lớp tương đương)	Trộn lẫn nhãn?	Khả năng giảm chiều
$\{S, H\}$	$\{u_1\}, \{u_2\}, \{u_3\}$	Không	Tập gốc (chưa giảm)
$\{S\}$	$\{u_1, u_3\}, \{u_2\}$	Không	Được (H là dư thừa)
$\{H\}$	$\{u_1\}, \{u_2, u_3\}$	Có	Không (S là Core)

Giảm chiều bằng tập thô: Các khái niệm-4

- Xét tập DL mục tiêu $X \subseteq U$ và $B \subseteq C$.

① **Xấp xỉ dưới** (Lower Approximation):

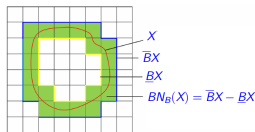
$$\underline{B}X = \{u \in U : [u]_B \subseteq X\}$$

② **Xấp xỉ trên** (Upper Approximation):

$$\overline{B}X = \{u \in U : [u]_B \cap X \neq \emptyset\}$$

③ **Vùng biên** (Boundary Region):

$$BN_B(X) = \overline{B}X - \underline{B}X$$



Giảm chiều bằng tập thô: Các khái niệm-4

- Ý nghĩa:

- ① $\underline{B}X$: tập các DT chắc chắn $\in X$ dựa trên tri thức B .
- ② $\overline{B}X$: tập các DT có khả năng $\in X$.
- ③ $BN_B(X)$: chứa các DT không thể phân loại rõ ràng là $\in X$ hay $\notin X$.

- Mệnh đề:

- ① $\underline{B}X \subseteq X \subseteq \overline{B}X$
- ② $\underline{B}\emptyset = \overline{B}\emptyset = \emptyset, \underline{B}U = \overline{B}U = U$
- ③ $X_1 \subseteq X_2 \Rightarrow \underline{B}X_1 \subseteq \underline{B}X_2, \overline{B}X_1 \subseteq \overline{B}X_2$
- ④ $\underline{B}(X_1 \cap X_2) = \underline{B}X_1 \cap \underline{B}X_2, \overline{B}(X_1 \cup X_2) = \overline{B}X_1 \cup \overline{B}X_2$
- ⑤ $\underline{B}(X_1 \cup X_2) \supseteq \underline{B}X_1 \cup \underline{B}X_2, \overline{B}(X_1 \cap X_2) \subseteq \overline{B}X_1 \cap \overline{B}X_2$
- ⑥ $\underline{B}(U - X) = U - \overline{B}X, \overline{B}(U - X) = U - \underline{B}X$
- ⑦ $\underline{B}(\underline{B}X) = \overline{B}(\underline{B}X) = \underline{B}X, \underline{B}(\overline{B}X) = \overline{B}(\overline{B}X) = \overline{B}X$

- X là **tập rõ (chính xác, tinh)**: $\underline{B}X = \overline{B}X$
 - Vùng biên rỗng. DL X cực kỳ rõ ràng.
- X là **tập thô**: $\underline{B}X \neq \overline{B}X$
 - Vùng biên khác rỗng. DL X có sự *nhập nhằng/nhiều*.

Giảm chiều bằng tập thô: Ví dụ-4

Bệnh nhân	Sốt (S)	Ho (H)	Bệnh (B)
u_1	cao	có	<i>có</i>
u_2	thấp	không	<i>không</i>
u_3	cao	không	<i>có</i>

- Xét $X = \{u_1, u_3\}$ (nhóm "có bệnh").

① Dựa trên toàn bộ $C = \{S, H\}$:

- $U/\{C\} = \{\{u_1\}, \{u_2\}, \{u_3\}\}$. Mỗi bệnh nhân là 1 hạt riêng biệt (không ai giống hệt ai về cả **Sốt** và **Ho**).
- $\underline{CX} = \{u_1, u_3\}$, $\overline{CX} = \{u_1, u_3\}$, $BN_C(X) = \emptyset$.
- $\Rightarrow X$: tập rõ (đối với C), DL X được phân loại chính xác 100%.

② Xét dựa trên $B = \{H\}$ (**Ho**):

- $U/\{B\} = \{\{u_1\}, \{u_2, u_3\}\}$
- $\underline{BX} = \{u_1\}$, $\overline{BX} = \{u_1, u_2, u_3\}$, $BN_B(X) = \{u_2, u_3\}$.
- $\Rightarrow X$: tập thô (đối với B).
- $\Rightarrow$ Nếu chỉ dùng **Ho** $\Rightarrow$ bị mơ hồ ở u_2, u_3 .

Giảm chiều bằng tập thô: Các khái niệm-5

- Xét BQĐ $DT = (U, C \cup \{d\})$ với $B \subseteq C$,
 $U/d = \{D_1, D_2, \dots, D_m\}$ (lớp).
- Miền dương** (Positive, khẳng định) của d đối với B :

$$POS_B(\{d\}) = \bigcup_{D_i \in U/\{d\}} \underline{B}(D_i)$$

- Độ** (bậc) phụ thuộc của d vào B :

$$\gamma_B(\{d\}) = \frac{|POS_B(\{d\})|}{|U|}$$

~

- $0 \leq \gamma_B(\{d\}) \leq 1$.
- $\gamma = 1 \Rightarrow d$ phụ thuộc hoàn toàn vào B .

Giảm chiều bằng tập thô: Các khái niệm-6

- *Thuộc tính dư thừa a trong C đối với d:*

$$POS_C(\{d\}) = POS_{C-\{a\}}(\{d\})$$

hoặc

$$\gamma_C(\{d\}) = \gamma_{C-\{a\}}(\{d\})$$

- $\leadsto$ Ngược lại a được gọi **thuộc tính cần thiết**.
- **Rút gọn** (tương đối) **của C đối với d**: tập con $R \subset C$ thoả mãn
 - ① *Tính bảo toàn*: $POS_R(\{d\}) = POS_C(\{d\})$
 - ② *Tính tối thiểu*: $\forall a \in R: POS_{R-\{a\}}(\{d\}) \neq POS_C(\{d\})$.
- *Reduct* (C): tất cả rút gọn của C.

Giảm chiều bằng tập thô: Các khái niệm-7

- **Lỗi (Core):**

$$\text{Core}(C) = \{a \in C \mid a \text{ là cần thiết trong } C \text{ đối với } d\}.$$

- **Mệnh đề:** $\text{Core}(C) = \bigcap \text{Reduct}(C).$

ĐT	Thu nhập (T)	Công việc (V)	Tài sản (S)	Quyết định (d)
u_1	cao	ổn định	có	<i>chấp nhận</i>
u_2	cao	ổn định	không	<i>chấp nhận</i>
u_3	thấp	không	có	<i>từ chối</i>
u_4	thấp	không	không	<i>từ chối</i>

- Xét tập $C = \{T, V, S\}$: $U/C = \{\{u_1\}, \{u_2\}, \{u_3\}, \{u_4\}\}$

- $U/\{d\} = \{\{u_1, u_2\}, \{u_3, u_4\}\} = \{d_{cn}, d_{tc}\}$
- $\underline{C}(d_{cn}) = \{u_1\} \cup \{u_2\} = d_{cn}$, $\underline{C}(d_{tc}) = \{u_3\} \cup \{u_4\} = d_{tc}$
- $\Rightarrow \text{POS}_C(\{d\}) = d_{cn} \cup d_{tc} = U$

Giảm chiều bằng tập thô: ví dụ-5

- Xét tập $B = \{T\}$ (Thu nhập):
 - $U/T = \{\{u_1, u_2\}, \{u_3, u_4\}\}$
 - $\underline{T}(d_{cn}) = \{u_1, u_2\}, \underline{T}(d_{tc}) = \{u_3, u_4\}$
 - $\Rightarrow \mathbf{POS}_{\{T\}}(\{d\}) = \mathbf{U}$, và $\{T\}$ cũng **tối thiểu**.
 - $\Rightarrow \{T\} \in \mathbf{Reduct}(C)$.
 - $\gamma_{\{T\}}(\{d\}) = \frac{|\mathbf{POS}_{\{T\}}(\{d\})|}{|U|} = \frac{4}{4} = 1$
- Xét $B = \{S\}$ (Tài sản):
 - $U/S = \{\{u_1, u_3\}, \{u_2, u_4\}\}$
 - $\underline{S}(d_{cn}) = \emptyset, \underline{S}(d_{tc}) = \emptyset$
 - $\Rightarrow \mathbf{POS}_{\{S\}}(\{d\}) = \emptyset$
 - $\gamma_{\{S\}}(\{d\}) = \frac{|\mathbf{POS}_{\{S\}}(\{d\})|}{|U|} = 0$
 - $\Rightarrow \{S\} \notin \mathbf{Reduct}(C)$.
 - $\Rightarrow$ Thuộc tính **Tài sản** không cung cấp bất kỳ thông tin nào để có thể đưa ra quyết định.

Giảm chiều bằng tập thô: Thuật toán Skowron-1

- $\hookrightarrow$ BQĐ $DT = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$.
- ***Ma trận phân biệt*** (Discernibility matrix) của DT :

$$M = (c_{ij})_{n \times n}$$

$$c_{ij} = \begin{cases} \{a \in C \mid a(u_i) \neq a(u_j)\}, & \text{nếu } d(u_i) \neq d(u_j) \\ \emptyset, & \text{nếu } d(u_i) = d(u_j) \end{cases}$$

- **Hàm phân biệt:** hàm Boolean $f(f_{DT})$

$$f = \bigwedge \left\{ \bigvee c_{ij} : 1 \leq j < i \leq n, c_{ij} \neq \emptyset \right\}$$

Giảm chiều bằng tập thô: Thuật toán Skowron-2

Thuật toán *Reduct* (Skowron, 1992)

- **Đầu vào:** BQĐ $DT = (U, C \cup \{d\})$.
- **Đầu ra:** $Reduct(C)$.
- **Phương pháp:**
- **Bước 1.** *Xây dựng MT phân biệt*
 - Duyệt qua tất cả các cặp ĐT khác nhau.
- **Bước 2.** *Lập hàm phân biệt*
 - Chuyển đổi MT thành biểu thức logic.
- **Bước 3.** *Tối giản biểu thức*
 - Sử dụng các luật đại số Boolean: luật hấp thụ, luật phân phối.
- **Bước 4.** *Kết luận*
 - Mỗi hội sơ cấp trong biểu thức tối giản là 1 tập rút gọn.

Giảm chiều bằng tập thô: Thuật toán Skowron-3

Bệnh nhân	Sốt (S)	Ho (H)	Bệnh (B)
u_1	cao	có	<i>có</i>
u_2	thấp	không	<i>không</i>
u_3	cao	không	<i>có</i>

• $\hookrightarrow$

$$M = \begin{pmatrix} \emptyset & \{S, H\} & \emptyset \\ \{S, H\} & \emptyset & \{S\} \\ \emptyset & \{S\} & \emptyset \end{pmatrix}$$

• $\hookrightarrow$

- Lập hàm phân biệt: $f = (S \vee H) \wedge S$
- Tối giản: $f = S$ (theo luật hấp thụ)
- $\implies \text{Reduct} = \{\{S\}\}, \text{Core} = \{S\}.$

Giảm chiều bằng tập thô: Thuật toán Skowron-4

ĐT	Thu nhập (<i>T</i>)	Công việc (<i>V</i>)	Tài sản (<i>S</i>)	Quyết định (<i>d</i>)
u_1	cao	ổn định	có	<i>chấp nhận</i>
u_2	cao	ổn định	không	<i>chấp nhận</i>
u_3	thấp	không	có	<i>từ chối</i>
u_4	thấp	không	không	<i>từ chối</i>

• $\hookrightarrow$

$$M = \begin{pmatrix} \emptyset & \emptyset & \{T, V\} & \{T, V, S\} \\ \emptyset & \emptyset & \{T, V, S\} & \{T, V\} \\ \{T, V\} & \{T, V, S\} & \emptyset & \emptyset \\ \{T, V, S\} & \{T, V\} & \emptyset & \emptyset \end{pmatrix}$$

• $\hookrightarrow$

- Lập hàm phân biệt:

$$f = (T \vee V) \wedge (T \vee V \vee S) \wedge (T \vee V \vee S) \wedge (T \vee V).$$

Giảm chiều bằng tập thô: Thuật toán Skowron-5

- $\hookrightarrow$
 - $f = (T \vee V) \wedge (T \vee V \vee S) \wedge (T \vee V \vee S) \wedge (T \vee V)$
 - Tối giản:
 - Hấp thu: $(T \vee V) \wedge (T \vee V \vee S) = T \vee V$
 - Hội với các thành phần còn lại:
 $(T \vee V) \wedge (T \vee V) = (T \vee V)$
 - $\Rightarrow f = (T \vee V)$.
- $\Longrightarrow \text{Reduct} = \{\{T\}, \{V\}\}, \text{Core} = \emptyset$.

Giảm chiều bằng tập thô: Thuật toán Shen-1

Thuật toán QuickReduct (*Shen & Chouchoulas, 2000*)

- **Đầu vào:** BQĐ $DT = (U, C \cup \{d\})$.
- **Đầu ra:** $R \in \text{Reduct}(C)$.
- **Phương pháp:**
- **Bước 1.** (*Khởi tạo*) Đặt $R := \emptyset$.
- **Bước 2.** (*Tính độ phụ thuộc gốc*) Tính $\gamma_C(\{d\})$.
- **Bước 3.** (*Lặp*) Chừng nào $\gamma_R(\{d\}) < \gamma_C(\{d\})$:
 - ① Tính $\gamma_{R \cup \{a\}}(d)$ với mỗi $a \in (C - R)$.
 - ② Chọn a sao cho $\gamma_{R \cup \{a\}}(\{d\}) \rightarrow \max$.
 - ③ Cập nhật $R := R \cup \{a\}$.
- **Bước 4.** (*Kết thúc*) R sau cùng là tập rút gọn cần tìm.

Giảm chiều bằng tập thô: Thuật toán Shen-2

ĐT	Thu nhập (T)	Công việc (V)	Tài sản (S)	Quyết định (d)
u_1	cao	ổn định	có	<i>chấp nhận</i>
u_2	cao	ổn định	không	<i>chấp nhận</i>
u_3	thấp	không	có	<i>từ chối</i>
u_4	thấp	không	không	<i>từ chối</i>

- $R := \emptyset$, $\gamma_C(\{d\}) = 1$.
- Xét từng thuộc tính:
 - $\gamma_{\{S\}}(\{d\}) = \frac{|POS_{\{S\}}(\{d\})|}{|U|} = \frac{0}{4} = 0$
 - $\gamma_{\{T\}}(\{d\}) = \frac{|POS_{\{T\}}(\{d\})|}{|U|} = \frac{4}{4} = 1$
 - $\gamma_{\{V\}}(\{d\}) = 1$
 - Vì $\gamma_{\{T\}}(\{d\}) = \gamma_C(\{d\}) \Rightarrow$ chọn $R = \{T\}$. TT dừng.
- Có thể thấy: $Reduct(C) = \{\{T\}, \{V\}\}$.

Giảm chiều bằng tập thô: Thuật toán Jensen-1

- **Entropy của tập quyết định $\{d\}$:**

$$H(\{d\}) = - \sum_{i=1}^m p(d_i) \log_2 p(d_i)$$

- **Entropy điều kiện của $\{d\}$ khi biết tập thuộc tính B :**

$$H(\{d\} | B) = \sum_{j=1}^k p(b_j) H(\{d\} | B = b_j)$$

- **Thông tin tương hỗ (Mutual Information):**

$$I(B; \{d\}) = H(\{d\}) - H(\{d\} | B)$$

Giảm chiều bằng tập thô: Thuật toán Jensen-2



- ① $I(B; \{d\}) = IG(\{d\}, B)$: tùy ngữ cảnh để sử dụng
- ② $\text{dom}(d) = \{d_1, d_2, \dots, d_m\}$ (nhãn).
- ③ b_j : các LTD (equivalence classes) được tạo ra bởi B .
- ④ Có thể hiểu $H(\{d\} | B)$ như là TB có trọng số: "*Sự hỗn loạn chung của HT bằng tổng sự hỗn loạn của từng nhóm nhỏ, sau khi đã nhân với tỉ lệ số lượng của nhóm đó.*"
- ⑤ $H(\{d\} | B) = 0$: B xác định hoàn toàn quyết định $\{d\}$ (không còn sự không chắc chắn).

Giảm chiều bằng tập thô: Thuật toán Jensen-3

Thuật toán Entropy-based QuickReduct (EBQR- Jensen & Shen, 2002- Stefanowski & Słowiński, 1994)

- **Đầu vào:** BQĐ $DT = (U, C \cup \{d\})$.
- **Đầu ra:** $R \in \text{Reduct}(C)$.
- **Phương pháp:**
- **Bước 1.** (Khởi tạo) Đặt $R := \emptyset$.
- **Bước 2.** (Tính Entropy gốc) Tính $H(\{d\} | C)$.
- **Bước 3.** (Lặp) Chừng nào $H(\{d\} | R) > H(\{d\} | C)$:
 - ① Tính $H(\{d\} | R \cup \{a\})$ với mỗi $a \in (C - R)$.
 - ② Chọn a sao cho $H(\{d\} | R \cup \{a\}) \rightarrow \min$.
 - ③ Cập nhật $R := R \cup \{a\}$.
- **Bước 4.** (Kết thúc) R sau cùng là tập rút gọn cần tìm.

Giảm chiều bằng tập thô: Thuật toán Jensen-4

Ví dụ:

ĐT	Thu nhập (T)	Công việc (V)	Tài sản (S)	Quyết định (d)
u_1	cao	ổn định	có	<i>chấp nhận</i>
u_2	cao	ổn định	không	<i>chấp nhận</i>
u_3	thấp	không	có	<i>từ chối</i>
u_4	thấp	không	không	<i>từ chối</i>

- $R := \emptyset$

$$\begin{aligned}
 H(\{d\} | C) &= \sum_{i=j}^4 p(c_j) H(\{d\} | C = c_j) \\
 &= \frac{1}{4}(0) + \frac{1}{4}(0) + \frac{1}{4}(0) + \frac{1}{4}(0) = 0
 \end{aligned}$$

Giảm chiều bằng tập thô: Thuật toán Jensen-5

Ví dụ: (tiếp theo)

- $p(d = \text{chapnhan}) = 2/4, p(d = \text{tuchoi}) = 2/4$

$\Rightarrow$

$$H(\{d\} | \emptyset) = H(d) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

- Xét S :

- $S = \text{co}$: có 2 ĐT $\{u_1$ ("chấp nhận"), u_3 ("từ chối") $\} \Rightarrow$

$$p(d = \text{chapnhan}) = 1/2, p(d = \text{tuchoi}) = 1/2 \Rightarrow$$

$$H(\{d\} | S = \text{co}) = 1$$

- $S = \text{khong}$: $H(\{d\} | S = \text{khong}) = 1$

$\Rightarrow$

$$\begin{aligned} H(\{d\} | S) &= p(S = \text{co})H(\{d\} | S = \text{co}) + \\ &\quad + p(S = \text{khong})H(\{d\} | S = \text{khong}) \\ &= ((2/4) \times 1) + ((2/4) \times 1) = 1 \end{aligned}$$

Giảm chiều bằng tập thô: Thuật toán Jensen-6

Ví dụ: (tiếp theo)

- Xét T :
 - $T = \text{cao}$: có 2 ĐT chấp nhận $\{u_1, u_2\} \Rightarrow$
 $p(d_{cn}) = 2/2 = 1, p(d_{tc}) = 0/2 = 0$
 $\Rightarrow H(\{d\} | T = \text{cao}) = 0$
 - $T = \text{thap}$: $H(\{d\} | T = \text{thap}) = 0$ $\Rightarrow H(\{d\} | T) = 0$ (không nhiễu)
- Xét V :
 - $V = \text{ondinh} \Rightarrow d = \text{chapnhan}$ (100%)
 - $V = \text{tuchoi} \Rightarrow d = \text{tuchoi}$ (100%) $\Rightarrow H(\{d\} | V) = 0$
- Chọn $T \Rightarrow R := \{T\}$. TT dừng.
- $\leadsto \text{Reduct}(C) = \{\{T\}, \{V\}\}$.

Giảm chiều bằng tập thô: Thuật toán Jensen-7

Ví dụ:

ĐT	Thu nhập (T)	Công việc (V)	Tài sản (S)	Quyết định (d)
u_1	cao	ổn định	có	<i>chấp nhận</i>
u_2	cao	ổn định	không	<i>từ chối</i>
u_3	thấp	không	có	<i>từ chối</i>
u_4	thấp	không	không	<i>từ chối</i>

- $R := \emptyset$
- Bảng vẫn nhất quán: $\Rightarrow$ Entropy mục tiêu
 $H(\{d\} | C) = 0$
- Entropy hiện tại $H(\{d\} | R)$ khi $R = \emptyset$:
 $H(d) = -\left(\frac{1}{4} \log_2 \left(\frac{1}{4}\right) + \frac{3}{4} \log_2 \left(\frac{3}{4}\right)\right) \simeq 0.81$

Giảm chiều bằng tập thô: Thuật toán Jensen-7

Ví dụ: (tiếp theo)

Lập 1:

- Xét S :
 - $S = co: \{u_1 (cn), u_3 (tc)\} \Rightarrow H(\{d\} | S = co) > 0 (= 1)$
 - $S = khong: \{u_2(tc), u_4(tc)\} \Rightarrow H(\{d\} | S = khong) = 0$
 $\Rightarrow$ Entropy: $H(\{d\} | S) = (0.5 \times 1) + (0.5 \times 0) = 0.5$
 $(> 0$ vẫn lẫn lộn)
- Xét V :
 - $V = ondingh: \{u_1 (cn), u_2 (tc)\} \Rightarrow Entropy > 0$ (0.5, vẫn lẫn lộn)
- Xét T :
 - $T = cao: \{u_1 (cn), u_2 (tc)\} \Rightarrow Entropy > 0$ (0.5, vẫn lẫn lộn)
- Không có thuộc tính nào làm Entropy về 0. GS chọn S và cập nhật $R := \{S\}$.

Giảm chiều bằng tập thô: Thuật toán Jensen-8

Ví dụ: (tiếp theo)

Lặp 2:

- Xét $T: \Rightarrow R = \{T, S\}$
 - $(T = \text{cao}, S = \text{co}): \{u_1(\text{cn})\} \Rightarrow \text{Entropy} = 0$
 - $(T = \text{cao}, S = \text{khong}): \{u_2(\text{tc})\} \Rightarrow \text{Entropy} = 0$
 - $(T = \text{thap}, S = \text{co}): \{u_3(\text{tc})\} \Rightarrow \text{Entropy} = 0$
 - $(T = \text{thap}, S = \text{khong}): \{u_4(\text{tc})\} \Rightarrow \text{Entropy} = 0$
 - $\Rightarrow H(\{d\} | \{T, S\}) = 0$. TT dừng,
- $\Rightarrow R = \{T, S\} \in \text{Reduct}(C)$.
- $\leadsto \text{Reduct}(C) = \{\{T, S\}, \{V, S\}\}$.

Nội dung môn học

- ① Tổng quan về học máy
- ② Toán, Python và Weka
- ③ Học có giám sát
 - Hàm mất mát
 - k -láng giềng gần nhất
 - Hồi quy tuyến tính
 - Cây quyết định
 - Bộ phân loại Naïve Bayes
 - Máy véc-tơ hỗ trợ
- ④ Học không giám sát
 - Phân cụm k -mean
 - Giảm chiều bằng tập thô
- ⑤ Học kết hợp
- ⑥ Tài liệu tham khảo

Chương 5. Học kết hợp

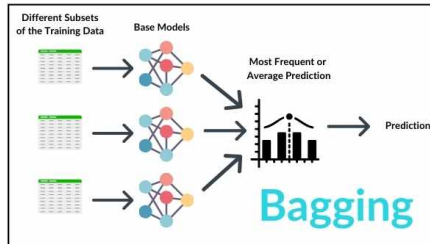
- 1 Phương pháp Bagging
- 2 Phương pháp Boosting
- 3 Phương pháp Voting
- 4 Phương pháp MultiClassClassifier



- $\leadsto$ Tự đọc: PP Voting, MultiClassClassifier

Phương pháp Bagging: Giới thiệu

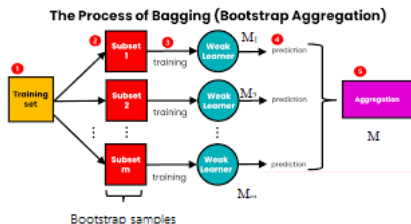
- Kết hợp nhiều MHHM (yếu) được HL trên các TDL khác nhau $\Rightarrow$ tạo ra 1 MHHM mạnh hơn.



- **Bagging** thường được áp dụng cho các MHHM có bias thấp và PS cao $\Rightarrow$ giảm PS và tăng độ ổn định của MH.
- MH phổ biến: **Random Forest**.

Phương pháp Bagging: Cách hoạt động

- ① THL $\Rightarrow$ tạo các TDL con (*Bootstrap*).
- ② TDL con $\Rightarrow$ HL các MHHM (thường CQĐ).
- ③ DD $\hat{y}$:
 - ① Bài toán hồi quy: TBC
 - ② Bài toán phân loại: bầu chọn đa số



Phương pháp Bagging: Ứng dụng

- **Y tế:**
 - Chẩn đoán bệnh, Dự đoán kết quả điều trị.
- **Tài chính:**
 - Dự đoán giá cổ phiếu, rủi ro tài chính.
- **Thị giác máy tính:**
 - Nhận diện hình ảnh, phân loại ảnh.
- **NTP:**
 - Phân loại văn bản, phát hiện spam.

Phương pháp Bagging: Ưu nhược điểm

- **Ưu điểm:**
 - ① **Giảm PS của MH:** Kết hợp nhiều MH $\Rightarrow$ giảm độ biến động của DĐ.
 - ② **Giảm overfitting:** Kết hợp nhiều MH $\Rightarrow$ giảm nguy cơ overfitting so với 1 MH đơn lẻ HL trên 1 TDL duy nhất.
 - ③ **Tăng độ chính xác của MH.**
- **Nhược điểm:**
 - ① **Tính toán phức tạp:** HL và DĐ với nhiều MH $\Rightarrow$ yêu cầu tài nguyên tính toán lớn hơn.
 - ② **Giải thích khó khăn:** Các MH kết hợp thường phức tạp hơn và khó giải thích hơn.

Phương pháp Bagging: Thực hành Weka - (1)

- Các bước tiến hành:

- ➊ Tải vào bộ nhớ **train set**
- ➋ Trong Classify lựa chọn Classifier là meta và chọn **Bagging**
 - Lựa chọn 1 classifier cho Bagging: J48, Naive Bayes ...
- ➌ Thiết lập chế độ HL:
 - 10-fold cross validation.
 - Các lựa chọn khác.
- ➍ Kiểm thử MH với **test set**
- ➎ Lưu MH.

Phương pháp Bagging: Thực hành Weka - (2)

- Thực hành trên tập *DL labor*:

① *labor_train.arff*

② *labor_test.arff*

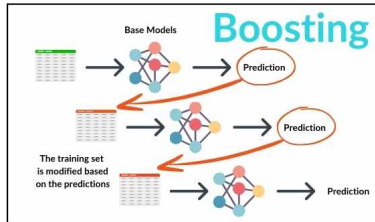
The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier is set to 'Bagging -P 100 -S 1 -num-slots 1 -I 100 -W weka.classifiers.bayes.NaiveBayes'. The test options are set to 'Cross-validation' with 'Folds' set to 10 and 'Percentage split' set to 66. The classifier output shows the following results:

weka.classifiers.bayes.NaiveBayes		
Time taken to build model: 0 seconds		
--- Stratified cross-validation ---		
--- Summary ---		
Correctly Classified Instances	38	97.4359 %
Incorrectly Classified Instances	1	2.5641 %
Kappa statistic	0.9349	
Mean absolute error	0.1139	
Root mean squared error	0.198	
Relative absolute error	27.6835 %	
Root relative squared error	43.8502 %	
Total Number of Instances	39	
--- Detailed Accuracy By Class ---		

The 'weka.gui.GenericObjectEditor' window is also open, showing the configuration for 'weka.classifiers.meta.Bagging'. The 'Class for bagging a classifier to reduce variance.' is set to 'NaiveBayes'. Other settings include 'bagSizePercent' (100), 'batchSize' (100), 'calcOutOfBag' (False), 'classifier' (NaiveBayes), 'debug' (False), 'doNotCheckCapabilities' (False), 'numDecimalPlaces' (2), 'numExecutionSlots' (1), and 'numIterations' (100).

Phương pháp Boosting: Giới thiệu

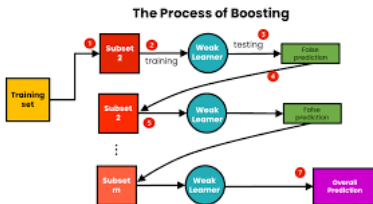
- Kết hợp nhiều MH yếu thành 1 MH mạnh, bằng cách xây dựng dãy các MH liên tiếp, trong đó mỗi MH sau tập trung sửa lỗi của MH trước đó.



- **Boosting** thường áp dụng cho các MH có PS thấp và bias cao $\Rightarrow$ giảm bias.
- PP HKH mạnh.

Phương pháp Boosting: Cách hoạt động

- ① TDL gốc $\Rightarrow$ HLMH đầu tiên, tính toán lỗi của MH này.
- ② Tăng trọng số cho các mẫu bị DĐ sai để MH tiếp theo tập trung vào chúng.
- ③ HLMH tiếp theo. Quá trình này lặp lại cho đến số lần mong muốn.
- ④ Kết hợp các MH bằng cách gán trọng số cho mỗi MH dựa trên độ chính xác của nó.



Phương pháp Boosting: Các thuật toán

- 1 **AdaBoost** (Adaptive Boosting)
- 2 **Gradient Boosting**
- 3 **XGBoost** (Extreme Gradient Boosting): cải tiến Gradient Boosting

Phương pháp Boosting: Ứng dụng

- **Y tế:**
 - Chẩn đoán bệnh từ DL bệnh án.
- **Tài chính:**
 - Dự báo tài chính.
 - Phát hiện gian lận .
- **Thị giác máy tính:**
 - Nhận dạng ảnh.

Phương pháp Boosting: Ưu nhược điểm

- **Ưu điểm:**
 - ① **Hiệu quả:** trên nhiều loại DL khác nhau.
 - ② **Tăng độ chính xác MH.**
 - ③ **XLDL:** mất cân bằng tốt hơn.
- **Nhược điểm:**
 - ① **Tính toán phức tạp hơn.**
 - ② **Dễ bị overfitting:** nếu số lượng vòng lặp quá lớn.

Phương pháp Boosting: Thực hành Weka - (1)

- Các bước tiến hành:

- ➊ Tải vào bộ nhớ **train set**
- ➋ Trong Classify lựa chọn Classifier là meta và chọn **AdaBoostM1**
 - Lựa chọn 1 classifier cho Bagging: J48, Hồi quy logistic
 - ...
- ➌ Thiết lập chế độ HL:
 - 10-fold cross validation.
 - Các lựa chọn khác.
- ➍ Kiểm thử MH với **test set**
- ➎ Lưu MH.

Phương pháp Boosting: Thực hành Weka - (2)

- Thực hành trên tập DL *labor*:
 - ① *labor_train.arff*
 - ② *labor_test.arff*

Nội dung môn học

- ① Tổng quan về học máy
- ② Toán, Python và Weka
- ③ Học có giám sát
 - Hàm mất mát
 - k -láng giềng gần nhất
 - Hồi quy tuyến tính
 - Cây quyết định
 - Bộ phân loại Naïve Bayes
 - Máy véc-tơ hỗ trợ
- ④ Học không giám sát
 - Phân cụm k -mean
 - Giảm chiều bằng tập thô
- ⑤ Học kết hợp
- ⑥ Tài liệu tham khảo

Tài liệu tham khảo

- ① Marc Peter Deisenroth, *Mathematics For Machine Learning*, Cambridge University Press, 2020.
- ② Vũ Hữu Tiệp, *Machine Learning cơ bản*, 2020.
- ③ Nguyễn Hoàng Sơn, *Giáo trình CSDL*, NXB ĐH Huế, 2018 (Download: 1 2).

