Trường Đại học Quy Nhơn
Khoa Công nghệ thông tin
Bộ môn Hệ thống thông tin

Trần Thị Liên
tranthilien@qnu.edu.vn

CHƯƠNG 2. THU THẬP &TIỀN XỬ LÝ DỮ LIỆU

Nội dung

2.1. Tổng quan về giai đoạn tiền xử lý dữ liệu

Tính quan trọng của tiền xử lý

- Không có dữ liệu tốt, không thể có kết quả khai phá tốt!
  - Quyết định chất lượng phải dựa trên dữ liệu chất lượng
    - Chẳng hạn, dữ liệu bội hay thiếu là nguyên nhân không chính xác, thậm chí gây hiểu nhầm.
  - Kho dữ liệu cần tích hợp nhất quán
- Phần lớn công việc xây dựng một kho dữ liệu là trích chọn, làm sạch và chuyển đổi dữ liệu —Bill Inmon .
- Dữ liệu có chất lượng cao nếu như phù hợp với mục đích sử dụng trong điều hành, ra quyết định, và lập kế hoạch

2.1. Tổng quan về giai đoạn tiền xử lý dữ liệu

Dữ liệu trong thực tế có chất lượng xấu

Dữ liệu thiếu, không đầy đủ: thiếu giá trị của thuộc tính, thiếu các thuộc tính quan tâm, hoặc chỉ chứa dữ liệu tích hợp

VD tuổi, cân nặng=“”

Dữ liệu bị tạp, nhiễu: chứa lỗi hoặc các sai biệt

VD : Lương =“-100 000”

- **Dữ liệu mâu thuẫn**: có sự không thống nhất
  - Vd salary = “abc” (không phù hợp với kiểu dữ liệu số của thuộc tính salary)

2.1. Tổng quan về giai đoạn tiền xử lý dữ liệu

- **Bài tập** : 20': thảo luận và tổng hợp
  - Tình huống: Bạn là người quản lý thông tin, bạn có công ty điện tử X (gồm rất nhiều chi nhánh trên toàn quốc). Bạn cần phân tích Dữ liệu bán hàng của tất cả các chi nhánh.
  - Sau khi thu thập DL từ các chi nhánh, bạn có thể gặp những vấn đề gì ? Ví dụ và tại sao?
  - *Tại sao DL trong thực tế thường có chất lượng xấu?*

2.1. Tổng quan về giai đoạn tiền xử lý dữ liệu

Giai đoạn tiền xử lý dữ liệu

- Quá trình xử lý dữ liệu thô/gốc nhằm cải thiện chất lượng dữ liệu và do đó, cải thiện chất lượng của kết quả khai phá.
  - Dữ liệu thô/gốc
    - Có cấu trúc, bán cấu trúc, phi cấu trúc
    - Được đưa vào từ các nguồn dữ liệu trong các hệ thống xử lý tập tin (file processing systems) và/hay các hệ thống cơ sở dữ liệu
  - Chất lượng dữ liệu: tính chính xác, tính hiện hành, tính toàn vẹn, tính nhất quán

2.1. Tổng quan về giai đoạn tiền xử lý dữ liệu

- Chất lượng dữ liệu (data quality)
  - Tính chính xác (accuracy): giá trị được ghi nhận đúng với giá trị thực.
  - Tính hiện hành (currency/timeliness): giá trị được ghi nhận không bị lỗi thời.
  - Tính toàn vẹn (completeness): tất cả các giá trị dành cho một biến/thuộc tính đều được ghi nhận.
  - Tính nhất quán (consistency): tất cả giá trị dữ liệu đều được biểu diễn như nhau trong tất cả các trường hợp.

2.1. Tổng quan về giai đoạn tiền xử lý dữ liệu

- **Các kỹ thuật tiền xử lý dữ liệu**
  - **Làm sạch dữ liệu** (data cleaning/cleansing): loại bỏ nhiễu, hiệu chỉnh những phần dữ liệu không nhất quán
  - **Tích hợp dữ liệu** (data integration): trộn dữ liệu từ nhiều nguồn khác nhau vào một kho dữ liệu
  - **Biến đổi dữ liệu** (data transformation): chuẩn hoá dữ liệu
  - **Rút gọn dữ liệu** (data reduction): thu giảm kích thước dữ liệu(nghĩa là giảm số phần tử) bằng kết hợp dữ liệu (data aggregation), loại bỏ các đặc điểm dư thừa(redundant features) (nghĩa là giảm số chiều/thuộc tính dữ liệu), gom cụm dữ liệu

Hiểu dữ liệu

- Hiểu dữ liệu là quá trình phân tích, xử lý và thu thập thông tin từ dữ liệu để đưa ra những quyết định hoặc kiến nghị hữu ích. Nó bao gồm việc xem xét và trả lời các câu hỏi về dữ liệu, khám phá các quan hệ giữa các trường và tìm ra các mẫu hoặc xu hướng trong dữ liệu. Hiểu dữ liệu cũng giúp chúng ta tìm ra những điểm mạnh và điểm yếu trong dữ liệu, và đưa ra các giải pháp để cải thiện hoặc sử dụng dữ liệu một cách hiệu quả hơn.

9

Hiểu dữ liệu

- Đo độ tập trung của dữ liệu: giúp chúng ta xác định mức độ tập trung của dữ liệu trong một tập dữ liệu.
- Để đo độ tập trung dữ liệu, chúng ta có thể sử dụng các chỉ số tập trung như trung bình, trung vị và độ tập trung của chuẩn (biểu diễn bằng độ lệch chuẩn hoặc phạm vi).
- Trung bình là giá trị trung bình của tất cả các giá trị trong tập dữ liệu. Trung vị là giá trị trung tâm của tập dữ liệu, được tìm bằng cách sắp xếp các giá trị trong tập dữ liệu và tìm giá trị giữa.

10

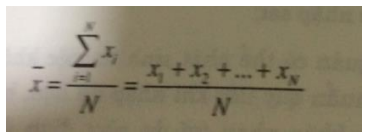
Hiểu dữ liệu

- Độ lệch chuẩn là sự khác biệt giữa các giá trị trong tập dữ liệu và giá trị trung bình. Phạm vi là khoảng cách giữa giá trị lớn nhất và giá trị nhỏ nhất trong tập dữ liệu.
- Chỉ số tập trung này cho chúng ta biết dữ liệu có tập trung chặt chẽ hay phân tán.

11

Hiểu dữ liệu

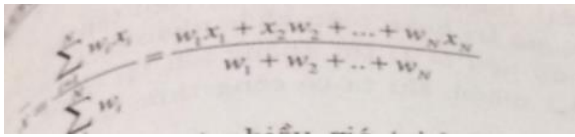
- Độ đo trung bình: gọi x là tập giá trị x=(x1, x2, ...xN) là N phần tử của dữ liệu ta có công thức tính trung bình như sau:



12

Hiểu dữ liệu

- Trường hợp có trobgj số wi thì công thức tính như sau:



- Để tính trung bình, chúng ta có thể sử dụng hàm mean() trong thư viện numpy trong python
- import numpy as np
- data = [1, 2, 3, 4, 5]
- mean_value = np.mean(data)
- print(mean_value)

13

Hiểu Dữ liệu

- Trung vị (Median) là một chỉ số thống kê cho biết giá trị giữa của một tập dữ liệu. Nó được tìm bằng cách sắp xếp các giá trị trong tập dữ liệu tăng dần hoặc giảm dần và chọn giá trị ở giữa.
- Ví dụ, nếu chúng ta có một tập dữ liệu gồm các chiều cao của một lớp học sinh, trung vị sẽ cho chúng ta biết chiều cao trung bình của lớp học sinh đó. Nếu chúng ta sắp xếp các chiều cao tăng dần, trung vị sẽ là chiều cao ở giữa của tập dữ liệu. Nếu số lượng phần tử trong tập dữ liệu là chẵn, thì trung vị sẽ là giá trị trung bình của hai giá trị giữa.

14

- Trong python sử dụng hàm median() trong thư viện numpy. Ví dụ:

import numpy as np

- data = [1, 2, 3, 4, 5]
- median_value = np.median(data)
- print(median_value)

15

Hiểu dữ liệu

- Mode: đo độ tập trung của dữ liệu đó là tập giá trị thường xuyên xuất hiện trong tập dữ liệu
- Trong python sử dụng hàm mode() trong thư viện numpy. Ví dụ:

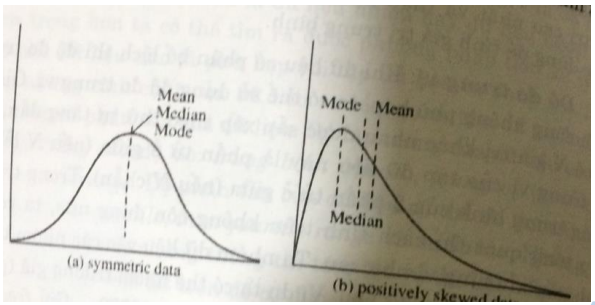
import numpy as np

- data = [1, 2, 3, 4, 5]
- mode_value = np.mode(data)
- print(mode_value)

16

Hiểu dữ liệu

- Dữ liệu phân bố đối xứng thì 3 giá trị trùng nhau. Nếu phân bố không đối xứng thì 3 giá trị khác nhau



Hiểu dữ liệu

- Đo độ phân tán dữ liệu
- Độ phân tán dữ liệu (Data Dispersion) là một chỉ số thống kê cho biết mức độ chênh lệch giữa các giá trị trong tập dữ liệu. Nó cho ta biết mức độ phân tán của dữ liệu so với giá trị trung bình.
- Ví dụ, nếu chúng ta có một tập dữ liệu gồm các lương của một nhóm nhân viên, độ phân tán dữ liệu sẽ cho ta biết mức độ chênh lệch giữa các lương trong nhóm nhân viên đó. Nếu tất cả lương trong nhóm nhân viên gần bằng nhau thì dữ liệu sẽ có độ phân tán thấp, còn nếu có một số nhân viên có lương rất cao so với các nhân viên khác thì dữ liệu sẽ có độ phân tán cao.

Hiểu dữ liệu

- Đo độ phân tán dữ liệu
- Một cách đo độ phân tán dữ liệu là sử dụng độ lệch chuẩn (Standard Deviation) hoặc phương sai (variance) , độ lệch chuẩn là sự khoảng cách giữa giá trị trung bình và các giá trị trong tập dữ liệu. Phương sai là mức độ chênh lệch giữa các giá trị trong tập dữ liệu với giá trị trung bình. Có thể sử dụng biểu đồ histogram và biểu đồ hộp để đo độ phân tán dữ liệu.
- Để vẽ biểu đồ histogram để đo độ phân tán dữ liệu, chúng ta sử dụng hàm hist() của pandas hoặc matplotlib. Để vẽ biểu đồ hộp đồ thị, chúng ta sử dụng hàm boxplot() của pandas hoặc matplotlib.
- Dùng lệnh

Hiểu dữ liệu

- Để tính độ lệch chuẩn, chúng ta có thể sử dụng hàm std() trong thư viện numpy. Ví dụ:
- import numpy as np
- data = [1, 2, 3, 4, 5]
- std_value = np.std(data)
- print(std_value)

Hiểu dữ liệu

- Tóm tắt dữ liệu: dung các dạng biểu đồ sau để thể hiện

1. Biểu đồ đường (line chart) : dùng để trình bày sự thay đổi của dữ liệu theo thời gian hoặc vị trí.
2. Biểu đồ scatter plot: dùng để trình bày sự tương quan giữa hai hoặc nhiều biến liên tục.
3. Biểu đồ histogram: dùng để trình bày sự phân bố của dữ liệu liên tục.
4. Biểu đồ box plot: dùng để trình bày sự phân bố của dữ liệu liên tục với thông tin về trung vị, phạm vi, phương sai và outliers.
5. Biểu đồ heatmap: dùng để trình bày sự tương quan giữa hai hoặc nhiều biến categorical hoặc numerical.

2.2.Làm sạch dữ liệu

Các vấn đề của dữ liệu?

- Dữ liệu thu được từ thực tế có thể chứa nhiễu, lỗi, không hoàn chỉnh, có mâu thuẫn
  - Không hoàn chỉnh (incomplete): Thiếu các giá trị thuộc tính, hoặc thiếu một số thuộc tính
    - Vd: salary = <undefined>
  - Nhiễu/lỗi (noise/error): Chứa đựng những lỗi hoặc các thể hiện bất thường (abnormal instances)
    - Vd: salary = "-525" (giá trị của thuộc tính không thể là một số âm)
  - Không nhất quán: Chứa đựng các mâu thuẫn
    - Vd: salary = "abc" (không phù hợp với kiểu dữ liệu số của thuộc tính salary)
    - Kiểu ngày: 2004/12/25 và 25/12/2004
- Giải quyết tính dư thừa tạo ra sau tích hợp dữ liệu.

Hiểu dữ liệu

- Độ lệch chuẩn (Standard Deviation) là một chỉ số thống kê cho biết mức độ phân tán của dữ liệu trong một tập dữ liệu. Nó đo lường sự khác biệt giữa các giá trị trong tập dữ liệu và giá trị trung bình.
- Ví dụ, nếu chúng ta có một tập dữ liệu gồm các điểm thi của một lớp học sinh, độ lệch chuẩn sẽ cho chúng ta biết mức độ phân tán của các điểm thi trong lớp. Nếu độ lệch chuẩn là thấp, có nghĩa là các điểm thi trong lớp gần giống nhau và tập trung gần với giá trị trung bình.

2.2. Làm sạch dữ liệu

Tại sao cần phải làm sạch dữ liệu?

- Đảm bảo chất lượng dữ liệu
- Nếu dữ liệu không sạch thì các kết quả khai phá dữ liệu sẽ bị ảnh hưởng và không đáng tin
- Các kết quả khai phá dữ liệu không chính xác sẽ dẫn đến các quyết định không chính xác, không tối ưu
  - Vd: Các dữ liệu chứa lỗi hoặc thiếu giá trị thuộc tính sẽ có thể dẫn đến các kết quả thống kê sai lầm

2.2. Làm sạch dữ liệu

- Là quá trình
  - Xác định tính không chính xác, không đầy đủ/tính bất hợp lý của dữ liệu
  - Chỉnh sửa các sai sót và thiếu sót được phát hiện
  - Nâng cao chất lượng dữ liệu.
- Quá trình bao gồm
  - Kiểm tra định dạng, tính đầy đủ, tính hợp lý, miền giới hạn
  - Xem xét dữ liệu để xác định ngoại lai (địa lý, thống kê, thời gian hay môi trường) hoặc các lỗi khác
  - Đánh giá dữ liệu của các chuyên gia miền chủ đề.
- Quá trình thường dẫn đến
  - loại bỏ, lập tài liệu và kiểm tra liên tiếp và hiệu chỉnh đúng bản ghi nghi ngờ.
  - Kiểm tra xác nhận có thể được tiến hành nhằm đạt tính phù hợp với các chuẩn áp dụng, các quy luật, và quy tắc.

2.2. Làm sạch dữ liệu

Xử lý dữ liệu bị thiếu (missing data)

- Dữ liệu không có sẵn khi cần được sử dụng
- Nguyên nhân gây ra dữ liệu bị thiếu
  - Khách quan (không tồn tại lúc được nhập liệu, sự cố,…)
  - Chủ quan (tác nhân con người)

2.2. Làm sạch dữ liệu

Xử lý dữ liệu bị thiếu (missing data)

- Giải pháp cho dữ liệu bị thiếu
  - Bỏ qua
  - Xử lý tay (không tự động, bán tự động)
  - Điền giá trị tự động:
    - Thay thế hằng số chung VD “chưa biết”, có thể thành một lớp mới trong DL
    - Trung bình giá trị thuộc tính các bản ghi hiện có
    - Trung bình giá trị thuộc tính các bản ghi cùng lớp: tinh hơn
    - Giá trị khả năng nhất: dựa trên suy luận như công thức Bayes hoặc cây quyết định
  - Ngăn chặn dữ liệu bị thiếu: thiết kế tốt CSDL và các thủ tục nhập liệu (các ràng buộc dữ liệu)

2.2. Làm sạch dữ liệu

Age	Income	Team	Gender
23	24,200	Red Sox	M
39	?	Yankees	F
45	45,390	?	F

Điền giá trị còn thiếu cho các thuộc tính
Điền giá trị trung bình cho thuộc tính income , hoặc điền giá trị mà đa số thuộc tính Income có trên cơ sở người có 39 tuổi
Điền giá trị thường xuyên xuất hiện cho thuộc tính team

2.2. Làm sạch dữ liệu

Tình huống

Thu thập dữ liệu Sinh viên của trường ĐHQN (ví dụ để phân tích mức sống)

Các thuộc tính nào có thể có trong CSDL?

Ví dụ thuộc tính bị thiếu giá trị là thuộc tính "tiền thuê nhà''

Cách giải quyết?

2.2. Làm sạch dữ liệu

Nhận diện phần tử biên (outliers) và giảm thiểu nhiễu(noisy data)

- Định nghĩa :
  - Outliers: những dữ liệu(đối tượng) không tuân theo đặc tính/hành vi chung của tập dữ liệu(đối tượng).
  - Noisy data: outliers bị loại bỏ (rejected/discarded outliers) như là những trường hợp ngoại lệ (exceptions).
- Nguyên nhân
  - Lỗi ngẫu nhiên
  - Biến dạng của một biến đo được

  Giá trị không chính xác do
  - Lỗi do thiết bị thu thập dữ liệu
  - Vấn đề nhập dữ liệu: người dùng hoặc máy có thể sai
  - Vấn đề truyền dữ liệu: sai từ thiết bị gửi/nhận/truyền
  - Hạn chế của công nghệ: ví dụ, phần mềm có thể xử lý không đúng
  - Thiếu nhất quán khi đặt tên: cũng một tên song cách viết khác nhau

2.2. Làm sạch dữ liệu

Nhận diện phần tử biên (outliers) và giảm thiểu nhiễu(noisy data)

- Giải pháp
  - Phương pháp phân khoảng (Binning):
    - Sắp dữ liệu tăng và chia "đều" vào các khoảng
    - Làm trơn: theo trung bình, theo biên…
  - Phân cụm (Clustering)
    - Phát hiện và loại bỏ ngoại lai (outliers)
  - Kết hợp kiểm tra máy tính và con người
    - Phát hiện giá trị nghi ngờ để con người kiểm tra (chẳng hạn, đối phó với ngoại lai có thể)
  - Hồi quy
    - Làm trơn: ghép dữ liệu theo các hàm hồi quy

2.2.Làm sạch dữ liệu

Phương pháp phân khoảng

- Phân chia với độ rộng (khoảng cách) bằng nhau
  - Chia khoảng giá trị thành N khoảng với kích thước (độ rộng) bằng nhau
  - Nếu min_i và max_i là giá trị nhỏ nhất và lớn nhất của thuộc tính, thì kích thước (độ rộng) của mỗi khoảng = $(max_i - min_i)/N$
  - Không phù hợp đối với các tập dữ liệu lệch (skewed data), hoặc

  có chứa các ngoại lai (outliers) – vì có thể một khoảng sẽ chỉ chứa một (hoặc một số) các ngoại lai
- Phân chia với độ sâu (tần xuất xuất hiện) bằng nhau
  - Chia khoảng giá trị thành N khoảng (không nhất thiết bằng nhau), sao cho mỗi khoảng chứa xấp xỉ bằng nhau số lượng (tần xuất xuất hiện) của các phần tử
  - Hiệu quả hơn cách phân chia với độ rộng (khoảng cách) bằng nhau

Ví dụ

* Dữ liệu được xếp theo giá: 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
* Chia thùng theo chiều sâu:
  - Bin 1: 4, 8, 9, 15
  - Bin 2: 21, 21, 24, 25
  - Bin 3: 26, 28, 29, 34
* Làm trơn thùng theo trung bình:
  - Bin 1: 9, 9, 9, 9
  - Bin 2: 23, 23, 23, 23
  - Bin 3: 29, 29, 29, 29
* Làm trơn thùng theo biên:
  - Bin 1: 4, 4, 4, 15
  - Bin 2: 21, 21, 25, 25
  - Bin 3: 26, 26, 26, 34

Bài tập phương pháp phân khoảng

- Thời gian : 10’
- Cho DL giá ($) : 5, 7, 9, 15, 24, 31, 35, 35, 37, 42, 42, 42, 48, 48, 50
- SỐ KHOẢNG : 4
- - Dùng phương pháp phân chia theo chiều sâu.
- Tính giá trị của các khoảng làm trơn theo biên
- Tính giá trị của các khoảng làm trơn theo trung bình

2.2. Làm sạch dữ liệu

- **Xử lý dữ liệu không nhất quán**
  - Định nghĩa của dữ liệu không nhất quán
    - Dữ liệu được ghi nhận khác nhau cho cùng một đối tượng/thực thể → discrepancies from inconsistent data representations
      - 2004/12/25 và 25/12/2004
    - Dữ liệu được ghi nhận không phản ánh đúng ngữ nghĩa cho các đối tượng/thực thể
      - Ràng buộc khóa ngoại

2.2. Làm sạch dữ liệu

- **Xử lý dữ liệu không nhất quán (inconsistent data)**
  - Nguyên nhân
    - Sự không nhất quán trong các qui ước đặt tên hay mã dữ liệu
    - Định dạng không nhất quán của các vùng nhập liệu
    - Thiết bị ghi nhận dữ liệu, …
  - Giải pháp
    - Tận dụng siêu dữ liệu, ràng buộc dữ liệu, sự kiểm tra của nhà phân tích dữ liệu cho việc nhận diện
    - Điều chỉnh dữ liệu không nhất quán bằng tay
    - Các giải pháp biến đổi/chuẩn hóa dữ liệu tự động

2.3. Tích hợp dữ liệu

- **Quá trình trộn dữ liệu từ các nguồn khác nhau vào một kho dữ liệu sẵn sàng cho quá trình khai phá dữ liệu**
  - Vấn đề nhận dạng thực thể (entity identification problem)
    - Tích hợp lược đồ (schema integration)
    - So trùng đối tượng (object matching)
  - Vấn đề dư thừa (redundancy)
  - Vấn đề mâu thuẫn giá trị dữ liệu (data value conflicts)
- → **Liên quan đến cấu trúc và tính không thuần nhất (heterogeneity) về ngữ nghĩa (semantics) của dữ liệu**
- → **Hỗ trợ việc giảm và tránh dư thừa và không nhất quán về dữ liệu → cải thiện tính chính xác và tốc độ quá trình khai phá dữ liệu**

2.3. Tích hợp dữ liệu

- **Vấn đề nhận dạng thực thể**
  - Các thực thể (object/entity/attribute) đến từ nhiều nguồn dữ liệu.
  - Hai hay nhiều thực thể khác nhau diễn tả cùng một thực thể thực.
  - Ví dụ ở mức lược đồ (schema): customer_id trong nguồn S1 và cust_number trong nguồn S2.
  - Ví dụ ở mức thể hiện (instance): "R & D" trong nguồn S1 và "Research & Development" trong nguồn S2. "Male" và "Female" trong nguồn S1 và "Nam" và "Nữ" trong nguồn S2.

  → Vai trò của siêu dữ liệu (metadata)

2.3. Tích hợp dữ liệu

- **Vấn đề dư thừa**
  - Hiện tượng: giá trị của một thuộc tính có thể được dẫn ra/tính từ một/nhiều thuộc tính khác, vấn đề trùng lắp dữ liệu (duplication).
  - Nguyên nhân: tổ chức dữ liệu kém, không nhất quán trong việc đặt tên chiều/thuộc tính.
  - Phát hiện dư thừa: phân tích tương quan (correlation analysis)
    - Dựa trên dữ liệu hiện có, kiểm tra khả năng dẫn ra một thuộc tính B từ thuộc tính A.
    - Đối với các thuộc tính số (numerical attributes), đánh giá tương quan giữa hai thuộc tính với các hệ số tương quan (correlation coefficient, aka Pearson's product moment coefficient).

2.3. Tích hợp dữ liệu

- **Phát hiện và xử lý các mâu thuẫn đối với giá trị dữ liệu**
  - Đối với cùng một thực thể trên thực tế, nhưng các giá trị thuộc tính từ nhiều nguồn khác nhau lại khác nhau. Các lý do có thể:
    - Các cách biểu diễn khác nhau
    - Mức đánh giá, độ đo (scales) khác nhau – Vd: hệ đo lường mét và hệ đo lường của Anh

2.4. Biến đổi dữ liệu (Data transformation)

- **Việc chuyển (ánh xạ) toàn bộ tập giá trị của một thuộc tính sang một tập mới các giá trị thay thế, sao cho mỗi giá trị cũ tương ứng với một trong các giá trị mới**
- **Các phương pháp biến đổi dữ liệu**
  - Làm trơn (Smoothing): Loại bỏ nhiễu/lỗi khỏi dữ liệu
  - Kết hợp (Aggregation): Sự tóm tắt dữ liệu, xây dựng các khối dữ liệu (data cubes)
  - Khái quát hóa (Generalization): Xây dựng các phân cấp khái niệm (concept hierarchies)
  - Chuẩn hóa (Normalization): Đưa các giá trị về một khoảng được chỉ định
    - Chuẩn hóa min-max
    - Chuẩn hóa z-score
    - Chuẩn hóa bởi thang chia 10
  - Xây dựng (tạo nên) các thuộc tính mới dựa trên các thuộc tính ban đầu

2.4. Biến đổi dữ liệu

- **Chuẩn hóa (normalization)**
  - min-max normalization
    - Giá trị cũ: v ∈[minA, maxA]
    - Giá trị mới: v' ∈ [new_minA, new_maxA]
    - →Ví dụ: chuẩn hóa điểm số từ 0-4.0 sang 0-10.0.
    - →Đặc điểm của phép chuẩn hóa min-max?

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- **→Hoặc v'= (v-min_A)/(max_A –min_A)**
- **→Ví dụ: Giả sử giá trị nhỏ nhất và lớn nhất cho thuộc tính "thu nhập bình quân" là 500.000 và 4.500.000. Chúng ta muốn ánh xạ giá trị 2.500.000 về khoảng [0.0, 1.0] sử dụng chuẩn hóa min- max. Giá trị mới thu được**

$$v' = \frac{2.500.000 - 500.000}{\quad}(1.0 - 0) + 0 = \frac{2.000.000}{\quad} = 0.5$$

4.Biến đổi dữ liệu

- **Chuẩn hóa (normalization)**
  - z-score normalization
    - Giá trị cũ: v tương ứng với mean Ā và standard deviation σ_A
    - Giá trị mới: v'
    - →Đặc điểm của chuẩn hóa z-score?

$$v' = \frac{v - \bar{A}}{\sigma_A}$$

- **Với ví dụ phía trên: Giả sử thu nhập bình quân có độ lệch tiêu chuẩn và trung bình là: 1.000.000 và 500.000. Sử dụng phương pháp z-score thì giá trị 2.500.000 được á**

$$v' = \frac{2.500.000 - 1.000.000}{500.000} = \frac{1.500.000}{500.000} = 3$$

2.4. Biến đổi dữ liệu

- **Chuẩn hóa (normalization)**
  - Normalization by decimal scaling
  - Phương pháp này sẽ di chuyển dấu phân các phần thập phân của các giá trị của thuộc tính A. Số chữ số sau dấu phân cách phần thập phân được xác định phụ thuộc vào giá trị tuyệt đối lớn nhất có thể có của thuộc tính A. Khi đó giá trị v sẽ được ánh xạ thành v' bằng cách tính
    - Giá trị cũ: v
    - Giá trị mới: v' với j là số nguyên nhỏ nhất sao cho Max(|v'|)< 1

$$v' = \frac{v}{10^j}$$

- **Ví dụ: Giả sử rằng các giá trị của thuộc tính A được ghi nhận nằm trong khoảng -968 đến 917. Giá trị tuyệt đối lớn nhất của miền là 968. Để thực hiện chuẩn hóa theo phương pháp án này, trước đó chúng ta mang các giá trị chia cho 1.000 (j = 3). Như vậy giá trị -968 sẽ chuyển thành - 0.968 và 917 được chuyển thành**

2.4. Biến đổi dữ liệu

- **Xây dựng thuộc tính/đặc tính (attribute/feature construction)**
  - Các thuộc tính mới được xây dựng và thêm vào từ tập các thuộc tính sẵn có.
  - Hỗ trợ kiểm tra tính chính xác và giúp hiểu cấu trúc của dữ liệu nhiều chiều.
  - Hỗ trợ phát hiện thông tin thiếu sót về các mối quan hệ giữa các thuộc tính dữ liệu.

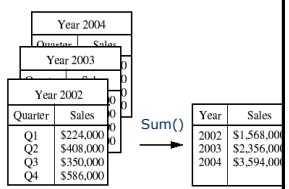
→ Các thuộc tính dẫn xuất

2.5. Thu giảm dữ liệu

- **Tập dữ liệu được biến đổi đảm bảo các toàn vẹn, nhưng nhỏ/ít hơn nhiều về số lượng so với ban đầu.**
- **Các chiến lược thu giảm**
  - Kết hợp khối dữ liệu (data cube aggregation)
  - Chọn một số thuộc tính (attribute subset selection)
  - Thu giảm chiều (dimensionality reduction)
  - Thu giảm lượng (numerosity reduction)

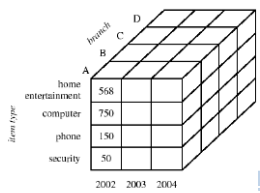
2.5. Thu giảm dữ liệu: Kết hợp khối dữ liệu

- **Mức thấp nhất của khối dữ liệu**
  - Tổng hợp dữ liệu thành một cá thể quan tâm
  - Chẳng hạn, một khách hàng trong kho dữ liệu cuộc gọi điện thoại.
  - Dạng dữ liệu: additive, semi-additive (numerical)
  - Kết hợp dữ liệu bằng các hàm nhóm: average, min, max, sum, count, …



2.5. Thu giảm dữ liệu: Kết hợp khối dữ liệu

- **Các mức phức hợp của tích hợp thành khối dữ liệu**
  - Giảm thêm kích thước dữ liệu
- **Tham khảo mức thích hợp**
  - Sử dụng trình diễn nhỏ nhất đủ để giải bài toán
- **Nên sử dụng dữ liệu khối lập phương khi trả lời câu hỏi tổng hợp thông tin**



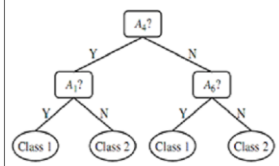
2.5.Thu giảm dữ liệu: Chọn một số thuộc tính

❖ **Chọn một số thuộc tính (attribute subset selection)**

- Phương pháp này rút gọn kích thước dữ liệu bằng cách loại bỏ các thuộc tính không hữu ích hoặc dư thừa (hoặc loại bỏ các chiều).
- Mục đích chính là tìm ra tập thuộc tính nhỏ nhất sao cho khi áp dụng các phương pháp khai phá dữ liệu thì kết quả thu được là gần sát nhất với kết quả khi sử dụng tất cả các thuộc tính.
- → Bài toán tối ưu hóa: vận dụng heuristics

2.5.Thu giảm dữ liệu

❖ **Chọn một số thuộc tính (attribute subset selection)**

Lựa chọn tăng dần	Loại bót	Cây quyết định
Tậpthuộc tính ban đầu {A1, A2, A3, A4, A5, A6} Tập rút gọn ban đầu {} => {A1} => {A1, A4} => Kết quả {A1, A4, A6}	Tậpthuộc tính ban đầu {A1, A2, A3, A4, A5, A6} => {A1, A3, A4, A5, A6} => {A1, A4, A5, A6} => Kết quả {A1, A4, A6}	Tậpthuộc tính ban đầu {A1, A2, A3, A4, A5, A6}  => Kết quả {A1, A4, A6}

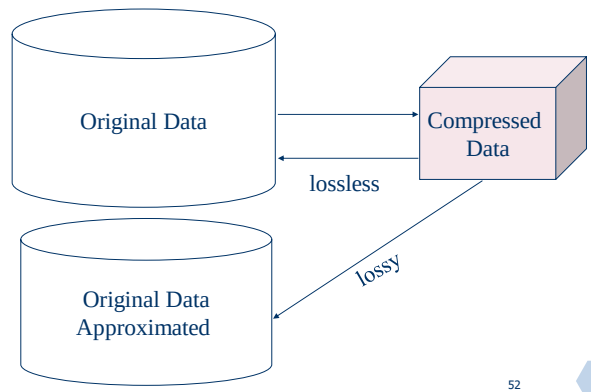
2.5. Thu giảm dữ liệu

Nén dữ liệu (Data Compression)

❖ **Mã hóa hoặc biến đổi dữ liệu**

- *Nén Không mất thông tin (lossless)*
  - *Nén có thể phục hồi lại được*
  - Nhưng chỉ các thao tác hạn hẹp mà không mở rộng
- Nén *mất thông tin (lossy)*
- Dữ liệu không phục hồi hoàn toàn
  - Dùng biến đổi wavelet, phân tích thành phần (Principal Component Analysis – PCA)

2.5. Thu giảm dữ liệu



2.5.Thu giảm dữ liệu

- **Giảm số lượng (numerosity reduction)**
- **Phương pháp tham số**
  - *Giả sử dữ liệu phù hợp với mô hình nào đó*, ước lượng tham số mô hình, lưu chỉ các tham số, và không lưu dữ liệu (ngoại trừ các ngoại lai có thể có)
  - Mô hình tuyến tính loga (Log-linear models): lấy giá trị tại một điểm trong không gian M-chiều như là tích của các không gian con thích hợp
  - Các phương pháp phi thông số (nonparametric): lưu trữ các biểu diễn thu giảm của dữ liệu
  - Biểu đồ (histograms), phân cụm (clustering), lấy mẫu (sampling)

2.5.Thu giảm dữ liệu

- **Giảm số lượng (numerosity reduction)**
  - Pp Biểu đồ
    - PP thông dụng của rút gọn DL
    - Phân chia DL vào các giỏ và chiều cao của các cột là số đối tượng nằm trong giỏ chỉ lưu giá trị trung bình của mỗi giỏ.
    - Hình dáng của biểu đồ phụ thuộc vào số lượng giỏ.

2.6.Rời rạc hóa và kiến trúc khái niệm

Rời rạc hóa

- Ba kiểu thuộc tính:
  - Định danh — giá trị từ một tập không có thứ tự
  - Thứ tự — giá trị từ một tập được sắp
  - Liên tục — số thực
- Rời rạc hóa
  - Rút gọn số lượng giá trị của thuộc tính liên tục bằng cách chia miền giá trị của thuộc tính thành các đoạn. Nhãn đoạn sau đó được dùng để thay thế giá trị thực.

2.6. Rời rạc hóa và kiến trúc khái niệm

- Phân cấp khái niệm
  - Rút gọn DL bằng tập hợp và thay thế các khái niệm mức thấp (như giá trị số của thuộc tính tuổi) bằng khái niệm ở mức cao hơn (như trẻ, trung niên, hoặc già)

2.6. Rời rạc hóa và kiến trúc khái niệm

- Ví dụ:
- Chuyển đổi gía trị logic thành 1.0
- Chuyển đổi giá trị ngày tháng thành số
- Chuyển đổi các cột có giá trị lớn thành tập các giá trị trong vùng nhỏ hơn, chẳng hạn chia chúng cho hệ số nào đó.
- Nhóm các giá trị có cùng ngữ nghĩa như: Hoạt động trước CMT8 là nhóm 1; từ 01/08/45 – 31/06/54:nhóm 2; từ 1/07/54 – 30/04/75 là nhóm 3
- Thay thế giá trị của tuổi = Trẻ, trung niên, già

2.6. Rời rạc hóa và kiến trúc khái niệm

Dữ liệu số

- Phân thùng (xem làm trơn khử nhiễu)
- Phân tích sơ đồ (đã giới thiệu)
- Phân tích cụm (đã giới thiệu)
- Rời rạc hóa dựa theo Entropy
- Phân đoạn bằng phân chia tự nhiên

Bài tập

- **Bài 1**
- **Cho dữ liệu về một thuộc tính A của một tập 12 đối tượng như sau: 4, 6, 5, 9, 8, 1, 3, 2, 7, 10, 12, 11.**
- **Số bin =3**
- **- Dùng phương pháp phân chia theo chiều sâu.**
- **Tính giá trị của các khoảng làm trơn theo biên**
- **Tính giá trị của các khoảng làm trơn theo trung bình**

Bài tập

Bài 2

Cho tập tuổi như sau:

13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

a)làm trơn dữ liệu (giảm thiểu nhiễu) sử dụng phương pháp chia khoảng theo chiều sâu với 4 bin (khoảng).

b) sử dụng chuẩn hóa min-max để chuyển giá trị tuổi 35 vào giới hạn [0-1].