**Data cleaning**

Nguồn:

Data Mining: Concepts and Techniques - Jiawei Han, Micheline Kamber, Jian Pei

**Tổng quan**

Dữ liệu trong thế giới thực thường có xu hướng không đầy đủ, nhiễu và không nhất quán. Các quy trình làm sạch dữ liệu cố gắng điền vào các giá trị bị thiếu, làm mịn nhiễu trong dữ liệu trong khi nhận diện các giá trị ngoại lệ và chỉnh sửa những sự không nhất quán trong dữ liệu.

**Các kỹ thuật chính**

I.Xử lý giá trị thiếu (Missing Values)

1.Bỏ qua bản ghi (tuple) có giá trị bị thiếu

* Phương pháp này thường được áp dụng khi nhãn lớp bị thiếu, đặc biệt nếu nhiệm vụ khai thác dữ liệu liên quan đến phân loại.
* Nhược điểm:
  + Hiệu quả không cao nếu tuple chỉ thiếu một vài thuộc tính còn lại đầy đủ thông tin hữu ích.
  + Phương pháp này đặc biệt kém khi tỉ lệ giá trị thiếu của từng thuộc tính có sự biến đổi lớn.
  + Bỏ qua tuple sẽ bỏ lỡ các giá trị hợp lệ ở các thuộc tính khác, có thể có ích cho nhiệm vụ phân tích.

2.Điền giá trị bị thiếu thủ công

* Cập nhật giá trị cho thuộc tính bị thiếu theo cách thủ công.
* Nhược điểm:
  + Tốn thời gian.
  + Không khả thi khi tập dữ liệu quá lớn và có nhiều giá trị bị thiếu.

3. Sử dụng một hằng số toàn cục để điền giá trị

* Thay thế tất cả các giá trị thiếu bằng một hằng số cố định, chẳng hạn như “Unknown” hoặc .
* Nhược điểm:
  + Khi thay thế bằng giá trị “Unknown”, chương trình khai thác dữ liệu có thể hiểu nhầm rằng giá trị này tạo thành một khái niệm quan trọng, do tất cả các giá trị đều giống nhau.
  + Dù đơn giản nhưng phương pháp này không đảm bảo độ chính xác cao.

4. Sử dụng giá trị trung tâm (trung bình hoặc trung vị) của thuộc tính

* Áp dụng các biện pháp thống kê mô tả để chọn giá trị thay thế.
  + Với phân phối dữ liệu đối xứng, giá trị trung bình **(Mean)** có thể được sử dụng.
  + Với phân phối lệch, giá trị trung vị **(Median)** sẽ là lựa chọn phù hợp hơn.
* Ví dụ: Nếu phân phối thu nhập của khách hàng AllElectronics là đối xứng và thu nhập trung bình là 56,000 USD, giá trị này có thể được sử dụng để thay thế giá trị bị thiếu.

5. Sử dụng giá trị trung tâm tính theo lớp của tuple

* Tính giá trị trung bình hoặc trung vị của thuộc tính cho tất cả các mẫu cùng thuộc một lớp (ví dụ: cùng nhóm rủi ro tín dụng).
* Ưu điểm:
  + Giúp tăng độ chính xác vì giá trị thay thế được tính theo đặc điểm của nhóm mà tuple thuộc về.
  + Với dữ liệu phân phối lệch, trung vị sẽ mang lại kết quả đáng tin cậy hơn.

6. Sử dụng giá trị có khả năng xảy ra nhất để điền giá trị thiếu

* Xác định giá trị dự đoán cho thuộc tính bị thiếu dựa trên hồi quy (Regression), các công cụ dựa trên suy luận sử dụng hình thức Bayesian (Bayesian formalism) hoặc quy nạp cây quyết định (Decision tree).
* Ưu điểm:
  + Phương pháp này được ưa chuộng do tận dụng được hầu hết thông tin từ các thuộc tính còn lại trong dữ liệu.
  + Giúp bảo toàn mối quan hệ giữa thu nhập và các thuộc tính khác trong quá trình dự đoán.

7.Đánh giá chung các phương pháp

* Các phương pháp từ 3 đến 6 đều có thể tạo ra sự thiên lệch cho dữ liệu vì giá trị đã điền có thể không chính xác hoàn toàn.
* Tuy nhiên, phương pháp 6 (sử dụng giá trị có khả năng xảy ra nhất) được đánh giá cao nhờ việc sử dụng nhiều thông tin có sẵn trong dữ liệu để dự đoán giá trị bị thiếu, từ đó bảo toàn mối quan hệ giữa các thuộc tính.

8. Lưu ý về giá trị bị thiếu

* Không phải tất cả giá trị bị thiếu đều là lỗi trong dữ liệu.

Ví dụ: Trong đơn xin thẻ tín dụng, người nộp đơn có thể không cung cấp số giấy phép lái xe nếu họ không có, và trường đó có thể hợp lý để để trống hoặc ghi “không áp dụng”.

* Các biểu mẫu nên cho phép người dùng chỉ định các giá trị như “không áp dụng”, “không biết”, hoặc để trống nếu sẽ được cung cấp ở bước sau.
* Thiết kế cơ sở dữ liệu và quy trình nhập liệu hợp lý có thể giúp giảm số lượng giá trị thiếu hoặc lỗi ngay từ đầu.

II. Xử lý dữ liệu nhiễu (Noisy Data)

1. Khái niệm về nhiễu (Noise)

Nhiễu là:

* Là sai số ngẫu nhiên hoặc biến động không mong muốn trong dữ liệu
* Thường biểu hiện qua các giá trị ngoại lệ (outliers)

2. Các kỹ thuật làm mịn dữ liệu nhằm Loại Bỏ Nhiễu

2.1 Phân nhóm (Binning):

a) Giới thiệu

Phân nhóm (Binning) là một kỹ thuật quan trọng trong tiền xử lý dữ liệu, được sử dụng để làm mịn dữ liệu số bằng cách giảm thiểu nhiễu và biến động ngẫu nhiên.

b) Các phương pháp phân nhóm chính

Phân nhóm tần số bằng nhau (Equal-frequency binning)

* Mỗi nhóm chứa cùng số lượng giá trị
* Ví dụ: Với kích thước nhóm = 3, dữ liệu giá cả [4, 8, 15, 20, 21, 24, 25, 28, 29] được chia thành:
  + Nhóm 1: [4, 8, 15]
  + Nhóm 2: [20, 21, 24]
  + Nhóm 3: [25, 28, 29]

Làm mịn bằng giá trị trung bình nhóm (smoothing by bin means)

* Thay thế mọi giá trị trong nhóm bằng trung bình của nhóm
* Ví dụ:
  + Nhóm 1: [4, 8, 15]
  + Trung bình: (4+8+15)/3 = 9
  + Kết quả: [9, 9, 9, ...]

Làm mịn bằng giá trị trung vị nhóm (smoothing by bin medians)

* Thay thế bằng median của nhóm
* Ví dụ:
  + Nhóm 1: [1, 2, 3, 4, 100]
  + Trung vị: 3
  + Kết quả: [3, 3, 3, ...]

Làm mịn bằng biên nhóm Smoothing by bin boundaries

* Xác định min/max của nhóm làm biên
* Thay thế mỗi giá trị bằng biên gần nhất
* Ví dụ:
  + Nhóm 1: [4, 8, 15]
  + Nhóm 1: Biên [4, 15]
  + Giá trị 8 → 4 (vì gần 4 hơn 15)
  + Kết quả: [4, 4, 15, ...]

2.2 Regression (Hồi Quy)

Hồi quy (Regression) là phương pháp làm mịn dữ liệu bằng cách điều chỉnh các giá trị dữ liệu theo một hàm số

Có hai dạng chính:

* Hồi quy tuyến tính (Linear regression):
  + Tìm đường thẳng "tốt nhất" để biểu diễn mối quan hệ giữa hai thuộc tính
  + Cho phép dự đoán một thuộc tính từ thuộc tính kia
* Hồi quy tuyến tính đa biến (Multiple linear regression):
  + Mở rộng của hồi quy tuyến tính
  + Xử lý nhiều hơn hai thuộc tính
  + Biểu diễn dữ liệu trên một bề mặt đa chiều

2.3 Phân tích ngoại lệ (Outlier Analysis):

Việc phát hiện giá trị ngoại lai có thể được thực hiện thông qua các phương pháp phân cụm, nơi các giá trị tương đồng được tổ chức thành các nhóm (clusters).

Phát hiện ngoại lệ bằng phân cụm:

* Các giá trị tương tự được tổ chức thành nhóm (cluster)
* Giá trị nằm ngoài các cụm được xem là ngoại lệ

A black and white drawing of three dots

AI-generated content may be incorrect.

Nhiều phương pháp làm mịn dữ liệu đồng thời cũng là phương pháp rời rạc hóa dữ liệu (một dạng biến đổi dữ liệu) (data discretization) và giảm dữ liệu(data reduction). Ví dụ:

* Các kỹ thuật phân nhóm (binning) đã mô tả ở trên làm giảm số lượng giá trị riêng biệt cho mỗi thuộc tính
* Điều này hoạt động như một dạng giảm dữ liệu cho các phương pháp khai thác dữ liệu dựa trên logic, như quy nạp cây quyết định - vốn thường xuyên so sánh các giá trị trên dữ liệu đã sắp xếp

Hệ thống phân cấp khái niệm (Concept hierarchies) là một dạng rời rạc hóa dữ liệu cũng có thể được sử dụng để làm mịn dữ liệu. Ví dụ:

* Một hệ thống phân cấp khái niệm cho thuộc tính *giá cả* có thể ánh xạ các giá trị *giá cả* thực tế thành các mức: *rẻ*, *trung bình* và *đắt*
* Qua đó giảm số lượng giá trị dữ liệu cần xử lý trong quá trình khai thác

III. Làm sạch dữ liệu như một quy trình trong khai thác dữ liệu (Data Cleaning as a Process)

1. Tổng quan

Làm sạch dữ liệu là một phần thiết yếu trong xử lý dữ liệu nhằm đảm bảo tính chính xác và nhất quán. Các giá trị bị thiếu, nhiễu, và không nhất quán có thể làm sai lệch kết quả phân tích. Trước đó, chúng ta đã tìm hiểu về cách xử lý dữ liệu bị thiếu và kỹ thuật làm mịn dữ liệu. Tuy nhiên, việc làm sạch dữ liệu không chỉ đơn thuần là sử dụng các kỹ thuật riêng lẻ mà còn là một quy trình toàn diện.

2. Phát hiện sai lệch (Discrepancy Detection)

a) Bước đầu tiên trong quá trình làm sạch dữ liệu là phát hiện sai lệch. Sai lệch có thể phát sinh từ nhiều nguyên nhân, bao gồm:

* Biểu mẫu nhập liệu thiết kế kém với nhiều trường tùy chọn.
* Lỗi do con người khi nhập dữ liệu.
* Lỗi cố ý (ví dụ người cung cấp không muốn tiết lộ thông tin cá nhân).
* Dữ liệu lỗi thời (ví dụ địa chỉ không còn chính xác).
* Sự không nhất quán trong cách biểu diễn dữ liệu và mã hóa.
* Lỗi từ thiết bị đo lường hoặc hệ thống ghi nhận dữ liệu.
* Sử dụng dữ liệu vào mục đích khác với mục đích ban đầu.
* Tích hợp dữ liệu từ nhiều nguồn với cách đặt tên thuộc tính khác nhau.

Để phát hiện sai lệch, người phân tích dữ liệu cần sử dụng metadata (dữ liệu về dữ liệu), bao gồm:

* Kiểu dữ liệu và miền giá trị của từng thuộc tính.
* Các giá trị hợp lệ.
* Mô tả thống kê cơ bản: trung bình, trung vị, mode, độ lệch chuẩn, phạm vi giá trị.
* Mối phụ thuộc giữa các thuộc tính.
* Các giá trị cách quá 2 lần độ lệch chuẩn (standard deviation) so với trung bình có thể được xem là ngoại lệ. Trong giai đoạn này, người dùng có thể viết script hoặc sử dụng các công cụ chuyên dụng.

Một số sai lệch điển hình có thể kể đến:

* Sử dụng không nhất quán mã hóa hoặc định dạng dữ liệu (ví dụ “2010/12/25” so với “25/12/2010”).
* Field overloading – khi nhà phát triển tận dụng phần chưa dùng của trường dữ liệu để lưu thông tin mới, gây lỗi.
* Không tuân thủ các quy tắc
* Quy tắc duy nhất (Unique rule): mỗi giá trị trong thuộc tính phải khác nhau.
* Quy tắc liên tục (Consecutive rule): giá trị không được thiếu trong khoảng nhất định.
* Quy tắc giá trị null (Null rule): cách xử lý giá trị trống, dấu hỏi, hoặc ký tự đặc biệt.

b) Xử lý giá trị null

Các lý do phổ biến gây ra giá trị null bao gồm:

* Người cung cấp từ chối trả lời hoặc thấy thông tin không phù hợp (ví dụ người không lái xe bỏ trống trường “số bằng lái”).
* Người nhập không biết giá trị đúng.
* Giá trị sẽ được bổ sung trong bước xử lý sau.

Quy tắc null cần xác định cách biểu diễn giá trị null (ví dụ: 0 với số, dấu cách với chuỗi, “?” hoặc “không biết”) và cách xử lý tương ứng.

c) Công cụ hỗ trợ phát hiện và xử lý sai lệch

* Data scrubbing tools: Sử dụng kiến thức miền như địa chỉ, kiểm tra chính tả để phát hiện và sửa lỗi.
* Data auditing tools: Phân tích dữ liệu để phát hiện quy luật và mối quan hệ, từ đó xác định các giá trị vi phạm. Có thể sử dụng phân tích thống kê hoặc phân cụm để phát hiện ngoại lệ.

3. Biến đổi dữ liệu (Data Transformation)

Sau khi phát hiện sai lệch, cần thực hiện biến đổi dữ liệu để sửa lỗi. Việc biến đổi có thể bao gồm:

* Sửa lỗi nhập liệu thủ công bằng cách đối chiếu giấy tờ.
* Dùng công cụ di chuyển dữ liệu (data migration tools) để thực hiện thay thế đơn giản (ví dụ đổi “gender” thành “sex”).
* Sử dụng ETL tools (Extraction/Transformation/Loading) với giao diện đồ họa để định nghĩa và thực hiện biến đổi. Tuy nhiên, chúng chỉ hỗ trợ tập hợp biến đổi giới hạn nên đôi khi cần viết script riêng.

4. Vấn đề và hướng tiếp cận mới

Quá trình phát hiện và biến đổi sai lệch là lặp đi lặp lại, dễ sai sót và tốn thời gian. Một số vấn đề gồm:

* Biến đổi sai có thể gây ra lỗi mới.
* Một số sai lệch lồng nhau chỉ lộ ra sau khi các lỗi khác được xử lý.
* Người dùng không nhận được phản hồi cho đến khi toàn bộ quy trình hoàn tất.
* Các dòng dữ liệu lỗi không được xử lý thường bị ghi vào file mà không giải thích lý do.

Để giải quyết vấn đề này, các công cụ mới tập trung vào tăng tính tương tác:

* Potter’s Wheel là một công cụ làm sạch dữ liệu công khai, tích hợp phát hiện sai lệch và biến đổi. Người dùng có thể thực hiện từng bước biến đổi trên giao diện dạng bảng tính, dễ dàng hoàn tác khi gặp lỗi. Công cụ này phát hiện sai lệch tự động dựa trên phiên bản dữ liệu mới nhất sau mỗi biến đổi.
* Một hướng khác là phát triển ngôn ngữ khai báo cho phép người dùng định nghĩa các toán tử biến đổi một cách hiệu quả (ví dụ mở rộng SQL với các cú pháp chuyên biệt).

5. Cập nhật metadata

Trong suốt quá trình làm sạch, việc cập nhật metadata là cực kỳ quan trọng nhằm lưu giữ các phát hiện mới về dữ liệu. Việc này giúp tăng tốc quá trình làm sạch cho các phiên bản dữ liệu sau.

6. Kết luận  
Quy trình làm sạch dữ liệu như một quy trình bao gồm hai giai đoạn chính:

* Phát hiện sự không nhất quán: Sử dụng kiến thức về metadata, các phương pháp thống kê cùng với công cụ chuyên dụng để nhận diện lỗi do nhập liệu, biểu diễn dữ liệu không nhất quán, …
* Biến đổi dữ liệu: Áp dụng các biến đổi cần thiết để sửa chữa các sai lệch, hỗ trợ bằng các công cụ thương mại và các script tùy chỉnh.

Quy trình này lặp đi lặp lại để đảm bảo dữ liệu sau cùng đạt được chất lượng cao. Việc tăng cường tính tương tác trong các công cụ và ngôn ngữ khai báo cũng giúp quá trình làm sạch trở nên hiệu quả hơn. Đồng thời, việc cập nhật metadata liên tục sẽ hỗ trợ việc làm sạch dữ liệu trong tương lai.