Tiền xử lý dữ liệu trong lĩnh vực học máy (Phần 2)

Các phương pháp xử lý dữ liệu chung

Như chúng ta đã thảo luận ở phần trước, một đặc trưng thường được phân loại vào một trong hai dạng là thuộc tính dạng số và thuộc tính dạng nhóm. Các xử lý dữ liệu bị khuyết cũng sẽ khác nhau ứng với hai loại thuộc tính này. Trong phần này, chúng ta cùng nhau xem xét các phương pháp điền vào dữ liệu bị khuyết để được bộ dữ liệu đầy đủ trước khi bước vào huấn luyện.

Với một đặc trưng dữ liệu dạng số, có rất nhiều lựa chọn mà chúng ta có thể xem xét khi điền vào một giá trị bị khuyết, ví dụ:

- Một giá trị hằng có ý nghĩa trong miền xác định của dữ liệu, ví dụ như 0.
- Một giá trị của một đặc trưng từ một mẫu dữ liệu ngẫu nhiên trong tập dữ liệu.
- Các giá trị thống kê cơ bản như giá trị trung bình, giá trị trung vị hay giá trị mốt (mode) của cột.
- Một giá trị được ước lượng từ một mô hình dự đoán khác.

```
### Mean, Median, Mode imputation ###

# create another dataset

mmmdataset = dataset

print(mmmdataset.describe())

# get values of dataset

values = mmmdataset.values

# create Imputer

imputer = Imputer(missing_values='NaN', strategy='mean') # strategy can be changed to "median" or "most_frequent"

# impute missing data by mean

transformed_values = imputer.fit_transform(values)

print(values)
```

Các đặc trưng dữ liệu dạng nhóm cần được tinh chỉnh một cách khéo léo hơn, vì vậy bạn cần chú ý nhiều đến hiệu năng của mô hình sau khi bạn tinh chỉnh (so sánh trước và sau khi áp dụng tinh chỉnh). Một số cách điền vào giá trị bị khuyết trong dữ liệu dạng nhóm là:

- Thay thế bằng giá trị xuất hiện nhiều nhất của đặc trưng đó trong toàn bộ tập dữ liệu. Tuy nhiên, cách này thường phát sinh độ lệch nhất định cho mô hình.
- Coi các giá trị bị khuyết là một giá trị mới trong tập giá trị đặc trưng nhóm.
- Sử dụng một mô hình dự đoán để ước lượng giá trị thay thế cho giá trị bị khuyết. Trong trường hợp này, chúng ta chia bộ dữ liệu ra thành hai phần bao gồm: Một phần chứa các dữ liệu đầy đủ để huấn luyện, phần còn lại chứa các điểm dữ liệu bị khuyết. Một số phương pháp có thể kể đến như hồi quy logistic (logistic regression), KNN hoặc phương pháp ANOVA.

Multiple Imputation

Multiple Imputaion (MI) là một cách để xử lý với dữ liệu bị khuyết hiệu quả. Với các phương

pháp điền đơn khả năng (giá trị) vào dữ liệu bị thiếu như là giá trị trung bình, trung vị hay bất kỳ một đặc trưng thống kê nào khác thì đều đi kèm một mức độ không chắc chắn nhất định về khả năng nhận định rằng những giá trị nào thì nên được điền vào. Phương pháp sử dụng tổ hợp đa khả năng để điền vào giá trị bị thiếu giúp giảm thiểu sự không chắc chắn trên bằng việc tính toán trên một vài lựa chọn khác nhau. Theo đó, một vài phiên bản dữ liệu hoàn thiện của dữ liệu được tạo ra. Cuối cùng, những phiên bản dữ liệu này được tổ hợp với nhau để tạo ra một phiên bản cuối cùng với các giá trị đã điền vào dữ liệu bị khuyết được coi là “chắc chắn” nhất.

Tiến trình thực hiện MI

Với phương pháp MI, các giá trị bị khuyết được thay thế bởi *m* khả năng khác nhau, với 1 < *m* < 10.

Item	Y	X
1	9	7
2	?	10
3	11	19
4	?	10
5	15	14
6	19	18
7	21	5
8	8	4
9	19	21
10	21	17

Bảng 3.2. Bảng dữ liệu bị khuyết

Năm 1987, Rubin đã tạo ra một phương pháp để thực hiện điền đa khả năng vào dữ liệu bị khuyết theo các bước sau:

- Đưa dữ liệu của bạn vào một mô hình thích hợp. Mô hình thích hợp lấy các mẫu dữ liệu và cố gắng để tìm ra mô hình phù tốt nhất, như là phân phối chuẩn hoặc phân phối chi bình

phương (chi-square distribution). Các mô hình này cũng có thể là một số mô hình tham số khác thu được từ dữ liệu. Như trong bảng dữ liệu trên, chúng ta có thể tạo ra hai mô hình đơn giản (tương ứng với hai khả năng có thể điền vào dữ liệu bị thiếu) là: *mô hình láng giềng gần nhất* (nearest neighbor), mô hình này lấy giá trị của láng giềng ở trên hoặc ở dưới và *mô hình làng giềng gần nhất + 25%* (nearest neighbor + 25%), mô hình này lấy các giá trị láng giềng gần nhất rồi tăng lên thêm 25%.

- Ước lượng một điểm dữ liệu bị khuyết sử dụng mô hình đã lựa chọn. Ví dụ, mô hình láng giềng gần nhất sẽ điền giá trị 9 vào Y2 (9 là giá trị của láng giềng gần nhất của Y2 là Y1.
- Lặp lại bước hai bước trên (bạn có thể sử dụng cùng một mô hình, hoặc các mô hình khác nhau) 2 – 5 lần cho mỗi điểm dữ liệu bị khuyết.

Model 1 (nearest neighbors)		
Item 2	9	11
Item 4	15	11
Model 2 (model 1 + 25%)		
Item 2	11	14
Item 4	19	14

- Thực hiện phân tích dữ liệu của ban. Bạn có thể chạy một phép kiểm thử t-test hoặc một phép kiểm thử ANOVA. Phép kiểm thử mà bạn sử dụng nên được chạy qua tất cả các bộ dữ liệu thu được từ các khả năng đã điền vào. Như trong ví dụ đang xét, chúng ta sẽ thu được bốn tập dữ liệu mới từ bước thứ ba, do vậy ta cũng cần kiểm thử bốn lần (mỗi tập một lần).

- Tính trung bình của các ước lượng tham số như phương sai, độ lệch chuẩn từ mỗi mô hình để thu được giá trị cuối cùng. Hay nói cách khác, bạn có thể tổng hợp các kết quả từ hai bộ dữ liệu đã tạo ra từ mô hình 1, và bạn cũng có thể tổng hợp kết quả của hai bộ dữ liệu đã tạo ra từ mô hình 2.

Mặc dù ví dụ chúng ta lấy ở bài viết này là tương đối trực quan, nhưng trong thực tế việc tính toán các giá trị bị khuyết gần đúng là rất phức tạp vì nó liên quan nhiều đến việc tổng hợp thông tin có trước về một tham số mà bạn quan tâm với các thông tin mới từ một mẫu trong phân tích Bayes (Bayesian analysis). Bên cạnh đó, việc tính toán cũng liên quan đến vấn đề lấy lại mẫu (resampling) trong các phân phối dự đoán, khi mà một số lớn các mẫu nhỏ cùng kích cỡ được liên tục rút ra (cùng với sự thay thế) từ một mẫu đơn ban đầu.

Người ta đã viết ra một module gọi là*fancyimpute* trong Python để hỗ trợ việc xử lý cho chúng ta:

```
### Multiple imputation by chained equations

from fancyimpute import MICE
```

```
# create another dataset for MI

mi = dataset

print(mi.describe())

# create Multiple imputation chained equations with 100 times of imputation and 5 nearest neighbors

mice = MICE(n_imputations=100, n_pmm_neighbors=5,init_fill_method='mean')

# impute data

mice.complete(mi)

print(mi.describe())
```

Maximum Likelihood trong xử lý dữ liệu bị khuyết

Giả định

Để tiến tới việc xử lý dữ liệu bị khuyết với phương pháp Maximum Likelihood, chúng ta phải tạo ra một số giả định rằng sự khuyết dữ liệu trên bất kỳ biến (đặc trưng) nào đều liên kết nhất định tới các biến khác. Người ta thường giả sử rằng dữ liệu thuộc dạng thiếu hoàn toàn ngẫu nhiêu (MCAR). MCAR giả sử rằng chỉ có một biến Y bị khuyết dữ liệu và tập các biến còn lại (gọi là vec-tơ X) đều có dữ liệu. Dữ liệu sẽ thuộc dạng MCAR nếu xác suất để Y bị khuyết không phụ thuộc vào vec-tơ X và chính đặc trưng Y (Rubin 1976). Biểu diễn dưới dạng công thức, ta gọi R là một “chỉ số phản hồi” có giá trị bằng 1 nếu Y bị khuyết và giá trị bằng 0 nếu Y được quan sát thấy. Khi đó, MCAR được biểu thị dưới dạng công thức như sau:

$$P(R = 1| X,Y) = P(R = 1)$$

Công thức trên ta có thể hiểu là xác suất để R = 1 (Y bị khuyết) trên điều kiện X, Y chính bằng xác suất để R = 1. Nghĩa là xác suất để R = 1 không phụ thuộc vào cả X và Y.

Ví dụ, nếu gọi Y biểu thị mức độ phạm pháp và X là số năm học (năm lớp 1, năm lớp 2, …) của một học sinh, MCAR sẽ có nghĩa là xác suất để dữ liệu bị khuyết trên mức độ phạm pháp là không liên quan tới cả đặc trưng mức độ phạm pháp và đặc trưng số năm học. Rất nhiều kĩ thuật xử lý dữ liệu bị khuyết truyền thống chỉ chạy được khi có giả định dữ liệu thuộc dạng MCAR (như hai phương pháp listwise và pairwise đã nói trong phần trước).

Một giả thiết yếu hơn (nhưng vẫn rất quan trọng) là dữ liệu thuộc dạng khuyết ngẫu nhiên (MAR). Cũng như MCAR, ta giả định rằng chỉ một biến Y bị khuyết dữ liệu còn tập các biến còn lại X đều được quan sát thấy. Khi đó, ta nói dữ liệu trên biến Y bị khuyết ngẫu nhiên nếu xác suất để Y bị khuyết không phụ thuộc vào Y, nhưng phụ thuộc vào X. Một cách công thức hóa, ta có:

$$Pr(R = 1| X,Y) = Pr (R = 1|X)$$

Trong đó, R vẫn là chỉ số phản hồi. Do vậy, MAR cho phép sự khuyết dữ liệu trên biến Y phụ thuộc vào các biến (quan sát thấy) khác. Nó chỉ không phụ thuộc vào chính nó (Y) mà thôi.

Vẫn với ví dụ nêu trên, nếu gọi Y là mức độ phạm pháp và X là số năm học của một học sinh thì MAR sẽ xảy ra nếu xác suất để mức độ phạm pháp bị khuyết phụ thuộc vào số năm học của một học sinh, nhưng trong mỗi năm học, xác suất để đặc trưng mức độ phạm pháp bị khuyết không phụ thuộc vào mức độ phạm pháp chung.

Về mặt bản chất, MAR cho phép dữ liệu bị khuyết phụ thuộc vào các đặc trưng không bị khuyết,

nhưng không phụ thuộc vào các giá trị bị khuyết. Do vậy, nếu một dữ liệu là MCAR thì nó cũng là MAR.

Việc kiểm tra xem liệu dữ liệu có phải là khuyết hoàn toàn MCAR hay không là không khó. Ví dụ, ta có thể so sánh giữa một người phụ nữ và một người đàn ông xem liệu họ có khác nhau về tỉ lệ các trường hợp bị khuyết dữ liệu về thu nhập hay không. Bất kỳ sự khác biệt nào cũng vi phạm đến MCAR. Tuy vậy, rất khó để kiểm tra xem liệu dữ liệu là khuyết ngẫu nhiên MAR nhưng không phải là khuyết hoàn toàn ngẫu nhiên MCAR. Lý do rất rõ ràng, chúng ta không thể chắc chắn rằng liệu những đứa trẻ phạm pháp có nhiều khả không cung cấp dữ liệu về mức độ phạm pháp hơn là những đứa trẻ không phạm pháp.

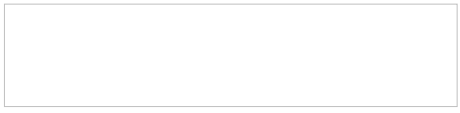
Điều gì sẽ xảy ra nếu dữ liệu bị khuyết không ngẫu nhiên (NMAR)? Thực tế, các trẻ em phạm pháp thường ít muốn chia sẻ về mức độ phạm pháp của chúng. Nếu dữ liệu thực sự là NMAR, thì cơ chế khuyết dữ liệu phải được mô hình hóa như một phần của tiến trình ước lượng để sinh ra các ước lượng tham số không lệch cho mô hình. Nghĩa là, nếu dữ liệu trên biến Y bị khuyết, ta phải xác định xem xác suất để Y bị khuyết phụ thuộc vào Y và các biến khác như thế nào. Điều này là không dễ bởi có vô vàn mô hình khác nhau mà ta có thể xác định được. Không có gì trong dữ liệu có thể giúp ta xác định được những mô hình nào là đúng. Và khi đó chúng ta sẽ phải lựa chọn các mô hình phù hợp theo cảm tính. Một nghiên cứu đã đề cập đến vấn đề về dữ liệu bị khuyết không ngẫu nhiên NMAR, và một số tiến trình đã được đề xuất. Nhưng các phương thức được đề xuất là rất phức tạp ngay cả trên những trường hợp dữ liệu đơn giản.

Vì những lý do đó, hầu hết các phương pháp nâng cao trong xử lý dữ liệu bị khuyết đều dựa trên giả định rằng dữ liệu bị khuyết ngẫu nhiên.

Maximum Likelihood

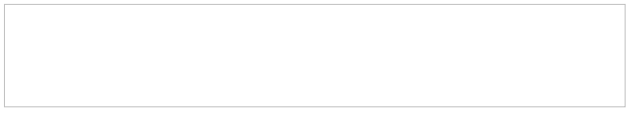
Bây giờ chúng ta đã sẵn sàng để xem xét maximum likelihood (ML), một phương pháp cạnh tranh với phương pháp multiple imputation. Cả hai phương pháp đều sinh ra các ước lượng có tính nhất quán và tương đối hiệu quả.

Bước đầu tiên của ước lượng ML là xây dựng hàm likelihood. Giả sử chúng ta có *n* quan sát độc lập (*i = 1, 2, …, n*) trên *k* biến (*yi1, yi2, …, yik*) mà không điểm dữ liệu nào bị khuyết. Khi đó, hàm likelihood được cho dưới dạng:



Trong đó, $fi(.)$ là hàm *xác suất chung (joint probability function)* hay *hàm mật độ xác suất (probability density function)*, của quan sát thứ *i*, và θ là một bộ tham số được ước lượng. Để thu được các ước lượng ML, chúng ta tìm giá trị của θ để *L* lớn nhất. Có rất nhiều phương pháp có thể giải quyết được vấn đề này và bất kỳ phương pháp đúng nào cũng đều sinh ra các kết quả giống nhau.

Bây giờ, ta giả sử rằng ở quan sát thứ *i*, hai biến *y1* và *y2* bị khuyết dữ liệu và chúng thỏa mãn giả định MAR. Xác suất chung cho quan sát đó chỉ là xác suất quan sát các biến còn lại từ *yi3* tới *yik*. Nếu *y1* và *y2* là rời rạc, xác suất chung ở trên là tổng trên tất cả các giá trị có thể có của hai biến với dữ liệu bị khuyết:



Nếu các biến bị khuyết dữ liệu thuộc dạng liên tục, chúng ta sử dụng tích phân tại chỗ của tổng:

Về mặt bản chất, với mỗi xác suất thành phần của một quan sát đóng góp vào hàm likelihood, chúng ta cộng tổng hoặc lấy tích phân trên các biến bị khuyết dữ liệu để thu được xác suất cận biên của giá trị các biến này mà quan sát được.

Như thông thường, giá trị likelihood tổng thể chỉ là sự kết hợp của các likelihood của toàn bộ quan sát. Ví dụ, nếu có *m* quan sát không bị khuyết dữ liệu và *n* - *m* quan sát bị khuyết dữ liệu trên hai biến y1 và y2, hàm likelihood cho toàn bộ tập dữ liệu sẽ là:

Trong đó, các quan sát được sắp xếp theo quy luật là *m* điểm dữ liệu đầu tiên không bị khuyết và *n* - *m* điểm dữ liệu tiếp theo bị khuyết. Likelihood này có thể được tối ưu để đạt cực đại để thu được các ước lượng tốt nhất của θ.

Dựa vào cơ sở lý thuyết đã trình bày, ta thấy rằng phương pháp sử dụng Maximum Likelihood không thay thế hoặc điền vào dữ liệu bị khuyết nhưng các dữ liệu bị khuyết sẽ được xử lý bên trong mô hình phân tích. Mô hình được ước lượng bởi phương pháp Full Information Maximum Likelihood (FIML), phương pháp này sử dụng tất cả các dữ liệu có sẵn (full information) để ước lượng mô hình. Qua đó, các thông số của bộ dữ liệu được ước lượng sẽ gần giống với các thông số của các dữ liệu mẫu đầy đủ đã được phân tích.

Chú ý: Chúng ta đã cùng nhau tìm hiểu rất nhiều cách xử lý dữ liệu bị khuyết khác nhau. Nhưng bạn cần lưu ý rằng dù bất kì cách xử lý dữ liệu nào đã thực hiện trên tập dữ liệu huấn luyện cũng sẽ phải được thực hiện trên dữ liệu kiểm thử hoặc dữ liệu mới trong tương lai để đảm bảo tính nhất quán của dữ liệu.

Mời bạn đọc tiếp chủ để tiền xử lý dữ liệu trong phần tiếp thep (phần 3).

References:

1. https://statisticalhorizons.com/wp-content/uploads/MissingDataByML.pdf
2. https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4
3. https://machinelearningmastery.com/handle-missing-data-python
4. https://arxiv.org/pdf/1710.01011.pdf
5. https://www.bu.edu/sph/files/2014/05/Marina-tech-report.pdf
6. *Time-Series Lecture at University of San Francisco by Nathaniel Stevens*