Zadanie klasyfikacji dla klas uporządkowanych

Ordinal classification problem

Warsztaty Badawcze 2

Karol Saputa, Małgorzata Wachulec, Aleksandra Wichrowska

Plan prezentacji

- Opis zagadnienia
- Metody rozwiązania
- Metryki
- Przykład dla omawianych metod - Wine Quality

Czym jest klasyfikacja klas uporządkowanych?

Klasyczny problem klasyfikacji wieloklasowej:

Rozpoznawanie choroby

vs.

Problem klasyfikacji klas uporządkowanych:

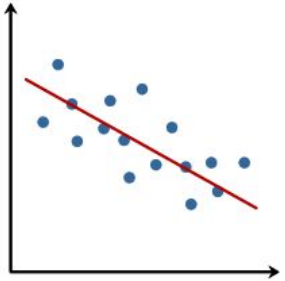
Ocena jakości wina

Możliwe rozwiązania

1. Zastosowanie zwykłego algorytmu klasyfikacji wieloklasowej

|good - average| ≠ |good - bad|

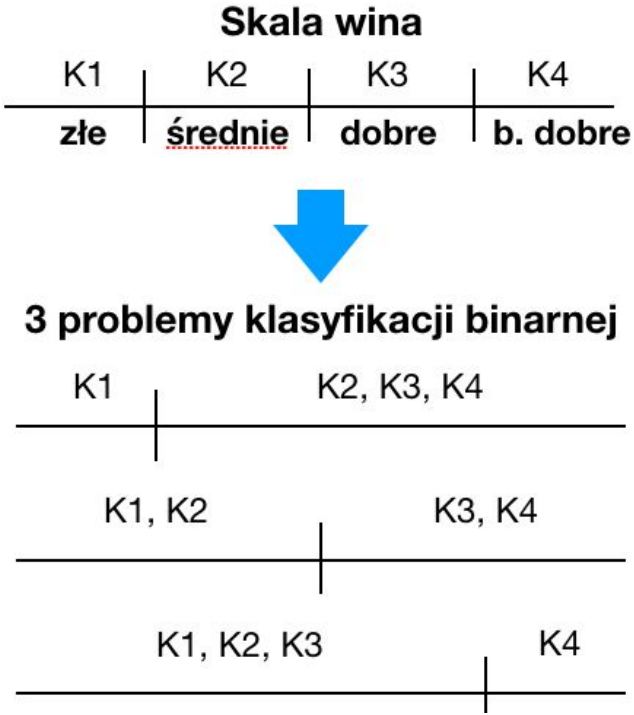
2. Zamiana zmiennej celu na zmienną numeryczną i dopasowanie modelu regresyjnego



3. Zamiana jednego problemu klasyfikacji m-klasowej na m-1 problemów klasyfikacji binarnej

m-1 zadań klasyfikacji binarnej

1. Nowe zmienne celu (ozn. y):
   - Czy wino jest lepsze niż "złe"/ "średnie"/ "dobre"?
2. Użycie klasyfikatora zwracającego dla każdej obserwacji:
   - P(y > "złe")
   - P(y > "średnie")
   - P(y > "dobre")
3. Wyliczenie prawdopodobieństw przynależenia do każdej z kategorii:
   - P(K1) = 1 - P(y > "złe")
   - P(K2) = P(y > "złe") - P(y > "średnie")
   - P(K3) = P(y > "średnie") - P(y > "dobre")
   - P(K4) = P(y > "dobre")



Źródło: Frank, E., & Hall, M. (2001). A Simple Approach to Ordinal Classification. *Machine Learning: ECML 2001 Lecture Notes in Computer Science*, 145–156. doi: 10.1007/3-540-44795-4_13

Uporządkowana regresja logistyczna

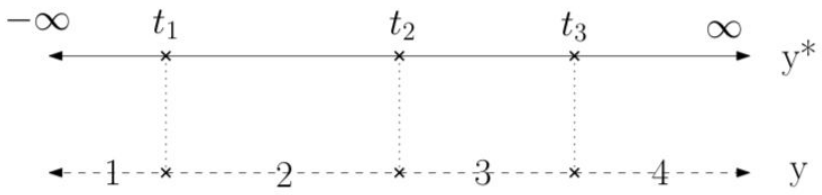
Problem zdefiniowany następująco:

- m - liczba klas
- y - prawdziwe etykiety
- $y^* = X\beta + \varepsilon$, $E\varepsilon = 0$
- $t_1, \dots, t_{m-1}$ - granice rozdzielające poszczególne klasy

Współczynniki β oraz $t_1, \dots, t_{m-1}$ są estymowane z danych tak, aby zachodziła zależność:

$$y = \begin{cases} 1 & \text{if } -\infty < y^* \leq t_1 \\ 2 & \text{if } t_1 < y^* \leq t_2 \\ \vdots & \vdots \\ m & \text{if } t_{m-1} < y^* < \infty \end{cases}$$

Przykład dla 4 klas:



Uporządkowana regresja logistyczna

P-stwo, że obserwacja przynależy do klasy k:

$$P(y_i = k| x_i) = P(t_{k-1} \leq y_i^* < t_k \mid x_i)$$

Po podstawieniu $y_i^* = \beta x_i + \varepsilon_i$:

$$P(y_i = k| x_i) = P(t_{k-1} \leq \beta x_i + \varepsilon_i < t_k \mid x_i)$$

$$P(y_i = k| x_i) = P(t_{k-1} - \beta x_i \leq \varepsilon_i < t_k - \beta x_i \mid x_i)$$

$$P(y_i = k| x_i) = F(t_k - \beta x_i) - F(t_{k-1} - \beta x_i)$$

gdzie F to dystrybuanta ε

Dla 4 klas dostajemy:

$$P(y_i = 1| x_i) = F(t_1 - \beta x_i)$$

$$P(y_i = 2| x_i) = F(t_2 - \beta x_i) - F(t_1 - \beta x_i)$$

$$P(y_i = 3| x_i) = F(t_3 - \beta x_i) - F(t_2 - \beta x_i)$$

$$P(y_i = 4| x_i) = 1 - F(t_3 - \beta x_i)$$

Miary oceny klasyfikacji

Podstawowe miary:

- accuracy (ACC)
- mean absolute error (MAE)
- mean squared error (MSE)

Inne miary:

- Accuracy within (ACC1, ACC2, itd.)
- Normalized Distance Performance Measure (NDPM)

Accuracy within (ACC1, ACC2, itd.)

- ACC within 1 (ACC1) $$\frac{\sum_{i=1}^{N} \mathbb{1}(|y_i - \widehat{y_i}| \in \{0, 1\})}{N}$$

- ACC within 2 (ACC2) $$\frac{\sum_{i=1}^{N} \mathbb{1}(|y_i - \widehat{y_i}| \in \{0, 1, 2\})}{N}$$

itd.

Uwaga: klasyczne ACC to ‘ACC within 0’

Normalized Distance Performance Measure (NDPM)

$$NDPM = \frac{2C^{-} + C^{u}}{2C}$$

System's ranking

Rank	Item
1	Item A
2	Item B
3	Item C
4	Item D
5	Item E

User's ranking

Rank	Item
1	Item A
2	Item B
2	Item C
4	Item E
5	Item D

C^{u}

C^{-}

C^{-} - liczba preferencji użytkownika przeciwstawnych względem preferencji systemu

C^{u} - liczba sytuacji, kiedy system rozróżnia przedmioty, a użytkownik traktuje je równoważnie.

C - liczba par preferencji użytkownika

Rezultaty uzyskane w artykule - poprawa dokładności

Wykonany eksperyment

- przekształcenie wartości numerycznych na kategoryczne przedziały dla 29 zbiorów danych
- wykorzystanie drzew decyzyjnych do rozwiązania podzadań klasyfikacji

Wyniki

- Zaproponowana metoda C4.5-ORD dla większości testowanych zbiorów danych uzyskuje lepsze wyniki niż klasyczne drzewa

	C4.5-ORD	C4.5	C4.5-1PC
C4.5-ORD	–	**4 (0)**	**6 (4)**
C4.5	**23 (18)**	–	15 (11)
C4.5-1PC	**22 (16)**	14 (7)	–

Tabela pokazuje jak często model w danej kolumnie osiąga istotnie wyższe wyniki niż modele w poszczególnych wierszach

Uzyskana w artykule poprawa wyników

Na zbiorach na których uzyskano istotnie wyższe wyniki dla proponowanej metody (oznaczone czarnymi kropkami) różnica wynosiła kilka punktów.
Proponowany wzrost dokładności jest więc stosunkowo niewielki, choć łatwy w uzyskaniu.

Table 2. Experimental results for target value discretized into five bins: percentage of correct classifications, and standard deviation

Dataset	C4.5-ORD	C4.5	C4.5-1PC
Abalone	48.08±0.48	46.34±0.73 •	49.55±0.65 ◦
Ailerons	59.24±0.30	56.97±0.35 •	55.58±0.34 •
Delta Ailerons	56.00±0.33	55.54±0.50 •	56.77±0.15 ◦
Elevators	50.34±0.28	47.76±0.29 •	50.72±0.33 ◦
Delta Elevators	50.01±0.38	47.63±0.42 •	50.34±0.29
2D Planes	75.37±0.11	75.37±0.06	75.29±0.07
Pole Telecom	95.05±0.12	95.05±0.10	94.94±0.07

Wykorzystanie metod dla zbioru Wine-Quality

Ocena jakości wina w skali 2-5 na podstawie parametrów chemicznych

1. Uporządkowana regresja logistyczna
- funkcja polr - proportional odds logistic regression

```
library(MASS)
m1 <- polr(Class ~ ., data = train_ds, Hess=TRUE)
m1_pred <- predict(m1, test_ds)
```

2. Implementacja metody z artykułu
- wykorzystanie lasów losowych (ranger) do określenie pomocniczych prawdopodobieństw
- możliwość zastosowania różnych modeli

Analiza wstępnych wyników

- dla małej liczby klas naturalnie występują wysokie wartości ACC1
- klasyczna regresja logistyczna uzyskuje istotnie większe wartości MSE
- dla implementacji artykułu istotny jest wybór metody rozwiązywania podzadań
- potrzeba wykonania dokładniejszych testów

Wartości metryk uzyskane dla zbioru Wine-Quality

	MSE	ACC	ACC1
Logistic Regression	0.61	0.53	0.96
Ordinal Logistic Regression	0.62	0.52	0.96
Implementacja artykułu (dla lasów)	0.37	0.69	0.98
Implementacja artykułu (dla regresji logistycznej)	0.61	0.54	0.96

Przydatne linki (i źródła)

1. https://www.cs.waikato.ac.nz/~eibe/pubs/ordinal_tech_report.pdf?fbclid=IwAR3d5m6JC3eZRnvajE-jQQ-d727b-r3rl021MWQTBb01LxCimW0s6uFfLc4
2. http://www.cs.uu.nl/docs/vakken/b3dar/ordinal-dict.pdf
3. https://link.springer.com/chapter/10.1007/978-3-642-01818-3_25
4. Jak to zrobić w Pythonie?
    a. https://github.com/sarvothaman/ordinal-classification/blob/master/ordinal-classification.ipynb
    b. https://pythonhosted.org/mord/