Implementacja algorytmu do odkrywania reguł asocjacyjnych

Mikołaj Bańkowski
Mateusz Kołacz

Prowadzący: dr inż. Robert Bembenik

Temat projektu

Podstawowo:
Implementacja algorytmu Apriori do odkrywania reguł asocjacyjnych wraz z implementacją obliczania miar: współczynnika podniesienia (*lift*) oraz trzech wybranych spośród: *conviction*, *cosine*, *gini*, *Jaccard/coherence*, *certainty factor*, *improvement*, *mutual information*, *odd ratio*. Przeprowadzić porównanie zachowania się wybranych miar w odniesieniu do wartości współczynnika podniesienia.

Dodatkowo:
Zaimplementować jeszcze jeden z algorytmów do odkrywania reguł asocjacyjnych:

- Algorytm AprioriRuleGen
- Algorytm Eclat
- Algorytm Partition

Zaimplementowane algorytmy do okrywania reguł asocjacyjnych

- Algorytm Apriori
- Algorytm Eclat

Dodatkowo przeprowadzona analiza

Przeprowadziliśmy porównanie własnej implementacji algorytmu Apriori z implementacją algorytmu Apriori znajdującą się w bibliotece języka Python – apyori oraz dokonaliśmy analizy porównawczej algorytmów Apriori i Eclat

Analiza porównawcza w obu przypadkach bierze pod uwagę:

- **Czasy wykonania** algorytmu przy różnych parametrach.
- **Uzyskiwane wyniki** dla różnych wartości progów wsparcia (*support*) i ufności (*confidence*).
- **Skuteczności algorytmu** na kilku różnych zbiorach danych wejściowych.

Dane

Dane będą pobierane z **UCI Machine Learning Repository** (https://archive.ics.uci.edu)

Wybrane zbiory danych:

Tic-Tac-Toe Endgame - Zbiór ten zawiera 958 rekordów

Car Evaluation - Zbiór ten zawiera 1728 rekordów

Nursery - Zbiór ten zawiera 12960 rekordów

Wszystkie zbiory danych mają reprezentację dyskretną

Pobieranie i preprocessing danych

Została zaimplementowana klasa data_manager.py

To klasa, która jest odpowiedzialna za dostarczenie danych do pozostałych obiektów. Umożliwia czerpanie danych zarówno z lokalnego pliku tekstowego oraz zbiorów danych udostępnianych przez repozytorium UCIML.

Następnie dane są przetwarzane i przygotowywane w taki sposób, aby były zgodne z wymaganiami wybranego algorytmu oraz umożliwiały jego efektywne działanie.

Apriori: Dane są przekształcane na poziome transakcje

Eclat: Dane są przekształcane w pionową strukturę TID-sets

Dane przekazywane do algorytmu **Apriori** są reprezentowane jako **lista transakcji**, gdzie każda transakcja to **frozenset**[1] zawierający unikalne elementy opisujące produkty lub cechy

```
[
    frozenset({'buns', 'hotdogs', 'ketchup'}),
    frozenset({'buns', 'hotdogs'}),
    frozenset({'chips', 'hotdogs', 'coke'}),
    frozenset({'chips', 'coke'}),
    frozenset({'chips', 'ketchup'}),
    frozenset({'chips', 'hotdogs', 'coke'}),
    frozenset({''}),
    frozenset({''}),
    frozenset({''})
]
```

python run.py -f DATASET.csv

Dane przekazywane do algorytmu **Eclat** są reprezentowane jako TID-Sets, gdzie kluczami są **frozensety**[1] zawierające pojedyncze unikalne elementy a wartościami są **zbiory identyfikatorów transakcji** które wskazują, w których transakcjach dany element występuje.

```
{
    frozenset({'buns'}): {0, 1},
    frozenset({'hotdogs'}): {0, 1, 2, 5},
    frozenset({'ketchup'}): {0, 4},
    frozenset({'coke'}): {2, 3, 5},
    frozenset({'chips'}): {2, 3, 4, 5},
    frozenset({''}): {6, 7, 8}
}
```

python run.py -f DATASET.csv -a eclat

frozenset[1] wybraliśmy ze względu na swoje właściwości niezmienności oraz możliwość wykorzystania jako klucz w strukturach danych takich jak słowniki czy zbiory frozenset obsługuje typowe operacje zbiorowe, takie jak przecięcie, suma , różnica oraz różnica symetryczna

Wybrane miary

- Współczynnik podniesienia (lift),

$$\mathrm{Lift}(A \to B) = \frac{\mathrm{support}(A \cap B)}{\mathrm{support}(A) \times \mathrm{support}(B)}$$

- Współczynnik pewności (certainty factor),

$$\text{Relative Support}(A \to B) = \mathrm{support}(A \cap B)$$

- Jaacard / coherence,

$$\mathrm{CF}(A \to B) = \frac{\mathrm{support}(A \cap B) - \mathrm{support}(A) \times \mathrm{support}(B)}{1 - \mathrm{support}(B)}$$

- iloraz szans (odds-ratio)

$$\text{Odd Ratio}(A \to B) = \frac{\mathrm{support}(A \cap B) \times \mathrm{support}(\neg A \cap \neg B)}{\mathrm{support}(A \cap \neg B) \times \mathrm{support}(\neg A \cap B)}$$

A→B reprezentuje **regułę asocjacyjną,** jeśli występuje zbiór A (np. pewne produkty w koszyku zakupowym), to z pewnym prawdopodobieństwem występuje także zbiór B

Implementacja sposobu na wyznaczanie miar

Proces obliczania miar:

1. Wygenerowanie reguł asocjacyjnych
2. Obliczenie wsparcia[1] i ufności[2]
3. Obliczenie poszczególnych miar jakości

Oba algorytmy, **Apriori** i **Eclat**, bazują na tym samym sposobie wyznaczania miar jakości reguł asocjacyjnych. Wykorzystują te same metody do obliczania wskaźników, takich jak współczynnik podniesienia (lift), wsparcie relatywne (relative support), czy certainty factor.

Wsparcie[1] określa, jak często zbiór elementów (np. produkty lub zdarzenia) pojawia się w całym zbiorze danych
Ufność[2] określa, jak często występuje zbiór B w transakcjach, które już zawierają zbiór A. Innymi słowy, jest to prawdopodobieństwo warunkowe P(B|A)

Sposób działania

Skrypt umożliwiający uruchomienie algorytmu do wykrywania często współwystępujących wzorców w zbiorze danych. Aby korzystać ze skryptu, należy uruchomić go z odpowiednimi flagami w linii poleceń:

-f, --input-file

-a, --algorithm

-u, --UCI-dataset

-s, --min-support

-c, --min-confidence

Przykład uruchomienia skryptu z kilkoma flagami jednocześnie:

python script.py -a eclat -u nursery -s 0.2 -c 0.7

Przeprowadzone testy

- Porównanie własnej implementacji algorytmu Apriori z algorytmem Apriori dostępnym w bibliotece Python - apyori
- Porównanie własnej implementacji algorytmu Apriori z algorytmem Eclat
- Porównanie zachowania się wybranych miar w odniesieniu do wartości współczynnika podniesienia

Apriori vs Apriori biblioteka

W ramach tej części testów stworzona implementacja algorytmu Apriori była porównywana z dostępną w języku Python biblioteką **apyori**.

Porównane zostały takie własności jak:

- **procent odnalezionych reguł asocjacyjnych**,
- **procent poprawnie wyznaczonych wartości współczynnika confidence** (w odniesieniu do wyników zwracanych przez implementację **apyori**)
- **czasy wykonywania obydwu rozwiązań**.

Porównanie odbyło się przy uwzględnieniu **różnych zbiorów danych** i wartości współczynników **min_support** oraz **min_confidence**. Czasy egzekucji były mierzone dla **1 000 000 wywołań funkcji** wyznaczającej reguły asocjacyjne dla każdego zestawu parametrów wejściowych.

Dataset	Min Support	Min Confidence	Rules Match	Confidence Match	My Exec Time	Apyori Exec Time
car_evaluation	0.15	0.15	100.00%	100.00%	2.041 ms	0.271 ms
car_evaluation	0.15	0.5	100.00%	100.00%	2.623 ms	0.311 ms
car_evaluation	0.15	0.8	100.00%	100.00%	2.622 ms	0.435 ms
car_evaluation	0.5	0.15	100.00%	100.00%	2.271 ms	0.353 ms
car_evaluation	0.5	0.5	100.00%	100.00%	2.919 ms	0.281 ms
car_evaluation	0.5	0.8	100.00%	100.00%	2.063 ms	0.268 ms
car_evaluation	0.8	0.15	100.00%	100.00%	2.473 ms	0.329 ms
car_evaluation	0.8	0.5	100.00%	100.00%	2.361 ms	0.328 ms
car_evaluation	0.8	0.8	100.00%	100.00%	2.139 ms	0.331 ms
tic_tac_toe_endgame	0.15	0.15	100.00%	100.00%	2.341 ms	0.415 ms
tic_tac_toe_endgame	0.15	0.5	100.00%	100.00%	2.844 ms	0.321 ms
tic_tac_toe_endgame	0.15	0.8	100.00%	100.00%	2.164 ms	0.268 ms
tic_tac_toe_endgame	0.5	0.15	100.00%	100.00%	2.242 ms	0.271 ms
tic_tac_toe_endgame	0.5	0.5	100.00%	100.00%	1.949 ms	0.264 ms
tic_tac_toe_endgame	0.5	0.8	100.00%	100.00%	2.214 ms	0.291 ms
tic_tac_toe_endgame	0.8	0.15	100.00%	100.00%	2.035 ms	0.276 ms
tic_tac_toe_endgame	0.8	0.5	100.00%	100.00%	1.994 ms	0.272 ms
tic_tac_toe_endgame	0.8	0.8	100.00%	100.00%	2.190 ms	0.341 ms
nursery	0.15	0.15	100.00%	100.00%	2.926 ms	0.259 ms
nursery	0.15	0.5	100.00%	100.00%	3.055 ms	0.271 ms
nursery	0.15	0.8	100.00%	100.00%	3.021 ms	0.263 ms
nursery	0.5	0.15	100.00%	100.00%	2.088 ms	0.258 ms
nursery	0.5	0.5	100.00%	100.00%	2.072 ms	0.257 ms
nursery	0.5	0.8	100.00%	100.00%	2.108 ms	0.264 ms
nursery	0.8	0.15	100.00%	100.00%	2.039 ms	0.257 ms
nursery	0.8	0.5	100.00%	100.00%	2.038 ms	0.265 ms
nursery	0.8	0.8	100.00%	100.00%	2.064 ms	0.262 ms

Wnioski (dotyczące Apriori)

Implementacja algorytmu Apriori w ramach projektu osiągnęła pełne pokrycie z wynikami uzyskanymi za pomocą popularnej biblioteki apyori, co świadczy o poprawności jej działania

Pomimo pełnego pokrycia, własna implementacja algorytmu Apriori działa średnio 10 razy wolniej niż biblioteka apyori. W przyszłości istnieje możliwość optymalizacji kodu w celu poprawy wydajności - jest to jednak akceptowalna różnica, ponieważ biblioteka apyori była tworzona znacznie dokładniej i ze znacznie większym naciskiem na doprecyzowanie niż niniejszy projekt.

Apriori vs Eclat

W ramach tej części testów stworzona implementacja algorytmu **Apriori** była porównywana z stworzoną implementacją algorytmu **Eclat**.

Porównane zostały takie własności jak:

- **procent odnalezionych reguł asocjacyjnych**,
- **procent poprawnie wyznaczonych wartości współczynnika confidence** (w odniesieniu do wyników zwracanych przez implementację **apyori**)
- **czasy wykonywania obydwu rozwiązań**.

Porównanie odbyło się przy uwzględnieniu **różnych zbiorów danych** i wartości współczynników **min_support** oraz **min_confidence**. Czasy egzekucji były mierzone dla **1 000 000 wywołań funkcji** wyznaczającej reguły asocjacyjne dla każdego zestawu parametrów wejściowych.

Dataset	Min Support	Min Confidence	Rules Match	Confidence Match	Apriori Exec Time	Eclat Exec Time
car_evaluation	0.15	0.15	100.00%	100.00%	1.742 s	1.182 s
car_evaluation	0.15	0.5	100.00%	100.00%	1.755 s	1.194 s
car_evaluation	0.15	0.8	100.00%	100.00%	1.795 s	1.252 s
car_evaluation	0.5	0.15	100.00%	100.00%	1.766 s	1.208 s
car_evaluation	0.5	0.5	100.00%	100.00%	1.743 s	1.237 s
car_evaluation	0.5	0.8	100.00%	100.00%	1.724 s	1.215 s
car_evaluation	0.8	0.15	100.00%	100.00%	1.767 s	1.211 s
car_evaluation	0.8	0.5	100.00%	100.00%	1.766 s	1.225 s
car_evaluation	0.8	0.8	100.00%	100.00%	1.788 s	1.217 s
tic_tac_toe_endgame	0.15	0.15	100.00%	100.00%	1.752 s	1.214 s
tic_tac_toe_endgame	0.15	0.5	100.00%	100.00%	1.833 s	1.312 s
tic_tac_toe_endgame	0.15	0.8	100.00%	100.00%	1.905 s	1.212 s
tic_tac_toe_endgame	0.5	0.15	100.00%	100.00%	1.728 s	1.194 s
tic_tac_toe_endgame	0.5	0.5	100.00%	100.00%	1.727 s	1.197 s
tic_tac_toe_endgame	0.5	0.8	100.00%	100.00%	1.749 s	1.201 s
tic_tac_toe_endgame	0.8	0.15	100.00%	100.00%	1.723 s	1.198 s
tic_tac_toe_endgame	0.8	0.5	100.00%	100.00%	1.733 s	1.197 s
tic_tac_toe_endgame	0.8	0.8	100.00%	100.00%	1.735 s	1.203 s
nursery	0.15	0.15	100.00%	100.00%	2.738 s	1.410 s
nursery	0.15	0.5	100.00%	100.00%	2.762 s	1.388 s
nursery	0.15	0.8	100.00%	100.00%	2.726 s	1.376 s
nursery	0.5	0.15	100.00%	100.00%	1.889 s	1.253 s
nursery	0.5	0.5	100.00%	100.00%	1.888 s	1.257 s
nursery	0.5	0.8	100.00%	100.00%	1.892 s	1.241 s
nursery	0.8	0.15	100.00%	100.00%	1.860 s	1.257 s
nursery	0.8	0.5	100.00%	100.00%	1.908 s	1.342 s
nursery	0.8	0.8	100.00%	100.00%	1.885 s	1.257 s

Wnioski (dotyczące Apriori vs Eclat)

W kontekście porównania algorytmów Apriori i Eclat wykazano, że oba algorytmy generują identyczne reguły asocjacyjne, co potwierdza ich spójność i poprawność działania natomiast algorytm Eclat jest znacznie szybszy niż algorytm Apriori

Przewaga czasowa algorytmu Eclat wynika z wykorzystania pionowej reprezentacji danych (TID-sets) oraz bardziej efektywnego przeszukiwania przestrzeni kombinacji poprzez operacje na zbiorach identyfikatorów transakcji.

W praktyce, dla dużych zbiorów danych, rekomenduje się użycie algorytmu Eclat, natomiast dla mniejszych zbiorów danych, oba algorytmy działają efektywnie i ich wybór może zależeć od innych czynników, takich jak dostępne zasoby obliczeniowe czy potrzeba interpretowalności wyników.

Porównanie zachowania się wybranych miar w odniesieniu do wartości współczynnika podniesienia

W ramach niniejszej analizy, własna implementacja algorytmu Apriori została użyta do wyznaczenia reguł asocjacyjnych i powiązanych z nimi metryk na trzech dyskretnych zbiorach danych pochodzących z UCIML repo.

Uzyskane wyniki zostały posortowane na podstawie wsp. podniesienia - lift a następnie zaprezentowane w postaci tabeli ogólnej i tabeli uśrednionych wartości metryk w zależności od wartości wsp. lift.

Wszystkie poniższe wyniki zostały uzyskane przy wartości min_support = 0.15 oraz min_confidence = 0.3.

Lift Range	Rule	Relative Support	Lift	Certainty	Jaccard	Odds Ratio
0.6-0.7	('safety_high',) ==> ('target_unacc',)	0.16	0.69	1.44	0.87	2.06
0.7-0.8	('persons_4',) ==> ('target_unacc',)	0.18	0.77	1.63	1.10	2.32
0.7-0.8	('persons_more',) ==> ('target_unacc',)	0.19	0.80	1.68	1.18	2.40
0.8-0.9	('safety_med',) ==> ('target_unacc',)	0.21	0.89	1.86	1.50	2.66
0.8-0.9	('maint_med',) ==> ('target_unacc',)	0.16	0.89	2.48	1.88	3.54
0.8-0.9	('maint_low',) ==> ('target_unacc',)	0.16	0.89	2.48	1.88	3.54
0.8-0.9	('buying_med',) ==> ('target_unacc',)	0.16	0.89	2.48	1.88	3.54
0.9-1.0	('doors_4',) ==> ('target_unacc',)	0.17	0.97	2.70	2.46	3.86
0.9-1.0	('doors_3',) ==> ('target_unacc',)	0.17	0.99	2.78	2.71	3.97
0.9-1.0	('target_unacc',) ==> ('lug_boot_big',)	0.21	0.91	0.43	0.42	1.30
0.9-1.0	('lug_boot_big',) ==> ('target_unacc',)	0.21	0.91	1.92	1.62	2.74
0.9-1.0	('lug_boot_med',) ==> ('target_unacc',)	0.23	0.97	2.04	1.93	2.92
0.9-1.0	('target_unacc',) ==> ('lug_boot_med',)	0.23	0.97	0.46	0.46	1.39
0.9-1.0	('doors_5more',) ==> ('target_unacc',)	0.17	0.97	2.70	2.46	3.86
1.0-1.1	('buying_high',) ==> ('target_unacc',)	0.19	1.07	3.00	3.75	4.28
1.0-1.1	('maint_high',) ==> ('target_unacc',)	0.18	1.04	2.91	3.25	4.15
1.0-1.1	('doors_2',) ==> ('target_unacc',)	0.19	1.08	3.02	3.86	4.31
1.1-1.2	('buying_vhigh',) ==> ('target_unacc',)	0.21	1.19	3.33	7.13	4.76
1.1-1.2	('target_unacc',) ==> ('lug_boot_small',)	0.26	1.12	0.53	0.56	1.59
1.1-1.2	('lug_boot_small',) ==> ('target_unacc',)	0.26	1.12	2.34	3.10	3.35
1.1-1.2	('maint_vhigh',) ==> ('target_unacc',)	0.21	1.19	3.33	7.13	4.76
1.4-1.5	('target_unacc',) ==> ('persons_2',)	0.33	1.43	0.68	0.85	2.04
1.4-1.5	('persons_2',) ==> ('target_unacc',)	0.33	1.43	3.00	29.79	4.28
1.4-1.5	('safety_low',) ==> ('target_unacc',)	0.33	1.43	3.00	29.79	4.28
1.4-1.5	('target_unacc',) ==> ('safety_low',)	0.33	1.43	0.68	0.85	2.04

Pełna tabela wyników dla zbioru danych Car Evaluation

Lift Range	No. Rules	Relative Support [Avg. / Std. dev.]	Certainty [Avg. / Std. dev.]	Jaccard [Avg. / Std. dev.]	Odds Ratio [Avg. / Std. dev.]
0.6-0.7	1	[0.160 / 0.000]	[1.443 / 0.000]	[0.870 / 0.000]	[2.060 / 0.000]
0.7-0.8	2	[0.183 / 0.000]	[1.651 / 0.001]	[1.140 / 0.001]	[2.358 / 0.001]
0.8-0.9	4	[0.168 / 0.000]	[2.326 / 0.073]	[1.785 / 0.027]	[3.322 / 0.148]
0.9-1.0	7	[0.199 / 0.001]	[1.863 / 0.898]	[1.723 / 0.778]	[2.862 / 1.120]
1.0-1.1	3	[0.186 / 0.000]	[2.975 / 0.002]	[3.619 / 0.069]	[4.249 / 0.005]
1.1-1.2	4	[0.234 / 0.001]	[2.385 / 1.309]	[4.479 / 7.824]	[3.615 / 1.696]
1.4-1.5	4	[0.333 / 0.000]	[1.840 / 1.346]	[15.323 / 209.370]	[3.162 / 1.260]

Tabela wyników uśrednionych dla zbioru danych Car Evaluation

Wnioski (dotyczące wsp. lift vs pozostałe miary)

Na podstawie powyższej tabeli można stwierdzić, że współczynnik wsparcia relatywnego (**relative support**) jest tym większy, im wyższy współczynnik podniesienia lift. Widać też, że wartości w poszczególnych kubełkach nie odbiegają znacząco od siebie (minimalne wartości odchylenia standardowego).

W przypadku współczynnika **Certainty**, nie widać dokładnej zależności pomiędzy wartościami wsp. lift, jednak można dostrzec znacznie wyższą wartość odchylenia standardowego

Współczynnik **Jaccarda** rośnie wykładniczo w stosunku do wartości współczynnika podniesienia. To samo dotyczy odchylenia standardowego jego wartości w zadanym przedziale.

W przypadku współczynnika **Odds Ratio** również widać tendencje wzrostową w stosunku do wartości wsp. lift, jednak nie jest to silnie widoczne zarówno w stosunku do wartości średniej jak i odchylenia standardowego.

Analiza zachowania się wybranych miar w odniesieniu do wartości współłczynnika podniesienia (lift) wykazała, że **większość miar (współczynnik Jaccarda, Certainty Factor, Odds Ratio) wykazuje tendencję do wzrostu wraz ze wzrostem wartości współczynnika** podniesienia.

Dokładność i wiarygodność wyników uzyskanych przez algorytm Apriori zależy od odpowiedniego dobrania parametrów, takich jak minimalne wsparcie (min_support) i minimalne zaufanie (min_confidence). Warto przeprowadzić analizę wrażliwości, aby określić optymalne wartości tych parametrów dla konkretnego zestawu danych.

Dziękujemy
za
uwagę