ZUM projekt - dokumentacja wstępna

Mateusz Szczęsny, Małgorzata Kubiak

1. Temat projektu

Nienadzorowana detekcja anomalii na podstawie niepodobieństwa do sąsiadów z możliwością użycia dowolnej miary niepodobieństwa. Porównanie z nienadzorowaną detekcją anomalii za pomocą algorytmów klasyfikacji jednoklasowej dostępnych w środowisku R lub Python.

Interpretacja tematu

Do nienadzorowanej detekcji anomalii wykorzystamy algorytm bazujący na KNN (*K* najbliższych sąsiadów). Algorytm został wprowadzony w pracy [5]. Przyjmuje się, że obiekt jest podobny do tych, które są mu najbliższe. Głównym krokiem jest wybór liczby *k* najbliższych sąsiadów, gdzie *k* jest parametrem. Dla danego punktu algorytm szuka *k* najbliższych sąsiadów w zestawie uczącym, a następnie przypisuje punktowi kategorię, która pojawia się najczęściej wśród tych sąsiadów. W naszej modyfikacji obiekty zbyt oddalone od swoich *k* sąsiadów będą uznawane za anomalie. Do oceny niepodobieństwa wykorzystamy kilka wybranych miar. Algorytm zostanie zaimplementowany w języku Python.

2. Zbiory danych

W projekcie wykorzystamy dwa zbiory danych. Pierwszy to „Thyroid Disease dataset” (https://odds.cs.stonybrook.edu/thyroid-disease-dataset/) - zestaw danych dotyczący chorób tarczycy, pochodzący z repozytorium UCI Machine Learning, przeznaczony do klasyfikacji w celu określenia, czy pacjent skierowany do kliniki jest chory na niedoczynność tarczycy. Zbiór zawiera 3772 próbki o 6 rzeczywistych atrybutach i 15 kategorycznych, ale na potrzeby detekcji anomalii wykorzystane zostały jedynie rzeczywiste. Zawiera trzy klasy: normalna (brak niedoczynności tarczycy), hiperfunkcja i nienormalne funkcjonowanie. Hiperfunkcja jest klasą odstającą (outlier), a pozostałe dwie stanowią normalne klasy (inlier). Anomalie stanowią 2.5% wszystkich przykładów.

Drugi zbiór danych to “Shuttle” ze strony https://odds.cs.stonybrook.edu/shuttle-dataset/, pochodzący z repozytorium UCI Machine Learning. Zawiera 49097 próbki, z czego 7% stanowią anomalie. Nominalnie służy do klasyfikacji wieloklasowej z dziewięcioma klasami. Na potrzeby detekcji anomalii pięć najmniejszych klas (2, 3, 5, 6, 7) zostało połączonych, tworząc klasę odstających (outlier), podczas gdy klasa 1 stanowi klasę normalną (inlier). Dane klasy 4 zostały odrzucone.

3. Plan implementacji

Algorytm niepodobieństwa do sąsiadów działa w następujących krokach:

1. Obliczanie miary podobieństwa do wszystkich przykładów trenujących
2. Znalezienie *K* najbardziej podobnych przykładów
3. Ocena niepodobieństwa

Na podstawie wartości podobieństwa oceniamy czy przykład jest na tyle podobny do zestawu trenującego, żeby uznać, że jest poprawny czy jest na tyle niepodobny, że uznajemy go za anomalie.

Argumentami funkcji będą *K* definiowane jako liczba najbliższych według wybranej miary podobieństwa przykładów trenujących, które zostaną wzięte pod uwagę przy decyzji czy dany przykład jest anomalią, funkcja, według której wyliczane będzie niepodobieństwo oraz funkcja mówiąca jak interpretować niepodobieństwo (zwraca podobny albo niepodobny).

Przykładowe miary, które mogą zostać użyte to proste miary, podobne do tych stosowanych w klasyfikatorze nadzorowanym jak odległość Lorenza albo Euklidesa [2], albo bardziej złożone metody jak Local Outlier Factor (LOF) [1]. W przypadku prostych miar jak odległość Euklidesa, sposób interpretacji miary odległości ograniczy się do porównania wartości miary z jakimś progiem (np. średnia odległość od siebie punktów w zbiorze treningowym). W przypadku LOF funkcja interpretujące wyniki może nawet ponownie przeszukać otoczenie sąsiadów i obliczyć dodatkowe metryki.

Problemem w używaniu tego algorytmu może być długi czas obliczeń dla dużych zbiorów danych, ponieważ przy każdej klasyfikacji, trzeba obliczyć miarę podobieństwa dla każdego przykładu w zbiorze trenującym. Wiele publikacji [3, 4] skupia się na aspekcie optymalizacji szybkości działania algorytmu dla dużych zbiorów danych. Na potrzeby tego projektu nie będziemy wprowadzać żadnych optymalizacji poza najprostszymi jeśli nasza implementacja okaże się zbyt wolna do przeprowadzenia testów w akceptowalnym czasie.

4. Lista algorytmów do eksperymentów

Lista algorytmów i bibliotek

1. Algorytmy z biblioteki SKlearn
    a. Algorytmy do porównania detekcji anomalii
        i. LOF
        ii. Isolation forest
        iii. Fitting an elliptic envelope
2. Matplotlib jako narzędzie do wizualizacji wyników
3. Numpy jako bilbioteka do obliczeń
4. Pandas dataframe

5. Plan badań

W badaniach zostanie wybrane kilka miar podobieństwa oraz sposobów ich oceny. W tym momencie rozważamy użycie:

- Dystansu Euklidesa
- Dystansu Lorenzian
- Dystansu Manhattan
- LOF

Algorytm zostanie nauczony na zbiorze złożonym z przykładów prawidłowych (nie anomalii). Parametry zostaną dostrojone na zbiorze walidacyjnym składającym się z przykładów będących zarówno anomaliami jak i nie. Dobranie właściwych parametrów jest kluczowe dla działania tego algorytmu, ponieważ trzeba np zdecydować, jaki jest akceptowalny próg niepodobieństwa. Na koniec wyniki zostaną ocenione przy pomocy metryki F1 na zbiorze testowym.

Dane zostaną podzielone na zbiór treningowy, walidacyjny i testowy. Zbiór treningowy będzie zawierał 60% przykładów, z czego wszystkie będą nie anomaliami. Zbiory walidacyjny i testowy będą stanowiły po 20% zbiory danych i będą miały podobną proporcję anomalii do przykładów prawidłowych.

Testy zostaną przeprowadzone na wszystkich wybranych zbiorach danych. Sprawdzane będą również wybrane algorytmy nienadzorowanej detekcji anomalii z biblioteki SKlearn. Zostaną one sprawdzone na tych samych danych w ten sam sposób.

Wyniki zostaną przedstawione na wykresach oraz w tabelach.

6. Bibliografia

1. Campos, Guilherme O., et al. "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study." *Data mining and knowledge discovery* 30 (2016): 891-927.
2. Prasatha, V. S., et al. "Effects of distance measure choice on knn classifier performance-a review." *arXiv preprint arXiv:1708.04321* (2017): 56.
3. Ramaswamy, Sridhar, Rajeev Rastogi, and Kyuseok Shim. "Efficient algorithms for mining outliers from large data sets." *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000.
4. Knorr, Edwin M., Raymond T. Ng, and Vladimir Tucakov. "Distance-based outliers: algorithms and applications." *The VLDB Journal* 8.3 (2000): 237-253.
5. Fix, E. and Hodges, J.L. (1951). "*Discriminatory analysis. Nonparametric discrimination; consistency properties*". Technical Report 4, USAF School of Aviation Medicine, Randolph Field, TX, USA, 1951.