Zuzanna Górecka

Kamil Szydłowski

ZAAWANSOWANE UCZENIE MASZYNOWE

NIENADZOROWANA DETEKCJA ANOMALII NA PODSTAWIE NIEPODOBIEŃSTWA DO SĄSIADÓW

Interpretacja tematu projektu

Temat 13: 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.

Projekt ma na celu zaznajomienie się z zagadnieniem nienadzorowanej detekcji anomalii. W ramach projektu zostanie zaimplementowany algorytm do wykrywania anomalii na podstawie niepodobieństwa do sąsiadów. Jako parametr będzie można podać dowolną miarę niepodobieństwa.

Poprawność implementacji będzie można sprawdzić porównując wyniki dla wskaźnika nieprawidłowości *Local outlier factor* z gotową implementacją z biblioteki *scikit-learn* z pakietu *sklearn.neighbors.LocalOutlierFactor*.

Następnie zaimplementowany przez nas algorytm zostanie zbadany oraz porównany z algorytmami klasyfikacji jednoklasowej z *scikit-learn* – lasem izolacyjnym oraz jednoklasowym SVM.

Część implementacyjna

Opracowana procedura do detekcji anomalii będzie składała się z 3 etapów:

1. Uruchomienia zaimplementowanego przez nas algorytmu k-NN do znalezienia k najbliższych sąsiadów dla każdego przykładu,
2. wyliczenia wskaźnika nieprawidłowości,
3. na podstawie wartości wskaźnika klasyfikacja przykładu jako odstającego lub nie.

Cały algorytm będzie znajdował się w klasie *NNAnomalyDetector*

Parametry:

- k: int
  liczba sąsiadów
- metric: string | Callable
  miara niepodobieństwa; może być podana jako 1 z nazw ze słownika sklearn.metrics.pairwise.distance_metrics() albo jako konkretna funkcja
- outlier_factor: str
  nazwa wskaźnika nieprawidłowości; zostaną zaimplementowane wskaźniki: niepodobieństwo do k-tego sąsiada, średnie niepodobieństwo do k sąsiadów, osiągalność x1 z x2, lokalna gęstość otoczenia x, LOF

Metody:

- detect_anom(X, thresh = None)

Parametr thresh przyjmuje wartość punktu odcięcia, dla której przykład klasyfikowany jest jako anomalia. Jeżeli wartość nie jest podana to algorytm zwróci wektor wartości wskaźnika nieprawidłowości dla każdego przykładu. Jeżeli parametr thresh jest podany to algorytm zwróci wektor z informacją o tym, czy dany przykład jest anomalią

Dodatkowo zostanie zaimplementowana klasa do znajdowania k najbliższych sąsiadów – *KNN*

Parametry:

- k: int
  liczba sąsiadów
- metric: string | Callable
  miara niepodobieństwa; może być podana jako 1 z nazw ze słownika sklearn.metrics.pairwise.distance_metrics() albo jako konkretna funkcja

Metody:

- get_kNN_dist(X)
  Zwróci wartości miary niepodobieństwa dla wszystkich k najbliższych sąsiadów dla każdego przykładu z podanych danych. W sytuacji, w której kilka przykładów jest tak samo odległych od przykładu referencyjnego jak k-ty sąsiad, sąsiadów może być więcej niż k

Algorytmy i narzędzia

Projekt zostanie wykonany w języku Python z wykorzystaniem bibliotek takich jak: *Pandas*, *Numpy*, *scikit-learn*, *Matplotlib*. Zostaną również wykorzystane algorytmy: *IsolationForest* oraz *OneClassSVM* z *scikit-learn* oraz miary niepodobieństwa z pakietu: *sklearn.metrics.pairwise.distance_metrics*.

Plan badań

Eksperymenty

Podstawowym celem eksperymentów będzie rozstrzygnięcie, czy nasza implementacja algorytmu k-NN w zadaniu detekcji anomalii przewyższa jakość klasyfikacji jednoklasowej ww. gotowych modeli.

W tym celu wytrenujemy i przeprowadzimy ocenę podanych modeli. Biorąc pod uwagę niepewność uczenia maszynowego każdy eksperyment powtórzymy pięć razy przy różnych ziarnach, a następnie obliczymy średnią.

Podczas eksperymentów przebadamy wpływ hiperparametru *K – liczba sąsiadów* w algorytmie k-NN na wynik jego działania. Ponadto wyznaczymy wartości różnych miar niepodobieństwa i wskaźników nieprawidłowości oraz czas działania danej metody.

Ten wynik umieścimy w kontekście działania gotowych modeli klasyfikacji jednoklasowej z domyślnymi parametrami.

Zbiory danych

Jako źródła danych wykorzystamy podane niżej zbiory z repozytorium Outlier Detection Datasets. Zbiory do oceny jakości (Speech, Satelite, ForestCover) będą miały realistyczne rozmiary (więcej niż 1000 przykładów, przynajmniej 10 atrybutów o rozkładzie ciągłym).

Nazwa	Przykłady	Wymiarowość	Przykłady odstające	Komentarz

Wine dataset	129	13	10 (7,7 %)	Mały zbiór do prototypowania
Speech	3686	400	61 (1,65%)	Duża wymiarowość.
Satellite	6435	36	2036 (32%)	Stosunkowo duża liczba anomalii.
ForestCover	286048	10	2747 (0.9%)	Stosunkowo mała liczba anomalii.

Wstępne przygotowanie danych będzie obejmowało ich przeskalowanie, aby średnia wynosiła 0 a odchylenie standardowe 1.

Miary jakości i procedury oceny modeli

Porównanie algorytmów k-NN ze względu na różne wartości k przeprowadzimy przy użyciu **wartości pola pod krzywą ROC (AUC)** z uwagi na to, że wskaźnik nieprawidłowości na wyjściu będzie miał postać ciągłą jednowymiarową.

Następnie wskaźnik nieprawidłowości przekształcimy na wartość binarną przy użyciu punktu odcięcia (threshold). Optymalny punkt odcięcia to taki, który znajduje się najbliżej lewego górnego rogu wykresu (maksymalizując wartość AUC).

Na tej podstawie porównamy między sobą nasz model i gotowe klasyfikatory, używając **miary F1**, która uwzględnia zarówno precyzję, jak i czułość.