

Algorytm k-NN

Słowa kluczowe: uczenie maszynowe, uczenie nadzorowane, python

Opis

Implementacja aplikacji w języku Python, która dokona klasyfikacji przypadków przy użyciu algorytmu k-NN (ang. k Nearest Neighbours).

Wymagania obowiązkowe (max 3/6 pkt)

- Algorytm ma być zaimplementowany całkowicie własnoręcznie, zabronione jest korzystanie z gotowych funkcji obliczających k-NN, odległości lub testowanie (w tym podział na zbiór uczący i testowy).
- Należy zaimplementować obliczanie odległości w metryce manhattan (odległość taksówkowa) i euklidesowej.
- Program powinien wypisać na wyjście podjęte decyzje dla zbioru testowanych przypadków (domyślnie wszystkich ze zbioru uczącego) oraz *trafność klasyfikacji* uzyskaną na tym zbiorze.
- Należy przetestować działanie algorytmu (tzn. sprawdzić *trafność klasyfikacji* utworzonego klasyfikatora) dla przynajmniej dwóch zbiorów danych:
 - a) Iris (<https://archive.ics.uci.edu/ml/datasets/Iris>)
 - b) Wine (<https://archive.ics.uci.edu/ml/datasets/Wine>)Należy zwrócić uwagę na to, że w Iris atrybut decyzyjny jest ostatni, a w Wine pierwszy.
- Format danych wykorzystywany na UCI (linki powyżej) to CSV z przecinkiem jako separatorem.
- Aplikacja powinna na wejście przyjmować następujące argumenty linii poleceń (konsolka):
 - a) **-k** k w k-NN, domyślnie k=5
 - b) **-m** metryka ('euclides' or 'manhattan'), domyślnie 'euclides'
 - c) **-t** rodzaj zbioru testowego ('train', 'split', 'cross'), domyślnie 'train' (bo na 3.0 nie trzeba implementować pozostałych)
 - d) **-d** indeks atrybutu decyzyjnego (domyślnie ostatni)
 - e) **[file with data]** plik z danymi, obowiązkowy argument
- Przykłady uruchomienia:
`python3 knn.py iris.data`
`python3 knn.py -k 1 -t cross iris.data`
`python3 knn.py -k 3 -m euclides -t train -d 0 wine.data`
- Do implementacji obsługi linii poleceń należy wykorzystać bibliotekę *argparse*.

Wymagania dodatkowe cz. 1 (max 4.5/6 pkt)

- Należy zaimplementować podział przypadków na zbiór uczący i zbiór testowy przy użyciu percentage split (wprowadzić na to dodatkowy parametr, wartość domyślna 0.25, czyli 25%).
- Klasyfikator trzeba nauczyć na zbiorze uczącym, a następnie przetestować go na zbiorze testowym.

Wymagania dodatkowe cz. 2 (max 6/6 pkt)

- Należy zaimplementować 10-fold cross-validation i ocenić za jej pomocą wyniki algorytmu, uśredniając uzyskane trafności klasyfikacji.