Metody sztucznej inteligencji 2
Projekt 1. — Algorytm k najbliższych sąsiadów
Konspekt

Bartłomiej Dach, Tymon Felski

27 maja 2018

Niniejszy dokument zawiera informacje wstępne dotyczące pierwszego projektu, którego celem jest zaimplementowanie algorytmu k najbliższych sąsiadów (k-NN, ang. *k nearest neighbors*) i analiza jego działania dla dostarczonych danych treningowych.

1. Skład grupy

Projekt realizowany będzie w dwuosobowej grupie, w składzie:

1. Bartłomiej Dach,
2. Tymon Felski.

2. Opis algorytmu

Algorytm k najbliższych sąsiadów jest jedną z wielu metod rozwiązywania **problemu klasyfikacji**, czyli predykcji wartości zmiennych jakościowych (zwanych również zmiennymi kategorycznymi lub dyskretnymi) na podstawie przykładowych **danych treningowych** [4, s. 9–10]. Klasycznym przykładem tego typu problemu jest klasyfikacja gatunków irysów z użyciem pomiarów rozmiarów działek kielicha oraz płatków kwiatu.

Klasyfikatory oparte na metodzie k najbliższych sąsiadów operują na punktach w przestrzeni n-wymiarowej. Proces klasyfikacji danego punktu $x_0 \in \mathbb{R}^n$ odbywa się w następujący sposób:

1. Ze zbioru treningowego wybierane jest k punktów znajdujących się najbliżej punktu x_0 [5, s. 261]. Odległość między punktami definiowana jest najczęściej jako odległość euklidesowa w przestrzeni $\mathbb{R}^n$.
2. Jeżeli większość z wybranych k punktów należy do jednej klasy, to punkt wejściowy jest przypisywany do tej samej klasy.
3. Potencjalne remisy w punkcie (2) są rozstrzygane losowo [4, s. 463–464].

Liczba punktów wybieranych ze zbioru treningowego (wartość k) stanowi parametr algorytmu. Jakość działania metody w dużym stopniu zależy od odpowiedniego doboru tego parametru dla danego zadania klasyfikacji [4, s. 468–470].

Ponieważ algorytm k-NN wymaga zapamiętania całości danych treningowych do swojego działania, zaliczany jest on do klasy klasyfikatorów **opartych na pamięci** (ang. *memory-based*) [4, s. 463].

3. Wybrane technologie

Językiem programowania, który zdecydowano się wykorzystać, jest język skryptowy **Python** w wersji 3.5.2. Jest to uwarunkowane między innymi sporymi możliwościami tego języka w zakresie przetwarzania i analizy danych, wygodą programowania i przenośnością stworzonych rozwiązań. W celu uproszczenia pracy z danymi oraz ich analizy, wybrano także pomocnicze moduły Pythona, którymi są **pandas**, **NumPy** oraz **Matplotlib**.

W poniższej tabeli zestawiono wybrane biblioteki wraz z ich wersjami oraz określono licencje, na których zostały udostępnione.

Nr	**Komponent, wersja**	**Opis**	**Licencja**	
1	Matplotlib, 2.1.0	Umożliwia tworzenie wykresów	Matplotlib License	[1]
2	NumPy, 1.13.3	Używana do efektywnych obliczeń na wektorach n-wymiarowych	BSD License	[3]
3	pandas, 0.21.0	Wspomaga ładowanie danych z plików CSV oraz ich analizę	BSD License	[2]

Tablica 1: Wykorzystane biblioteki wraz z określeniem licencji

Literatura

[1] Matplotlib Development Team: Matplotlib. Oficjalna strona: `https://matplotlib.org`. [Dostęp 26 lutego 2018]

[2] McKinney W.: pandas – Python Data Analysis Library. Oficjalna strona: `https://pandas.pydata.org`. [Dostęp 26 lutego 2018]

[3] Oliphant T.: NumPy. Oficjalna strona: `http://www.numpy.org`. [Dostęp 26 lutego 2018]

[4] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Nowy Jork: Springer-Verlag, 2009. [Online]
Dostępne: `https://web.stanford.edu/~hastie/ElemStatLearn/`. [Dostęp 26 lutego 2018]

[5] V.N. Vapnik, *Statistical learning theory*. Nowy Jork: John Wiley and Sons, 1998.