

Metody sztucznej inteligencji 2

Projekt

Maciej Żelaszczyk

21 lutego 2018

Zakład Sztucznej Inteligencji i Metod Obliczeniowych
Wydział Matematyki i Nauk Informacyjnych

`m.zelaszczyk@mini.pw.edu.pl`

**Warsaw University
of Technology**

- Grupa 2.
- Środa, godz. 12:15, s. 201.
- Projekt: maksymalnie 60 punktów, zaliczenie od 30.
- Ćwiczenia: maksymalnie 40 punktów, zaliczenie od 20.
- Warunek dodatkowy: $\sum p \geq 51$.

- Dwa zadania projektowe.
- Projekt 1: krótkie zadanie na oswojenie z ideami, 15pkt.
- Projekt 2: dłuższe zadanie na stosowanie bardziej złożonych pomysłów, 45pkt.
- Realizacja w grupach dwuosobowych.

Harmonogram

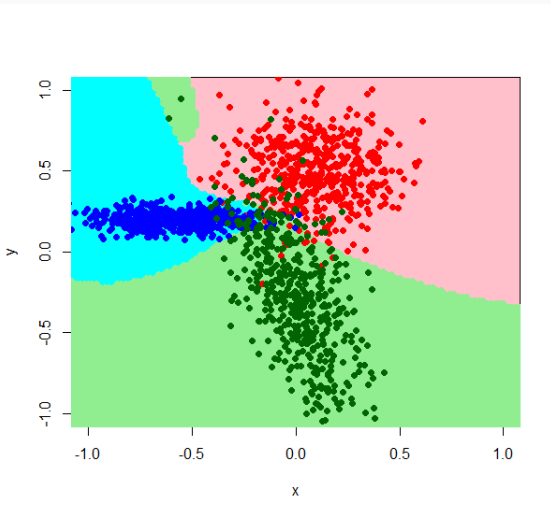
1. 21.02 - organizacyjne, wyjaśnienie Projektu 1.
2. 28.02 - konspekt do Projektu 1.
3. 07.03 - konsultacje.
4. 14.03 - zebranie Projektu 1. + raport + tematy do Projektu 2.
5. 21.03 - konspekt Projektu 2.
6. 28.03 - konsultacje.
7. 04.04 - konsultacje.
8. 11.04 - pierwsze wyniki Projektu 2.
9. 18.05 - konsultacje.
10. 25.04 - konsultacje.
11. 09.05 - kontrola prac nad Projektem 2.
12. 23.05 - konsultacje.
13. 30.05 - zaawansowane wyniki Projektu 2.
14. 06.06 - ostateczne wyniki Projektu 2 + raport.
15. 13.06 - prezentacje (ok. 10 minut) - jaki był Projekt 2, jakie wyniki.

Projekt 1

- Grupy dwuosobowe.
- Implementacja algorytmu k-nearest neighbors (k-NN) dla problemu klasyfikacji.
- Implementacja manualna, bez korzystania z bibliotek.
- Format zadany przez dostarczony plik.
- Podział train/test.
- Wybór parametrów algorytmu.
- Wybór wag do problemu (wagi równe, wagi proporcjonalne do odległości).
- Możliwość zmiany parametrów w trakcie testu.
- Wizualizacja wyniku.
- Jaki wpływ na wynik ma dobór k , wag, proporcji train/test?

Projekt 1

- Wizualizacja.



Projekt 1

- Oddanie konspektu:
 - Członkowie grupy.
 - Python/Java/C++/...?
 - Opis algorytmu k-NN.
 - Literatura.
 - Krótko.
- Oddanie projektu:
 - Raport z opisem implementacji, parametrów, wyników.
 - Kody użyte do implementacji, pliki wynikowe.

- Grupy dwuosobowe.
- Propozycje tematów przedstawione na czwartym spotkaniu.
- Możliwość wyboru własnego tematu.
- Konspekt:
 - Opis problemu badawczego.
 - Cel badania.
 - Planowane modele, pomysły, algorytmy.
 - Opis danych.
 - Metody weryfikacji rezultatów.

Projekt 2

- Implementacja.
- Wczytywanie zbioru danych (train) do aplikacji.
- Parametryzacja trenowanego modelu.
- Efektywne przetwarzanie zbioru lub równoległe przetwarzanie na różnych zestawach parametrów.
- Przetwarzanie danych testowych.
- Prezentacja wyników uzyskanych na danych testowych.

- Wyniki:
 - Opracowane statystycznie (średnie, mediany, odchylenia standardowe, itp.) rezultaty dla poszczególnych parametrów.
 - Komentarz wraz ze wskazaniem planowanych eksperymentów (z uzasadnieniem).
- Raport:
 - Opis problemu badawczego, cel badania oraz faktycznie wykorzystane techniki wraz z opisem dokonanych modyfikacji.
 - Instrukcja do aplikacji testowej wraz z przykładowymi zbiorami danych i opisem sposobu uruchomienia i uzyskania wyników.
 - Wybrane, opracowane statystycznie wyniki wraz z ich opisem.
 - Omówienie uzyskanych wyników, wskazanie sukcesów i problemów, propozycje dalszych prac badawczych.

- Do oddania:
 - Raport.
 - Aplikacja w formie kodu.
 - Dane.
 - Prezentacja.

k-nearest neighbors

- Rozważamy problem klasyfikacji.
- Bierzemy $(x_1, y_1), \dots, (x_n, y_n)$, gdzie $x_i \in \mathbb{R}^p, y_i \in \{0, 1, \dots, c\}$.
- p to wymiar przestrzeni danych, $c + 1$ to liczba klas.
- Miara odległości $d(x_i, x_j)$.
- Przykładowo, $d(x_i, x_j) = \|x_i - x_j\|$.
- Dla wybranego punktu x_i o klasie decydujemy przez głosowanie wśród k najbliższych mu punktów.
- Podstawowy schemat głosowania to liczebność.
- Możliwe inne schematy, np. ważone w oparciu o odległość.
- Jak zdecydować o klasach bez wiedzy o $(x_1, y_1), \dots, (x_n, y_n)$?
- Czy to jest efektywny algorytm?