

Wymagania do projektu Uczenie Maszynowe (UMA)

6 marca 2024

prowadzący: Stanisław Kozdrowski

Konsultacje: sroda, godz. 15-17, pokój 25a. Może być też w innym terminie po wcześniejszym uzgodnieniu mailem (stanislaw.kozdrowski@pw.edu.pl).

Ważne informacje:

1. Projekty realizowane są w Zespołach 2-osobowych, w wyjątkowych przypadkach, po uzgodnieniu z prowadzącym może to być Zespół 3-osobowy. Zapisów dokonujemy przy użyciu systemu zapisy (uwierzytelnianie wydziałowe). Po zalogowaniu należy wybrać przedmiot **Uczenie Maszynowe (UMA)**. System ułatwia również znalezienie drugiego członka zespołu (widać kto zapisał się samotnie). O przydziale tematu decyduje kolejność zgłoszeń.
2. Dane do większości eksperymentów można pobrać z Internetu, np. ze strony UCI lub Kaggle.
3. Implementacja programu w jednym z języków programowania - Python, R, ew. C++. Programy muszą działać pod kontrolą systemu Ubuntu Linux. Chciałbym zasugerować, aby tam, gdzie to możliwe, w części eksperymentalnej projektu dokonać także porównania z implementacjami algorytmów dostępnymi w środowiskach R lub Python.
4. Dokumentację wstępną oraz końcową proszę przysłać w formie elektronicznej.
5. Wszelka korespondencja e-mail związana z przedmiotem powinna mieć w temacie e-maila napis [UMA], po czym powinna wystąpić dalsza część tematu.
6. Konsultacje mogą się odbywać zarówno w formie stacjonarnej jak i zdalnej. Przed planowanymi konsultacjami proszę wysłać maila do propoadzającego projekt - a w przypadku wersji zdalnej również link do zoom'a lub teams.

Ważne terminy:

1. **do 08.03.2024:** projekty zostaną upublicznione
2. **do 26.03.2024:** ostateczny termin przydziału tematów do Zespołów
3. **do 14.05.2024:** przygotowanie wstępnej dokumentacji
4. **11.06.2024:** ostateczna data oddania projektu.

Projekt Wstępny powinien zawierać:

1. opis projektu, wskazujący na zrozumienie problemu
2. precyzyjny opis algorytmów, które będą wykorzystane
3. plan eksperymentów, który może się zmienić - nie musi być ostateczny
4. dokładnie opisane zbiory danych, które będą używane do badań.
5. **Dokumentację wstępną należy wysłać mailem na adres: stanislaw.kozdrowski@pw.edu.pl.**

Projekt Końcowy powinien zawierać:

1. krótkie streszczenie założeń z Projektu Wstępnego
2. pełen opis funkcjonalny
3. precyzyjny opis algorytmów oraz opis zbiorów danych
4. raport z przeprowadzonych testów oraz wnioski
5. opis wykorzystanych narzędzi, bibliotek, itp.

Oddawanie projektów: Zespół wysyła e-mail z załącznikiem Projektu Końcowego i czeka na ocenę lub na zaproszenie na rozmowę. Do pakowania Projektu Końcowego (wraz z Dokumentacją Końcową) należy używać formatu .zip (nie .rar)

Punktacja:

za Projekt można uzyskać maksymalnie 50 punktów:

- z czego za Projekt Wstępny 10p
- za implementację i testy 20p
- za badania i dokumentację 20p.

Tematy Projektów

umieszczone są na stronie: <http://assigner.ii.pw.edu.pl/>

Po zalogowaniu się należy wybrać przedmiot [UMA] oraz jeden z projektów.

Tematy Projektów:

SK.UMA.1 – Algorytm AQ ze zbiorem walidacyjnym

Zaimplementować zmodyfikowaną wersję algorytmu indukcji reguł AQ, która do oceny dodawanego w i -tej iteracji kompleksu stosuje osobny zbiór walidacyjny.

SK.UMA.2 – Heurystyczny algorytm eliminacji

Zaimplementować dwa heurystyczne algorytmy eliminacji nieistotnych atrybutów, uwzględniające zależność wyższych rzędów.

SK.UMA.3 – Drzewo klasyfikacyjne + ruletka

Przygotować algorytm konstruujący drzewo klasyfikacyjne (techniką zstępującą - 'top down'), w którym wybór testu dla dodawanego węzła drzewa odbywa się ruletkowo, tzn. każdy test proporcjonalną do swojej jakości szansą na bycie wybranym.

SK.UMA.4 – Szeregowanie zadań

Zaimplementować problem kombinatoryczny (szeregowanie zadań) w sposób umożliwiający zastosowanie do niego metod uczenia ze wzmocnieniem.

SK.UMA.5 – Drzewo regresji + ruletka

Przygotować algorytm konstruujący drzewo regresji (technika zstępująca - 'top down'), w którym wybór testu dla dodawanego węzła drzewa odbywa się ruletkowo, tzn. każdy test jest proporcjonalną do swojej jakości szansą na bycie wybranym.

SK.UMA.6 – Ranking atrybutów

Przygotować własny algorytm konstruujący drzewa klasyfikacyjne i zastosować go do przygotowania metody tworzącej ranking atrybutów. Ranking ma być tworzony przez wielokrotną konstrukcję drzewa na losowych podzbiorach zbioru uczącego oraz zliczenie częstości występowania poszczególnych atrybutów w testach.

SK.UMA.7 – Układanie rozkładu jazdy

Zaimplementować problem kombinatoryczny (układanie rozkładu jazdy) w sposób umożliwiający zastosowanie do niego metod uczenia ze wzmocnieniem.

SK.UMA.8 – Klasyfikacyjny las losowy

Zaimplementować zmodyfikowaną wersję algorytmu generowania lasu losowego, w której do generowania kolejnych drzew losowane są częściej elementy ze zbioru uczącego, na których dotychczasowy model się mylił.

SK.UMA.9 – Problem komiwojażera

Zaimplementować problem kombinatoryczny (problem komiwojażera) w sposób umożliwiający zastosowanie do niego metod uczenia ze wzmocnieniem.

SK.UMA.10 – Regresyjny las losowy

Zaimplementować zmodyfikowaną wersję algorytmu generowania lasu losowego regresji, w której do generowania kolejnych drzew losowane są częściej elementy ze zbioru uczącego, na których dotychczasowy model popełniał większe błędy.

SK.UMA.11 – Inkrementacyjna indukcja drzew

Zaimplementować inkrementacyjną indukcję drzew decyzyjnych. Przebudowa drzewa na podstawie sekwencyjnie nadchodzących porcji danych lub pojenynych przykładów.

SK.UMA.12 – Las losowy i naiwny klasyfikator bayesa

Zaimplementować zmodyfikowaną wersję algorytmu generowania lasu losowego, w której zamiast drzew składowymi są naiwne klasyfikatory bayesa.

SK.UMA.13 – Inkrementacyjna indukcja reguł

Zaimplementować inkrementacyjną indukcję reguł. Przebudowa zbioru reguł na podstawie sekwencyjnie nadchodzących porcji danych lub pojedynczych przykładów.

SK.UMA.14 – Wahadło

Przy pomocy technik uczenia ze wzmocnieniem (bazujących na algorytmie Q-Learning) przygotować agenta sterującego [wahadłem]

SK.UMA.15 – Drzewa i lasy losowe

Dowolny problem z wskazanych baz.

SK.UMA.16 – Kolejka górską

Przy pomocy technik uczenia ze wzmocnieniem (bazujących na algorytmie Q-Learning) przygotować agenta sterującego [wagonikiem].

SK.UMA.17 – Funkcja ucząca

Funkcja ucząca i funkcja kosztu dla modeli regresyjnych. Przykłady implementacji.

SK.UMA.18 – IRep++

Przygotować implementację algorytmu do indukcji reguł [IRep++] oraz przeprowadzić eksperymenty na wybranym zadaniu z bazy.

SK.UMA.19 – Techniki oceny klasyfikacji

Zaimplementuj techniki oceny klasyfikacji dla zestawów danych dotyczących raka piersi, które dostępne są w <http://archive.ics.uci.edu/ml/datasets>

SK.UMA.20 – Estymacja jakości sieci

Przygotować implementację programu do estymacji jakości sieci teleinformatycznej za pomocą wybranych kilku modeli UM. Na podstawie literatury: DOI:10.1117/12.2536656 i DOI:10.3390/e23010007. Baza danych: <http://sndlib.zib.de/home.action>