

Systemy uczące się – Lab 4

dr Przemysław Juszczuk

Katedra Inżynierii Wiedzy, Uniwersytet Ekonomiczny

26 X 2018

Projekt zaliczeniowy

Podstawą zaliczenia ćwiczeń jest indywidualne wykonanie projektu uwzględniającego trzy aspekty: opracowanie teoretyczne zagadnienia, przygotowanie aplikacji w dowolnym języku, przeprowadzenie eksperymentów i wizualizacja wyników.

Projekt zaliczeniowy – opracowanie teoretyczne

Opracowanie teoretyczne powinno zawierać krótki opis oraz zasadę działania wybranej metody. W szczególności:

- wskazanie pozycji bibliograficznej / stron internetowych, na podstawie których wykonano opracowanie;
- podstawowe założenia metody;
- próbę zapisu formalnego wraz z opisem (całej metody), lub wybranego elementu. Np. dla sieci neuronowej zapis formalny może dotyczyć funkcji aktywacji neuronu;
- pseudokod wraz z opisem – przy czym pseudokod może dotyczyć już bezpośrednio rozwiązywanego zadania (jeżeli celem jest klasyfikacja konkretnego zbioru danych, to pseudokod może dotyczyć właśnie tego zbioru).

Aplikacja może zostać przygotowana w dowolnym języku. Nie należy jednak stosować żadnych bibliotek wbudowanych pozwalających na realizację całego zadania poprzez wywołanie konkretnej metody – przykładowo biblioteka umożliwiająca klasyfikację zbioru danych przy pomocy drzewa decyzyjnego, gdzie wejściem jest zbiór danych, a wyjściem klasyfikator.

- na podstawie pseudokodu opracowanego w części teoretycznej należy przygotować poszczególne metody i funkcje, które następnie posłużą do budowy wybranego narzędzia (na przykład klasyfikatora);
- nie jest narzucona żadna metoda – w szczególności nie jest konieczne stosowanie paradygmatu obiektowego;
- celem jest poznanie zasady działania wybranej metody, a nie traktowanie jej jako tzw. "black box";
- można korzystać z bibliotek wbudowanych, kiedy potrzebne jest wykonanie podstawowych operacji na danych takich jak: sortowanie, wyznaczenie średniej, wyznaczenie mediany;
- aplikacja może działać w trybie konsolowym, przy czym konieczne jest opracowanie wyników (patrz część trzecia projektu).

Projekt zaliczeniowy – część eksperymentalna

W części trzeciej należy pokazać, że przygotowana aplikacja opracowana na podstawie pseudokodu z części pierwszej działa. W tym celu należy przeprowadzić przykładowe zadanie klasyfikacji / predykcji (lub inne wskazane w projekcie) na podstawie wybranego zbioru danych. Zbiór danych może być dowolny, spreparowany na potrzeby zadania. Można też skorzystać z repozytorium dostępnego w internecie.

Wymagania

- ocena dostateczna – przedstawiony został podstawowy opis teoretyczny wraz z ogólnym pseudokodem. Pominięto zagadnienie dotyczące zapisu formalnego. Kod jest nieczytelny, brak jest komentarzy (wystarczy komentować metody, nie jest konieczne opisywanie każdej zmiennej), a fragmenty kodu noszą znamiona tak zwanego god object, gdzie szereg funkcjonalności przypisanych jest do jednej klasy / jednej metody. W części eksperymentalnej wykorzystano prosty, spreparowany zbiór danych, pominięto zagadnienie preprocessingu danych a wizualizacja części eksperymentalnej nie uwzględnia symulacji krokowej.
- ocena dobra – część teoretyczna uwzględnia krótkie wprowadzenie, próby zapisu formalnego wybranych fragmentów systemu (np. zdefiniowany jest graf, czy też pojedynczy neuron w sieci). Pseudokod jest dokładnie opisany i wyjaśniony. W części praktycznej fragmenty kodu odpowiadające za wybrane elementy systemu są od siebie wyraźnie oddzielone, kod zawiera komentarze. W części eksperymentalnej przeprowadzono prostą symulację krokową.

Wymagania

- ocena bardzo dobra – uwzględnienie wymagań na ocenę dobrą z zapisem formalnym metody, przygotowaniem prostego przykładu (w części teoretycznej). Wykonanie części praktycznej, w której dodatkowo uwzględnione są procedury związane z preprocessingiem: fuzyfikacją, uzupełnieniem danych niepełnych i dyskretyzacją. Możliwość przeprowadzenia symulacji krokowej. Przedstawienie wybranych wyników w formie wykresów wraz z krótkim komentarzem.

- sieć Kohonena (mapa samoorganizująca się) – celem projektu jest opracowanie aplikacji umożliwiającej klasyfikację danych wejściowych. Mechanizm bazuje na uczeniu nienadzorowanym, gdzie początkowy zestaw obiektów opisanych przy pomocy n atrybutów pozwala nauczyć sieć neuronową. Tak przygotowana sieć pozwala klasyfikować kolejne dane przychodzące do systemu w taki sposób, że ich położenie przedstawione jest na płaszczyźnie dwuwymiarowej. Zatem oryginalne obiekty o n atrybutach rzutowane są na płaszczyznę dwuwymiarową.
- algorytm ID3 – celem projektu jest przygotowanie aplikacji umożliwiającej budowę drzewa decyzyjnego (a następnie przetestowanie jego skuteczności jako klasyfikatora) na podstawie algorytmu ID3. Drzewo decyzyjne budowane jest na podstawie atrybutów dających największy przyrost informacji. W celu wyznaczenia kolejnych atrybutów, względem których budowane są kolejne fragmenty drzewa. W każdej iteracji należy wskazać atrybut o najmniejszej entropii. W kolejnej iteracji analizowany jest zbiór pomniejszony o ten atrybut.

- rozmyta sieć kognitywna – celem projektu jest przygotowanie aplikacji umożliwiającej (dla zadanego zestawu atrybutów) zbudowanie rozmytej sieci kognitywnej. Dla obiektów opisanych maksymalnie pięcioma atrybutami budowana jest sieć bazująca na grafie pełnym. W przeciwnym wypadku losowo wybierane jest $n \cdot (n - 1)$ krawędzi pomiędzy wierzchołkami, które uznane będą jako istotne (wskazane przez eksperta). Następnie, na tak zbudowanej sieci kognitywnej na podstawie zestawu danych (po wcześniejszym ewentualnym uzupełnieniu danych niepełnych) należy przeprowadzić proces uczenia adaptacyjnego. Testowanie powinno odbywać się na 10 ostatnich odczytach – to znaczy dla zestawu danych zawierających 100 następujących po sobie odczytów uczenie przebiega na 90 pierwszych odczytach. Dla 10 ostatnich odczytów następuje symulacja działania sieci, gdzie wartości pojęć wyznaczone są na podstawie iloczynów pojęć i wag wpływających na te pojęcia. Wreszcie następuje ocena – porównanie wartości pojęć uzyskanych na drodze testowania oraz rzeczywistych wartości pojęć z danych.

- klasyfikator k -nn – celem projektu jest przygotowanie aplikacji umożliwiającej klasyfikację obiektów opisanych n atrybutami liczbowymi przy pomocy algorytmu k -nn dla dowolnego k (w tym dla $k = 1$). Klasyfikator ma umożliwiać wyznaczenie odległości pomiędzy obiektami na podstawie odległości euklidesowej, manhattan oraz Czebyszewa. Każdy nowy obiekt w systemie klasyfikowany jest na podstawie klasy decyzyjnej przypisanej do jego k najbliższych sąsiadów.
- algorytm ewolucyjny – celem projektu jest przygotowanie aplikacji bazującej na dowolnej wersji algorytmu ewolucyjnego (lub populacyjnego). Populacja ma zawierać n osobników, gdzie każdy osobnik (genotyp osobnika) składa się z dwóch współrzędnych. Funkcja oceny umożliwia wyznaczenie wartości fenotypu osobnika oraz wskazanie rozmieszczenia osobników w przestrzeni. Celem jest wyznaczenie ekstremum funkcji (minimum, lub maksimum). Aplikacja ma działać na trzech dowolnych funkcjach (lub alternatywnie do aplikacji ma zostać dołączony parser umożliwiający wprowadzenie własnej funkcji).