Instrukcja użytkowania programu FuzzyDataMiner 1.0

Spis treści

1. Wprowadzenie

Program FuzzyDataMiner powstał w ramach implementacji metod analizy danych opisanych w rozprawie habilitacyjnej dr Urszuli Bentkowskiej i jest dostępny w sieci INTERNET. Informacje na jego temat, jak i samo oprogramowanie, można pozyskać ze strony:

http://diagres.ur.edu.pl/~fuzzydataminer/

Głównym celem tego oprogramowania jest zaprezentowanie dwóch zastosowań praktycznych metod opisanych w rozprawie. Pierwsze z nich dotyczy klasyfikowania obiektów testowych z wykorzystaniem klasyfikatora skonstruowanego metodą k-NN w sytuacji, gdy w obiekcie testowym występują wartości nieokreślone (missing values). Natomiast drugie stanowi propozycje konstrukcji klasyfikatora dla danych mikromacierzowych, które cechują się bardzo dużą liczbą atrybutów i niewielką liczbą obiektów.

2. Aspekty techniczne konstrukcji programu

Program został zaimplementowany w języku Java. Dlatego do poprawnego działania niezbędne jest zainstalowanie pakietu oprogramowania Java SE Development Kit 8 lub przynajmniej Java SE Runtime Environment 8 firmy Oracle.

Dla przetestowania programu FuzzyDataMiner z wymienionej wyżej strony WWW należy pobrać archiwum zip o nazwie fuzzydataminer.zip

Po rozpakowaniu tego pliku na dysku pojawi się katalog fuzzydataminer zawierający następujące pliki:

- fuzzydataminer.jar - uruchamialne archiwum jar programu FuzzyDataMiner,

- biodeg.tab - przykładowy zbiór danych do przetestowania metody z parametrem -S (patrz niżej),
- biodeg_config.txt - przykładowa konfiguracja 3 eksperymentów do przetestowania metody z parametrem -S (patrz niżej),
- leukemia.tab - przykładowy zbiór danych do przetestowania metody z parametrem -M (patrz niżej),
- leukemia_config.txt - przykładowa konfiguracja 2 eksperymentów do przetestowania metody z parametrem -S (patrz niżej),
- start_s.bat - plik wsadowy do uruchomienia eksperymentu z metodą -S (plik ten może być uruchomiony z okna managera plików Windows bez użycia terminala),
- start_m.bat - plik wsadowy do uruchomienia eksperymentu z metodą -M (plik ten może być uruchomiony z okna managera plików Windows bez użycia terminala).

W katalogu fuzzydataminer pojawi się także podkatalog lib zawierający bibliotekę WEKA API w wersji 3.8, z której korzysta program fuzzydataminer.

Ponadto z wymienionej wyżej strony można pobrać dwa zestawy zbiorów danych używanych do eksperymentów opisanych w rozprawie. Pierwszy zbiór o nazwie **SDataSets.zip** to zestaw danych do eksperymentów z dodawaniem wartości missing. Natomiast drugi, o nazwie **MDataSets.zip** to zestaw danych mikromacierzowych.

3. Korzystanie z programu

Wykorzystanie programu odbywa się poprzez uruchomienie za pomocą maszyny wirtualnej języka Java archiwum jar o nazwie **fuzzydataminer.jar** z odpowiednimi opcjami. Ponieważ wykorzystanie programu wymaga podawania opcji, wygodne jest uruchamianie programu bezpośrednio w terminalu tekstowym systemu lub za pomocą odpowiednich nakładek na system operacyjny. Dla prostego testu można także wykorzystać pliki wsadowe start_s.bat i start_m.bat (patrz wyżej).

Ogólnie są dwa następujące sposoby wykorzystania programu do eksperymentów.

1. Wykorzystanie programu do wykonywania eksperymentów w zakresie testowania klasyfikatora k-NN na danych testowych mających wartości missing. Ogólny sposób wywołania programu dla tego przypadku jest następujący:

   java -jar fuzzydataminer.jar -S data.tab config.txt results.txt

Gdzie znaczenie parametrów jest następujące:

- java - fuzzydataminer.jar - uruchomienie jara fuzzydataminer.jar,

- -S - opcja mówiąca o pierwszym sposobie wykorzystania programu (testowanie obiektów z wartościami missing),
- data.tab - nazwa pliku z danymi, który podczas eksperymentu jest dzielony na część treningową i testową,
- config.txt - nazwa pliku z konfiguracją obliczeń,
- results.txt - nazwa pliku z wynikami obliczeń.

2. Wykorzystanie programu do wykonywania eksperymentów w zakresie testowania klasyfikatora k-NN na danych mikromacierzowych. Ogólny sposób wywołania programu dla tego przypadku jest następujący:

java -jar fuzzydataminer.jar -M data.tab config.txt results.txt

Gdzie znaczenie parametrów jest podobne jak w poprzednim przypadku:

- java - fuzzydataminer.jar - uruchomienie jara fuzzydataminer.jar,
- -M - opcja mówiąca o drugim sposobie wykorzystania programu (analiza danych mikromacierzowych),
- data.tab - nazwa pliku z danymi, który podczas eksperymentu jest dzielony zgodnie z zasadami metody leave-one-out,
- config.txt - nazwa pliku z konfiguracją obliczeń,
- results.txt - nazwa pliku z wynikami obliczeń.

4. Pliki konfiguracyjne

Jak opisano w poprzednim podrozdziale, do poprawnego wykorzystania programu, oprócz podania poprawnych parametrów wywołania, trzeba jeszcze przygotować plik konfiguracyjny, którego nazwa jest drugim parametrem programu. Plik konfiguracyjny jest plikiem tekstowym, którego struktura wygląda w ten sposób, że w każdej linii tego pliku opisane są parametry pojedynczego eksperymentu. Podczas obliczeń program wykonuje kolejno eksperymenty i ich wyniki zapisuje do pliku o nazwie będącej ostatnim parametrem wywołania programu.

Każdy z eksperymentów opisywany jest za pomocą ciągu parametrów, przy czym każdy z parametrów podawany jest w formacie:

nazwa_parametru=wartość_parametru

Kolejność parametrów jest dowolna, jednak konieczne jest, aby podać wszystkie wymagane parametry do danego eksperymentu. Opisy parametrów są oddzielane spacją.

Zestawy parametrów do obydwu typów eksperymentów (opcja -S i opcja -M) znacząco różnią się. Dlatego poniżej opisujemy dokładnie obydwa zestawy parametrów.

Obliczenia z parametrem -S

W przypadku wywołania programu z parametrem -S, konfiguracja każdego eksperymentu powinna zawierać następujące parametry:

- Partition - to parametr informujący, w jakiej proporcji podczas eksperymentu ma być dzielona tablica danych na część treningową i testową, np. parametr: Partition=0.6 oznacza, że tablica treningowa będzie liczyć 60% obiektów, a testowa 40% obiektów.
- Missing - to parametr informujący, jaki procent wybranych losowo wartości atrybutów w danych testowych zostanie podczas eksperymentu podmieniona na wartość MISSING, np. parametr Missing=0.3 oznacza, że 30% wartości będzie podmieniona.
- Decision - to parametr informujący o głównej wartości decyzji atrybutu decyzyjnego, np. parametr Decision=[RB] oznacza, że główna wartością decyzji jest wartość RB (nawiasy kwadratowe dodaje się z powodów technicznych dla ułatwienia parsowania tej wartości)
- Repeat - to parametr informujący o tym, ile razy eksperyment ma być powtórzony celem uzyskania średniej wartości wyniku oraz odchylenia standardowego, np. parametr Repeat=10 oznacza, że zostanie wykonane 10 powtórzeń.
- Method - to parametr informujący czy i jaka jest używana metoda agregacji klasyfikatorów; w szczególności wartość C oznacza, że jest to pojedynczy klasyfikator (bez agregacji), wartość M oznacza agregację za pomocą średniej arytmetycznej oraz wartość F oznacza agregację przedziałów niepewności.
- k - to parametr informujący o liczbie sąsiadów w metodzie k-NN; parametr ten jest używany jedynie w metodzie C, bo w metodzie M i F agregowana jest informacja pochodząca z klasyfikatorów z różnych k.
- MonteCarlo - to parametr informujący o liczbie losowo wybranych obiektów wykorzystywanych do obliczenia przedziału niepewności w metodzie F.
- FMeasure - to parametr informujący o sposobie agregacji w metodzie F (6 wartości ponumerowanych od 1 do 6).

Oto przykładowa konfiguracja w przypadku wywołania programu z parametrem -S:

Partition=0.6 Missing=0.1 Decision= [RB] Repeat=10 Method=F MonteCarlo=10 FMeasure=3

Jest to eksperyment przy podziale wejściowych danych w proporcji 60% do 40%, wypełnieniem wartością Missing 10% wartości atrybutów w tablicy testowej, wartością główną atrybutu decyzyjnego RB, dziesięciokrotnym powtarzaniem eksperymentu, metodą F agregacji niepewności opartej na agregacji przedziałów niepewności oraz sposobie numer 3 agregacji w metodzie F.

Po wykonaniu eksperymentu z parametrem -S, w pliku wynikowym zostaną zapisane wyniki w następujący sposób. W pierwszym wierszu pliku zapisywane są oddzielone od siebie etykiety parametrów konfiguracyjnych wraz z etykietami obliczonych wyników, przy czym są dwie etykiety wyników: AUC i STDDEV (AUC oznacza średnie accuracy, zaś STDDEV to odchylenie standardowe accuracy). Następnie w kolejnych wierszach zapisywane są wyniki

poszczególnych eksperymentów. Na przykład dla sformułowanych wyżej parametrów w pliku wynikowym zostanie wypisany następujący tekst.

Partition;Missing;Decision;Repeat;Method;k;MonteCarlo;FMeasure;AUC;STDDEV
0.6;0.1;RB;10;F;0;5;2;0.807;0.014

Ostatnie dwa parametry (0.807 i 0.014) oznaczają odpowiednio średnie accuracy i odchylenie standardowe. Zauważmy, że k ma tutaj wartość 0 i nie wnosi nic do eksperymentu, a występuje jedynie w celu ujednolicenia pliku do formatu CSV, co pozwala wczytywać wyniki np. do arkusza kalkulacyjnego w celu ich dalszej analizy.

Obliczenia z parametrem -M

W przypadku wywołania programu z parametrem -M, konfiguracja każdego eksperymentu powinna zawierać następujące parametry:

- Decision1 - to parametr informujący o głównej wartości decyzji atrybutu decyzyjnego, np. parametr Decision=[Cancer] oznacza, że główna wartością decyzji jest wartość Cancer.
- Decision2 - to parametr informujący o podrzędnej wartości decyzji atrybutu decyzyjnego, np. parametr Decision=[Normal] oznacza, że podrzędną wartością decyzji jest wartość Normal.
- Repeat - to parametr informujący o tym, ile razy eksperyment ma być powtórzony celem uzyskania średniej wartości wyników oraz odchylenia standardowego, np. parametr Repeat=10 oznacza, że zostanie wykonane 10 powtórzeń.
- NoClassHoriz - to parametr informujący o tym, ile jest agregowanych przedziałów niepewności.
- NoClassVert - to parametr informujący o tym ile klasyfikatorów k-NN jest używana do obliczenia danego przedziału niepewności.
- Method - to parametr informujący czy i jaka jest używana metoda agregacji klasyfikatorów; w szczególności: wartość M oznacza agregację za pomocą średniej arytmetycznej oraz wartość F oznacza agregację przedziałów niepewności.
- k - to parametr informujący o liczbie sąsiadów w metodzie k-NN.
- FMeasure - to parametr informujący o sposobie agregacji w metodzie F (6 wartości ponumerowanych od 1 do 6).

Oto przykładowa konfiguracja w przypadku wywołania programu z parametrem -M:

Decision1=[Cancer] Decision2=[Normal] Repeat=20 NoClassHoriz=5 NoClassVert=10 Method=F k=5 FMeasure=6

Jest to eksperyment, gdzie wartością główną atrybutu decyzyjnego jest Cancer a podrzędną Normal, 20-krotnym powtarzaniem eksperymentu, przy agregacji 5 przedziałów niepewności, przy czym każdy przedział jest generowany z użyciem 10 klasyfikatorów k-NN, gdzie k=5, przedziały niepewności są agregowane metodą F, typem agregacji numer 3.

Po wykonaniu eksperymentu z parametrem -M w pliku wynikowym zostaną zapisane wyniki w następujący sposób. W pierwszym wierszu pliku zapisywane są oddzielone od siebie etykiety parametrów konfiguracyjnych wraz z etykietami obliczonych wyników, przy czym jest 6 etykiet wyników: ACC;SD_ACC;SEN;SD_SEN;SPEC;SD_SPEC (ACC oznacza średnie accuracy wraz z jego odchyleniem standardowym SD_ACC, SEN oznacza średnią czułość wraz z jej odchyleniem standardowym SD_SEN, SPEC oznacza średnią specyficzność wraz z jej odchyleniem standardowym SD_SPEC).

Następnie w kolejnych wierszach zapisywane są wyniki poszczególnych eksperymentów. Na przykład dla sformułowanych wyżej parametrów w pliku wynikowym zostanie wypisany następujący tekst.

Decision1;Decision2;Repeat;NoClassHoriz;NoClassVert;Method;k;FMeasure;ACC;SD_ACC;SEN;SD_SEN;SPEC;SD_SPEC
[Tumor];[Normal];20;5;10;F;5;6;0.574;0.056;0.583;0.064;0.559;0.057

Ostatnie sześć parametrów (**0.574, 0.056, 0.583, 0.064, 0.559, 0.057**) oznaczają odpowiednio średnie accuracy i jego odchylenie standardowe, średnią czułość i jej odchylenie standardowe oraz średnią specyficzność i jej odchylenie standardowe.

W przypadku, gdy zestaw parametrów nie jest zgodny z powyższym opisem lub pojawią się jakieś problemy techniczne z wykonaniem programu (np. niepoprawne nazwy plików) działanie programu jest przerwane z odpowiednim komunikatem. Wtedy należy poprawić parametry i uruchomić program jeszcze raz.

5. Format pliku z danymi

W programie FuzzyDataMiner wczytywane są zbiory danych z plików tekstowych w następującym formacie. W pierwszym wierszu pliku, po słowie ATTRIBUTES, podawana jest liczba atrybutów (kolumn) w zbiorze danych. W kolejnych wierszach podawany jest opis typów wartości, jakie mogą się pojawić w zbiorze danych, przy czym informacja ta podawana jest następująco. Najpierw nazwa atrybutu, a później po spacji typ wartości atrybutu w postaci jednego z trzech następujących słów: **double** (typ zmiennoprzecinkowy, inaczej ułamek dziesiętny), **integer** (typ całkowity) lub **nominal** (typ tekstowy). Po podaniu typów atrybutów, po słowie OBJECTS podawana jest liczba obiektów. Wreszcie, w kolejnych wierszach podawane są obiekty poprzez podanie wartości wszystkich atrybutów w tej samej kolejności, co przy prezentacji ich typów.

Oto przykład prostego pliku z danymi:

ATTRIBUTES 3
weight double
growth integer
overweight nominal

OBJECTS 3
95 176 yes
65 180 no
85 160 yes

W zbiorze danych mamy 3 atrybuty. Pierwszy o nazwie **weight** jest atrybutem o wartościach zmiennoprzecinkowych. Drugi o nazwie **growth** jest atrybutem o wartościach całkowitych. Wreszcie trzeci, o nazwie **overweight** to atrybut o wartościach tekstowych. Po opisie atrybutów następuje opis 3 obiektów poprzez podanie wartości atrybutów.