



WYŻSZA SZKOŁA INFORMATYKI

I ZARZĄDZANIA

Z SIEDZIBĄ W RZESZOWIE

Przedmiot: Sztuczna Inteligencja

Dokumentacja Inicjowania Projektu (PRINCE2)

Algorytm LEM2

Prowadzący: mgr inż. Rafał Niemiec

Osoby realizujące projekt:

ZESPÓŁ C:

(1 - Kierownik) Khrystyna Mylkus 53425

(2) Viktor Kozenko 53406

(3) Mykola Rudnytskyi 53450

(4) Yevhen Reminetskyi 53448

(5) Ihor Shchupliak 53452

(6) Andrzej Bogacz 53681

(7) Dmytro Khomenko 53395 (IOS)

Rzeszów 2017

Spis treści

Cel	3
Skład	3
Ogólną postać algorytmu LEM2	3
Charakterystyka algorytmu LEM2	3
Opis działania algorytmu Algorytm LEM2	4
Definicja projektu	4
Sekcje zmienne.....	5
Struktura zespołu	5
Harmonogram	5
Określenie celu biznesowego	6
Funkcjonalne	6
Niefunkcjonalne–	6
Kryteria jakości	6
Bibliografia	7

Cel

Cel projektu polega na implementacji oraz analizie jednego z najbardziej znanych i skutecznych algorytmów indukcji reguł, zaproponowanego przez J.W. Grzymała-Busse. Algorytm **LEM2** pozwala generować reguły również w przypadku zbiorów sprzecznych. Wykorzystuje w tym celu teorię zbiorów przybliżonych. Algorytm **LEM2** tworzy minimalny opis dyskryminujący przybliżenia danej klasy decyzyjnej za pomocą specyficznej zasady generowania kolejnych pokryć. Algorytm ten został zaimplementowany w systemie uczenia maszynowego **LEERS** (*Learning from Examples based on Rough Sets*).

Skład

Ogólną postać algorytmu LEM2

Algorytm 1: LEM2

Data: zbiór X ;
Result: pojedyncze lokalne pokrycie \mathcal{T} zbioru X ;
begin
 $G := X; \mathcal{T} := \emptyset$;
 while $G \neq \emptyset$ **do**
 $T := \emptyset; T_G := \{t : [t] \cap G \neq \emptyset\}$;
 while $T = \emptyset$ **or** $[T] \not\subseteq X$ **do**
 wybierz $t \in T_G$ takie, że $|[t] \cap G|$ jest maksymalne; jeżeli jest więcej niż jedno t spełniające warunek, to wybierz spośród nich pierwsze znalezione t takie, że $|[t]|$ jest minimalne;
 $T := T \cup \{t\}; G := [t] \cap G; T_G = \{t : [t] \cap G \neq \emptyset\} \setminus T$;
 foreach $t \in T$ **do**
 if $[T \setminus \{t\}] \subseteq X$ **then** $T := T \setminus \{t\}$;
 $\mathcal{T} := \mathcal{T} \cup \{T\}; G := X \setminus \bigcup_{T \in \mathcal{T}} [T]$;
 foreach $T \in \mathcal{T}$ **do**
 if $\bigcup_{T' \in \mathcal{T} \setminus \{T\}} [T'] = X$ **then** $\mathcal{T} := \mathcal{T} \setminus \{T\}$;
 end

Rys. 1 Algorytm LEM2

Charakterystyka algorytmu LEM2

Oznaczenia i definicje:

- X – niepusta dolna lub górna aproksymacja klasy decyzyjnej;
- $t = (a, v)$ – warunek, gdzie a – atrybut warunkowy, v – wartość przyjmowana przez a ;
- $[t]$ – blok, tj. zbiór obiektów spełniających warunek $t = (a, v)$;

- T - zbiór warunków t ;
- $[T] = \{t \in T \mid t \text{ spełnia każdy z warunków } t \in T\}$ - zbiór obiektów spełniających każdy z warunków $t \in T$;
- Zbiór X zależy od zbioru T wtedy i tylko wtedy, gdy $\emptyset \neq [T] \subseteq X$;
- T jest minimalnym kompleksem zbioru X wtedy i tylko wtedy, gdy X zależy od T i nie istnieje $T_0 \subset T$ takie, że X zależy od T_0 ;
- T - rodzina zbiorów T ;
- T jest lokalnym pokryciem zbioru X wtedy i tylko wtedy, gdy: 1. Każdy zbiór $T \in T$ jest minimalnym kompleksem zbioru X ; 2. $\bigcup_{T \in T} T = X$; 3. T jest minimalny, tj. składa się z najmniejszej możliwej liczby zbiorów T .

Opis działania algorytmu Algorytm LEM2

Opis działania algorytmu Algorytm LEM2 na wejściu otrzymuje aproksymację rozpatrywanej klasy decyzyjnej. Jeżeli jest to dolna aproksymacja, to są generowane reguły pewne, jeżeli górna, to możliwe (Zamiast aproksymacji można rozpatrywać obszar brzegowy, wtedy generowane są reguły przybliżone). Algorytm przy generowaniu każdej reguły bierze pod uwagę tylko te warunki, które są spełnione przynajmniej przez jeden obiekt z rozpatrywanego zbioru (tj. $TG := \{t : [t] \cap G \neq \emptyset\}$). W każdym kroku generowania reguły (zbiór T reprezentuje regułę) wybierany jest taki warunek, który jest spełniany przez największą liczbę obiektów (tj. $|[t] \cap G|$ jest maksymalne). Jeżeli jest więcej takich warunków spełniających podane kryterium, to spośród nich wybierany jest ten, który jest spełniony przez najmniejszą liczbę wszystkich obiektów z U (tj. $|[t]|$ jest minimalne). W tym przypadku jest to równoważne temu, że taki warunek jest spełniony przez najmniejszą liczbę obiektów spoza zbioru G . Reguła jest tworzona (tj. $T := T \cup \{t\}$), dopóki nie jest dodany ani jeden warunek do reguły (tj. $T = \emptyset$) lub te warunki które już są dodane do reguły są spełniane nie tylko przez obiekty z rozpatrywanej aproksymacji (tj. $[T] \subseteq X$). Po wygenerowaniu każdej reguły, następuje jej przycinanie poprzez usunięcie.

Sekcje stabilne

Definicja projektu

1. Zaimplementować algorytm LEM2.
2. Zaimplementować klasyfikator .
3. Za pomocą algorytmu LEM2 wygenerować dla każdej klasy decyzyjnej reguły decyzyjne z 2/3 losowo wybranych obiektów zbioru iris.tab.

4. Za pomocą klasyfikatora dokonać klasyfikacji pozostałych obiektów zbioru iris.tab z strony: <http://archive.ics.uci.edu/ml/>

Ad 1. 1. Zaimplementować funkcję wyznaczającą $[t] \cap G$ dla dowolnego warunku t i dowolnego zbioru G obiektów. 2. Zaimplementować funkcję wyznaczającą zbiór $TG = \{t : [t] \cap G \neq \emptyset\}$. 3. Zaimplementować funkcję, która wybierze najlepszy warunek do dodania do reguły.

4. Zaimplementować funkcję usuwającą zbędny warunek z reguły.

5. Zaimplementować funkcję usuwającą zbędną regułę ze zbioru reguł.

Ad 2. Klasyfikatora przypisać każdy obiekt do klasy tej reguły, którą obiekt spełnia. Jeżeli występuje konflikt klas, tzn. obiekt spełnia reguły dotyczące różnych klas, to jest on przypisywany do klasy tych reguł, których spełnia więcej. W przypadku, gdy obiekt spełnia tyle samo reguł każdej ze spornych klas, to nie jest on przypisywany do żadnej klasy.

Sekcje zmienne

Struktura zespołu

Khrystyna Mylkyus – Kierownik projektu, programista

Viktor Kozenko – Projektant, programista

Mykola Rudnytskyi - Testowania oraz weryfikacja danych, programista

Yevhen Reminetskyi – Analitik, programista

Ihor Shchupliak – Ekspert techniczny, programista

Andrzej Bogacz- Specjalista zarządzania jakością oprogramowania

Dmytro Khomenko – Lider techniczny, programista

Harmonogram:

Data rozpoczęcia projektu: 23 luty 2017r.

Data zakończenia projektu: 25 maj 2017r.

- 23.02.2017 – 18.03.2017 – Opracowanie wymagań projektu
- 18.03.2017 – 30.03.2017 – Projektowanie struktury projektu
- 30.03.2017- 2.04.2017 – Opracowanie technologii w której będzie tworzony projekt
- 2.04.2017 – 12.05.2017 – Tworzenie oprogramowania oraz testowanie
- 12.05.2017 – 23.05.2017- Opracowanie dokumentacji
- 25.04.2017 – Zamykanie projektu

Określenie celu biznesowego projektu

Cel biznesowy określa korzyści, jakie osiągną udziałowcy projektu dzięki jego realizacji. Aplikacja nie posiada wymagań biznesowych, jest stworzona w celu ukształtowanie umiejętności programowania, w celach edukacyjnych.

Identyfikacja wymagań

Funkcjonalne – system ma realizować takie zadanie:

- ☐ Generowanie reguł
- ☐ Klasyfikować zbiór danych

Niefunkcjonalne – wymagania dotyczące, jakości i sposobu działania systemu, inaczej wymagania jakościowe.

To jest:

- ☐ Dostępność,
- ☐ Niezawodność
- ☐ Możliwości utrzymania
- ☐ Łatwość użytkowania.

Strategia Zarządzania Komunikacją

Plan Komunikacji - opis środków oraz częstotliwości komunikacji pomiędzy zespołem projektowym a interesariuszami projektu. Element Dokumentacji Inicjowania Projektu opisujący, jak interesariusze projektu i inne zainteresowane strony będą informowani w trakcie projektu. Komunikacja pomiędzy członkami zespołu odbywa się w formie spotkań w czasie rzeczywistym oraz poprzez komunikacji on-line via Skype. Wymiana plików planowana poprzez gitlab.rzeszow.pl.

Kryteria jakości

Zarządzanie jakością oprogramowania jest tematyką powiązaną z różnymi metodykami zarządzania projektami, wytwórczymi i operacyjnymi. Metodyka zarządzania projektami PRINCE2 definiuje komponent „Jakość w środowisku projektowym”, który opisuje konieczność zastosowania podwójnej, obiektywnej kontroli wytwarzanych produktów oraz proponuje zastosowanie 4 składników – Systemu Zarządzania Jakością, Funkcji zapewnienia jakości, Planowaniu jakości i

Kontroli jakości oraz technikę „Przeglądy jakości” mają na celu weryfikację czy wytwarzane produkty spełniają przyjęte w projekcie kryteria jakości.

Bibliografia

Managing Successful Projects with PRINCE2 (2009 ed.), TSO, ISBN 978-0-11-331059-3.

[http://aragorn.pb.bialystok.pl/~dmalyszko/SI/projects/Algorytm%20LEM2%20\(2%20os.\).pdf](http://aragorn.pb.bialystok.pl/~dmalyszko/SI/projects/Algorytm%20LEM2%20(2%20os.).pdf)

https://pl.wikipedia.org/wiki/Zarz%C4%85dzanie_jako%C5%9Bci%C4%85_oprogramowania