



METODY INŻYNIERII WIEDZY

METHODS OF KNOWLEDGE ENGINEERING

**Ćwiczenia Laboratoryjne,
Zaliczenie i Egzamin**



Adrian Horzyk
horzyk@agh.edu.pl



AGH
Akademia Górniczo-Hutnicza
w Krakowie

1. WŁASNA APLIKACJA DO EKSPLORACJI DANYCH



(4 ćwiczenia)

Zbuduj własną aplikację do eksploracji danych (do wykorzystania w przyszłości):

- ✓ Realizacja w preferowanym obiektowym języku programowania (np. C#, C++, Java, Python, PHP), lecz pod warunkiem zapewnienia sobie stosownego kompilatora i środowiska programistycznego, jeśli takiego nie ma na pracowni komputerowej, gdzie do dyspozycji są kompilatory MS Visual Studio, Java i Python.
- ✓ Aplikacja powinna posiadać spójny interfejs odczytywania danych z plików.
- ✓ Umożliwiać porównania czasowe i prezentację wyników oraz wizualizację sposobu ich działania w postaci graficznej, pokazując również wyniki pośrednie lub kolejne kroki działania algorytmów, demonstrując sposób i poprawność ich działania.

Wartościowe prace i aplikacje można będzie rozszerzyć do [prac magisterskich](#).

Na zajęciach wolno korzystać z własnych laptopów, jeśli tak jest wygodniej.

EKSPLORACJA DANYCH



Zadania do realizacji z eksploracji danych:

- ✓ Zaimplementuj metodę Apriori, Eclat i FPGrowth (oraz ew. jedną z najszybszych [D-Club](#) [dla ambitnych – realizacja doceniona będzie w ramach egzaminu]) dla zbiorów transakcji, dobierz odpowiednio wsparcie i napisz funkcje wyznaczające:
 - Wszystkie wzorce częste
 - Wszystkie wzorce zamknięte
 - Wszystkie wzorce maksymalne
 - Określ wszystkie silne reguły asocjacyjne dla analizowanych transakcji
- ✓ Zaimplementuj metodę Apriori dla wzorców sekwencyjnych, a następnie wykorzystaj ją do znalezienia wielosłownych terminów, fraz oraz określeń idiomatycznych w analizowanych tekstach, np. wykorzystując [korpusy tekstów Gutenberga](#), poszukując wzorców częstych i maksymalnych.
- ✓ Dokonaj pomiarów czasowych działania tych metod i porównaj szybkość działania tych metod na różnej [wielkości zbiorach danych](#).
- ✓ Wykonaj wizualizację graficzną działania poszczególnych metod uwidaczniając sposób ich działania (dla mniejszych zbiorów) oraz uzyskane wyniki.

2. WŁASNA APLIKACJA DO BUDOWY MODELU WIEDZY I EKSPLORACJI



(5 ćwiczeń)

Zbuduj własną aplikację do budowy modelu wiedzy na bazie asocjacyjnych grafowych struktur danych AGDS oraz wykorzystaj ją eksploracji danych:

- ✓ Realizacja w preferowanym obiektowym języku programowania (np. C#, C++, Java, Python, PHP), lecz pod warunkiem zapewnienia sobie stosownego kompilatora i środowiska programistycznego, jeśli takiego nie ma na pracowni komputerowej, gdzie do dyspozycji są kompilatory MS Visual Studio, Java i Python.
- ✓ Wolno korzystać z własnych laptopów, jeśli tak wygodniej.
- ✓ Aplikacja powinna posiadać spójny interfejs odczytywania danych z plików.
- ✓ Umożliwiać porównania czasowe i prezentację wyników oraz wizualizację sposobu ich działania w postaci graficznej, pokazując również wyniki pośrednie lub kolejne kroki działania algorytmów, demonstrując sposób i poprawność ich działania.

Wartościowe prace i aplikacje można będzie rozszerzyć do [prac magisterskich](#).

Na zajęciach wolno korzystać z własnych laptopów, jeśli tak jest wygodniej.

BUDOWA MODELU WIEDZY ORAZ EKSPLORACJA DANYCH



Zadania do realizacji z budowy modelu wiedzy i eksploracji:

- ✓ Zbuduj asocjacyjną grafową strukturę danych AGDS dla wybranych zbiorów danych oraz zrealizuj następujące funkcje:
 - Wyszukiwanie wzorców podobnych
 - Wyszukiwanie wzorców częstych (Asocjacyjny algorytm Apriori)
 - Wyszukiwanie wzorców zamkniętych
 - Wyszukiwanie wzorców maksymalnych
 - Określ silne reguły asocjacyjne w analizowanych transakcjach
 - Wykorzystaj asocjacyjną strukturę danych do klasyfikacji i klasteryzacji danych.
- ✓ Dokonaj pomiarów czasowych działania tej metody wykorzystującej model wiedzy w stosunku do wcześniej realizowanych metod eksploracji danych i porównaj szybkość ich działania na różnej wielkości zbiorach danych.
- ✓ Wykonaj wizualizację graficzną struktury AGDS (dla mniejszych zbiorów) oraz sposobu jej działania, jak również uzyskanych wyników.

ZBUDUJ ASOCJACYJNĄ GRAFOWĄ STRUKTURĘ DANYCH POZWALAJĄCĄ PRZECHOWYWAĆ DANE I RELACJE



Zbuduj graf AGDS dla wybranych zbiorów danych (np. Iris i Wine, ...) wykorzystując listy lub tablice sortujące, funkcje haszujące, słowniki, wyszukiwanie połówkowe albo AVB+drzewa itp.

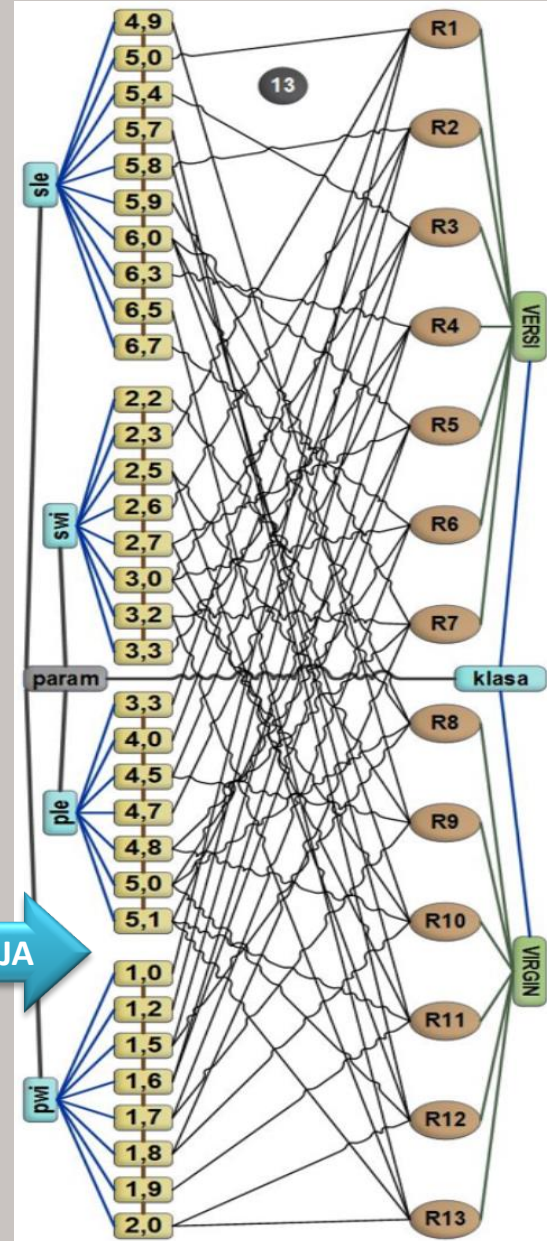
Napisz metody pozwalające szybko znaleźć grupę wzorców najbardziej podobnych do wybranego wzorca, pewnego ich podzbioru albo podanego podzbioru wartości atrybutów przez użytkownika.

Napisz metody pozwalające szybko znaleźć grupę wzorców spełniających pewne kryteria filtrowania (grupę, klastry), np. dane atrybutu lub atrybutów mieszczą się w pewnym zakresie, oraz dokonać klasyfikacji.

Napisz metodę pozwalającą szybko określić wszystkie wzorce częste wykorzystując asocjacje i ilość połączeń dla poszczególnych wartości. Porównaj sposób działania z metodami eksploracji bezpośredniej operujących na strukturach tabelarycznych, następnie zmierz i porównaj czasy wykonania tych operacji dla różnych zbiorów danych.

param	sle	swi	ple	pwi	klasa
R1	5,0	2,3	3,3	1,0	VERSI
R2	5,8	2,6	4,0	1,2	VERSI
R3	5,4	3,0	4,5	1,5	VERSI
R4	6,3	3,3	4,7	1,6	VERSI
R5	6,0	2,7	5,1	1,6	VERSI
R6	6,7	3,0	5,0	1,7	VERSI
R7	5,9	3,2	4,8	1,8	VERSI
R8	6,0	2,2	5,0	1,5	VIRGIN
R9	4,9	2,5	4,5	1,7	VIRGIN
R10	6,0	3,0	4,8	1,8	VIRGIN
R11	5,8	2,7	5,1	1,9	VIRGIN
R12	5,7	2,5	5,0	2,0	VIRGIN
R13	6,5	3,2	5,1	2,0	VIRGIN

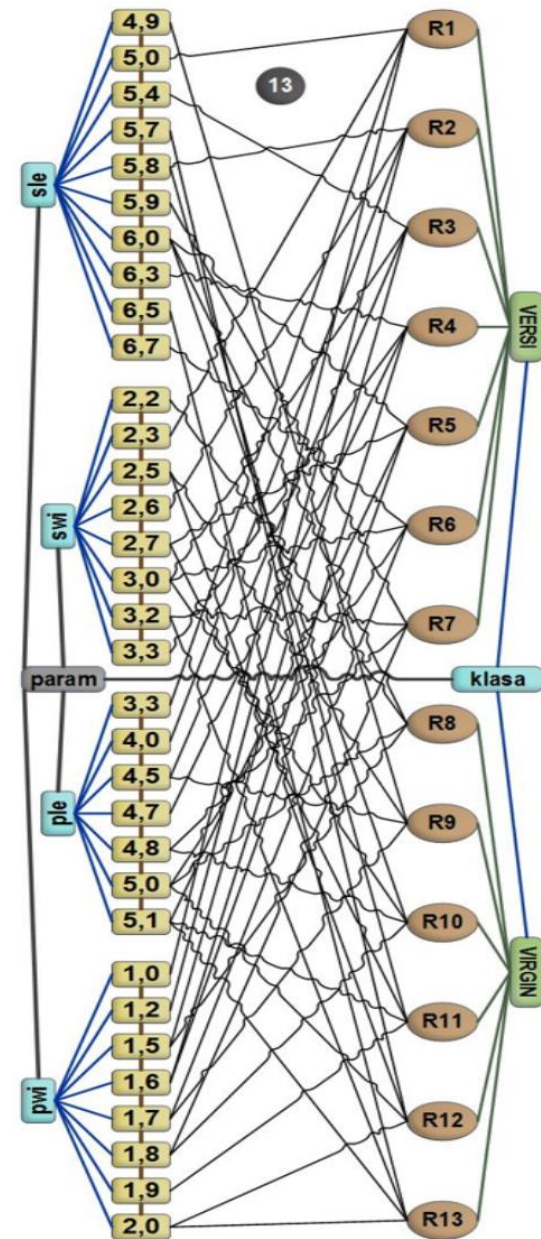
TRANSFORMACJA



Uwagi implementacyjne: konwersja tabeli do AGDS



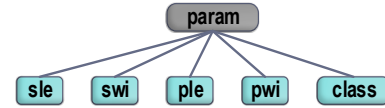
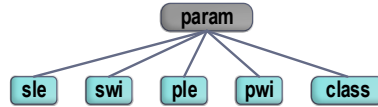
1. Wykorzystaj programowanie zorientowane obiektowo i zaimplementuj każdy węzeł grafu jako klasę zawierająca reprezentowaną wartość lub nazwę oraz listę wskaźników do innych połączonych węzłów w grafie.
2. Warto potraktować węzeł PARAM jako wyróżniony w grafie, poprzez który uzyskujemy szybki (w czasie stałym) pośredni dostęp do wszystkich elementów w grafie. Graf AGDS można oprzeć na strukturze drzewiastej: korzeniem jest PARAM, na 1. poziomie nazwy atrybutów: pwi, ple, swi, sle, klasa, na 2. poziomie posortowane wartości atrybutów, na 3. poziomie etykiety rekordów/encji, przy czym tutaj pojawiają się cykle.
3. Do reprezentacji danych dla poszczególnych atrybutów posługujemy się słownikami, tablicami lub listami sortującymi, można wykorzystać też AVB-drzewa lub funkcję haszujące czy też korzystamy z sortowania i wyszukujemy połówkowo (w wersji na tablicach). W przypadku tablic odpada też konieczność reprezentacji połączeń pomiędzy sąsiadującymi wartościami za pomocą wskaźników, lecz tablica jest gorszym rozwiązaniem w przypadku przewidywanych częstych operacji dodawania lub usuwania nowych wartości atrybutów. Preferowane są więc struktury dynamiczne lub samoorganizujące się, np. AVB-drzewa.
4. Jeśli zbiór obiektów (rekordów) jest stały i z góry zdefiniowany, można pójść na skróty i wykorzystać dla wartości każdego atrybutu szybkie sortowanie stogowe (kopcowe) z usuwaniem duplikatów wartości. Następnie przekształcić posortowane tablice wartości w tablice klas reprezentujące węzły i połączyć je z pozostałymi węzłami grafu przeglądając zbiór uczący ponownie.
5. Opcjonalnie można utworzyć połączenia pomiędzy sąsiadującymi etykietami wzorców: R1 – – R13, w celu odwzorowania ich kolejności.



Uwagi implementacyjne: Grafowa struktura AGDS oparta na drzewie



TRANSFORMACJA ASOCJACYJNA



param	sle	swi	ple	pwi	class
R1	5,0	2,3	3,3	1,0	VERSI
R2	5,8	2,6	4,0	1,2	VERSI
R3	5,4	3,0	4,5	1,5	VERSI
R4	6,3	3,3	4,7	1,6	VERSI
R5	6,0	2,7	5,1	1,6	VERSI
R6	6,7	3,0	5,0	1,7	VERSI
R7	5,9	3,2	4,8	1,8	VERSI
R8	6,0	2,2	5,0	1,5	VIRGIN
R9	4,9	2,5	4,5	1,7	VIRGIN
R10	6,0	3,0	4,8	1,8	VIRGIN
R11	5,8	2,7	5,1	1,9	VIRGIN
R12	5,7	2,5	5,0	2,0	VIRGIN
R13	6,5	3,2	5,1	2,0	VIRGIN



param	sle	swi	ple	pwi	class
R1	5,0	2,3	3,3	1,0	VERSI
R2	5,8	2,6	4,0	1,2	VERSI
R3	5,4	3,0	4,5	1,5	VERSI
R4	6,3	3,3	4,7	1,6	VERSI
R5	6,0	2,7	5,1	1,6	VERSI
R6	6,7	3,0	5,0	1,7	VERSI
R7	5,9	3,2	4,8	1,8	VERSI
R8	6,0	2,2	5,0	1,5	VIRGIN
R9	4,9	2,5	4,5	1,7	VIRGIN
R10	6,0	3,0	4,8	1,8	VIRGIN
R11	5,8	2,7	5,1	1,9	VIRGIN
R12	5,7	2,5	5,0	2,0	VIRGIN
R13	6,5	3,2	5,1	2,0	VIRGIN



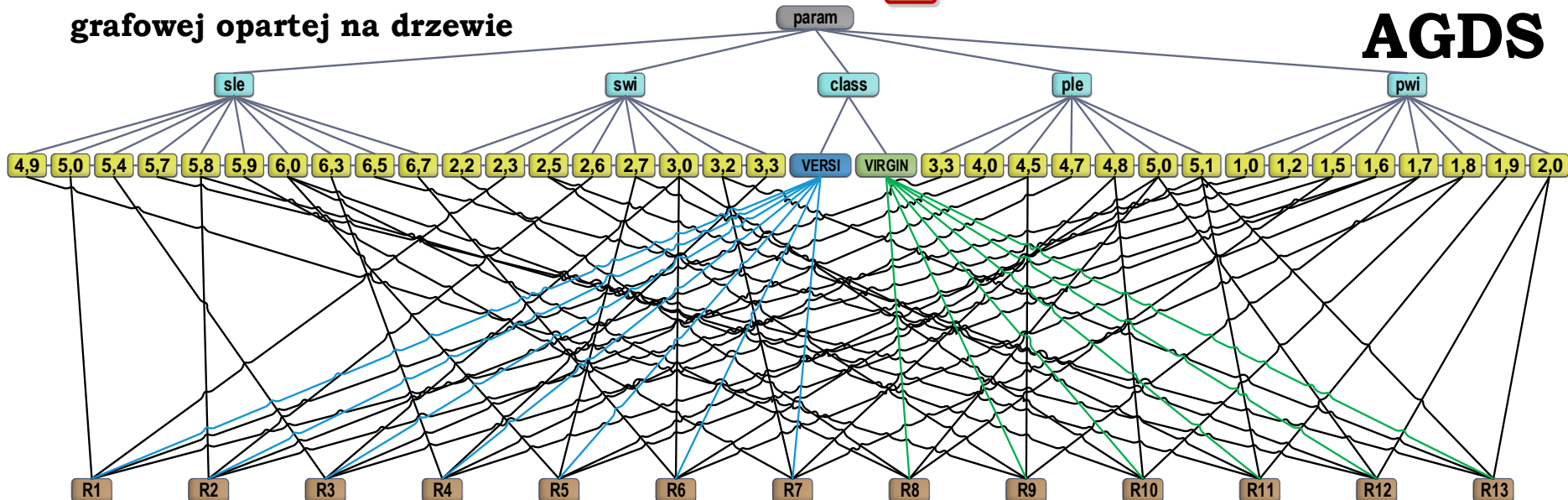
param	sle	swi	ple	pwi	class
R1	5,0	2,3	3,3	1,0	VERSI
R2	5,8	2,6	4,0	1,2	VERSI
R3	5,4	3,0	4,5	1,5	VERSI
R4	6,3	3,3	4,7	1,6	VERSI
R5	6,0	2,7	5,1	1,6	VERSI
R6	6,7	3,0	5,0	1,7	VERSI
R7	5,9	3,2	4,8	1,8	VERSI
R8	6,0	2,2	5,0	1,5	VIRGIN
R9	4,9	2,5	4,5	1,7	VIRGIN
R10	6,0	3,0	4,8	1,8	VIRGIN
R11	5,8	2,7	5,1	1,9	VIRGIN
R12	5,7	2,5	5,0	2,0	VIRGIN
R13	6,5	3,2	5,1	2,0	VIRGIN



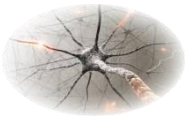
R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11 R12 R13

WZORCE IRIS w strukturze grafowej opartej na drzewie

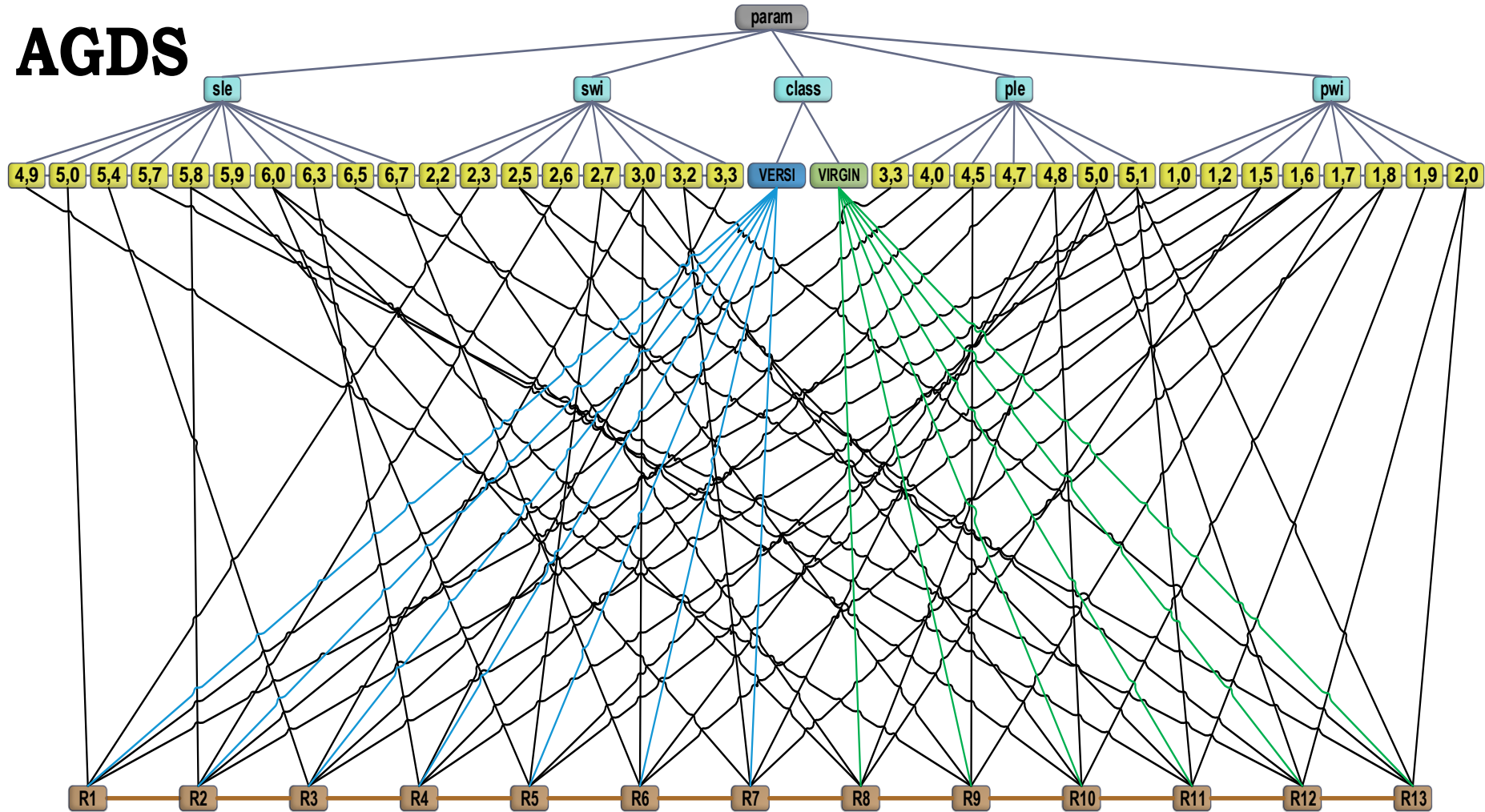
AGDS



Dostęp do elementów grafu uzyskuje za pośrednictwem wyróżnionego węzła głównego param, który posiada połączenia z wszystkimi parametrami (atrybutami, cechami), a one z wartościami, a te etykietami wzorców.



AGDS



Struktura grafowa oparta na drzewie pozwala uzyskać bardzo szybki dostęp do dowolnych danych oraz reprezentowanych relacji pomiędzy tymi danymi.

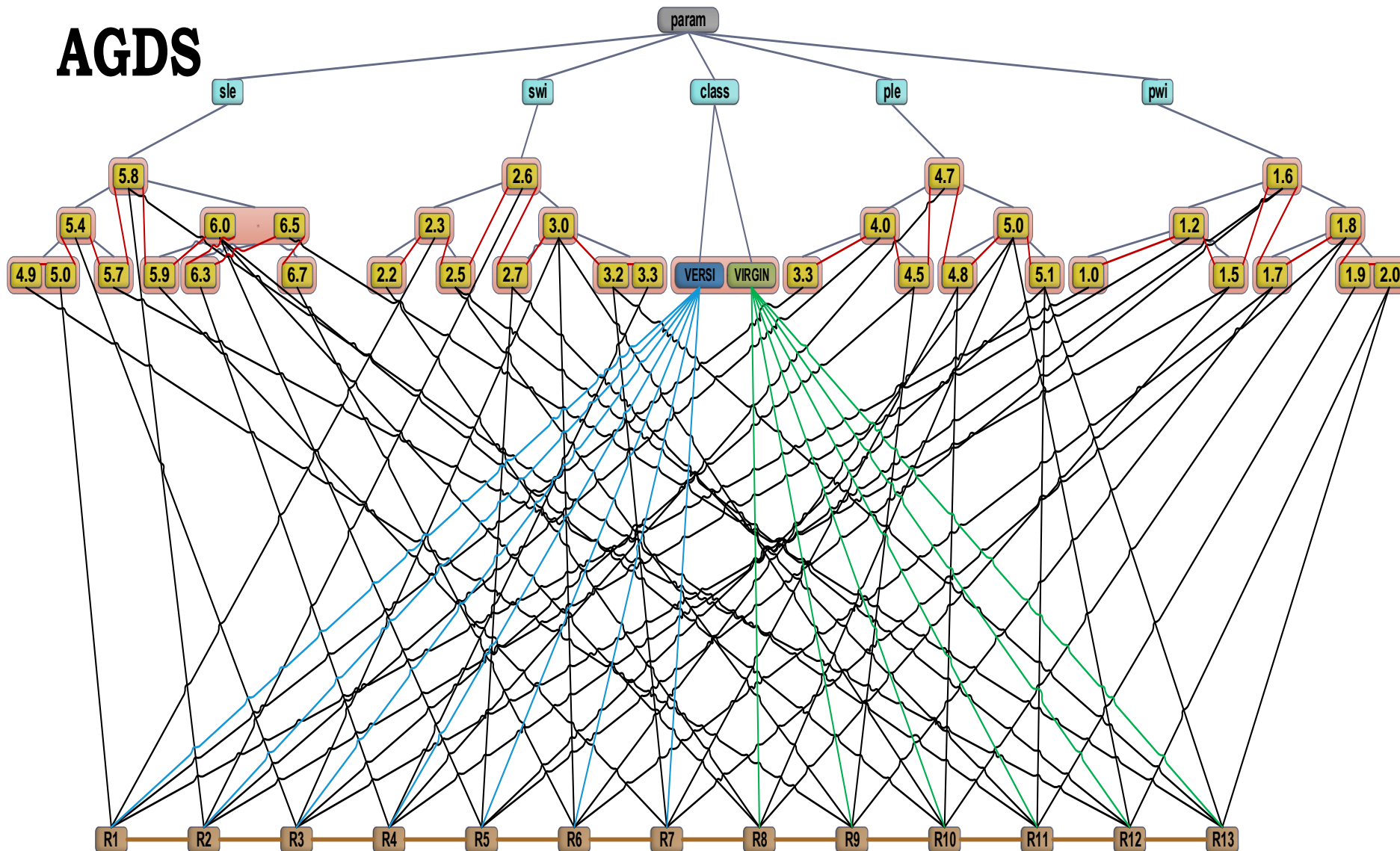
Jeśli kolejność rekordów w tabeli bazodanowej ma znaczenie, dodajemy połączenia pomiędzy etykietami rekordów, celem odwzorowania tej kolejności.



PASYWNA ASOCJACYJNA GRAFOWA STRUKTURA DANYCH



AGDS



AVB+DRZEWA mogą zawierać dodatkowe krawędzie (bordowe) łączące ze sobą sąsiednie (w sensie porządku) wartości poszczególnych atrybutów.

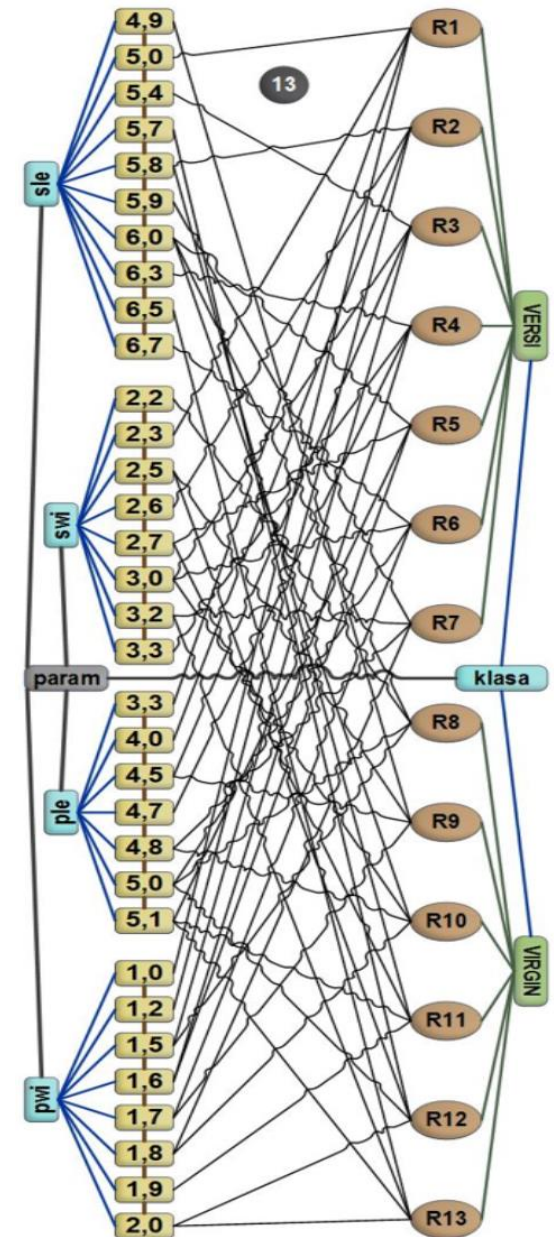
Uwagi implementacyjne: wnioskowanie i eksploracja



- ✓ Każda metoda operująca na grafie AGDS porusza się po ograniczonej ilości krawędzi tego grafu korzystając z lokalnych list wskaźników znajdujących się w poszczególnych węzłach grafu.
- ✓ Węzły reprezentujące atrybuty posiadają dwa dodatkowe wskaźniki do węzłów reprezentujących minimalną MIN i maksymalną MAX wartość tego atrybutu. Ponadto w węźle atrybutu można dla przechowywać wartość przedziału liczb $RANGE$ reprezentowanych przez powiązane wartości definiujące ten atrybut.
- ✓ W podstawowej wersji implementacji struktury AGDS wartości wag pomiędzy sąsiednimi posortowanymi wartościami atrybutów określamy wg wzoru:

$$w_{a,b}^{P_i} = 1 - \frac{|a - b|}{MAX_{P_i} - MIN_{P_i}} = 1 - \frac{|a - b|}{RANGE_{P_i}}$$

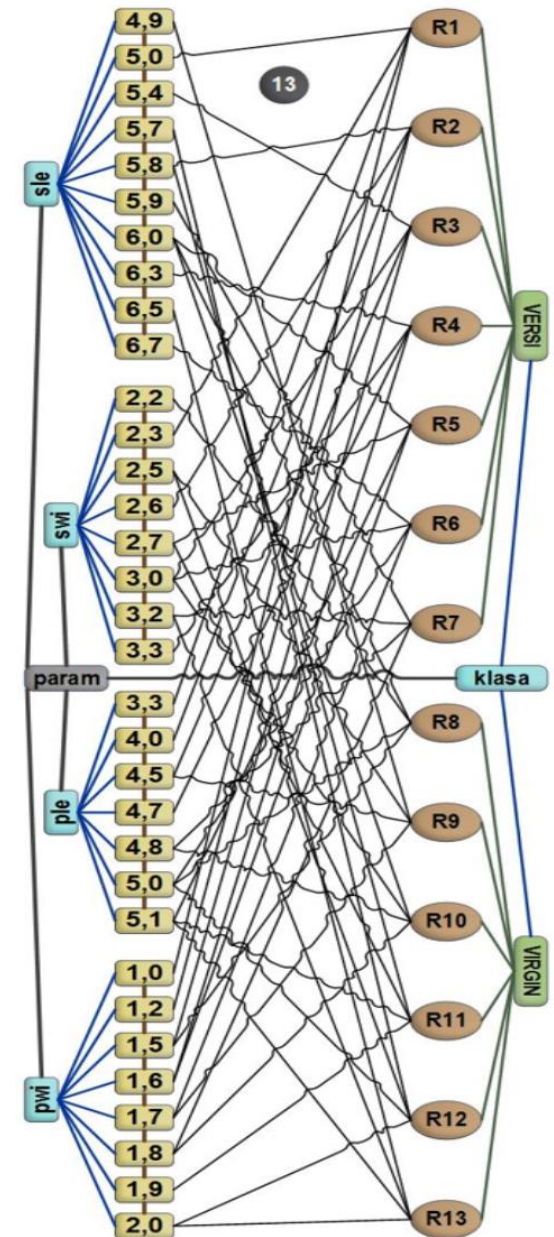
- ✓ W podstawowej wersji implementacji struktury AGDS wartości wag pomiędzy węzłami wartości i węzłami wzorców (obiektów) przyjmują wartości jednostkowe ($w = 1$), co umożliwia łatwe policzenie podobieństw dodając $+1$ do zmiennych w połączonych węzłach, ew. przemnożone przez wagę w połączenia z węzłami reprezentującymi podobne wartości.
- ✓ Przy poszukiwaniu podobnych obiektów bierzemy pod uwagę wartości podobne, tzn. jeśli $\frac{|a-b|}{RANGE_{P_i}} \leq p$ gdzie p % podobieństwa.



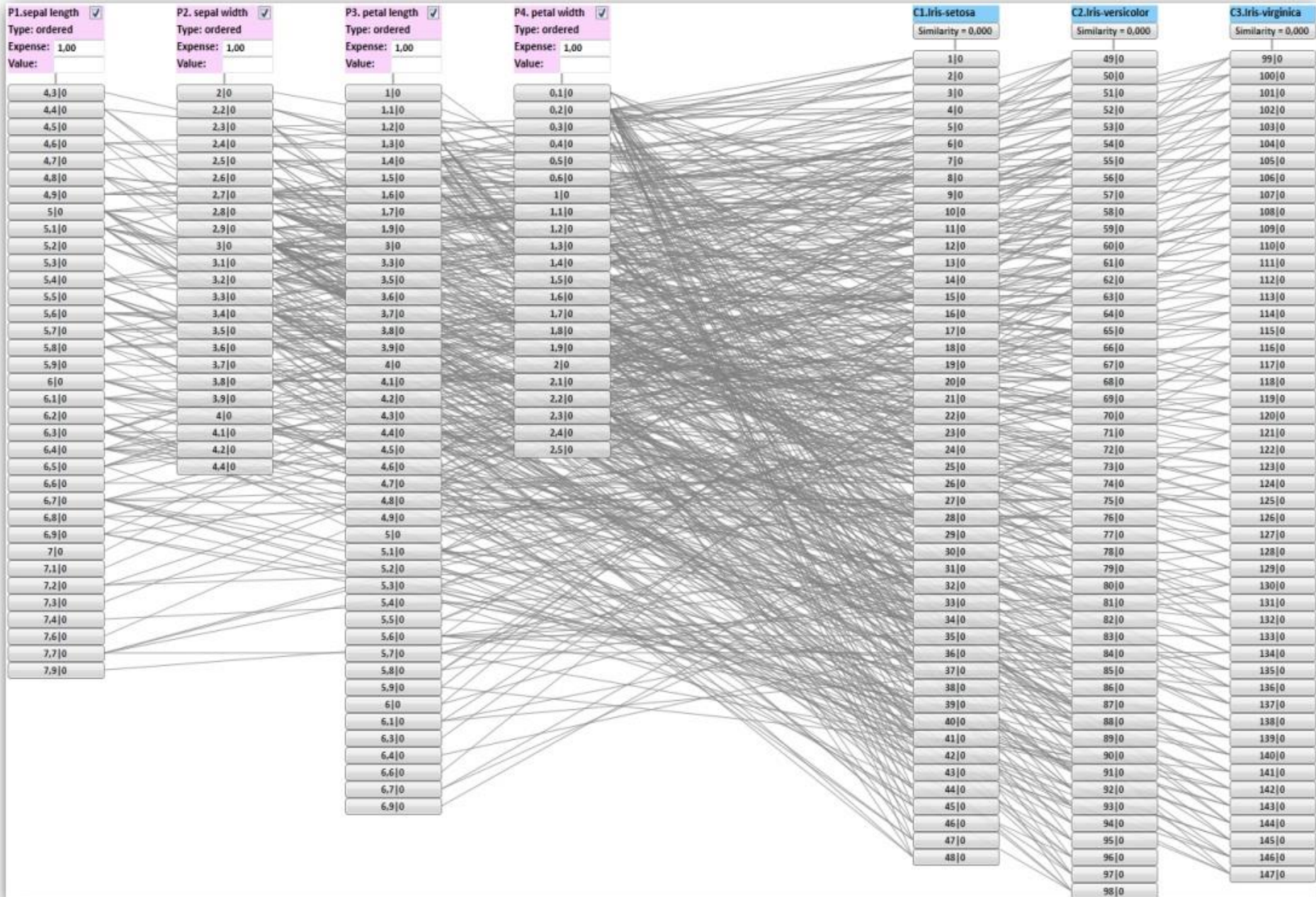
Wyznaczanie i udostępnianie wyników wnioskowania



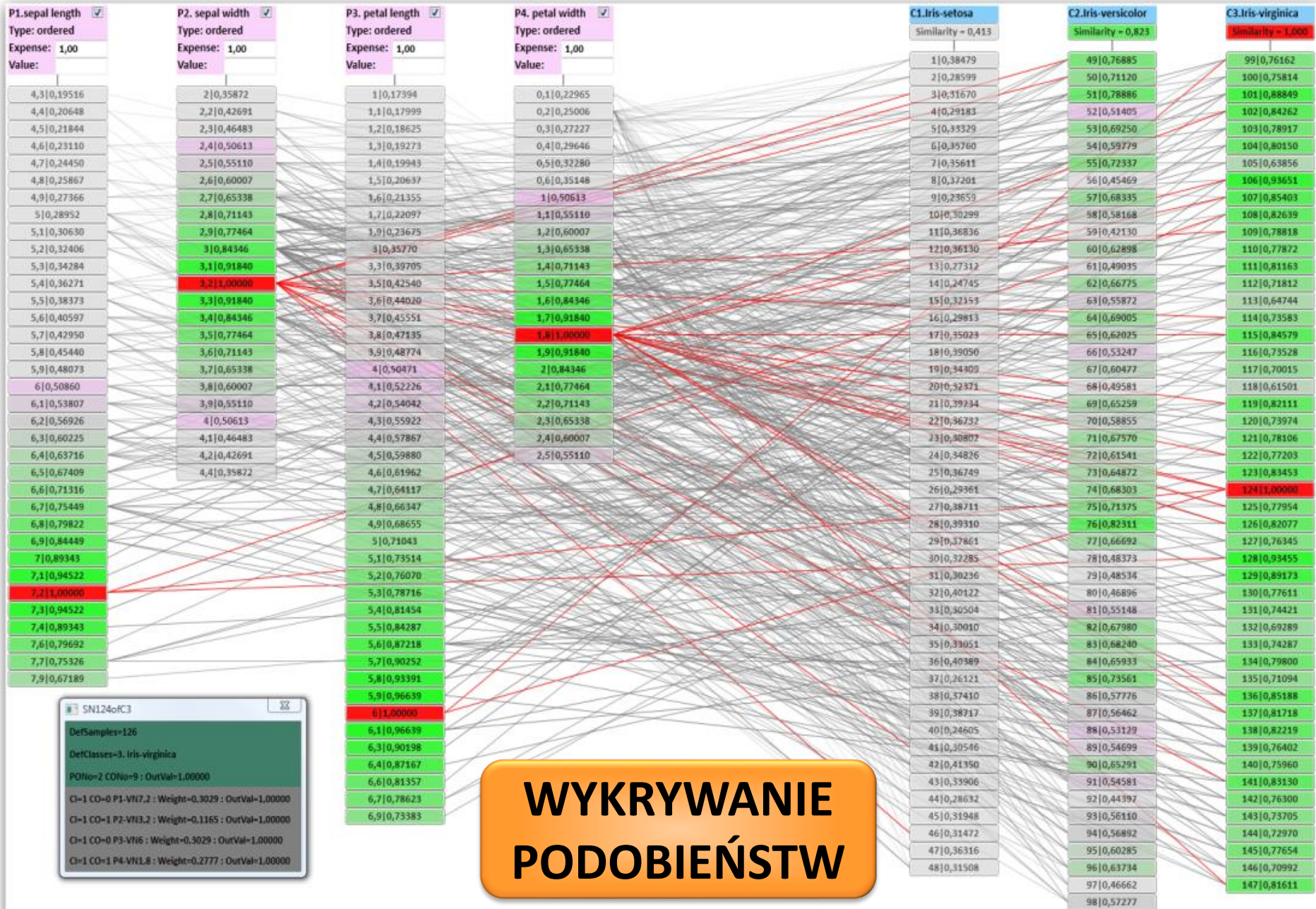
- ✓ Do reprezentacji wartości parametrów warto zastosować mapę w C#, gdyż wykorzystuje ona **strukturę drzewa** i wyszukuje elementy w czasie **logarytmicznym**, co w przypadku wykorzystania **słownika** lub **listy** realizowane jest przy złożoności **liniowej**. Ze względu na symulację AGDS na maszynie nie równoległej, występuje konieczność określenia węzła, który definiuje wartość danego atrybutu.
- ✓ Obliczanie podobieństwa obiektów przeprowadzamy w taki sposób, że obiekt R_n wysyła żądanie wyznaczenia podobieństw do węzłów definiujących jego wartości, czyli uruchamia metodę w węzłach wartości. Te metody dodają 1 (**metoda dodawania**) do wszystkich węzłów R_i , z którymi są powiązane oraz do węzłów sąsiednich wartości zgodnie z wagą obliczoną wg wzoru z poprzedniego slajdu. Następnie te sąsiednie węzły dodają (**metoda dodawania**) tą wartość (wynikającą z wagi) do wszystkich połączonych z nimi węzłów R_i . Można rozważyć tylko najbliższe sąsiedztwo lub dalsze, wtedy przechodzimy do kolejnych węzłów wartości dalej oddalonych aż do osiągnięcia pewnej określonej z góry odległości maksymalnej albo do MIN lub MAX wartości tego atrybutu. Każdy węzeł R_i , w którym była wywołana **metoda dodawania** wartości przez węzeł wartości dodaje do listy węzłów wynikowych wskaźnik do siebie, o ile jeszcze w tej liście nie istniał. Uzyskujemy więc krótką wynikową listę wskaźników do węzłów R_i , w których wartości są niezerowe i stanowią wynik działania metody. Tą listę wskaźników najlepiej od razu sortować wg wartości sum R_i przy aktualizacji wartości w węzłach R_i . W taki sposób uzyskujemy posortowaną listę obiektów od najbardziej podobnych do najmniej podobnych zgodnie ze zdefiniowanymi kryteriami.



Pełna struktura AGDS dla zbioru Iris z ML Repository



ZNAJDYWANIE PODOBIEŃSTW POMIĘDZY DANYMI I ICH GRUPAMI



WYKRYWANIE PODOBIEŃSTW

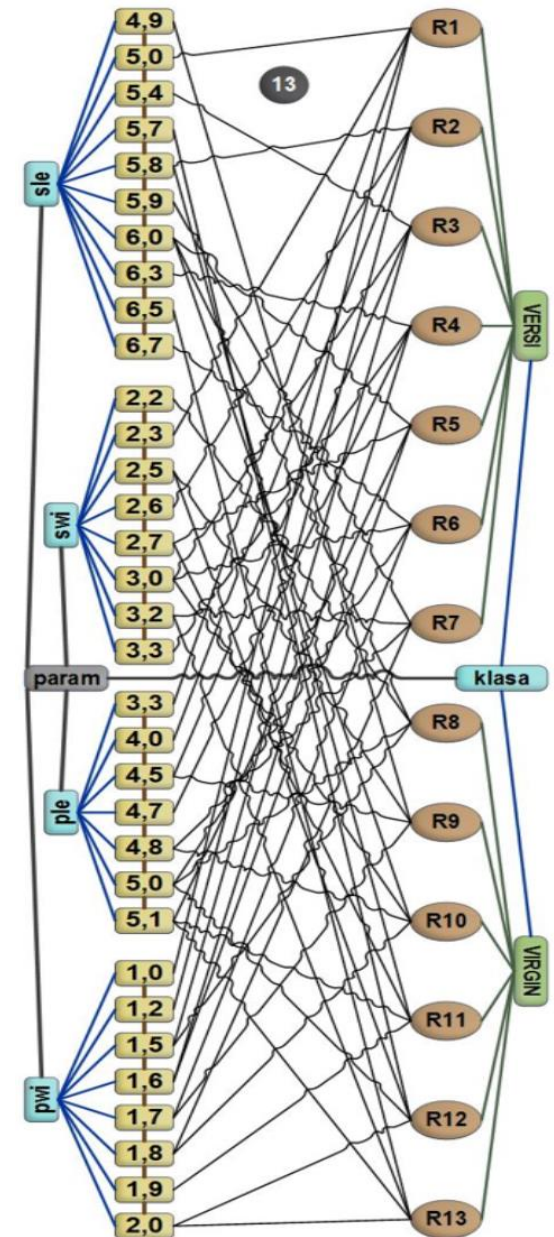
Ćwiczenia do wykonania na strukturach AGDS



1. Znajdź najbardziej podobny/e obiekt(y) do wskazanego (uczącego lub dowolnego innego), dodatkowo można określić ile takich najbardziej podobnych obiektów nas interesuje. Jeśli jakiejś wartości nie ma na wejściu wśród reprezentowanych wartości danego atrybutu pobudzamy pozostałe wartościami obliczonymi według tego samego wzoru, który stosujemy do wyznaczenia wag:

$$w_{a,b}^{P_i} = 1 - \frac{|a - b|}{MAX_{P_i} - MIN_{P_i}} = 1 - \frac{|a - b|}{RANGE_{P_i}}$$

2. Znajdź najbardziej podobne obiekty do pewnej grupy obiektów (uczących lub dowolnych innych). Realizujemy to podobnie jak poprzednie zadanie, sumując w neuronach Ri otrzymane wartości podobieństw dla wszystkich obiektów ze wskazanej grupy i na tej podstawie wyznaczając najbardziej podobne obiekty do podanej grupy. Można to zadanie też nieco sprytniej uprościć w taki sposób, że na wejściu określamy sobie jeden hiperwzorzec będący sumą wartości wszystkich wzorców tej sumy składający się ze wszystkich wzorców sumy, a następnie nim pobudzić węzły wartości i obliczyć podobieństwa obiektów przy jednym przejściu przez ten graf. Obydwa sposoby powinny dać ten sam wynik.
3. Znajdź pozostałe cechy obiektów charakteryzujących się pewnymi cechami. Mamy określone pewne cechy lub przedziały cech dla części atrybutów. Poprzez nie przechodzimy do obiektów, wyszukując najbardziej podobne, a następnie od obiektów przechodzimy do pozostałych atrybutów/cech i wyznaczamy te najsilniejsze dokonując w nich również sumowania w zależności od ich połączeń z obiektami.
4. Proszę eksperymentować na różnych zbiorach i zaproponować inne...



WNISKOWANIE ASOCJACYJNE Z WYKORZYSTANIEM AGDS

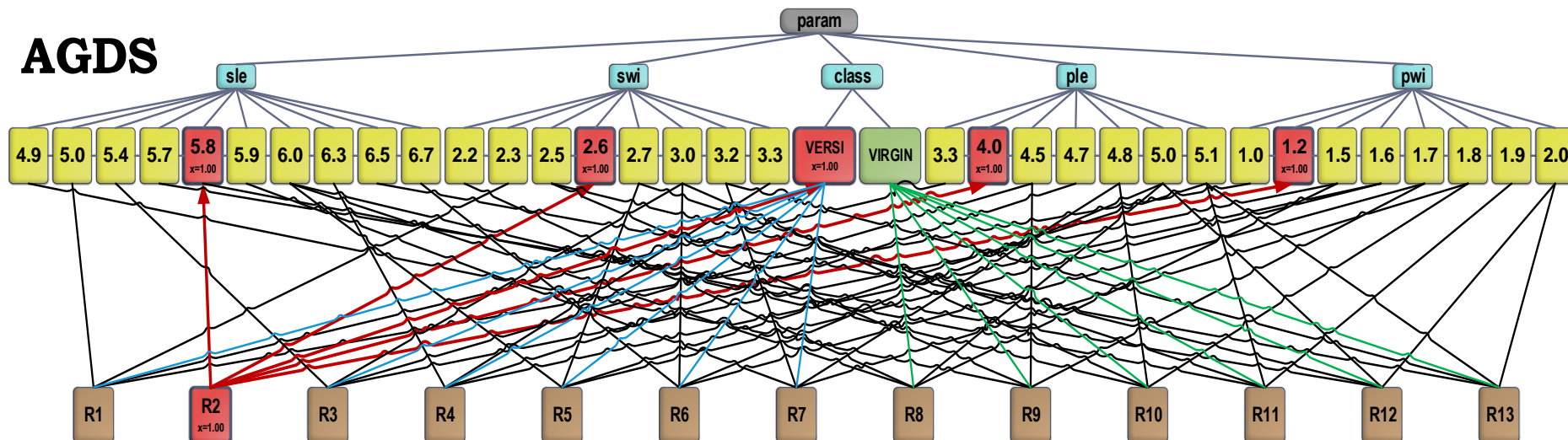


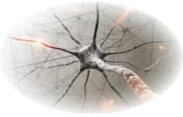
Struktury asocjacyjne możemy teraz wykorzystać do **asocjacyjnego wnioskowania**, które polega na tym, że przechodzimy w odpowiedni sposób po krawędziach grafu wyznaczając wartości dla innych wierzchołków tego grafu i na tej podstawie otrzymujemy informację na temat np. podobieństwa lub obiektów grafu, które spełniają zadane kryteria.

Wykorzystajmy wobec tego nasz graf AGDS utworzony dla 13 Irysów do takiego wnioskowania poszukując obiektów (Irysów) najbardziej podobnych do wskazanego R2. Przechodzimy więc od węzła R2 do wszystkich innych połączonych z nim krawędzią i wyznaczamy wartości dla tych węzłów z wykorzystaniem wag dla poszczególnych połączeń. Najprostszy sposób polega na przemnażaniu wartości przez wartość wagi, aczkolwiek istnieją również inne sposoby. Natomiast jeśli do danego węzła schodzi się wiele krawędzi, wtedy w węźle wyznaczamy sumę ważoną tych wartości:

1. Wychodzimy z węzła R2, który przyjmuje wartość $x=1,0$, gdyż jest w 100% podobny sam do siebie.
2. Wyznaczamy wartości dla połączonych z nim węzłów: 5,8, 2,6, VERSI, 4,0 oraz 1,2 mnożąc wartość w węźle R2 przez wagę połączeń prowadzących do tych węzłów – w tym wypadku wszystkie te wagi równe są jedności, gdyż są to połączenia do wartości definiujących obiekt R2.

AGDS



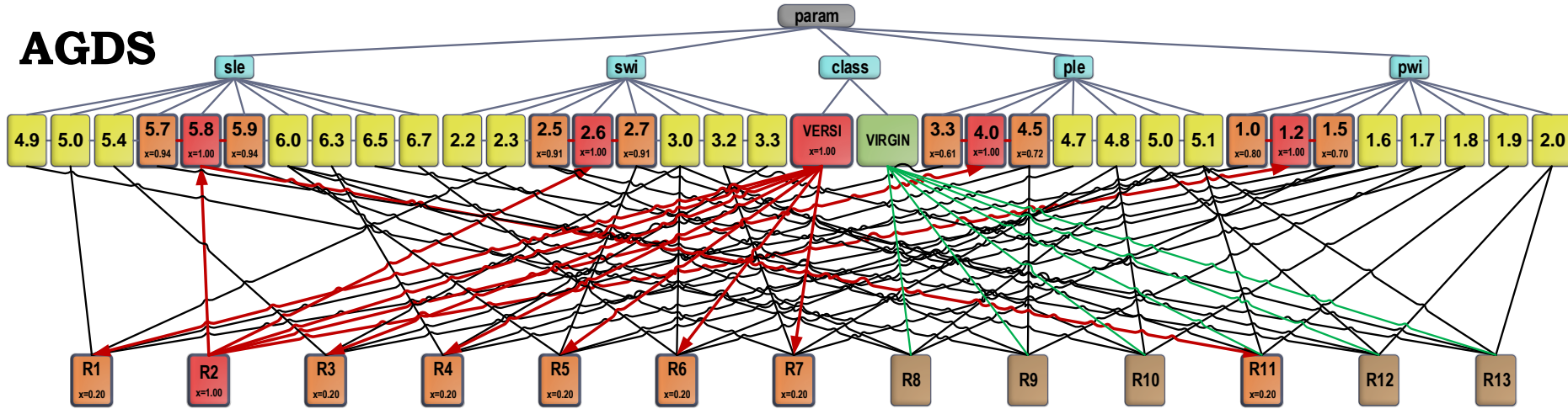


WNISKOWANIE ASOCJACYJNE Z WYKORZYSTANIEM AGDS

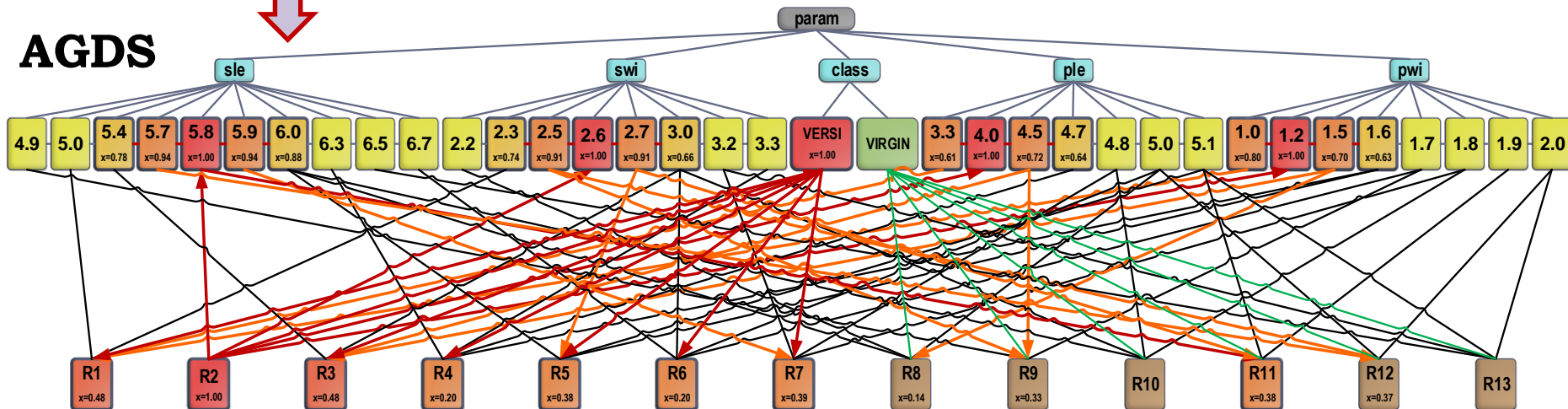


3. W węzłach 5,8, 2,6, VERSI, 4,0 oraz 1,2 otrzymujemy więc wartości równe jedności ($x=1,0$).
4. Wartości w tych węzłach przemnażamy przez wagi połączeń tych węzłów z innymi węzłami (sąsiednimi) i obliczamy wartości x dla nich. Podobnie postępujemy z połączonymi węzłami reprezentującymi obiekty, przy czym dla nich dodatkowo sumujemy wartości ważne przesyłane im przez węzły reprezentujące wartości atrybutów.

AGDS



AGDS



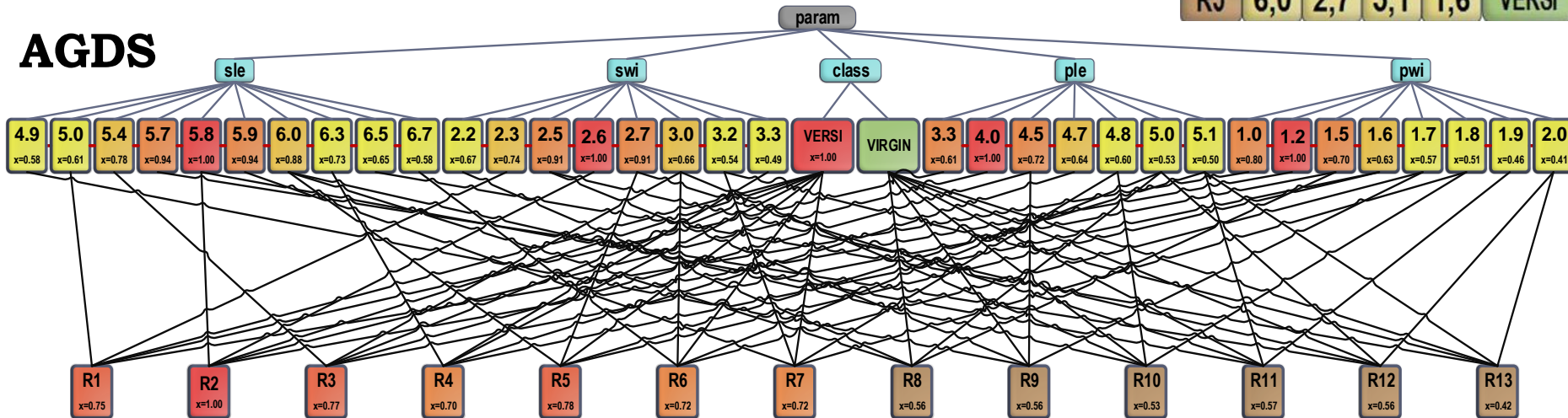
WNOSKOWANIE ASOCJACYJNE Z WYKORZYSTANIEM AGDS



5. W końcu, gdy przejdziemy po wszystkich powiązanych (skojarzonych) wartościach atrybutów wyznaczając dla nich miarę podobieństwa (poprzez przemnożenie wartości reprezentowanej przez sąsiada przez wartość łączącej ich wagi) uzyskamy, oraz przemnożymy obliczone wartości dla węzłów wartości atrybutów przez wagi łączące je z węzłami obiektów (tutaj $w = 1/5 = 0,2$), wyznaczamy sumaryczne wartości (sumy ważone) dla węzłów obiektów, które stanowią pewną miarę podobieństwa obiektu R2 do pozostałych obiektów Rx. Z poniższego grafu AGDS wynika, iż najbardziej podobne objekty to: R5 (78%), R3 (77%), R1 (75%), ...

R1	5,0	2,3	3,3	1,0	VERSI
R2	5,8	2,6	4,0	1,2	VERSI
R3	5,4	3,0	4,5	1,5	VERSI
R4	6,3	3,3	4,7	1,6	VERSI
R5	6,0	2,7	5,1	1,6	VERSI

AGDS



Warto również zaznaczyć, iż grafy AGDS nie są strukturami neuronowymi, wobec tego nie musimy wartości węzłów przemnażać przez wagi przechodząc do połączonych węzłów, lecz możemy zastosować również inne sposoby, np. możemy od wartości reprezentowanej przez węzeł odejmować wartość równą $x' = x - (1 - w)$, gdzie x reprezentuje wartość węzła wysyłającego wartość do sąsiada x' . Wtedy uzyskamy inną miarę podobieństwa, a wartości wyznaczone dla węzłów Rx będą się bardziej różnić! Możemy też zastosować wzory na obliczanie wag dla grafów DASNG do wyznaczenia wartości wag pomiędzy węzłami wartości atrybutów i węzłami obiektów: $w = 1 / \text{ilość połączeń wychodzących}$.

3. WŁASNA APLIKACJA DO KLASYFIKACJI I KLASTERYZACJI DANYCH



(4 ćwiczenia)

Zbuduj własną aplikację do klasyfikacji i klasteryzacji danych:

- ✓ Realizacja w preferowanym obiektowym języku programowania (np. C#, C++, Java, Python, PHP), lecz pod warunkiem zapewnienia sobie stosownego kompilatora i środowiska programistycznego, jeśli takiego nie ma na pracowni komputerowej, gdzie do dyspozycji są kompilatory MS Visual Studio, Java i Python.
- ✓ Wolno korzystać z własnych laptopów, jeśli tak wygodniej.
- ✓ Aplikacja powinna posiadać spójny interfejs odczytywania danych z plików.
- ✓ Umożliwiać porównania czasowe i prezentację wyników oraz wizualizację sposobu ich działania w postaci graficznej, pokazując również wyniki pośrednie lub kolejne kroki działania algorytmów, demonstrując sposób i poprawność ich działania.

Wartościowe prace i aplikacje można będzie rozszerzyć do [prac magisterskich](#).

Na zajęciach wolno korzystać z własnych laptopów, jeśli tak jest wygodniej.

PREZENTACJA KOŃCOWA



Przygotuj 15-min. prezentację końcową wszystkich zrealizowanych zadań, aplikacji oraz uzyskanych wyników i przedstaw ją na ostatnich zajęciach w semestrze:

- ✓ Opisz zastosowane metody i porównaj sposób ich działania.
- ✓ Pokaż strukturę, model i wyniki ich działania korzystając z graficznych wizualizacji w swoich aplikacjach.
- ✓ Porównaj wyniki i zinterpretuj je.

Wartościowe prace i aplikacje można będzie rozszerzyć do [prac magisterskich](#).

EGZAMIN



Egzamin z Metod inżynierii wiedzy może być przeprowadzony w dwóch alternatywnych postaciach (każdy student może sobie wybrać bardziej dogodną dla siebie formę):

1. **Teoretycznej (ustny)** z zakresu całego materiału z wykładów i z ćwiczeń.
 2. **Praktycznej (w formie projektu)** rozwijając zakres materiału z wykładów i ćwiczeń o realizację ciekawego projektu (nieco trudniejszego zadania niż na ćwiczeniach), realizując implementację jakiejś trudniejszej metody z zakresu eksploracji danych (np. D-Club), czy też z zakresu modelowania i eksploracji wiedzy (implementując jakąś asocjacyjną strukturę do reprezentacji i eksploracji wiedzy) itp., wykonując porównania oraz finalną prezentację rozwiązania i osiągniętych wyników. Oczywiście zakres takich projektów egzaminacyjnych należy ustalić z prowadzącym przedmiot.
- ✓ Dodatkową korzyścią płynącą z podjęcia egzaminu w formie praktycznej jest możliwość dalszego rozwinięcia wybranego projektu do [pracy magisterskiej](#), której temat i zakres może zostać ustalony z prowadzącym przedmiot. Ponadto jeśli praca magisterska będzie miała charakter badawczy i zostanie zrealizowana na odpowiednio wysokim poziomie merytorycznym, algorytmy i zgromadzone wyniki mogą zostać we współpracy studenta z prowadzącym przedmiot opisane w [krótkim artykule naukowym](#) o zasięgu światowym (w języku angielskim) i dopisane do dorobku (C.V.) studenta, którym będzie się mógł pochwalić! W przypadku dalszej fascynacji tematem, można podjąć też [studia doktoranckie](#), które według nowej formuły Szkół Doktorskich (od. 2019 r.) powiązane są z [atrakcyjnymi stypendiami i możliwościami rozwojowymi](#) dla młodych naukowców.