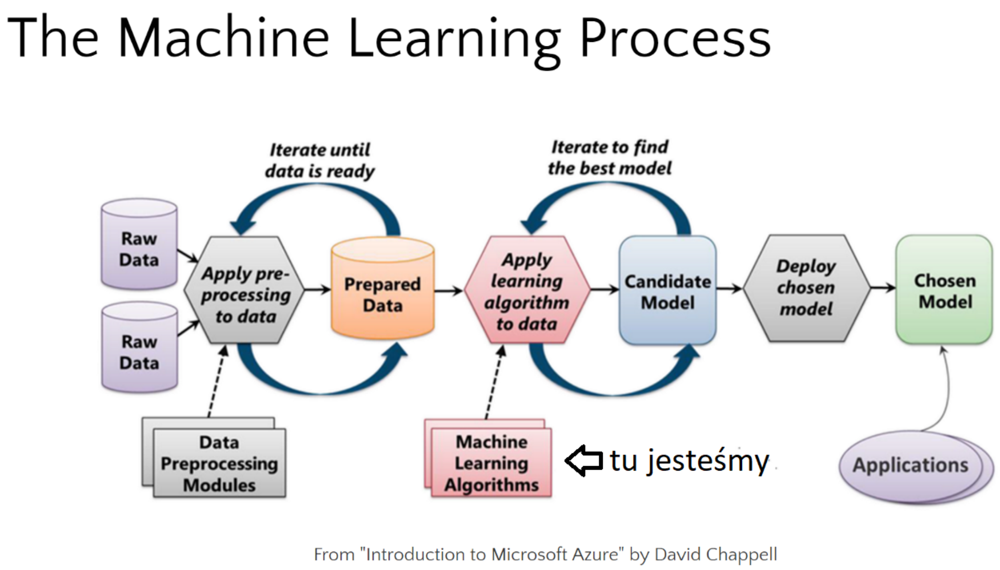
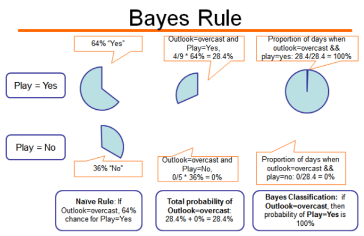
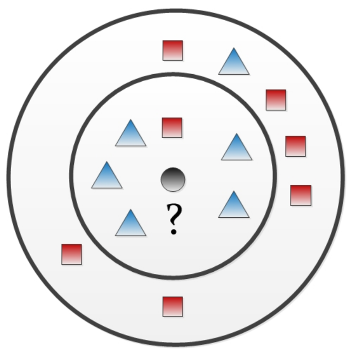
Ćwiczenie - 5 Sztuczna Inteligencja

Budowanie modeli klasyfikacji: k-NN oraz Naiwny klasyfikator Bayesa







Krótkie wprowadzenie (Short introduction)

W ćwiczeniu wprowadzamy kolejne dwa modele klasyfikacji, techniki modelowania procesu decyzyjnego (in the task we have two more classifiers to implement) - metodę $k-NN$ (k Nearest Neighbours) oraz Naiwny Klasyfiaktor Bayesa (Naive Bayes classifier). Przejdźmy do opisania zadania i wprowadzenia teoretycznego z przykładami. (Let us to show theory and toy examples of both algorithms)

Zadania do wykonania (Set of tasks to do)

1) Tworzymy na pulpicie katalog w formacie Imię_nazwisko, w którym umieszczamy wszystkie pliki związane z ćwiczeniem (Create the folder to collect your all files),

2) Czytamy teorię związaną z klasyfikacją metodą k najbliższych sąsiadów ($k-NN$), oraz Naiwnym Klasyfikatorem Bayesa, w razie problemów ze zrozumieniem, analizujemy przykłady na kartce. (Read the theory about k-NN and Naive Bayes classifer, in case of problems resolve simple examples on the piece of paper),

3) Generujemy system treningowy (australian_TRN.txt) oraz testowy (australian_TST.txt) za pomocą programu data_splitter.exe. (Generate test and training decision systems using data_splitter.exe program)

4) Otrzymany system testowy klasyfikujemy systemem treningowym metodą $2-NN$, za pośrednictwem programu napisanego w dowolnym języku programowania. Implementujemy algorytm dla: Metryki Euklidesa, Canberra, Czebyszewa, Manhattan oraz dla Bezwzględnego współczynnika korelacji Pearsona. (Classifiy your tst system based on trn system with use of k-NN classifer - prepare implementation in chosen programming language)

5) Implementujemy w dowolnym języku programowania (preferowany C++) klasyfikator Bayesa. Wynikiem działania programu powinny być dwa pliki:

a) dec_bayes.txt - zawierający podjęte decyzje dla obiektów systemu australian_TST.txt na podstawie obiektów systemu australian_TRN.txt

b) acc_bayes.txt - zawierający dwie wartości: Accuracy Globalne klasyfikacji i Accuracy Zbalansowane klasyfikacji.

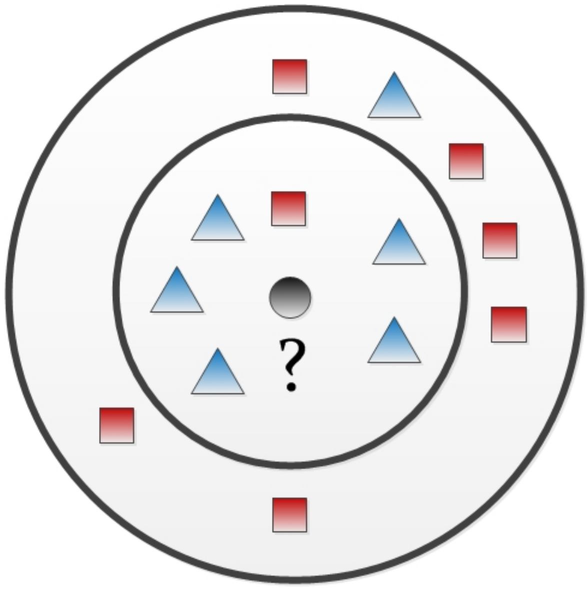
(Classify your tst system based on trn system with use of Naive Bayes classifier - prepare implementation in chosen programming language)

6) W przypadku programowania w C++, ułatwieniem może być użycie programu demonstracyjny znajdujący się na stronie http://wmii.uwm.edu.pl/∼artem w zakładce Dydaktyka/Sztuczna Inteligencja (Its the link to exemplary auxiliary program).

Klasyfikator k-NN - część teoretyczna

Przyjmując, że nie znamy figury w centrum Rys. 1 możemy przeprowadzać dedukcję na podstawie obserwacji figur stojących w jej sąsiedztwie prowadzącą do jej zdefiniowania. Przykładowo, gdy rozważamy po dwie najbliższe figury spośród trójkątów i kwadratów, widzimy, że sumaryczna odległość dwóch trójkątów od figury nieznanej jest mniejsza niż odległość pary kwadratów, stąd możemy przypuszczać, że naszą ukrytą figurą jest trójkąt. Tego typu sposób klasyfikacji nazywamy metodą k najbliższych sąsiadów, w naszym przykładzie rozważaliśmy k=2 w sensie wyboru po dwa obiekty z każdej dostępnej klasy obiektów.

• Dla danego systemu testowy (X, A, c) i treningowego (Y, A, c), gdzie X, Y to odpowiednio uniwersum obiektów testowych i treningowych, $A = (a_1, a_2, \ldots, a_n)$ jest zbiorem atrybutów warunkowych, $c \in D = \{c_1, c_2, \ldots, c_m\}$ jest atrybutem decyzyjnym.



Rysunek 1: Wizualizacja problemu klasyfikacji metodą k najbliższych sąsiadów

Dla obiektów $x \in X, y \in Y$ postaci,

$$x = a_1(x)\ a_2(x) \dots a_n(x)\ c(x)$$

$$y = a_1(y)\ a_2(y) \dots a_n(y)\ c(y)$$

zdefiniujmy podstawowe metryki,

Metryka Manhattan przedstawia się następująco,

$$d(x, y) = \sum_{i=1}^{n} |a_i(x) - a_i(y)|$$

Metryka Euklidesowa szczególnym przypadkiem metryki Minkowskiego,

$$d(x, y) = \sqrt{(a_1(x) - a_1(y))^2 + (a_2(x) - a_2(y))^2 + ... + (a_n(x) - a_n(y))^2}$$

czyli zapisując ogólnie:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (a_i(x) - a_i(y))^2}$$

Metryka Canberra jest postaci,

$$d(x, y) = \sum_{i=1}^{n} |\frac{a_i(x) - a_i(y)}{a_i(x) + a_i(y)}|$$

Metryka Czebyszewa określana jest wzorem,

$$d(x, y) = max(|a_i(x) - a_i(y)|),\ dla\ i = 1, 2, \dots, n$$

Bezwzględny współczynnik korelacji Pearsona może być używany w poniższy sposób,

$$d(x, y) = 1 - |r_{x,y}|$$

$$r_{x,y} = \frac{1}{n} * \sum_{i=1}^{n} (\frac{a_i(x) - \overline{x}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i(x) - \overline{x})^2}})(\frac{a_i(y) - \overline{y}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i(y) - \overline{y})^2}})$$

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} a_i(x), \overline{y} = \frac{1}{n}\sum_{i=1}^{n} a_i(y)$$

Procedura algorytmu k-NN z równym uwzględnianiem klas decyzyjnych

• Wczytujemy system testowy (X, A, c) i treningowy (Y, A, c), gdzie X, Y to odpowiednio uniwersum obiektów testowych i treningowych, $A = (a_1, a_2, \ldots, a_n)$ jest zbiorem atrybutów warunkowych, $c \in D = \{c_1, c_2, \ldots, c_m\}$ jest atrybutem decyzyjnym.

• Ustalamy metrykę d liczenia odległości między obiektami, oraz liczbę najbliższych sąsiadów decydujących o klasyfikacji k,

• Klasyfikujemy wszystkie obiekty testowe za pomocą k najbliższych obiektów, każdej z klas systemu treningowego, (decyzję przekazuje klasa, której obiekty są najbliżej testowego w sensie metryki d),

• Po zakończeniu klasyfikacji, tworzymy Macierz Predykcji, zawierającą informacje o jakości klasyfikacji systemu testowego X:

Parametry mówiące o jakości przeprowadzonej klasyfikacji, które należy umieścić w raporcie klasyfikacji (w Macierzy Predykcji) są definiowane następująco:

Dla $\bigwedge_{c \in D}$

$$acc_c = \frac{liczba\ obiektów\ poprawnie\ sklasyfikowanych\ w\ klasie\ decyzyjnej\ c}{liczba\ obiektów\ chwyconych\ w\ klasie\ c}$$

$$cov_c = \frac{liczba\ obiektów\ chwyconych\ w\ klasie\ c}{liczba\ obiektów\ klasy\ c}$$

$$TPR_c = \frac{x}{x + liczba\ obiektów\ z\ pozostałych\ klas\ błędnie\ trafiających\ do\ klasy\ c}$$

przyjmujemy, że $x = liczba\ obiektów\ poprawnie\ sklasyfikowanych\ w\ klasie\ decyzyjnym\ c$

Ostatecznie wyliczamy wartości globalne, które umieszczamy pod Macierzą Predykcji,

$$acc_{global} = \frac{ilość\ obiektów\ poprawnie\ sklasyfikowanych\ w\ całym\ systemie\ TST}{ilość\ obiektów\ chwyconych\ w\ systemie\ TST}$$

$$cov_{global} = \frac{ilość\ obiektów\ chwyconych\ w\ całym\ systemie\ TST}{ilość\ obiektów\ systemu\ TST}$$

Przykładowa klasyfikacja 2-NN Wczytujemy system testowy postaci,

Tabela 1: System Testowy (X, A, c)

	a_1	a_2	a_3	a_4	c
x_1	2	4	2	1	4
x_2	1	2	1	1	2
x_3	9	7	10	7	4
x_4	4	4	10	10	2

oraz system treningowy

Tabela 2: System Treningowy (Y, A, c)

	a_1	a_2	a_3	a_4	c
y_1	1	3	1	1	2
y_2	10	3	2	1	2
y_3	2	3	1	1	2
y_4	10	9	7	1	4
y_5	3	5	2	2	4
y_6	2	3	1	1	4

Ustalmy k=2 i d jako metrykę Euklidesa

Metryka Euklidesa działa następująco, dla obiektów

$$x = a_1(x)\ a_2(x) \dots a_n(x)\ c(x)$$

$$y = a_1(y)\ a_2(y) \dots a_n(y)\ c(y)$$

$$d(x, y) = \sqrt{(a_1(x) - a_1(y))^2 + (a_2(x) - a_2(y))^2 + ... + (a_n(x) - a_n(y))^2}$$

czyli zapisując ogólnie:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (a_i(x) - a_i(y))^2}$$

Przechodzimy do klasyfikacji obiektów testowych:

Dla x_1 2 4 2 1 4

$d(x1, y1) = \sqrt{(2-1)^2 + (4-3)^2 + (2-1)^2 + (1-1)^2} = \sqrt{3}$
$d(x1, y2) = \sqrt{65}$
$d(x1, y3) = \sqrt{2}$
$d(x1, y4) = \sqrt{114}$
$d(x1, y5) = \sqrt{3}$

$d(x1, y6) = \sqrt{2}$

Dwóch najbliższych sąsiadów obiektu testowego x_1 w klasie decyzyjnej 2 to y_3, y_1
Klasa 2 głosuje z mocą $\sqrt{2} + \sqrt{3}$
Najbliższymi sąsiadami x_1 w klasie decyzyjnej 4 są y_6, y_5
Klasa 4 głosuje z mocą $\sqrt{2} + \sqrt{3}$
$\sqrt{2} + \sqrt{3} = \sqrt{2} + \sqrt{3}$
Stąd obiekt x_1 nie jest chwytany, nie jesteśmy w stanie powiedzieć, która klasa jest bliżej w sensie dwóch najbliższych sąsiadów.

Dla x_2 1 2 1 1 2

$d(x2, y1) = 1$
$d(x2, y2) = \sqrt{84}$
$d(x2, y3) = \sqrt{2}$
$d(x2, y4) = \sqrt{166}$
$d(x2, y5) = \sqrt{15}$
$d(x2, y6) = \sqrt{2}$

Klasa 2 głosuje z mocą $1 + \sqrt{2}$
Klasa 4 głosuje z mocą $\sqrt{2} + \sqrt{15}$
$1 + \sqrt{2} < \sqrt{2} + \sqrt{15}$
Obiekt x_2 otrzymuje decyzję 2, jest poprawnie sklasyfikowany.

Dla x_3 9 7 10 7 4

$d(x3, y1) = \sqrt{197}$
$d(x3, y2) = \sqrt{117}$
$d(x3, y3) = \sqrt{182}$
$d(x3, y4) = \sqrt{50}$
$d(x3, y5) = \sqrt{129}$
$d(x3, y6) = \sqrt{182}$

Klasa 2 głosuje z mocą $\sqrt{117} + \sqrt{182}$
Klasa 4 głosuje z mocą $\sqrt{50} + \sqrt{129}$
$\sqrt{50} + \sqrt{129} < \sqrt{117} + \sqrt{182}$
Obiekt x_3 otrzymuje decyzję 4, jest poprawnie sklasyfikowany.

Dla x_4 4 4 10 10 2

$d(x4, y1) = \sqrt{172}$
$d(x4, y2) = \sqrt{182}$
$d(x4, y3) = \sqrt{167}$
$d(x4, y4) = \sqrt{151}$
$d(x4, y5) = \sqrt{130}$
$d(x4, y6) = \sqrt{167}$

Klasa 2 głosuje z mocą $\sqrt{167} + \sqrt{172}$

Klasa 4 głosuje z mocą $\sqrt{130} + \sqrt{151}$
$\sqrt{130} + \sqrt{151} < \sqrt{167} + \sqrt{172}$
Obiekt x_4 otrzymuje decyzję 4, jest błędnie sklasyfikowany.

Podsumowując klasyfikację:
Obiekt x_1 nie jest chwytany
Obiekt x_2 otrzymuje decyzję 2, jest poprawnie sklasyfikowany
Obiekt x_3 otrzymuje decyzję 4, jest poprawnie sklasyfikowany
Obiekt x_4 otrzymuje decyzję 4, jest błędnie sklasyfikowany.

Macierz Predykcji, stanowiąca raport z klasyfikacji możemy zobaczyć w Tab. 3.

Tabela 3: Macierz Predykcji

	2	4	*No. of obj.*	*Accuracy*	*Coverage*
2	1	1	2	0.5	1.0
4	0	1	2	1.0	0.5
True Positive Rate	1.0	0.5			

Naiwny Klasyfikator Bayesa - część teoretyczna

Klasyfikator z rodziny metod probabilistycznych. Wymaga założenia, że atrybuty systemu decyzyjnego są niezależne od siebie, takie założenie często jest niezgodne z sytuacją rzeczywistą, stąd klasyfikator nazywamy naiwnym. Pomimo opisanego założenia klasyfikator działa w wielu przypadkach zaskakująco dobrze. Istnieje teoria, że tego typu klasyfikacją może kierować się nasz umysł - patrz artykuł:
http://reverendbayes.wordpress.com/2008/05/29/bayesian-theory-in-new-scientist/.

Załóżmy, że mamy dane systemy decyzyjne: treningowy (U_{trn}, A, d) oraz testowy (U_{tst}, A, d), gdzie U jest zbiorem obiektów, $A = \{a_1, a_2, ..., a_n\}$ zbiorem atrybutów warunkowych, $d \in D = \{d_1, d_2, ..., d_k\}$ atrybutem decyzyjnym.

Klasyfikacja obiektu testowego $v \in U_{tst}$ opisanego jako $a_1(v), a_2(v), ..., a_n(v)$ polega na obliczeniu dla każdej z klas decyzyjnych parametru:

$$P(d = d_i | b_1 = a_1(v), b_2 = a_2(v), ..., b_n = a_n(v))$$

gdzie w postaci Twierdzenia Bayesa mamy,

$$\frac{P(b_1 = a_1(v), b_2 = a_2(v), ..., b_n = a_n(v) | d = d_i) * P(d = d_i)}{P(b_1 = a_1(v), b_2 = a_2(v), ..., b_n = a_n(v))}$$

Mianownik możemy pominąć ponieważ jest stały dla wszystkich klas decyzyjnych. Przy założeniu że atrybuty są niezależne licznik możemy obliczyć jako

$$P(b_1 = a_1(v), b_2 = a_2(v), ..., b_n = a_n(v) | d = d_i) * P(d = d_i) = P(d = d_i) * \prod_{m=1}^{n} P(b_m = a_m(v) | d = d_i$$

W praktyce możemy zastosować oszacowanie częściowe,

$$P(b_m = a_m(v)|d = d_i) = \frac{\text{liczba wystąpien wartosci } b_m = a_m(v) \text{ w klasie } d_i}{\text{liczność klasy treningowej } d_i}$$

Każda klasa głosuje wartością parametru:

$$Param_{d=d_i} = P(d = d_i) * \prod_{m=1}^{n} P(b_m = a_m(v)|d = d_i)$$

W tym wariancie, w przypadku gdy dla klasy d_i uzyskamy wartość $P(b_m = a_m(v)|d = d_i) = 0$, wyszukujemy w pozostałych klasach najmniejszej niezerowej wartości $P(b_m = a_m(v)|d = d_j)$ i przypisujemy ją do $P(b_m = a_m(v)|d = d_i)$ po odpowiednim zmniejszeniu (np dzieląc przez 2). Jeżeli klas z zerowym wystąpieniem $b_m = a_m(v)$ jest więcej, każdej z nich przypisujemy tę zmniejszoną wartość. Innym sposobem na radzenie sobie z problemem zerowych wartości $P(b_m = a_m(v)|d = d_i)$ jest faworyzowanie pozostałych klas zawierających wartość $b_m = a_m(v)$, następuje zwiększanie liczników o jeden podczas obliczania prawdopodobieństwa. W przypadku gdy żadna z klas nie zawiera wartości $b_m = a_m(v)$, prawdopodobieństwa zerowe są pomijane w iloczynie.

Aby zapobiec problemowi małych bliskich zeru liczb, możemy poszczególne prawdopodobieństwa logarytmować. W praktycznym zastosowaniu, dopuszcza się użycie sumy prawdopodobieństw, (ponieważ mnożenie przez małe wartości może doprowadzić do zmniejszenia parametru klasy do zera, przy ograniczonej dokładności) w naszym wariancie algorytmu każda klasa decyzyjna będzie głosowała za pomocą parametru:

$$Param_{d=d_i} = P(d = d_i) * \sum_{m=1}^{n} P(b_m = a_m(v)|d = d_i)$$

W przypadku problemu z zerową licznością deskryptora $b_m = a_m(v)$, stosujemy faworyzowanie klas, podobnie jak w wariancie z iloczynem prawdopodobieństw.

Gdy podczas klasyfikacji parametry klas są jednakowe, konflikt rozwiązujemy nadając obiektowi testowemu losową decyzję.

W przypadku gdy atrybuty są ciągłe, oraz zakładając że mają rozkład normalny, poszczególne prawdopodobieństwa $P(b_m = a_m(v)|d = d_i)$ szacujemy za pomocą funkcji Gaussa.

$$f(x) = \frac{1}{\sqrt{(2 * \pi * \sigma_c^2)}} * e^{\frac{-(x-\mu_c)^2}{2*\sigma_c^2}}$$

Do jej obliczenia potrzebujemy średnie z klas oraz wariancje wartości w klasach.

$$\mu_c = \frac{\sum_{i=1}^{licznosc\ klasy\ c} a(v_i)}{licznosc\ klasy\ c}$$

$$\sigma_c^2 = \frac{1}{licznosc\ klasy\ c} * \sum_{i=1}^{licznosc\ klasy\ c} (a(v_i) - \mu_c)^2$$

Przykład działania Naiwnego Klasyfikatora Bayesa

Wczytujemy system testowy (problemy do rozwiązania z ukrytymi decyzjami eksperta) postaci,

Tabela 4: System Testowy (X, A, c)

	a_1	a_2	a_3	a_4	c
x_1	2	4	2	1	4
x_2	1	2	1	1	2
x_3	9	7	10	7	4
x_4	4	4	10	10	2

oraz system treningowy (bazę wiedzy służącą do rozwiązywania problemów)
Widzimy, że $P(c = 2) = \frac{3}{6} = \frac{1}{2}$, $P(c = 4) = \frac{1}{2}$.

Rozpoczynamy od klasyfikacji obiektu testowego
x_1 2 4 2 1 4

Wyliczamy $Param_{c=2} = P(c = 2) * \sum_{i=1}^{4} P(a_i = v_i|c = 2)$,
$P(a_1 = 2|c = 2) = \frac{1}{3}$
$P(a_2 = 4|c = 2) = \frac{0}{3}$ nie da się obsłużyć tej wartości, żadna z klas nie zawiera $a_2 = 4$
$P(a_3 = 2|c = 2) = \frac{1}{3}$
$P(a_4 = 1|c = 2) = \frac{3}{3}$

Tabela 5: System Treningowy (Y, A, c)

	a_1	a_2	a_3	a_4	c
y_1	1	3	1	1	2
y_2	10	3	2	1	2
y_3	2	3	1	1	2
y_4	10	9	7	1	4
y_5	3	5	2	2	4
y_6	2	3	1	1	4

Ostatecznie $Param_{c=2} = \frac{1}{2} * (\frac{1}{3} + \frac{0}{3} + \frac{1}{3} + \frac{3}{3}) = \frac{5}{6}$;

oraz $Param_{c=4} = P(c = 4) * \sum_{i=1}^{4} P(a_i = v_i|c = 4)$,
$P(a_1 = 2|c = 4) = \frac{1}{3}$
$P(a_2 = 4|c = 4) = \frac{0}{3}$ nie da się obsłużyć tej wartości, żadna z klas nie zawiera $a_2 = 4$
$P(a_3 = 2|c = 4) = \frac{1}{3}$
$P(a_4 = 1|c = 4) = \frac{2}{3}$
Ostatecznie $Param_{c=4} = \frac{1}{2} * (\frac{1}{3} + \frac{0}{3} + \frac{1}{3} + \frac{2}{3}) = \frac{2}{3}$;

$Param_{c=2} > Param_{c=4}$, obiekt x_1 dostaje decyzję 2, ta decyzja nie jest zgodna z ukrytą decyzją eksperta stąd obiekt jest błędnie sklasyfikowany,

Podczas klasyfikacji pierwszego obiektu napotykamy liczność zero dla $a2 = 4$, której nie możemy obsłużyć, ponieważ żadna z istniejących klas nie zawiera wartości $a2 = 4$.

Przejdźmy do klasyfikacji obiektu testowego
x_2 1 2 1 1 4

Wyliczamy $Param_{c=2} = P(c = 2) * \sum_{i=1}^{4} P(a_i = v_i|c = 2)$,
$P(a_1 = 1|c = 2) = \frac{1}{3}$ zwiększamy licznik o 1, ponieważ $P(a_1|c = 4) = 0$, $P(a_1 = 1|c = 2) = \frac{2}{3}$
$P(a_2 = 2|c = 2) = \frac{0}{3}$ nie da się obsłużyć tej wartości, żadna z klas nie zawiera $a_2 = 2$
$P(a_3 = 1|c = 2) = \frac{1}{3}$
$P(a_4 = 1|c = 2) = \frac{3}{3}$
Ostatecznie $Param_{c=2} = \frac{1}{2} * (\frac{2}{3} + \frac{0}{3} + \frac{1}{3} + \frac{3}{3}) = 1$;

oraz $Param_{c=4} = P(c = 4) * \sum_{i=1}^{4} P(a_i = v_i|c = 4)$,
$P(a_1 = 1|c = 4) = \frac{0}{3}$ w tej sytuacji zwiększam o 1 licznik $P(a_1 = 1|c = 2)$, faworyzując klasy zawierające przynajmniej jedną wartość $a_1 = 1$
$P(a_2 = 2|c = 4) = \frac{0}{3}$ nie da się obsłużyć tej wartości, żadna z klas nie zawiera $a_2 = 2$
$P(a_3 = 1|c = 4) = \frac{1}{3}$
$P(a_4 = 1|c = 4) = \frac{2}{3}$
Ostatecznie $Param_{c=2} = \frac{1}{2} * (\frac{0}{3} + \frac{0}{3} + \frac{1}{3} + \frac{2}{3}) = \frac{1}{2}$;

$Param_{c=2} > Param_{c=4}$, obiekt x_2 dostaje decyzję 2, ta decyzja jest zgodna z ukrytą decyzją eksperta stąd obiekt jest poprawnie sklasyfikowany,

Przejdźmy do klasyfikacji obiektu testowego
x_3 9 7 10 7 4

Wyliczamy $Param_{c=2} = P(c=2) * \sum_{i=1}^{4} P(a_i = v_i | c = 2)$,
$P(a_1 = 9|c = 2) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_2 = 7|c = 2) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_3 = 10|c = 2) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_4 = 7|c = 2) = \frac{0}{3}$ nie da się obsłużyć tej wartości
Ostatecznie $Param_{c=2} = \frac{1}{2} * (\frac{0}{3} + \frac{0}{3} + \frac{0}{3} + \frac{0}{3}) = 0$;

oraz $Param_{c=4} = P(c=4) * \sum_{i=1}^{4} P(a_i = v_i | c = 4)$,
$P(a_1 = 9|c = 4) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_2 = 7|c = 4) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_3 = 10|c = 4) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_4 = 7|c = 4) = \frac{0}{3}$ nie da się obsłużyć tej wartości
Ostatecznie $Param_{c=2} = \frac{1}{2} * (\frac{0}{3} + \frac{0}{3} + \frac{0}{3} + \frac{0}{3}) = 0$;

$Param_{c=2} == Param_{c=4}$, obiekt x_3 dostaje decyzję losową $los(2, 4) = 4$, ta decyzja jest zgodna z ukrytą decyzją eksperta stąd obiekt jest poprawnie sklasyfikowany,

Przejdźmy do klasyfikacji obiektu testowego
x_4 4 4 10 10 4

Wyliczamy $Param_{c=2} = P(c=2) * \sum_{i=1}^{4} P(a_i = v_i | c = 2)$,
$P(a_1 = 4|c = 2) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_2 = 4|c = 2) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_3 = 10|c = 2) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_4 = 10|c = 2) = \frac{0}{3}$ nie da się obsłużyć tej wartości
Ostatecznie $Param_{c=2} = \frac{1}{2} * (\frac{0}{3} + \frac{0}{3} + \frac{0}{3} + \frac{0}{3}) = 0$;

oraz $Param_{c=4} = P(c=4) * \sum_{i=1}^{4} P(a_i = v_i | c = 4)$,
$P(a_1 = 4|c = 4) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_2 = 4|c = 4) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_3 = 10|c = 4) = \frac{0}{3}$ nie da się obsłużyć tej wartości
$P(a_4 = 10|c = 4) = \frac{0}{3}$ nie da się obsłużyć tej wartości
Ostatecznie $Param_{c=2} = \frac{1}{2} * (\frac{0}{3} + \frac{0}{3} + \frac{0}{3} + \frac{0}{3}) = 0$;

$Param_{c=2} == Param_{c=4}$, obiekt x_4 dostaje decyzję losową $los(2, 4) = 4$, ta decyzja nie jest zgodna z ukrytą decyzją eksperta stąd obiekt jest poprawnie sklasyfikowany,

Ostatecznie wyliczamy parametry:

$$Gobal\_Accuracy = \frac{\text{liczba obiektów tst poprawnie sklasyfikowanych w całym systemie}}{\text{liczba obiektów sklasyfikowanych w całym systemie}}$$

$$Balanced\_Accuracy = \frac{\sum_{i=1}^{liczba\ klas} \frac{\text{liczba obiektów tst poprawnie sklasyfikowanych w klasie } c_i}{\text{liczba obiektów sklasyfikowanych w klasie } c_i}}{liczba\ klas}$$

$$Gobal\_Accuracy = \frac{2}{4} = \frac{1}{2}$$

$$Balanced\_Accuracy = \frac{\frac{1}{2} + \frac{1}{2}}{2} = \frac{1}{2}$$

Obiekt tst	Ukryta decyzja eksperta	Decyzja naszego kasyfikatora
x_1	4	2
x_2	2	2
x_3	4	4
x_4	2	4

***

the source of image from the first page: http://www.simafore.com/