Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Magdalena Waśniowska
Nr albumu: 305964

Analiza skupień a problem dynamicznych kręgów zbiorowisk roślinnych

Praca licencjacka
na kierunku MATEMATYKA

Praca wykonana pod kierunkiem
dra inż. Przemysława Biecka
Instytut Matematyki Stosowanej i Mechaniki

Lipiec 2013

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora (autorów) pracy

Streszczenie

W pracy przedstawiono algorytmy: medoidów, Isomap, Isopam, a także sylwetkę podziału. Do ilustracji ich działania wykorzystano dane o zbiorowiskach roślinnych z doliny Bogdanki w Poznaniu, które przeanalizowano za pomocą pakietu statystycznego R.

Słowa kluczowe

Isopam, Isomap, algorytm medoidów, sylwetka podziału, zbiorowisko roślinne

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.2 Statystyka

Klasyfikacja tematyczna

62H30, 62P10

Tytuł pracy w jzyku angielskim

Cluster analysis and the problem of the dynamic vegetation circles

Spis treści

Spis rysunków

Spis tablic

Wprowadzenie

Grupowanie, czyli analiza skupień jest metodą polegającą na podziale zbioru obiektów na jak najbardziej jednorodne klasy. Jest ona pomocna np. przy redukcji dużej liczby danych początkowych do kilku podstawowych grup, traktowanych później jako odrębne przedmioty w dalszej pracy. Analiza skupień ma szerokie zastosowanie w wielu dziedzinach nauki m.in. biologii. Była ona niezbędna w pracy badawczej Pani Moniki Zgrabczyńskiej, doktorantki Zakładu Ekologii Roślin i Ochrony Środowiska wydziału Biologii Uniwersytetu im. Adama Mickiewicza w Poznaniu. Celem niniejszej pracy jest podział 85 zbiorowisk roślinnych na 5 rozłącznych grup, w których obiekty będą jak najbardziej podobne do wcześniej wybranych elementów wyróżnionych, a w różnych jak najbardziej odmienne. Uzyskanie jednorodnych klas w zbiorze danych, ma ułatwić Pani Monice Zgrabczyńskiej wyodrębnienie ich zasadniczych wspólnych cech.

Praca złożona jest z trzech rozdziałów. W pierwszym z nich zostanie omówiona teoria dotycząca algorytmu Isopam, na podstawie którego zostanie dokonany ostateczny podział danych. Na początku zostaną jednak przedstawione dwa inne algorytmy: Isomap i PAM, będące częścią składową Isopam. Pierwszy z nich odpowiada za obliczenie rzeczywistych odległości między obiektami z próby oraz przekształcenie początkowej przestrzeni danych, drugi za podział zbioru danych na rozłączne skupienia. Dodatkowo w tym rozdziale zostanie przedstawiona sylwetka podziału, metoda która pomaga dokonać oceny jakości dokonanej klasyfikacji. W drugim rozdziale opisane zostaną funkcje z pakietu R służące do analizy danych przy pomocy algorytmów Isomap, PAM, Isopam, a także do stworzenia wykresu sylwetki podziału. W ostatnim rozdziale zostaną poddana analizie dane o zbiorowiskach roślinnych z doliny Bogdanki w Poznaniu zebrane i opracowane przez Panią Monikę Zgrabczyńską, we współpracy z którą została dokonana analiza otrzymanych wyników.

Rozdział 1

Teoria

W poniższym rozdziale zostanie omówiony Isopam (ang. *Isometric Feature Mapping and Partitioning Around Medoids*), metoda podziału zbioru danych na rozłączne skupienia. W dużej mierze korzysta ona z dwóch innych algorytmów Isomap (ang. *Isometric Feature Mapping*) oraz PAM (ang. *Partitioning Around Medoids*), dlatego zostaną one najpierw wyjaśnione. Natomiast w dalszej części zostanie przedstawiona sylwetka podziału (ang. *silhouette*), czyli funkcja pomagająca ocenić poprawność dokonanego podziału.

1.1. Algorytm Isomap

Zbiór danych $X = \{x_1, x_2, \ldots, x_n\}$ stanowiący n-obserwacji, z których każda jest opisana przez d cech w skali liczbowej ($n, d \in \mathbb{N}_+$) można przedstawić jako n punktów w przestrzeni $\mathbb{R}^d$. Ponadto punkty te stanowią pewną strukturę, o której można założyć, że jest rozmaitością M wymiaru k.

Definicja 1 (Homeomorficzność) *Dwie przestrzenie nazwane są homeomorficzne jeśli istnieje homeomorfizm (funkcja różnowartościowa, „na", ciągła, której odwrotność też jest ciągła) przekształcający jedną na drugą. [3]*

Definicja 2 (Rozmaitość) *Rozmaitością M wymiaru k nazywa się taki podzbiór przestrzeni, którego każdy punkt ma otwarte otoczenie homeomorficzne z przestrzenią euklidesową $\mathbb{R}^k$. [3]*

Im większa liczba obserwacji tym lepiej można opisać M, która lokalnie przypomina przestrzeń euklidesową, ale globalnie może być bardziej zróżnicowana np. nieliniowa. Jednak im większe d tym trudniejsze jest zilustrowanie tej rozmaitości. Rozwiązanie tego problemu polega na zamianie wejściowej przestrzeni danych na taką, której wymiar jest niższy. Isomap jest metodą, która ma na celu zredukować liczbę wymiarów danych wejściowych, tak by każdy element z X był opisany przez mniejszą liczbę cech, jednak w taki sposób by ich wspólna struktura została zachowana.

Isomap zdecydowanie lepiej (w porównaniu do powszechnie używanych metod) radzi sobie z odtworzeniem nieliniowych struktur. [10] Jest tak za sprawą sposobu oszacowania rzeczywistych odległości między punktami. W przypadku obserwacji umieszczonych blisko siebie w przestrzeni, zwykła odległość rozumiana jako metryka na przestrzeni danych jest jej dobrym przybliżeniem. Natomiast w przypadku oddalonych punktów odległość może być aproksymowana poprzez długość ścieżki wiodącej między poszczególnymi elementami, z przystankami będącymi innymi elementami.

1.1.1. Opis algorytmu

Celem Isomap jest znalezienie niskowymiarowej reprezentacji danych, w której odległości rzeczywiste między elementami z próby są jak najbliższe tym z oryginalnej przestrzeni wysokowymiarowej.

Poniżej wprowadzone oznaczenia, będą wykorzystywane w dalszej części rozdziału:

- n - liczba obserwacji wejściowych,
- d - wymiar przestrzeni danych wejściowych,
- $X = \{x_1, x_2, \ldots, x_n\}$ - zbiór danych,
- x_i - element z próby przed zastosowaniem Isomap, $x_i \in \mathbb{R}^d$,
- $t < d$ - zredukowany wymiar przestrzeni danych wejściowych,
- $Y = \{y_1, y_2, \ldots, y_n\}$ - niskowymiarowy reprezentant zbioru danych,
- y_i - element z próby po zastosowaniu Isomap, $y_i \in \mathbb{R}^t$,
- $d_s(i, j)$ - odległość od elementu i-tego do j-tego w przestrzeni $\mathbb{R}^s$ z ustaloną metryką.

Dla Isomap elementy z próby X są punktami w przestrzeni $\mathbb{R}^d$ umieszczonymi na rozmaitości M. Algorytm na początku oszacowuje jej wewnętrzną strukturę na podstawie ścieżek poprowadzonych z każdego punktu do elementów sąsiadujących z nim, a następnie przeprowadza skalowanie wielowymiarowe, by na końcu wskazać punkty Y z przestrzeni niższego wymiaru.

Cały algorytm, którym posługuje się Isomap można podsumować w trzech krokach, które w dalszej części rozdziału zostaną wyjaśnione.

1. Dla każdego elementu z próby znajdź jego sąsiadów (w zależności od wersji algorytmu k lub ε najbliższych) i oblicz odległości między nimi.
2. Znajdź najkrótsze ścieżki pomiędzy wszystkimi elementami próby i oblicz ich długość.
3. Zastosuj skalowanie wielowymiarowe do otrzymanej w kroku 2 macierzy odległości.

Krok 1

Aby policzyć rzeczywiste odległości między elementami próby, potrzeba na początku określić, które elementy leżą blisko siebie na rozmaitości M, tak że ich odległość może być przybliżona odległością z danej przestrzeni. Należy również określić, które są daleko od siebie i trzeba w inny sposób określić odległość je dzielącą. W pierwszym etapie Isomap ustala sąsiadów każdego elementu z X. Są dwie możliwości wyboru sąsiadów. Odpowiadają za to odpowiednie wersje algorytmu: ε-Isomap oraz k-Isomap.

Definicja 3 (Sąsiad w wersji ε-Isomap) *Sąsiadem x_j elementu x_i $(i \neq j)$ nazywa się taki element z X, który spełnia*

$$x_j \in B(x_i, \varepsilon) \quad ,$$

gdzie $B(x_i, \varepsilon)$-kula o środku w x_i i promieniu ε. [10]

Definicja 4 (Sąsiad w wersji k-Isomap) *Sąsiadem x_j elementu x_i $(i \neq j)$ nazywa się taki element z X, który jest wśród k $(k < n)$ elementów z X, położonych najbliżej x_i względem ustalonej odległości. [10]*

Sąsiedztwo jest reprezentowane przez graf ważony G. Wierzchołki to elementy z X połączone krawędziami wyłącznie ze swoimi sąsiadami. Waga $w(i,j)$ pomiędzy wierzchołkiem x_i oraz x_j jest równa $d_d(i,j)$, ponieważ zostały one zakwalifikowane jako leżące dostatecznie blisko siebie i ich odległość będzie mierzona w zwykły sposób. Elementy, które nie są sąsiadami, czyli leżą daleko od siebie, nie są połączone krawędziami w grafie G.

Krok 2

W drugim etapie następuje oszacowanie rzeczywistych odległości między wszystkimi parami punktów rozmieszczonymi na rozmaitości M na podstawie otrzymanego w kroku 1 grafu G. Jeśli elementy leżały dostatecznie blisko, to zostały zakwalifikowane jako sąsiedzi i odległość między nimi jest już obliczona. Jeśli natomiast leżały daleko od siebie, to szukana jest najkrótsza ścieżka w grafie G, po której można przejść od jednego elementu do drugiego. Wynikiem kroku 2 jest macierz odległości $A = \{a_{i,j}\}_{i,j=1,2,\ldots,n}$. Wyraz $a_{i,j}$ określa odległość między elementem x_i a x_j. Macierz A jest symetryczna, ma wszystkie wyrazy nieujemne oraz zera na głównej przekątnej.

Do znalezienia najkrótszej drogi między elementami, które nie są ani ε-sąsiadami, ani k-sąsiadami może posłużyć algorytm Floyd'a.

Algorytm Floyd'a

Na wstępie macierz A określona jest następująco:

$$a_{i,j} = \begin{cases} w(i,j) & \text{gdy } x_i \text{ oraz } x_j \text{ są bezpośrednio połączone w grafie } G, \\ \infty & \text{w przeciwnym przypadku.} \end{cases}$$

Algorytm posługuje się rekurencyjną formułą. Poniżej przyjęte zostało oznaczenie, w którym $NS(x_i, x_j, k)$ jest długością najkrótszej ścieżki łączącej wierzchołki x_i oraz x_j poprowadzonej po drodze złożonej wyłącznie z wierzchołków należących do zbioru $\{x_1, x_2, \ldots, x_k\}$.

Znając $NS(x_i, x_j, k)$ celem jest policzyć $NS(x_i, x_j, k+1)$. Ale dla każdej pary wierzchołków z G tą najkrótszą ścieżką jest albo ścieżka poprowadzona po wierzchołkach ze zbioru $\{x_1, x_2, \ldots, x_k\}$ albo ścieżka idąca z x_i do x_{k+1}, a następnie z x_{k+1} do x_j. W pierwszym przypadku jest to $NS(x_i, x_j, k)$, w drugim suma $NS(x_i, x_{k+1}, k)$ i $NS(x_{k+1}, x_j, k)$.

Przyjmując

$$NS(x_i, x_j, 0) = a_{i,j} \quad ,$$

dla każdego $k \in \{1, 2, \ldots, n-1\}$ stosując rekurencję oblicza się

$$NS(x_i, x_j, k+1) = \min(NS(x_i, x_j, k), NS(x_i, x_{k+1}, k) + NS(x_{k+1}, x_j, k))$$

ostatecznie otrzymując

$$a_{i,j} = NS(x_i, x_j, n) \quad .$$

Krok 3

Ostatnim etapem jest rzutowanie elementów z próby X z przestrzeni $\mathbb{R}^d$ do $\mathbb{R}^t$, tak aby punkty w otrzymanej podprzestrzeni w jak największym stopniu zachowywały strukturę punktów z rozmaitości M i jednocześnie najlepiej oddawały zmienność obserwacji. Do rozwiązania tego zadania Isomap wykorzystuje skalowanie wielowymiarowe - MDS (ang. *multidimensional scaling*). MDS najpierw szuka takiej pośredniej podprzestrzeni $\mathbb{R}^s$ $(t < s < d)$ oraz punktów w tej podprzestrzeni tak, by A była macierzą odległości euklidesowych między tymi punktami (przyjmuje się, że $a_{i,j} = d_s(i,j)^2$). Następnie podprzestrzeni $\mathbb{R}^t$ oraz umieszczonych w niej elementów Y, takich że $d_t(i,j)$ minimalizują wskaźnik σ określony następującym wzorem [5]

$$\sigma = \sum_{i=1}^{n}\sum_{j=1}^{n}((d_s(i,j)^2) - (d_t(i,j)^2)) \quad . \tag{1.1}$$

Ale, czy dla każdej macierzy A można znaleźć taką podprzestrzeń, w której będzie ona macierzą odległości? Okazuje się, że jeśli A spełnia pewne założenia to istnieje rozwiązanie tego problemu. Poniżej udowodnione zostanie twierdzenie mówiące, kiedy możliwe jest znalezienie takiej podprzestrzenie $\mathbb{R}^s$, a następnie zostaną omówione kolejne kroki MDS. Jednak najpierw zostanie wprowadzone pojęcie macierzy centralnej, które w dalszej części rozdziału będzie często wykorzystywane.

Definicja 5 (Macierz centralna) *Macierz centralna $H = \{h_{i,j}\}_{i,j=1,2,\ldots,n}$ jest postaci:*

$$H = Id - Q \quad ,$$

gdzie Id jest macierzą jednostkową, a $Q = \frac{1}{n}\boldsymbol{1}\cdot\boldsymbol{1}^T$, $\boldsymbol{1}$ jest wektorem kolumnowym n jedynek. [12]

Macierze centralne mają tą własność, że pomnożone z lewej strony przez macierz usuwają z każdego jej elementu średnią wartość dla kolumny, w której znajduje się dany wyraz. Natomiast z prawej - dla wierszy.

Przykład:

$$\mathbf{\Gamma} = \begin{pmatrix} 1 & 0 & 5 \\ 2 & 3 & 4 \\ 1 & 2 & 3 \end{pmatrix}$$

w - wektor średnich dla wierszy

$$\mathbf{w} = \begin{pmatrix} 2 \\ 3 \\ 2 \end{pmatrix}$$

$$\mathbf{\Gamma}\cdot\mathbf{H} = \begin{pmatrix} -1 & -2 & 3 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

Ponadto

$$H\cdot\Gamma\cdot H = (Id - Q)\cdot\Gamma\cdot(Id - Q) = \Gamma - \Gamma^W - \Gamma^K + \Gamma^{WK} \quad , \tag{1.2}$$

gdzie
$\Gamma^W = \Gamma \cdot Q$ (każdy wyraz w macierzy Γ został zastąpiony wartością średnią wiersza, w którym się znajduje),
$\Gamma^K = Q \cdot \Gamma$ (każdy wyraz w macierzy Γ został zastąpiony wartością średnią kolumny, w której się znajduje),
$\Gamma^{WK} = Q \cdot \Gamma \cdot Q$ (każdy element w macierzy Γ został zastąpiony średnią wszystkich wyrazów macierzy).
Przykładowo

$$\mathbf{\Gamma} \cdot \mathbf{Q} = \begin{pmatrix} 2 & 2 & 2 \\ 3 & 3 & 3 \\ 2 & 2 & 2 \end{pmatrix} \quad .$$

Dla pojedynczego wyrazu macierzy $H \cdot \Gamma \cdot H$ oznacza to

$$(h \cdot \gamma \cdot h)_{i,j} = \gamma_{i,j} - \gamma^W_{i,j} - \gamma^K_{i,j} + \gamma^{WK}_{i,j} \tag{1.3}$$

(od każdego elementu $\gamma_{i,j}$ odjęta jest średnia elementów i-tego wiersza oraz j-tej kolumny macierzy Γ oraz dodana średnia wartość wszystkich elementów macierzy). [12]

Twierdzenie 6 *Niech $A = \{a_{i,j}\}_{i,j=1,2,\ldots,n}$ będzie macierzą symetryczną, taką że $a_{i,j} \geqslant 0$, $a_{i,i} = 0$ dla każdego $i, j = 1, 2, \ldots, n$ oraz niech $H = \{h_{i,j}\}_{i,j=1,2,\ldots,n}$ będzie macierzą centralną. Wówczas A jest macierzą odległości euklidesowych w $\mathbb{R}^s$ dla pewnego $s \in \mathbb{N}_+$ wtedy i tylko wtedy gdy $K = -H \cdot A \cdot H$ jest dodatnio określona. [12]*

Dowód.
$\Longrightarrow$
Skoro macierz A jest macierzą odległości euklidesowych w przestrzeni $\mathbb{R}^s$ oznacza to, że

$$a_{i,j} = d_s(i,j)^2 \quad ,$$

czyli dla pewnych $\alpha_i \in \mathbb{R}^s$, $i = 1, 2, \ldots, n$ zachodzi

$$a_{i,j} = \| \alpha_i - \alpha_j \|^2 = < \alpha_i - \alpha_j, \alpha_i - \alpha_j > \quad , \tag{1.4}$$

gdzie $\| \cdot \|$ jest normą euklidesową oraz $< \cdot, \cdot >$ standardowym iloczynem skalarnym. Dlatego

$$a_{i,j} = \| \alpha_i - \alpha_j \|^2 = \| \alpha_i \|^2 - 2 < \alpha_i, \alpha_j > + \| \alpha_j \|^2 \quad . \tag{1.5}$$

Wobec tego

$$A = C - 2B \cdot B^T + C^T \quad ,$$

gdzie

$$B = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \ldots \\ \alpha_n \end{pmatrix} \quad ,$$

$$C = \begin{pmatrix} \| \alpha_1 \|^2 & \ldots & \| \alpha_1 \|^2 \\ \vdots & \ddots & \vdots \\ \| \alpha_n \|^2 & \ldots & \| \alpha_n \|^2 \end{pmatrix} \quad .$$

Zatem

$$H \cdot A \cdot H = H \cdot C \cdot H - 2H \cdot B \cdot B^T \cdot H + H \cdot C^T \cdot H \quad .$$

Na mocy własności (1.3) pierwszy i ostatni składnik sumy są równe zeru. Ponadto dzięki symetryczności macierzy centralnych

$$H \cdot A \cdot H = -2H \cdot B \cdot B^T \cdot H = -2H \cdot B \cdot (H \cdot B)^T = -2\tilde{B} \cdot \tilde{B}^T \tag{1.6}$$

dla

$$\tilde{B} = \begin{pmatrix} \alpha_1 - \mu \\ \alpha_2 - \mu \\ \cdots \\ \alpha_n - \mu \end{pmatrix} = \begin{pmatrix} \tilde{\alpha_1} \\ \tilde{\alpha_2} \\ \cdots \\ \tilde{\alpha_n} \end{pmatrix}$$

i pewnej stałej μ. Dzięki (1.6)

$$\frac{K}{2} = -\frac{H \cdot A \cdot H}{2} = \tilde{B} \cdot \tilde{B}^T = \tilde{A} \quad . \tag{1.7}$$

Sprawdzając warunek dodatniej określoności, dla każdego wektora v zachodzi

$$v^T \tilde{A} v = v^T \tilde{B} \cdot \tilde{B}^T v = (\tilde{B}^T v)^T \cdot \tilde{B}^T v =< \tilde{B}^T v, \tilde{B}^T v >\geqslant 0 \quad ,$$

czyli $\tilde{A}$, a tym samym $2\tilde{A} = K = -H \cdot A \cdot H$ jest dodatnio określona.

$\Longleftarrow$

Skoro $-H \cdot A \cdot H = K = \{k_{i,j}\}_{i,j=1,2,\ldots,n}$ jest dodatnio określona to jest ona macierzą Grama dla pewnego układu wektorów.

Definicja 7 (Macierz Grama) *Niech $\gamma_1, \gamma_2, \ldots, \gamma_n$ będzie układem wektorów w przestrzeni euklidesowej V. Macierzą Grama układu $\gamma_1, \gamma_2, \ldots, \gamma_n$ nazywa się macierz $G = \{g_{i,j}\}_{i,j=1,2,\ldots,n}$ postaci*

$$g_{i,j} =< \gamma_i, \gamma_j > \quad ,$$

gdzie $< \cdot, \cdot >$ jest określonym na V iloczynem skalarnym. [9]

Wobec tego istnieją $\beta_i \in \mathbb{R}^m$ dla $i = 1, 2, \ldots, n$ oraz $m \in \mathbb{N}_+$, takie że $k_{i,j} =< \beta_i, \beta_j >$ ze standardowym iloczynem skalarnym. Wtedy na mocy własności 1.3 dla iloczynu macierzy z macierzami centralnymi

$$< \beta_i, \beta_j >= k_{i,j} = -(h \cdot a \cdot h)_{i,j} = -(a_{i,j} - a^W_{i,j} - a^K_{i,j} + a^{WK}_{i,j}) = -a_{i,j} + a^W_{i,j} + a^K_{i,j} - a^{WK}_{i,j} \quad , \tag{1.8}$$

gdzie
$a^W_{i,j}$-średnia elementów i-tego wiersza macierzy A,
$a^K_{i,j}$-średnia elementów j-tej kolumny macierzy A,
$a^{WK}_{i,j}$-średnia wszystkich elementów macierzy A.
Wówczas

$$\| \beta_i - \beta_j \|^2 =< \beta_i, \beta_i > -2 < \beta_i, \beta_j > + < \beta_j, \beta_j > \quad . \tag{1.9}$$

Wykorzystując teraz 1.8 oraz to, że $a_{i,i} = 0$ (z założenia), $a^K_{i,j} = a^K_{j,j}$, $a^W_{i,j} = a^W_{i,i}$ oraz $a^W_{i,j} = a^K_{j,i}$ (z symetryczności macierzy A) ciąg dalszy 1.9 wygląda następująco

$$(-a_{i,i} + a^W_{i,i} + a^K_{i,i} - a^{WK}_{i,i}) - 2(-a_{i,j} + a^W_{i,j} + a^K_{i,j} - a^{WK}_{i,j}) + (-a_{j,j} + a^W_{j,j} + a^K_{j,j} - a^{WK}_{j,j}) =$$

$$= -a_{i,i} + a^W_{i,i} + a^K_{i,i} - a^{WK}_{i,i} + 2a_{i,j} - 2a^W_{i,j} - 2a^K_{i,j} + 2a^{WK}_{i,j} - a_{j,j} + a^W_{j,j} + a^K_{j,j} - a^{WK}_{j,j} = 2a_{i,j} \quad .$$

Ostatecznie

$$\| \beta_i - \beta_j \|^2 = 2a_{i,j}$$

$$(\frac{1}{\sqrt{2}} \parallel \beta_i - \beta_j \parallel)^2 = a_{i,j}$$

$$(\parallel \frac{\beta_i}{\sqrt{2}} - \frac{\beta_j}{\sqrt{2}} \parallel)^2 = a_{i,j}$$

co oznacza, że A jest macierzą odległości euklidesowych dla układu wektorów $\alpha_i = \frac{\beta_i}{\sqrt{2}}$, $i = 1, 2, \ldots, n$.

■

Macierz A powstała w skutek algorytmu Floyd'a spełnia założenia powyższego twierdzenia, bo jest ona symetryczna oraz dla każdego $i, j = 1, 2, \ldots, n$ $a_{i,j} \geqslant 0$, $a_{i,i} = 0$. Zakładając ponadto, że $-H \cdot A \cdot H$ jest dodatnio określona wynika, że A jest macierzą odległości euklidesowych. By móc znaleźć zbiór Y w $\mathbb{R}^t$ potrzeba najpierw znaleźć układ $\alpha_i \in \mathbb{R}^s$, $i = 1, 2, \ldots, n$, dla którego A jest macierzą odległości. Pozostając przy oznaczeniach z dowodu powyższego twierdzenia, problem znalezienia α_i można sprowadzić do problemu znalezienia $\tilde{\alpha_i} = \alpha_i - \mu$. Jest on o tyle prostszy, że wyrazy macierzy $\tilde{A}$ są iloczynami skalarnym ($\tilde{A} = \tilde{B} \cdot \tilde{B}^T$). $\tilde{A}$ jest macierzą Grama dla układu $\tilde{\alpha_1}, \tilde{\alpha_2}, \ldots, \tilde{\alpha_n}$.

Dzięki temu, że $\tilde{A}$ jest dodatnio określona wszystkie jej wartości własne są nieujemne i można je uporządkować nierosnąco:

$$\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_n \geqslant 0 \quad .$$

A następnie znaleźć odpowiadające im wektory własne $w_i = (w_i^1, w_i^2, \ldots, w_i^n)$, $i = 1, 2, \ldots, n$ (wektor w_i odpowiada wartości własnej λ_i). Wówczas [3]

$$\tilde{A} = W \cdot J \cdot W^T = W \cdot J^{\frac{1}{2}} \cdot J^{\frac{1}{2}} \cdot W^T = W \cdot J^{\frac{1}{2}} \cdot (W \cdot J^{\frac{1}{2}})^T \quad , \tag{1.10}$$

gdzie W jest macierzą, w której kolumnach są kolejne wektory własne. Natomiast J jest macierzą diagonalną z odpowiadającymi wektorom wartościami własnymi na głównej przekątnej, a $J^{\frac{1}{2}}$ z pierwiastkami wartości własnych na głównej przekątnej. Przyrównując do siebie (1.7) i (1.10)

$$\tilde{A} = \tilde{B} \cdot \tilde{B}^T = W \cdot J^{\frac{1}{2}} \cdot (W \cdot J^{\frac{1}{2}})^T \quad , \tag{1.11}$$

czyli

$$\tilde{B} = W \cdot J^{\frac{1}{2}} \quad . \tag{1.12}$$

Oznacza to, że

$$\tilde{\alpha_i} = (\sqrt{\lambda_1} w_1^i, \sqrt{\lambda_2} w_2^i, \ldots, \sqrt{\lambda_n} w_n^i) \quad \text{dla} \quad i = 1, 2, \ldots, n \quad , \tag{1.13}$$

czyli $\tilde{\alpha_i}$ jest i-tym wierszem macierzy $W \cdot J^{\frac{1}{2}}$. Jednak odpowiedź $\alpha_i = \tilde{\alpha_i} + \mu$ można jeszcze uprościć, ponieważ

$$\parallel \alpha_i - \alpha_j \parallel^2 = \parallel \tilde{\alpha_i} + \mu - (\tilde{\alpha_j} + \mu) \parallel^2 = \parallel \tilde{\alpha_i} + \mu - \tilde{\alpha_j} - \mu \parallel^2 = \parallel \tilde{\alpha_i} - \tilde{\alpha_j} \parallel^2 .$$

Ostatecznie

$$\alpha_i = \tilde{\alpha_i} = (\sqrt{\lambda_1} w_1^i, \sqrt{\lambda_2} w_2^i, \ldots, \sqrt{\lambda_n} w_n^i) \tag{1.14}$$

co można rozumieć jako, że kolejne współrzędne przeskalowanego i-tego wektora własnego stanowią współrzędne wzdłuż i-tej osi w przestrzeni euklidesowej $\mathbb{R}^n$ otrzymanego układu wektorów.

Jeśli $r(\tilde{A}) = s < n$ to $n - s$ ostatnich wartości własnych jest zerowych co oznacza, że została znaleziona dokładnie s-wymiarowa podprzestrzeń, na którą zrzutowano X. Wystarczy

nie brać pod uwagę ostatnich $n - s$ współrzędnych, które są zerami. Jeśli jednak $\tilde{A}$ ma większy rząd, a celem jest znalezienie t wymiarowej podprzestrzeni, to szukany wektor będzie wyrażony przez t pierwszych wartości i wektorów własnych

$$y_i = (\sqrt{\lambda_1} w_1^i, \sqrt{\lambda_2} w_2^i, \ldots, \sqrt{\lambda_t} w_t^i) \quad . \tag{1.15}$$

W ten sposób otrzymuje się układ obiektów Y powstały z X za pomocą algorytmu Isomap.

Podczas rzutowania X konieczna jest minimalizacja straty informacji. Im wyższa wartość własna, tym odpowiadający jej wektor własny jest słabiej skorelowany z pozostałymi. Jeśli ostatnich $n - t$ wartości własnych jest znacznie mniejsza niż pierwszych t, to najlepsza będzie reprezentacja dana przez t pierwszych wektorów własnych. Dla takiej reprezentacji $\sigma = 2n(\lambda_{t+1} + \ldots + \lambda_n)$, a więc będzie najmniejsza. [12] Dla przypomnienia σ zdefiniowany w 1.1 to wskaźnik, który należy zminimalizować.

Co jednak jeśli $-HAH$, a tym samym $\tilde{A}$ nie jest dodatnio określona? Wówczas po zastosowaniu powyższej formuły współrzędne α_i będą zespolone. Jednak problem ten można rozwiązać dodając dodatkowy krok. Wystarczy zastąpić wszystkie ujemne wartości własne zerami. [3] Wówczas wzór 1.14 pozostaje bez zmian.

Ze względu na skomplikowaną formułę kroku 3 poniżej zostało umieszczone jego podsumowanie.

1. Dla macierzy A oblicz $\tilde{A} = -\frac{HAH}{2}$.
2. Znajdź wartości własne macierzy $\tilde{A}$.
3. Zamień ujemne wartości własne na zera, a następnie uporządkuj je w sposób nierosnący. Znajdź odpowiadające wartościom własnym wektory własne.
4. Oblicz $\tilde{B} = W \cdot J^{\frac{1}{2}}$ (W macierz, w której kolumnach są wektory własne, $J^{\frac{1}{2}}$ macierz diagonalna, w której na głównej przekątnej są pierwiastki kolejnych wartości własnych).
5. Szukane rozwiązanie w przestrzeni t-wymiarowej stanowią kolejne wiersze macierzy $\tilde{B}$, z pominięciem $n - t$ ostatnich kolumn.

1.2. Algorytm PAM

Podział zbioru danych na rozłączne skupienia jest to proces w którym dąży się do pogrupowania elementów w większe grupy w obrębie których są one jak najbardziej do siebie podobne, a w różnych jak najbardziej odmienne od siebie. Mając już jednorodne klasy łatwiej jest wyodrębnić cechy wspólne obiektów w danej grupie. PAM (algorytm medoidów) jest jedną z wielu metod podziału zbioru danych na skupienia. Jej celem jest znalezienie takiego rozbicia, w którym zostanie zminimalizowana suma odmienności elementów od najbliższych im elementów wyróżnionych.

1.2.1. Niepodobieństwo obiektów

Zanim $X = \{x_1, x_2, \ldots, x_n\}$ zostanie podzielone na $k, k \in \mathbb{N}_+$ rozłącznych skupień należy najpierw określić stopień podobieństwa/niepodobieństwa między wszystkimi parami elementów z X. Na podstawie danych o obiektach np. położenia w przestrzeni, miara podobieństwa oddaje jak „blisko" siebie znajdują się obserwacje. Swoistym przeciwieństwem podobieństwa jest odmienność. Może być ona wyrażona odległością w dowolnej metryce, ale równie dobrze

może to być miara zdefiniowana przez badacza np. w naukach społecznych zdecydowanie łatwiej jest samemu określić miarę odmienności między badanymi obiektami niż szukać metryki, która będzie spełniać odpowiednie założenia. W pierwszym przypadku miara odmienności będzie funkcją

$$o : X \times X \longrightarrow \mathbb{R}^+$$

spełniającą następujące warunki:

1. $o(x_i, x_j) \geqslant 0, \quad o(x_i, x_j) = 0 \Leftrightarrow x_i = x_j$
2. $o(x_i, x_j) = o(x_j, x_i)$
3. $o(x_i, x_j) \leqslant o(x_i, x_k) + o(x_k, x_j)$

dla każdej pary $x_i, x_j \in X$ oraz $i, j = 1, 2, \ldots, n$.

Dla zbioru $x_1, x_2, \ldots, x_n$ będącego zbiorem obserwacji opisanych wektorami z przestrzeni $\mathbb{R}^d$, czyli $x_i = (x_i^1, x_i^2, \ldots, x_i^d)$, $x_i^k \in \mathbb{R}$ dla $k = 1, 2, \ldots, d$, $i = 1, 2, \ldots, n$ wśród najczęściej używanych odległości między elementami x_i a x_j wyróżnia się:

1. odległość Minkowskiego (norma w L_p)

$$o(x_i, x_j) = (\sum_{k=1}^{d} (\mid x_i^k - x_j^k \mid)^p)^{\frac{1}{p}} \quad ,$$

2. odległość Manhattan (norma w L_1)

$$o(x_i, x_j) = \sum_{k=1}^{d} \mid x_i^k - x_j^k \mid \quad ,$$

3. odległość Euklidesowa (norma w L_2)

$$o(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_i^k - x_j^k)^2} \quad ,$$

4. odległość maksimum wymiarów (norma w L_∞)

$$o(x_i, x_j) = \max_{k=1,2,\ldots,d} \mid x_i^k - x_j^k \mid \quad ,$$

5. odległość Canberra

$$o(x_i, x_j) = \sum_{k=1}^{d} \frac{\mid x_i^k - x_j^k \mid}{\mid x_i^k + x_j^k \mid} \quad .$$

Jeśli natomiast zostanie przyjęte, że odmienność nie będzie odległością zadaną przez żadną metrykę, to można ją wówczas utożsamić z funkcją

$$o : X \times X \longrightarrow \mathbb{R}^+$$

spełniającą:

1. $o(x_i, x_j) = 0$ dla $x_i = x_j$

2. $o(x_i, x_j) = o(x_j, x_i)$

dla każdej pary $x_i, x_j \in X$ oraz $i, j = 1, 2, \ldots, n$.

Niezależnie od przyjętej konwencji na odmienność (jest czy nie jest odległością między elementami) zakładając dodatkowo, że $o(x_i, x_j) \in [0, 1]$ można określić podobieństwo między x_i a x_j. Oznaczone przez $p(x_i, x_j)$ wyraża się wzorem

$$p(x_i, x_j) = 1 - o(x_i, x_j) \quad .$$

Po zdefiniowaniu miary niepodobieństwa, można skonstruować macierz niepodobieństwa. Dla $X = \{x_1, x_2, \ldots, x_n\}$ będzie to macierz o wymiarach $n \times n$, w której wyraz z i-tego wiersza oraz j-tej kolumny jest miarą odmienności elementów x_i oraz x_j, czyli $o(x_i, x_j)$. Macierz ta będzie kwadratowa, symetryczna z zerami na głównej przekątnej. Dlatego dla przejrzystości może być przedstawiona jako macierz dolnotrójkątna.

W dalszej części pracy zostało przyjęte, że odmienność między elementami jest wyrażona ustaloną miarą odległości.

1.2.2. Opis algorytmu

PAM jest metodą podziału na skupienia zbioru obserwacji, która korzysta z macierzy odmienności i elementów centralnych, czyli medoidów.

Definicja 8 (Medoid) *Medoidem nazywa się wyróżniony element x_w należący do zbioru danych $X = \{x_1, x_2, \ldots, x_n\}$, którego średnia odmienność względem innych elementów jest najmniejsza. Jest elementem wyznaczanym jednoznacznie, który jest reprezentantem swojego skupienia. [8]*

Mając $k \in \mathbb{N}_+$ oraz zbiór danych $X = \{x_1, x_2, \ldots, x_n\}$ algorytm PAM zwraca listę elementów z X podzielonych na k rozłącznych grup w taki sposób, że suma niepodobieństw elementów do ich najbliższych medoidów jest zminimalizowana. Medoid jest zawsze elementem z próby i podobnie jak centroid jest najbardziej centralnie umieszczonym obiektem, w tym przypadku, w skupieniu. Każda grupa ma dokładnie jeden medoid, a kiedy zostaną one już wyróżnione, to ostateczny podział odbywa się poprzez przypisanie pozostałych elementów do tych grup, które reprezentują najbardziej podobne do nich medoidy. Oznacza to, że celem PAM jest znalezienie takich k medoidów $\{m_1, m_2, \ldots, m_k\} \in \{x_1, x_2, \ldots, x_n\}$, które zminimalizują

$$\sum_{i=1}^{n} \min_{j=1,2,\ldots,k} o(x_i, m_j) \quad , \tag{1.16}$$

gdzie $o(x_i, m_j)$ jest miarą niepodobieństwa elementów x_i oraz m_j (wykorzystane jest to, że medoid jest elementem z próby). [8]

Niech wszystkie elementy próby to $X = \{x_1, x_2, \ldots, x_n\}$ oraz wybrane elementy na medoidy to $M = \{m_1, m_2, \ldots, m_k\}$. Oczywiście elementy nie wybrane to $X - M$. Na początku PAM dzieli intuicyjnie elementy na założoną z góry liczbę klas, a potem ten podział jest poprawiany by uzyskać jak najniższy wskaźnik 1.16. Metodę działania algorytmu PAM można przedstawić w dwóch krokach. [8]

1. Faza Budowy (ang. *Build Phase*).
   Inicjalizacja poprzez wybór k elementów na medoidy, tak że:

- m_1 obiekt z X z najmniejszą sumą

$$\sum_{i=1}^{n} o(x_i, m_1) \quad ,$$

- m_2 obiekt z X, który najbardziej obniża 1.16,
  $\vdots$
- m_k obiekt z X, który najbardziej obniża 1.16.

Po wyborze medoidów, wszystkie pozostałe obiekty są przypisane do skupień z najbliższymi im medoidami.

2. Faza Zamiany (ang. *Swap Phase*).
   Rozważane są wszystkie pary (m_i, x_k), gdzie $m_i \in M$, $x_k \in X - M$. Jeśli któryś x_k obniża 1.16 bardziej niż m_i to następuje zamiana. Proces jest powtarzany do momentu, w którym 1.16 nie może być już bardziej obniżona, czyli nie nastąpi kolejna zamiana.

Poniżej został przedstawiony algorytm, według którego jest przeprowadzana każda z faz. [20] Jednak najpierw dla elementu x_j zostaną zdefiniowane
$\mu_j :=$ najmniejsza odległość elementu x_j do elementu z M,

$$\mu_j = \min_{1,2,\ldots,k} o(m_k, x_j) \quad ,$$

$\tilde{\mu}_j :=$ druga najmniejsza odległość elementu x_j do elementu z M.

Faza Build

1. Ustal $M = \{m_1\}$, gdzie dla m_1 zachodzi

$$\min_{j=1,2,\ldots,n} \sum_{i=1}^{n} o(x_j, x_i) = \sum_{i=1}^{n} o(m_1, x_i) \quad .$$

2. Poniższe kroki (3 i 4) powtórz $k-1$ razy, by znaleźć odpowiednio medoidy $m_2, m_3, \ldots, m_k$.

3. Po kolei rozważ obiekty $x_i \in X - M$ jako kandydatów na kolejne medoidy.

   (a) Dla pozostałych $x_j \in X - M$ oblicz μ_j.

   (b) Jeśli $\mu_j > o(x_i, x_j)$, czyli niepodobieństwo x_j do x_i jest mniejsze niż x_j do jego najbliższego medoida należy rozważyć x_i jako element mogący obniżyć 1.16. Wówczas zysk z możliwej zamiany to

$$z_{i,j} = \max(\mu_j - o(x_i, x_j), 0) \quad .$$

   (c) Oblicz sumę wszystkich zysków Z_i dla x_i, czyli

$$Z_i = \sum_{j \in X - M} z_{i,j} \quad .$$

4. Wybierz ten obiekt x_0, który maksymalizuje wskaźnik Z i dodaj go do M

$$M := M \cup \{x_0\} \quad .$$

5. Przypisz pozostałe obiekty do najbliższych im medoidów.

Faza Swap

Ten etap modyfikuje wyniki otrzymane w kroku pierwszym, a tym samym polepsza poprawność grupowania. Rozważane są wszystkie pary $(m_i, x_k) \in M \times X - M$ i korzyści płynące z ich zamiany.

1. Niech w_{jik} oznacza wkład obiektu x_j do zamiany m_i z x_k, czyli w przypadku jeśli x_k przejąłby funkcję medoidu za element m_i. Jeśli wkład jest ujemny to zamiana jest korzystna, jeśli dodatni to taka zamiana nie polepszy wyników grupowania, a tym samym wskaźnika 1.16. Rozważane są następujące przypadki.
   (a) $\mu_j < o(x_j, m_i)$ element x_j ma dalej do medoidu m_i niż do swojego najbliższego elementu z M .
      i. Jeśli $o(x_j, x_k) \geqslant \mu_j$ to nie rób nic, bo x_j ma również daleko do x_k, czyli $w_{jik} = 0$.
      ii. Jeśli $o(x_j, x_k) < \mu_j$, czyli element x_j ma bliżej do x_k niż do swojego medoidu, to policz możliwą korzyść płynącą z zamiany, czyli $w_{jik} = o(x_j, x_k) - \mu_j$.
   (b) $\mu_j = o(x_j, m_i)$, czyli medoid m_i jest medoidem elementu x_j.
      i. Jeśli $o(x_j, x_k) < \tilde{\mu}_j$, czyli x_j ma bliżej do x_k niż do drugiego najbliższego medoidu, to $w_{jik} = o(x_j, x_k) - \mu_j$ co może być zarówno dodatnie jak i ujemne.
      ii. Jeśli $o(x_j, x_k) \geqslant \tilde{\mu}_j$, czyli x_j ma dalej do x_k niż do drugiego najbliższego medoidu, to $w_{jik} = \tilde{\mu}_j - \mu_j$ co zawsze jest dodatnie z definicji μ_j.
2. Oblicz całkowity efekt zamiany m_i i x_k
$$W_{ik} = \sum_{x_j \in X - M} w_{jik} \quad .$$
3. Wybierz parę (m_i, x_k), która zminimalizuje sumę W_{ik}.
4. Jeśli dla tej pary $W_{ik} < 0$ to dokonaj zamiany. Element m_i przechodzi na miejsce x_k i na odwrót. Jeśli jest przeciwnie, czyli $W_{ik} \geqslant 0$ to grupowania nie można ulepszyć poprzez taką zamianę.

Dzięki temu, że
$$\mu_j = \min_{1,2,\ldots,k} o(m_k, x_j)$$
nie może zachodzić przypadek
$$\mu_j > o(m_i, x_j)$$
dla pewnego $m_i \in M$, który został pominięty w powyższym opisie fazy zamiany. Należy jeszcze zauważyć, że jeśli w fazie Swap nastąpi zamiana elementów to dla każdego obiektu $x_j \in X$ musi być na nowo policzone μ_j oraz $\tilde{\mu}_j$.

1.3. Algorytm Isopam

Algorytm wykorzystywany przez Isopam jest metodą podziału zbioru danych na skupienia powstałą poprzez połączenie Isomap oraz PAM. Może być zastosowany w wersji podziału hierarchicznego, jak i niehierarchicznego. W tej pracy zostanie opisany tylko ten drugi przypadek, ponieważ w rozdziale dotyczącym analizy danych rzeczywistych właśnie ta wersja będzie wykorzystywana.

Isopam został wymyślony do grupowania zbiorowisk roślinnych oraz zwierzęcych na podstawie występowania gatunków na ich terenie, ze względu na częste rozbieżności między wiedzą ekspercką, a wynikami wykorzystywanych do tej pory metod analizy skupień. Ten algorytm optymalizuje podział na skupienia i ewentualnie liczbę skupień, biorąc pod uwagę wszystkie lub tylko wskazane typowe gatunki tzw. gatunki wyróżniające dla danego skupienia. [19]

Definicja 9 (Gatunek wyróżniający) *Gatunek wyróżniający to taki, który głównie występuje w danym zbiorowisku lub grupie zbiorowisk. [6]*

Definicja 10 (Zbiorowisko roślinne/zwierzęce) *Zbiorowiskiem nazywa się zgrupowanie osobników różnych gatunków roślin/zwierząt zajmujących wspólnie określony teren np. zbiorowisko lasu liściastego, zbiorowisko łąki wilgotnej. [6]*

Isopam ma lepiej radzić sobie z takimi trudnościami jak np. to, że samo występowanie danego gatunku w zbiorowisku nie musi być zawsze istotną informacją. W niektórych przypadkach może być to „szum", czyli zakłócenie. Dany gatunek przypadkowo pojawił się w zbiorowisku i nie ma wpływu na jego skład, a jedynie zakłóca odkrycie prawdziwej struktury łączącej zbiorowiska. Badacz jest w stanie dostrzec taką sytuację, ale algorytm nie. [7]

Działanie Isopam dzieli się na kilka rozłącznych etapów. Pierwszym z nich polega na usystematyzowaniu danych (przyjęte jest poniżej, że elementami z próby są zbiorowiska, a cechami gatunki). Służy do tego Isomap. Odległości między elementami są policzone na zasadzie k najbliższych sąsiadów. Dzięki tej metodzie uwidocznione są znaczące odległości między zbiorowiskami, które nie mają wspólnych gatunków, ale połączone są siecią innych, pośrednich zbiorowisk. Oznacza to, że zastosowanie przybliżonych rzeczywistych odległości do pomiaru odmienności między zbiorowiskami oddaje dystans je dzielący (rozumiany w sensie ekologicznym). Praca Isomap jest zakończona w momencie zrzutowania danych do przestrzeni niższego wymiaru, przestrzeni t wymiarowej. Kolejnym krokiem jest podział zbiorowisk na rozłączne skupienia. PAM wykorzystuje niskowymiarowego reprezentanta danych. Przy pomocy macierzy odmienności (są to odległości euklidesowe w niskowymiarowej podprzestrzeni) następuje wybór m medoidów oraz dalszy podział. Zakończeniem tych dwóch kroków jest ocena wyników. W trakcie działania algorytmu brane są pod uwagę trzy parametry:

1. k liczba sąsiadów w algorytmie k-Isomap,
2. t wymiar podprzestrzeni, na którą następuje zrzutowanie,
3. m liczba medoidów, czyli liczba skupień, które mają zostać wyodrębnione.

Uwzględniane są wszystkie możliwe ich kombinacje, by znaleźć najkorzystniejszy podział, chyba że któryś z parametrów został wcześniej ustalony. Najniższe możliwe wartości są zawsze równe 2. Po przejściu przez wszystkie etapy następuje zmiana parametrów (k, t lub m) i ponowne działanie Isomap, PAM oraz ocena. Ostateczny podział jest wyłaniany poprzez wybór tego, który ma najwyższą ocenę.

Ocena dokonanego podziału na skupienia może odbywać się na dwa sposoby, w zależności od tego czy zostały wyodrębnione gatunki wyróżniające czy nie. W pierwszym przypadku systematyzowanie (Isomap) i podział (PAM) są powtarzane w poszukiwaniu podziału zapewniającego wysoką wierność gatunków względem zbiorowisk w skupieniach. W drugim przypadku poszukiwany jest taki podział, w którym zostaje wyodrębnionych jak najwięcej gatunków typowych (również o wysokiej wierności). [7]

Definicja 11 (Wierność gatunkowa) *Wierność gatunkowa jest to stopień powiązania gatunków z określonym zespołem roślinności. Gatunki o dużym stopniu wierności występują wyłącznie lub prawie wyłącznie w danym zespole roślinnym. [6]*

Do oceny podziału wykorzystywana jest wartość statystyki testowej G.

1.3.1. Test G

Test G nazywany również testem ilorazu wiarogodności lub G^2 służy do weryfikowania hipotez. Może być stosowany w sytuacji, gdy badana jest zależność dwóch mierzalnych zmiennych - test niezależności (ang. *test of independence*). [14]

Dla tabeli T wymiaru $w \times k$, gdzie w jest liczbą wierszy, k liczbą kolumn, w której jest przedstawiona zależność między dwiema zmiennymi z minimum dwoma cechami każda, wzór na statystykę testową G ma następującą postać

$$G = 2\sum_{i=1}^{w}\sum_{j=1}^{k} o_{ij} \ln \frac{o_{ij}}{\hat{o}_{ij}} \quad , \tag{1.17}$$

o_{ij} - liczebności empiryczne, czyli obserwowane (miejsce w i-tym wierszu i j-tej kolumnie w tabeli),
$\hat{o}_{ij}$ - liczebności teoretyczne,

$$\hat{o}_{ij} = \frac{o_{i\cdot} o_{\cdot j}}{N} \quad , \tag{1.18}$$

$$o_{i\cdot} = \sum_{j=1}^{k} o_{ij} \quad , \quad o_{\cdot j} = \sum_{i=1}^{w} o_{ij} \quad .$$

Statystykę G można również zapisać jako

$$G = 2N\sum_{i=1}^{w}\sum_{j=1}^{k} p_{ij}(\ln(p_{ij}) - \ln(p_{i\cdot}) - \ln(p_{\cdot j})) \quad , \tag{1.19}$$

gdzie

$$N = \sum_{i=1}^{w}\sum_{j=1}^{k} o_{ij} \quad ,$$

$$p_{ij} = \frac{o_{ij}}{N} \quad , \quad p_{i\cdot} = \frac{o_{i\cdot}}{N} \quad , \quad p_{\cdot j} = \frac{o_{\cdot j}}{N} \quad .$$

Dla ustalonego poziomu istotności wartość statystyki G jest porównywana do dystrybuanty rozkładu chi-kwadrat z $(w-1)(k-1)$ stopniami swobody.

Kryterium optymalizacji w Isopam stanowi wartość statystyki testowej G. Dla ustalonego podziału na m skupień i wszystkich gatunków osobno tworzona jest tabela T wymiaru $2 \times m$, której kolumny odnoszą się do kolejnych skupień, natomiast wiersze do obecności, bądź też braku w nich danego gatunku. Wyrazy tabeli określają liczbę zbiorowisk w danym skupieniu, w których gatunek wystąpił (pierwszy wiersz) oraz tych, w których nie wystąpił (drugi wiersz). Wartość statystyki testowej G jest obliczona z następującego wzoru

$$G = 2\left(\sum_{i=1}^{w}\sum_{j=1}^{k} o_{ij}\ln(o_{ij}) - \sum_{j=1}^{k} o_{\cdot j}\ln(o_{\cdot j}) - \sum_{i=1}^{w} o_{i\cdot}\ln(o_{i\cdot}) + N\ln(N)\right) \tag{1.20}$$

opisanego w załączniku numer 3 do [7]. Wyprowadzenie powyższego wzoru z 1.19 zostało umieszczone w Dodatku A. Kolejnym krokiem jest przeprowadzenie korekty Williama [7]

$$G_W = \frac{G}{q} \quad , \tag{1.21}$$

gdzie

$$q = 1 + \frac{1}{6N(k-1)} \left(N \sum_{j=1}^{k} \frac{1}{o_{\cdot j}} - 1 \right) \left(N \sum_{i=1}^{w} \frac{1}{o_{i\cdot}} - 1 \right)$$

oraz standaryzacji Botta-Dukát [7]

$$G_s = \frac{G_W - k - 1}{\sqrt{2(k-1)}} \quad . \tag{1.22}$$

Standaryzacja jest konieczna, by wyeliminować wpływ liczby skupień otrzymywanych w różnych podziałach.

Jeśli na początku zostały wskazane typowane gatunki wyróżniające to tylko dla nich jest obliczona wartość G_s. Jeśli nie, to dla wszystkich, a wartością progową jest 3.5. Gatunki, które ją przekroczą traktowane są jako gatunki wyróżniające. Mając daną listę typowych gatunków Isopam dokonuje oceny i szuka optymalnego podziału. Dla różnych grupowań obliczony jest iloczyn liczby gatunków wyróżniających oraz ich średniej wartości G_s. Podział, który maksymalizuje ten iloczyn jest najkorzystniejszy, ponieważ wartość G_s może być interpretowana jako miara wierności gatunkowej.

1.4. Sylwetka podziału

Po zakończonej analizie skupień warto jest zbadać poprawność otrzymanego podziału, czyli określić elementy (o ile takie istnieją) leżące na pograniczu dwóch lub więcej skupień, czy wręcz błędnie zakwalifikowane. W tym celu używana jest sylwetka (ang. *silhouette*). Jest to graficzna metoda reprezentacji przeprowadzonego grupowania. Dla każdego skupienia powstaje jego sylwetka przedstawiająca jakość dokonanego podziału w jego obrębie, ale porównanie wszystkich sylwetek daje ogólny widok jakości podziału całego zbioru danych.

1.4.1. Metoda obliczania

Sylwetka dla zbioru danych $X = \{x_1, x_2, \ldots, x_n\}$ może być policzona albo za pomocą macierzy odmienności, albo za pomocą macierzy podobieństwa. Na początku zostanie rozważony pierwszy przypadek. Niech $O = \{o_{i,j}\}_{i,j=1,2,\ldots,n}$ będzie macierzą, w której wyraz z i-tego wiersza oraz j-tej kolumny jest miarą niepodobieństwa elementów x_i oraz x_j

$$o_{i,j} = o(x_i, x_j) \quad .$$

Ponadto niech będzie znany ustalony podział X na rozłączne skupienia (minimum 2) wykonany dowolną metodą.

Sylwetka jest obliczana dla każdego elementu $x_i \in X$ $i = 1, 2, \ldots, n$ osobno. Wartością liczbową, która jest zwracana jest wskaźnik $s(x_i)$. Dodatkowo jest podana informacja o skupieniu, do którego należy x_i oraz skupieniu dla niego sąsiednim, czyli takim do którego x_i najbardziej pasuje (nie uwzględniając tego do którego został przydzielony).

Niech σ oznacza skupienie, do której został przypisany obiekt x_i. Jeśli w σ są jeszcze inne elementy to zostaje zdefiniowana

$a(x_i)$:=średnia odmienność elementu x_i względem pozostałych elementów z σ,

czyli

$$a(x_i) = \frac{1}{\#\sigma - 1} \sum_{x_j \in \sigma \setminus \{x_i\}} o(x_i, x_j) \quad . \tag{1.23}$$

Następnie dla każdego skupienia $\tilde{\sigma} \neq \sigma$ zostaje obliczona

$c(x_i, \tilde{\sigma})$:=średnia odmienność elementu x_i względem elementów z $\tilde{\sigma}$,

czyli

$$c(x_i, \tilde{\sigma}) = \frac{1}{\#\tilde{\sigma}} \sum_{x_j \in \tilde{\sigma}} o(x_i, x_j) \tag{1.24}$$

i wybrana najmniejsza z wartości

$$b(x_i) = \min_{\tilde{\sigma} \neq \sigma} c(x_i, \tilde{\sigma}) \quad . \tag{1.25}$$

Skupienie σ^* dla którego zachodzi $b(x_i) = c(x_i, \sigma^*)$ jest skupieniem sąsiednim dla x_i. Wynika z tego, że skupienie sąsiednie to takie skupienie, do którego zostałby przypisany element x_i jeśli nie mógłby być przypisany do swojego skupienia. Dzięki dodatniej wartości miary odmienności zarówno $a(\cdot)$ jak i $b(\cdot)$ są zawsze nieujemne. Ostatecznie wskaźnik sylwetki wyraża się wzorem

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max(a(x_i), b(x_i))} \quad . \tag{1.26}$$

Ponadto jeśli $\sigma = \{x_i\}$ to przyjmuje się $s(x_i) = 0$.

Sylwetka może być również policzona za pomocą podobieństwa obiektów. Przyjmując oznaczenia jak wyżej, dla każdego elementu x_i z próby X $a(x_i)$ oraz $c(x_i, \tilde{\sigma})$ jest zdefiniowane tak samo jak w (1.23) i (1.24). Natomiast

$$b = \max_{\tilde{\sigma} \neq \sigma} c(x_i, \tilde{\sigma}) \tag{1.27}$$

oraz

$$s(x_i) = \frac{a(x_i) - b(x_i)}{\max(a(x_i), b(x_i))} \quad . \tag{1.28}$$

Poniżej będzie wykorzystany wskaźnik sylwetki policzony za pomocą macierzy odmienności.

Lemat 12 *Dla każdego elementu z próby jego wskaźnik sylwetki należy do przedziału* $[-1, 1]$.

Dowód. Należy udowodnić, że

$$\forall x_i \in X \quad s(x_i) \in [-1, 1] \quad .$$

Niech zostanie ustalony element $x_i \in X$ przydzielony do skupienia σ.

1. Jeśli $\#\sigma = 1$ to
$$s(x_i) = 0 \in [-1, 1] \quad .$$

2. Jeśli $\#\sigma > 1$, czyli
$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max(a(x_i), b(x_i))}$$
to możliwe są następujące przypadki.

(a) $b(x_i) > a(x_i)$
Oznacza to, że $\max(a(x_i), b(x_i)) = b(x_i)$, czyli $s(x_i) = 1 - \frac{a(x_i)}{b(x_i)}$.
Załóżmy przeciwnie czyli, że $s(x_i) \in (-\infty, -1) \cup (1, +\infty)$.
Jeśli

$$1 - \frac{a(x_i)}{b(x_i)} > 1$$

to dzięki temu, że $b(x_i)$ jest nieujemne

$$0 > \frac{a(x_i)}{b(x_i)}$$
$$0 > a(x_i) \quad .$$

Co daje sprzeczność, bo $a(x_i)$ jest również nieujemne.
Podobnie jeśli

$$1 - \frac{a(x_i)}{b(x_i)} < -1$$
$$2 < \frac{a(x_i)}{b(x_i)}$$
$$2b(x_i) < a(x_i) \quad .$$

Sprzeczność z założeniem, że $b(x_i) > a(x_i)$.

(b) $b(x_i) = a(x_i)$, więc

$$s(x_i) = 0 \in [-1, 1] \quad .$$

(c) $b(x_i) < a(x_i)$
Oznacza to, że $\max(a(x_i), b(x_i)) = a(x_i)$, czyli $s(x_i) = \frac{b(x_i)}{a(x_i)} - 1$.
Załóżmy przeciwnie czyli, że $s(x_i) \in (-\infty, -1) \cup (1, +\infty)$.
Jeśli

$$\frac{b(x_i)}{a(x_i)} - 1 > 1$$

z nieujemności $a(x_i)$

$$\frac{b(x_i)}{a(x_i)} > 2$$
$$b(x_i) > 2a(x_i) \quad .$$

Sprzeczność z założeniem $b(x_i) < a(x_i)$.
Podobnie jeśli

$$\frac{b(x_i)}{a(x_i)} - 1 < -1$$
$$\frac{b(x_i)}{a(x_i)} < 0$$
$$b(x_i) < 0 \quad .$$

Sprzeczność z nieujemnością $b(x_i)$.

■

1.4.2. Interpretacja

Przy interpretacji wartości wskaźnika $s(\cdot)$ rozważone zostaną trzy charakterystyczne przypadki, w których wskaźnik sylwetki dla ustalonego elementu x_i jest blisko $1, -1$ oraz 0 odpowiednio.

- $s(x_i) \approx 1$
  Po pierwsze oznacza to, że
  $$s(x_i) > 0 \quad ,$$
  czyli
  $$b(x_i) - a(x_i) > 0$$
  $$b(x_i) > a(x_i) \quad .$$
  Co dalej
  $$\max(a(x_i), b(x_i)) = b(x_i)$$
  oraz
  $$s(x_i) = 1 - \frac{a(x_i)}{b(x_i)} \quad .$$
  Jeśli
  $$s(x_i) \approx 1$$
  to
  $$\frac{a(x_i)}{b(x_i)} \approx 0 \quad ,$$
  czyli
  $$a(x_i) \lll b(x_i) \quad .$$
  Oznacza to, że w obrębie σ, skupienia elementu x_i, ma on znacznie mniejszą średnią odmienność w porównaniu do σ^*, który jest jemu sąsiedni. Nawet drugi najlepszy wybór obiektu x_i, czyli σ^* nie jest porównywalnie blisko co σ. Wobec tego element x_i został dobrze przydzielony.

- $s(x_i) \approx -1$
  Oznacza to, że
  $$s(x_i) < 0 \quad ,$$
  czyli
  $$b(x_i) - a(x_i) < 0$$
  $$b(x_i) < a(x_i) \quad .$$
  Co dalej
  $$\max(a(x_i), b(x_i)) = a(x_i)$$
  oraz
  $$s(x_i) = \frac{b(x_i)}{a(x_i)} - 1 \quad .$$
  Jeśli
  $$s(x_i) \approx -1$$
  to
  $$\frac{b(x_i)}{a(x_i)} \approx 0 \quad ,$$

czyli

$$b(x_i) \lll a(x_i) \quad .$$

Średnia odmienność x_i względem σ^* jest zdecydowanie mniejsza niż względem σ, więc lepiej byłoby przypisać x_i do skupienia σ^*. Oznacza to, że element x_i mógł zostać źle przydzielony.

- $s(x_i) \approx 0$
  Zatem

$$b(x_i) - a(x_i) \approx 0 \quad ,$$

  czyli

$$b(x_i) \approx a(x_i) \quad .$$

  Oznacza to, że średnia odmienność elementu x_i względem σ oraz σ^* jest porównywalnej wielkości, czyli x_i pasuje równie dobrze do σ^* jak i do σ. Ten element leży na granicy dwóch skupień.

Wskaźnik sylwetki może być przedstawiony na wykresie. Oś pozioma jest przedziałem [-1,1] na którym zaznaczone są wartości $s(\cdot)$. Natomiast na pionowej umieszczone są po kolei wszystkie skupienia z wypunktowanymi elementami do nich należącymi. Elementy w danej grupie są uporządkowane względem malejącego wskaźnika $s(\cdot)$, co daje poglądowy obraz podziału dla całego skupienia, ale również dla całego podziału, ponieważ na jednym wykresie są umieszczone wszystkie skupienia jedno pod drugim.

Sylwetka jest pomocny przy ocenie dokonanego podziału. Może być użyty do poprawy wyników grupowania. Istnieją następujące możliwości.

1. Elementy o ujemnym wskaźniku sylwetki mogą zostać przeniesione do swoich skupień sąsiednich. Jednak jeśli miałoby to dotyczyć wielu obiektów metoda ta nie jest odpowiednia, ponieważ w takim przypadku cała struktura skupień zostanie zmieniona.

2. Mogą zostać porównane wyniki analizy skupień dla tej samej próby X, ale wykonanej innymi algorytmami, a następnie zostanie wybrany najkorzystniejszy podział.

3. Jest pomocny przy wyborze k - liczby skupień w podziale. Jeśli k zostanie ustalone za nisko to wewnętrzne niepodobieństwa obiektów w skupieniu będą wysokie, a to oznacza duży $a(\cdot)$, a tym samym niski $s(\cdot)$. Przeciwnie, jeśli k ustalono za wysoko to część naturalnych skupień musiała zostać podzielona by wyodrębnić dokładnie k grup. W rezultacie część elementów ma równie blisko do swojego skupienia jak i do skupienia sąsiedniego, czyli $s(\cdot)$ jest niski.

Dla ustalonego skupienia definiuje się średnią szerokość sylwetki (ang. *average silhouette width*), która określa średnią wartość silhouette dla wszystkich elementów z tego skupienia. Podobnie dla ustalonego podziału określa się łączną średnią szerokość sylwetki (ang. *overall average silhouette width*), która jest średnią wartością sylwetek wszystkich elementów ze zbioru danych. Te wskaźniki pozwalają odróżnić dobrze wyodrębnione skupienia (z dużą średnią szerokością sylwetki) od tych słabszych (z niską średnią szerokością sylwetki). Natomiast poszukiwanie najodpowiedniejszej liczby skupień, można sprowadzić do znalezienia takiego podziału, którego łączna średnia szerokość sylwetki będzie największa.

Rozdział 2

Funkcje pakietu R

2.1. Pakiet vegan

Pakiet `vegan` nazwany pakietem ekologów (ang. *Community Ecology Package*) ma zapisane funkcje służące m.in. do analizy zróżnicowania czy uporządkowywania obiektów. [1] Omówiona w rozdziale 1.1 metoda zmiany początkowej przestrzeni danych została zaimplementowana w funkcji `isomap()` zapisanej w tym pakiecie. Co ważne istnieje możliwość zastosowania dowolnej wersji Isomap (ε-, k-).

2.1.1. Isomap

Funkcja `isomap()` służy do umieszczenia zbioru obiektów z przestrzeni wysokowymiarowej w podprzestrzeni niższego wymiaru, przy jednoczesnym zachowaniu ich struktury odległości. Jej najistotniejszym etapem jest moment liczenia odmienności między obiektami, na zasadzie sąsiadów i sieci połączeń między nimi. To ją odróżnia od skalowania wielowymiarowego. Z tego względu funkcja `isomap()` wywołuje najpierw funkcję `isomapdist()` liczącą te nowe odmienności między obiektami, a następnie stosuje funkcję `cmdscale()` odpowiedzialną za skalowanie wielowymiarowe. Składnia funkcji `isomap()` z domyślnie zapisanymi w programie `R` wartościami, która w dużej mierze pokrywa się ze składnią funkcji `isomapdist()`, została przedstawiona poniżej

```
isomap(dist, ndim=10, epsilon, k, path = "shortest", fragmentedOK =FALSE)
```

natomiast tabela 2.1 zawiera opis ważniejszych argumentów tej funkcji.

Podstawą do zainicjowania algorytmu Isomap jest macierz odmienności między obiektami. Niezależnie od tego, która wersja (ε- czy k-) zostanie użyta, pierwszym krokiem jest określenie dla każdego obiektu jego sąsiadów właśnie na podstawie macierzy odmienności. Funkcja `isomap()` jako wejściową macierz danych akceptuje tylko macierz niepodobieństw. W przypadku jej braku w tym samym pakiecie dostępna jest funkcja `dist()`, która umożliwia jej policzenie. [1]

Jeśli jest stosowana wersja ε-Isomap to zachowywane są odmienności między obiektami, które są oddalone od siebie o mniej niż ε. Oznacza to, że w macierzy odmienności zostają zatrzymane tylko wyrazy mniejsze co do wartości od ε. W przypadku k-Isomap wybieranych jest k najmniejszych wartości dla każdego obiektu osobno (czyli k najmniejszych wyrazów dla każdego wiersza). Ważne jest, że minimum jeden z tych argumentów musi zostać podany w deklaracji składni funkcji. Jeśli zarówno `k` jak i `epsilon` są określone to będzie użyty tylko `epsilon`. Może się jednak zdarzyć, że nie znając dokładnie rozmieszczenia obiektów w przestrzeni wartość `epsilon` zostanie ustalona za nisko. Wówczas np. dla danego obiektu

Argument	Opis
`dist`	Macierz odmienności.
`ndim`	Maksymalny wymiar niskowymiarowej podprzestrzeni, w której mają zostać umieszczone obiekty, liczba naturalna mniejsza od wejściowej przestrzeni wysokowymiarowej.
`epsilon`	Wartość ε w wersji ε-Isomap.
`k`	Wartość k w wersji k-Isomap.
`path`	Metoda znajdowania ścieżki między obiektami niebędącymi sąsiadami, przyjmuje wartości `"shortest"` lub `"extended"`.
`fragmentedOK`	Wartość logiczna: wykonaj Isomap na największym nierozdrobnionym zbiorze obiektów (`TRUE`), przerwij funkcję z wynikiem error (`FALSE`).

Tablica 2.1: Argumenty funkcji `isomap()`.

w kole o promieniu ε nie będzie żadnego innego obiektu, czyli nie będzie on miał żadnego sąsiada. Taka sytuacja może prowadzić do rozdrobnienia danych i zbiór obiektów zostanie podzielony na kilka rozłącznych fragmentów. Wówczas nie powstanie spójny graf G przez co nie będzie możliwa dalsza wspólna analiza danych. O tym jak ma postępować funkcja `isomap()` w takiej sytuacji mówi argument `fragmentedOK` opisany w tabeli 2.1.

Wartość argumentu `path` odnosi się do sposobu znalezienia ścieżki łączącej obiekty nie będące sąsiadami, a tym samym znalezienia odległości je dzielących. W obu przypadkach (`"shortest"` i `"extended"`) w macierzy odmienności wartości odpowiadające odległościom obiektów nie będących sąsiadami są zamienione na `NA`. Dla `"shortest"` ścieżka jest szukana z wykorzystaniem wszystkich pozostałych wyrazów macierzy, czyli po ścieżkach wiodących po osiągalnych obiektach. Cała macierz jest uzupełniona w jednym etapie. Natomiast dla `"extended"` szukanie odbywa się fazami. W każdej z nich funkcja znajduje ścieżki, ale tylko takie, które wiodą po jednym obiekcie pośrednim. Odmienności zostają zmienione i w efekcie po każdym etapie powstaje nowa macierz odmienności. Fazy dla `path="extended"` są powtarzane do momentu zamienienia wszystkich brakujących wartości `NA` liczbami.

Wartością funkcji `isomap()` jest obiekt klasy `isomap`. Jednak by wyświetlić otrzymane wyniki należy zastosować do niego inną funkcję, ponieważ samo jego wywołanie dostarcza tylko informacji o użytych wartościach w funkcji `isomap()` do jego wygenerowania. Do obiektu tej klasy można użyć funkcje: `summary()`, `plot()` oraz `rgl.isomap()`. Dla uproszczenia opisu wymienionych funkcji przyjmuje się, że została zastosowana funkcja `isomap()`, w efekcie której powstała zmienna o nazwie `iso`.

Funkcja `summary()` użyta do obiektu klasy `isomap` zwraca dwie informacje. Pierwsza z nich to współrzędne otrzymanych punktów w niskowymiarowej podprzestrzeni. Współrzędne są podane do czwartego wymiaru włącznie, chyba że zostało ustalone `ndim`< 4. Przy skalowaniu n obiektów zostanie wypisana macierz o n wierszach (każdy z nich odpowiada innemu obiektowi) oraz maksymalnie 4 kolumnach (każda odpowiada innemu wymiarowi). Druga wartość funkcji `summary()` to siatka połączeń między obiektami na szkielecie kwadratowej macierzy dolnotrójkątnej (dla ustalenia uwagi nazwanej d), której poszczególne wiersze i odpowiadające im numerem kolumny odnoszą się do obiektów. O tej macierzy można myśleć jak o macierzy odmienności. Sieć połączeń jest przedstawiona w formie macierzy (nazwanej dla odróżnienia macierzą m) o 2 kolumnach. Pierwsza z nich to numery wierszy, druga numery kolumn odnoszące się do macierzy d. Patrząc na ustalony wiersz macierzy m (niech będzie on równy a, b) zwracana jest informacja o tym, że a-ty wiersz w macierzy d, a zatem

a-ty obiekt poddany skalowaniu jest połączony siecią z b-tą kolumną macierzy d, a więc b-tym obiektem w skalowaniu. By wyświetlić tylko macierz m z wyników `summary()` należy wpisać

```
summary(iso)$net   .
```

Natomiast dla

```
summary(iso)$points
```

wyświetlona zostanie tylko macierz ze współrzędnymi.

Funkcja `plot()` wyświetla dwuwymiarowy wykres zrzutowanych punktów. Przykładowy wzór na nią to

```
plot(iso, net = TRUE, type = "text", choice=c(1,6), n.col = "black")   .
```

Pierwszym argumentem jest obiekt klasy `isomap`. Kolejny (`net`) odpowiada, czy sieć połączeń między obiektami (ta sama, która jest wartością `summary()`) ma być umieszczona na wykresie (`TRUE`), czy nie (`FALSE`). Argument `type` określa sposób reprezentacji obiektów na wykresie. Przyjmuje wartości: `text`, `points`, `none`, które odpowiednio oznaczają, że obiekty będą przedstawione za pomocą napisów z ich nazwami, punktów lub nie będą widoczne. Możliwe jest również samodzielne wybranie, na podstawie których wymiarów zostanie wykonany wykres. Odpowiada za to `choice` - wektor długości dwa, którego wartości określają, które wymiary mają zastać wykorzystane. Ostatni argument, czyli `n.col` jest odpowiedzialny za kolor sieci połączeń, jeśli jest ona wyświetlona.

Funkcja `rgl.isomap()` pozwala wykonać dynamiczny wykres trójwymiarowy (z możliwością jego obrotu na ekranie za pomocą myszki). Podobnie jak w przypadku funkcji `plot()` możliwy jest wybór, które osie zostaną przedstawione. Przykładowa składnia tej funkcji wygląda następująco

```
rgl.isomap(iso, choice=c(1,3,4), type="p", ax.col="blue", web="red")   .
```

Argumenty powtarzające się z funkcją `plot()` to `choice` oraz `type`. Pierwszy z nich jest to wektor długości 3 odpowiadający wymiarom, które mają zostać zaprezentowane na wykresie. `type` tym razem przyjmuje wartości: p, t, n oznaczające odpowiednio: punkty, tekst, nic. Argument `ax.col` odnosi się do koloru osi współrzędnych, natomiast `web` do koloru sieci połączeń.

2.2. Pakiet `cluster`

W pakiecie `cluster` są zapisane m.in. funkcje mające zastosowanie w analizie skupień (podziale obiektów na względnie jednorodne grupy). [11] Poniżej omówiona zostanie jedna z nich służąca do klasyfikacji obiektów za pomocą algorytmu PAM (1.2). Ponadto zostanie przedstawiona dostępna również w tym pakiecie funkcja pozwalająca obliczyć sylwetkę podziału (1.4).

2.2.1. PAM

Kod wywołujący funkcję `pam()`, z domyślnie zapisanymi w programie `R` wartościami poszczególnych argumentów, ma następującą postać

```
pam(dat, k, diss = inherits(x, "dist"), metric = "euclidean",
medoids = NULL, stand = FALSE, do.swap=TRUE, cluster.only = FALSE,
pamonce = FALSE)
```

Argument	Opis
`dat`	Macierz danych.
`k`	Liczba skupień, `k`$\in \mathbb{N}_+$, `k`$<$ liczba obserwacji.
`diss`	Wartość logiczna określająca czy `dat` jest macierzą odmienności (`TRUE`), czy nie (`FALSE`).
`metric`	Miara odległości użyta do obliczenia odmienności między obiektami, do wyboru `"euclidean"` i `"manhattan"`.
`medoids`	Wektor długości `k` przyjmujący wartości naturalne.
`stand`	Wartość logiczna określająca czy wartości w macierz `dat` są wystandaryzowane przed policzeniem macierzy odmienności (`TRUE`), czy nie (`FALSE`). Tylko przy `diss=FALSE`.
`cluster.only`	Wartość logiczna określająca czy ma być wykonany wyłącznie podział (`TRUE`), czy policzone również pozostałe wartości funkcji `pam()` (`FALSE`).
`do.swap`	Wartość logiczna określająca czy faza zamiany (1.2.2) ma być pominięta (`FALSE`), czy nie (`TRUE`).
`pamonce`	Wartość logiczna lub liczba naturalna z przedziału [0,2] określająca liczbę kroków w fazie zamiany.

Tablica 2.2: Ważniejsze argumenty funkcji `pam()`.

natomiast opis jej ważniejszych argumentów znajduje się w tabeli 2.2.

Funkcja `inherits(x, "dist")` z pakietu `base` zwraca wartość prawdziwą (`TRUE`), jeśli obiekt `x` jest klasy `dist` lub fałszywą (`FALSE`), jeśli obiekt `x` jest innej klasy. [15]

Macierz danych wykorzystana do analizy - `dat` może być zarówno macierzą obserwacji (w wierszach są obiekty, które zostaną podzielone na skupienia, w kolumnach cechy) jak i macierzą odmienności (odmienności między elementami są odległościami pomiędzy nimi). W zależności od tego, która opcja zostanie zadeklarowana wymagane są od tej macierzy inne warunki do spełnienia. W pierwszym przypadku dopuszczalne są `NA`, ale pod warunkiem, że każda para obserwacji ma przynajmniej jedną wspólną cechę z niepominiętymi wartościami. Jeśli podawana jest macierz odmienności, `NA` nie są dozwolone. W obu przypadkach wszystkie wpisane wartości muszą być liczbowe.

Podanie macierzy odmienności jako `dat` może być istotne w przypadku, kiedy wymagane jest by policzyć odmienności między elementami za pomocą innej niż Euklidesowa lub Manhattan miary odległości, ponieważ funkcja `pam()` korzysta tylko z tych dwóch metryk. Do osobnego obliczenia macierzy odmienności z użyciem innej miary odległości służy funkcja `dist()`. [17]

W przypadku `stand=TRUE` standaryzacja odbywa się dla każdej cechy (kolumny) osobno. Dla wartości x_{ij} w macierz danych o n obiektach ilustruje ją poniższy wzór

$$\tilde{x}_{ij} = \frac{x_{ij} - \hat{x}_j}{D} \quad . \tag{2.1}$$

Od elementu x_{ij} z macierzy `dat` odjęta jest wartość średnia dla cechy, czyli $\hat{x}_j$

$$\hat{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij} \quad , \tag{2.2}$$

a następnie wynik podzielony jest przez średnie odchylenie bezwzględne oznaczone jako D

$$D = \frac{1}{n} \sum_{i=1}^{n} |x_{ij} - \hat{x}_j| \quad . \tag{2.3}$$

Wartość	Opis
`medoids`	Jeśli `pam()` działał z macierzą odmienności to jest to wektor ze współrzędnymi lub nazwami (jeśli były podane) wybranych w podziale medoidów, jeśli nie to fragment macierzy danych `dat`, w której wierszach są wyłącznie ustalone medoidy.
`id.med`	Wektor liczbowy odpowiadający numerom porządkowym medoidów jako elementów ze zbioru obserwacji.
`clustering`	Wektor podziału na skupienia, każdemu obiektowi zostaje przypisana liczba naturalna odpowiadająca numerowi skupienia, do którego został przydzielony.
`isolation`	Wektor długości odpowiadającej liczbie skupień przyjmujący wartości: `no, L, L*` (i-ta współrzędna wektora odpowiada i-temu skupieniu).
`clusinfo`	Macierz z odpowiednimi wartościami, w której wierszach są powstałe skupienia, a w kolumnach odpowiednio: `size, max_diss, av_diss, diameter, separation`.
`silinfo`	Lista ważniejszych wartości `silhouette()` (2.2.2).
`diss`	Macierz odmienności między obiektami.
`call`	Wzór funkcji `pam()` użyty do wygenerowania podziału.
`data`	Macierz danych.

Tablica 2.3: Ważniejsze wartości funkcji `pam()`.

Jak zostało opisane w rozdziale 1.2 algorytm PAM składa się z dwóch faz: budowy i zamiany. Zmieniając odpowiednie argumenty w funkcji `pam()` można modyfikować działanie, a nawet obecność każdej z nich. Domyślnie w funkcji `pam()` zarówno faza budowy, jak i zamiany są wykonywane. Oznacza to, że `medoids` przyjmuje wartość `NULL`, ponieważ medoidy są wyłaniane w fazie budowy. Jednak jeśli zostaną one wcześniej określone, to ta faza jest automatycznie pominięta. Kolejność podawania medoidów nie ma znaczenia, ponieważ cały algorytm działa niezależnie od kolejności obiektów, dotyczy to również kolejności występowania w macierzy `dat`.

Jeśli `do.swap=FALSE` to podział na skupienia, kończy się po wyborze medoidów i przypisaniu pozostałych elementów do medoidów, które są im najbliższe. Oznacza to, że faza zamiany nie jest wykonywana. Ale zamiast całkowicie ją pomijać można wykonać tylko kilka jej pierwszych etapów, jednak musi być wówczas zadeklarowane `do.swap=TRUE`. Domyślnie `pamonce=FALSE` co jest równoważne `pamonce=0` i oznacza, że faza zamiany zostanie w pełni wykonana. Dla `pamonce=1` lub `pamonce=TRUE` faza zamiany jest zakończona po pierwszej turze zamian, natomiast dla `pamonce=2` po drugiej. Należy zauważyć, że rezygnując z fazy zamiany można otrzymać podział, który nie będzie w pełni minimalizował sumy 1.16. Jednak taki wybór może mieć znaczenie, jeśli potrzeba podzielić zbiór obiektów wokół konkretnych medoidów i nie jest dopuszczalna możliwość ich zamiany.

Wynikiem opisywanej funkcji jest obiekt klasy `pam`, który jest listą. Jego ważniejsze wartości przedstawia tabela 2.3. W przypadku `cluster.only=TRUE` wartością funkcji `pam` będzie jednoelementowa lista, której elementem będzie wyłącznie `clustering`.

W zależności od tego jakie zostały użyte argumenty do wywołania funkcji `pam()`, będą również zwrócone różne wartości dla elementu `data`. Jeśli `stand=TRUE` to będzie to wystandaryzowana macierz wejściowa, a jeśli `diss=TRUE`, to ta wartość nie będzie w ogóle podana. W przeciwnych przypadkach `data`=`dat`.

Dla każdego skupienia, które powstało w efekcie zastosowania funkcji `pam()`, `clusinfo` dostarcza podstawowych informacji na jego temat. Są to: `size` (wielkość) - liczba obserwacji w skupieniu, `max_diss` (maksymalna odmienność) - maksymalna odmienność między obiektami w skupieniu a medoidem tego skupienia, `av_diss` (średnia odmienność) - średnia odmienność między obiektami w skupieniu a medoidem tego skupienia, `diameter` (średnica) - maksymalna odmienność między dwoma dowolnymi obiektami ze skupienia, `separation` (oddzielenie) - minimalna odmienność między obiektem ze skupienia oraz obiektem spoza skupienia.

Wartość `isolation` zwraca informację czy skupienia są izolowane (`L` lub `L*`), czy nie (`no`). Skupienie jest zakwalifikowane jako `L*`-izolowane jeśli jego średnica jest mniejsza niż jego oddzielenie. Natomiast jako `L`-izolowane jeśli dla każdej obserwacji x_i z tego skupienia maksymalna odmienność między x_i oraz każdym innym obiektem z tego skupienia jest mniejsza niż minimalna odmienność między x_i oraz każdą inną obserwacją spoza tego skupienia. W przeciwnym przypadku skupienie nie jest izolowanie (`no`).

Do ilustracji poprawności wyników podziału na skupienia może posłużyć sylwetka omówiona w rozdziale 1.4. W programie `R` do jej obliczenia służy oddzielna funkcja `silhouette()` omówiona w rozdziale 2.2.2. Jednak używając do podziału funkcji `pam()` wśród jej wartości są również podstawowe informacje o sylwetce, ale jedynie przy założeniu, że liczba powstałych skupień jest większa od 1 i jednocześnie mniejsza niż liczba obiektów w zbiorze danych.

Wśród wartości `silinfo` są wyróżnione:

1. `widths` (szerokość) - macierz o wymiarach $n \times 3$, w której wierszach są wszystkie obiekty ze zbioru danych, a kolumnach odpowiednio: numer skupienia do którego zakwalifikowano obiekt, numer skupienia sąsiedniego, wartość sylwetki,

2. `clus.avg.widths` (średnia szerokość skupienia) - średnia wartość sylwetki policzona dla każdego skupienia osobno,

3. `avg.width` (średnia szerokość) - średnia wartość sylwetki wszystkich obiektów ze zbioru danych.

Są to najważniejsze informacje dotyczące sylwetki, jednak nie wystarczające np. do wykonania jej wykresu. By go otrzymać trzeba użyć funkcji `silhouette()` (2.2.2).

2.2.2. Silhouette

Do sprawdzenia poprawności podziału można wykorzystać funkcję `silhouette()` z pakietu `cluster`. Oblicza ona sylwetkę na podstawie wzoru 1.26 przedstawionego w 1.4. Jej składnia ma następującą postać

```
silhouette(x, dist, dmatrix)   ,
```

natomiast tabela 2.4 przedstawia opis jej argumentów.

Głównym, a w niektórych przypadkach jedynym argumentem funkcji `silhouette()` jest `x`. Może to być obiekt np. klasy `pam` omówiony w rozdziale 2.2.1 lub inny powstały w skutek podziału na skupienia funkcją dostępną w pakiecie `R`. Może być również wektorem o kolejnych wartościach odpowiadającym przyporządkowaniu kolejnych obiektów do skupień. Jest też możliwość by tylko argumentowi x przypisać wartość `$clustering` jako jedyną wartość spośród wszystkich argumentów funkcji `silhouette()`. W zależności od tego jak został zadeklarowany pierwszy argument, determinuje to obecność pozostałych. Jeśli argumentowi `x` przypisano obiekt odpowiedniej klasy np. `pam` to pozostałe argumenty nie są konieczne,

Argument	Opis
`x`	Obiekt odpowiedniej klasy powstały z funkcji dokonującej podział, wektor lub wartość `$clustering`.
`dist`	Macierz odmienności powstała z `dist()`, funkcji pakietu `cluster`. [11]
`dmatrix`	Macierz odmienności.

Tablica 2.4: Argumenty funkcji `silhouette()`.

bo ma ona w swoich wartościach zapisaną macierz odmienności. W pozostałych przypadkach niezbędne jest jej ręczne zadeklarowanie, obowiązkowo musi być podana wartość `dist` lub `dmatrix` do wyboru.

Wynikiem funkcji `silhouette()` jest obiekt klasy `silhouette`. Jest to macierz o 3 kolumnach i tylu wierszach, ile było obiektów poddanych analizie skupień. Wiersze odpowiadają poszczególnym elementom i określają odpowiednio numer skupienia, do którego został przydzielony (pierwsza kolumna), numer skupienia sąsiedniego (druga) oraz wartość $s(\cdot)$ danego obiektu (trzecia). W tej macierzy najpierw jest lista elementów z pierwszego skupienia, potem drugiego itd.

Obiekt klasy `silhouette` może być użyty jako argument dwóch funkcji. Pierwsza z nich to `summary()`. Składnia tej funkcji ma następującą postać

```
summary(sil, FUN)   ,
```

gdzie `sil` jest obiektem klasy `silhouette`, a `FUN` jest funkcją wykorzystaną do działania na wartościach $s(\cdot)$. Domyślnie `FUN=mean`. Oznacza to, że obliczana jest średnia arytmetyczna dla wartości $s(\cdot)$. Przy omówieniu wartości funkcji `silhouette()` przyjęte zostało, że wartość tego argumentu nie została zmieniona.

Wynikiem tej funkcji jest obiekt klasy `summary.silhouette`, który jest listą o sześciu elementach. Pierwszy to `si.summary` będący podsumowaniem wartości $s(\cdot)$ dla wszystkich obiektów. Jest to wartość funkcji `summary()` z pakieta `base` zastosowana do wektora wszystkich wartości $s(\cdot)$. [15] Kolejny argument - `clus.avg.widths` podaje średnią szerokość sylwetki dla każdego skupienia, natomiast `avg.width` dla całego zbioru obiektów. Liczba elementów w poszczególnych skupieniach jest wartością `clus.sizes`. Wartość `call` odpowiada za wzór na funkcję `silhouette()` dla `x` będącego wektorem lub wartością `$clustering`. Jeśli `x` jest innej klasy to zostanie zwrócony wzór na zastosowaną funkcję podziału do jego wygenerowania. Ostatni argument to `Ordered`, który jest wartością logiczną równą `TRUE` jeśli elementy w `sil` zostały posortowane w sposób nierosnący względem $s(\cdot)$ w obrębie swoich skupień oraz `FALSE` jeśli nie.

Drugą funkcją, którą można zastosować do obiektu klasy `silhouette` jest `plot()`. Wykonuje ona wykres sylwetki podziału w sposób opisany w 1.4.2. Kod funkcji z domyślnie użytymi wartościami w programie `R` wygląda następująco

```
plot(x, nmax.lab = 40, max.strlen = 5, main = NULL, sub = NULL,
xlab = expression("Silhouette width "* s[i]), do.n.k = TRUE, col = "gray",
do.col.sort = length(col) > 1, border = 0, do.clus.stat = TRUE)   .
```

Podstawowym argumentem funkcji `plot()` jest obiekt klasy `silhouette` przypisany do argumentu `x`. Kolejny to `nmax.lab`, który określa maksymalną dopuszczalną liczbę obiektów przy jakiej będą wypisane ich nazwy na wykresie obok ich wartości $s(\cdot)$. Natomiast `max.strlen` to liczba naturalna określająca ile pierwszych znaków z nazw obiektów ma być

wyświetlonych na etykietach. Nazwę tytułu określa `main`, a podtytułu `sub`. Dodatkowo `xlab` jest podpisem pod osią odciętych. Kolejny argument to `do.n.k` mówiący czy na wykresie ma być napisane jaka była liczba obiektów w zbiorze danych oraz liczba otrzymanych skupień (`TRUE`), czy nie (`FALSE`). Za kolor wykresu odpowiada `col`, który może być zarówno wektorem z nazwami kolorów, jak i np. funkcją `heat.colors()` z pakietu `grDevices`. [16] Natomiast `do.col.sort` określa czy kolory mają być przypisane każdy do innego skupienia (`TRUE`), czy mają być używane na zmianę do kolejnych słupków z wartościami $s(\cdot)$ (`FALSE`). Argument `border` ustala czy te słupki mają być obrysowane na czarno (`TRUE`), czy nie (`FALSE`). Ostatni, czyli `do.clus.stat` determinuje, czy na wykresie mają być umieszczone informacja o liczbie obiektów w każdym skupieniu oraz ich średniej wartości $s(\cdot)$.

2.3. Pakiet `isopam`

Do prawidłowego działania pakietu `isopam` niezbędne są jeszcze pakiety: `cluster`, `permute` oraz `vegan` instalowane automatycznie przy załadowywaniu pakietu `isopam`. Pierwszy z nich odpowiada za część algorytmu związaną z PAM, drugi za generowanie permutacji danych, natomiast ostatni m.in. za użycie Isomap.

Pakiet `isopam` jest wyposażony w dwie funkcje: `isopam()` oraz `isotab()`. [19] Poniżej omówiona zostanie pierwsza z nich odnosząca się do algorytmu Isopam opisanego w 1.3.

2.3.1. Isopam

Do podziału zbioru obiektów na rozłączne skupienia za pomocą algorytmu Isopam służy funkcja `isopam()`. W programie `R` można zastosować jej wersję hierarchiczną, jak i niehierarchiczną. Jednak w tym rozdziale uwaga będzie skupiona na tej drugiej, podobnie jak zostało to zrobione w części teoretycznej. Poniżej przedstawiona jest jej składnia z domyślnie stosowanymi wartościami

```
isopam (dat, c.fix = FALSE, c.opt = TRUE, c.max = 6, l.max = FALSE,
stopat = c(1,7), sieve = TRUE, Gs = 3.5, ind = NULL, distance = "bray",
k.max = 100, d.max = 7)
```

natomiast opis argumentów znajduje się w tabeli 2.5.

Najważniejszym argumentem tej funkcji jest macierz danych. Ze względu na to, że Isopam służy głównie do podziału na skupienia zbiorowisk czy to roślinnych, czy zwierzęcych, najczęściej ma ona w wierszach obszary (zmienne użyte do podziału), a w kolumnach gatunki (cechy), natomiast w poszczególnych polach częstotliwość ich występowania. Macierz ta musi być numeryczna z wypełnionymi wszystkimi miejscami, czyli nie są dozwolone brakujące wartości - `NA`. Dodatkowo musi mieć minimum 3 wiersze, tak by w podziale brały udział minimum 3 obiekty. Nie może być ona również macierzą odmienności, ponieważ przy szukaniu optymalnego rozwiązania Isopam korzysta z informacji zawartych w macierzy częstotliwości występowania.

Zastosowanie pakietu `vegan` daje możliwość użycie różnych miar odległości w funkcji `isopam()`. [1] Do wyboru są m.in. : Manhattan, Euklidesowa, Canberra, Bray-Curtis'a, Kulczyńskiego, Jaccard'a, ale również każda inna po wcześniejszym jej zapisaniu. Do części z nich należy dodatkowo zainstalować pakiet `proxy`. [2] Wybór odpowiedniej miary odmienności jest istotny, ponieważ powinna ona oddawać różnice w składzie gatunkowym zbiorowisk, czyli ekologiczny dystans je dzielący. Zbiorowiska, które mają wspólnych wiele gatunków powinny być mało oddalone od siebie, natomiast te, które mają kilka lub wcale wspólnych gatunków powinny być daleko od siebie. Taką własność ma miara Bray–Curtis domyślnie stosowana przez

Argument	Opis
`dat`	Macierz danych.
`c.fix`	Liczba skupień, które mają powstać.
`c.max`	W przypadku podziału hierarchicznego maksymalna liczna skupień w jednym etapie podziału. Tylko przy `c.opt=TRUE`.
`c.opt`	Wartość logiczna, prawdziwa (`TRUE`) lub fałszywa (`FALSE`).
`l.max`	Maksymalna liczba poziomów w podziale hierarchicznym.
`stopat`	Wektor dwuwymiarowy z regułami zatrzymania w podziale hierarchicznym.
`Gs`	Wartość progowa wystandaryzowanej statystyki G (1.3.4), domyślnie 3.5.
`sieve`	Wartość logiczna, prawdziwa (`TRUE`) lub fałszywa (`FALSE`).
`ind`	Wektor z nazwami kolumn gatunków uznanych za wskaźnikowe.
`distance`	Miara odległości użyta w Isomap.
`k.max`	Maksymalna wartość k w k-Isomap.
`d.max`	Maksymalna liczba wymiarów uzyskana w Isomap.

Tablica 2.5: Argumenty funkcji `isopam()`.

`isopam()`. [4] Jest ona liczona w następujący sposób

$$o(x_i, x_j) = \frac{\sum_{k=1}^{n} \mid x_i^k - x_j^k \mid}{\sum_{k=1}^{n} (x_i^k + x_j^k)} \tag{2.4}$$

dla $x_i = (x_i^1, x_i^2, \ldots, x_i^n)$, $x_i \in \mathbb{R}^n$. Podobną własność ma miara Kulczyńskiego, która musi być już dodatkowo deklarowana wśród argumentów funkcji `isopam()`. Jest on liczona w następujący sposób

$$o(x_i, x_j) = 1 - \frac{1}{2}\left(\frac{\sum_{k=1}^{n} \min(x_i^k, x_j^k)}{\sum_{k=1}^{n} x_i^k} + \frac{\sum_{k=1}^{n} \min(x_i^k, x_j^k)}{\sum_{k=1}^{n} x_j^k}\right) \quad . \tag{2.5}$$

W przypadku podziału hierarchicznego w procesie powstawania dendrogramu w każdym jego etapie mogą następować podziały na różną liczbę grup. Jeśli argument `c.opt` ma wartość fałszywą (`FALSE`) to następują podziały na dwie grupy, jeśli prawdziwą (`TRUE`) to liczba skupień w podziale jest wybrana najkorzystniej spośród $\{2, 3, \ldots$ `c.max`$\}$.

Argument `c.fix` jest używany wyłącznie w przypadku podziału niehierarchicznego. Wówczas `c.opt, c.max` oraz `l.max` nie są brane pod uwagę. Jeśli `c.fix` nie jest podane to wykonany jest podział hierarchiczny. Ponadto dla `l.max`= 1 powstaje podział niehierarchiczny.

Wektorem z regułami zatrzymania w przypadku podziału hierarchicznego jest `stopat`. Jego pierwszą wartością jest liczba gatunków wskaźnikowych, które muszą zostać wybrane w obrębie skupienia. Druga to wystandaryzowana wartość G, którą muszą osiągnąć te gatunki (1.3.1).

Przy optymalizacji brane są pod uwagę gatunki wraz z wyliczoną dla nich statystyką G. Jeśli `sieve` jest ustalone na wartość prawdziwą (`TRUE`) to tylko gatunki przekraczające `Gs` są brane pod uwagę, jeśli fałszywą (`FALSE`) to wszystkie.

Efektem działania funkcji `isopam()` jest obiekt klasy `isopam`. Jego wartości zostały przedstawione w tabeli 2.6. Najważniejszą z nich jest uzyskany podział. Jest on zapisany pod `flat` i przedstawia obserwacje z przypisanymi numerami skupień, do których zostały zakwalifikowane. Do tej wartości można zastosować funkcję `plot()`, która przedstawi przyporządkowanie

Wartość	Opis
`call`	Wzór funkcji isopam użyty do wygenerowania podziału.
`distance`	Użyta miara odmienności.
`flat`	Wynik podziału.
`hier`	W przypadku podziału hierarchicznego tabela obserwacji.
`medoids`	Medoidy wykorzystane w podziale.
`analytics`	Tabela z parametrami użytymi do podziału.
`dendro`	W przypadku podziału hierarchicznego obiekt klasy `hclust` pozwalający na graficzną prezentację dendrogramu.
`dat`	Macierz danych.

Tablica 2.6: Wartości funkcji `isopam()`.

obiektów utożsamionych z odpowiadającymi im numerami porządkowymi w zbiorze danych (oś odciętych) do skupień (oś rzędnych).

Wartość `analytics` jest tabelą zawierającą takie dane jak: powstała liczba skupień, użyty wymiar podprzestrzeni w Isomap, minimalna możliwa wartość k w k-Isomap, użyta wartość k w k-Isomap, maksymalna możliwa wartość k w k-Isomap, liczbę gatunków wskaźnikowych przekraczających wartość progową `Gs`, średnia wartość `Gs` dla gatunków wskaźnikowych czy średnia wartość `Gs` dla wszystkich gatunków. Tabelą jest również `hier`. Generowana tylko w podziale hierarchicznym, zawiera obserwacje z przypisanymi im numerami skupień z wyszczególnieniem poszczególnych poziomów w podziale (poziomów musi być więcej niż jeden).

Rozdział 3

Analiza danych rzeczywistych

W poniższym rozdziale zostanie opisane zastosowanie w praktyce algorytmu Isopam do rzeczywistych danych z zakresu fitosocjologii, czyli działu botaniki zajmującego się badaniem zbiorowisk roślinnych oraz ich klasyfikowaniem. Dane te dotyczą zbiorowisk roślinnych zaobserwowanych w dolinie Bogdanki w Poznaniu. Zostały one zebrane i opracowane przez Panią Monikę Zgrabczyńską, we współpracy z którą zostanie dokonana analiza otrzymanych wyników grupowania.

3.1. Wprowadzenie tematyczne

Podstawowym pojęciem wielokrotnie wykorzystywanym w poniższej analizie skupień jest zbiorowisko roślinne zdefiniowane w rozdziale 1.3 oraz krąg zbiorowisk, który jest zestawem zbiorowisk zarówno tych naturalnych, jak i antropogenicznych.

Kiedyś cały badany teren był lasem, jedynymi zbiorowiskami były naturalne zbiorowiska leśne. Oczywiście było ich wiele rodzai. Ale pod wpływem czasu, ale głównie pod wpływem działalności człowieka nastąpiło zróżnicowanie szaty roślinnej. Znaczna część zbiorowisk naturalnych uległa zniszczeniu. Jednak powstały w ich miejsce nowe, dostosowane do zmienionych warunków, zbiorowiska zastępcze. Pomimo zmian zbiorowiska zastępcze cały czas wykazują silny związek ze zbiorowiskami leśnymi, od których się wywodzą. Postawiony problem, do którego rozwiązania posłuży Isopam, polega na wyodrębnieniu wśród listy zbiorowisk dynamicznych kręgów zbiorowisk roślinnych.

Definicja 13 (Dynamiczny krąg zbiorowisk roślinnych) *Dynamiczny krąg zbiorowisk to wszystkie występujące na danym siedlisku zbiorowiska zastępcze wraz z właściwym dla nich zbiorowiskiem naturalnym - leśnym zbiorowiskiem końcowym. [22]*

Definiowanie dynamicznego kręgu zbiorowisk polega na znalezieniu grupy zbiorowisk zastępczych, dla danego naturalnego zbiorowiska leśnego, wykazujących związek z nim wynikający z sąsiedztwa występowania, podobieństwa składu florystycznego oraz warunków siedliska. [22] Innymi słowy dynamiczny krąg zbiorowisk będzie tworzyła grupa zbiorowisk złożona ze zbiorowiska leśnego oraz zbiorowisk zastępczych, które są podobne do niego np. ze względu na występowanie tych samych gatunków roślinnych. Oznacza to, że podział na skupienia będzie odbywał się wokół naturalnych zbiorowisk leśnych, natomiast pozostałe zostaną przypisane do grupy z tymi, z którymi są najsilniej związane. W ten sposób powstaną pełne składy skupień.

Wyodrębnienie dynamicznych kręgów zbiorowisk roślinnych ma zastosowanie planistyczne dla badanego terenu. Daje ono możliwość potencjalnego odtworzenia naturalnego zbiorowiska

leśnego. Sadząc odpowiednie gatunki roślin jest możliwe ponowne zalesienie obszaru. Nowy las nie będzie identyczny z pierwotnym, ale bardzo podobny. Natomiast sadzenie nieodpowiednich roślin, czyli charakterystycznych dla zbiorowiska leśnego, od którego nie pochodzi dane zbiorowisko zastępcze, może trwale uszkodzić florę.

3.2. Opis danych

Wyodrębnienie dynamicznych kręgów zbiorowisk roślinnych odbędzie się na podstawie porównania składu florystycznego zbiorowisk. Zbiór danych stanowi 85 zbiorowisk roślinnych z terenu doliny Bogdanki (są to obiekty, które zostaną poddane grupowaniu) z wyróżnionymi wartościami systematycznymi dla 31 grup gatunków roślin (cechy, na podstawie których odbędzie się grupowanie). Brane są pod uwagę całe grupy gatunków np. gatunki bagienne, a nie pojedynczy gatunki, by móc odrzucić gatunki nieistotne, czyli takie które nie zakwalifikowały się do żadnej z grup i stanowiłyby „szum" (1.3). Dodatkowym argumentem jest uproszczenie zbioru danych początkowo składającego się z 394 typów roślin z uwzględnieniem ich stałości fitosocjologicznych względem 85 zbiorowisk.

Definicja 14 (Stałość fitosocjologiczna) *Względna częstość występowania danego gatunku w obrębie zespołu roślinnego.[21]*

Stałości są liczone na podstawie wcześniej wykonanych zdjęć fitosocjologicznych terenu. Oddają one częstotliwość występowania gatunku na badanym obszarze. Może ona być przedstawione w postaci pięciostopniowej skali, w której V oznacza składniki stałe tj. występujące w 81—100% wszystkich zdjęć, IV składniki częste występujące w 61—80% zdjęć, III składniki średnio częste występujące w 41—60% zdjęć, II składniki niezbyt częste występujące w 21—40% zdjęć oraz I składniki rzadkie lub sporadyczne występujące w 1—20% zdjęć. Stałość może też być przedstawiona jako ułamek, którego licznikiem jest liczba zdjęć, w których występował dany gatunek, a mianownikiem liczba uwzględnianych zdjęć. [21]

Wartość systematyczną grupy gatunków oblicza się na podstawie stałości biorąc pod uwagę poszczególne grupy gatunków według następującego wzoru [21]

$$D(\%) = \frac{G \cdot S}{100} \quad ,$$

gdzie
S - przeciętna stałość grupy,
G - udział zbiorowy danej grupy.
Wartości S i G obliczane są natomiast według następujących wzorów

$$S(\%) = \frac{g}{z} \cdot n \cdot 100 \quad ,$$

$$G(\%) = \frac{g}{t} \cdot 100 \quad ,$$

gdzie
g - suma liczników z ułamków stałości gatunków danej grupy w zbiorowisku,
z - liczba gatunków w danej grupie,
n - liczba zdjęć dla zbiorowiska,
t - suma wystąpień wszystkich gatunków w zbiorowisku.

Tego typu dane dostarczają informacji w jaki sposób różnią się od siebie zbiorowiska. Jednak można je również przedstawić w formie binarnej. Wówczas zwrócona zostanie informacja o tym, czy zbiorowiska się różnią. Tabela 3.1 przedstawia fragment analizowanych danych z uwzględnionymi wartościami systematycznymi grup gatunków, natomiast tabela 3.2 te same dane w formie binarnej. Kolumny odpowiadają zbiorowiskom, natomiast wiersze grupom gatunków.

	All-Cha	Car-Rub	Fal-Hum	Con-Agr	Con-Brom	···
Querco-Fagetea	0,0034	0,0099	0,0076	0,0017	0,0000	
Carpinion betuli	0,0022	0,0000	0,0038	0,0050	0,0000	
Alnion incane	0,0011	0,0000	0,0057	0,0000	0,0015	
Alnetea glutinosae	0,0011	0,0042	0,0000	0,0000	0,0030	
Artemisietea vulgaris	0,0061	0,0085	0,0189	0,0297	0,0106	
···						

Tablica 3.1: Fragment analizowanych danych w formie niebinarnej.

	All-Cha	Car-Rub	Fal-Hum	Con-Agr	Con-Brom	···
Querco-Fagetea	1	1	1	1	0	
Carpinion betuli	1	0	1	1	0	
Alnion incane	1	0	1	0	1	
Alnetea glutinosae	1	1	0	0	1	
Artemisietea vulgaris	1	1	1	1	1	
···						

Tablica 3.2: Fragment analizowanych danych w formie binarnej.

3.3. Podział na skupienia

Do podziału zostanie wykorzystana niehierarchiczna wersja funkcji `isopam()` z wcześniej określonymi medoidami, a tym samym liczbą skupień. Analiza odbędzie się zarówno na podstawie danych binarnych, jak i zwykłych.

3.3.1. Metoda podziału

Podział zbiorowisk na skupienia zostanie wykonany za pomocą algorytmu Isopam. Powstałe grupy będą odpowiadały dynamicznym kręgom zbiorowisk roślinnych. Te kręgi będą budowane wokół naturalnych zbiorowisk leśnych, do czego posłuży druga część algorytmu Isopam, czyli PAM. Zbiorowiska leśne będą medoidami, natomiast pozostałe zbiorowiska (czyli zbiorowiska zastępcze) zostaną przydzielone do grup reprezentowanych przez te medoidy, z którymi są najsilniej związane. W ten sposób powstaną skupienia. W algorytmie PAM po pierwszym podziale obiektów następuje faza zamiany (1.2.2), w trakcie której wszystkie medoidy są ponownie rozważane i jeśli jest możliwość otrzymania podziału bardziej minimalizującego sumę 1.16 to zostają one zastąpione. W przypadku dynamicznych kręgów nie ma możliwości zamiany medoidów. To zbiorowiska naturalne stanowią najbardziej centralne obiekty skupień i pozostałe zbiorowiska, czyli zbiorowiska zastępcze muszą zostać przydzielone do nich, a więc do lasów, od których pochodzą. Nie są one centralne w sensie położenia

w skupieniu, ale ze względu na fakt, że są reprezentantami swoich grup. Nie ma więc możliwości, by zbiorowisko zastępcze było medoidem, ponieważ jego szata roślinna została już w znacznym stopniu zmieniona, a jedynie zbiorowiska naturalne oddają pierwotną roślinność. Oznacza to, że w PAM należy nie wykonywać fazy zamiany.

Pośrednia miara odmienności zastosowana w Isopam jest pomocna w przypadku dużej beta różnorodności, czyli dużych różnic między zbiorowiskami w ich składzie gatunkowym, a nie w liczbach gatunków. Znaczące odległości powstaną między zbiorowiskami, które nie mają wspólnych żadnych gatunków, ale są połączone ścieżką pośrednich zbiorowisk. [7]

Naturalne zbiorowiska leśne, które będą pełniły funkcje medoidów zostały wskazane przez eksperta. Są to

1. ols porzeczkowy (łac. *Carici elongatae-Alnetum*) - Car.Aln (35),
2. murawa lepnicowo-kostrzewowa (łac. *Silene otitae-Festucetum trachyphyllae*) - Sil.Fes (36),
3. łęg jesionowo-olszowy (łac. *Fraxino-Alnetum*) - Fra.Aln (40),
4. grąd środkowoeuropejski (łac. *Galio sylvatici-Carpinetum*) - Gal.Car (42),
5. łęg dębowo-wiązowy (łac. *Querco-Ulmetum*) - Que.Ulm (46).

Po nazwach zbiorowisk przedstawione są skróty użyte w tabeli danych, a w nawiasach numery kolumn, w których występują te zbiorowiska. W zbiorze nie ma innych naturalnych zbiorowisk leśnych oprócz wskazanych. Pozostałe występujące zbiorowiska leśne nie mają charakteru zbiorowisk naturalnych np. są to leśne zbiorowiska zastępcze.

3.3.2. Użyte argumenty funkcji `isopam()`

W programie `R` pełne tabele danych, których fragmenty przedstawiają 3.1 i 3.2, zostały zapisane pod zmiennymi `daneb` i `dane` odpowiednio dla danych binarnych i zwykłych. Jednak w funkcji `isopam()` argument `dat` odpowiadający macierzy danych ma w wierszach obiekty, a w kolumnach cechy. Dlatego niezbędne jest transponowanie zmiennych. Odpowiadają za to poniższe formuły

```
data<-t(dane)
datab<-t(daneb)   .
```

Tak wygenerowane zmienne `data` oraz `datab` zostaną użyte jako argumenty `dat` w funkcji podziału.

W funkcji `isopam()` nie ma możliwości wskazania medoidów. Ale w funkcji `pam()` (jest częścią funkcji `isopam()` i na jej podstawie dokonany jest podział) taka możliwość już jest. Bardzo istotnym fragmentem jest możliwość przypisania naturalnym zbiorowiskom leśnym funkcji medoidów, dlatego na potrzeby tego podziału funkcja `isopam()` została zmodyfikowana w sposób opisany w Dodatku B. Składnia nowej funkcji wygląda następująco

```
i(dat, c.fix = FALSE, c.opt = TRUE, c.max = 6, l.max = FALSE,
stopat = c(1, 7), sieve = TRUE, Gs = 3.5, ind = NULL, distance = "bray",
medoids=NULL, k.max = 100, d.max = 7)   .
```

Dodanym argumentem jest `medoids`. Jest on użyty w dokładnie taki sam sposób jak został opisany w rozdziale 2.2.1, ponieważ jest on wykorzystywany tylko w funkcji `pam()` będącej fragmentem `isopam()`. Deklarując medoidy deklarowana jest jednocześnie liczba skupień,

która ma powstać. Dlatego też należy przypisać argumentowi `c.fix` wartość 5, czyli liczbę naturalnych zbiorowisk naturalnych.

Ostatnim zadeklarowanym argumentem jest rodzaj wykorzystanej odległości. Dla danych binarnych będzie to miara Kulczyńskiego, natomiast dla zwykłych Bray-Curtis. Pozostałe argumenty zostaną niezmienione z domyślnie przypisanymi im wartościami.

Ostatecznie do wygenerowania podziału, czyli wyodrębnienia dynamicznych kręgów zbiorowisk roślinnych, zostały wykorzystane następujące deklaracje funkcji

```
podzial<-i(data, c.fix = 5, medoids = c(35, 36, 40, 42, 46))
```

dla podziału na danych z wartościami systematycznymi grup gatunkowych oraz

```
podzial_bin<-i(datab, c.fix = 5, medoids = c(35, 36, 40, 42, 46),
distance = "kulczynski")
```

dla podziału danych binarnych.

3.4. Wyniki

W wyniku zastosowania funkcji `isopam()` powstały dwie zmienne `podzial` i `podzial_bin`. Są one klasy `isopam`. Poniżej zostaną przedstawione ich najważniejsze wartości.

W obu podziałach powstało po 5 skupień, każde reprezentowane jest przez swój medoid. Nazwami skupień są kolejne cyfry 1, 2, 3, 4, 5. Nie oznacza to jednak, że skupienie 1 w przypadku `podzial` odpowiada skupieniu 1 dla `podzial_bin`. Należy zwrócić uwagę na medoidy je reprezentujące. Tabele 3.3 oraz 3.4 przedstawiają przyporządkowanie skupień do medoidów dla `podzial` oraz `podzial_bin` odpowiednio. Zostały one wyświetlone na ekranie po wpisaniu

```
podzial$medoids
```

i

```
podzial_bin$medoids
```

.

Kolejność podawania medoidów przy deklarowaniu wartości funkcji nie ma znaczenia i nie ma ona wpływu na ostateczną kolejność skupień. Cały algorytm PAM, a tym samym Isopam działa niezależnie od kolejności obiektów, dotyczy to również kolejności występowania w macierzy danych.

Skupienie 1	Skupienie 2	Skupienie 3	Skupienie 4	Skupienie 5
Gal.Car	Que.Ulm	Sil.Fes	Fra.Aln	Car.Aln

Tablica 3.3: Skupienia i reprezentujące je medoidy dla `podzial`.

Skupienie 1	Skupienie 2	Skupienie 3	Skupienie 4	Skupienie 5
Fra.Aln	Que.Ulm	Sil.Fes	Gal.Car	Car.Aln

Tablica 3.4: Skupienia i reprezentujące je medoidy dla `podzial_bin`.

Najważniejszą wartością funkcji `isopam()` jest jednak lista ze składem powstałych skupień. By ją wyświetlić należy wpisać

```
podzial$flat
```

dla danych z wartościami systematycznymi grup gatunków oraz dla danych binarnych

`podizl_bin$flat` .

Po wypisaniu powyższych formuł zostanie wyświetlona lista zbiorowisk z przypisanymi im numerami skupień, do których zostały zakwalifikowane. Zbiorowiska będą posortowane w tej samej kolejności w jakiej występowały w macierzy danych. Jest to jednak nieczytelna forma, ze względu na dużą liczbę obiektów poddanych analizie skupień. By wyświetlić uporządkowaną listę zbiorowisk względem skupień wystarczy użyć funkcji `sort()`. Po wypisaniu formuł

```
sort(podzial$flat)
```

oraz

```
sort(podizl_bin$flat)
```

zostanie wyświetlona lista zbiorowisk z ich przynależnością do skupień w takiej kolejności, że najpierw będą przedstawione zbiorowiska ze skupienia 1, potem zbiorowiska ze skupienia 2 itd. Tabele 3.5 oraz 3.6 zawierają pełen skład powstałych skupień. Medoidy zostały w nich wyróżnione **pogrubioną** czcionką.

Skupienie 1	Skupienie 2	Skupienie 3	Skupienie 4	Skupienie 5
All.Cha	Car.Rub	Con.Agr	Eup.can	Car.gra
Lol.Pla	Fal.Hum	Con.Brom	Imp.par	Cic.Car
Pot..Fes	Hel.dec	Dau.Pic	Urt.Con	Cla.mar
Pru.Pla	Son.arch	Geo.Che	Ang.Cir	The.phr
Pot.ans	Poe.ann	Arr.ela	Fil.Ger	Typ.ang
Aeg.Sam	Cor.Cor	**Sil.Fes**	Sel.Mol	Typ.lat
Gal.Car	C.Cagr	Tor.jap	Ste.Desc	Car.Ber
LZZ..GC.	Arm.Fes..p.	Sap.off	Car.Chr	Sal.cin
LZZ.p.GC.	LZZ..QU.	Rud.Sol	zb..CA	**Car.Aln**
LZZ..PQ.	**Que.Ulm**	Ber.inc	**Fra.Aln**	Cal.can
	Rub.ida	Cal.epi	Fra.Aln.c	Car.rip
	Rum.obt			Iri.pse
	Leo.Bal			Gly.pli
	Arc.lap			Sci.lac
	Aeg.Pet			Bid.cer
	Agr.Aeg			Bid.Pol
	Ant.syl			Bid.Rum
	Chaaro			Cor.mas
	Sol.can			zb..Pa
	zb..Jun			LZZ..C.Al..
	Ran.rep			Pol.cus
	Car.pan			Phr.com
	Vic.tet			Car.acu
	Vit.vin			
	Am.Fru			
	Ant.nit			
	Euo.Pru			
	Tan.Art			
	Epi.hir			
	Che.Rob			

Tablica 3.5: Skład skupień na podstawie danych z wartościami systematycznymi.

Do oceny grupowania posłużyła tabela wykonana metodą intuicyjno-ekspercką. Dynamiczne kręgi zbiorowisk na terenie doliny Bogdanki w Poznaniu zostały wyodrębnione na

Skupienie 1	Skupienie 2	Skupienie 3	Skupienie 4	Skupienie 5
All.Cha	Fal.Hum	Con.Agr	Geo.Che	Imp.par
Car.Rub	Son.arch	Dau.Pic	Hel.dec	Urt.Con
Con.Brom	Car.acu	**Sil.Fes**	**Gal.Car**	Cic.Car
Eup.can	Aeg.Sam	Cor.Cor	LZZ..QU.	Cla.mar
Ang.Cir	**Que.Ulm**	C.Cagr	LZZ..GC.	The.phr
Arr.ela	Rum.obt	Arm.Fes..p.	Leo.Bal	Typ.ang
Fil.Ger	Aeg.Pet	Tor.jap	Arc.lap	Typ.lat
Lol.Pla	Pol.cus	Sap.off	Agr.Aeg	Poe.ann
Pot..Fes		Rud.Sol	Ant.syl	Car.Ber
Pru.Pla		Sol.can	Chaaro	Car.Chr
Sel.Mol		zb..Jun	Vit.vin	zb..CA
Ste.Desc		Vic.tet	Ant.nit	Sal.cin
Car.gra		Am.Fru	Che.Rob	**Car.Aln**
Pot.ans		Ber.inc		Fra.Aln.c
Fra.Aln		Tan.Art		Rub.ida
LZZ.p.GC.				Ran.rep
LZZ..PQ.				Cal.can
Euo.Pru				Car.pan
Cal.epi				Car.rip
				Iri.pse
				Gly.pli
				Sci.lac
				Bid.cer
				Bid.Pol
				Bid.Rum
				Cor.mas
				zb..Pa
				LZZ..C.Al..
				Epi.hir
				Phr.com

Tablica 3.6: Skład skupień na podstawie danych binarnych.

podstawie dostępnej literatury i opisanych w niej badań oraz obserwacji autorów o występowaniu zbiorowisk na różnych typach siedlisk. Metoda ta polega na tym, że zostały zebrane informacje o tym, w jakich kręgach dane zbiorowiska były obserwowane, z uwzględnieniem charakteru terenu badań. Dlatego została wprowadzona trzystopniowa skala:

1. • zbiorowisko roślinne przewodnie danego kręgu,
2. ∘ zbiorowisko roślinne często obecne w danym kręgu, ale nie wskaźnikowe,
3. · zbiorowisko roślinne pojawiające się w różnych kręgach - towarzyszące.

Skala ta oznacza stwierdzony dotąd (poprzez obserwację) zakres występowania zbiorowisk w odniesieniu do dynamicznych kręgów. Tabela początkowo była wymiarów 85×5, jednak podzielono ją na dwie części. W jej wierszach są kolejne zbiorowiska, natomiast w kolumnach naturalne zbiorowiska leśne, czyli wybrane medoidy. Każda komórka w tabeli odnosi się do zakresu zmienności występowania danego zbiorowiska w obrębie dynamicznego kręgu reprezentowanego przez dany medoid. Pierwsza część przedstawia listę potencjalnych zbiorowisk przewodnich dla pięciu kręgów. W drugiej zestawiono pozostałe zbiorowiska charakteryzujące się szerszą skalą ekologiczną, czyli takie które mogą okazać się diagnostyczne dla dwóch i więcej dynamicznych kręgów.

O jakości grupowania będzie świadczył stopień pokrycia zmienności występowania zbiorowisk przedstawiony w tabeli eksperckiej ze stopniem zmienności wygenerowanym dla wyników `podzil` oraz `podzial_bin`.

Przykładowo Sal.cin to zbiorowisko wierzby szarej. Może występować zazwyczaj w kręgu olsu, czyli Car.Aln. Zostało zatem oznaczone w tabeli eksperckiej stopniem •. Ale może też występować na nieco suchszych siedliskach łęgowych Fra.Aln, zostało zatem oznaczone ∘. Nie występuje na siedliskach suchych typu Gal.Car i nigdy nie zostanie spotkane na siedlisku suchym Sil-Fes, dlatego te pola zostały puste. Jeśli wyniki grupowania przypisałyby to zbiorowisko do skupienia reprezentowanego przez Car.Aln lub Fra.Aln (4 lub 5 dla `podzial` oraz 1 lub 5 dla `podzial_bin`) będzie to miało sens ekologiczny. W obu podziałach to zbiorowisko trafiło do skupienia razem z Car.Aln, więc został osiągnięty pozytywny wynik klasyfikacji w odniesieniu do metody intuicyjno-eksperckiej.

Natomiast biorąc pod uwagę zbiorowisko Car.acu (szuwar turzycy błotnej) występujący w takim samym stopniu w kręgach dla Car.Aln i Fra.Aln jak zbiorowisko Sal.cin. W `podzial` zostało przydzielone do skupienia numer 5, czyli z medoidem Car.Aln, więc poprawnie. Jednak dla `podzial_bin` zostało zakwalifikowane do 2, czyli skupienia reprezentowanego przez Que.Ulm. Oddaje to brak pozytywnego wyniku w odniesieniu do metody intuicyjno-eksperckiej. Przedstawione przykłady zostały zaprezentowane w tabelach 3.7, 3.8 oraz 3.9, w których kolumny odpowiadają skupieniom reprezentowanym przez naturalne zbiorowiska leśne, natomiast wiersze zbiorowiskom zastępczym.

	Gal.Car	Que.Ulm	Sil.Fes	Fra.Aln	Car.Aln
Sal.cin				∘	•
Car.acu				∘	•
...					

Tablica 3.7: Fragment tabeli intuicyjno-eksperckiej.

Zgodne z wiedzą ekspercką dopasowanie zostało zaznaczone znakiem $\surd$, niezgodne $\times$.

	Gal.Car	Que.Ulm	Sil.Fes	Fra.Aln	Car.Aln
Sal.cin					$\surd$
Car.acu					$\surd$
...					

Tablica 3.8: Fragment porównanych wyników `podzial` z tabelą intuicyjno-ekspercką.

	Gal.Car	Que.Ulm	Sil.Fes	Fra.Aln	Car.Aln
Sal.cin					$\surd$
Car.acu		$\times$			
...					

Tablica 3.9: Fragment porównanych wyników `podzial_bin` z tabelą intuicyjno-ekspercką.

Okazuje się, że w przypadku zbiorowisk możliwie diagnostycznych dla dynamicznych kręgów zbiorowisk (pierwsza część tabeli intuicyjno-eksperckiej) wszystkie z analizowanych dla `podzial` powtórzyły się. Dla `podzial_bin` było to 86%. Pozostałe zbiorowiska (druga część tabeli) mają szeroką skalę występowania i tym samym słabą siłę przywiązania do konkretnego kręgu, dlatego słabsze wyniki porównania (81% dla `podzial` i 83% dla `podzial_bin`) nie wpływają w dużym stopniu na pozytywną ocenę grupowań z pierwszej części tabeli. Negatywne przyporządkowania może wynikać zarówno z tego, że jeszcze nikt nie opisał danego zbiorowiska jako obecne w danym kręgu, jak i z niepoprawnego wyniku funkcji `isopam()`.

3.5. Poprawność podziału

Do sprawdzenia poprawności otrzymanych wyników podziału na skupienia zostanie wykorzystana funkcja `silhouette()` omówiona w rozdziale 2.2.2. Jej głównym argumentem musi być obiekt odpowiedniej kategorii, jednak zarówno `podzial` jak i `podzial_bin` są to obiekty klasy `isopam`. Oznacza to, że po pierwsze atrybut przechowujący informację o podziale nie jest typu `$clustering`, a po drugie wśród jej wartości nie ma macierzy odmienności powstałej po zastosowaniu Isomap. Jednak funkcja `isopam()` najpierw korzysta z `isomap()`, a ostatecznego podziału dokonuje za pomocą `pam()`, dla której wartości można już obliczyć sylwetkę. W Dodatku B została przedstawiona dokonana zmiana w oryginalnym kodzie funkcji `isopam()`, która pozwoliła na wyznaczenie $s(\cdot)$. Polega ona na dodatkowym wywołaniu funkcji `silhouette()` w każdym miejscu, w którym została wcześniej wywołana funkcja `pam()` oraz zapisaniu sylwetki podziału wśród wartości funkcji `isopam()`.

Wykresy 3.1 i 3.2 przedstawiają otrzymane sylwetki podziałów dla wyników `podzial` i `podzial_bin`. Zostały one otrzymane po wprowadzeniu następujących deklaracji

```
plot(podzial$sil, col=c("maroon1","chartreuse4", "lightseagreen",
"darkorchid2", "gold1"), border=TRUE, main="", do.n.k=FALSE)
```

dla podziału względem wartości systematycznych grup gatunków oraz

```
plot(podzial_bin$sil, col=c("maroon1","chartreuse4", "lightseagreen",
"darkorchid2", "gold1"), border=TRUE, main="", do.n.k=FALSE)
```

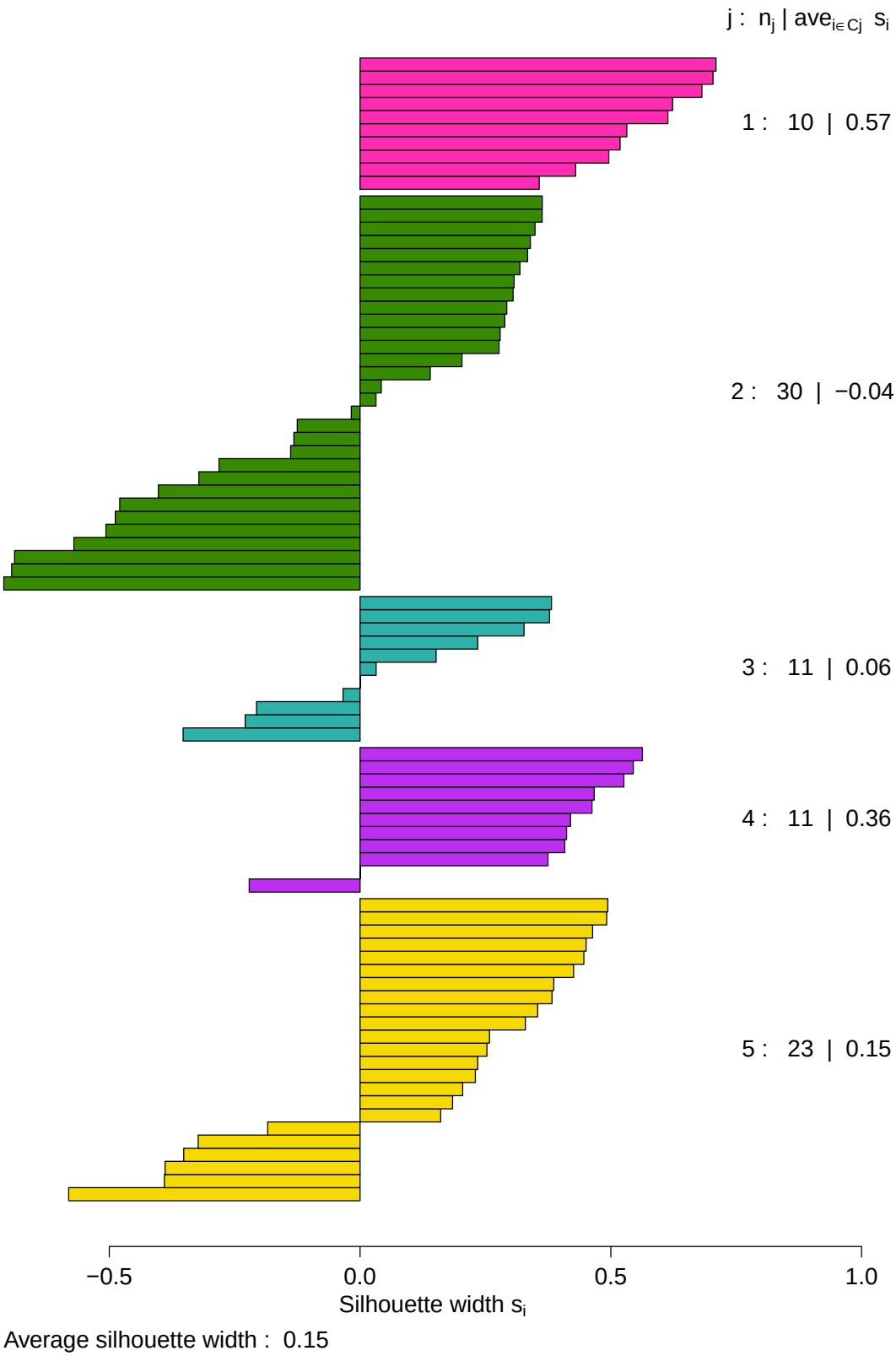
dla danych binarnych.

Przed analizą wykresów należy wziąć pod uwagę w jaki sposób wykonano podział. W PAM została całkowicie pominięta faza zamiany, która odpowiada za optymalizację grupowania. Faza budowy też została zastąpiona przez podanie ustalonych medoidów. Przez te operacje w PAM nie było możliwości obniżania sumy 1.1, a składniki tej sumy mają wpływ na wartość sylwetki. W samym Isopam można zmieniać wartości trzech argumentów:

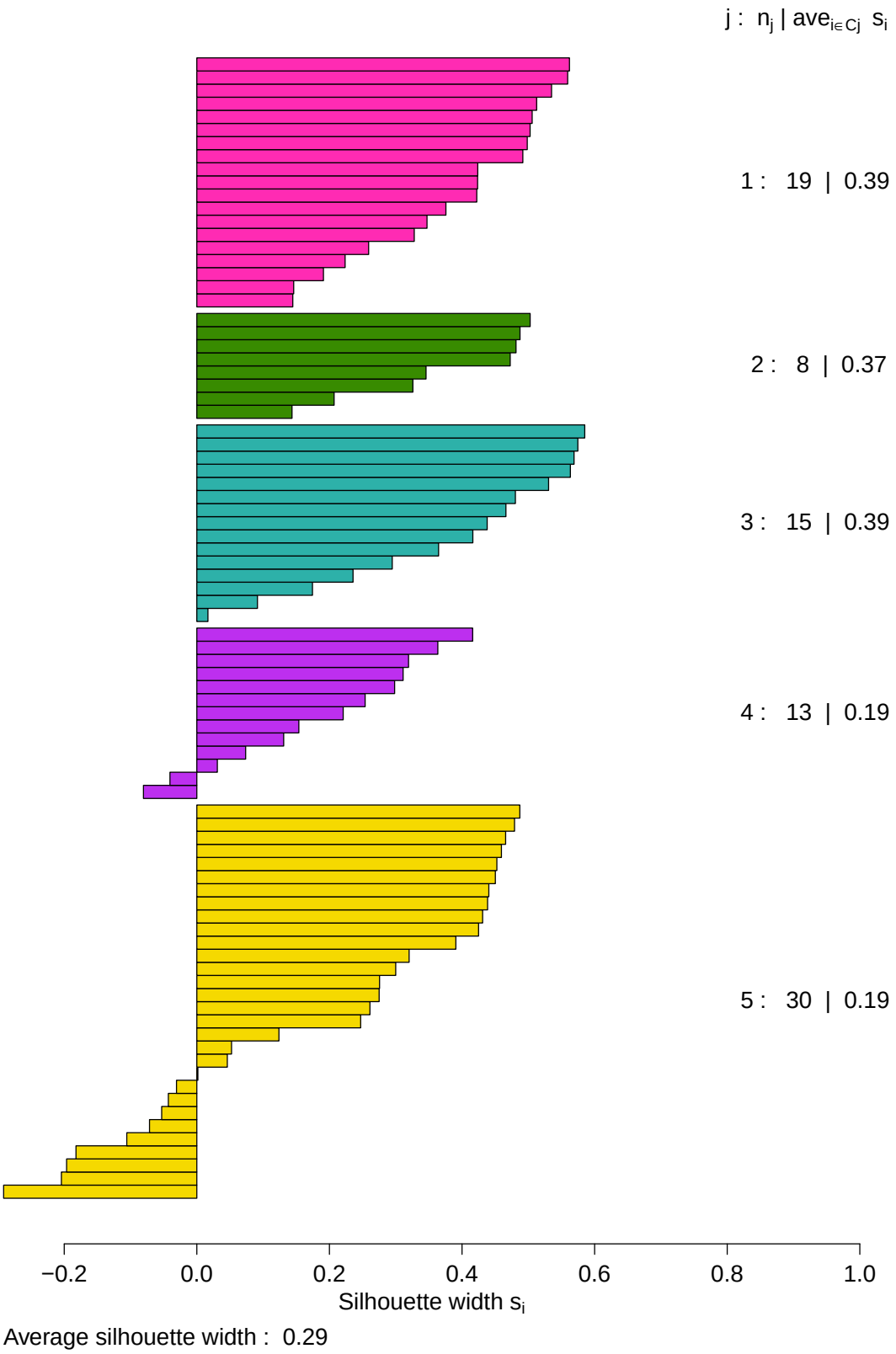
1. k (k-Isomap),
2. m (liczba skupień),
3. d (wymiar podprzestrzeni w Isomap).

W omawianym przykładzie druga z nich została ustalona i nie podlegała zamianie. Ograniczało to możliwości znalezienia podziału, który być może obniżyłby sylwetkę, ale niekoniecznie miałby sens w zakresie fitosocjologii.

W obu przypadkach (`podzial` i `podzial_bin`) występują obiekty o ujemnych wartościach $s(\cdot)$. Jedną z możliwych przyczyn takiej sytuacji może być błędna klasyfikacja, będąca efektem działania samego algorytmu, a raczej przyjętej zasady konieczności przyporządkowania obiektu do jednego skupienia. Oznacza to, że nie każdy element z wartością ujemną $s(\cdot)$ musi być „błędnie" przyporządkowany, ale jak wynika z analizy rozpatrywanych danych, obiekty te (szczególnie w przypadku zbiorowisk o szerszej skali) należy traktować z pewną rezerwą. Ma to miejsce w wynikach `podzial`. Trzy zbiorowiska suchych muraw napiaskowych z ujemną sylwetką zostały przypisane do kręgu Que.Ulm, skupienie 2. Jednak ich najbliższe sąsiedztwo (można je odczytać z wartości `podzial$sil`) stanowi już właściwy dla nich krąg Sil.Fes, skupienie 3. Ale nie można też wykluczyć możliwości błędnego przypisania zbiorowiska do skupienia, w efekcie którego jego wartość $s(\cdot)$ jest ujemna.



Rysunek 3.1: Wykres silhouette dla `podzial`.



Rysunek 3.2: Wykres silhouette dla `podzial_bin`.

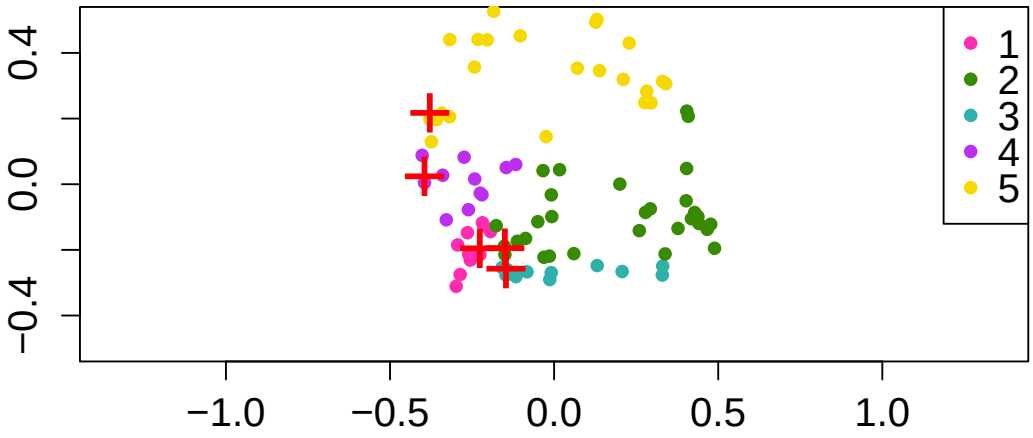
Dodatkową pomoc przy analizie wyników stanowią rysunki 3.3 oraz 3.4. Przedstawiają one przyporządkowanie zbiorowisk do skupień (każde reprezentowane innym kolorem). W pierwszym fragmencie Isopam zbiorowiska będące początkowo punktami w przestrzeni $\mathbb{R}^{31}$ odwzorowuje do przestrzeni niższego wymiaru za pomocą Isomap. Najniższy możliwy wymiar to $\mathbb{R}^2$. Wybór tej podprzestrzeni pozwala na graficzną prezentacje zrzutowanych punktów. Otrzymanych w ten sposób reprezentantów wysokowymiarowej przestrzeni (każdy punkt to inne zbiorowisko) można przedstawić na wykresie z uwzględnieniem grup, do których zostali przydzieleni po zadziałaniu PAM. Widoczna jest wówczas struktura całego skupienia np. czy jego elementy stanowią punkty znajdujące się blisko siebie, czy może punkty o szerokim zakresie występowania. Dodatkowo na wykresach znakiem + zostały wyróżnione medoidy.

Na rysunku 3.3 widać, że skupienie numer 2 jest złożone z wielu zbiorowisk o szerokim zasięgu występowania. Jednak część z nich (łącznie z medoidem) jest położona bardzo blisko skupienia numer 1 oraz 3, które są złożone z niewielkiej liczby zbiorowisk o wąskim zakresie występowania. Dlatego dla tych zbiorowisk ze skupienia numer 2 wartość $s(\cdot)$ wyszła ujemna (średnia odległość do obiektów ze skupienia sąsiedniego jest niższa niż do obiektów z własnego skupienia). Podobnie jest w przypadku skupienia numer 5. Oznacza to, że z ujemnej wartości sylwetki dla części zbiorowisk nie musi wynikać ich błędne przyporządkowanie. W przypadku skupień 3 i 4, w których tylko kilka zbiorowisk ma ujemne wartości $s(\cdot)$ (wszystkie z nich są na pograniczu skupień), po konsultacji z ekspertem, poprawność przyporządkowania można polepszyć przepisując te zbiorowiska do ich skupień sąsiednich.

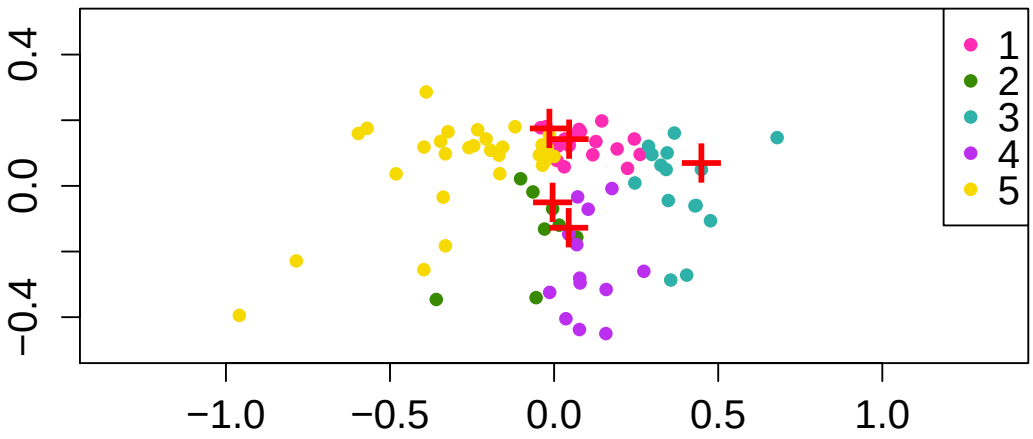
Należy jeszcze zwrócić uwagę, że w rozpatrywanym przypadku jest bardzo duże podobieństwo między medoidami ze skupień 1, 2 oraz 3 (leżą one bardzo blisko siebie), natomiast zasięg występowania wszystkich zbiorowisk jest zdecydowanie większy. Jednak ze względu na zalecenia eksperta nie ma możliwości zmiany liczby skupień, czy też elementów pełniących funkcje medoidów.

W przypadku `podzial_bin` zdecydowana większość zbiorowisk ma dodatnie wartości $s(\cdot)$. Dzięki rysunkowi 3.4 można zobaczyć jak wygląda struktura skupienia 5, w którym najwięcej obiektów miało ujemną wartość sylwetki. Jest to skupienie charakteryzujące się największym zakresem zmienności swoich komponentów. Część z nich znajduje się bardzo blisko zbiorowisk ze skupienia numer 1, które cechuje się małym zakresem zmienności. Efektem czego są ich ujemne wartości sylwetki. Natomiast w przypadku skupienia numer 4 tylko dwa zbiorowiska mają ujemną wartość $s(\cdot)$. Ich skupieniem sąsiednim jest skupienie numer 2, na granicy z którym leżą. Zgodnie z wiedzą ekspercką nie byłoby błędne przeniesienie ich do tego skupienia.

Jednak uzyskany podział (dla danych binarnych i zwykłych) nie zostanie zmieniony tzn. zbiorowiska o ujemnej wartości sylwetki nie zostaną przeniesione do swoich skupień sąsiednich, ponieważ sylwetka podziału miała pełnić tylko funkcję informacyjną o poprawności dokonanego podziału. Podobną rolę odgrywa tabela ekspercka. W odniesieniu do zbiorowisk, które nie zostały zgodnie z zakładanym przez nią zakresem zmienności zakwalifikowane, przyczyną może być luka we współczesnej wiedzy w rozpoznaniu stopnia diagnostyczności, obecności i występowania niektórych typów zbiorowisk. Przykładem tego typu jest zbiorowisko Rud.Sol. przypisane we wszystkich klasyfikacjach do kręgu Sil.Fes, choć w literaturze zbiorowisko to jest opisywane zazwyczaj jako typowo łęgowe (Sil.Fes jest kręgiem dąbrów).



Rysunek 3.3: Graficzna prezentacja podziału na skupienia dla `podzial`.



Rysunek 3.4: Graficzna prezentacja podziału na skupienia dla `podzial_bin`.

Podsumowanie

Celem niniejszej pracy był podział 85 zbiorowisk roślinnych na 5 rozłącznych grup na podstawie zgromadzonych danych w formie binarnej i zwykłej. Zamierzeniem było wyodrębnienie dynamicznych kręgów zbiorowisk roślinnych na terenie doliny Bogdanki w Poznaniu. Wyniki badań przeprowadzonych za pomocą pakietu statystycznego R pokazały, że są nieznaczne różnice między obecną wiedzą ekspercką (reprezentowaną przez Panią Monikę Zgrabczyńską), a rezultatami pracy algorytmu Isopam. Nie są to jednak na tyle duże odchylenia by podważać poprawność dokonanej analizy skupień.

Punktem referencyjnym dla wyników uzyskanych poprzez analizę skupień byłą tabela z przewidywanym składem skupień dla badanego obszaru badań opracowana na podstawie opisanych w literaturze badań i obserwacji autorów. W stosunku do zbiorowisk, które nie zostały zgodnie z nią zakwalifikowane, przyczyną może być luka we współczesnej wiedzy. Dlatego niezgodność w porównywanych wynikach nie musi oznaczać błędu po stronie klasyfikacji za pomocą Isopam.

Dodatkową pomoc w analizie wyników stanowi wskaźnik sylwetki oraz wartość łącznej średniej szerokości sylwetki, która jest na poziomie 0.29 oraz 0.15 dla danych binarnych i zwykłych odpowiednio. Wynik sylwetki na poziomie 0.2 sugeruje, że otrzymana struktura jest słaba. Z jednej strony odzwierciedla to charakter kręgów, z drugiej jednak wzrasta niepewność co do otrzymanych wyników. Jednak algorytm Isopam naturalnie (bez ustalenia argumentu `c.fix=5`) wybrałby zaledwie 3 skupienia. Wymuszając na nim wydzielenie 5 skupień dodatkowo z ustalonymi medoidami, należy oczekiwać obniżenia wyników sylwetki. Dodatkowo tabele na podstawie, których został dokonany podział stanowią zbiór danych rzeczywistych z terenu doliny Bogdanki, w żaden sposób nie modyfikowany. Zatem wyniki sylwetki na takim poziomie nie są zaskakujące.

Dzięki policzonej wartości sylwetki dla każdego zbiorowisk osobno można rozważać poprawność jego przyporządkowania do danego skupienia. Jednak ujemny wynik jakiegoś obiektu nie przesądza o jego złym zakwalifikowaniu. Po konsultacji z ekspertem część z nich można przenieść do ich skupień sąsiednich. Uzyskane podziały są najlepsze jakie mogły być przy zadanych warunkach początkowych.

Warte uwagi jest jeszcze porównanie składu skupień dla obu podziałów. W przypadku kręgu z medoidem Gal.Car w bardzo małym stopniu pokrywa się lista zbiorowisk. Jednak nie oznacza to rozbieżności, ponieważ nawet jeśli te zbiorowiska występują w innych skupieniach, to tylko w skupieniach najbardziej zbliżonych ekologicznie.

Dodatek A

Dowód tożsamości dla statystyki testowej G

Poniżej zostanie przedstawione wyprowadzenie wzoru 1.20 z 1.19. Jest on używany przez algorytm Isopam do obliczenia wartości statystyki G.

Teza:

$$2N\sum_{i=1}^{w}\sum_{j=1}^{k}p_{ij}[\ln(p_{ij})-\ln(p_{i\cdot})-\ln(p_{\cdot j})]=2[\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(o_{ij})-\sum_{j=1}^{k}o_{\cdot j}\ln(o_{\cdot j})-\sum_{i=1}^{w}o_{i\cdot}\ln(o_{i\cdot})+N\ln(N)]$$

Dowód.

$$2N\sum_{i=1}^{w}\sum_{j=1}^{k}p_{ij}[\ln(p_{ij})-\ln(p_{i\cdot})-\ln(p_{\cdot j})]=2[N\sum_{i=1}^{w}\sum_{j=1}^{k}p_{ij}\ln(p_{ij})-N\sum_{i=1}^{w}\sum_{j=1}^{k}p_{ij}\ln(p_{i\cdot})$$
$$-N\sum_{i=1}^{w}\sum_{j=1}^{k}p_{ij}\ln(p_{\cdot j})]=2[N\sum_{i=1}^{w}\sum_{j=1}^{k}\frac{o_{ij}}{N}(\ln(o_{ij})-\ln(N))-N\sum_{i=1}^{w}\sum_{j=1}^{k}\frac{o_{ij}}{N}(\ln(o_{i\cdot})-\ln(N))$$
$$-N\sum_{i=1}^{w}\sum_{j=1}^{k}\frac{o_{ij}}{N}(\ln(o_{\cdot j})-\ln(N))]=2[\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(o_{ij})-\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(N)-\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(o_{i\cdot})$$
$$+\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(N)-\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(o_{\cdot j})+\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(N)]=2[\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(o_{ij})-\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(o_{i\cdot})$$
$$-\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(o_{\cdot j})+N\ln(N)]=2[\sum_{i=1}^{w}\sum_{j=1}^{k}o_{ij}\ln(o_{ij})-\sum_{j=1}^{k}o_{\cdot j}\ln(o_{\cdot j})-\sum_{i=1}^{w}o_{i\cdot}\ln(o_{i\cdot})+N\ln(N)]$$

■

Dodatek B

Kod zmienionej funkcji `isopam()`

W związku z potrzebą podziału zbiorowisk roślinnych wokół ustalonych medoidów funkcja `isopam()` została zmieniona w takie sposób, by była możliwość ich deklaracji. Nowy argument `medoids` jest wykorzystywany w funkcji `pam()` (będącej fragmentem funkcji `isopam()`), która dokonuje podziału na skupienia. Posłużyło do tego zadeklarowanie jego wśród argumentów funkcji `isopam()`, a także nowej zmiennej `m`, której przypisano wartość `medoids`. Jest ona później wykorzystana we wszystkich miejscach, w których funkcja `pam()` była wywoływana. Druga zmiana polega na rezygnacji z fazy zamiany podczas działania algorytmu PAM. Jest wykonana poprzez deklarację `do.swap=FALSE` wśród argumentów `pam()`. Kolejnym dodanym fragmentem jest funkcja `silhouette()`. Po wykonaniu podziału za pomocą `pam()` następuje obliczenie sylwetki podziału.

Ze względu na długość całego kodu funkcji `isopam()` zostaną przedstawione jedynie zmienione jej fragmenty. W nawiasach zostały podane numery linii, w których wystąpiły zmiany, natomiast *kursywą* zostały oznaczone dodane fragmenty.

```
(1) function (dat, c.fix=FALSE, c.opt=TRUE, c.max=6, l.max=FALSE,
(2) stopat=c(1,7), sieve=TRUE, Gs=3.5, ind=NULL, distance="bray", medoids=NULL,
(3) k.max=100, d.max=7, ..., juice=FALSE)
(4) m=medoids
(118) cl.iso <- pam(isodiss, k=e, medoids=m, diss=TRUE, do.swap=FALSE)
(119) s<-silhouette(cl.iso)
(263) cl.iso <- pam(d.iso, k=mc, medoids=m, diss=TRUE, do.swap=FALSE)
(264) s<-silhouette(cl.iso)
(317) out <- list(medoids=MDS, clusters=CLS, sizes=CLI,
(318) is.ok=fine, k.min=k.min, k.max=k.max,
(319) k=mk, d=md, noi=noi, ivx=ivx, ivi=ivi, sil=s)
(322) out <- list(medoids=NULL, clusters=NULL, sizes=NULL,
(323) is.ok=FALSE, k.min=NULL, k.max=NULL, k=NULL,
(324) d=NULL, noi=NULL, ivx=NULL, ivi=NULL, sil=NULL)
(603) OUT <- list(call=sys.call(), distance=distance,
(604) flat=ctb.flat, hier=NULL, medoids=med, analytics=summ, sil=output$sil,
(605) dendro=NULL, dat=dat)
(607) OUT <- list(call=sys.call(), distance=distance,
(608) flat=ctb.flat, hier=ctb, medoids=med, analytics=summ, sil=NULL,
(609) dendro=dendro, dat=dat)
```

Po konsultacji z autorem kodu funkcji prof. dr. Sebastianem Schmidtlein w najbliższych

miesiącach pojawi się nowa wersja `isopam()` dysponująca możliwością wyboru medoidów. Wydłużany czas oczekiwania jest spowodowany pracą nad zmianami dotyczącymi przyspieszenia działania funkcji, które będą równocześnie zaproponowane.

Bibliografia

[1] Blanchet F., Kindt R., Legendre P., Minchin P., O'Hara R., Oksanen J., Simpson G., Solymos P., Stevens M., Wagner H., (2013), *Package „vegan"*, http://vegan.r-forge.r-project.org/, dostęp 31.05.2013.

[2] Buchta C., Meyer D., (2013), *Package „proxy"*, http://cran.r-project.org/web/packages/proxy/proxy.pdf, dostęp 31.05.2013.

[3] Cayton L., (2005), *Algorithms for manifold learning.*

[4] Coe R., Kindt R., (2005), *Tree diversity analysis*, World Agroforestry Centre, Nairobi, Kenya.

[5] Ćwik J., Koronacki J., (2008) *Statystyczne systemy uczące się*, Akademicka Oficyna Wydawnicza EXIT, Warszawa.

[6] Encyklopedia Leśna, http://www.encyklopedialesna.pl/, dostęp 31.05.2013.

[7] Faude U., Feilhauer H., Schmidtlein S., Tichy L., (2010), *A brute-force approach to vegetation classification*, Journal of Vegetation Science 21, 1162—1171.

[8] Hubert M., Rousseeuw P., Struyf A., (1997), *Clustering in an Object-Oriented Environment*, Journal of Statistical Software Vol. 1, Issue 4, 6–11.

[9] Koźniewski T., (2006), *Wykłady z algebry liniowej II*, Uniwersytet Warszawski, Warszawa.

[10] Langford J., de Silva V., Tenenbaum J., (2000), *A Global Geometric Framework for Nonlinear Dimensionality Reduction*, Science 22, 2319–2323.

[11] Maechler M., (2012), *Package „cluster"*, http://cran.r-project.org/web/packages/cluster/cluster.pdf, dostęp 31.05.2013.

[12] Maimon O., Rokach L., (2010), *Data Mining and Knowledge Discovery Handbook*, Springer, New York London.

[13] Mazur D. (2005), *Metody grupowania i ich implementacja do eksploracji danych postaci symbolicznej.*

[14] McDonald J., (2012), *G-test of independence*, http://udel.edu/∼mcdonald/statgtestind.html, dostęp 31.05.2013.

[15] R Development Core Team, (2013), *Package „base"*, http://stat.ethz.ch/R-manual/R-patched/library/base/html/00Index.html, dostęp 31.05.2013.

[16] R Development Core Team, (2013), *Package „grDevices”*, http://stat.ethz.ch/R-manual/R-devel/library/grDevices/html/00Index.html, dostęp 31.05.2013.

[17] R Development Core Team, (2013) , *Package „stats”*, http://stat.ethz.ch/R-manual/R-patched/library/stats/html/00Index.html, dostęp 31.05.2013.

[18] Rousseeuw P., (1987), *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*, Journal of Computational and Applied Mathematics 20, 53–65.

[19] Schmidtlein S., (2013), *Package „isopam”*, http://www.geographie.uni-bonn.de/biogeo/resources/code/isopam, dostęp 31.05.2013.

[20] Simovici D., (2011), *The PAM Clustering Algorithm*.

[21] Słownik Ekologiczny, http://slownik.ekologia.pl/, dostęp 31.05.2013.

[22] Zgrabczyńska M., (2012), *Problematyka definicji i możliwości zastosowania dynamicznych kręgów zbiorowisk w diagnozie potencjalnej roślinności naturalnej - przegląd literatury*, Dokonania naukowe doktorantów - nauki przyrodnicze, w druku.