PRACE NAUKOWE AKADEMII EKONOMICZNEJ WE WROCŁAWIU
Nr 1126 2006
TAKSONOMIA 13
Klasyfikacja i analiza danych – teoria i zastosowania

Marek Walesiak, Andrzej Dudek
Akademia Ekonomiczna we Wrocławiu

SYMULACYJNA OPTYMALIZACJA WYBORU PROCEDURY KLASYFIKACYJNEJ DLA DANEGO TYPU DANYCH – OPROGRAMOWANIE KOMPUTEROWE I WYNIKI BADAŃ

1. Wstęp

W literaturze przedmiotu w typowej procedurze klasyfikacyjnej wyodrębnia się osiem etapów (por. np. [Milligan 1996, s. 342-343; Walesiak 2005]): 1) wybór obiektów do klasyfikacji, 2) wybór zmiennych charakteryzujących obiekty, 3) wybór formuły normalizacji wartości zmiennych, 4) wybór miary odległości[1], 5) wybór metody klasyfikacji, 6) ustalenie liczby klas, 7) walidację wyników klasyfikacji, 8) opis (interpretację) i profilowanie klas. Do newralgicznych zalicza się etapy dotyczące wyboru: formuły normalizacji wartości zmiennych, miary odległości, metody klasyfikacji i ustalenia liczby klas, które mają w znacznej mierze charakter arbitralny.

W artykule krótko scharakteryzowano dziewięć ścieżek w symulacyjnej optymalizacji wyboru procedury klasyfikacyjnej dla danego typu danych, (zaproponowanych w pracy Walesiaka i Dudka [2005]). Następnie zaprezentowano podstawowe funkcje programu komputerowego clusterSim, służącego realizacji wyodrębnionych ścieżek, oraz wybrane wyniki obliczeń symulacyjnych przy wzrastającej liczbie obiektów i zmiennych w macierzy danych. Wszystkie procedury opracowano w języku R oraz pomocniczo w języku C++.

2. Charakterystyka ścieżek w symulacyjnej optymalizacji wyboru procedury klasyfikacyjnej dla danego typu danych

Punktem wyjścia w symulacyjnej optymalizacji wyboru procedury klasyfikacyjnej dla danego typu danych jest macierz danych. Przy opracowywaniu poszczególnych ścieżek uwzględniono następujące elementy:

[1] W metodach bazujących na macierzy danych (np. w metodzie *k*-średnich) etap ten nie występuje.

a) typ skali pomiaru zmiennych w macierzy danych – z tego tytułu wyróżniono dziewięć ścieżek w procedurze symulacyjnej,

b) typ formuły normalizacyjnej dla zmiennych mierzonych na skali przedziałowej i (lub) ilorazowej – uwzględniono 11 formuł normalizacyjnych (por. [Gatnar, Walesiak i in. 2004, s. 35-38]),

c) miary odległości właściwe dla poszczególnych typów skal pomiaru zmiennych,

d) typ metody klasyfikacji – rozważania ograniczono do najczęściej wykorzystywanych w praktyce metod klasyfikacji (siedem hierarchicznych metod aglomeracyjnych i metoda *k-medoids*, bazujące na macierzy odległości, oraz metoda *k*-średnich, bazująca na macierzy danych),

e) miernik oceny jakości klasyfikacji – do oceny zastosowano pięć indeksów pozwalających na wyznaczenie optymalnej liczby klas.

Do oceny wyników symulacji, w zaproponowanej optymalizacji wyboru procedury klasyfikacyjnej dla danego typu danych, wprowadzono trzy najlepsze mierniki globalne z eksperymentu Milligana i Cooper [1985]: Calińskiego i Harabasza [1974], Bakera i Huberta [Hubert 1974; Baker, Hubert 1975], Huberta i Levine [1976], oraz dwa indeksy wykorzystywane często w literaturze w testach porównawczych[2]: Silhouette ([Rousseeuw 1987; Kaufman, Rousseeuw 1990]), Krzanowskiego i Lai [1985].

Liczba rozpatrywanych wariantów procedury klasyfikacyjnej zależy od liczby formuł normalizacyjnych, liczby typów miar odległości i liczby metod klasyfikacji. Szczegółowe charakterystyki ścieżek zaprezentowano w tab. 1.

3. Charakterystyka oprogramowania komputerowego clusterSim

Program służący optymalizacji wyboru procedury klasyfikacyjnej dla danego typu danych został napisany w języku R. Niektóre fragmenty w celu przyspieszenia obliczeń zostały napisane w języku C i skompilowane kompilatorem gcc 3.2.3 w środowisku MINGW w systemie Windows. Wszystkie funkcje użyte do obliczeń połączone zostały w pakiet clusterSim. Jego kod źródłowy i skompilowaną wersję działającą w systemie Microsoft Windows przewiduje się umieścić w formie pakietu w repozytorium pakietów systemu R – CRAN.

Pakiet składa się z funkcji podstawowej `cluster.Sim` oraz dziesięciu funkcji pomocniczych. Funkcja `cluster.Sim` ma następującą składnię:

```
cluster.Sim (x, p, minClusterNo, maxClusterNo, icq="S",
             outputHtml="", outputCsv="")
```

Parametry:

`x` macierz danych

[2] Por. [Tibshirani, Walther i Hastie 2001; Dudoit, Fridlyand 2002; Sugar, James 2003; Mufti, Bertrand, El Moubarki 2005].

Tabela 1. Ścieżki w symulacyjnej optymalizacji wyboru procedury klasyfikacyjnej dla danego typu danych

Nr	Etapy typowej procedury klasyfikacyjnej	Numer ścieżki w procedurze symulacyjnej								
		1	2	3	4	5	6	7	8	9
I	Wybór obiektów i zmiennych	macierz danych [x_{ij}]								
II	Skala pomiaru zmiennych	ilorazowa	ilorazowa	przedziałowa[1]	porządkowa	nominalna wielostanowa	binarna	ilorazowa	przedziałowa[1]	ilorazowa	przedziałowa[1]
	Wybór formuły normalizacji[2]	n6 – n11	n1 – n5	n1 – n5	NA	NA		bez normalizacji		n6-n11/ n1-n5	n1-n5
	Skala pomiaru zmiennych po normalizacji	ilorazowa	przedziałowa	przedziałowa	porządkowa	nominalna wielostanowa	binarna	ilorazowa	przedziałowa[1]	ilorazowa/ przedziałowa	przedziałowa
III	Wybór miary odległości[3]	d1 – d7	d1 – d5	d1 – d5	d8	d9	b1 – b10	d1 – d7	d1 – d5	NA	
IV	Wybór metody klasyfikacji	1. Pojedynczego połączenia 3. Średniej klasowej 5. *k*-medoids (PAM) 7. Centroidalna[4] 2. Kompletnego połączenia 4. Ważonej średniej klasowej 6. Warda[4] 8. Medianowa[4]								*k*-średnich	
V	Liczba możliwości	[(6 x 7 x 5)+ (6 x 1 x 3)] + [(5 x 5 x 5) +(5 x 1 x 3)] = 368	(5 x 5 x 5) + (5 x 1 x 3) = 140	1 x 5 = 5	1 x 5 = 5	10 x 5 = 50	(7 x 5) + (1 x 3) = 38	(5 x 5) + (1 x 3) = 28	11	5
	Miernik jakości klasyfikacji	1. Caliński & Harabasz (G1) 2. Baker & Hubert (G2) 3. Hubert & Levine (G3) 4. Silhouette (S) 5. Krzanowski & Lai (KL)		1. NA 2. G2 3. G3 4. S 5. NA			1. G1 2. G2 3. G3 4. S 5. KL		1. G1 2. NA 3. NA 4. NA 5. KL	

[1] Lub ilorazowa i przedziałowa.
[2] n1 (n2) – standaryzacja klasyczna (Webera), n3 – unitaryzacja, n4 – unitaryzacja zerowana, n5 – normalizacja w przedziale [–1; 1], n6-n11 – przekształcenia ilorazowe.
[3] d1 – miejska, d2 – euklidesowa, d3 – Czebyszewa, d4 – kwadrat euklidesowej, d5 – GDM1, d6 – Canberra, d7 – Braya-Curtisa, d8 – GDM2, d9 – Sokala i Michenera dla zmiennych nominalnych; odległości dla zmiennych binarnych (dostępne w procedurze dist.binary): b1 = Jaccard; b2 = Sokal & Michener; b3 = Sokal & Sneath (1); b4 = Rogers & Tanimoto; b5 = Czekanowski; b6 = Gower & Legendre (1); b7 = Ochiai; b8 = Sokal & Sneath (2); b9 = Phi of Pearson; b10 = Gower & Legendre (2).
[4] Metody klasyfikacji przyjmujące założenie, że odległości między obiektami zostały wyznaczone za pomocą kwadratu odległości euklidesowej, tylko bowiem w tym przypadku metody te mają interpretację geometryczną zgodną z ich nazwami.
NA – nie stosuje się.

Źródło: opracowanie własne (opisy metod znajdują się m.in. w pracach: [Gordon 1999; Everitt, Landau, Leese 2001; Gatnar, Walesiak 2004]).

`p`	ścieżka symulacji (zgodnie z tab. 1)
`minClusterNo`	minimalna liczba klas w symulacji
`maxClustersNo`	maksymalna liczba klas w symulacji
`icq`	indeks jakości podziału: S, G1, G2, G3, KL (tab. 1)
`outputHtml`	parametr opcjonalny – nazwa (bez rozszerzenia) pliku HTML z wynikami symulacji
`outputCsv`	parametr opcjonalny – nazwa (bez rozszerzenia) pliku tekstowego typu CSV z wynikami symulacji

Funkcja `cluster.Sim` przed rozpoczęciem symulacji sprawdza, czy są zachowane następujące warunki:

- indeks G1 Calińskiego i Harabasza oraz KL Krzanowskiego i Lai może być użyty tylko dla danych mierzonych na skali ilorazowej lub przedziałowej, a więc można stosować je tylko dla ścieżek 1, 2, 6, 7, 8 i 9,
- dla ścieżek 8 i 9 program zezwala jedynie na użycie indeksów G1 i KL,
- minimalna liczba klas jest większa lub równa 2, a maksymalna liczba klas jest mniejsza lub równa liczbie obiektów minus jeden (minus dwa dla indeksu G3 oraz minus trzy dla indeksu KL),
- program sprawdza dla ścieżki numer 5, czy wszystkie dane są danymi binarnymi, a dla ścieżek 1, 6 i 8, czy dane są mierzone na skali ilorazowej.

Optymalny wynik symulacji otrzymany przy użyciu funkcji `cluster.Sim` jest obiektem złożonym o następującej strukturze:

- path – numer ścieżki z tab. 1,
- result – optymalna wartość indeksu jakości podziału,
- normalization – metoda normalizacji, dla której uzyskana została optymalna wartość indeksu jakości podziału,
- distance – miara odległości, dla której uzyskana została optymalna wartość indeksu jakości podziału,
- method – metoda klasyfikacji, dla której uzyskana została optymalna wartość indeksu jakości podziału,
- classes – liczba klas, dla której uzyskana została optymalna wartość indeksu jakości podziału,
- time – całkowity czas działania funkcji.

Wszystkie warianty procedury klasyfikacyjnej (ustalone na podstawie liczby formuł normalizacyjnych, typów miar odległości oraz metod klasyfikacji) wraz z wartościami indeksu jakości podziału można zapisać w plikach w formacie tekstowym typu CSV lub w formacie HTML. W formacie pliku CSV oraz HTML w wierszach znajdują się poszczególne warianty procedury klasyfikacji badane w symulacji, a w kolumnach ich charakterystyki:

- No. – numer procedury klasyfikacyjnej,
- No. of clusters – liczba klas,
- Normalization formula – typ formuły normalizacyjnej,

- Distance measure – typ miary odległości,
- Clustering method – typ metody klasyfikacji,
- Silhouette (lub nazwa odpowiedniego indeksu) – wartość indeksu jakości podziału,
- Rank – pozycja *i*-tej procedury klasyfikacji według wartości indeksu jakości podziału (1 oznacza pozycję najlepszą według zadanego indeksu jakości podziału).

Wyniki symulacji zapisane w formacie HTML przedstawiane są w formie nieuporządkowanej i uporządkowanej tablicy rezultatów dla każdej klasyfikacji:
RESULTS OF CLASSIFICATIONS
PATH = 1 (Ratio data)
INDEX = Silhouette
NO. OF CLUSTERS = <2; 15>

Unsorted results (fragment tablicy)

	No.	No. of clusters	Normalization formula	Distance measure	Clustering method	Silhouette	Rank
1	1	2	n1	Manhattan	single	0.0237886519943069	4537
2	2	3	n1	Manhattan	single	–0.0192343072750229	4791
3	3	4	n1	Manhattan	single	–0.0473224230840262	4897
4	4	5	n1	Manhattan	single	–0.0538015977854346	4924
5	5	6	n1	Manhattan	single	–0.0481570597359684	4902

Sorted results (fragment tablicy)

	Rank	No.	No. of clusters	Normalization formula	Distance measure	Clustering method	Silhouette
1	1	3980	5	n9	GDM1	average	0.536841338004683
2	2	4512	5	n10	GDM1	average	0.536841338004682
3	3	4010	7	n9	GDM1	pam	0.528232736071232
4	4	4542	7	n10	GDM1	pam	0.528232736071231
5	5	3982	7	n9	GDM1	average	0.527746457551768

Na końcu pliku znajduje się informacja o całkowitym czasie wykonywania funkcji (*Estimated calculation time*: np. 6.15).

Dane z formatu CSV mogą być bezpośrednio przeniesione do arkusza kalkulacyjnego MS Excel lub do popularnych pakietów statystycznych. Dane w formacie HTML mogą zostać przeniesione do popularnych edytorów tekstu.

Pakiet clusterSim zawiera dziesięć funkcji pomocniczych niezbędnych do symulacyjnej optymalizacji wyboru procedury klasyfikacyjnej dla danego typu danych[3]:

[3] Oprócz funkcji `cluster.Description`.

`data.Normalization (x,type="n0")` – funkcja dokonująca normalizacji danych według jednej z 11 formuł normalizacyjnych oznaczonych `n1-n11` (`x` – macierz danych, `type` – typ formuły normalizacyjnej z tab. 1). Opcja `n0` oznacza pozostawienie danych bez normalizacji.

`dist.BC (x)` – funkcja obliczająca macierz odległości według formuły Braya–Curtisa dla zmiennych ilorazowych.

`dist.GDM (x,method="GDM1")` – funkcja obliczająca macierz odległości według miary `GDM` (zob. Walesiak [2002]), gdzie `GDM1` oznacza miarę odległości GDM dla danych metrycznych, a `GDM2` dla danych porządkowych.

`dist.SM(x)` – funkcja obliczająca macierz odległości według miary Sokala–Michenera dla zmiennych nominalnych.

`index.G1 (x,cl)` – funkcja obliczająca indeks Calińskiego–Harabasza (pseudostatystykę *F*) dla macierzy danych `x` i wektora liczb całkowitych `cl` informujących o przynależności poszczególnych obiektów do klas.

`index.G2(d,cl)` – funkcja obliczająca indeks Bakera–Huberta dla macierzy odległości `d` i ustalonego podziału zbioru obiektów na klasy `cl`.

`index.G3(d,cl)` – funkcja obliczająca indeks Huberta–Levine dla macierzy odległości `d` i ustalonego podziału zbioru obiektów na klasy `cl`.

`index.S(d,cl)` – funkcja obliczająca indeks Silhouette Rousseeuwa dla macierzy odległości `d` i ustalonego podziału zbioru obiektów na klasy `cl`.

`index.KL (x,clall)` – funkcja obliczająca indeks Krzanowskiego–Lai dla macierzy danych `x` oraz trzech wektorów liczb całkowitych informujących o przynależności poszczególnych obiektów do klas w podziale na $u-1$, u i $u+1$ klas (`clall`).

`cluster.Description(x, cl)` – funkcja obliczająca osobno dla każdej klasy i zmiennej z ustalonego podziału zbioru obiektów na klasy `cl` następujące statystyki opisowe: średnia arytmetyczna, odchylenie standardowe, mediana, medianowe odchylenie bezwzględne, dominanta (dla zmiennych nominalnych i porządkowych; jeśli występuje więcej wartości o maksymalnej częstości występowania zwracana jest wartość „N.A.”).

W żadnej z funkcji pakietu clusterSim nie występują ograniczenia co do wielkości zarówno macierzy danych, macierzy odległości, jak i liczby klas, na które może być podzielony zbiór. Każda z funkcji może być wykorzystana w innych procedurach programu R po uruchomieniu pakietu clusterSim.

4. Wybrane wyniki obliczeń symulacyjnych w programie clusterSim

Obliczenia symulacyjne zostały wykonane w programie R wersji 2.1.1 z wykorzystaniem pakietu clusterSim oraz pakietów ade4, cluster i R2HTML. Ograniczo-

no je do ścieżki pierwszej, obejmującej zmienne mierzone na skali ilorazowej, z uwagi na największą liczbę rozpatrywanych możliwości. Ścieżka ta obejmuje 368 wariantów klasyfikacji przy podziale na ustaloną liczbę klas. W klasyfikacji hierarchicznej rozważa się podział na $n-1$ klas. Ponieważ badacz zwykle w praktyce dysponuje wiedzą aprioryczną o przedziale, w jakim powinna znaleźć się wyodrębniona liczba klas, w programie istnieje możliwość arbitralnego podania minimalnej i maksymalnej liczby klas. Zatem liczba rozpatrywanych klasyfikacji LK wynosi:

$$LK = (\text{maxClusterNo} - \text{minClusterNo} + 1) \times LW_p ,$$

gdzie: maxClusterNo (minClusterNo) – maksymalna (minimalna) liczba klas w procedurze symulacyjnej,

LW_p – liczba rozpatrywanych wariantów klasyfikacji dla *p*-tej ścieżki.

Przykładowa składnia poleceń z wykorzystaniem funkcji `cluster.Sim` jest następująca:

```
> library(clusterSim)
> x <- read.csv2(C:/Dane_ratio_75_5.csv,
  header=TRUE, strip.white = TRUE, row.names=1)
> wynikratio75_5<-cluster.Sim(x,1,2,15,"S","wynik_ratio75_5")
> print(wynikratio75_5)
```

Nie ma bezpośrednich procedur wczytywania zawartości skoroszytów MS Excel do programu R. W przypadku tego typu danych najwygodniej używać pośredniego formatu CSV. Zapisanie danych w Excelu wymaga wybrania polecenia *Plik* | *Zapisz jako...* i wybrania odpowiedniego formatu. Do wczytania tych danych w programie R służy instrukcja `read.csv2` (ponieważ dla języka polskiego symbolem ułamka dziesiętnego jest przecinek). Podane parametry instrukcji `read.csv2` oznaczają, że dane w pliku CSV zawierają nagłówek (`header=TRUE`) oraz w pierwszej kolumnie znajdują się nazwy obiektów (`row.names=1`). Zawsze należy podać pełną ścieżkę dostępu do pliku z danymi z separatorem nazwy oznaczonym „/”. Wczytywane dane zazwyczaj zawierają macierz danych, choć mogą zawierać np. obliczoną w zewnętrznym programie macierz odległości.

Dla ścieżki pierwszej przeprowadzono obliczenia symulacyjne przy założeniach:

- liczba obiektów w macierzy danych wynosiła odpowiednio 50, 75, 100, 150, 200 i 400,
- liczba zmiennych w macierzy danych wynosiła odpowiednio 5, 10 i 30,
- założono minimalną oraz maksymalną liczbę klas odpowiednio równą 2 i 15.

Dla tych założeń oraz przyjętego miernika oceny jakości klasyfikacji w tab. 2 przedstawiono przybliżone czasy obliczeń.

Tabela 2. Przybliżone czasy obliczeń w minutach dla przyjętych założeń oraz danego miernika oceny jakości klasyfikacji*

Liczba obiektów × liczba zmiennych	Miernik oceny jakości klasyfikacji				
	S	G1	G2	G3	KL
50 × 5	3,72	4,12	2,13	0,52	10,83
50 × 10	3,45	6,63	2,30	0,50	18,08
75 × 5	6,15	4,37	4,00	0,90	11,68
75 × 10	6,15	7,02	3,93	0,87	19,55
100 × 5	8,40	4,63	6,63	1,22	12,32
100 × 10	9,23	7,70	7,25	1,47	21,32
150 × 5	16,82	5,95	22,83	2,53	15,10
150 × 10	16,23	9,82	26,05	3,12	25,97
200 × 5	21,68	7,30	62,83	4,72	16,80
200 × 10	22,28	11,53	68,30	5,90	27,97
400 × 30	97,58	73,35	**	49,93	149,68

* Obliczenia wykonano na komputerze z procesorem Intel Pentium IV CPU 1,5 GHz i pamięcią RAM 512 MB.
** Zbyt długi czas obliczeń z praktycznego punktu widzenia.

Źródło: obliczenia własne.

5. Wnioski końcowe

W typowym studium klasyfikacyjnym etapy dotyczące wyboru formuły normalizacji wartości zmiennych, miary odległości, metody klasyfikacji oraz ustalenia liczby klas mają zwykle charakter arbitralny. Niewątpliwą zaletą programu clusterSim jest obiektywizacja problemu ich wyboru. Uzyskuje się to w wyniku przeprowadzenia symulacyjnej optymalizacji wyboru procedury klasyfikacyjnej dla danego typu danych. Miernikami oceny wszystkich procedur klasyfikacyjnych badanego zbioru obiektów są globalne indeksy oceny jakości klasyfikacji, pozwalające na wyznaczenie optymalnej liczby klas.

Zaprezentowane podejście niesie ze sobą również pewne ograniczenia:

- w literaturze funkcjonuje ponad 40 mierników oceny jakości klasyfikacji. W programie cluster.Sim możliwe było uwzględnienie tylko indeksów globalnych;
- spośród indeksów globalnych w procedurze clusterSim uwzględniono pięć najważniejszych. Jednak ostateczny wybór jednego z nich pozostaje nadal arbitralny.

Literatura

Baker F.B., Hubert L.J. (1975), *Measuring the Power of Hierarchical Cluster Analysis*, „Journal of the American Statistical Association" vol. 70, nr 349, s. 31-38.

Calinski R.B., Harabasz J. (1974), *A Dendrite Method for Cluster Analysis*, „Communications in Statistics" vol. 3, s. 1-27.

Dudoit S., Fridlyand J. (2002), *A Prediction-Based Resampling Method for Estimating the Number of Clusters in A Dataset*, „Genome Biology" vol. 3, nr 7, s. 1-20.
Everitt B.S., Landau S., Leese M. (2001), *Cluster Analysis*, Edward Arnold, London.
Gatnar E., Walesiak M. (red.) (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych*, AE, Wrocław.
Gordon A.D. (1999), *Classification*, Chapman and Hall/CRC, London.
Hubert L.J. (1974), *Approximate Evaluation Technique for the Single-Link and Complete-Link Hierarchical Clustering Procedures*, „Journal of the American Statistical Association" vol. 69, nr 347, s. 698-704.
Hubert L.J., Levine J.R. (1976), *Evaluating Object Set Partitions: Free Sort Analysis and Some Generalizations*, „Journal of Verbal Learning and Verbal Behaviour" vol. 15, s. 549-570.
Kaufman L., Rousseeuw P.J. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, New York.
Krzanowski W.J., Lai Y.T. (1985), *A Criterion for Determining the Number of Groups in A Data Set Using Sum of Squares Clustering*, „Biometrics" nr 44, s. 23-34.
Milligan G.W. (1996), *Clustering Validation: Results and Implications for Applied Analyses*, [w:] P. Arabie, L.J. Hubert, G. De Soete (red.), *Clustering and Classification*, World Scientific, Singapore, s. 341-375.
Milligan G.W., Cooper M.C. (1985), *An Examination of Procedures for Determining the Number of Clusters in A Data Set*, „Psychometrika" nr 2, s. 159-179.
Mufti G.B., Bertrand P., El Moubarki L. (2005), *Determining the Number of Groups from Measures of Cluster Stability*, [w:] J. Janssen, P. Lenca (red.), *Applied Stochastic Models and Data Analysis*, ENST Bretagne, Brest, s. 404-413.
Rousseeuw P.J. (1987), *Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis*, „Journal of Computational and Applied Mathematics" nr 20, s. 53-65.
Sugar C.A., James G.H. (2003), *Finding the Number of Clusters in a Dataset: an Information-Theoretic Approach*, „Journal of the American Statistical Association" vol. 98, nr 463, s. 750-763.
Tibshirani R., Walther G., Hastie T. (2001), *Estimating the Number of Clusters in A Data Set Via the Gap Statistic*, „Journal of the Royal Statistical Society", ser. B, vol. 63, cz. 2, s. 411-423.
Walesiak M. (2002), *Uogólniona miara odległości w statystycznej analizie wielowymiarowej*, AE, Wrocław.
Walesiak M. (2005), *Rekomendacje w zakresie strategii postępowania w procesie klasyfikacji zbioru obiektów*, [w:] A. Zeliaś (red.), *Przestrzenno-czasowe modelowanie i prognozowanie zjawisk gospodarczych*, AE, Kraków, s. 185-203.

Walesiak M., Dudek A. (2005), *Symulacyjna optymalizacja wyboru procedury klasyfikacyjnej dla danego typu danych – charakterystyka problemu*, Zeszyty Naukowe Uniwersytetu Szczecińskiego (w druku).

DETERMINATION OF OPTIMAL CLUSTERING PROCEDURE FOR A DATA SET – COMPUTER PROGRAM AND EMPIRICAL RESULTS

Summary

In typical cluster analysis study eight major steps are distinguished (see Milligan [1996, 342-343]; Walesiak [2005]). Four of them represent the critical steps: decisions concerning variable normalisation formula, selection of a distance measure, selection of clustering method, determining the number of clusters.

The article presents:

a) determination of optimal clustering procedure for a data set by varying all combinations of normalization formulas, distance measures, and clustering methods. Nine paths of simulation was separated depends on variable scale of measurement in a data set;

b) clusterSim computer program written in R and C++ languages;

c) some empirical results of simulation study based on data matrix with growing number of objects and variables.