УДК 004.032.26 **doi: 10.26906/SUNZ.2019.5.045**

О. А. Кобилін, С. О. Вискребенцева, Р. В. Петрова

Харківський національний університет радіоелектроніки, Харків, Україна

ОБРОБКА ДАНИХ, ЩО МІСТЯТЬ ПРОПУСКИ В ЗАДАЧАХ КЛАСТЕРИЗАЦІЇ

Предметом досліджень є методи підготовки та обробки вхідних даних, що містять пропущені значення, для їх подальшого аналізу та кластеризації. **Метою дослідження** є розгляд існуючих методів позбавлення від пропусків у даних в задачах кластеризації та доцільність їх використання у реальних задачах. **Завдання:** аналіз переваг та недоліків кожного з методів, що направлені на відновлення даних, для визначення доцільності використання їх в задачах кластеризації та виділення оптимального методу, порівняльний аналіз методів, оцінка результативності за наслідками порівняння кластеризації відновлених даних з результатами кластеризації еталонних даних. **Методи**: FCM - для проведення безпосередньо кластеризації даних, метод видалення всіх рядків, що містять пропуски, заповнення пропусків вибірковими статистиками, заповнення пропусків з урахуванням структури зв'язків. **Результати**: ефективність застосування методів при підготовці даних для подальшої кластеризації залежить від кількості наявних пропусків в похідному наборі. Якщо таких рядків досить мало, то кожен, з розглянутих методів, може бути використаний і дати необхідні результати. Але, якщо рядків з пропусками досить багато, наприклад 30%, тоді найбільш прийнятними для використання можна назвати методи, що пов'язані з заміною значень, однак слід враховувати, що така заміна може призвести до спотворення даних, а з рештою і результатів. **Висновки. Наукова новизна –** дослідження проблеми кластеризації даних, що містять пропущені значення та розгляд методів, які дозволяють розв'язати цю задачу. Проведення експериментів та порівняння результатів кожного з методів, висновки про доцільність використання того чи іншого методу та побічні ефекти. **Практична значущість** роботи полягає у визначені можливості використання в реальних задачах, що зазвичай не є ідеальними і з великою ймовірністю міститимуть пропуски, методів обробки даних для використання їх в задачах кластеризації.

Ключові слова: кластеризація, неповні дані, обробка та аналіз даних, Data Mining, FCM, методи відновлення даних, мова програмування R.

Вступ

З швидким розвитком комп'ютерних технологій і наук задачі, що постають перед науковцями змінюються, так раніше обчислювальні машини, а, разом з ними і комп'ютерні науки, розвивалися досить повільно і основний інтерес становив розвиток саме апаратної бази: збільшення пам'яті, як оперативної так і жорстких дисків, загальна швидкість обчислень та інше. Розвиток програмного забезпечення був обмежений саме апаратними характеристиками, необхідно було розв'язувати проблеми збільшення працездатності програми та вирішувати задачі скорочення ресурсів, а саме зменшення програмного коду.

На сьогодні проблема з пам'яттю або обчислювальними ресурсами не стоїть так гостро, апаратна база стрімко розвивається і більше не становить основний інтерес. Ще однією рушійною силою є те, що комп'ютери стали все більш доступними і немалий вклад в розвиток технологій роблять пересічні користувачі. Враховуючи це, збільшується кількість інформації, що потрапляє в комп'ютер і безпосередньо в Інтернет.

Зі збільшенням об'ємів даних стали виникати нові задачі, все більший інтерес становить робота з даними і розв'язання проблем, що пов'язані з їх обробкою і подальшим аналізом. Оскільки можливості створення нового контенту мають всі бажаючи – обсяги даних непомірно зростають, а їх впорядкованість слабка. З'являється необхідність пошуку даних, в тому числі зображень, їх обробки, для маркетингових і статистичних досліджень, використанню у інших сферах, наприклад повсякденного життя пересічного користувача.

Найпопулярнішими напрямками досліджень частіше стають: Big Data, Data Mining, Machine Learning, постає питання пошуку даних їх глобальний та інтелектуальний аналіз. Серед таких актуальних завдань знаходять своє місце і поняття класифікації та кластеризації даних.

Класифікація встановлює закономірності для розбиття даних на заздалегідь визначені підмножини (класи).

Кластеризація є процес розбиття заданої вибірки об'єктів (спостережень) на підмножини (як правило, непересічні), які називаються кластерами, так, щоб кожен кластер складався з схожих об'єктів, а об'єкти різних кластерів істотно відрізнялися [1].

Задачі подібні, але основна відмінність полягає у тому, що кластеризація передбачає розбиття за умови початкової невизначеності щодо конкретних груп, вона може мати критерії щодо кількості кінцевих кластерів, але не їх зміст, тобто, можна сказати, що це навчання без вчителя.

Виділяють наступні основні завдання кластерного аналізу [2]:

- розробка типології або класифікації;
- дослідження корисних концептуальних схем групування об'єктів;
- породження гіпотез на основі дослідження даних;
- перевірка гіпотез для визначення, чи дійсно типи (групи) виділені тим чи іншим способом, присутні в наявних даних.

Отже, кластеризація становить інтерес, як спосіб попередньої обробки даних, для більш зручного подальшого аналізу. Отримавши необхідні групи, а також їх центроїди можна продовжувати роботу вже з конкретними представниками, а не з усім набором

даних, що особливо актуально в умовах безкінечно зростаючого об'єму інформації. Даний підхід дозволяє краще зрозуміти дані, шляхом використання для кожного кластеру найбільш оптимального алгоритму аналізу; провести стиснення, виділивши найбільш типових представників, за умов збитковості даних; виявлення новизни, шляхом виділення об'єктів, що не потрапили до жодного з кластерів.

Сфера застосування може бути доволі широка, використання в сегментації зображень [8], аналізу відео [9], прогнозування, аналіз текстів, оптимізація, машинне навчання, інтелектуальному аналізі даних [7]. Таким чином вивчення та використання методів кластеризації для вирішення багатьох важливих питань є досить цікавою задачею, але вона має низку своїх недоліків і проблем, що потребують вирішення.

Так, наприклад, однією з проблем кластеризації можна виділити роботу з пропусками даних. Існує безліч методів, але вони не передбачають відсутності якоїсь кількості інформації. Але, як тільки ми виходимо за рамки тестових даних і переходимо до обробки реальних - стикаємося з цією проблемою, адже в дійсності ідеальних даних не існує і всі вони містять шуми (некорисну інформацію, яка може зашкодити результату), пропуски, невідповідні формати та інше.

Таким чином, доцільним є розгляд варіантів рішення проблеми кластеризації даних з пропусками.

Кластеризація та класичні методи кластеризації

Описуючи процес кластеризації, позначимо множину об'єктів, що складається з набору атрибутів $x_i = \left\{t_1^i, t_2^i, ..., t_m^i\right\}$, як $\tau = \left\{x_i\right\}_{i=1}^n$, де t_v^i приймає значення із заданої множини T_v^i. Завдання кластеризації полягає в побудові множини $C = \left\{c_v\right\}_{v=1}^k$ і відображення $F : \tau \to C$ заданої множини об'єктів на множину кластерів.

Кластер містить записи з τ подібними (за заданим критерієм) один на одного

$$x_i \in c_v, x_j \in c_v \Rightarrow d(x_i, x_j) < \varepsilon, \qquad (1)$$

де $d(x_i, x_j)$ – міра близькості між об'єктами (відстань), а ε - максимальне значення порогу, що формує один кластер [1].

Виходячи з [3], можна сказати, що методи кластеризації поділяються на ієрархічні та неієрархічні (алгоритми розподілу) алгоритми.

В процесі ієрархічної кластеризації відбувається злиття та поділ кластерів під час побудови дерева вкладених кластерів (дендрограми). Ієрархічні методі також поділяються на:

- агломеративні, на початку алгоритму кожен елемент це окремий кластер, при подальших кроках кластери поєднуються в один. Таким чином, метод полягає в об'єднанні кластерів, зменшенні їх кількості;

- дівізивні, тобто один кластер на початку і подальше його розділення на більшу кількість кластерів.

Неієрархічні методи відрізаються тим, що потребують наявність умови зупинки і кількість кластерів. Це ітеративний процес поділу на кластери до тих пір доки не буде виконана умова зупинки. Неієрархічні методи кластерного аналізу більш придатні у випадку великої кількості спостережень.

Найбільш популярним методом неієрархічної кластеризації є метод найменших квадратів

$$\sum_{j=1}^{k}\sum_{i=1}^{n_j}\left|x_i - s_j\right|^2 \to \min \qquad (2)$$

і його чисельна реалізація називається методом *k*-середніх.

Алгоритм *k*–середніх полягає у тому, що на початку обирається *k* довільних центри, далі, за цими центрами, решта множини розбивається на групи. На наступному кроці обчислюються нові центри для отриманих кластерів таким чином, щоб квадрат евклідової відстані від елемента кластера до його центроїду був меншим ніж відстань до центроїдів решти кластерів [4] .

Модифікацією цього методу можна вважати *fuzzy k*-means або *c*-means [5], який відрізняється тим, що кожен елемент кластеру належить до нього з певною ймовірністю, і його не обов'язково можна чітко віднести до одного з кластерів, тобто групи можуть перетинатися.

Неієрархічні методи більш стійкі до шумів, неправильної метрики та наявності незначимих параметрів у порівнянні з ієрархічними методами, які виграють, у випадках з невизначеною кількістю кластерів, ітерацій, або умови зупинки. Також ієрархічні методи дозволяють більш детально вивчити структуру даних.

Проблеми кластеризації, робота з пропусками

Розбиття множини на групи подібних об'єктів це потужний механізм підготовки даних до подальшого аналізу, але існує проблема обробки даних, що не є повними, тобто мають пропуски. Відсутні дані створюють багато труднощів, оскільки більшість процедур аналізу даних не були для них призначені [6]. Зважаючи на те, що в реальних даних велика ймовірність отримати таку ситуацію, коли інформація, що аналізується не повна, можна сказати, що відмовитися від кластеризації за цією причиною не вихід, тому така задача потребує вирішення і є актуальною.

Отже, існують декілька можливих рішеньпроблеми обробки даних з пропусками. Їх умовно можна поділити на такі, що націлені на попередню підготовку даних і ті, що вимагають адаптації (модифікації) стандартних алгоритмів кластеризації для обробки безпосередньо даних з пропусками. Кожен з цих варіантів має свої недоліки та переваги, і не є універсальним для рішення проблеми.

Розглянемо деякі варіанти попередньої обробки вхідних даних:

1. Виключення рядків з наявністю пропусків

Це метод, що легко реалізувати, але він може призвести до суттєвої втрати важливих даних. Його можна використовувати лише тоді, коли пропуски в даних розміщені випадковим чином і їх доволі мало, щоб вплинути на кінцевий результат.

2. Метод найближчих сусідів

Базується на тому, щоб знайти найближчий, за певним критерієм, рядок, схожий на рядок з пропуском. Далі, для позбавлення від пропуску, значення схожого рядка усереднюється за певним критерієм. Даний метод, у випадку великої кількості пропусків, допускає ряд похибок.

Практичні результати застосування методів обробки даних з пропусками

Таблиця 1 – **Похідні дані для дослідження**

№	Назва	Опис
1	age	Вік, в роках
2	sex	Стать (1 = чоловік; 0 = жінка)
3	cp	Тип болю в грудях
4	trestbps	Артеріальний тиск у стані спокою (в мм рт.ст. при надходженні в госпіталь)
5	fbs	Рівень цукру в крові натще> 120 мг / дл (1 = істина; 0 = брехня)
6	restecg	Результати електрокардіографії в спокої
7	thalach	Досягнуто максимальний пульс
8	exang	Розвинути стенокардію, викликану фізичним навантаженням (1 = так; 0 = ні)

3. Заповнення пропусків середніми значеннями у стовпці

Відбувається заміна пропусків на їх оцінки, найчастіше це може бути середнє значення вибірки, мода, медіана і регресивні моделі, коли невідоме значення ознаки обчислюється за допомогою знайденої функції регресії за відомим ознаками. До недоліків слід віднести спотворення, що вносяться в розподіл даних та зменшення дисперсії.

4. Регресійний аналіз

Так само потребує випадкового розподілення пропусків і залежить від правильно обраного методу регресійного аналізу. Для початку проведемо кластеризацію похідної таблиці, яка не містить пропущених даних. Для кластеризації оберемо метод – *fuzzy k-means* і мову програмування R. Виконаємо побудову "нечітких" кластерів з використанням функції fanny() з пакету cluster:

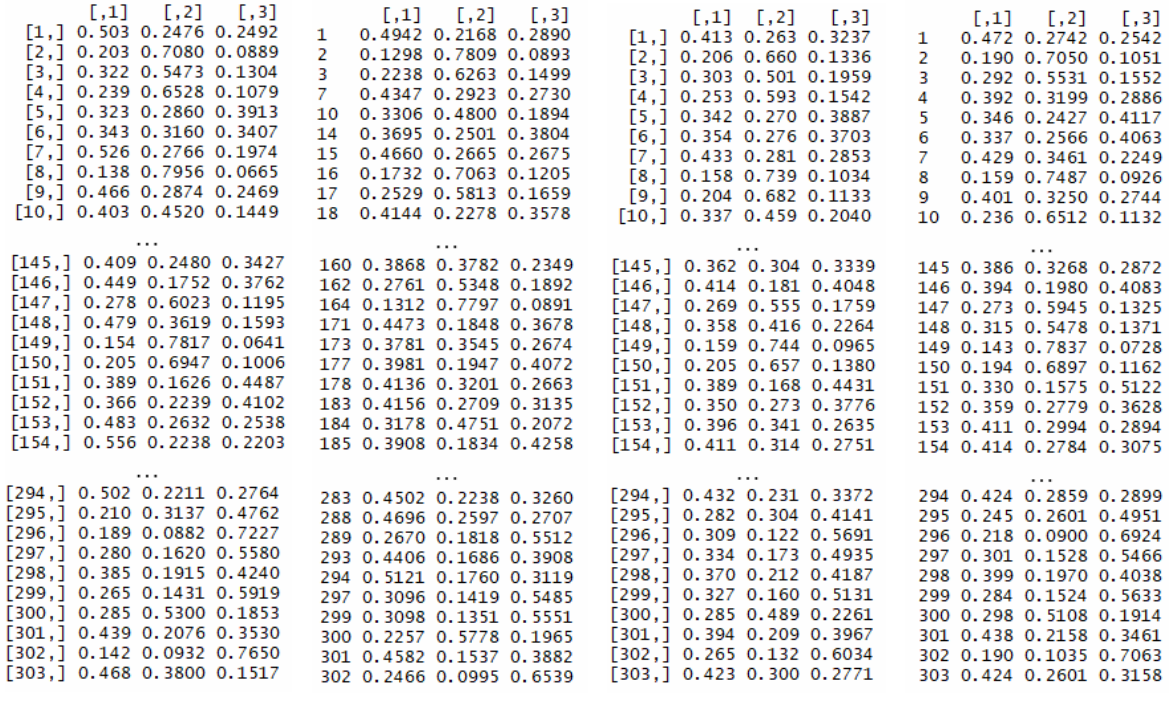
```
res.fanny <- fanny(data, k = 3, memb.exp = 1.3,
metric = "euclidean", stand = TRUE, maxit = 500)
print(res.fanny$membership,3)
```

- обрано 3 кластери (k = 3);

- memb.exp обрано таким чином, що memb.exp -> 1 дає більш чітку кластеризацію, тоді як memb.exp -> Inf призводить до повної нечіткості;

- metric - рядок символів, який вказує метрику, що буде використовуватися для розрахунку відмінностей між спостереженнями. Варіанти «euclidean», «manhattan» і «SqEuclidean». Евклідові відстані - це корінь суми квадратів різниць, манхетенські відстані – сума абсолютних різниць, а «SqEuclidean», квадрат евклідових відстаней – сума квадратів різниць.

- stand – параметр логічного типу, якщо true, вимірювання в x стандартизируются перед обчисленням відмінностей. Вимірювання стандартизовані для кожної змінної (стовпчик) шляхом вирахування середнього значення змінної і ділення на середнє абсолютне відхилення змінної: maxit – максимальна кількість ітерацій, у нашому випадку – 500; перші, середні та останні 10 рядків, отримані за результатами кластеризації, наведено на рис. 1, а.



а

```
        [,1]  [,2]  [,3]
[1,]   0.503 0.2476 0.2492
[2,]   0.203 0.7080 0.0889
[3,]   0.322 0.5473 0.1304
[4,]   0.239 0.6528 0.1079
[5,]   0.323 0.2860 0.3913
[6,]   0.343 0.3160 0.3407
[7,]   0.526 0.2766 0.1974
[8,]   0.138 0.7956 0.0665
[9,]   0.466 0.2874 0.2469
[10,]  0.403 0.4520 0.1449
          ...
[145,] 0.409 0.2480 0.3427
[146,] 0.449 0.1752 0.3762
[147,] 0.278 0.6023 0.1195
[148,] 0.479 0.3619 0.1593
[149,] 0.154 0.7817 0.0641
[150,] 0.205 0.6947 0.1006
[151,] 0.389 0.1626 0.4487
[152,] 0.366 0.2239 0.4102
[153,] 0.483 0.2632 0.2538
[154,] 0.556 0.2238 0.2203
          ...
[294,] 0.502 0.2211 0.2764
[295,] 0.210 0.3137 0.4762
[296,] 0.189 0.0882 0.7227
[297,] 0.280 0.1620 0.5580
[298,] 0.385 0.1915 0.4240
[299,] 0.265 0.1431 0.5919
[300,] 0.285 0.5300 0.1853
[301,] 0.439 0.2076 0.3530
[302,] 0.142 0.0932 0.7650
[303,] 0.468 0.3800 0.1517
```

б

```
      [,1]  [,2]  [,3]
1   0.4942 0.2168 0.2890
2   0.1298 0.7809 0.0893
3   0.2238 0.6263 0.1499
7   0.4347 0.2923 0.2730
10  0.3306 0.4800 0.1894
14  0.3695 0.2501 0.3804
15  0.4660 0.2665 0.2675
16  0.1732 0.7063 0.1205
17  0.2529 0.5813 0.1659
18  0.4144 0.2278 0.3578
          ...
160 0.3868 0.3782 0.2349
162 0.2761 0.5348 0.1892
164 0.1312 0.7797 0.0891
171 0.4473 0.1848 0.3678
173 0.3781 0.3545 0.2674
177 0.3981 0.1947 0.4072
178 0.4136 0.3201 0.2663
183 0.4156 0.2709 0.3135
184 0.3178 0.4751 0.2072
185 0.3908 0.1834 0.4258
          ...
283 0.4502 0.2238 0.3260
288 0.4696 0.2597 0.2707
289 0.2670 0.1818 0.5512
293 0.4406 0.1686 0.3908
294 0.5121 0.1760 0.3119
297 0.3096 0.1419 0.5485
299 0.3098 0.1351 0.5551
300 0.2257 0.5778 0.1965
301 0.4582 0.1537 0.3882
302 0.2466 0.0995 0.6539
```

в

```
        [,1]  [,2]  [,3]
[1,]   0.413 0.263 0.3237
[2,]   0.206 0.660 0.1336
[3,]   0.303 0.501 0.1959
[4,]   0.253 0.593 0.1542
[5,]   0.342 0.270 0.3887
[6,]   0.354 0.276 0.3703
[7,]   0.433 0.281 0.2853
[8,]   0.158 0.739 0.1034
[9,]   0.204 0.682 0.1133
[10,]  0.337 0.459 0.2040
          ...
[145,] 0.362 0.304 0.3339
[146,] 0.414 0.181 0.4048
[147,] 0.269 0.555 0.1759
[148,] 0.358 0.416 0.2264
[149,] 0.159 0.744 0.0965
[150,] 0.205 0.657 0.1380
[151,] 0.389 0.168 0.4431
[152,] 0.350 0.273 0.3776
[153,] 0.396 0.341 0.2635
[154,] 0.411 0.314 0.2751
          ...
[294,] 0.432 0.231 0.3372
[295,] 0.282 0.304 0.4141
[296,] 0.309 0.122 0.5691
[297,] 0.334 0.173 0.4935
[298,] 0.370 0.212 0.4187
[299,] 0.327 0.160 0.5131
[300,] 0.285 0.489 0.2261
[301,] 0.394 0.209 0.3967
[302,] 0.265 0.132 0.6034
[303,] 0.423 0.300 0.2771
```

г

```
      [,1]  [,2]  [,3]
1   0.472 0.2742 0.2542
2   0.190 0.7050 0.1051
3   0.292 0.5531 0.1552
4   0.392 0.3199 0.2886
5   0.346 0.2427 0.4117
6   0.337 0.2566 0.4063
7   0.429 0.3461 0.2249
8   0.159 0.7487 0.0926
9   0.401 0.3250 0.2744
10  0.236 0.6512 0.1132
          ...
145 0.386 0.3268 0.2872
146 0.394 0.1980 0.4083
147 0.273 0.5945 0.1325
148 0.315 0.5478 0.1371
149 0.143 0.7837 0.0728
150 0.194 0.6897 0.1162
151 0.330 0.1575 0.5122
152 0.359 0.2779 0.3628
153 0.411 0.2994 0.2894
154 0.414 0.2784 0.3075
          ...
294 0.424 0.2859 0.2899
295 0.245 0.2601 0.4951
296 0.218 0.0900 0.6924
297 0.301 0.1528 0.5466
298 0.399 0.1970 0.4038
299 0.284 0.1524 0.5633
300 0.298 0.5108 0.1914
301 0.438 0.2158 0.3461
302 0.190 0.1035 0.7063
303 0.424 0.2601 0.3158
```

Рис. 1. Результат застосування функції *fanny*: а – всі дані; б – при видалених 30% даних; в – при заміні 30% значень на медіани; г – при заміні 30% значень з урахуванням структури зв'язків

В результаті виводиться матриця коефіцієнтів приналежності, максимальний з яких визначає цільовий кластер.

Для оцінки міри нечіткості, отриманої класифікації, використовується коефіцієнт поділу Dunn:

$$F_k = \sum_{i=1}^{n}\sum_{r=1}^{k}\frac{\mu_{ir}^2}{k} \qquad (3)$$

Даний коефіцієнт приймає значення *1* в разі чіткої кластеризації:

dunn_coeff	normalized
0.4203087	0.1304630

В даному випадку $F_k = 0,42$, а його нормована версія, що змінюється від 0 до 1 – 0.13.

Побудуємо діаграму. На рис. 2, а наведено ординаційну діаграму з результатами кластеризації.

```
fviz_cluster(res.fanny, frame.type = "norm", frame.level = 0.7)
```

Для подальшого аналізу внесемо у похідні дані 30% пропусків випадковим чином.

1. Кластеризація з видаленням пропусків

Перші, середні та останні 10 рядків, отримані за результатами кластеризації, наведено на рис. 1, б.

```
heartMissedNO <- subset(heartMissed, exang != "")
```

Коефіцієнт Dunn приймає значення:

dunn_coeff	normalized
0.4153731	0.1230597

На рис. 2, б наведено ординаційну діаграму з результатами кластеризації.

2. Кластеризація із заповненням пропусків вибірковими статистиками

Отже, введемо припущення, що взаємозв'язку між змінними, у даному випадку, немає, тоді ефективним способом заповнення пропусків буде використання середніх значень, для цього оберемо медіану.

```
heartMissedProcess<-heartMissed
ind <- apply(heartMissedProcess, 1, function(x) sum(is.na(x))) > 0
heartMissedProcess[ind, 1:8]
pPmI <- preProcess(heartMissedProcess[, 1:8], method = 'medianImpute')
heartMissedProcess[, 1:8] <- predict(pPmI, heartMissedProcess[, 1:8])
(Imp.Med <- heartMissedProcess[ind, 1:8])
```

Перші, середні та останні 10 рядків, отримані за результатами кластеризації, наведено на рис. 1, в.

Коефіцієнт Dunn приймає значення:

dunn_coeff	normalized
0.38203172	0.07304758

На рис. 2, в наведено ординаційну діаграму з результатами кластеризації.

3. Заповнення пропусків з урахуванням структури зв'язків

Попередній метод передбачав відсутність зв'язків між параметрами, це так званий "наївний" метод. Альтернативою йому можна назвати метод, що враховує зв'язки між параметрами.

Приклад для заповнення поля age:

```
lm(age ~ trestbps, data = heartMissKor)
ageTres <- function(oP) {if (is.na(oP)) return(NA)
  else return(34.7253 + 0.1508   * oP)
}
heartMissKor[is.na(heartMissKor$age), 'age'] <-
  sapply(heartMissKor[is.na(heartMissKor$age),
'trestbps'], ageTres)
heartMissKor[ind, 10]
```

Перші, середні та останні десять рядків, отримані за результатами кластеризації, наведено на рис. 1, г.

На рис. 2, г наведено ординаційну діаграму з результатами кластеризації при заміні 30% значень з урахуванням структури зв'язків.

Для проведення кластеризації і отримання параметрів на кожному етапі використовувалася функція:

```
kmean <- function(data) {
  res.fanny <- fanny(data, k = 3, memb.exp = 1.3,
         metric = "euclidean", stand = TRUE, maxit = 500)
  print(res.fanny$membership,3)
  res.fanny$coeff
  print(  res.fanny$coeff)
  Dunn <- res.fanny$membership^2
  fviz_cluster(res.fanny, ellipse.type = "norm", ellipse.level
= 0.7)
```

Висновки

В ході роботи було розглянуто методи кластеризації даних з пропусками.

Для проведення аналізу було обрано медичні дані, що складаються з 303 рядків. Для проведення кластеризації оброблених даних обрано метод нечіткої кластеризації *k* середніх.

За результатами проведеної роботи можна зробити висновок, що найгіршим варіантом є варіант видалення всіх рядків, які містять пропуски. Даний метод можливий лише у випадках коли вибірка містить мінімальну кількість пропусків, або тоді коли було попередньо проведено інший вид обробки і відбувається видалення залишків пустих значень. Найкращим вважається метод боротьби з пропусками з урахуванням взаємозв'язків між полями, але на даній вибірці він не значно перевершує метод заміни на середні значення.

Якщо порівнювати порівняльні таблиці, що є результатами методу *fanny*, то можна сказати, що обидва методи впоралися добре на даному наборі даних з урахуванням 30% пропусків.

Якщо аналізувати графіки, то можна сказати, що є похибки в кластеризації, відновлення відбулося не ідеально, тому слід враховувати втрату повної достовірності, при виборі одного з таких методів.

Отже, ідеального методу боротьби з неповними даними немає.

Для кожного окремого випадку необхідно детально аналізувати похідний набір даних, що буде найбільш прийнятний у конкретному випадку.

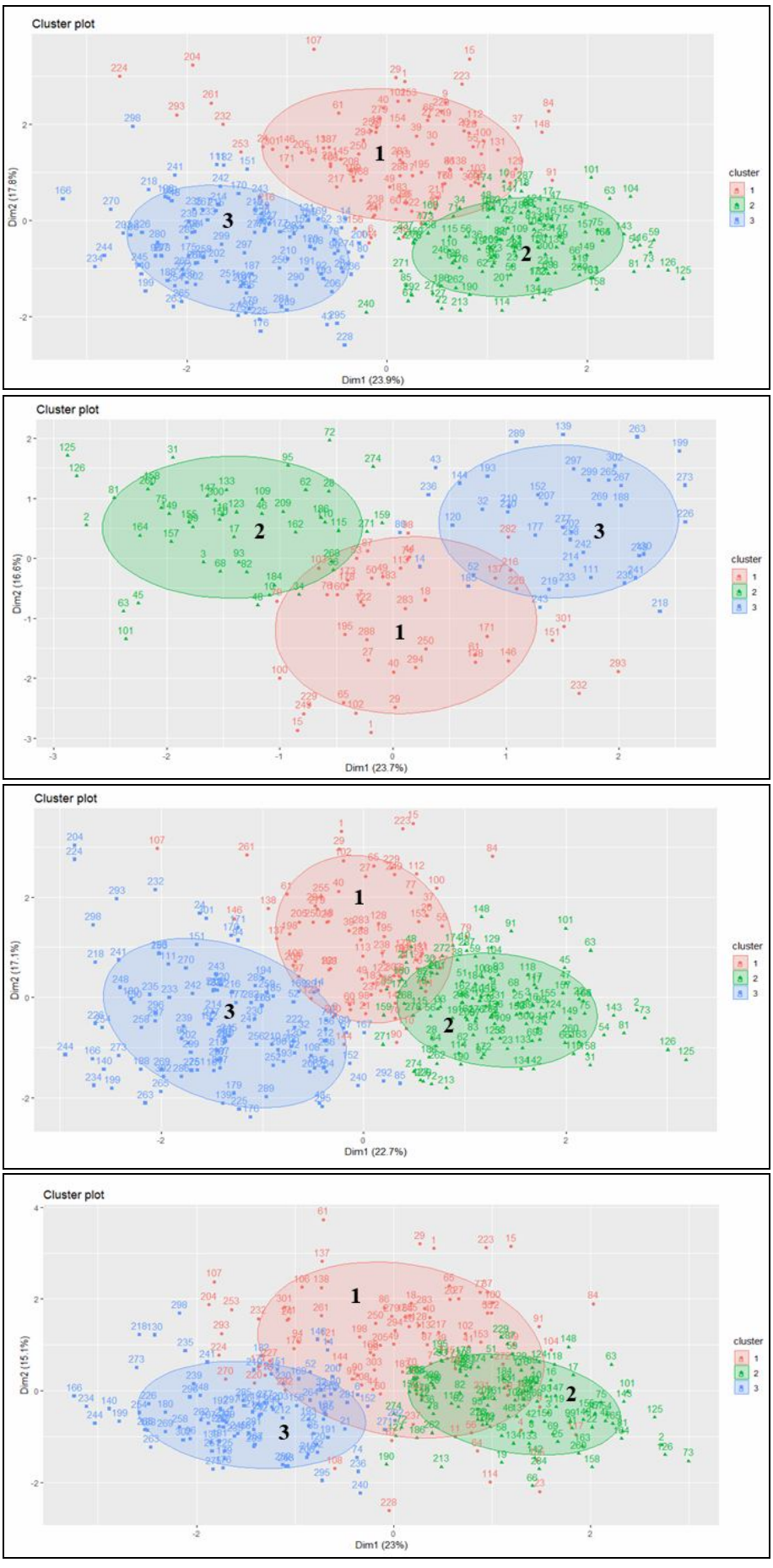


Рис. 2. Графік результатів нечіткої кластеризації: а – всі дані; б – при видалених 30% даних; в – при заміні 30% значень на медіани; г – при заміні 30% значень з урахуванням структури зв'язків

СПИСОК ЛІТЕРАТУРИ

1. Шумейко, А. А., & Сотник, С. Л. (2012). Интеллектуальный анализ данных. *Днепропетровск: Белая ЕА, 212*.
2. Жамбю, М., & Айвазян, С. А. (1988). *Иерархический кластер-анализ и соответствия*. Финансы и статистика.
3. Jain, A. K., Murty, M. N., Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264-323.
4. Steinley, D. (2006). K means clustering a half century synthesis. *British Journal of Math. and Stat. Psychology, 59*(1), 1-34.
5. Huang, Z., & Ng, M. K. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, *7*(4), 446-452..
6. Schafer, J. L., & Graham, J. W. (2002). Missing data: our view of the state of the art. *Psychological methods*, *7*(2), 147.
7. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed observations in Data Stream Mining tasks. Information Technology and Management Science, 19(1), 23-28.
8. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In 2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET) (pp. 665-670). IEEE.
9. Oleg, K., Sergii, M., & Mykhailo, S. (2017, October). Video Clustering via Multidimensional Time-Series Analysis. *In Proceedings of the 9th International Conference on Information Management and Engineering* (pp. 60-63). ACM.

Рецензент: д-р техн. наук, проф. С. Г. Семенов,
Національний технічний університет «Харківський політехнічний інститут», Харків
Received (Надійшла) 04.02.2019
Accepted for publication (Прийнята до друку) 21.03.2019

Обработка данных, содержащих пропуски, в задачах кластеризации

О.А. Кобилин, С. А. Выскребенцева, Р. В. Петрова

Предметом исследований являются методы подготовки и обработки входных данных, содержащих пропущенные значения, для их дальнейшего анализа и кластеризации. **Целью** является рассмотрение существующих методов избавления от пропусков в данных в задачах кластеризации и целесообразность их использования в реальных ситуациях. **Задачи:** анализ преимуществ и недостатков каждого из методов, направленных на восстановление данных, для определения целесообразности использования в задачах кластеризации и выделение наиболее подходящего к применению, сравнение их между собой, оценка результативности по результатам сравнения кластеризации восстановленных данных с результатами кластеризации эталонных данных. Применяемыми **методами** являются: FCM, для проведения непосредственно кластеризации данных, метод удаления всех строк, содержащих пропуски, заполнение пропусков выборочными статистиками, заполнение пропусков с учетом структуры связей. Полученные **результаты**: эффективность применения методов при подготовке данных для дальнейшей кластеризации зависит от количества имеющихся пропусков в исходном наборе. Если таких строк достаточно мало, то каждый, из рассмотренных методов, может быть использован и дать необходимые результаты. Но, если строк с пропусками достаточно много, например, 30%, тогда наиболее приемлемыми для использования можно назвать методы, связанные с заменой значений, однако следует учитывать, что такая замена может привести к искажению данных, а в итоге и результатов. **Выводы. Научная новизна -** исследование проблемы кластеризации данных, содержащих пропущенные значения и рассмотрение методов, которые позволяют решить эту задачу. Проведение экспериментов и сравнения результатов каждого из методов, выводы о целесообразности использования того или иного метода и побочные эффекты. **Практическая значимость** работы заключается в определении возможности использования в реальных задачах, которые обычно не являются идеальными и с большой вероятностью содержат пустые значения, методов обработки данных для использования их в задачах кластеризации.

Ключевые слова: кластеризация, неполные данные, обработка и анализ данных, Data Mining, FCM, методы восстановления данных, язык программирования R.

Processing incomplete data in cluster tasks

O.A. Kobylin, S.O. Vyskrebentseva, R.V. Petrova

The subjects of research is the methods of preparation and processing of input data containing missing values for their further analysis and clustering. **The goal** is to consider existing methods of getting rid of data gaps in clustering problems and the appropriateness of their use in real situations. **The tasks** include: analysis of advantages and disadvantages of each of the methods aimed at recovering data, to determine the appropriateness of use in clustering tasks and highlighting the most suitable for use, comparing them with each other; performance evaluation by comparing the recovered data clustering with the clustering results of the reference data. **The used methods**: FCM method for direct data clustering, methods of deleting all lines containing omissions, filling in omissions with selective statistics, filling in omissions taking into account the structure of links. The obtained **results:** efficiency of applying the methods to preparing data for further clustering depends on the number of omissions in the original set. If there are few such lines, then each of the considered methods can be used to obtain the necessary results. But, if there are a lot of lines with omissions , for example, 30%, then the methods that are associated with the replacement of values can be called the most acceptable for use, however, it should be borne in mind that this replacement can lead to distortion of the data, and ultimately the results. **Conclusions. Scientific novelty -** investigation of the problem of incomplete data clustering and consideration of methods that can solve this problem. Conducting experiments and comparing the results of each of the methods, conclusions about the advisability of using one of them and side effects. **The practical significance** of the paper consists in determining the possibility to use it in real tasks, which are usually not ideal and most likely contain empty values, data processing methods for using them in clustering tasks.

Keywords: clustering, incomplete data, data processing and analysis, Data Mining, FCM, data recovery methods, programming language R.