РОЗДІЛ «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»

УДК 002.704:303.7

ШУМЕЙКО О.О., д.т.н., професор
ШЕВЧЕНКО Г.Я.*, к.т.н., зав. наук. відділу
ПАНКРАТОВА Д.В., магістр

Дніпровський державний технічний університет, м. Кам'янське
*ТОВ «Ноосфера», м. Дніпро

ВИКОРИСТАННЯ МЕТОДІВ КЛАСТЕРИЗАЦІЇ В ЗАДАЧАХ ІНФОГРАФІКИ

Вступ. У сучасному світі людині щодня доводиться стикатися зі сприйняттям і обробкою величезного масиву інформації, тому тенденція до максимальної візуалізації змісту стає однією з найбільш пріоритетних у процесі ефективної комунікації [1]. Візуалізація даних є важливим елементом аналізу числової інформації, що дозволяє забезпечити чіткість і системність в сприйнятті інформації, а також швидке і правильне її декодування. До візуалізації відносяться діаграми різного виду, графіки і, звичайно, інфографіка.

Інфографіка дозволяє зв'язати різнорідні дані і представити їх в зручному візуальному вигляді. Проблема зв'язування даних різної природи сама по собі нетривіальна, і не може бути єдиного універсального підходу.

Постановка задачі. У даній роботі ми розглянемо застосування методів кластеризації до побудови елементів інфографіки. Формально під задачею кластерного аналізу заданої множини об'єктів розуміється завдання знаходження деякого розбиття цієї множини об'єктів на підмножини таким чином, щоб елементи, що відносяться до однієї підмножини, розрізнялися між собою в значно меншій мірі, ніж елементи з різних підмножин. Підмножини, які володіють подібними властивостями, називаються кластерами [2].

Результати роботи. Одним з найпопулярніших методів чисельного аналізу є метод найменших квадратів. Для завдання кластеризації він виглядає наступним чином:

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

по всіх s_j та k.

Чисельна реалізація цієї задачі називається методом k-середніх.

Метод k-середніх. Ідея метода складається з наступного: на початку вибирається k довільних початкових центрів із множини $\Im$. Далі всі об'єкти розбиваються на k груп, найбільш близьких до відповідного центру. На наступному кроці обчислюються центри знайдених кластерів. Процедура повторюється ітераційно до тих пір, поки центри кластерів не стабілізуються.

Алгоритм розбиття об'єктів $x_i\left(i = 0,1,...,n\right)$ заснований на мінімізації між кластерної відстані, у випадку, якщо в якості відстані використовується середньо квадратична норма ℓ_2, функція мети буде мати вигляд

$$S = \sum_{j=1}^{k}\sum\left\{\left|x_i - \mu_j\right|^2 \middle| x_i \in c_j\right\},$$

де x_i – i-й об'єкт, а c_j представляє собою j-й кластер з центром μ_j.

Структура алгоритму складається з наступних кроків.

1. Для ініціалізації алгоритму вибираємо *k* центрів кластерів.
2. Кожному з *n* об'єктів ставимо у відповідність кластер, виходячи з мінімізації ℓ_2 норми між об'єктом і центром відповідного кластеру.
3. Перерахуємо центри нових отриманих кластерів.
4. Для кожного *i,* такого, що $x_i \in c_j$, знайдемо

$$h = \arg\min\left\{\frac{n_r \left\| x_i - \mu_r \right\|_2}{n_r - 1}\right\},$$

де n_r – число об'єктів кластеру C_r.

Для розв'язання цієї задачі серед всіх елементів кластеру $x \in c_i$ знайдемо елемент z, який мінімізує похибку $\sum_{x \in c_i} \|x - z\|_2^2$, для чого знайдемо рішення задачі

$$\frac{\partial}{\partial z}\sum_{x \in c_i}\|x - z\|_2^2 = \frac{\partial}{\partial z}\sum_{x \in c_i}\left(\|x\|_2^2 - 2x^T z + \|z\|_2^2\right) = \sum_{x \in c_i}(-x + z) = 0, \text{ тобто } z = \frac{1}{n_i}\sum_{x \in c_i} x.$$

5. Якщо виконується умова

$$\frac{n_h \left\| x_i - \mu_h \right\|_2}{n_h - 1} < \frac{n_j \left\| x_i - \mu_j \right\|_2}{n_j - 1},$$

то треба перемістити об'єкт x_i з кластеру c_j в кластер c_h, після чого переобчислити значення центрів кластерів.

6. Якщо $i < n$, то переходимо до кроку 4, інакше – до кроку 3.

Критерієм зупинки алгоритму може бути або досягнення заданого числа ітерацій алгоритму, або досягнення функцією цілі заданого значення порога.

Метод ефективний в разі, якщо дані діляться на компактні групи, які можна описати сферою. Використання індикаторної функції дозволяє спростити запис базового алгоритму і записати в наступному вигляді.

Нехай $C = \{c_i\}_{i=1}^k$ – множина кластерів з центрами

$$\mu_i = \frac{\sum\left\{x_j \,\middle|\, x_j \in c_i\right\}}{\sum\left\{1 \,\middle|\, x_j \in c_i\right\}} = \frac{\sum_{j=1}^n u_j^i x_j}{\sum_{j=1}^n u_j^i},$$

де u_j^i – індикаторна функція, тобто

$$u_j^i = \begin{cases} 1, & \text{якщо } x_j \in c_i, \\ 0, & \text{в іншому випадку.} \end{cases}$$

Функція мети

$$S(C, \Im) = \sum_{i=1}^k \sum_{j=1}^n u_j^i d\left(x_j, \mu_i\right)$$

і умови

$$\sum_{i=1}^n u_j^i = 1, 0 < \sum_{j=1}^k u_j^i \le n,$$

тобто, кожен елемент може бути тільки в одному кластері, і кластер не може бути порожнім або містити елементів більше, ніж їх кількість.

Умова зупинки виконання алгоритму після ν-го кроку буде мати вигляд

$$\left|S^{\nu}(C,\Im)-S^{\nu-1}(C,\Im)\right|<\varepsilon,$$

де ε – обраний поріг.

Зауважимо, що при обчисленні критерію належності можна враховувати розмір кластеру, що дозволяє поліпшити ефективність алгоритму. Критерій того, що *j*-й елемент належить *i*-му, а не *k*-му кластеру, матиме вигляд:

$$\frac{n_i}{n_i-1}d(x_j,\mu_i)<\frac{n_k}{n_k-1}d(x_j,\mu_k),$$

де n_i – кількість елементів, які віднесені до кластеру c_j. Швидкість збігання методу – $O(n)$.

Розглянуті завдання інфографіки спираються на метод *k*-середніх, тому їм притаманні недоліки методу, до яких, перш за все, потрібно віднести:

- наявність апріорної інформації про кількість кластерів;
- чутливість до ізольованих віддалених елементів;
- істотна залежність швидкості збіжності методу від початкового вибору центрів кластерів.

Алгоритм стрибаючих жаб (SFLA). Для ефективного використання інфографіки дуже важливо мати можливість швидкого аналізу даної інформації, що істотно залежить від вибору центрів кластерів. Для вирішення цього завдання використовуємо алгоритм стрибків жаб (Shuffled Frog-Leaping Algorithm) [3, 4]. Цей алгоритм є оптимізацією метаеврістики, який імітує еволюцію групи жаб, які, стрибаючи випадково по камінню на ставку, шукають місце, де є максимальна кількість їжі. Кожна жаба приносить якесь рішення проблеми. Загальна популяція ділиться на групи жаб, які еволюціонують самостійно, щоб переглядати простір рішення в різних напрямках.

Алгоритм стрибаючих жаб (SFLA) дозволяє вирішити проблему центрів мас кластеру. В класичному варіанті алгоритму *k*-середніх використовується випадковий розподіл центрів мас кластерів, що дуже часто є джерелом похибки. А результатом роботи алгоритму стрибаючих жаб буде оновлення розташування на кожній ітерації центрів мас, що дасть можливість отримати краще рішення.

Припустимо, що початкова популяція жаб *F* випадковим чином визначається в просторі $(Xn, n=1,2,...,F)$, *fn* представляє значення придатності *n*-ї жаби. Всі жаби сортуються в порядку убування придатності і діляться на *q* груп, кожна з яких містить *p* жаб $(F=p*q)$. У кожній групі є жаби з кращою і гіршою придатністю, які позначаються *Xb* і *Xw* відповідно. Крім того, жаба з кращою придатністю у всіх популяціях позначається *Xg* .

Під час еволюції придатність регулюється наступним чином:

$$S=H*(Xb-Xw); \tag{1}$$

$$Xnew=Xw+S, \tag{2}$$

де *S* представляє значення зміни положення $(-Smax<S<Smax)$, *H* – випадкове число від 0 до 1. Якщо цей процес покращить рішення, то він замінює найгірше, інакше, те ж правило застосовується при заміні *Xb* глобальним рішенням *Xg :*

$$S=H*(Xg-Xw). \tag{3}$$

Після того, як отримали S, перерахуємо $Xnew$ згідно з (2). Якщо це нове рішення гірше, ніж найгірша жаба, тоді довільно генеруємо краще рішення, ніж Xw, і замінюємо Xw на $Xnew$.

Процес локального пошуку і перетасовки триває до виконання певного критерію збіжності. У нашому випадку кожна жаба складається з представника μ_i кожного регіону або класу, званого гравітаційним центром.

Отже, стрибок жаби дає кілька можливих кластерів, що представляють кандидата рішення. Тому необхідно зберегти тільки один, оцінюваний як найкращий, відповідно до визначеної мети $f = 1/\varepsilon$, де ε – квадратична помилка, мінімум якої є показником відповідної кластеризації:

$$\varepsilon = \sum_{i-1}^{k} \sum_{j-1}^{N_i} d\left(x_j^i, \mu_i\right), \tag{4}$$

де k – кількість кластерів, N_i – кількість елементів у кластері i, d представляє відстань між елементом $x_j(i)$ класу i і центром мас μ_i цього класу.

Для кращої кластеризації необхідно максимізувати f. Максимальне значення відповідності відповідає кластеризації з мінімальним відстанню між елементами, які належать до тієї ж множини.

Основні етапи алгоритму SFLA для кластеризації можна узагальнити наступним чином (рис.1).

- Крок 1 (задаємо початкові параметри). Ініціалізуємо розмір популяції F, кількість q, кількість жаб p, параметр H, кількість ітерацій n_1 для локального пошуку і кількість ітерацій n_2 для виконання програми.

- Крок 2 (генерація популяції жаб). Для реалізації алгоритму SFLA початкова популяція генерується F жабами. Кожна жаба Xn відповідає вектору B розмірності $D*k$ такій, що D є розмірність пошукового простору і k – кількість кластерів. Дійсно, кожна жаба складається з одного представника μ_i кожної області. Генерація цих гравітаційних центрів проводиться випадковим чином.

- Крок 3 (оцінка придатності кожної жаби). Після генерації початкової популяції кожен елемент призначається кластеру з найближчим центром. Потім всі жаби оцінюються, використовуючи функцію придатності (4). Кожній жабі асоційовано значення f.

- Крок 4 (сортування популяції). Популяція жаб сортується в порядку спадання згідно зі значенням придатності по порядку і визначається краща жаба Xg в цій популяції.

- Крок 5 (поділ популяції). Після сортування популяції жаб кожна група містить p жаб.

- Крок 6 (локальний пошук). У кожній групі визначаємо найкращу жабу Xb і гіршу жабу Xw. Найгірша жаба змінює положення, її нове місце $Xnew$ розраховується і оцінюється відповідне значення фітнес-функції f. Якщо $f(Xnew) > f(Xw)$, то це рішення замінює найгірше, інакше, Xw робить інший

стрибок згідно з (3). Тому ми перераховуємо нову позицію і її придатність f. Нове положення жаби, якщо воно дасть краще рішення, замінить найгірше, інакше генерують випадковим чином $Xnew$ кращим, ніж Xw.

- Крок 7. Різні групи об'єднуються, щоб знову сформувати популяцію жаб.
- Крок 8. Перейти до кроку 4, якщо число максимальної ітерації n_2 не досягнуто.

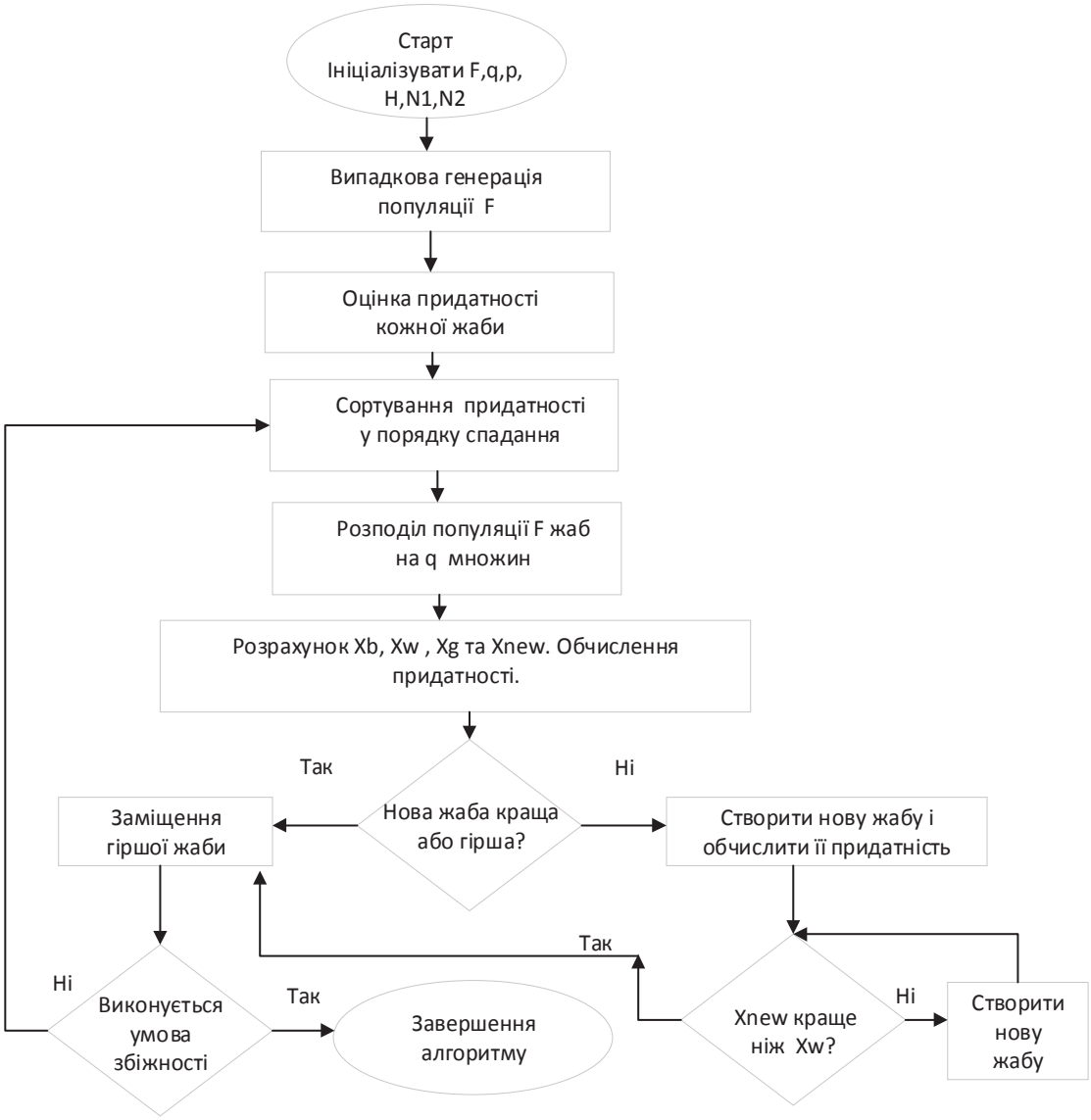

Рисунок 1 – Блок-схема алгоритму стрибаючих жаб (SFLA)

Приклади інфографіки, які використовують кластеризацію. *Перше завдання* полягає в кластеризації даних, які представляють собою незалежну випадкову величину, закон розподілу якої невідомий. Для цього прикладу використовуються дані про те, куди переселилися біженці з Донбасу (рис.2) і Криму. Переселення людей є незалежною випадковою величиною. При цьому потрібно врахувати, що якщо центри кластерів можна знайти, використовуючи традиційний метод *k*-середніх, то розміри кластеру залежать від кількості біженців в тому чи іншому кластері. Для побудови кластерів було вирише-

но використовувати множину багатокутників Парето, причому розміри багатокутника обчислюються попарно з сусідніми елементами відносно $m[i]/m[j]$, де $m[i]$ – кількість біженців в i-му кластері. Таким чином, чим більша площа багатокутника, що визначає кластер, тим більше в цьому районі біженців.

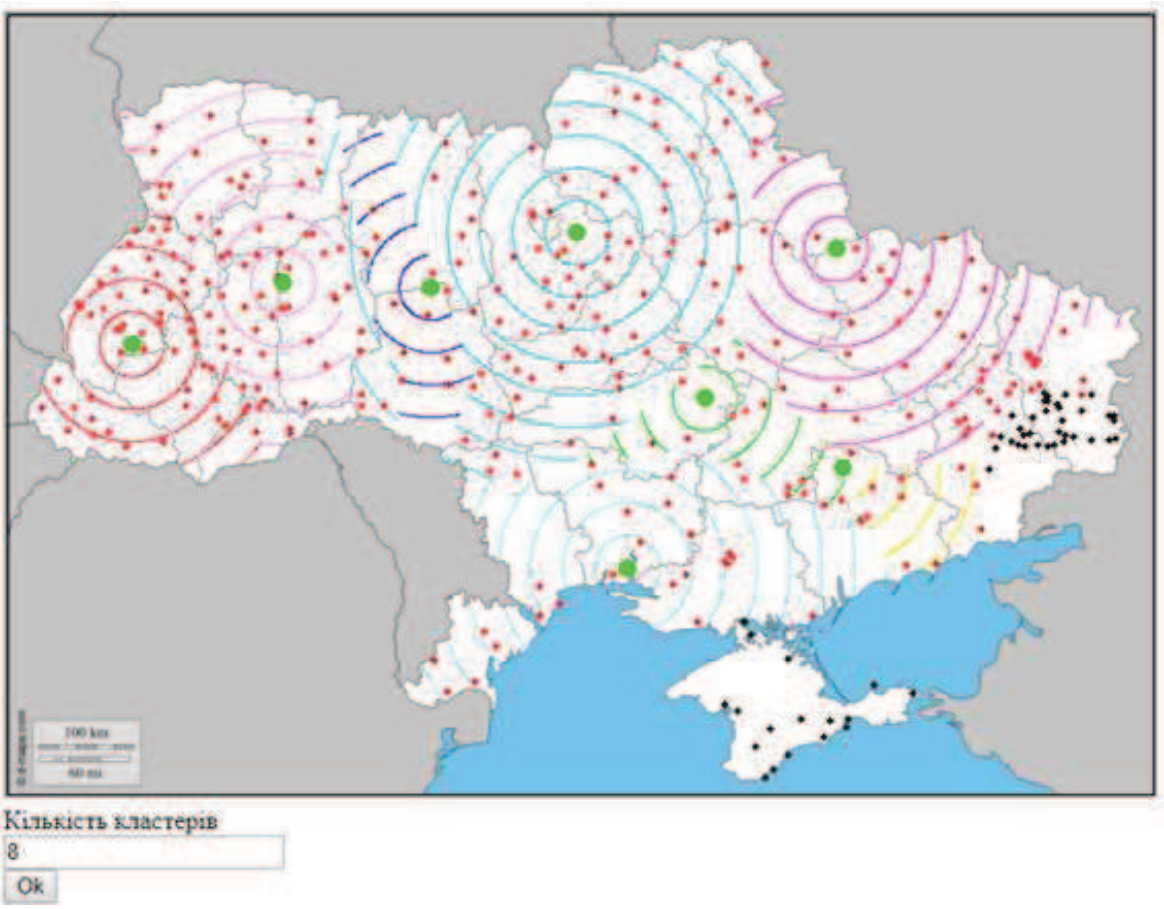

Рисунок 2 – Кластери розподілу біженців з Донбасу

Друге завдання має принципово іншу природу. Є вищі навчальні заклади. Потрібно визначити, які регіони України тяжіють до яких навчальних центрів (рис.3). Розв'язання цього завдання пов'язане з тим, що дані не є незалежною випадковою величиною. Якщо в регіоні мале число навчальних закладів, то звідки взятися ресурсу для зростання нових ВНЗ. І якщо є наявність великої кількості навчальних закладів, то цей факт сприяє формуванню в регіоні великої кількості як викладачів, так і студентів, частина з яких стає викладачами, що підсилює даний регіон з точки зору привабливості для надання освітніх послуг. Таким чином, вищі навчальні заклади «тяжіють» один до одного, що принципово відрізняється від традиційних методів кластеризації. Тобто в рамках кластеру модель нагадує галактику, що обертається біля деякого центра мас, в якому може і не бути ніякого елемента кластеру. Запропоновано для побудови кластеру використовувати закон тяжіння

$$F = \gamma \frac{mM}{r^2},$$

де в якості маси M елемента кластеру береться величина a^n, де n – кількість вузів, маса одного вузу m дорівнює одиниці, «гравітаційна стала» γ=1. Параметр a вибирається виключно з метою візуалізації. Центр кластеру є центром мас елементів кластеру,

а дозвільним правилом є умова, що дана точка більше тяжіє до центру того чи іншого кластеру.

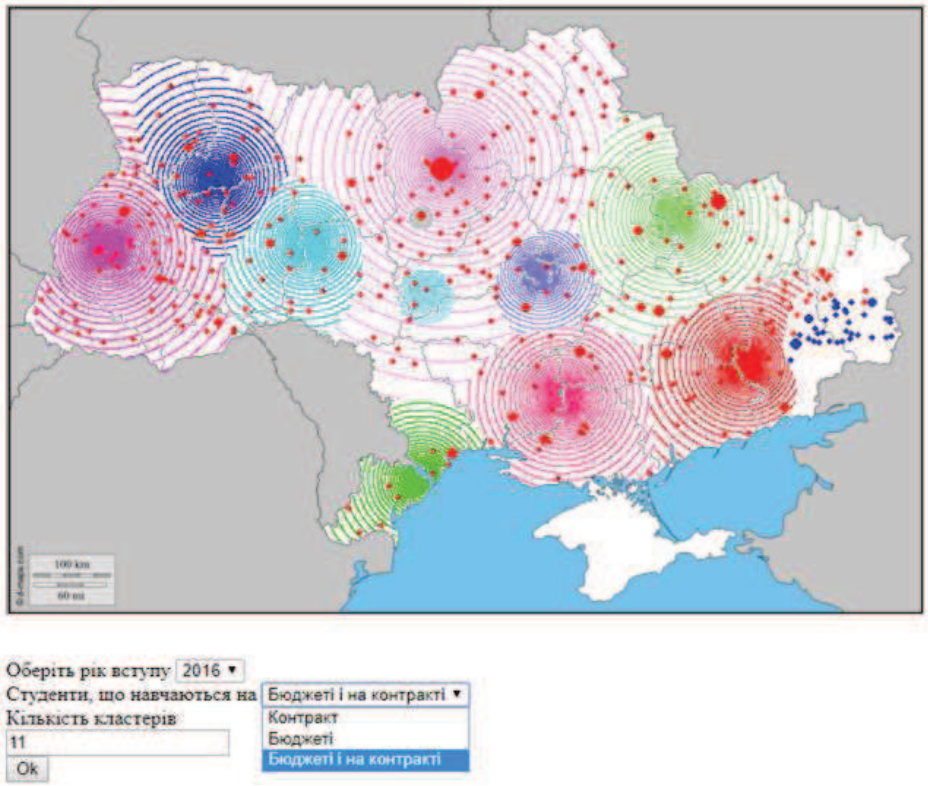

Рисунок 3 – Кластери розподілу студентів першого курсу

Наступний приклад виконано в рамках ідеології попереднього випадку, але для візуалізації наукових центрів України (рис.4). В даному випадку вищі навчальні заклади розглядаються як наукові центри (рис.5) і, крім того, в якості елементів використовується інформація про науково-дослідні інститути.

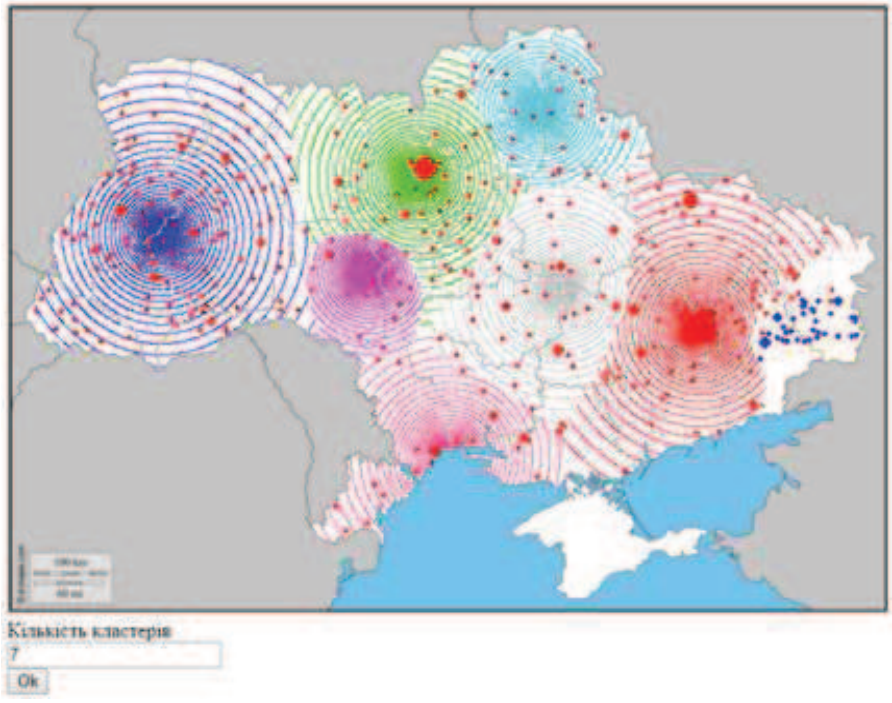

Рисунок 4 – Кластери розподілу науково-дослідних установ

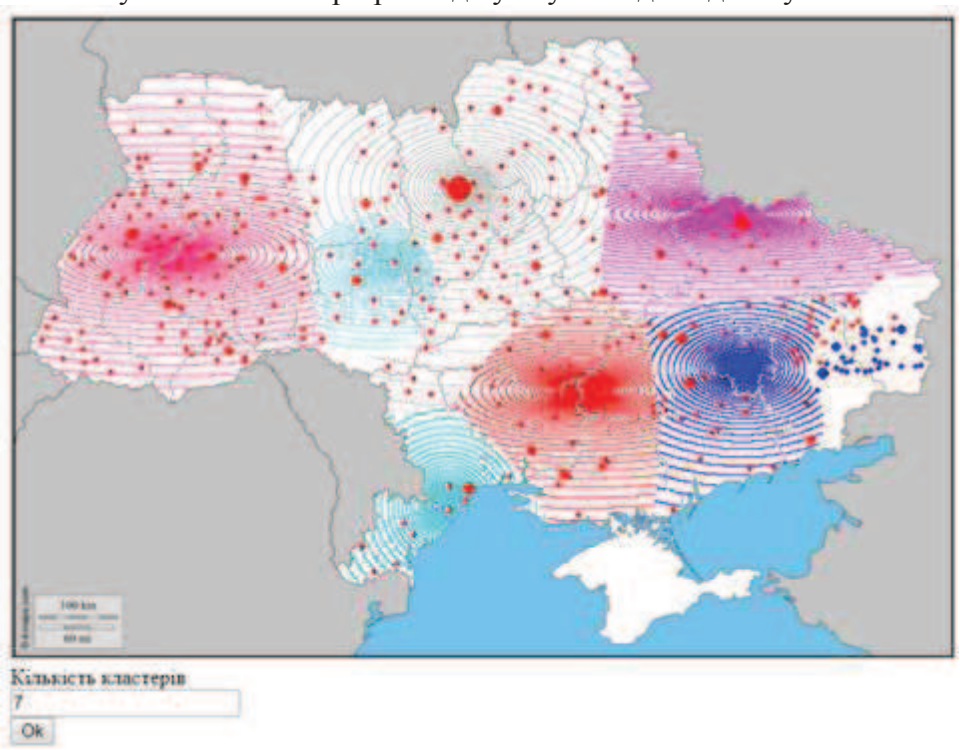

Рисунок 5 – Кластери розподілу вищих навчальних закладів

Навіть в такій грубій формі інфографіка досить цікава, особливо цікаво відзначити при великій кількості кластерів наявність ізольованих анклавів, де науковці «варяться у власному соку».

Цей метод допускає узагальнення. Так як кластер складається з декількох елементів (точок з вагою), то має сенс враховувати їх вплив один на одного, виділяючи їх головні компоненти. В цьому випадку кластер буде формуватися з урахуванням взаємного впливу матеріальних точок (елементів) і буде видно, в яких напрямках йде велика залежність елементів кластеру, в яких – менша.

Висновки. Використання кластерного аналізу дозволяє поліпшити якість інформаційного наповнення елементів інфографіки, дозволяючи виявити приховані, латентні зв'язки між візуалізованими елементами. Із застосуванням описаних методів можна ознайомитися на ресурсі http://sciencehunter.net/Services/visualization/.

ЛІТЕРАТУРА

1. Швед О.В. Инфографика как средство визуальной коммуникации / О.В.Швед // Science and Education, Philology (III). – 2013. – С.189-194. – Режим доступу: http://er.nau.edu.ua:8080/handle/NAU/15220.
2. Шумейко А.А. Интеллектуальный анализ данных (введение в Data Mining) / А.А.Шумейко, С.Л.Сотник. – Днепропетровск: Белая Е.А., 2012. – 212с. – Режим доступу: http://pzs.dstu.dp.ua/Data/dm.pdf.
3. Eusuff M. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization / M.Eusuff, K.Lansey, F.Pasha // Engineering Optimization. – 2006. – №2, vol. 32. – P.129-154.
4. Narimani M.R. A new modified shuffle frog leaping algorithm for non-smooth economic dispatch / M.R.Narimani // World Applied Sciences Journal. – 2011. – №6, vol. 12. – P.803-814.

Надійшла до редколегії 09.10.2017.