УДК 339.7:339.982

Бригадир В.О.
аспірант кафедри міжнародного менеджменту та маркетингу
Тернопільського національного економічного університету

ТЕОРЕТИЧНІ ЗАСАДИ КЛАСТЕРНОГО АНАЛІЗУ КРАЇН ЄС

THEORETICAL PRINCIPLES OF CLUSTER ANALYSIS OF THE EU COUNTRIES

АНОТАЦІЯ

У статті здійснено обгрунтування методу кластерного аналізу, що був обраним для стратифікації країн ЄС за рівнем соціально-економічного розвитку. Обрано програму Deductor Studio для вирішення завдання кластеризації країн-членів ЄС на основі використання самоорганізуючих карт Кохонена. Розглянуто послідовність кроків, необхідних для вирішення завдання кластеризації.

Ключові слова: кластеризація, кластерний аналіз, карта Кохонена, Deductor Studio, кластер.

АННОТАЦИЯ

В статье осуществлено обоснование метода кластерного анализа, который был выбран для стратификации стран ЕС по уровню социально-экономического развития. Выбрана программа Deductor Studio для решения задания кластеризации стран – членов ЕС на основе использования самоорганизующих карт Кохонена. Рассмотрена последовательность шагов, необходимых для решения задания кластеризации.

Ключевые слова: кластеризация, кластерный анализ, карта Кохонена, Deductor Studio, кластер.

ANNOTATION

The ground of method of cluster analysis that was select for the EU countries stratification after the level of socio-economic development is carried out in the article. The program Deductor Studio is selected for the decision of task to the clusterization of the EU countries on the basis of the use of the selforganising Kohonan maps. The sequence of steps necessary for the decision of task to the clusterization is considered.

Keywords: clusterization, cluster analysis, Kohonan map, Deductor Studio, cluster.

Постановка проблеми. Завдання кластеризації є вже давно відомим, фахівці у різних сферах науки використовують для цього низку інших термінів – групування, сегментація, таксономія тощо. Переважно під кластеризацією розуміють групування об'єктів за близькістю їхніх властивостей, коли елементи одного кластера (групи) подібні, а різних кластерів істотно відрізняються [1; 2; 3]. Її використовують через відсутність апріорних даних стосовно класів (страт), до яких можна віднести елементи досліджуваного набору даних, або коли цих елементів надзвичайно багато, це утруднює проведення ручного аналізу.

Загалом, сама постановка завдання кластеризації є нетривіальним завданням, оскільки у загальному випадку оптимальна кількість кластерів є невідомою; крім того, вибір міри (критерію) близькості властивостей об'єктів має суб'єктивний характер. Метод кластеризації застосовується для вирішення різноманітних наукових завдань. Нас цей підхід цікавитиме з точки зору стратифікації країн – членів ЄС за рівнем соціально-економічного розвитку. Розбиття множини країн ЄС на кластери (страти) допоможе виявити внутрішні закономірності у групах, покращити наочність представлення даних, висунути нові гіпотези, зрозуміти інформативність соціально-економічних властивостей цих країн.

Аналіз останніх досліджень і публікацій. Проблемам дослідження соціальної стратифікації країн ЄС присвячено роботи зарубіжних та вітчизняних науковців. Теоретичні основи соціальної стратифікації закладені А. Пігу та П. Сорокіним, згодом розвинуті у наукових працях А. Анурина, Н. Бердсолл, А. Гальчинського, В. Добренькова, О. Євтушенко, М. Іконникова, В. Іноземцева, Н. Коваліско, С. Макеєва та ін. Детальніше досліджено процеси соціальної стратифікації країн ЄС у роботах Д. Лук'яненка, М. Коніуш, Я. Столярчук та ін. Все ж недостатньо дослідженим є питання кластерного аналізу країн ЄС.

Мета статті полягає в аналізі та обґрунтуванні теоретичних засад соціальної стратифікації країн ЄС.

Виклад основного матеріалу дослідження. Зазвичай кластеризація є початковим етапом економіко-математичного дослідження об'єктів, за яким слідують подальші кроки, такі як оптимізація і прогнозування. Оскільки сама кластеризація не дає конкретних результатів аналізу, тому для отримання ефекту необхідно виконати змістовну інтерпретацію кожного кластера. Для цього аналітик має детально дослідити кожен кластер: його статистичні характеристики, кількість об'єктів у кластері, розподіл значень ознак цих об'єктів, іншими словами – з'ясувати, що об'єднує елементи кластера. Така інтерпретація значно полегшується за наявності засобів візуалізації, таких як дендрограми, діаграми, карти тощо.

У сфері інтелектуального аналізу даних на сьогодні запропоновано декілька десятків алгоритмів кластеризації та їхніх модифікацій [4; 5; 6]. Проте, незважаючи на такий широкий спектр, у реальній практиці економіко-математичного моделювання насамперед застосовуються алгоритми, які прості у використанні й одночасно забезпечують адекватні результати. Такими алгоритмами є:

1) метод k-means (k-середніх);
2) самоорганізуючі карти Кохонена;
3) EM-кластеризація.

Алгоритм k-means заснований на оптимізації цільової функції, котра визначає оптимальне

в певному сенсі розбиття множини об'єктів на кластери. Як цільова функція використовується сума квадратів зважених відхилень координат об'єктів від центрів шуканих кластерів. Алгоритм оптимізації цільової функції має ітеративний характер, і на кожній ітерації потрібно розраховувати матрицю відстаней між об'єктами.

Одним із недоліків алгоритму *k*-means є відсутність чіткого критерію для вибору оптимальної кількості кластерів. Зазвичай така апріорна інформація відсутня і дослідникові приходиться діяти методом проб і помилок. Для вирішення цієї проблеми було розроблена низка методів, які автоматично вибирають кількість кластерів (*k*), оптимальну згідно з обраним критерієм [5; 7; 8]. Переважно в них будуються моделі для різних значень *k*, а потім з них вибирається оптимальна. До найпопулярніших алгоритмів даного класу належить алгоритм *G*-means, в основі якого лежить припущення, що досліджувані дані мають гаусівський закон розподілу [8]. Фактично *G*-means є ітераційним алгоритмом, де *k*-means буде виконано *k* разів.

Крім того, підхід на основі *k*-means добре працює, коли дані у просторі утворюють компактні згустки, що добре відрізняються один від одного (сферичної або еліпсоїдної форми). А якщо дані мають вкладену форму, то жоден із алгоритмів сімейства *k*-means не впорається з таким завданням. Також алгоритм погано працює, коли один кластер значно більший за інший і вони розташовані поряд – виникає ефект «розщеплення» великого кластера. Тому доцільно провести кластерний аналіз з використанням нейронної мережі Кохонена, яка позбавлена згаданих недоліків.

Мережі Кохонена (вперше запропоновані фінським ученим Тойво Кохоненом [9; 10]) належать до самоорганізуючих нейронних мереж, які дають змогу виявляти кластери вхідних векторів, що володіють деякими загальними властивостями.

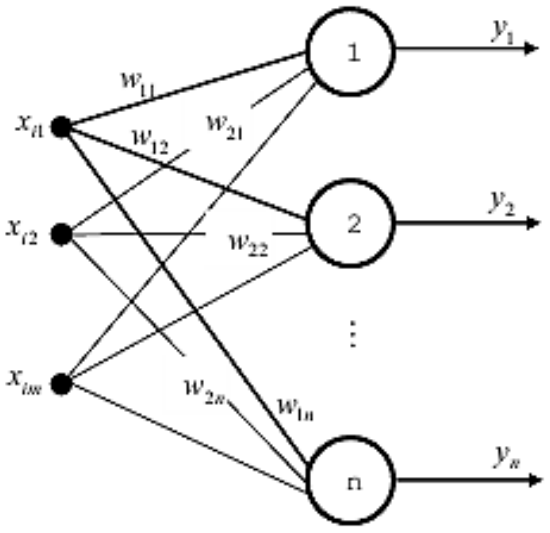

Рис. 1. Структура мережі Кохонена [9]

Мережа Кохонена (рис. 1) – це одношарова мережа, кожен нейрон якої з'єднаний з усіма компонентами *n*-мірного вхідного вектора. Під вхідним вектором розуміють опис одного із об'єктів, котрі підлягають кластеризації. Кількість нейронів збігається з кількістю кластерів, яку має виокремити мережа. Кожен *j*-й нейрон описується вектором ваг

$$\boldsymbol{w}_j = (w_{1j}, w_{2j}, w_{mj}),$$

де m – кількість компонентів вхідних векторів.

Вхідний вектор має вигляд:

$$\boldsymbol{x}_j = (x_{1j}, x_{2j}, x_{mj}).$$

У мережах Кохонена використовується навчання без учителя із застосовуванням механізмів конкуренції. При подачі на вхід мережі вектора $\underline{x}$ перемагає той нейрон, вектор ваг якого найменше відрізняється від вхідного вектора. Для нейрона-переможця виконується співвідношення

$$d(\mathbf{x}, \mathbf{w}_j) = \min_{1<i<n} d(\mathbf{x}, \mathbf{w}_i),$$

де n – кількість нейронів, j – номер нейрона-переможця, d ($\mathbf{x}$, $\mathbf{w}$) – відстань (згідно з обраною метрикою) між векторами $\mathbf{x}$ і $\mathbf{w}$. Найчастіше в якості міри відстані використовується евклідова міра.

Навколо нейрона-переможця утворюється так званий радіус навчання, який визначає скільки нейронів крім нейрона-переможця навчаються (тобто коректують свої ваги) на цій ітерації. Іншими словами, будь-який нейрон, у котрого відстань від вектора ваг до вектора ваг нейрона-переможця менша за радіус навчання, бере участь у згаданій корекції ваг.

Ваги нейрона-переможця і всіх нейронів, що лежать у межах його радіуса навчання, піддаються навчанню (адаптації) за правилом Кохонена

$$\mathbf{w}_i^{(k+1)} = \mathbf{w}_i^{(k)} + \eta_i^{(k)} \left[\mathbf{x} - \mathbf{w}_i^{(k)} \right],$$

де $\mathbf{x}$ – вхідний вектор, k – номер циклу навчання, $\eta_i^{(k)}$ – коефіцієнт швидкості навчання i-го нейрона з радіуса навчання в k-му циклі навчання.

Ваги нейронів, розташовані поза межами радіуса навчання, не змінюються.

Карти Кохонена (самоорганізуючі карти, або SOM – self-organizing map) [9; 11] призначені для візуального представлення багатовимірних властивостей об'єктів на двох осях. Карта Кохонена складається з комірок прямокутної або шестикутної форми, кожній з яких відповідає нейрон мережі Кохонена. Навчання нейронів відбувається аналогічно до нейронів мережі Кохонена. Об'єкти, вектори ознак яких близькі, потрапляють в одну комірку або в сусідні комірки.

Карти Кохонена дають змогу також представити отриману інформацію у простій і наочній формі шляхом нанесення розфарбування. Для цього розфарбовують вузли карти кольорами, що відповідають вибраними ознаками об'єктів.

Кожна ознака даних породжує своє розфарбування комірок карти – за величиною середнього значення цієї ознаки у даних, котрі потрапили в дану комірку.

Отже, як інструмент кластерного аналізу в подальших наших дослідженнях будемо використовувати самоорганізуючу карту Кохонена, перевагами якої порівняно з іншими алгоритмами є можливість візуального аналізу багатовимірних даних, стійкість до зашумлених даних, швидке і некероване навчання. До найпоширеніших застосувань карт Кохонена належать розвідувальний аналіз даних і виявлення нових явищ.

Серед широкого спектра програмних пакетів загального призначення варто виокремити продукти, які надають можливість повномасштабно реалізовувати мережі Кохонена: IBM SPSS Modeler [12], RapidMiner [13], Statistica Automated Neural Networks [14], Deductor [15], SOM Toolbox for Matlab [16] та інші. Переважно це додатки до загальновідомих громіздких статистичних і моделюючих пакетів.

З цього широкого набору програмних продуктів для завдань кластеризації країн – членів ЄС обрано програмний продукт Deductor компанії BaseGroup Labs. Вибір аналітичної платформи Deductor обумовлений такими її перевагами та можливостями:

– оступність не лише експертам з нейронних мереж, а і новачкам у сфері нейромережевих обчислень;

– простота у використанні у поєднанні з аналітичною потужністю; так набір майстрів проведе дослідника через усі етапи створення самоорганізуючих мереж Кохонена;

– багаті графічні та статистичні можливості, які полегшують інтерактивний дослідницький аналіз;

– російськомовний інтерфейс користувача та документація;

– наявність вільної програмної версії Deductor Academic.

У Deductor Studio включено повний набір механізмів, який дає змогу отримати інформацію з довільного джерела даних, провести весь цикл опрацювання (очищення і трансформацію даних, побудову моделей), найзручніше відобразити отримані результати (карти, таблиці, діаграми, дендрограми) і експортувати результати. Уся робота з аналізу даних в Deductor Studio базується на виконанні послідовності дій: імпорт → оброблення → візуалізація → експорт. Керування цими діями здійснюють відповідні програми-майстри.

Мережі та карти Кохонена будуються в Deductor Studio за допомогою обробника «Карта Кохонена», де реалізовано сам алгоритм Кохонена та спеціальний візуалізатор «Карта Кохонена». Це обумовлює відповідно два основні етапи розглядуваного підходу: 1) кластеризація об'єктів алгоритмом Кохонена; 2) побудова та інтерпретація карти Кохонена.

Отже, для проведення кластерного аналізу нами використано робоче місце аналітика Deductor Studio, яке входить до складу аналітичної платформи Deductor і містить набір механізмів імпорту, оброблення, візуалізації та експорту даних для швидкого й ефективного аналізу інформації.

Розглянемо послідовність кроків, необхідних для вирішення завдання кластеризації країн – членів ЄС, на основі використання самоорганізуючих карт Кохонена, реалізованих у рамках програми Deductor Studio (рис. 2).

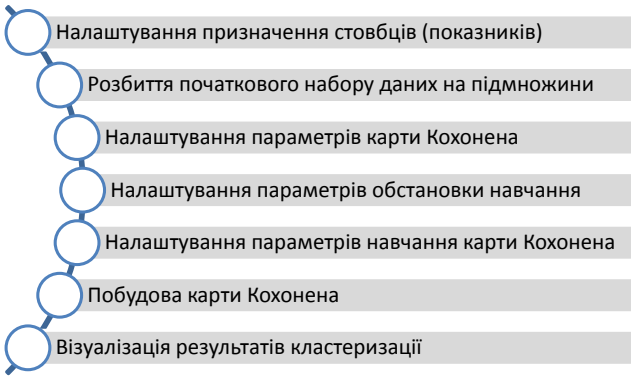

Рис. 2. Етапи кластеризації країн за допомогою карт Кохонена в Deductor Studio [9]

1. Налаштування призначення стовбців. Після імпорту початкових даних необхідно визначити призначення стовбців. У нашому випадку всі 12 стовбців з числовими даними (показниками) позначимо як вхідні, а інформаційний стовбець з назвами країн – вихідний. Останній не бере участь в розрахунках, лише інформативно допомагає в процесі аналізу. У програмі також можна встановити значимість кожного вхідного поля (показника), тобто кожен такий показник може мати ваговий коефіцієнт від 0 до 100%, який впливає на розрахунок відстані між векторами. Вважатимемо усі показники рівнозначними.

2. Розбиття початкового набору даних на підмножини. Оскільки методи кластеризації, у т.ч. й алгоритм Кохонена, є суб'єктивними, потреба виділяти окрему тестову множину відсутня. Тому розглядатимемо лише навчальну множину, сформовану із 100% записів.

3. Налаштування параметрів карти Кохонена. На цьому кроці реалізації алгоритму SOM заздалегідь задається конфігурація (розмір і форма) карти Кохонена, а також кількість нейронів у мережі. Виберемо шестикутні комірки, оскільки в цьому випадку відстані між центрами суміжних комірок будуть однаковими, що підвищує коректність візуалізації карти.

4. Налаштування параметрів обстановки навчання. Вказують умови припинення навчання (якщо помилка менше 0,05 або досягнуто 500 епох).

5. Налаштування параметрів навчання карти Кохонена. Результати кластеризації значною

мірою залежать від початкового ініціювання карти. Нами обрано спосіб ініціювання на основі власних векторів – саме цей спосіб найкраще підходить при першому ознайомленні з даними.

На цьому кроці обираються такі параметри, як швидкість і радіус навчання, а також функція сусідства. Функція сусідства визначає, які нейрони і в якій мірі будуть вважатися сусідніми щодо нейрона-переможця. В якості такої функції доцільно вибрати функцію Гауса, при використанні якої навчання проходить більш плавно і рівномірно, оскільки одночасно змінюються ваги усіх нейронів, що дає трохи кращий результат, ніж при застосуванні ступінчастої функції.

Також обираємо автоматичне визначення кількості кластерів і рівень значимості (0,01%). Автоматичний вибір означає, що кількість кластерів визначатиметься на основі методу *G*-means. Рівень значимості – параметр автоматичного визначення кластерів. Чим він більший, тим більшу кількість кластерів буде отримано. Зі статистичної точки зору, рівень значимості є ймовірністю справедливості нульової гіпотези про те, що значення у наявному наборі даних розподілені за нормальним законом. Цей параметр є ключовим для виділення кластерів алгоритмом *G*-means.

Таким чином, у процесі кластеризації нейронною мережею Кохонена спочатку навчається мережа Кохонена та будується відповідна їй карта, внаслідок чого записи розподіляються по комірках. Потім отримані комірки об'єднуються у кластери алгоритмом *G*-means.

6. Побудова карти Кохонена. На даному кроці проводиться власне навчання карти із заданими параметрами. Навчання може вважатися успішним, якщо відсоток розпізнаних прикладів досить великий (близький до 100%).

7. Візуалізація результатів кластеризації. До навченої мережі Кохонена пропонується спеціалізований візуалізатор – «Карта Кохонена».

За результатами формування карти отримуємо набір вузлів, який можна відобразити як двовимірний малюнок. При цьому кожному вузлові карти відповідає ділянка на малюнку (чотири- чи шестикутна), координати якої визначаються координатами відповідного вузла на решітці. Для візуалізації залишається лише визначити колір клітинок цієї картинки шляхом використання значення компонентів. Найпростішим варіантом є використання градацій сірого кольору. Тоді комірки, котрі відповідають вузлам карти, до яких потрапили елементи з мінімальними значеннями компонента або не потрапило взагалі жодного запису, відображатимуться чорним кольором, а комірки, в які потрапили записи з максимальними значеннями, відповідатимуть комірки білого кольору.

Зібравши воєдино карти усіх досліджуваних показників, отримаємо свого роду топографічний атлас, який дає інтегральне уявлення про структуру багатовимірних даних тобто відображає розташування компонентів, зв'язки між ними, а також відносне розташування різних значень компонентів.

Результати роботи алгоритму Кохонена відображаються на картах. Кожному вхідному полю (у нашому випадку – соціально-економічному показникові) відповідає своя карта. За результатами навчання нейронної мережі отриману карту можна представити у вигляді багатошарового куба, кожен шар якого є розфарбуванням, породженим одним із згаданих показників. Отриманий набір розфарбувань використовуватимемо для аналізу закономірностей, наявних між компонентами набору даних.

Результати кластеризації алгоритмом Кохонена можна побачити не лише на карті, а і на спеціальному візуалізаторі «Профілі кластерів». Тут можна подивитися загальну структуру сформованих кластерів, оскільки відображаються всі розглянуті показники разом із характером їхнього впливу на склад кластера. Для кожної розглянутої у кластері властивості обчислюються: довірчий інтервал, середнє та стандартне відхилення, стандартна похибка.

Важливим результатом для аналізу тут є значимість атрибутів (показників), яка показує їхній ступінь впливу на утворення того чи іншого кластера і виражається у відсотках. Для розрахунку значимості вхідних полів у пакеті Deductor використовується *t*-критерій Стьюдента, а для загальної значимості застосовується *F*-критерій Фішера.

Пакет Deductor дає змогу побачити ступінь подібності між кластерами за допомогою візуалізатора «Матриця порівнянь».

Висновки. Таким чином, можна зробити висновок, що кластеризацію країн ЄС доцільно проводити за допомогою алгоритму Кохонена, що дозволить більш детально проаналізувати утворені кластери країн ЄС.

БІБЛІОГРАФІЧНИЙ СПИСОК:

1. Паклин Н.Б. Бизнес-аналитика: от данных к знаниям (+CD): учеб. пособие, 2-е изд., перераб. и доп. / Паклин Н.Б., Орешков В.И. – СПб.: Питер, 2013. – 704 с.
2. Мандель И.Д. Кластерный анализ: Пер. с англ. / Мандель И.Д. – М.: Финансы и Статистика, 1988. – 176 с.
3. Барсегян А.А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP / А.А. Барсегян, М.С. Куприянов, В.В. Степаненко, И.И. Холод. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2007. – 384 с.
4. Воронцов К.В. Алгоритмы кластеризации и многомерного шкалирования: курс лекций [Електронний ресурс] / Воронцов К.В. – МГУ, 2007. – Режим доступу: www.ccas.ru/voron/download/clustering.pdf.
5. Jain A. Data Clustering: A Review / Jain A., Murty M., Flynn P. // ACM Computing Surveys. – 1999. – Vol. 31, No. 3. – P. 264-323.
6. Котов А. Кластеризация данных [Електронний ресурс] / Котов А., Красильников Н. – 2006. – Режим доступу: http://logic. pdmi.ras.ru/~yura/internet/02ia-seminar-note.pdf.
7. Tibshirani R. Estimating the Number of Clusters in a Dataset via the Gap Statistic / Tibshirani R., Walther G., Hastie T. //

Journal of the Royal Statistical Society. – 2001. – Vol. 63. – P. 411-423.
8. Hamerly G. Learning the k in k-means / Hamerly G., Elkan C. // Advanced in Neural Information Processing Systems 16. – 2004. – P. 281-288.
9. Кохонен Т. Самоорганизующиеся карты: Пер. с англ. / Тойво Кохонен. – М.: Бином, 2008. – 656 с.
10. Дебок Г. Анализ финансовых данных с помощью самоорганизующихся карт: Пер. с англ. / Г. Дебок, Т. Кохонен. – М.: Альпина, 2001. – 317 с.
11. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1104 с.
12. Офіційний сайт компанії IBM [Електронний ресурс]. – Режим доступу: http://www.ibm.com/software/ analytics/spss/products/modeler/.
13. Офіційний сайт RapidMiner [Електронний ресурс]. – Режим доступу: https://rapidminer.com.
14. Офіційний сайт компанії StatSoft [Електронний ресурс]. – Режим доступу: http://statsoft.ru/products/ STATISTICA_ Neural_ Networks/.
15. Офіційний сайт компанії BaseGroup Labs [Електронний ресурс]. – Режим доступу: http://basegroup.ru/deductor/description.
16. SOM Toolbox for Matlab [Електронний ресурс]. – Режим доступу: http://www.cis.hut.fi/projects/somtoolbox/.