ЛАБОРАТОРНА РОБОТА №4. КЛАСТЕРИЗАЦІЯ ДАНИХ. ПОБУДОВА СИСТЕМИ РЕКОМЕНДАЦІЇ КНИГ.

Теоретична довідка

Кластеризація — це процес групування багатовимірних наборів даних у тісно пов'язані групи. Класичними алгоритмами кластеризації є k-середніх(means) і **S**elf-**O**rganizing **M**ap (SOM).

Ще одним менш відомим алгоритмом для кластеризації є Centroid Neural Network (CentNN) for Unsupervised Competitive Learning.

CentNN — це алгоритм неконтрольованого змагального навчання, заснований на класичному алгоритмі кластеризації k-середніх, який оцінює центроїди пов'язаних груп кластерів у даних для навчання. CentNN не вимагає ні попередньо визначеного розподілу для коефіцієнта навчання, ні загальної кількості ітерацій для кластеризації. У дослідженні [1] показують, що CentNN сходиться набагато швидше, ніж звичайні алгоритми із схожою якістю кластеризації, тоді як інші алгоритми можуть давати нестабільні результати залежно від початкових значень коефіцієнта навчання та загальної кількості ітерацій.

Опис алгоритму

Algorithm CNN(M,N) [*M: number of clusters, N: number of data vectors*]
[*Initialize weights* $\mathbf{w}_1$ *and* $\mathbf{w}_2$]
Find the centroid, $\mathbf{c}$, of all data vector
Initialize $\mathbf{w}_1$ and $\mathbf{w}_2$ around $\mathbf{c}$ with small $\boldsymbol{\varepsilon}$:
$\mathbf{w}_1 := \mathbf{c} + \boldsymbol{\varepsilon},\ \mathbf{w}_2 := \mathbf{c} - \boldsymbol{\varepsilon}$
$N_1 := 0,\ N_2 := 0$
$k := 2,\ epoch := 0$
for $(k \le M)$
do
$loser := 0$
for $(n \le N)$
Apply a data vector $\mathbf{x}(n)$ to the network
Find the winner neuron, j, in this epoch for $1 \le j \le k$
if $(epoch \ne 0)$, **then** Set i is winner neuron, i, for $\mathbf{x}(n)$ in previous epoch.
if $i \ne j$, **then** neuron, i, is loser neuron.
if $(epoch == 0$ or $i \ne j)$
Run UpdateCNNWeight$(\mathbf{x}(n), \mathbf{w}_i, \mathbf{w}_j, epoch)$
$loser := loser + 1$
endif
$n := n + 1$
endfor [*check for all data*]
$epoch := epoch + 1$
while $loser \ne 0$
if $k \ne M$
Split group with the most error, j, by adding small vector, $\boldsymbol{\varepsilon}$, nearby group j :
$\mathbf{w}_j = \min_i E_i = \min_i \sum_{k=1}^{N_i} \|\mathbf{x}_i(k) - \mathbf{w}_i\|^2,\ \ 1 \le i \le M$
$\mathbf{w}_{k+1} := \mathbf{w}_j + \boldsymbol{\varepsilon}$
endif
$k := k + 1,\ N_k := 0$
endfor
end

Procedure UpdateCNNWeight$(\mathbf{x}, \mathbf{w}_i, \mathbf{w}_j, epoch)$
Update winner neuron : $\mathbf{w}_j(n) := \mathbf{w}_j(n) + \frac{1}{N_j+1}[\mathbf{x}(n) - \mathbf{w}_j(n)]$
if $(epoch \ne 0)$ [*loser neuron occurred only when epoch* $\ne 0$]
Update loser neuron : $\mathbf{w}_i(n) := \mathbf{w}_i(n) - \frac{1}{N_i-1}[\mathbf{x}(n) - \mathbf{w}_i(n)]$
endif
end

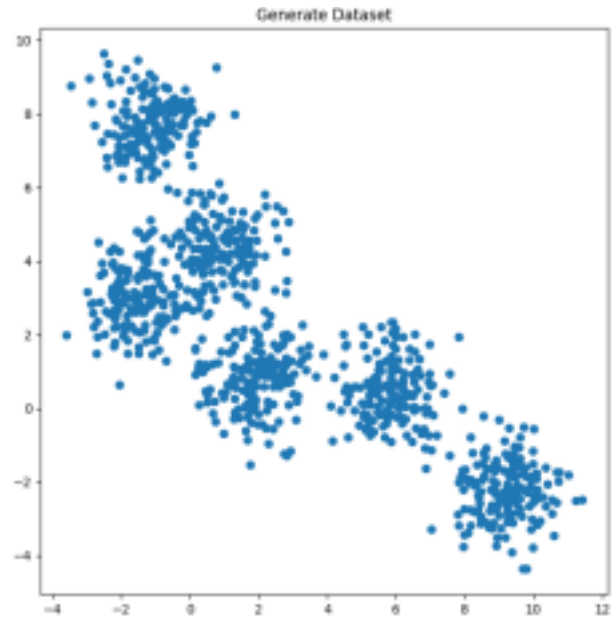
Для детальнішого опису алгоритму дивіться наприклад [2].

Варіанти завдань для кластеризації з допомогою CentNN

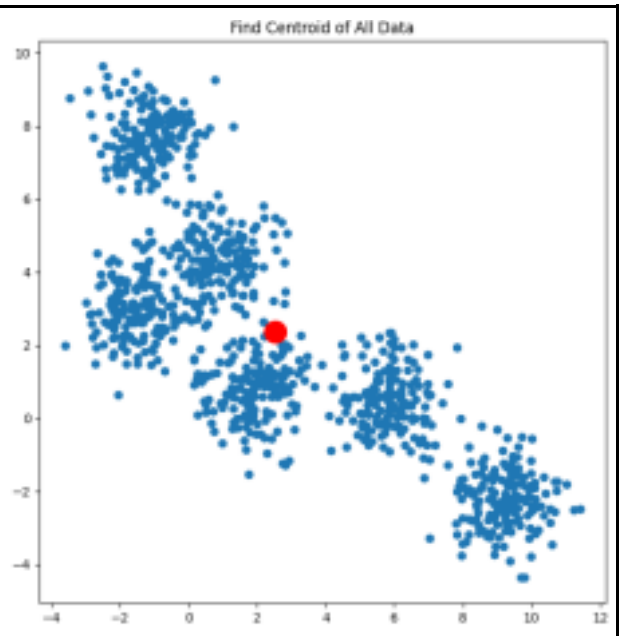
	Набори даних
1	https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/unbalance2.txt https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/s2.txt https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/a2.txt
2	https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/s1.txt https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/a2.txt https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/D31.txt
3	https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/s2.txt https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/dim032.txt https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/dim064.txt
4	https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/unbalance2.txt https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/D31.txt https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/dim128.txt
5	https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/a2.txt https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/a3.txt https://web.archive.org/web/20230316224609/https://cs.joensuu.fi/sipu/datasets/D31.txt

* Першим кроком є встановлення кількості кластерів відповідно до вашого варіанту.

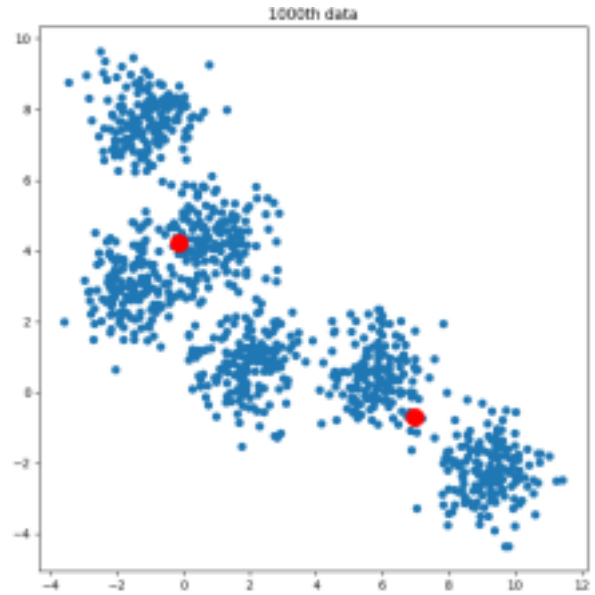
Очікуваний результат



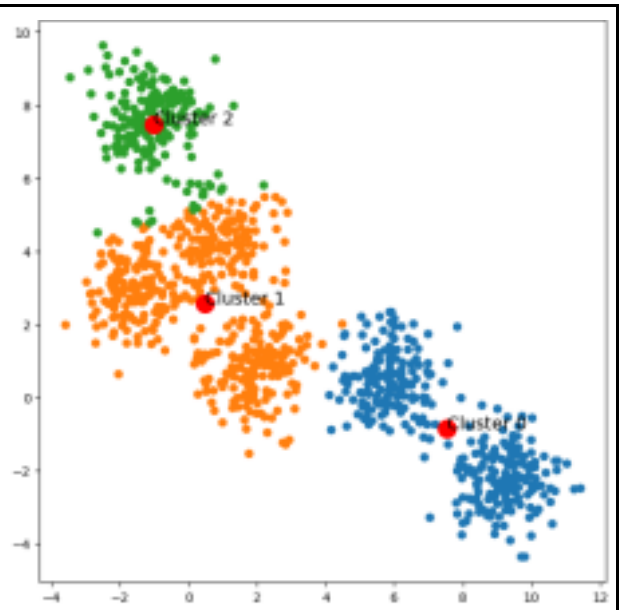
Вхідні дані



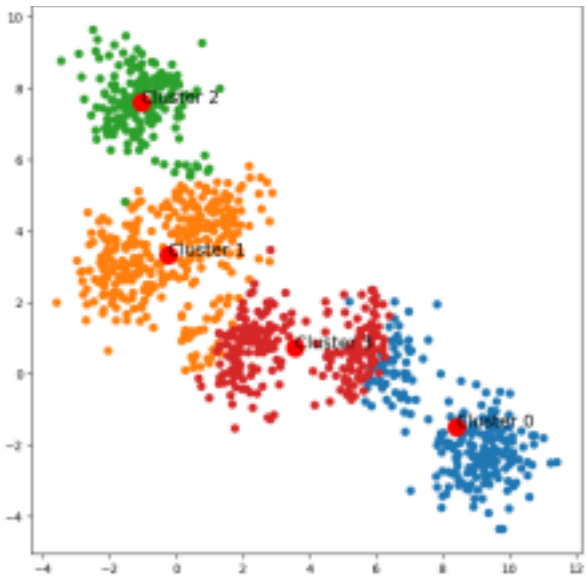
Вибір першого центру



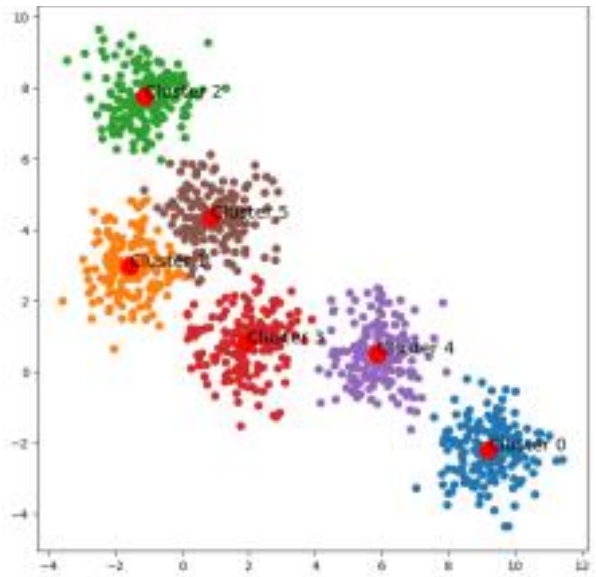
Наступний розрахунок центру



Проміжна ітерація (1)



Проміжна ітерація (2)



Остання ітерація

Підсумки виконання

```
Loser = 0
Num of Elements in Clusers: [165 163 168 171 167 166]
Loser = 0, reach the desired num of clusters
Stop at Epoch 22
```

Методи побудови систем рекомендацій

Механізм рекомендацій — це клас машинного навчання, який пропонує відповідні до його вподобань пропозиції клієнту. Оскільки, тепер системи на яких вибираєте товари чи послуги збирають достатньо інформації про користувачі на основі яких можна спробувати автоматизувати генерацію рекомендацій на основі вашої історії пошуку, історії переглядів або історії покупок.

Система рекомендацій допомагає організації створювати лояльних клієнтів і зміцнювати довіру до бажаних продуктів і послуг. Сучасні системи рекомендацій можуть також генерувати рекомендації навіть для нових клієнтів, які відвідали сайт уперше. Вони, зокрема, рекомендують продукти, які зараз є трендовими або високо оціненими, а також можуть рекомендувати продукти, які приносять максимальний прибуток компанії.

Система рекомендацій зазвичай створюється за допомогою 3 методів: фільтрації на основі вмісту, фільтрації на основі співпраці та комбінації обох.

1) Фільтрування на основі вмісту

Алгоритм рекомендує продукт, схожий на той, який використовувався під час перегляду. Простими словами, у цьому алгоритмі ми намагаємося знайти схожий предмет. Наприклад, людина рекомендує товари які схожі теги та схожі категорії. Тобто, тільки вміст виглядає схожим і не фокусується більше на людині, яка це дивиться. Лише він рекомендує продукт, який має найвищу оцінку

на основі попередніх уподобань.

2) Фільтрування на основі співпраці

Рекомендовані системи фільтрації на основі спільної роботи базуються на попередніх взаємодіях користувачів і цільових елементів. Простими словами, ми намагаємося шукати схожих клієнтів і пропонувати продукти на основі того, що вибрав він чи користувач зі схожими характеристиками з точки зору системи. Розберемося на прикладі. X і Y є двома схожими користувачами, і користувач X переглянув фільм A, B і C. Якщо користувач Y переглянув фільм B, C і D, тоді ми будемо рекомендувати фільм A користувачеві Y і фільм D користувачеві X.

3) Гібридний метод фільтрації

В основному це поєднання обох вищевказаних методів. Це доволі складна модель, яка рекомендує продукт на основі історії користувача, а також на основі подібних користувачів.

Побудова системи рекомендації книг

Система рекомендацій книг – це тип системи рекомендацій, коли ми маємо рекомендувати схожі книги читачеві на основі його інтересу. Система рекомендацій книг використовується онлайн-сайтами, які надають електронні книги.

Тут використаємо метод фільтрації на основі співпраці, щоб побудувати систему рекомендацій книг. Набір для подальшої роботи даних можна завантажити тут.

Опис набору даних

У наборі даних є 3 файли, отримані з деяких веб-сайтів з продажу книг.

Books – перший містить усю інформацію, пов’язану з книгами, як-от автор, назва, рік видання тощо.

Users – другий файл містить інформацію про зареєстрованого користувача, як-от ідентифікатор користувача, місцезнаходження.

Ratigns – рейтинги містять інформацію про те, який користувач скільки оцінив яку книгу.

Попередня обробка даних

У файлі, який містить книги наявні додаткові стовпці, які не потрібні для завдання, наприклад URL-адреси зображень. Також варто перейменувати стовпці кожного файлу, оскільки ім’я стовпця містить пробіли та великі літери, щоб зробити його зручнішим у використанні.

У першому файлі рекомендується залишити тільки такі стовпці: 'ISBN', 'Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publisher' і перейменувати їх відповідно на 'isbn', 'title', 'author', 'year', 'publisher'.

У другому файлі рекомендується залишити тільки такі стовпці: 'User-ID', 'Location', 'Age' і перейменувати їх відповідно на 'user_id', 'location', 'age'.

У третьому файлі рекомендується залишити тільки такі стовпці: 'User-ID', 'Book-Rating' і перейменувати їх відповідно на 'user_id', 'rating'.

Набір даних надійний і може розглядатися як великий набір даних.

Оскільки, тут не шукатиметься подібності між користувачами чи книгами, а визначатиметься чи є користувач А, який прочитав і сподобав В1 та В2 книжки, і користувач В, який також сподобав ці дві книги, а тепер користувач А прочитав і сподобався деяку книгу В3, яку В не читає, тому ми повинні рекомендувати В3 користувачеві В. Ось що таке фільтрація на основі співпраці.

Такого можна досягти за допомогою матричної факторизації, створивши одну матрицю, де стовпці будуть користувачами, а індекси – книгами. А значення – рейтингом. Наприклад, потрібно створити зведену таблицю (pivot table).

Перед тим як приступити до побудови зведеної таблиці варто зауважити, що якщо взяти всі книги та всіх користувачів для моделювання це може створити певні проблеми. Отже, нам потрібно зменшити кількість користувачів і книг, тому що не варто розглядати користувача, який лише зареєструвався на веб-сайті або прочитав лише одну чи дві книги. На такого користувача варто покладатися, щоб рекомендувати книги іншим, тому потрібно отримувати знання з даних з достатнім рівнем достовірності. Отже, для початку, потрібно обмежити цю кількість і взяти користувачів, які **оцінили принаймні 200 книг, а також обмежити книги, і візяти лише ті книги, які отримали принаймні 50 оцінок від користувача**.

Після формування нового набору даних для моделювання потрібно:

1. використати алгоритм найближчих сусідів (K-nearest neighbors), який використовується для кластеризації на основі евклідової відстані.
2. використати розоблений CentNN алгоритм.

Для кожного з двох алгоритмів потрібно вибрати **10** різних книг для яких показати 10 рекомендованих книг. Також, потрібно зменшити обмеження і взяти користувачів, як і **оцінили принаймні 20, 50, 100, 150 книг, і взяти лише ті книги, які отримали принаймні 10, 25, 50 оцінок від користувача**, та поглянути на потенційні зміни в рекомендаціях.

Джерела:

1. https://ieeexplore.ieee.org/document/839021
2. https://www.researchgate.net/publication/288767369_Centroid_neural_network_based_clustering_technique_using_competitive_learning
3. http://vigir.missouri.edu/~gdesouza/Research/Conference_CDs/IEEE_SMC_2009/PDFs/33.pdf
4. http://techlab.bu.edu/files/resources/articles_tt/Lin-Yu_2003.pdf
5. https://fulmanski.pl/zajecia/seminarium_katedry/tenn/TymoszczukPrace/old/1.pdf