Міністерство освіти і науки
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
"КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ"
Факультет інформатики
Кафедра інформатики

Курсова робота на тему :

"Кластеризація даних"

Керівник курсової роботи
доцент Глибовець А.М.

_____

(підпис)

____ травня 2016 р.

Виконав студент 1 року навчання
Каплунський Я.О.

Київ 2016

ЗМІСТ

Стр.

ВСТУП 4

РОЗДІЛ 1 Постановка задачі 5

РОЗДІЛ 2 Алгоритми кластеризації 8
2.1 Статистичні 8
2.1.1 EM - алгоритм 8
2.1.2 k - mean алгоритм 9
2.2 Ієрархічні 10
2.2.1 Формула Ланса-Уільямса 11
2.3 Графові 14
2.3.1 Алгоритм знаходження зв'язних компонентів 14
2.3.2 Алгоритм знаходження найкоротшого незамкненого шляху 15

РОЗДІЛ 3 Практична частина 17
ВИСНОВКИ 20
ДОДАТОК 22

ВСТУП

Аналіз даних на сьогоднішній день лежить в основі багатьох прикладних програм та сервісів. Кластерний аналіз допомагає згрупувати інформацію за спільними ознаками. Інтуєтивно зрозуміло, що об'єкти з одного кластеру мають більшу міру схожості ніж з об'єктом з іншого. Дуже важливо розрізняти кластеризацію(unsupervised classification) і дискретний аналіз(supervised classification). В навчані з вчителем ми оперуємо з розміченою колекцією документів. Тобто експерт завчасно підготував дані до подальшої обробки, вказавши вагові коефіцієнти для важливих слів чи відніс їх наперед заданої категорії. Ця розмічена колекція документів дозволяє моделі класифікувати нові дані та виділяти нові ознаки для класифікації. У випадку кластеризації проблема полягає у групуванні нерозміченої колекції документів в кластери. Програмі доведеться самостійно визначити, які дані є важливими, а які - ні.

Кластеризація корисна в багатьох прикладних задачах: виділення ознак, групуванні, прийнятті рішень, пошуку інформації, сегментуванні зображення та в класифікації ознак.[2]

В цій роботі ми розглянемо основні алгорими кластеризації даних, а також побудуємо систему, яка буде кластеризувати текстову інформацію для організації пошукової системи для форуму.

РОЗДІЛ 1

Постановка задачі

Задача кластеризації (або навчання без вчителя) полягає в наступному. Нехай є вибірка $X^l = \{x_1, ..., x_n\} \subset X$ і функція відстанні між об'єктами $\varrho(x, x')$. Потрібно розбити вибірку на підмножини, що не перетинаються, які будемо називати кластерами, так, щоб кожний кластер складався лише з об'єктів, які близькі за метрикою ϱ, а об'єкти різних кластрерів суттєво відрізнялися. При чому кожному об'єкту $x_i \in X^l$ приствоюється мітка (номер) кластера y_i. [1]

Алгоритм кластризації - це функція $a : X \to Y$, яка будь - якому $x \in X$ ставить у відповідність мітку кластера $y \in Y$. Множина міток Y в деяких випадках відома наперед, але частіше ставиться задача визначити оптимальне число кластерів, з точки зору того чи іншого критерію якості кластеризації. [1]

Рішення задачі кластеризації принципово неоднозначно, і цьому є декілька причин. По-перше, не існує однозначно найкращого критерію якості кластеризації. Відомий цілий ряд досить розумних критеріїв, а також багато алгоритмів, які не мають чітко вираженого критерію, але роблять розумну кластеризацію "по побудові". Всі вони можуть давати різні результати. По-друге, кількість кластерів, як правило, заздалегідь невідомо і визначаються вони відповідно до деяких суб'єктивних критеріїв. По-третє, результат кластеризації істотно залежить від метрики ϱ, вибір якої, як правило, також суб'єктивний і визначається експертом.

Цілі кластеризації можуть бути різними і залежати від особливості прикладної задачі:

- Зрозуміти структуру множини об'єктів X^l, розбивши на групи схожих об'єктів. Спростити подальшу обробку даних та прийняття рішень, працюючи з кожним кластером окремо(стратегія "Розділяй та владарюй").

- Зменшити об'єм даних в випадку надвеликої вибірки X^l, залишивши по одному найбільш типовому представнику з кожного кластера.

- Виділити нетипові об'єкти, які не підходять жодному кластеру. Цю задачу називають однокласовою класифікацією, знаходженням нетиповості(novely detection).

- Побудувати ієрархію множини об'єктів

В першому випадку число кластерів намагаються зробити меншим. У другому випадку важливіше забезпечити високу міру схожості об'єктів всередині кожного кластера, а кластерів може бути будь-яка кількість. В третьому випадку найбільший інтерес представляють окремі об'єкти, що не вписуються ні в один із кластерів.

У всіх цих випадках може використовуватись ієрархічна кластеризація, коли великі кластери розділяються на менші, а ті в свою чергу на ще менші. Такі задачі називаються задачами таксономії(taxonomy). Результатом таксономії є не просте розбиття на множини об'єктів на кластери, а деревоподібна ієрархічна структура. Замість номера кластера об'єкт характеризується перечисленням всіх кластерів, яким вони належать, від великого до малого. Класичним прикладом таксономії на основі схожості є систематизація живих істот, запропонована Карлом Лінеєм в середині XVII століття. В сучасному вигляді біологічна ієрархія має близько 30 рівнів 6 з яким вважається основними: царство, тип, клас, сімейство, рід, вид. Таксономія будується в багатьох областях знань, щоб упорядкувати інформацію про велику кількість об'єктів.

Кожний метод кластеризації має свої обмеження та може виділяти лише деякі типи структур. Саме поняття "тип кластерної структури "залежить від методу яким вони були знайдені та не має чіткого формулювання.

Розглянемо основні типи кластерних структур:

Рис. 1.1: Стрічкові

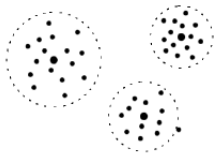


Рис. 1.2: З центром

Рис. 1.3: З'єднані перемичками

Рис. 1.4: Накладені на шум

Рис. 1.5: Ті що перетинаються

Рис. 1.6: Об'єднані не за схожістю, а за іншими ознаками

Рис. 1.7: Кластерів не існує

РОЗДІЛ 2

Алгоритми кластеризації

2.1 Статистичні

Статистичні алгоритми побудовані на припущенні, що кластери непогано описуються деяким сімейством ймовірнісних розподілів. Тоді задача кластеризації зводиться до поділу розподілів за кінцевою вибіркою.

2.1.1 EM - алгоритм

Гіпотеза(про природу випадковості даних). Об'єкти вибірки X^l обираються випадковим чином відповідно до їхнього розподілу описуючи розподіл:

$$p(x) = \sum_{y \in Y} w_y p_y(x), \ \sum_{y \in Y} w_y = 1,$$

де $p_y(x)$ - функція густини розподілу кластера y, w_y - невідома апріорна ймовірність появи об'єктів з кластера y. Зазвичай коли конкретизують вид розподілу $p_y(x)$, частіше за все обирають сферичні розподіли. Для того, щоб нам було зручно нормувати наші клстери, ми припускаємо, що наші кластери мають вигляд еліпсів, які витягнуті вздовж осі x або y. При такому підході оптимальне нормування обирається самим алгоритмом кластеризації, окремо для кожного кластера.[1]

Гіпотеза(про форму кластерів). Об'єкти описуються за допомогою n числових ознак $f_1(x), ..., f_n(x), X = R^n$. Кожний кластер $y \in Y$ описується n-мірним вектором та гаусівською густиною $p_y(x) = N(x; \mu_y, \sum_y)$ з центром $\mu_y = (\mu_{y1}, ..., \mu_{yn})$ і діагональною матрицею коваріації $\sum_y = diag(\delta^2_{y1}, ..., \delta^2_{yn})$.

EM - алгоритм

- E - крок (expectation)
  $g_{iy} = P(y|x_i) = \dfrac{w_y p_y(x_i)}{\sum_{z \in Y} w_y p_z(x_i)}, y \in Y, i = 1, ..., l;$

- M - крок (maximization):
  $w_y = \frac{1}{l}\sum_{i=1}^{l} g_{iy}, y \in Y;$
  $\mu_y i = \frac{1}{l w_y}\sum_{i=1}^{l} g_{iy} f_j(x_i), y \in Y, j = 1, ..., n;$
  $\delta_{yj}^2 = \frac{1}{l w_y}\sum_{i=1}^{l} g_{iy}(f_j(x_i) - \mu_{yj})^2, y \in Y, j = 1, ..., n;$
- $y_i = arg\max_{y \in Y} g_{iy}, i = 1, ..., l;$

2.1.2 k - mean алгоритм

Даний алгоритм є спрощеною формою EM - алгоритм. Головна відмінність полягає в тому, що в EM-алгоритмі кожний об'єкт x_i, розподіляється по всім кластерам з ймовірністю $g_{iy} = P\{y_i = y\}$. В алгоримі k-середніх(k-mean) кожний об'єкт закріплюється за одним кластером.

k-mean алгоитрм:

- Дослідник визначає кількість кластерів, що необхідно утворити
- Випадковим чином обирається k спостережень, які на цьому кроці вважаються центрами кластерів
- Кожне спостереження x_i «приписується» до одного з n кластерів, відстань до якого найкоротша Аналог E - кроку:
  $y_i = arg\min_{y \in Y} \varrho(x_i, \mu_y), i = 1, .., l;$
- Розраховується новий центр кожного кластера як елемент, ознаки якого розраховуються як середнє арифметичне ознак об'єктів, що входять у цей кластер Аналог M - кроку:

$$\mu_{yj} = \frac{\sum_{i=1}^{l}[y_i = y] f_j(x_i)}{\sum_{i=1}^{l}[y_i = y]}, y \in Y, j = 1, ..., n;$$

- Відбувається така кількість ітерацій (повторюються кроки 3-4), поки кластерні центри не стануть стійкими (тобто при кожній ітерації в

кожному кластері опинятимуться одні й ті самі об'єкти), дисперсія всередині кластера буде мінімізована, а між кластерами — максимізована

Недоліки алгоритму:

- Результат класифікації сильно залежить від випадкових початкових позицій кластерних центрів
- Алгоритм чутливий до викидів, які можуть викривлювати середнє
- Кількість кластерів повинна бути заздалегідь визначена дослідником

Існує два "канонічних варіанта"k-mean. Варіант Болла-Холла [4, ст. 110] ми описали попередньо. Варіант МакКіна [4, ст. 98] відрізняється тим, що кроки 3 та 4 виконуються в середині одного циклу по об'єктам вибірки. Коли знаходиться об'єкт, що переходить з одного кластера в інший, центри обох кластерів переобчислюються.

K-mean алгоритм дуже чутливий до вибірки початкових наближень центрів. Випадкова ініціалізація на першому кроці може привести до поганої кластеризації. Для формування початкового наближення краще за все виділити k найбільш віддалених точок вибірки: перші дві точки виділяються по максимуму всіх попарних відстанней, кожна наступна точка обирається так, щоб відстань від неї до найближчої уже виділеної була максимальною.

Кластеризація може виявитись некорекною в тому випадку, якщо спочатку не вгадати правильну кількість кластерів. Зазвичай рекомендують - провести кластеризацію при різних значеннях k і обрати те, при якому різко покращується кластеризація по заданому функціоналу.

2.2 Ієрархічні

Ієрархічні алгоритми кластеризації, які також називають таксономією, будують не одне розбиття вибірки на класи, що не перетинаються, а систему вкладених розбиттів. Результатом таксонімії зазвичай представляють у

вигляді дендрограми. Класичним прикладом такого дерева є є ієрархічна класифікація тварин та рослин.

Ієрархічні алгоритми кластеризації розділяють на два основних типи. Низхідні алгоритми розбивають вибірку на все більш і більш дрібні кластери. Більше поширені агломеративні або висхідні алгоритми, в яких об'єкти об'єднуються у все більш і більш великі кластери.

2.2.1 Формула Ланса-Уільямса

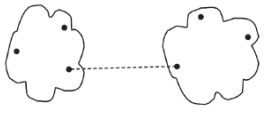
Спочатку кожний об'єкт представляє окремий кластер. Для одноелементних кластерів функція відстані визначається наступним чином:
$R(\{x\},\{x'\}) = \varrho(x,x')$.

Потім запускається процес злиття. На кожній ітерації замість пари найближчих кластерів U та V створюється новий кластер $W = U \cup V$. Відстань від нового кластера W до будь-якого кластера S обчислюється по відстаннях $R(U,V)$, $R(U,S)$ та $R(V,S)$, які на момент обчислення мають бути відомими:
$R(U \cup V, S) = \alpha_u R(U,S) + \alpha_v R(V,S) + \beta R(U,V) + \gamma |R(U,S) - R(V,S)|$,
де $\alpha_u, \alpha_v, \beta, \gamma$ - числоаві параметри. Ця універсальна формула узагальнює майже усі способи визначення відстаней між кластерами.

На практиці використовують наступні способи обчислення відстаней $R(W,S)$ між кластерами W та S. Для кожного з них існує доведення відповідності формулі Ланса-Уільямса при заданих параметрах[5].

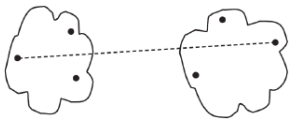
- Відстань ближнього сусіда

$$R^n(W,S) = \min_{w \in W, s \in S} \varrho(w,s); \qquad \alpha_u = \alpha_v = \frac{1}{2}, \beta = 0, \gamma = -\frac{1}{2}$$



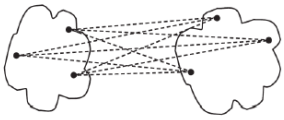
- Відстань дальнього сусіда

$$R^f(W,S) = \max_{w\in W, s\in S} \varrho(w,s); \qquad \alpha_u = \alpha_v = \frac{1}{2}, \beta = 0, \gamma = -\frac{1}{2}$$



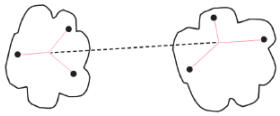
- Середня відстань

$$R^c(W,S) = \frac{1}{|W||S|}\sum_{w\in W}\sum_{s\in S}\varrho(w,s); \qquad \alpha_u = \frac{|U|}{|W|}\alpha_v = \frac{|V|}{|W|}, \beta = \gamma = 0$$



- Відстань між центрами

$$R^m(W,S) = \varrho^2(\sum_{w\in W}\frac{w}{|W|}\sum_{s\in S}\frac{s}{|S|}) \quad \alpha_u = \frac{|U|}{|W|}\alpha_v = \frac{|V|}{|W|}, \beta = -\alpha_u\alpha_v, \gamma = 0$$



- Відстань Уорда

$$R^u(W,S) = \frac{|S||W|}{|S|+|W|}\varrho^2(\sum_{w\in W}\frac{w}{|W|}\sum_{s\in S}\frac{s}{|S|})$$

$$\alpha_u = \frac{|S|+|U|}{|S|+|W|}\alpha_v = \frac{|S|+|V|}{|S|+|W|}, \beta = \frac{-|S|}{|S|+|W|}, \gamma = 0$$

Алгоритм Ланса-Уільямса:

- Спочатку усі кластери одноелементні $t := 1; C_t = \big\{\{x_1\}, ..., \{x_l\}\big\}$;
- Для усіх $t = 2, ..., l$(t - номер ітерації)

- Знайти в C_{t-1} два найближчиш кластера:
  $(U, V) = arg\min_{U \neq V} R(U, V)$;
  $R_t = R(U, V)$;

- Злити в один кластер:
  $W = U \cup V$;
  $C_t = C_{t-1} \cup W \; U, V$

- Для всіх $S \in C_t$
  обрахувати $R(W, S)$ по формулі Ланса-Уільямса.

Деякі відстанні мають характеристику розтягу. По мірі того, як кластер росте, то відстань від нього до других кластерів також росте, так ніби простір навколо кластера розтягується. Характеристика розтягу є бажаною, так як вона сприяє більш чіткому відділеню кластерів. З іншого боку, при сильно великих розтягах можна знайти кластери там, де їх не було. Розтягуючими є відстані: R^f , R^u.

Деякі відстані, навпаки, характерезуються стискання. По мірі того, як кластерна відстань зростає від інших кластерів вона зменшується, і здається, що простір навколо кластера стискається. Природня кластеризація при цьому "розмивається". Відстань ближнього сусіда R^n є стискаючим.

Характеристика стискання та розтягу визначається через відношення $R_t/\varrho(\mu_U, \mu_V)$, де $R_t = R(U, V)$ - відстань між ближніми кластерами, що об'єднуються на кроці t, μ_U, μ_V - центри цих кластерів. Якщо це відношення на кожному кроці більше одиниці, то R є розтягуючим, якщо менше - стискаючим. Є такі відношення які є ні стисканням ні розтягом R^c, R^m. Про них говорять, що вони зберігають метрику простору.

На практиці використовують гнучку відстань, яка являє собою компроміс між методами ближнього сусіда, дальнього сусіда та середньої відстані. Вона визначається одним параметром β замість чотирьох:
$\alpha_U = \alpha_V = (1 - \beta)/2, \gamma = 0, \beta < 1.$
Гнучка відстань є стискаючим при $\beta > 0$ і розтягуючим при $\beta < 0$. Стандартна рекомендація : $\beta = -0.25$.[4]

Достатки та недоліки: Точної відповіді на питання, який алгоритм кластеризації підходить найкраще, не існує. Кожна, із вище перерахованих відстаней, має свої переваги та недоліки в контексті конкретної задачі. [1]

Метод найближчого сусіда характеризується ланцюговим ефектом, коли незалежно від форми кластера до нього приєднуються найближчі об'єкти. В деяких випадках це приводить до того, що кластири "відрощують щупальці". В залежності від задачі ця характеристика може бути як корисною так і поганою. Цей метод найкраще підходить для виділеня стрічкових кластерів.

Метод дальнього сусіда ланцюгового ефекту не має, але на ранньому етапі він може об'єднати несхожі групи.

Відстань між центрами мас не монотонне та не руктивне, тому воно на практиці майже не застосовується, хоча інтуєтивно воно здається золотою серединою у виборі алгоритму.

Метод Уорда експерементально виявився найкращим з усіх перерахованих методів. Він найкраще розбиває дані на інтуєтивні кластери.[4]

2.3 Графові

Великий клас алгоритмів кластеризації базується на предствленні вибірки у вигляді графа. Вершина графа відповідає об'єкту вибірки, а ребара - попарні відстані між об'єктами $\varrho_{ij} = \varrho(x_i, x_j)$.

Перевагою графових алгоритмів кластеризації є наглядність, простота реалізації, можливість оптимізовувати алгоритм, спираючись на прості геометричні міркування.

2.3.1 Алгоритм знаходження зв'язних компонентів

Задається параметр R і в графі видаляються ребра (i, j), для яких $\varrho_{ij} > R$. З'єднаними залишаються лиш найближчі пари об'єктів. Ідея алгоритма полягає в тому, щоб підібрати таке значення $R \in [min\ \varrho_{ij}, max\ \varrho_{ij}]$,

при якому граф розбивається на декілька компонентів зв'язності. Знайдені зв'язні компоненти - і є кластери. [3]

Компонентою зв'язності графа називається підмножина його вершин, в якій будь - які дві вершини можна з'єднати шляхом. Для пошуку зв'язних компонентів можна використовувати усім відомий BFS (алгоритм Дейкстри) або DFS.

Для вибору параметра R рекомендують побудувати гістограму розподілу попарних відстаней ϱ_{ij}. В задачах з вираженою кластерною структурою ця гістограма має два чітких піка: зона найбільших внутрішніх міжкласових відстаней та найменших внутрішніх міжкласових відстаней. Параметр R задається як відстань, що відповідає точці мінімума між цими піками.

Недоліки алгориму:

- Обмежена область використання. Алгоритм знаходження зв'язних компонентів найбільше підходить для виділення кластерів класу концентрації чи стрічок. Наявність розрідженого фоне чи "вузьких перемичок"між кластерами призведе до некоректної кластеризації.

- Погана керованість числом кластерів. Для багатьох задач зручніше не задавати параметр R, а вказувати кількість кластерів. Керувати числом кластерів за допомогою параметра R важко. Доводиться вирішувати цю задачу декілька раз при різних значеннях параметра R,що негативно впливає на чис вирішення задачі.

2.3.2 Алгоритм знаходження найкоротшого незамкненого шляху

Алгоритм будує гарф з $l-1$ ребер так, щоб вони з'єднували всі l точок і мали найменшу сумарну довжину. Такий граф називається найкоротшим незамненем шляхом, або остове дерево.

На відмінно від попереднього алгоритму, тут число кластерів K задається як вхідний параметр. Недоліки алгориму:

- Обмежена область використання. Наявність розрідженого фоне чи

"вузьких перемичок"між кластерами призведе до некоректної кластеризації.

- Для побудови найкоротшого незамкнутого шляху потрібно виконати $O(l^3)$ операцій.

РОЗДІЛ 3

Практична частина

В якості практичної частини розглянемо побудову пошукової системи для форуму. Коли користувач буде шукати на нашому форумі інформацію, наша пошукова система, скоріш за все видасть йому правильну відповідь на постевлене запитання, якщо така існує - в іншому випадку вона виведе найбільш схожу відповідь.

Для вирішення цієї задачі ми і будемо використовувати кластеризацію. Для реалізації ціє ідеї нам потрібно буде виконати наступні дії:

1. Зробити лексичний аналіз тексту та розбити його на лексеми
2. Відкинути слова, що зустрічаються дуже часто і не допомагають знаходити релевантні повідомлення
3. Відкинути слова, що зустрічаються дуже рідко, і наврядчи зустрінуться в наступних повідемленнях
4. Обрахувати TF-IDF з урахуванням усього корпусу тексту.
5. Виділити характерні ознаки з кожного повідомлення і зберегти його в вигляді асоційованого масиву з повідомленням
6. Побудувати кластери з цих векторів
7. Визначити кластер до якого відноситься повідомлення
8. Обрати з цього кластеру кілька повідомлень, що маю достатню міру схожості з повідомленням. Це підвищить різноманіття видачі.

Для реалізації програми ми будемо використовувати бібліотеку SciKit.

Для перетворення тексту в вектор числових значення будемо використовувати інвертований індекс. При використанні цього підходу перше з чим ми стикаємось - попередня обробка даних. Для підрахунку використаємо метод CountVectorizer з бібліотеки SciKit.

```
CountVectorizer(analyzer= 'word', binary= False,
 charset = None,
charset_error= None, decode_error='strict',
dtype= <class 'numpy.int64'>, encoding = 'utf-8',
input='content',
lowercase= True, max_df=1.0, max_features=None, min_df=1,
ngram_range=(1, 1), preprocessor=None, stop_words=None,
strip_accents= None, token_pattern='(?u)\\b\\w\\w+\\b',
tokenizer=None, vocabulary= None)
```

Для обрахунку міри схожості між векторами будемо обраховувати, як відстань між нормованими векторами.

```
>>> def dist_norm(v1, v2):
v1_normalized = v1/sp.linalg.norm(v1.toarray())
v2_normalized = v2/sp.linalg.norm(v2.toarray())
delta = v1_normalized - v2_normalized
return sp.linalg.norm(delta.toarray())
```

Наступним кроком для підготовки тексту є видалення так званих "стоп-слів". Для цього нам необхідно передати відповідний параметр в наш CounteVectorizer.

```
>>> vectorizer = CountVectorizer(min_df = 1, stop_words='englis
```

Якщо параметр stop _ words = english, то в цей список буде складатися 318 англійських слів.

```
>>> sorted(vectorizer.get_stop_words()) (0:20)
['a','about','above','across','after','afterwards','again',
'against','all','almost','alone','along','already','also',
'although','always','am','among','amongst','amoungst']
```

Наступним етапом обробки даних буде стемінг. Стемінг - це процес скорочення слова до основи шляхом відкидання допоміжних частин, таких як закінчення чи суфікс. Результати стемінгу іноді дуже схожі на визначен-

ня кореня слова, але його алгоритми базуються на інших принципах. Тому слово після обробки алгоритмом стемінгу (стематизації) може відрізнятися від морфологічного кореня слова.

Для роботи з стемером ми будемо використовувати бібліотеку NLTK. З цього пакету будемо використовувати SnowballStemmer.

Також для узагальнення концепції стоп-слів, для часто вживаних слів, ми будемо використовувати TF-IDF(term frequency - inverse document frequency). Для цього ми використаємо TdfidVectorizer з пакету SciKit.

Для перевірки реалізації нашої системи будемо використовувати один із стандартних, для машинного навчання, набір даних 20newsgroup, який складається з 18826 повідомлень та із 20 новинних груп за різними тематиками. Увесь код реалізації можна знайти у додатку до даної роботи.

ВИСНОВКИ

У цій роботі ми розглянули основні алгоритми кластеризації інформації, а також побудували систему, яка здатна обробити зашумлений текст в компактне векторне представлення, яке ми вдало кластеризували. За допомогою кластеризації наша система може злегкістю видавати релевантні відповіді користувачу, що ми і прагнули зробити. Також варто відмітити те, що на сьогоднішній день існує багато допоміжних засобів для роботи з даними. Так у цій роботі були використані бібліотека Sci-learn в якій є реалізовані напопулярніші методи обробки даних на мові python. Також в цьому пакеті існує багато зручних інструментів для роботи зі стандартними, для машинного навчання, вибіроками даних, наприклад http://mlcomp.org.

Бібліографія

[1] Воронцов К.В. Алгоритмы кластеризации и многомерного шкалирования. Курс лекцій. МГУ, 2007.

[2] Luis P. C., Willi R. Building Machine Learning Systems With Python, Second Edition - PACKT Publishing, 2015

[3] Jain A., Murty M., Flynn P. Data Clustering: A Review. ACM Computing Surveys. 1999. Vol. 31, no. 3.

[4] Мандель І. Д. Кластерний аналіз - М.: Фінанси та Статистика, 1998

[5] Уілліамс У. Т., Ланс Д. Н. Методи ієрархічної класифікації Статистичні методи для ЕОМ Під ред. М. Б. Малюта. М .: Наука, 1986. С. 269-301.

ДОДАТОК

https://github.com/kapyar/course