Групиране на документи по общи признаци

Автоматизираното групиране на документи е основна задача, която може да се разглежда както като *допълнение към търсенето* на подобни документи, така и като *алтернативен начин за търсене*.

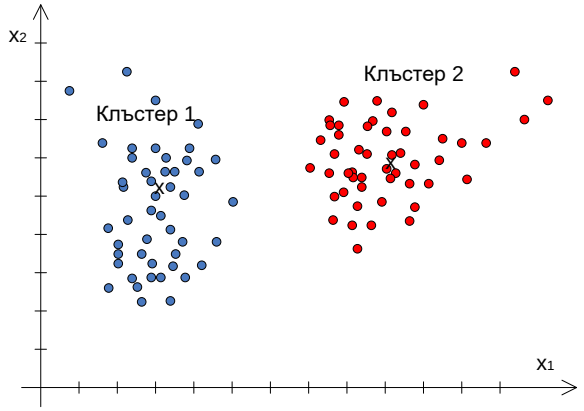
Съществуват редица ситуации, в които работата на потребителя може значително да се улесни, ако върнатите от системата резултати са допълнително групирани по определени признаци. Например, ако потребителят търси документи свързани с „време“, системата вероятно ще открие и върне резултати, свързани с „метеорологично време“, „астрономическо време (часово)“ и „историческо време“. Възможно е обаче потребителят да е заинтересуван само в едно от значенията на думата „време“. Но ако резултатите са смесени, той/тя трябва да ги прегледа всичките. Много по-удобно би било, ако върнатите резултати са групирани допълнително по съответните значения на критерия за търсене. В този случай, когато групирането се извършва върху върнатите от системата резултати, тогава то се счита за допълнение към търсенето, което има за цел да повиши както адекватността на получените резултати, така и удобството за работа на потребителя.

Възможно е търсенето да се организира така, че изобщо да не е необходимо потребителят да въвежда заявка за търсене. Вместо това, документите могат предварително да се групират по общи признаци (на множество нива, ако е необходимо), групите да се представят на потребителя, той да ги прегледа и веднага да се насочи към съответната група, която го интересува. На английски това действие би могло да се класифицира като „searching by browsing“ и представлява алтернативен подход за търсене.

Извън търсенето, групирането на документите по общи признаци е изключително полезно за реализация на автоматизирана подредба и категоризация на документите в колекцията. Процесът може да се автоматизира с помощта на т.нар. ***клъстерен анализ***. Той цели да групира n на брой обекти в k (k>1) на брой групи, наречени *клъстери*, на базата на p (p>0) на брой признаци (характеристики). Характеристиките могат да бъдат явно посочени от потребителите (например ключови думи) или неявно извлечени от самите документи чрез анализ на съдържанието им (например векторите от уникални думи в документа).

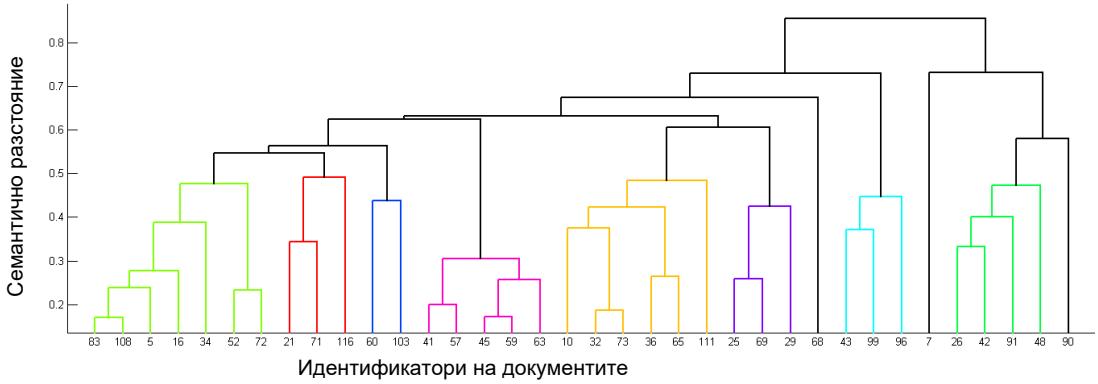
Методите и алгоритмите за клъстеризация могат да се разделят на 2 големи групи:

- ***Плоска клъстеризация*** - обектите се групират в плоски клъстери (фиг. 1). *Знае се кой обект на коя група принадлежи, но не и колко близки са обектите помежду си вътре в самия клъстер*. При този тип клъстеризация е необходимо *предварително да се укаже броят на клъстерите*, в които да бъдат групирани обектите.



Фигура 1. Пример за плоска клъстеризация

- ***Йерархична клъстеризация*** - обектите се групират в йерархични клъстери (фиг. 2). *Знае се не само кои обекти принадлежат на даден клъстер, но и колко близки са обектите помежду си вътре в самия клъстер*. От своя страна алгоритмите за йерархична клъстеризация могат също да се разделят на два вида – агломеративна (сливаща) и разделяща. При агломеративната клъстеризация групирането става „отдолу нагоре“, т.е. започва се с множеството отделни обекти, всеки формиращ собствен клъстер, и на всяка итерация двете най-близки групи се сливат. При разделящата клъстеризация е обратно (отгоре надолу) – започва се с един голям общ клъстер и на всяка итерация, от него се изваждат най-различните елементи. В практиката, агломеративният клъстерен анализ е по-популярен и по-често реализиран. При йерархичната клъстеризация *не е необходимо предварително да се посочва броя на клъстерите*, а алгоритъмът получава свободата да следва естественото разпределение на данните. Затова и групирането е по-точно, в сравнение с методите за плоска клъстеризация. Броят на клъстерите може да се определи на по-късен етап след анализ на генерираната *дендрограма* или чрез задаване на прагово разстояние при обединяване на клъстерите.



Фигура 2. Пример за йерархична клъстеризация. Групиране на доклади от научната конференция CompSysTech 2017 в тематични направления.
Фигурата представлява оцветена дендрограма (двоично дърво).

1. Плоска клъстеризация

Най-популярният алгоритъм за плоска клъстеризация е *k-means* (*алгоритъм на k-средните*, известен още и като *алгоритъм на k-вътрешно-групираните средни*). Лесен е за реализация и дава добри резултати. Работи по следния начин:

1. Потребителят указва *броя клъстери*, в които да бъдат разпределени обектите/документите.
2. Алгоритъмът присвоява *произволни центрове* (произволни координати) на посочения брой клъстери.
3. За всеки обект/документ:
   a. Изчислява се разстоянието от обекта до центровете на всеки клъстер. Могат да се използват различни мерки за сходство. Оригиналният алгоритъм разчита на Евклидово разстояние, но могат да се използват и други мерки.
   b. Обектът се зачислява към клъстера, до чийто център е най-близо.
   c. *Преизчисляват се координатите на центъра* на клъстера, към който е причислен обектът. Координатите се изчисляват така, че *сумата от квадратите на разстоянията* между центъра и всички обекти, принадлежащи на клъстера, *да бъде минимална.* В статистиката това е известно като метод на най-малките квадрати (МНМК) и се използва като основен критерий за оптимизация. Затова и при добавяне на нов обект, центровете се преизчисляват.

Тъй като алгоритъмът започва с произволни центрове, ако се използва за групирането на малко на брой обекти, неговата точност няма да бъде особено висока, заради произволните координати в началото. Но *с увеличаване на броя на групираните обекти, центровете непрекъснато ще се преизчисляват и координатите им ще започнат да клонят към оптималните* за дадените клъстери. Това е важно да се знае и отчита! Алгоритъмът на k-средните е представител на алгоритмите за неконтролирано обучение (unsupervised learning) от областта на машинното обучение. Т.е. колкото повече данни бъдат обработени, толкова по-точно ще бъдат групирани. Ако с този алгоритъм се планира да бъдат групирани малко на брой обекти (например 5-10), няма как да се очаква висока точност. В този случай е по-добре обектите да бъдат групирани ръчно от човек.

Алгоритъмът на k-вътрешно-групираните средни работи директно с характеристиките на обектите, а не с разстоянията между тях. Това означава, че всеки обект/документ трябва да се описва с характеристичен вектор, чийто стойности могат да бъдат:

- *Бинарни стойности,* показват наличието или отсъствието на дадена характеристика
  `objectId -> {0, 1, 0, 1, 1, 0}`
- *Реални числа*, показващи степента на приложимост на всяка характеристика или tf-idf тегла
  `objectId -> {0.75, 0, 0.25, 0.1, 0.5, 0}`
- *Семантична близост/разстояние* на текущия документ до останалите
  `objectId -> {0.65, 0.34, 0.09, 0.17, 0, 0.2}`

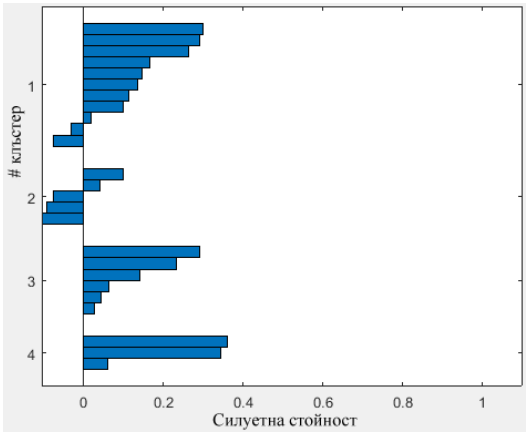
- *Ключови думи* в явен вид (т.е. разписани като стрингове)
  `objectId -> {сладко, десерт, храна}`

  Алгоритъмът не може да работи директно с думи, затова трябва множеството от думи да се преобразува до характеристичен вектор с бинарни стойности, които показват наличието или отсъствието на дадената ключова дума. Разбира се, трябва да се въведе подредба на думите, така че позициите им във всички вектори да бъдат идентични. В противен случай няма как да се разбере наличието (1) или отсъствието (0) за коя точно дума се отнася.

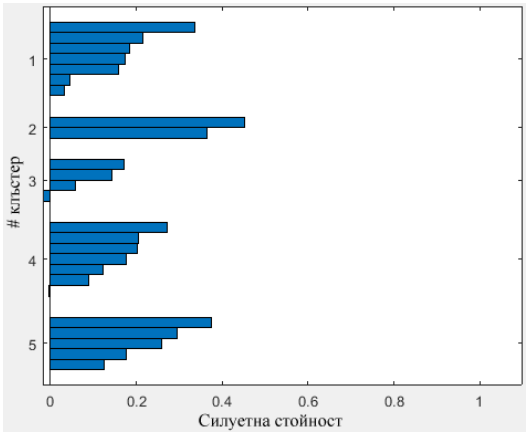
По принцип алгоритъмът не е предвиден и да работи директно с разстояния или подобия между документите, а само с техните характеристики. Но *на практика може да работи и с разстояния*, което е експериментално потвърдено от собствени изследвания. За целта, характеристичните вектори трябва да се конструират така, че да съдържат разстоянията (подобията) от даден обект до всички останали, като тук, както и при реалните числа и бинарните стойности, е важно елементите да бъдат позиционно подредени по един и същ начин във всички вектори.

Един от основните проблеми на плоската клъстеризация и съответно на алгоритъма на k-средните е, че потребителят трябва предварително да посочи броя клъстери, в които да се групират обектите. Но потребителят, по принцип, няма от къде да знае какъв би бил оптималният брой групи, който да доведе до най-точно групиране. Затова трябва да се експериментира с различен брой клъстери, резултатите да се сравнят и така да се определи най-подходящият брой. За сравнение на резултатите (при различния брой клъстери) може да се използва т.нар. *графика на силуетите*. Тя показва колко близо се намира всеки представител на клъстер до съседните клъстери. Стойността на силуета варира от 1 (обектът в клъстера е много далеч от съседните клъстери), през 0 (обектът е много близо до друг клъстер или клъстери и на практика може да се причисли и към тях) до -1 (обектът е грешно причислен към дадения клъстер, много по-близо е до друг).

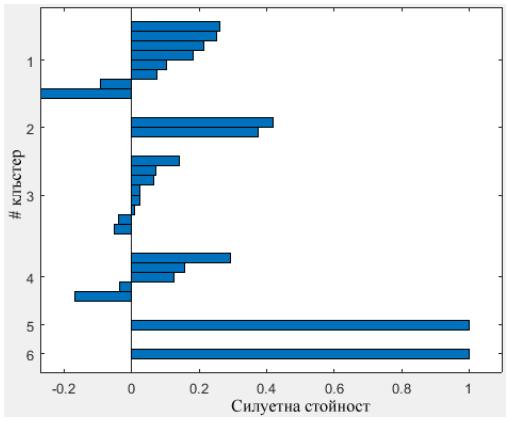
Целесъобразно е графиката на силуетите да бъде представена с пример: На групиране подлежат резултатите от анкета, попълнена от 25 човека относно техните хобита. Всеки респондент може да избере произволен брой интереси от списък с 30 предварително дефинирани такива. Целта е хората да се групират по интереси/хобита. Анкетата е проведена от студентката Деница Василева през пролетта на 2020 г. Резултатите от групирането при 4, 5 и 6 групи е представено чрез графика на силуетите на фиг. 3. Тъй като интересите на хората се описват в 30-мерното пространство, то очевидно няма как групирането да се представи с диаграма в равнината като фиг. 1.



а) при 4 клъстера



б) при 5 клъстера

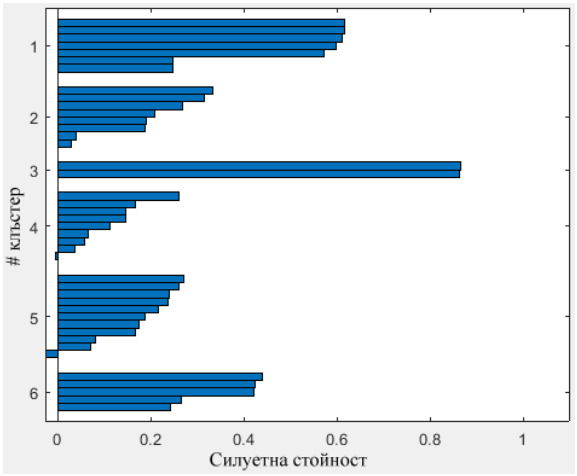


в) при 6 клъстера

Фигура 3. Графика на силуетите, показваща качеството на групиране на хора по интереси, при 4, 5 и 6 клъстера

При 4 клъстера, в групи 1, 2 и 3 има обекти, които се намират много близо до други клъстери и спокойно могат да бъдат причислени към тях, т.е. не може надеждно да се каже дали са правилно класифицирани или не. В същото време в клъстери 1 и 2 има очевидно грешно групирани обекти, като в клъстер 2, грешно групираните са дори повече от вярно причислените. Явно 4 клъстера не е подходящ брой. При 5 клъстера, резултатите са значително по-добри, като има само един документ (в група 3), който може да бъде прехвърлен към друга група. При 6 клъстера, резултатите отново се влошават. Очевидно, оптималният брой клъстери, при които групирането изглежда най-адекватно в случая е 5.

Като втори тест, с помощта на алгоритъма на k-средните са групирани 42 научни доклада от конференцията CompSysTech 2016, от всички секции без тази за електронно обучение, по тематични направления. Подобно автоматизирано групиране е изключително полезно и може да се използва като база за бързо и лесно генериране на програмата на конференцията. Според графиката на силуетите, най-добро групиране се постига при 6 клъстера (фиг. 4).



Фигура 4. Графика на силуетите, показваща качеството на групиране на 42 научни доклада по тематични направления, в 6 групи

Всеки клъстер включва определени доклади, които се идентифицират чрез техните входящи номера, които играят ролята на уникални идентификатори. Целенасочено ще бъдат посочени точните доклади, които изграждат съответните клъстери, тъй като в последствие същите 42 доклада ще бъдат групирани и чрез агломеративен йерархичен клъстерен анализ и съответно резултатите от двата типа клъстеризация ще бъдат сравнени.

Шестте групи се състоят от следните доклади:

Група 1: 7, 10, 24, 87, 89, 94, 106

Група 2: 3, 33, 67, 82, 92, 100, 119, 101

Група 3: 35, 75

Група 4: 5, 19, 23, 29, 57, 59, 68, 107, 111

Група 5: 6, 17, 20, 38, 40, 41, 64, 65, 76, 95, 121

Група 6: 12, 25, 42, 60, 103

2. Йерархична клъстеризация

За разлика от плоската клъстеризация, *при йерархичната се знае* не само кой обект/документ на коя група принадлежи, но и *каква е взаимовръзката между обектите вътре в самия клъстер*, т.е. *колко близки са те помежду си*. Това е изключително важна информация, която позволява на потребителя да прави допълнителни анализи и евентуално *да размества документите*, *да променя размерите на клъстерите*, *да разделя или слива клъстери*, както и *да подрежда документите вътре в самия клъстер*. Всичко това при плоската клъстеризация е невъзможно, защото потребителят не знае колко близки са документите помежду си вътре в самия клъстер.

Йерархичното клъстеризиране (групиране) обикновено се извършва чрез т.нар. агломеративен йерархичен клъстерен анализ (*agglomerative hierarchical clustering*), който се реализира „отдолу нагоре”. *В началото третира всички обекти като самостоятелни клъстери*, след което *на всяка стъпка слива два от съществуващите (двата най-близки) клъстери*, за да формира един по-голям (фиг. 5). Ако не бъде прекъснат или не са посочени допустими граници за размери на клъстерите, или други критерии за прекратяване на

групирането, процесът продължава докато всички клъстери бъдат обединени в един. Съществува и обратния подход „отгоре надолу“ (*divisive clustering*), при който в началото всички обекти са в един общ клъстер и алгоритъмът започва итеративно да го разделя на подгрупи. На всяка итерация най-големият клъстер се разделя на две части, докато не се достигне до състояние, при което всеки обект се намира в собствен клъстер.

Алгоритъмът за реализиране на агломеративния йерархичен клъстерен анализ е представен на фигура 5. Групирането се извършва по описания по-горе начин – отдолу нагоре. Алгоритъмът използва следните по-важни структури от данни:

`distMatrixClusters[clusterId1][clusterId2]` – матрица на разстоянията между клъстерите. Първоначално се инициализира с матрицата на разстоянията между отделните обекти, защото в началото всеки обект формира собствен клъстер.

`clusters[clusterId]` – масив, чийто индекси са идентификатори-те на клъстерите, а стойностите представляват масиви, съдържащи отделните документи, принадлежащи на съответния клъстер.

`calculateDistance()` – функция, която изчислява разстоянието между два клъстера по някой от описаните по-долу методи (най-близкия съсед, най-отдалечения съсед или средно-аритметичното разстояние).

```
-----------------------------------------------------------------------
// матрица на разстоянията между клъстерите
// distMatrixClusters[clusterId1][clusterId2]
distMatrixClusters = distMatrix; // distMatrix – разстоянията
// между документите

// clusters[clusterId] = масив от идентиф. на докум. в клъстера
// първоначално, всеки документ образува собствен клъстер
i=0;
for всеки документ с идентификатор doc_i {
   clusters[i] = doc_i; i++;
}
nbClusters = брой на документите за групиране;

do {
   // обхождат се всички клъстери (първоначално документи)
   // и се търси кои са двата на близки, за да се слеят
   // минималното разстояние между клъстерите се инициализира
   // с някаква голяма (произволна) стойност
   minDist = 1000;
   // closest_i и closest_j са идентификаторите
   // на двата най-близки клъстера
   closest_i = null;
   closest_j = null;
   // намиране на двата най-близки клъстера. за по-ефективно:
   // да се обхожда само горната дясна диагонална част
   for всеки клъстер cluster_i от distMatrixClusters  {
      for всеки друг клъстер cluster_j, различен от cluster_i {
         if (distMatrixClusters[cluster_i][cluster_j] < minDist)
            minDist = distMatrixClusters[cluster_i][cluster_j];
```

```
            closest_i = cluster_i;
            closest_j = cluster_j;
         }
      }
   }
   // след като е ясно кои са най-близките клъстери
   // closest_i и closest_j, те трябва да се слеят. За целта
   // този с по-малък номер поглъща този с по-голям номер.
   absorbing = min(closest_i, closest_j);
   absorbed = max(closest_i, closest_j);
   // сливане
   clusters[absorbing] = clusters[absorbing]+clusters[absorbed];
   // изтриване на погълнатия
   // и от clusters масива и от матрицата на разстоянията
   delete clusters[absorbed];
   delete редовете и колоните в distMatrixClusters,
                              свързани с absorbed;
   // след като е изтрит погълнатият клъстер от clusters
   // и от матрицата на разстоянията,
   // трябва да се (пре)изчисли разстоянието между
   // новия по-голям клъстер (absorbing) и всички останали
   for всеки клъстер cluster_j в distMatrixClusters,
                                          различен от absorbing {
      distMatrixClusters[absorbing][cluster_j]=calculateDistance(
              clusters[absorbing],clusters[cluster_j],distMatrix);
      distMatrixClusters[cluster_j][absorbing] =
                        distMatrixClusters[absorbing][cluster_j];
   }
   nbClusters--;
} while (nbClusters > 1)
-----------------------------------------------------------------------
```

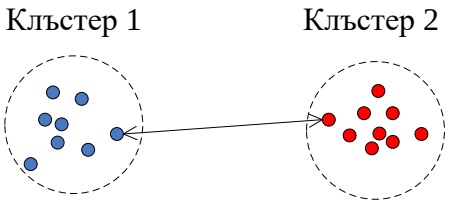
Фигура 5. Алгоритъм за реализация
на агломеративен йерархичен клъстерен анализ

При това движение отдолу-нагоре, алгоритъмът генерира *граф на свързаност* между отделните клъстери и елементите в тях. Към всяка дъга в този граф е асоциирано *тегло*, което показва *семантичната близост между клъстерите,* които свързва. По същество генерираният граф представлява *двоично дърво*, графичното представяне на което се нарича *дендрограма* (фиг. 2). Тя съдържа важна информация, чийто анализ отговаря на въпросите: кой обект с кои други да бъде групиран; в колко клъстера да бъдат разпределени обектите; как да бъдат подредени отделните клъстери.

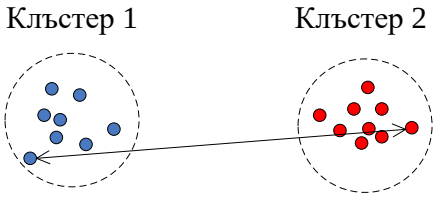
Йерархичният клъстерен анализ, за разлика от алгоритъма на k-средните, не работи директно с обектите, а с матрицата на разстоянията между тях. За да се формира тя, е необходимо да се изчислят подобията между всички обекти чрез някоя от разгледаните вече мерки за сходство. Това ще генерира матрица на подобията, която лесно се преобразува до матрица на разстоянията, като се извади от 1.

При изчисляване на разстоянието между два клъстера, могат да се използват различни методи (критерии) за свързване. Най-подходящи са методът на най-близкия съсед (*single*

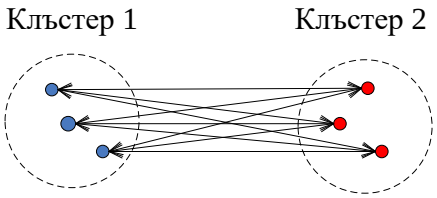
linkage clustering) – фиг. 6; методът на най-отдалечения съсед, известен още като пълно включване (*complete linkage clustering*) – фиг. 7; и методът на средно аритметичното разстояние (*average linkage clustering*) – фиг. 8, при който разстоянието се изчислява като средно-аритметично между всяка двойка обекти от двата клъстера. Задължително единият обект е от първия клъстер, а другият обект от втория. Последният е известен още и като метод на между груповото разстояние (*between groups linkage*).



Фигура 6. Метод на най-близкия съсед (single linkage clustering) при изчисляване на разстоянието между два клъстера



Фигура 7. Метод на най-отдалечения съсед / пълно включване (complete linkage clustering) при изчисляване на разстоянието между два клъстера



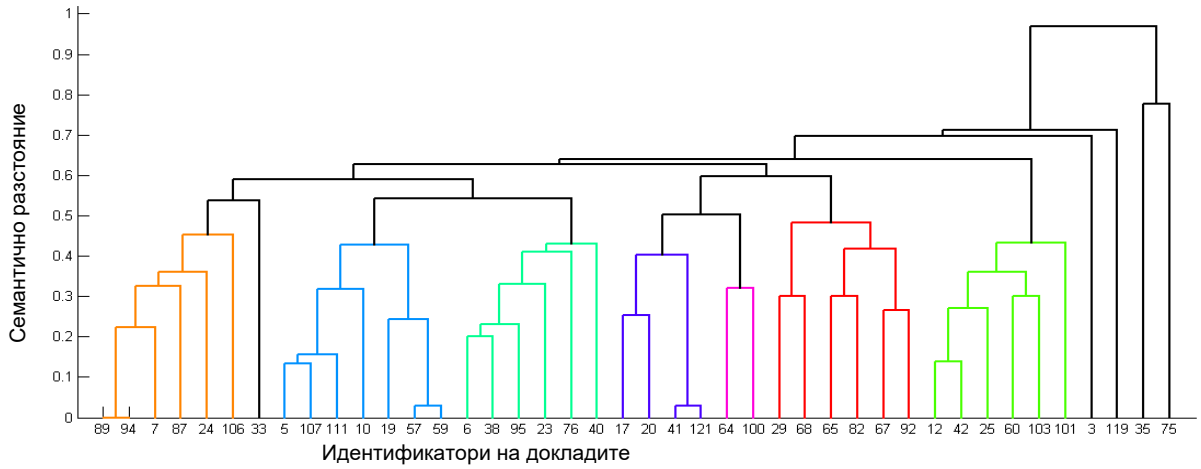
Фигура 8. Метод на средно аритметичното разстояние (average linkage clustering) при изчисляване на разстоянието между два клъстера

Кой от трите метода за свързване е най-подходящ в конкретна ситуация може лесно да се оцени чрез. т.нар. коефициент на *кофенетична корелация*. Както е видимо от дендрограмата на фиг. 8.2, всеки два обекта, или клъстера, са свързани помежду си с *връзка, притежаваща определена височина, която показва разстоянието между тях*. Тази височина се нарича *кофенетично разстояние* (*cophenetic distance*). Ако групирането е адекватно би трябвало връзките между клъстерите в двоичното дърво да съответстват на разстоянията в матрицата. От там, за да се оцени точността на избрания метод за свързване, трябва да се изчисли степента на корелация между височината на връзките (*кофенетичните разстояния*) между отделните обекти и разстоянията между тях във входната матрица. Линейният корелационен коефициент на Пиърсън е много подходящ за

целта. Колкото коефициентът на корелация е по-близък до 1, толкова по-точно е групирането, т.е. в по-голяма степен съответства на естественото деление на данните. Множество собствени експериментални изследвания, при групиране на научни доклади по тематични направления, показват, че методът на средно-аритметичното разстояние води до най-добри резултати и най-висок коефициент на кофенетична корелация.

Алгоритъмът приключва когато всички клъстери се обединят в един или когато кофенетичното разстояние между клъстерите стане по-голямо от предварително зададен праг. Не е необходимо потребителят да задава предварително броя на клъстерите, а алгоритъмът получава свободата да следва естественото разпределение на данните. Затова и групирането е по-точно, в сравнение с методите за плоска клъстеризация. Броят на клъстерите може да се определи на по-късен етап след анализ на генерираната дендрограма или чрез задаване на прагово разстояние при обединяване на клъстерите.

За илюстрация, чрез агломеративен йерархичен клъстерен анализ са групирани същите 42 научни доклада от конференцията CompSysTech 2016, които бяха групирани и чрез алгоритъма на k-средните в предходната точка. Това позволява да се направи обективно сравнение между двата подхода. При свързването на отделните клъстери е използван методът на средно-аритметичното разстояние между тях. Тъй като документите (докладите) са описани чрез множество от ключови думи, избрани от таксономия, първоначалните разстояния между тях са изчислени по предложената формула за изчисляване на семантична близост между две множества от възли в обща таксономия. Тя всъщност изчислява степента на подобие, но разстоянието се намира лесно като 1-подобието. Групирането е направено чрез собствена реализация на алгоритъма (на php). Резултатите са представени на оцветената дендрограма (фиг. 9), която е генерирана от Matlab на база на разстоянията между клъстерите.



Фигура 9. Оцветена дендрограма, показваща предложението на агломеративния йерархичен клъстерен анализ за групиране на докладите от конференцията CompSysTech 2016 (без тези от секцията за електронно обучение) в тематични направления

При сравнение на дендрограмата с клъстерите, получени в резултат на алгоритъма на k-средните, прави впечатление, че групирането наистина е подобно, но не и идентично. То не би могло да бъде съвсем идентично, защото агломеративният клъстерен анализ работи

директно с матрицата на разстоянията между документите, изчислени по предложената мярка за сходство, докато алгоритъмът на k-средните допълнително изчислява Манхатъново разстояние върху нейните вектор-редове. Тук отново трябва да се отбележи, че по принцип алгоритъмът на k-средните не работи с разстоянията между документите, а с техните характеристични вектори.

При сравнението се забелязва, че оранжевата група доклади (89, 94, …) в дендрограмата е почти идентична с Група 1, получена от алгоритъма на k-средните. Според последния, доклад 10 също трябва да попадне в тази група, макар че според дендрограмата е добре там, където си е. Подобно, светло-синята група също е почти идентична с Група 4 от плоската клъстеризация. Според алгоритъма на k-средните, в тази група трябва да се включи и доклад 68, който обаче е доста далече в дендрограмата. Комбинацията от първата зелена група (6, 38, 95, …), както и лилавата (17, 20, …) изграждат Група 5, получена от алгоритъма на k-средните. Вероятно, ако при него, докладите бяха групирани не в 6, а в повече групи, тези два клъстера е можело да бъдат отделно. Но пък графиките на силуетите показват, че при 7 и повече групи, грешно класифицираните доклади са повече. И при двата подхода за клъстеризация, полученото групиране е подобно, но агломеративният йерархичен клъстерен анализ предоставя много важна допълнителна информация – семантичното разстояние между всички клъстери и между документите вътре в тях. На базата на тази информация, в последствие, хора експерти биха могли допълнително да разделят или сливат клъстери, да разместват и преподреждат както клъстерите, така и документите в тях. Благодарение на всичко това, с помощта на агломеративния клъстерен анализ би могло да се получи по-точно и по-надеждно групиране на документите, отколкото с методите за плоска клъстеризация и в частност алгоритъма на k-средните. Това твърдение обаче важи предимно за случаите, когато броят на групираните обекти е сравнително малък и хора експерти биха погледнали и модифицирали групирането. Ако броят на документите е много голям, то ползите от йерархичната клъстеризация се обезценяват, защото едва ли някой ще прави допълнителни анализи и ръчно преструктуриране на групирането.

3. Сравнение на клъстеризации (групирания)

В определени ситуации се налага две или повече групирания да бъдат сравнявани помежду си. И не само при използването на алтернативни настройки (брой клъстери, различни метрики за разстояние и т.н.), но най-вече за оценката на качеството на групиранията, получени от съответните автоматизирани методи. Оценката на качеството може да стане чрез т.нар. *вътрешна оценка* и *външна оценка*. Вътрешната оценка отчита колко точно групирането отговаря на входните данни и разстоянията между документите, но напълно игнорира всякакви субективни фактори. Примери за вътрешна оценка са *графиката и стойностите на силуетите*, които отчитат качеството на групирането при методите за плоска клъстеризация, и *коефициентът на кофенетична корелация*, при йерархичната клъстеризация. С тяхна помощ може да се определи оптималният брой клъстери при плоската клъстеризация или най-подходящият метод за свързване при йерархичната, но и двата показателя не могат да кажат какво е качеството на полученото групиране, спрямо субективната човешка преценка. Освен това, за изчисляването тези

показатели са необходими съответно разстоянията от всеки един обект до центъра на съседните клъстери, или разстоянията между отделните клъстери. Много често такива данни просто няма. Те са налични в процеса на работа на клъстеризиращия алгоритъм, но след това (особено при методите за плоска клъстеризация) подобна информация не се съхранява.

Външната оценка, за разлика от вътрешната, разчита изцяло на хора експерти, за да определи точността на полученото групиране от даден автоматизиран метод. Обикновено *експертите предоставят* свое собствено *групиране*, което *се приема за еталон* и полученото групиране се сравнява с него. Очевидно, при еталона не е възможно да има допълнителна информация, като разстояния между клъстерите и/или между документите, затова и показателите за вътрешна оценка като графика на силуетите и коефициент на кофенетична корелация са напълно неприложими.

Съществуват метрики, които позволяват да се оцени доколко дадено групиране съответства на друго групиране. Те проверяват дали всеки един от обектите принадлежи на един и същ клъстер в двете групирания или не. Понеже клъстерите нямат имена по подразбиране, те се идентифицират от елементите в тях, т.е. метриките отчитат отношенията между всички двойки обекти, и проверяват дали тези двойки обекти принадлежат на един и същ или на различни клъстери в едното и в другото групиране. Да, това води до усложняване, но няма друг начин. В този контекст четирите основни показателя (TP, TN, FP, FN) имат следния смисъл:

TP (*true positives, вярно положителни*) – брой на двойките обекти, които принадлежат на един и същ клъстер, както в едното множество A, така и в другото (еталонното) B.

TN (*true negatives, вярно отрицателни*) – брой на двойките обекти, които принадлежат на различни клъстери, както в полученото множество A, така и в еталонното B.

FP (*false positives, фалшиво положителни*) – брой на двойките обекти, които принадлежат на един и същ клъстер в полученото групиране A, но на различни клъстери в еталона B.

FN (false negatives, фалшиво отрицателни) – брой на двойките обекти, които принадлежат на различни клъстери в полученото групиране A, но на един и същ в еталона B.

Подобно на метриките за оценка на върнатите резултати при търсенето на документи, комбинацията от посочените 4 показателя може да се използва за оценка на съответствието/подобието между две групирания (клъстеризации). За целта могат да се приложат: индексът на Rand (1), коефициентът на Jaccard (2) и индексът на Fowlkes–Mallows (3).

Индекс на Rand:

$$RI = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Индекс на Jaccard:

$$JI = \frac{TP}{TP + FP + FN} \tag{2}$$

Индекс на Fowlkes–Mallows:

$$FMI = \sqrt{\frac{TP}{TP + FP} \cdot \frac{TP}{TP + FN}} \tag{3}$$

Индексът на Rand се изчислява по същия начин като точността при върнатите резултати. Тъй като в числителя си отчита и броя на вярно отрицателните (TN), стойността му винаги ще бъде много висока, което го прави не достатъчно прецизен и информативен. На другия полюс пък е индексът на Jaccard, който напълно игнорира вярно отрицателните отношения. Това му дава висока прецизност, но го прави свръх чувствителен. Ако един голям клъстер в оценяваното групиране е разделен на два по-малки в референтното, индексът на Jaccard би имал неправомерно ниска стойност, имайки предвид, че документите все пак са групирани заедно, просто единия голям клъстер е разделен на два по-малки, поради някаква причина. Индексът на Fowlkes–Mallows представлява средно-геометричното между прецизността и откриваемостта, и дава по-балансирани резултати.

Към посочените метрики може да се добави и *процентът на коректно класифицирани (причислени) документи*. Приема се, че даден документ е правилно класифициран, ако е причислен към същия клъстер, към който принадлежи и в референтното групиране.

При подготовката на този файл са използвани материали от:

Калмуков, Й. Методи и алгоритми за търсене и извличане на документи. Издателство Primax Русе, 2022 г., ISBN 978-619-7242-93-5