МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА
УНИВЕРСИТЕТ „ПРОФ. Д-Р АСЕН ЗЛАТАРОВ“ - БУРГАС

Станислав Константинов Попов

Обобщени мрежи и Data Mining

АВТОРЕФЕРАТ

На дисертационен труд за присъждане на образователна и научна степен „доктор” по научна специалност „Компютърни системи и технологии”,
Област на висшето образование 5. Технически науки,
Професионално направление 5.3. Комуникационна и компютърна техника

Научни ръководители

чл.-кор. проф. дмн дтн Красимир Т. Атанасов

проф. д-р Евдокия Н. Сотирова

Бургас
2020

Представеният дисертационен труд беше обсъден на разширен катедрен съвет на катедра „Компютърни системи и технологии“, в Университет „Проф. д-р Асен Златаров“ - Бургас, на заседание, състояло се на 21.09.2020 г. и е насрочен за разкриване на процедура за защита пред научно жури със заповед УД-199 / 08.09.2020 г. на Ректора на Университет „Проф. д-р Асен Златаров“ – Бургас.

Дисертационният труд съдържа 112 страници, от които 20 фигури и са използвани 139 литературни източника. Резултатите са публикувани в 6 статии.

Защитата на дисертационния труд ще се състои на …… от ……. часа в зала ….., Университет „Проф. д-р Асен Златаров“ – гр. Бургас.

Материалите по защитата са предоставени за заинтересованите в деловодството на Университет „Проф.д-р Асен Златаров“-Бургас.

Автор: Станислав Константинов Попов
Заглавие: Обобщени мрежи и Data Mining

Изказвам искрената си благодарност на моите научни ръководители – чл.-кор. проф. дмн дтн Красимир Атанасов и проф. д-р Евдокия Сотирова за споделените знания и опит, ценните им съвети и препоръки по време на изготвянето на този дисертационен труд.

Благодаря също на всички мои колеги от катедра „Компютърни системи и технологии“, както и на моето семейство за безусловната подкрепа.

Увод

Терминът *Data Mining* или *извличане на знания от данни* може да се определи като проучване на събирането, предварителната обработка, същинската обработка, анализирането и получаването на полезна информация от данните. Това, което отличава „извличането на знания от данни“ от останалите дисциплини, е интердисциплинарният характер, който се заключава в интегрирането на знанията за бази данни, аналитични методи и средства, и бизнес познанията. Следователно, „извличането на знания от данни" е широкообхватен термин, който се използва за описание на тези различни аспекти на обработката на данни.

От аналитична гледна точка, извличането на знания от данни е сериозно предизвикателство поради големите несъответствия, които се срещат в отделните типове данни,. Дори в рамките на свързани класове, разликите са значителни. Въпреки разликите, приложенията за извличане на знания от данни често са тясно свързани с един от четирите основни проблема в извличането на знания: извличане на асоциативни модели, клъстеризация, класификация и откриване на екстремални стойности. Тези проблеми са толкова важни, защото се използват като градивни в по-голямата част от приложенията. Това е полезна абстракция, защото ни помага да се концептуализира и структурира по-ефективно областта на извличането на знания от данни.

Данните могат да имат различни формати или *типове*. Типът може да бъде количествен (напр. възраст), категория (напр. етнос), текстов, пространствен, времеви или графичен. Най-често срещаната форма на данните е многомерната, но все пак голяма част принадлежи и към по-сложни типове данни. През последните години преобладаващата тенденция, свързана с непрекъснато събираните данни, води до нарастващ интерес в областта на *потоците от данни*. Например, интернет трафикът генерира големи потоци, които дори не могат да бъдат съхранени ефективно, освен ако не се изразходват значителни ресурси за съхранение. Това води до единствени по рода си предизвикателства от гледна точка на обработката и анализа. В случаите, когато не е възможно данните да бъдат съхранени правилно и точно, цялата обработка трябва да се извършва в реално време.

Цел и задачи на дисертационния труд

Основната цел на изследванията, представени в дисертационния труд, е да се изследват различни процеси от теорията на извличането на знания от данни (Data Mining) чрез моделирането им с помощта на обобщени мрежи и програмната реализация на основните от тях. За да се постигне тази цел, са поставени следните задачи:

1. Да се анализират методите за извличане на закономерности от данни чрез алгоритми за Data Mining процеси;
2. Да се анализират алгоритми за клъстеризация на данни;
3. Разработване на обобщеномрежов модел MapReduce изчислителен модел;
4. Разработване на обобщеномрежов модел на стохастичен Expectation-Maximization алгоритъм.
5. Разработване на обобщеномрежов модел на Deep Learning невронна мрежа.
6. Разработване на обобщеномрежови модели за клъстерен анализ: по CLIQUE (клъстеризация в QUEst), йерархична клъстеризация, клъстеризация по метода STING.
7. Програмна реализация и тестване на MapReduce и K-means алгоритмите.

В текста с [n*] са означени статиите на автора, включени в дисертационния труд.

1. Въведение в обобщените мрежи и Data Mining

В тази глава са дадени основни дефиниции, които са необходими за изложението по-нататък, свързани с теорията на обобщените мрежи и на Data Mining.

1.1 Data Mining

1.1.1 Същност на процеса на извличане на знания от данни

Представено е понятието *Data Mining*, както и основните фази, през които преминава процесът на извличане на данни.

1.1.2 Преглед на основните градивни блокове при процеса на извличане на знания от данни

Представени са проблеми, считани за фундаментални при извличането на информация, свързани с клъстеризация, класификация, извличане на асоциативни шаблони и откриване на екстремални стойности, като те се срещат многократно в контекста на много приложения за извличане на знания.

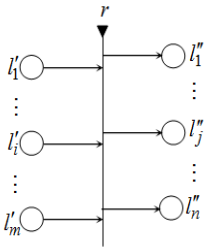
1.2 Обобщени мрежи

1.2.1 Въведение в теорията на обобщените мрежи

Обобщените мрежи (ОМ) са средства за моделиране на паралелно протичащи във времето процеси. Те включват като частни случаи мрежите на Петри и всички техни модификации.

Дефиниция на преход и обобщена мрежа

Неформално е описано понятието Обобщена Мрежа (ОМ) и са описани преходите й. Графично преходът ОМ в се представя като съвкупност от два елемента:  и (Фиг. 1).



Фиг. 1 Представяне на преход в ОМ

Дадена е формална дефиниция на понятията преход и ОМ.

1.2.2 Алгоритми за функциониране на преход и обобщена мрежа

Описани са основни алгоритмите за движение на ядрата през дадени преход и ОМ.

1.2.3 Извод

В Глава първа от дисертационния труд е направен литературен обзор на същността на процеса на извличане на знания и теорията на обобщените мрежи, като по този начин се полага основата на анализът и разработването на обобщеномрежови модели, които ще бъдат представени в следващите две глави на разработката.

2. Обобщеномрежови модели на алгоритми за Data Mining процеси

2.1 MapReduce computational model

Представените тук резултати са публикувани в [1*].

2.1.1 Представяне на Big Data концепцията

Big Data предоставя инструменти за съхранение, управление и манипулиране на огромно количество от данни с подходящите скорост и време. Данните могат да са от вътрешни и външни източници, в различни формати и генерирани от множество приложения. Източниците могат да бъдат различни – транзакции, социални медии, сензори, цифрови изображения и др. Тези съвременни технологии за събиране на данни увеличават обема на информация.

Моделът за програмиране *MapReduce* е мощен инструмент от областта на *Big Data*]. Той не е изцяло нов модел, защото се използва във функционалното програмиране отпреди много години (*Lisp, Haskell, Prolog, R*). Днес съществуват няколко разширения на *MapReduce*, включени в различни платформи с различен дизайн на системата (разпределение, споделени ресурси, комуникация) за подобряване на скоростта.

2.1.2 Разработване на обобщеномрежов модел на MapReduce

Моделът на обобщената мрежа на изчислителния модел *MapReduce,* представен на следващата страница, съдържа следния набор от преходи (Фиг. 2):

– Z_1 - „Входни файлове“;

– Z_2 - „Определяне на входния формат на файла, разделянето му на дялове и трансформирането му в <ключ, стойност> двойки“;

– $\{Z_{3,1}\ldots\ Z_{3,n}\}$ – „Прилагане на *map* функцията за всеки дял информация”;

– $\{Z_{4,1}\ldots\ Z_{4,n}\}$ – „Прилагане на незадължителната комбинаторна функция върху *map* функцията”;

– $\{Z_{5,1}\ldots\ Z_{5,n}\}$ - „Разделяне, разбъркване и сортиране на информацията”;

– Z_6 - „Прилагане на редукторна функцията върху информацията в разделите”;

– Z_7 - „Записване на изходния формат на информацията във файлове”.

Първоначално в позиция L_2 има едно α-ядро. То ще бъде на своето място през цялото време на функциониране на ОМ. То има следната характеристика: *„файлове, съхранявани в разпределена файлова система“*. Също така, първоначално в позиция L_{18} има едно ядро с характеристика „*текущо състояние на редукторните функции и броя на редукторите“.*

Ядро влиза в мрежата през позиция L_1 с начална характеристика „*входни файлове*“. Преходът Z_1 има следния вид:

$$Z_1 = \langle \{L_1, L_{21}, L_2\},\ \{L_2, L_3\},\ R_1,\ \vee(L_1, L_{21}, L_2)\rangle,$$

където

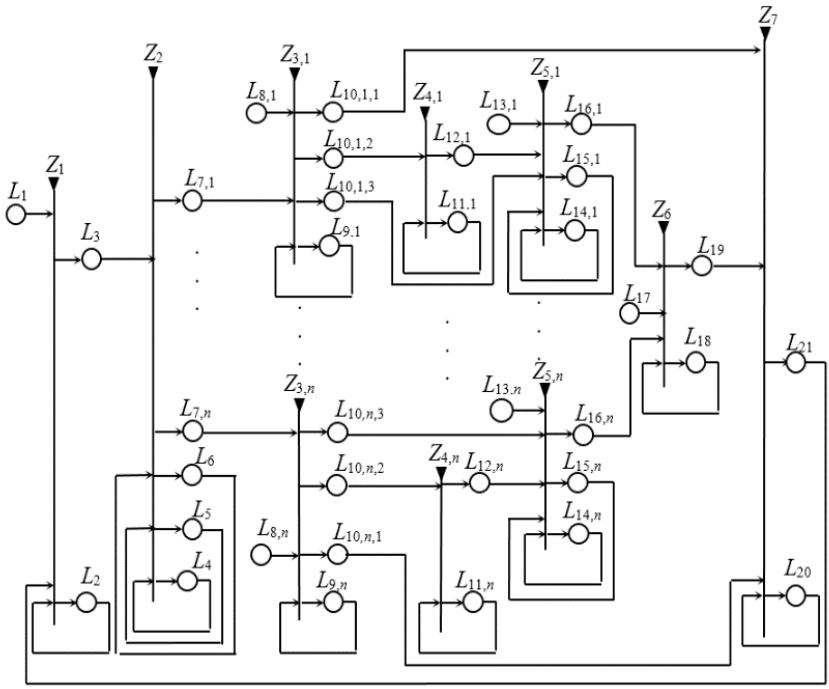
$$R_1 = \begin{array}{c|cc} & L_2 & L_3 \\ \hline L_1 & true & false \\ L_{21} & true & false \\ L_2 & W_{2,2} & W_{2,3} \end{array}$$

и:

- $W_{2,3}$ = „Избран е вход за *MapReduce* обработка“;
- $W_{2,2} = \neg W_{2,3}$.;

Ядрата влизащи в позиция L2 от позиция L1 и L21 не придобиват нови характеристики. Ядрото от позиция L2 преминава в L3 със следната характеристика:

„избран файл за MapReduce обработка“.



Фиг. 2 Обобщеномрежов модел на MapReduce

На последния преход Z_7 , който има вида:

$Z_7 = \langle \{L_{10,1,1}, \ldots, L_{10,n,1}, L_{19}, L_{20}\}, \{L_{20}, L_{21}\}, R_7, \vee(L_{10,1,1}, \ldots, L_{10,n,1}, L_{19}, L_{20}) \rangle$, матрицата е:

$$R_7 = \begin{array}{c|cc} & L_{20} & L_{21} \\ \hline L_{10,1,1} & true & false \\ \ldots & \ldots & \ldots \\ L_{10,n,1} & true & false \\ L_{19} & true & false \\ L_{20} & W_{20,20} & W_{20,21} \end{array}$$

и:

- $W_{20,21}$ = „Изходният файл е записан“;
- $W_{20,20} = \neg W_{20,21}$.

Ядрата, преминаващи в позиция L_{20} не придобиват нови характеристики. Ядрото, влизащо в позиция L_{21} придобива следната характеристика:

„записан изходен файл“.

2.1.3 Реализация на MapReduce алгоритъма в MATLAB

Описанието на алгоритъма в MATLAB изглежда по следния начин:

1)Mapreduce чете блок от данни от входното хранилище на информация, използвайки *[data, info] = read(ds)* и след това извиква *map*-функцията да работи върху същия блок.

2)Map-функцията получава блока от данни, организира го и след това използва *add* и *addmulti* функции, за да добави *key*-*value* двойки към междинен обект за съхранение на данни, наречен *KeyValueStore*. Броят на извикванията на *map*-функцията от *mapreduce* е равен на броя на блоковете във входното хранилище.

*3)*След като *map*-функцията премине през всички блокове от данни, *mapreduce* групира всички стойности в междинния *KeyValueStore* обект по уникален ключ.

4)mapreduce извиква *reduce*-функцията за всеки уникален ключ, добавен от *map*-функцията. Всеки уникален ключ може да има много асоциирани стойности. *mapreduce* предава стойностите към *reduce*-функцията като *ValueIterator* обект, който е обект, използван за обхождане на стойностите. *ValueIterator* обектът за всеки уникален ключ съдържа всички асоциирани стойности за този ключ.

5)Reduce-функцията използва *hasnext* и *getnext* функции за обхождане на стойностите в *ValueIterator* обекта една по една. След обобщаване на междинните резултати от *map*-функцията, reduce функцията добавя финалните двойки *key*-*value* към изхода, използвайки *add* и *addmulti* функциите. Подредбата на ключовете в изхода е същата както тази, по която *reduce*-функцията ги добавя към крайния *KeyValueStore* обект, т.е. *mapreduce* не сортира изрично изхода.

2.1.4 Моделиране на вероятности чрез логистична регресия и MapReduce

Целта, която трябва да се постигне при Data Mining задачите, свързани с прогнозиране, е да се намери модел, чрез който да се предсказват стойностите на една от променливите на база на известните стойности на други променливи. Тези задачи могат да бъдат разделени на два вида – класификация и регресия. Класификацията е процес на изграждане на модели, описващи съществуващи класове от обекти, с цел използването им за определяне класа на обекти с неизвестен клас.

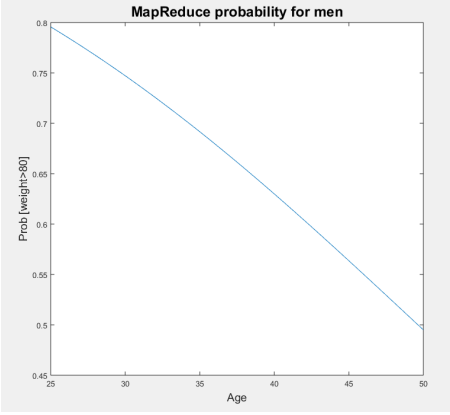
Логистичната регресия е начин да се моделира вероятност за дадено събитие като функция на друга променлива. В следващите редове е показано действието на *MapReduce* чрез свързването на множество повиквания за *mapreduce* за изпълнение на итеративен алгоритъм. Тъй като всяка итерация изисква отделно преминаване през данните, анонимната функция предава информация от една итерация в следващата, за да предостави информацията директно на *mapper*-а. За да завършат логистичната регресия, *map* и *reduce* функциите трябва изпълнят претеглена линейна регресия, базирана на текущите коефициентни стойности.

Моделиране на вероятност – Реализация 1

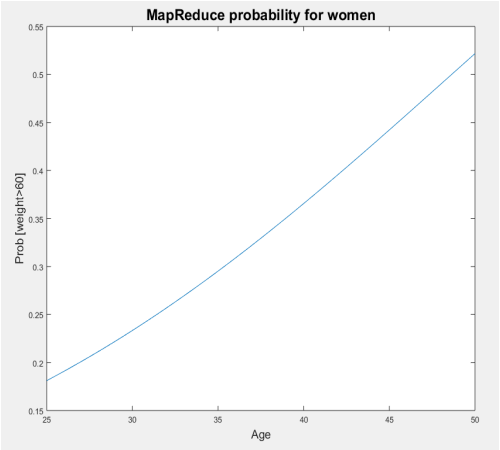
Целта е да се моделира вероятността за мъже и жени между 25 и 50 години, да имат тегло над определени килограми. Използваните данни са за пациенти на болница. В случая вероятността *„Prob”* е функция на променливата *„Age”*. Резултатът са две отделни графики - една за мъжете и една за жените, като за мъжете вероятността е за над 80 кг., а за жените – над 60 кг. В тестването са заложени пет на брой итерации, след извършването на които алгоритъмът трябва да изчисли вероятността.

Резултати: След петата итерация моделите на вероятност са представени на Фиг. 3 и Фиг. 4. Тъй като критерият за теглото е количествен и е съобразен с пола, в кода на програмата е заложено резултатите да се представят в две отделни графики

.



Фиг. 3 Вероятностен модел 1



Фиг. 4 Вероятностен модел 2

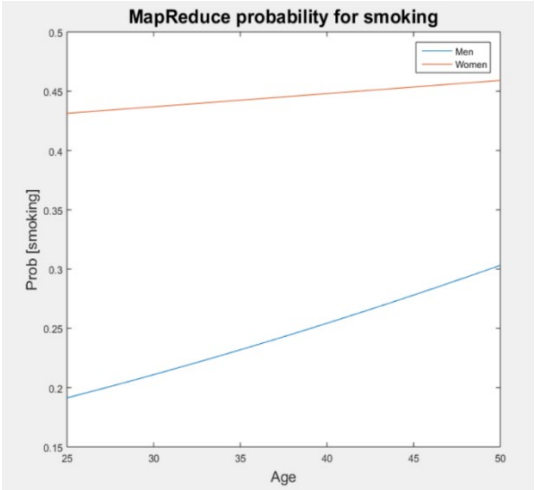
Моделиране на вероятност – Реализация 2

В посочения по-долу пример се моделира вероятността хората да пушат в зависимост от тяхната възраст. Тъй като критерият е един и същ, резултатът е показан в една обща графика.

Кодът в *reduce*-функцията остава същият (Reduce_fit.m). В *map*-функцията (Mapper_fit_all) се премахва променливата *control_wgt* и вместо *wgt*, използваме *smoke*.

Резултат: След извършване на петте итерации, графиката на вероятност е следната:

На Фиг. 5 се вижда, че и при двата пола с увеличаването на възрастта между 25 и 50 години, вероятността за пушене също нараства. При мъжете кривата е по-стръмна, но като цяло вероятността е в по-малки граници. При жените кривата е по-полегата, но стойностите на вероятност са по-високи.



Фиг. 5 Вероятностен модел 3

2.2 Стохастичен Expectation Maximization алгоритъм

Представените тук резултати са публикувани в [4*].

EM-алгоритъмът се прилага успешно в областта на извличането на знания като итеративен метод, който се стреми да намери максималния вероятностен оценител на параметър θ, принадлежащ на дадено параметрично разпределение на вероятностите. Основната цел е да се максимизира Q-функцията, която е резултат от изваждането на целевата променлива и първоначалното изчисление на θ.

Стандартната EM спомагателна функция е най-доброто изчисление на логаритмичната вероятност на наблюдаваните данни в смисъла на условната средно-квадратична грешка [64]. Идеята, която стои зад Stochastic Expectation-Maximization (SEM), подобно на други стохастични варианти на EM е, че може да не е необходимо да се изисква т .нар. „най-добра“ оценка. Следователно, SEM заменя стандартната спомагателна функция с:

$$\hat{Q}(\theta \mid \theta') = \log p(z' \mid \theta'), \tag{2.1}$$

където z' е случайна извадка от последващото разпределение на неизвестната променлива $p(z \mid y, \theta)$. Това води до следната модифицирана итерация; при дадена текуща оценка θ_n:

- Симулационна стъпка: изчисляване на $p(z \mid y, \theta_n)$ и изтегляне на неизвестна извадка z_n от $p(z \mid y, \theta_n)$;
- Максимизираща стъпка: намиране на $\theta_{n+1} = \arg\max_{\theta} p(z_n \mid \theta)$.

2.2.1 Разработване на обобщеномрежов модел на стохастичен ЕМ алгоритъм

Обобщеномрежовият модел на стохастичния ЕМ алгоритъм съдържа 8 прехода и 23 позиции (Фиг. 6):

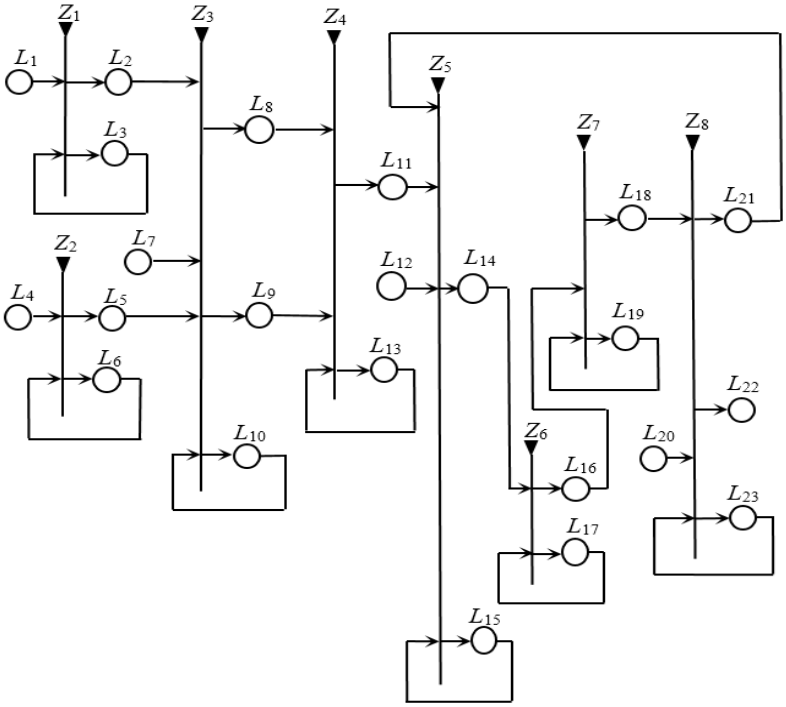
$$A = \{Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8\},$$

където:

- Z_1 – „Избиране на случайни наблюдавани данни (y)";
- Z_2 – „Избиране на латентна променлива (z)";
- Z_3 – „Формулиране на параметричната плътност на y и параметричната плътност на z";
- Z_4 – „Формиране на агрегатната параметрична плътност на y и z, $p(z \mid y, \theta)$";
- Z_5 – „Изчисляване на $p(z \mid y, \theta_n)$";
- Z_6 –„Извличане на ненаблюдавана извадка (z_n) от $p(z \mid y, \theta_n)$;
- Z_7 – „Формиране на заместващата функция $\hat{Q}(\theta / \theta_n)$";
- Z_8 – „Намиране на $\theta_{n+1} = \arg\max_{\theta} p(z_n \mid \theta)$";

Първоначално, α-ядро навлиза в ОМ през позиция L_1. Ядрото има първоначална характеристика:

„наблюдавани данни".



Фиг. 6 Обобщеномрежов модел на SEM алгоритъм

Ядрото, влизащо в позиция L_3 не придобива нова характеристика. След активирането на прехода Z_1, α-ядрото се придвижва към позиция L_2 и придобива следната характеристика:

„избрани наблюдавани данни”.

Преход Z_1 има следния вид:

$$Z_1 = \langle \{L_1, L_3\}, \{L_2, L_3\}, R_1, \vee(L_1, L_3) \rangle,$$

където:

$$R_1 = \begin{array}{c|cc} & L_2 & L_3 \\ \hline L_1 & false & true \\ L_3 & W_{3,2} & true \end{array}$$

и

- $W_{3,2}$ = „Наблюдаваните данни са избрани”.

β-ядро влиза в мрежата през позиция L_4 с начална характеристика:

„латентни данни“.

Ядрото, влизащо в позиция L_6 не придобива нова характеристика. След активирането на прехода Z_2, β -ядрото се придвижва към позиция L_5 и придобива следната характеристика:

„избрани латентни данни”.

.
.
.

$$Z_8 = \langle \{L_{18}, L_{20,} L_{23}\}, \{L_{21}, L_{22,} L_{23}\}, R_8, \vee(\wedge (L_{18}, L_{20}), L_{23})\rangle,$$

където:

$$R_8 = \begin{array}{c|ccc} & L_{21} & L_{22} & L_{23} \\ \hline L_{18} & false & false & true \\ L_{20} & false & false & true \\ L_{23} & W_{23,21} & W_{23,22} & true \end{array}$$

и

- $W_{23,21}$ = „n-тото изчисление на θ е увеличено с 1”;
- $W_{23,22}$ = „Намерен е параметър θ , който максимизира Q-функцията”.

2.3 Deep Learning невронна мрежа

Представените тук резултати са публикувани в [2*].

2.3.1 Невронни мрежи и Data Mining

Невронните мрежи са нелинейни инструменти за моделиране на статистически данни, разпознаване на обекти и прогнозиране. Те могат да се използват за моделиране на сложни отношения между входовете и изходите или за намиране на зависимости в данните.

Изкуствените невронни мрежи предлагат качествени методи за бизнес и икономически системи, които традиционните количествени инструменти в статистиката и иконометрията не могат да определят количествено поради сложността при превеждането на системите в прецизни математически функции. Следователно използването на невронни мрежи при извличане на знания данни е обещаващо поле за изследване, особено като се има предвид доказаната способност на невронните мрежи да откриват и асимилират връзки между голям брой променливи.

Deep Learning обикновено се отнася до методи, които отбелязват (картографират) данните чрез множество нива на абстракция, където по-високите нива представляват по-абстрактни позиции. Целта е алгоритъмът автоматично да усвоява сложни функции, които съотнасят входовете към изходите. Една от реализациите на алгоритъма се явява под

формата на невронни мрежи с право предаване, където нивата на абстракция се моделират от множество нелинейни скрити слоеве.

2.3.2 Многослойни невронни мрежи

В m-слойните невронни мрежи, изходът на един слой става вход на следващия. Уравнението, описващо тази операция е следното:

$$a^m = f^m(w^m \ldots f^2(w^2 f^1(w^1 p + b^1) + b^2) + \ldots + b^m), \quad (2.2)$$

където:

- a^m е изходът на m-слойя на невронната мрежа за $m = 1, 2$;
- w^m е матрицата на тегловните коефициенти за всеки вход на m-ия слой;
- b е отклонението на входа на неврона;
- f^1 е трансферната функция на първия слой;
- f^2 е трансферната функция на втория слой;

.
.
.

- f^m е трансферната функция на m-ия слой.

Невронът в първия слой получава външни входове p. Изходите на неврона от последния слой определя изхода на невронната мрежа a. Обучаващите множества се предават на алгоритъма (входна стойност и цел за постигане – на изхода на мрежата):

$\{p_1, t_1\}, \{p_2, t_2\}, ..., \{p_Q, t_Q\}$, където:

$Q \in (1, ..., n)$, n – брой на обучаваната двойка, където p_Q е входната стойност (на входа на мрежата), а t_Q е изходната стойност, съответстваща на целта. Всеки вход на мрежата е предварително установен и е постоянен, а изходът трябва да отговаря на целта. Разликата между входните стойности и целта е грешката $e = t–a$.

Когато се обучава многослойна невронна мрежа, обикновено наличните данни трябва да бъдат разделени на три подмножества. Първото подмножество се нарича „обучаващо множество“ и се използва за изчисляване на градиента и актуализиране на теглата и отклоненията. Второто множество се нарича „валидиращо множество“. Грешката при него се следи по време на процеса на обучение. Валидиращата грешка обикновено намалява по време на началната фаза на обучението. Третото множество е „тестово множество“. Сумата от трите множества трябва да е равна на 100% от обучаващите се двойки.

Условието за обучена мрежа е когато $e^2 < E_{max}$, където E_{max} е максималната квадратична грешка.

2.3.3 Разработване на обобщеномрежов модел на Deep Learning невронна мрежа

Обобщената мрежа на процеса на работа на DLNN е показан на Фиг. 7 (на следващата страница). Първоначално следните ядра влизат в мрежата:

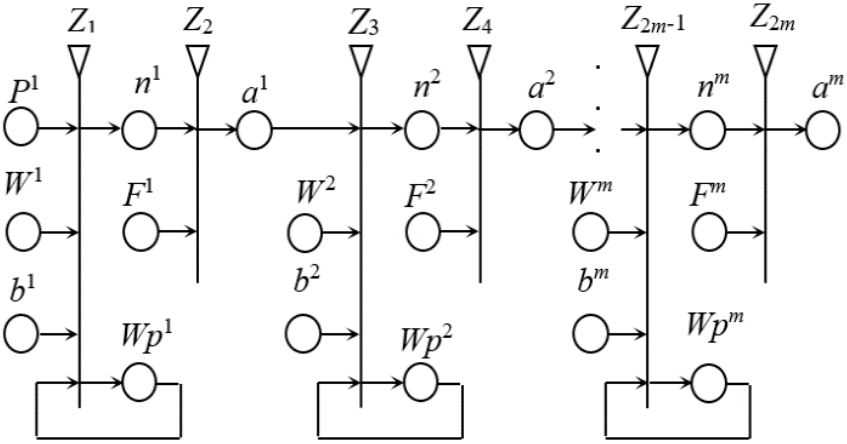
- в позиция P^1 – α^1-ядро с начална характеристика $x_0^{\alpha^1} = p^1$;
- в позиция W^1 – β^1- ядро с начална характеристика $x_0^{\beta^1} = w^1$;
- в позиция b^1 – γ^1- ядро с начална характеристика $x_0^{\gamma^1} = b^1$;
- в позиция F^1 –едно δ^1- ядро с начална характеристика $x_0^{\delta^1} = F^1(n) = a$;
- в позиция W^2 – β^2- ядро с начална характеристика:

$$x_0^{\beta^2} = W^2 = \begin{vmatrix} W_{1,1} & W_{1,2} & \dots & W_{1,R} \\ W_{2,1} & W_{2,2} & \dots & W_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ W_{S,1} & W_{S,2} & \dots & W_{S,R} \end{vmatrix};$$

- в позиция b^2 – γ^2- ядро с начална характеристика $x_0^{\gamma^2} = b_1, b_2, ..., b_S$;
- в позиция F^2 – едно δ^2- ядро с начална характеристика $a = \frac{1}{1+e^{-n}}$

;

- в позиция W^{m} – β^{m}- ядро с начална характеристика $x_0^{\beta^m} = W^m = W_{1,1}, W_{1,2}, ..., W_{1,R}$;
- в позиция b^m – γ^m- ядро с начална характеристика $x_0^{\gamma^m} = b_1, b_2, ..., b_S$;
- в позиция F^m – едно δ^m- ядро с начална характеристика $x_0^{\delta^m} = F^m(n) = a.$



Фиг. 7 Обобщеномрежов модел на DLNN

Обобщената мрежа има следния вид:

$$A = \{Z_1, Z_2, Z_3, Z_4, Z_{2m-1}, Z_{2m}\},$$

където преходите описват следните процеси:

- Z_1 – „Изчисляване влиянието на първия слой на DLNN (n^1)“;
- Z_2 – „Изчисляване изхода на първия слой на DLNN (a^1)“;
- Z_3 – „Изчисляване влиянието на втория слой на DLNN (n^2)“;
- Z_4 – „Изчисляване изхода на втория слой на DLNN (a^2)“;
- Z_{2m-1} – „Изчисляване влиянието на третия слой на DLNN (n^m)“;
- Z_{2m} – „Изчисляване изхода на третия слой на DLNN (a^m)“.

Преходите на обобщената мрежа имат следния вид:

$$Z_1 = \langle \{P^1, W^1, b^1, Wp^1\}, \{n^1, Wp^1\}, R_1, \wedge(\vee(P^1, W^1), \vee(b^1, Wp^1)) \rangle,$$

където:

$$R_1 = \begin{array}{c|cc} & n^1 & Wp^1 \\ \hline p^1 & false & true \\ W^1 & false & true \\ b^1 & true & false \\ Wp^1 & true & false \end{array}.$$

Ядрата α^1, β^1 и γ^1 се сливат в χ^1-ядро с характеристика: $x_{\text{cu}}^{\chi^1} = x_{\text{cu}}^{\beta^1} x_{\text{cu}}^{\alpha^1} + x_{\text{cu}}^{\gamma^1}$

.
.
.

$Z_{2m-1} = \langle\{a^2, W^m, b^m, Wp^m\}, \{n^m, Wp^{2m}\}, R_{2m-1}, \wedge(\vee(a^2, W^m), \vee(b^m, Wp^m))\rangle$,
където:

$$R_5 = \begin{array}{c|cc} & n^m & Wp^m \\ \hline a^2 & false & true \\ W^m & false & true \\ b^m & true & false \\ Wp^m & true & false \end{array}.$$

Ядрата σ^2, β^3 и γ^3 се обединяват в χ^3-ядро с характеристика: $x_{\text{cu}}^{\chi 3} = x_{\text{cu}}^{\beta 3} x_{\text{cu}}^{\sigma 2} + x_{\text{cu}}^{\gamma 3}$.

$$Z_{2m} = \langle\{n^m, F^m\}, \{a^m\}, R_{2m}, \wedge(n^m, F^m)\rangle,$$

където:

$$R_{2m} = \begin{array}{c|c} & a^m \\ \hline n^m & true \\ F^m & true \end{array}.$$

Ядрата δ^3 и χ^3 се сливат в σ^3-ядро с характеристика: $x_{\text{cu}}^{\sigma 3} = x_0^{\delta 3}(x_{\text{cu}}^{\chi 3})$.

2.4 Извод

В Глава втора от дисертационния труд са разработени модели на процеси, свързани с извличане на знания от данни (*MapReduce computational model, Стохастичен EM-алгоритъм* и *Deep learning невронна мрежа*) с помощта на апарата на обобщените мрежи. По този начин може да се анализират отделните стъпки и да се вникне по-дълбоко в начина им на действие.

Обобщеномрежовият модел на процеса на Mapreduce може да бъде използван за изследване и наблюдение. Също така, той може да се допълни със споменатите в началото на настоящата глава разширения към процеса. В допълнение, в MATLAB е направена демонстрация на действието на MapReduce, като са моделирани две отделни вероятности на база логистична регресия върху данни за пациенти.

EM-алгоритъмът и неговите варианти се използват редовно за решаване на широк кръг от съвременни задачи за оценка - от откриване на зависимости в ДНК последователности до приспособяване на смесени

модели към еднородни цели в клъстера. Обобщеномрежовият модел, представен в настоящото изследване, показва как действа SEM-алгоритъмът, като разделя двете му основни стъпки на по-малки и ги описва в детайли.

Предимствата на невронните мрежи се състоят в това, че те са: ефективен метод за създаване на модели с голяма точност на предсказване; справят се успешно при наличието на взаимовръзки между входните променливи, Представената в настоящата глава Deep Learning невронна мрежа, използвана за обучение, се състои от входен слой, скрит слой и изходен слой. Нейният обобщеномрежов модел е подходящ за отправна точка при моделирането на други типове невронни мрежи.

3. Обобщеномрежови модели на алгоритми за клъстеризация

3.1 Обобщеномрежов модел на клъстерен анализ, използващ CLIQUE: клъстеризация в QUEst

Представените тук резултати са публикувани в [5*].

Клъстерният анализ намира прилики между данните според характеристиките, които се намират в тях, и групира сходни обекти с данни в клъстери. Клъстерът е група от подобни обекти, които се различават от обектите, включени в други клъстери. В зависимост от подхода, използван за определяне на разстоянието между два обекта, са разработени различни алгоритми за клъстерен анализ.

CLIQUE алгоритъмът автоматично намира подпространства от най-високо измерение, така че в тези подпространства да съществуват клъстери с висока плътност; не е чувствителен към реда на входните записите; не предполага някакво канонично разпределение на данните и се увеличава линейно с размера на входните данни. Слабата страна на алгоритъма е, че точността на клъстерния резултат може да се намали за сметка на простотата на метода. Основните стъпки на CLIQUE алгоритъма са:

- Разделяне на пространството на данни и намиране на броя точки, които лежат във всяка клетка на дяла;
- Определяне на подпространствата, които съдържат клъстери, като се използва принципа Apriori;
- Идентифициране на клъстерите:

– Определяна на плътни единици във всички подпространства, които представляват интерес;

– Определяне на свързани плътни единици във всички подпространства, които представляват интерес;

- Генериране на минимално описание за клъстерите:

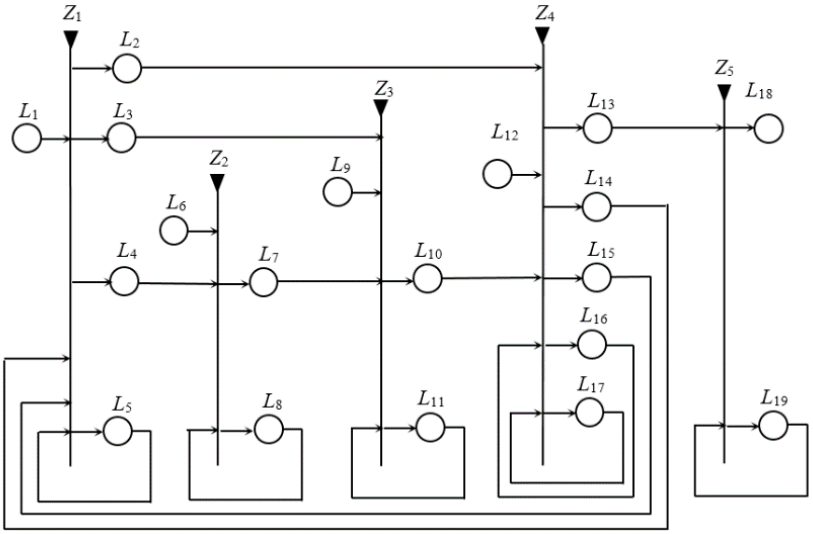
– За всеки клъстер се определят максималната му област от свързани плътни единици, която той покрива;

– Определяне на минималното покритие за всеки клъстер.

Разработване на обобщеномрежовия модел на CLIQUE алгоритъма

Представеният по-долу обобщеномрежов модел на процеса на прилагане на клъстерен анализ, използвайки CLIQUE метода съдържа 5 прехода и 19 позиции (Фиг.8). Преходите представляват следните процеси:

- Z_1 – „Многомерна база от данни“;
- Z_2 – „Предварителна обработка“;
- Z_3 – „Разделяне на пространството на данни и намиране на броя точки, които лежат във всяка клетка на дяла“;
- Z_4 – „Определяне на подпространствата, които съдържат клъстери, като се използва принципа Apriori“;
- Z_5 – „Генериране на минимално описание за клъстерите“.



Фиг. 8 Обобщена мрежа на клъстерен анализ използващ CLIQUE

Преходът Z_1 има следния вид:

$$Z_1 = \langle \{L_1, L_{14}, L_{15}, L_5\}, \{L_2, L_3, L_4, L_5\}, R_1, \vee(L_1, L_{14}, L_{15}, L_5) \rangle,$$

където

$R_1 =$

	L_2	L_3	L_4	L_5
L_1	*false*	*false*	*false*	*true*
L_{14}	*false*	*false*	*false*	*true*
L_{15}	*false*	*false*	*false*	*true*
L_5	$W_{5,2}$	$W_{5,3}$	$W_{5,4}$	$W_{5,5}$

и:

- $W_{5,2}$ = „Има избрани многомерни данни за извършване на Apriori процедура за генериране на *кандидати“*
- $W_{5,3}$ = „Има избрани многомерни данни за извършване на процедура по разделяне“;
- $W_{5,4}$ = „Има избрани многомерни данни за извършване на предварителна обработка“
- $W_{5,5} = \neg (W_{5,2} \wedge W_{5,3} \wedge W_{5,4})$.

α-ядрата преминаващи в позиция L_5 не получават нова характеристика. α_1-ядрото в позиция L_5 генерира нови α-ядра, които влизат в позиции L_2, L_3 и L_4 с характеристики, както следва:

„избрани многомерни данни за извършване на Apriori процедура за генериране на кандидати“,

„избрани многомерни данни“,

„избрани многомерни данни за извършване на предварителна обработка“.

β_2-ядро навлиза в мрежата през позиция L_6. То има начална характеристика:

„методи за предварителна обработка“.

.

.

.

Преходът Z_5 има следния вид:

$$Z_5 = \langle \{L_{13}, L_{19}\}, \{L_{18}, L_{19}\}, R_5, \vee(L_{13}, L_{19}) \rangle,$$

където

$R_5 =$

	L_{18}	L_{19}
L_{13}	*false*	*true*
L_{19}	$W_{19,18}$	$W_{19,19}$

и:

- $W_{19,18}$ = „Има минимално описание за клъстерите“;
- $W_{19,19} = \neg W_{19,18.}$

Ядрата преминаващи в позиция L_{19} не получават нова характеристика. α-ядрото в позиция L_{19} генерира ново ядро, което влиза в позиция L_{18} с характеристика:

„резултат от процеса на клъстеризация“.

3.2 Обобщеномрежов модел на процеса на йерархичен клъстерен анализ

Представените тук резултати са публикувани в [6*].

Йерархичната клъстеризация е вид клъстерен анализ, който групира обекти от данни в дърво от клъстери. Тази клъстеризация се формира по един от следните начини: отдолу нагоре (сливане) или отгоре надолу (разделяне). Йерархичното събирателно клъстеризиране започва с всеки обект от данни като отделен клъстер и стъпка по стъпка двата най-близки клъстера се сливат итеративно. Разделителното клъстеризиране започва с всички обекти групирани в един клъстер. Клъстерите се разделят, докато всеки обект не бъде в отделен клъстер. В клъстерния анализ, измерването на разстоянието и приликите се използва за определяне на сходството между две точки.

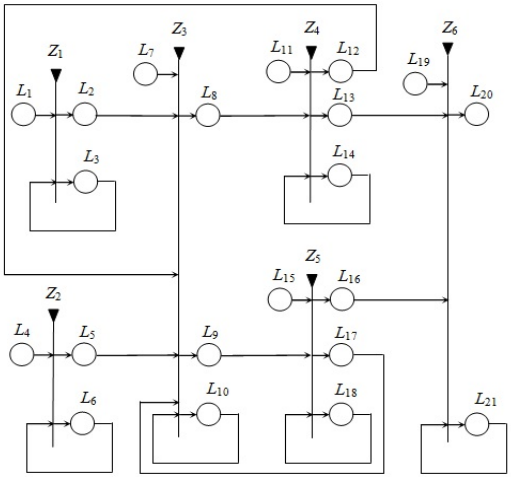
Разстоянието между обектите с данни може да се намери чрез различни мерки за разстояния – Евклидово разстояние, Манхатанско разстояние (името идва от разстоянието между градските блокове), разстояние на Чебишев и др. Разработване на обобщеномрежовия модел на йерархичен клъстерен анализ

Обобщеномрежовият модел съдържа 6 прехода и 21 позиции. (Фиг. 9). Преходите са следните:

$$A = \{Z_1, Z_2, Z_3, Z_4, Z_5, Z_6\},$$

и описват процесите:

- Z_1 – „Избор на данни”;
- Z_2 – „Определяне типа на йерархична клъстеризация”;
- Z_3 – „Изчисляване на сходствата/различията”;
- Z_4 – „Извършване на йерархичен събирателен клъстерен анализ”;
- Z_5 – „Извършване на йерархичен разделящ клъстерен анализ”;
- Z_6 – „Построяване на дендрограма (или диаграма на Вен)”;



Фиг. 9 ОМ модел на процеса на йерархичен клъстерен анализ

Преходът Z_1 има следния вид:

$$Z_1 = \langle \{L_1, L_3\}, \{L_2, L_3\}, R_1, \vee(L_1, L_3) \rangle,$$

където:

$$R_1 = \begin{array}{c|cc} & L_2 & L_3 \\ \hline L_1 & false & true \\ L_3 & W_{3,2} & W_{3,3} \end{array}$$

и:

- $W_{3,2}$ = „Избрани са входни данни за клъстерен анализ“;
- $W_{3,3} = \neg W_{3,2}$.

α-ядрото влизащо в позиция L_3 (от L_1) не придобива нова характеристика. α-ядрото в позиция L_3 генерира ново α-ядро, което влиза в позиция L_2 с характеристика: „*избрани данни за йерархичен клъстерен анализ“*. Това α-ядро се придвижва към преход Z_3, където ще се слее с β-ядрата от позиции L_5 и L_7 в позиция L_{10}.

β-ядрото навлиза в мрежата през позиция L_4 и има начална характеристика: „*тип на йерархичния клъстерен анализ*“.

.

.

.

Преходът Z_6 има следния вид:

$$Z_6 = \langle \{L_{19}, L_{13}, L_{16}, L_{21}\}, \{L_{20}, L_{21}\}, R_6, \vee (L_{19}, L_{13}, L_{16}, L_{21}) \rangle,$$

където:

	L_{20}	L_{21}
L_{19}	*false*	*true*
L_{13}	*false*	*true*
L_{16}	*false*	*true*
L_{21}	$W_{21,20}$	$W_{21,21}$

и:

- $W_{21,20}$ = „Клъстерите са визуализирани”,
- $W_{21,21}$= ¬ $W_{21,20.}$

Освен йерархична клъстеризация, описана по-горе, при *data mining* процесите може да се използва и нейрархичен клъстерен анализ. Ето защо, да да бъде пълен анализът в следващата част от настоящата глава програмно е демонстриран начинът на действие на *k-means (К-средните величини*) клъстеризация.

3.3 K-means клъстерен анализ

При k-means клъстерния анализ се отчита разстоянието на всяка единица до центровете на отделните клъстери, като най-близкото разстояние определя принадлежността на единицата към съответния клъстер. Методът изисква предварително да се определи броят на клъстерите.

Работата на алгоритъма е разделена на няколко етапа:

1) Избират се случайни *к* точки, които са началните центрове на клъстерите.

2) Всеки обект се свързва с най-близкия клъстерен център.

3) Преизчисляват се центровете на клъстерите според сегашния им състав.

4) Ако критерият за спиране на алгоритъма не е изпълнен, се връща на стъпка **2**.

Като критерий за спиране на работата на алгоритъма обикновено се избира минималната промяна в средната квадратична грешка.

3.4 Графична симулация на k-means алгоритъма в MATLAB

С помощта на MATLAB (R2015a) е представено действието на K-Means Cluster Analysis. За целта са разработени два варианта на клъстеризация на данните: със случайни числа, обединени в 5 класа с по 10 елемента в тях и таблични данни за пациенти. Програмната реализация

на двата варианта е сходна, като една от основните разлики е, че при болничните данни (представени в табличен вид), последните трябва да се трансформират в матрица, за да работи правилно програмата.

➢ **Вариант I:**

Генерират се случайни числа (10 елемента, обединени в 5 класа), като целта е да се намерят центровете на клъстерите. Програмната реализация се състои от един основен файл, който се изпълнява и пет функции.

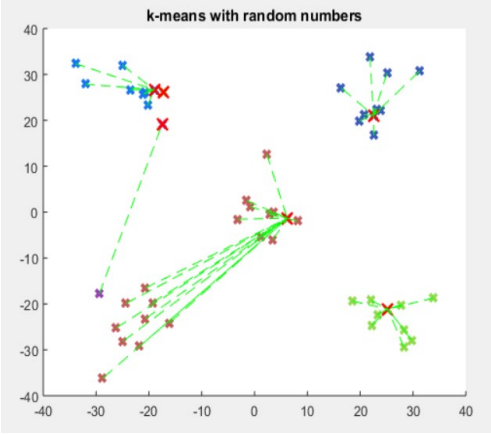
Резултат:

При максимален брой на итерациите равен на 6 и брой на клъстерите равен на 6 се получават резултатите, изобразени на фигури 10, 11 и 12.

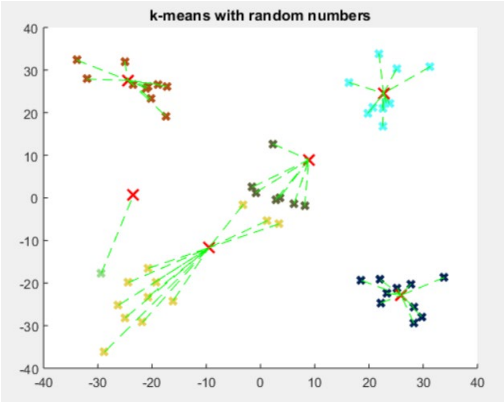
При зададените в началото параметри, алгоритъмът завършва след 3 итерации (от максимални 6). Както се вижда на графиките, с „ ***X*** ” е отбелязан центъра на клъстерите, с „***** ” случайните числа, които трябва да се съотнесат към някой от клъстерите и с „ **- - -** “ е показана тяхната принадлежност. Цялата визуализация на действието на алгоритъма е осъществена чрез програмен код.

На Фиг. 12 могат да се видят петте формирани клъстера с центрове:

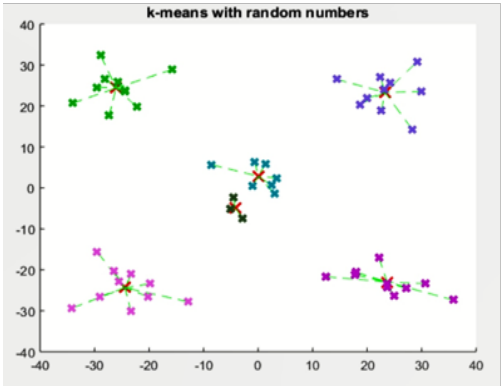
- 24.6767372846014, -24.4336054611274;
- 0.471178593955773, -1.10496173346548;
- 25.8047635903735, 22.2068631421341;
- -22.3858309083016, 22.8223928202951;
- -25.1606687149199, -26.5989557975283
- -21.46356256363626, -19.34168327728218.



Фиг. 10 Итерация 1



Фиг. 11 Итерация 2



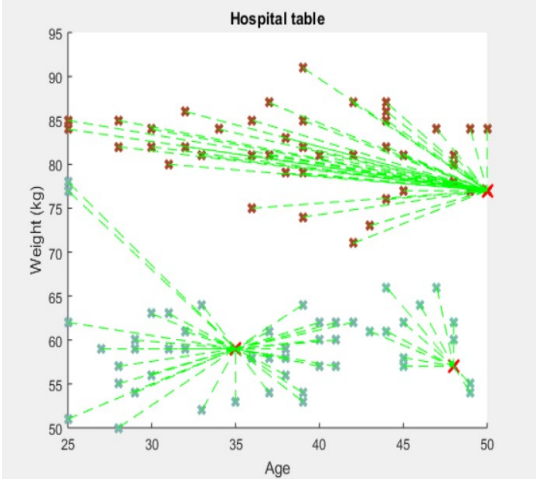
Фиг. 12 Итерация 3 - край

➢ **Вариант II:**

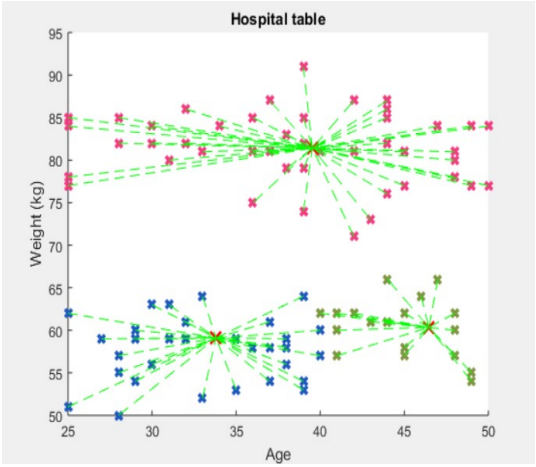
Използват се болнични данни за възрастта и теглото на пациенти. Таблицата с информацията е създадена по данни от Световната здравна организация. В случая броят на клъстерите е 3, а максималните итерации - 6. Тъй като таблицата с данните е от тип *dataset*, първо трябва да се преобразуват в *double.*

Резултат:

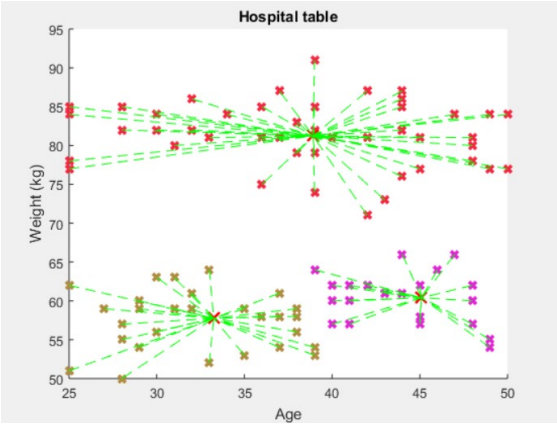
Алгоритъмът завършва след четвъртата стъпка, т.е. той успява да съотнесе данните към центровете на формираните клъстерите по-рано от зададените му максимални шест итерации.



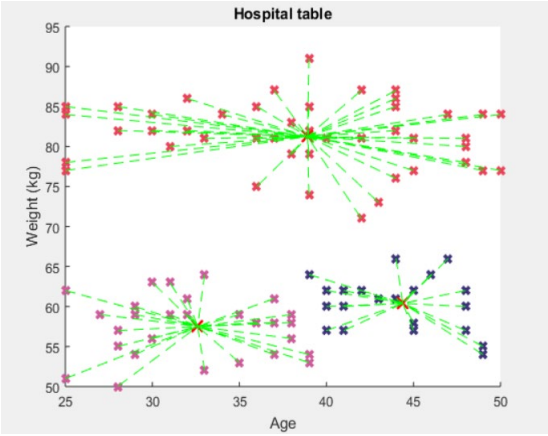
Фиг. 13 Итерация 1



Фиг. 14 Итерация 2



Фиг. 15 Итерация 3



Фиг. 16 Итерация 4 -край

На последната графика (Фиг. 16) могат да се видят трите формирани клъстера с центрове:

- 37.7169811320755, 58.7547169811321;
- 31.1111111111111, 82.1666666666667;
- 43.7586206896552, 80.6551724137931.

3.5 Обобщеномрежов модел на процеса на клъстерен анализ, използващ STING: статистически информационен мрежов подход към пространствено извличане на знания

Представените тук резултати са публикувани в [3*].

STING (Statistical Information Grid) алгоритъм

STING алгоритъмът съдържа следните стъпки:

1) Определяне на слой за начало на алгоритъма.

2) За всяка клетка от този слой се изчислява доверителния интервал (или прогнозиран такъв) на вероятността тази клетка да съответства на заявката.

3) От интервалът изчислен по-горе, всяка клетка се обозначава като подходяща или неприложима.

4) Ако този слой е най-долният, се преминава към стъпка 6, ако не – към стъпка 5.

5) Слизане с едно ниво надолу в йерархичната структура. Връщане към стъпка 2 за тези клетки, които формират подходящите такива от по-горния слой.

6) Ако спецификацията на заявката е изпълнена, се преминава към стъпка 8, ако не – към стъпка 7.

7) Извличане на данните, намиращи се в съответстващите клетки и извършване на допълнителна обработка. Връщане на резултата, който отговаря на изискванията на заявката. Преминаване към стъпка 9.

8) Намиране на области със съответстващи клетки. Тези области, които отговарят на изискванията на заявката се връщат обратно. Преминаване към стъпка 9.

9) Край.

Разработване на обобщеномрежовия модел на STING

Представената обобщена мрежа моделира процеса на прилагане на клъстерен анализ върху пространствени данни, като се използва метода *STING*. Обобщеномрежовият модел (Фиг. 17), представен на следващата страница, съдържа 6 прехода и 29 позиции. Преходите представляват следните процеси:

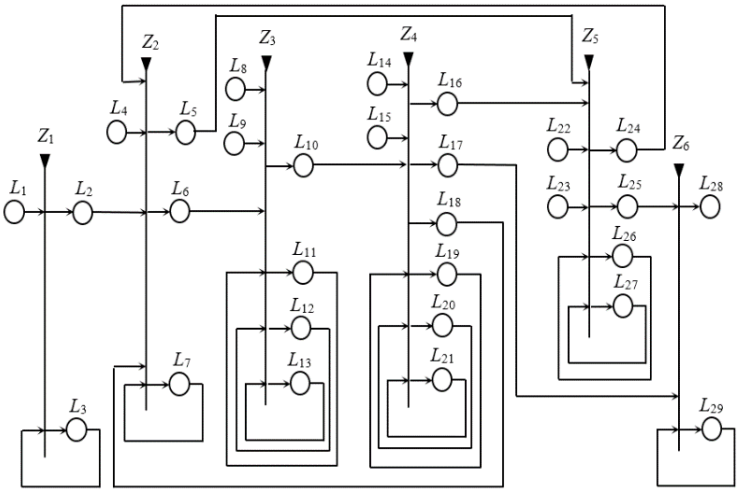
- Z_1 – „Действия с пространствени заявки“;
- Z_2 – „Действия с пространствена база от данни“;

• Z_3 – „Разделяне на пространствената площ на правоъгълни клетки, изграждайки йерархична структура и определяне на слой, от който да се стартира алгоритъма“;

• Z_4 – „Определяне на зависими и независими от атрибутите параметри за всяка клетка“;

• Z_5 – „Изчисляване на доверителния интервал за вероятност за всяка клетка, която е в съответствие със заявката“;

• Z_6 – „Извличане на данните, които се намират в съответстващите клетки и извършване на допълнителна обработка“.



Фиг. 17 Обобщена мрежа на процеса на клъстерен анализ, използващ метода STING

Преход Z_1 има следния вид:

$$Z_1 = \langle \{L_1, L_3\}, \{L_2, L_3\}, R_1, \vee(L_1, L_3) \rangle,$$

където

$$R_1 = \begin{array}{c|cc} & L_2 & L_3 \\ \hline L_1 & false & true \\ L_3 & W_{3,2} & W_{3,3} \end{array},$$

и:

• $W_{3,2}$ = „Има избрана заявка“;

• $W_{3,3} = \neg W_{3,2}$.

β_2-ядрата, които влизат в позиция L_3 от L_1 не получават нова характеристика. β-ядрото в позиция L_3 генерира ново такова, което влиза в позиция L_2 с характеристика:

„избрана пространствена заявка“.

Всяко ново α_2-ядро навлиза в мрежата през позиция L_4 с начална характеристика:

„пространствени данни“.

.
.
.

Преходът Z_6 има следния вид:

$$Z_6 = \langle \{L_{25}, L_{17}, L_{29}\}, \{L_{28}, L_{29}\}, R_6, \vee(L_{25}, L_{17}, L_{29}) \rangle,$$

където

$$R_6 = \begin{array}{c|cc} & L_{28} & L_{29} \\ \hline L_{25} & false & true \\ L_{17} & false & true \\ L_{29} & W_{29,28} & W_{29,29} \end{array},$$

и:

- $W_{29,29}$ = „Има данни за извличане или се извършва допълнителна обработка“;
- $W_{29,28} = \neg (W_{29,29})$.

α-ядрата, които влизат в позиция L_{29} не придобиват нови характеристики. Те генерират ново α-ядро, което влиза в позиция L_{28} с характеристика:

„резултат от пространствената заявка“.

3.6 Извод

В Глава трета от дисертационния труд са разработени и представени обобщеномрежови модели на алгоритми за клъстеризация (CLIQUE, йерархичен клъстерен анализ, STING). Освен това е направена програмна симулация на K-means алгоритъма в MATLAB.

Обобщеномрежовият модел на метода *CLIQUE* се използва за анализ и наблюдение на поведението му. Симулацията чрез ОМ отразява паралелната работа и позволява нейния анализ и контрол. Изграденият обобщеномрежов модел може да способства за постигането на по-нататъшни подобрения на реалния процес.

Изграденият обобщеномрежов модел на процеса на йерархичен клъстерен анализ може да бъде полезен за анализирането, управлението и оптимизацията на йерархичното клъстеризиране.

В MATLAB е реализирана симулация на *K-means* алгоритъма, като идеята е да се съпостави работата му с тази на вече описаната йерархичната клъстеризация. Целта е да се проследи поетапно процесът на работа на алгоритъма като се използват графични елементи за онагледяването на действието му. Демонстрирани са два варианта – със случайни множества от числа и с таблични данни за пациенти.

Посоченият в текущата глава вид клъстеризация е един от основните методи за клъстеризиране, прилагани в пространственото извличане на знания. *STING* представлява статистически базиран подход за проучване на пространствени бази от данни. Той изгражда йерархична структура на клетките и ги изследва за клъстери. Изграденият обобщеномрежов модел може да се използва за анализ и наблюдение на процеса на клъстеризация на алгоритъма *STING*.

Приноси към дисертационния труд

Приносите в настоящия дисертационен труд са научни и научно-приложни.

Към *научните* приноси се причисляват:

– Създадени са обобщеномрежови модели на действието на следните алгоритми за извличане на знания: MapReduce, Deep Learning невронна мрежа и Stochastic Expectation-Maximization;

– Разработени са обобщеномрежови модели на процеса на клъстерен анализ като следва: по CLIQUE (клъстеризация в QUEst), йерархична клъстеризация, клъстеризация по метода STING. В последствие изпълнението на алгоритмите може да бъде категоризирано чрез обща обобщена мрежа;

Научно-приложните приноси към дисертационния труд са:

– На базата на изчислителните възможности на MapReduce алгоритъма и създадения обобщеномрежов модел са реализирани две тествания, свързани с визуализацията на вероятност чрез логистична регресия в MATLAB. Използваните данни са за пациенти на болница.

– В MATLAB е реализирана е графична програмна демонстрация на действието на k-means алгоритъма, като по този начин може да се направи паралел между него и йерархичната клъстеризация.

Насоки за бъдещи изследвания

При изготвянето на дисертационния труд се появиха редица идеи за **бъдещи изследвания**:

1. Изследване на възможности за конструиране на общ обобщеномрежов модел на процеса на клъстеризация, с възможност за избор на клъстеризиращ алгоритъм в зависимост от различни изследователски задачи;
2. Тестване на моделите със симулатор на обобщени мрежи.
3. Разширяване на моделите с интуиционистки размита логика.

Публикации по дисертационния труд

1*. Bureva, V., **S. Popov**, E. Sotirova, K. Atanassov, Generalized Net of MapReduce Computational Model, IWIFSGN 2016: Uncertainty and Imprecision in Decision Making and Decision Support: Cross-Fertilization, New Models and Applications, Advances in Intelligent Systems and Computing, Vol. 559, 2018, pp. 305-315 (SJR=0.174)

2*. Sotirov, S., E. Sotirova, A. Shannon, V. Bureva, T. Petkov, **S. Popov**, H. Bozov, D. Tsolova, V. Georgieva, A Generalized Net Model of the Deep Learning Neural Network, ANNA '18: Advances in Neural Networks and Applications 2018, September 15-17, 2018, St. St. Konstantin and Elena Resort, Bulgaria, ISBN 978-3-8007-4756-6, pp. 64-67.

3*. Bureva, V., E. Sotirova, **S. Popov**, D. Mavrov, V. Traneva, Generalized Net of Cluster Analysis Process Using STING: A Statistical Information Grid Approach to Spatial Data Mining, International Conference on Flexible Query Answering Systems, Lecture Notes in Computer Science, Vol. 10333, 2017, pp. 239-248 (SJR=0.315)

4*. Bureva, V., **S., Popov**, E. Sotirova, B. Miteva, Generalized Net Of The Process Of Hierarchical Cluster Analysis, Annual Of Assen Zlatarov University, Burgas, Bulgaria, V. XLV, 2017, pp. 107-111.

5*. **Popov, S.**, Generalized Net Model of Stochastic Expectation-Maximization Algorithm, Annual Of Assen Zlatarov University, Burgas, Bulgaria, Vol. XLVIII, Book 1, 2019, pp. 86-89.

6*. V. Bureva, **S. Popov**, V. Traneva, S. Tranev, Generalized Net Model of Cluster Analysis Using CLIQUE: Clustering in Quest, June 2019, International Journal Bioautomotion 23(2), DOI: 10.7546/ijba.2019.23.2.000506, pp. 131-138 (SJR=0.267)

Цитати

1. Bureva, V., S. Popov, E. Sotirova, K. Atanassov, Generalized Net of MapReduce Computational Model, IWIFSGN 2016: Uncertainty and Imprecision in Decision Making and Decision Support: Cross-Fertilization, New Models and Applications, Advances in Intelligent Systems and Computing, Vol. 559, 2018, pp. 305-315

1.1. Ai S., Rong C., Cao J. (2020) Utilization of Big Data in Energy Internet Infrastructure. In: Zobaa A., Cao J. (eds) Energy Internet. Springer, Cham;

1.2. Zoteva, D., Krawczak, M., Generalized Nets as a Tool for the Modelling of Data Mining Processes. A Survey, Issues in IFSs and GNs, Vol. 13, 2017, pp. 1–60.

2. Sotirov, S., E. Sotirova, A. Shannon, V. Bureva, T. Petkov, S. Popov, H. Bozov, D. Tsolova, V. Georgieva, A Generalized Net Model of the Deep Learning Neural Network, ANNA '18: Advances in Neural Networks and Applications 2018, September 15-17, 2018, St. St. Konstantin and Elena Resort, Bulgaria, ISBN 978-3-8007-4756-6, pp. 64-67.

2.1 Y. Yang and Y. Yang, "Hybrid Method for Short-Term Time Series Forecasting Based on EEMD," in *IEEE Access*, vol. 8, pp. 61915-61928, 2020.

3. Bureva, V., E. Sotirova, S. Popov, D. Mavrov, V. Traneva, Generalized Net of Cluster Analysis Process Using STING: A Statistical Information Grid Approach to Spatial Data Mining, International Conference on Flexible Query Answering Systems, Lecture Notes in Computer Science, Vol. 10333, 2017, pp. 239-248

3.1.Zoteva, D., Krawczak, M., Generalized Nets as a Tool for the Modelling of Data Mining Processes. A Survey, Issues in IFSs and GNs, Vol. 13, 2017, pp. 1–60.

3.2.Videv, Tihomir; Bozveliev, Boris; Sotirov, Sotir, Modelling of Smart Home Cyber System with Intuitionistic Fuzzy Estimation, Information & Security, 2019, Vol. 43, pp. 45-53.

3.3.Videv T., Sotirov S., Bozveliev B. (2020) Generalized Net Model of the Network for Automatic Turning and Setting the Lighting in the Room with Intuitionistic Fuzzy Estimations. In: Castillo O., Melin P., Kacprzyk J. (eds) Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications. Studies in Computational Intelligence, vol 862. Springer, Cham.