Multidiszciplináris tudományok, 11. kötet. (2021) 4 sz. pp. 81-86 **https://doi.org/10.35925/j.multi.2021.4.9**

KLASSZIKUS KLASZTEREZŐ ALGORITMUSOK MÓDOSÍTÁSA KÖRÚT ALAPON

Agárdi Anita
tanársegéd, Miskolci Egyetem, Informatika Intézet, Általános Informatikai Intézeti Tanszék
3515 Miskolc, Miskolc-Egyetemváros, e-mail: agardianita@iit.uni-miskolc.hu

Absztrakt

Jelen cikkben a klasszikus klaszterező algoritmusok egy módosítását mutatom be. A cikkben egy olyan módszert mutatok be, amellyel a klaszterező algoritmusok maguk határozzák meg a klaszterhatárokat, azt, hogy hány csoportra bontsák az adatsor elemeit. A klaszterezés egy olyan adatbányászati módszer, ahol az egymással hasonló elemek azonos klaszterbe, míg az egymástól különböző elemek külön klaszterbe kerülnek. Jelen cikkben egy partíciós algoritmust (K-Means) és a hierarchikus módszereket (Single Linkage, Complete Linkage, Average Linkage, Ward, Centroid) mutatom be. A futási eredmények azt mutatják, hogy a klaszterezési algoritmusoknak többé-kevésbé sikerült kialakítaniuk a klasztereket anélkül, hogy bemenetként a klaszterszámot várnánk.

Kulcsszavak: *klaszterezés, K-Means, hierarchikus módszer*

Abstract

In this paper, a modification of classical clustering algorithms is presented. In this paper, I present a method by which clustering algorithms themselves determine cluster boundaries, the number of groups in which to break down the elements of a data set. Clustering is a data mining method where similar elements are placed in the same cluster, while different elements are placed in a separate cluster. In this paper, a partition algorithm (K-Means) and hierarchical methods (Single Linkage, Complete Linkage, Average Linkage, Ward, Centroid) are presented. The running results show that the clustering algorithms were more or less able to form the clusters without waiting for the cluster number as input.

Keywords: *clustering, K-Means, hierarchical method*

1. Bevezetés

Jelen cikk során a klasszikus klaszterező algoritmusok módosított változatainak eredménye kerül bemutatásra. A cikkben az alábbi klasszikus klaszterező algoritmusokat alkalmaztam: K-Means, Single Linkage, Complete Linkage, Ward, Centroid.

A klaszterezés elemek csoportosítása. Azon elemek, amelyek hasonlítanak egymásra azonos klaszterbe kellene hogy kerüljenek, míg azok amelyek nagyban különböznek külön klaszterbe. A klaszterezésnek számos felhasználási területe van, például képfeldolgozás [1], diákok jegyeinek csoportosítása [2], gyorséttermek csoportosítsa [3], kémia [4] stb.

A K-Means [5] algoritmus kezdetben kiválaszt *k* darab elemet, ezek lesznek a középpontok. Majd az adatsor minden elemét a hozzá legközelebbi klaszter középponthoz társítja. Ezután újraszámolja a

klaszter középpontokat, majd újból minden elemet a hozzá legközelebbi középponthoz társít. A klaszter középpont újraszámolás és az elemek klaszter középpontokhoz rendelése iteratív folyamat, addig folytatódik amíg kis mértékű a változás. Az algoritmus hátránya, hogy sokszor lokális optimumban ragad, ezért többször is le szokták futtatni, és a legjobb eredményt szokták elfogadni.

A hierarchikus klaszterezés [6] egy fát (dendogram) épít fel. A klaszterezés során minden elem kezdetben egy-egy klasztert fog jelenteni. Ezután folyamatosan vonjuk össze az egyes klasztereket, míg minden elem egyetlen klasztert nem fog jelenteni. Az alábbi stratégiákat különböztetjük meg a klaszterek távolságára [6]:

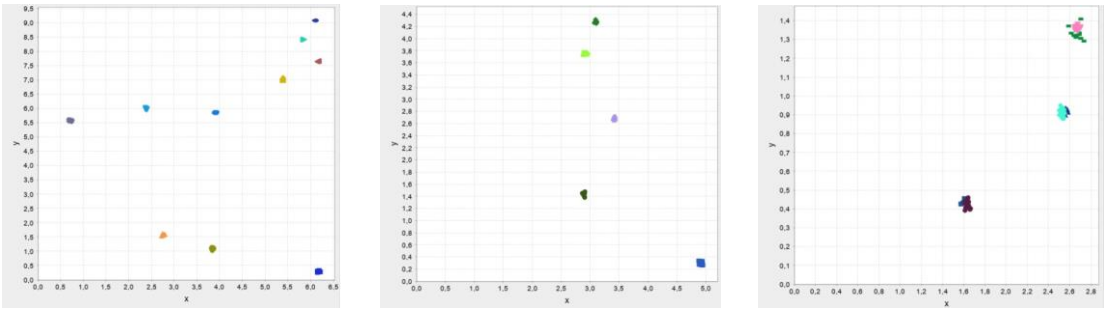
- Single Linkage: a klaszterek távolságát a két klaszter legközelebbi tagjai közti távolságként értelmezi.
- Complete Linkage: a klaszterek távolságát a két klaszter legtávolabbi tagjai közti távolságként értelmezi.
- Centroid módszer: a két klaszter távolságát a két klaszter középpontjaként értelmezi.
- Ward módszer: azt a két klasztert vonja össze, amely legkisebb négyzetes hibanövekedést okozza.
- Average Linkage: a két klaszter távolságát úgy számolja ki, hogy a két klaszter pontjainak a távolságát elosztja a klaszterek számosságának a szorzatával.

A következő fejezetben a futási eredményeket mutatom be három adatsoron keresztül.

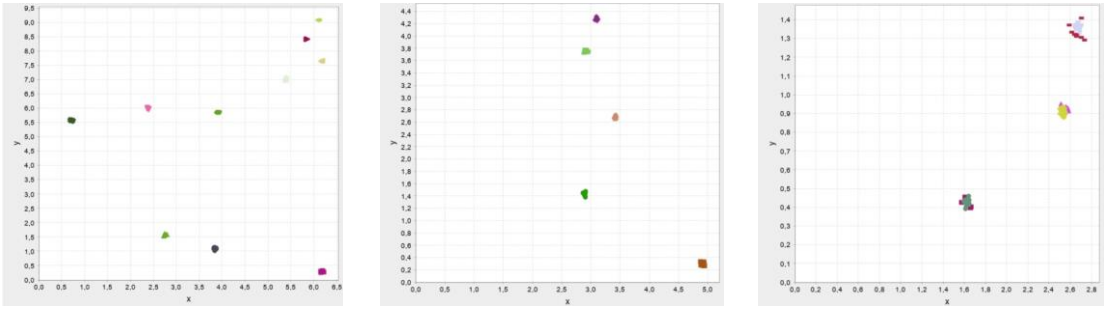
2. Futási eredmények

A klaszterező algoritmusok eredményeit úgy készítettem el, hogy kezdetben lefuttattam a klaszterezési algoritmusokat kis klaszterszámra. Ez adott egyfajta kezdeti klaszterezést, pontsorrendet (permutációt), körutat. Majd a végső klasztereket úgy alakítottam ki, hogy kiszámoltam a pontok közötti átlagos távolságokat. Sorban vettem a permutáció egyes elemeit, és azonos klaszterekhez soroltam, míg egy bizonyos távolságon belül vannak. Ezt a távolságot az átlagos távolságok negyedik gyökének vettem. Ha ezen értéktől nagyobb két pont távolsága, akkor azok külön klaszterbe fognak kerülni.

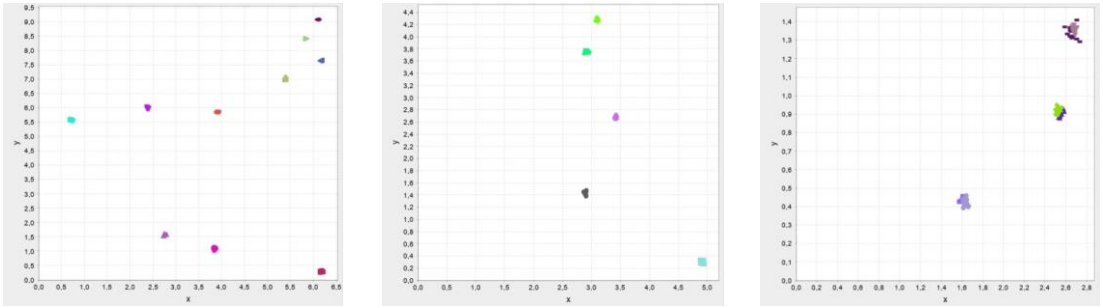
A futási eredmények során három adatsort készítettem. Az adatsorokat mesterségesen generáltam, ezek kétdimenziós jól klaszterezhető (kompakt) pontok. Az első adatsor 100 pontot tartalmaz, és az optimális eset az, ha 10 klaszterre kerül bontásra. A második adatsor 50 pontot tartalmaz, és az optimális eset az, ha 5 klaszterre kerül bontásra. A harmadik adatsor 90 pontot tartalmaz, és az optimális eset az, ha 3 klaszterre kerül bontásra.



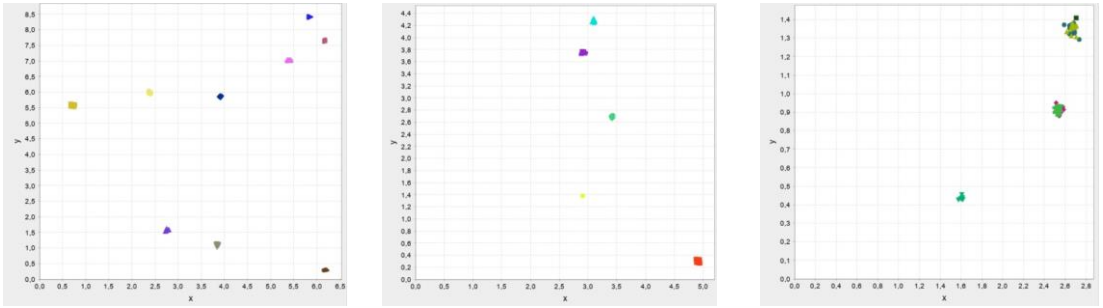
1. ábra. *Average Linkage eredménye az első, második és harmadik adatsorra*



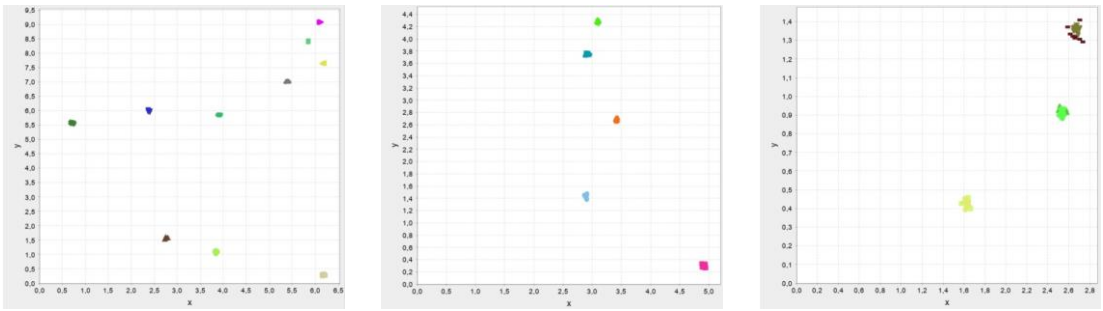
2. ábra. *Centroid módszer eredménye az első, második és harmadik adatsorra*



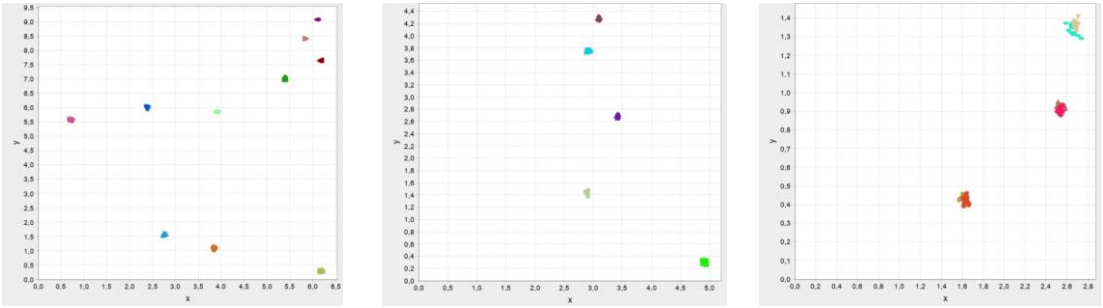
3. ábra. *Complete Linkage eredménye az első, második és harmadik adatsorra*



4. ábra. *K-Means eredménye az első, második és harmadik adatsorra*



5. ábra. *Single Linkage eredménye az első, második és harmadik adatsorra*



6. ábra. *Ward módszer eredménye az első, második és harmadik adatsorra*

Az 1-6. ábrákon látható, hogy az összes vizsgált módszer megtalálta a klaszter határokat az első és a második esetben. A harmadik esetben három klaszter helyett többet találtak az algoritmusok az ábrák alapján. Most táblázatban is bemutatom az eredményeket, amely a fitnesz értéket és a futási időt tartalmazza. A fitnesz érték az azonos klaszterbeli elemek egymástól vett távolságainak összege.

1. táblázat. *Futási eredmények az első adatsorra*

Algoritmus	Klaszterek száma	Fitnesz érték	Futási idő (perc)
Average Linkage	10	0.3495	$3.3708*10^{-4}$
Centroid módszer	10	0.3495	$6.8386*10^{-4}$
Complete Linkage	10	0.3495	$1.3987*10^{-4}$
K Means	9	0.3644	0.0011
Single Linkage	10	0.3495	$8.1784*10^{-4}$
Ward	10	0.3495	$3.1161*10^{-4}$

Az *1. táblázat* alapján elmondható, hogy a K-Means kivételével az algoritmusoknak sikerült 10 csoportra bontani az adatsort. A fitnesz értékük is egyezik. A futási idő minden esetben alacsony volt.

2. táblázat. *Futási eredmények a második adatsorra*

Algoritmus	Klaszterek száma	Fitnesz érték	Futási idő (perc)
Average Linkage	5	0.3447	$1.1716*10^{-4}$
Centroid módszer	5	0.3447	$8.8033*10^{-5}$
Complete Linkage	5	0.3447	$2.6763*10^{-5}$
K Means	5	0.3354	0.0012
Single Linkage	5	0.3447	$7.9233*10^{-5}$
Ward	5	0.3447	$4.3885*10^{-5}$

A második adatsort az összes algoritmusnak sikerült 5 csoportra bontania, ezt mutatja a *2. táblázat*. A fitnesz értékek is azonosak, tehát az algoritmusok ugyanazt a csoportosítást találták meg. A futási idő itt is nagyon alacsony volt.

3. táblázat. *Futási eredmények a harmadik adatsorra*

Algoritmus	Klaszterek száma	Fitnesz érték	Futási idő (perc)
Average Linkage	6	0.6187	$1.5920*10^{-4}$
Centroid módszer	6	0.6079	$1.8359*10^{-4}$
Complete Linkage	6	0.5452	$1.6098*10^{-4}$
K Means	6	0.5442	$8.3133*10^{-5}$
Single Linkage	6	0.6095	$5.0281*10^{-5}$
Ward	6	0.5568	$1.2962*10^{-4}$

A harmadik adatsort 3 csoportra kellett volna bontania az algoritmusoknak, ehelyett minden algoritmus 6 csoportra bontotta. A fitnesz értékek is eltérőek, a legkisebb fitnesz értékű csoportosítást a K-Means adta, így ez az algoritmus oldotta meg leghatékonyabban a harmadik adatsor klaszterezését.

3. Összegzés

Jelen cikkben a klassszikus klaszterező algoritmusok módosításait mutattam be. Az alábbi klaszterező algoritmusokat tárgyaltam: K-Means, hierarchikus klaszterezés (Single Linkage, Complete Linkage, Ward módszer, Centroid módszer, Average Linkage). Három adatsoron teszteltem az algoritmusokat. Az első két adatsorra tökéletesen klasztereztek, a harmadik adatsornál nem teljesen sikerült a klaszterhatárokat megtalálniuk az algoritmusoknak.

Köszönetnyilvánítás

„A cikkben/előadásban/tanulmányban ismertetett kutató munka az EFOP-3.6.1-16-2016-00011 jelű „Fiatalodó és Megújuló Egyetem – Innovatív Tudásváros – a Miskolci Egyetem intelligens szakosodást szolgáló intézményi fejlesztése” projekt részeként – a Széchenyi 2020 keretében – az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg."

Irodalom

[1] Koster, K., & Spann, M. (2000). MIR: An approach to robust clustering-application to range image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(5), 430-444. **https://doi.org/10.1109/34.857001**

[2] Oyelade, O. J., Oladipupo, O. O., & Obagbuwa, I. C. (2010). Application of k Means Clustering algorithm for prediction of Students Academic Performance. arXiv preprint arXiv:1002.2425.

[3] Austin, S. B., Melly, S. J., Sanchez, B. N., Patel, A., Buka, S., & Gortmaker, S. L. (2005). Clustering of fast-food restaurants around schools: a novel application of spatial statistics to the study of food environments. American journal of public health, 95(9), 1575-1581. **https://doi.org/10.2105/AJPH.2004.056341**

[4] Lawson, R. G., & Jurs, P. C. (1990). New index for clustering tendency and its application to chemical problems. Journal of chemical information and computer sciences, 30(1), 36-41. **https://doi.org/10.1021/ci00065a010**

[5] Hamerly, G., & Elkan, C. (2004). Learning the k in k-means. In Advances in neural information processing systems (pp. 281-288).

[6] Olson, C. F. (1995). Parallel algorithms for hierarchical clustering. Parallel computing, 21(8), 1313-1325. **https://doi.org/10.1016/0167-8191(95)00017-I**