Avrupa Bilim ve Teknoloji Dergisi
Sayı 23, S. 280-287, Nisan 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 23, pp. 280-287, April 2021

Research Article

K-Ortalamalar Kümeleme Yöntemi İçin Çift K Başlatma Algoritması

Aziz Mahmut Yücelen[1*], Abdullah Baykal[2]

[1*] Dicle Üniversitesi, Diyarbakır Teknik Bilimler Meslek Yüksekokulu, Diyarbakır, Türkiye, (ORCID: 0000-0001-7160-6614), ayucelen@msn.com
[2] Dicle Üniversitesi, Fen Fakültesi, Matematik Bölümü, Diyarbakır, Türkiye (ORCID: 0000-0001-8011-024X), baykal.abdullah@gmail.com

(İlk Geliş Tarihi 22 Ocak 2021 ve Kabul Tarihi 28 Mart 2021)

(**DOI:** 10.31590/ejosat.866830)

ATIF/REFERENCE: Yücelen, A. M., Baykal, A. (2021). K-Ortalamalar Kümeleme Yöntemi İçin Çift K Başlatma Algoritması. *Avrupa Bilim ve Teknoloji Dergisi*, (23), 280-287.

Öz

Veri madenciliğinin en dikkat çekici konularından biri olan kümeleme yöntemleri, bu alanın en yoğun araştırma sahası olup kümeleme üzerine bir çok teknik ve bağlı yöntemler bulunmaktadır.Bu alandaki çalışmaların bir kısmı daha önce mevcut olan algoritmaların güncellenmesiyle elde edilmiş ve performansları değerlendirilmiştir.Kümelemenin en çok ilgi duyulan konusu K-Ortalamalar yöntemidir.K-Ortalamalar algoritması her çalıştırıldığında, başlangıç merkezlerinin rastgele seçilmesi nedeniyle farklı küme çıktıları döndürür.Bu nedenle, sonuçların güvenilirliği olumsuz etkilenir ve kümeleme doğruluğu için yineleme sayısı artar.Bu yöntemin doğruluğunu arttırmaya çalışan yöntemlerden biri de K-Ortalamalar++ yöntemidir. Bu çalışmada K-Ortalamalar algoritmasının başarısının arttırılması hedeflenmiştir. Sentetik veri kümesine çift k olarak adlandırdığımız önerilen yöntem uygulanmıştır.Çift k yönteminin, nihai kümeleme etiketlerini bulmada K-Ortalamalar ve K-Ortalamalar++ yöntemine göre daha başarılı olduğu gözlenmiştir.

Anahtar Kelimeler: Veri Madenciliği, Kümeleme, Başlangıç Ağırlık Merkezleri, K-Ortalamalar, K-Ortalamalar++.

Double K Initializing Algorithm For K-Means Clustering Method

Abstract

Clustering methods which is one of the most striking subjects of data mining are the most intensive research area of this field and there are many techniques and related methods on it. Some of the studies in this field have been obtained by updating the algorithms previously available and their performance has been evaluated. The most interesting topic of clustering techniques is K-Means method. Every initializing of K-Means algorithm return different cluster outputs because of random selection of the initial centers. Therefore, the reliability of the results is adversely affected and the number of iterations increase for clustering accuracy.One of the methods that tries to increase accuracy of this method is the k-means ++ method. In this study, it was aimed to increase the success of the K-means algorithm. The proposed method that we called double k was applied to synthetic dataset. It has been observed that double k method is more successful in finding final clustering labels than the K-Means and K-Means++ methods.

Keywords: Data Mining, Clustering, Initial Centroids,K-Means, K-Means++

[*] Sorumlu Yazar: ayucelen@msn.com

1. Giriş

Veri madenciliği, ilerleyen teknolojik gelişmeler doğrultusunda artan veriler açısından her geçen gün ihtiyacın hissedildiği bir alan olmaktadır. Tıbbi görüntüleme ve teşhis alanındaki çalışmalardan otonom araç altyapısına, biyolojiden finansal uygulamalara kadar görüntü, sinyal ve veri işleme adına birçok disiplin tarafından sıkça kullanılan bir alan olarak karşımıza çıkmaktadır. Veri madenciliği temel olarak veri yığınlarından anlamlı bilgiler çıkarmayı esas alır ve çıkarılacak sonuçlar bir çok disiplin tarafından anlamlandırılarak teknolojiye katkı sunar. Yığınlardan kullanışlı bilgilerin ortaya çıkarılması için birçok teknik ve yöntem bulunmaktadır. Her teknik, işlenmesi gereken verilerin türüne uygun olarak seçilir. Kullanılan bu tekniklerden biri de kümelemedir.

Kümeleme, tanım olarak bu alanda çalışan araştırmacılar tarafından değişik şekillerde ifade edilse de temel olarak herhangi bir anlamsız ham veri kümesinin, uygun istatistiksel kriterler, benzerlik ve benzemezlik ölçütleri ile önceden belirlenmiş sayıda veri gruplarına ayırılması işlemi olarak tanımlanabilir. Bir anlamda, bir dizi benzer nesneyi aynı gruplara ve benzer olmayan nesneleri farklı gruplara parçalama tekniği olarak kullanılır [1]. Böylece işlemin kendisi, belirli bir benzerlik kriterine göre bir uzay içinde bulunan bir kümeye ait parçalanmayı kapsar [2]. Bu haliyle veri nesnelerinin örüntülerin gruplara gözetimsiz yani etiketsiz bir sınıflandırması da gerçekleşmektedir [3]. Birçok branştan araştırmacı, kümeleme probleminin güçlüklerini ortadan kaldırmak için yeni yöntemler ve eniyilemeler geliştirmektedir. Ham yığınlar içindeki nesnelerin yoğunlukları diğer bölgelerde bulunanlara göre yerel veya global olarak daha yüksek veya düşük olabilir [4]. Bu alandaki çalışmalarda temel hedef üretilen sonuçlarda yerel minimumdan çok, global minimuma erişmektir. Kümeleme, birçok disiplinde kullanılan makine öğrenmesi, istatistik, veri madenciliği ve örüntü tanıma gibi birçok alanda bilginin ortaya çıkarılmasına zemin oluşturmuştur [5]. Mühendislikte makina öğrenmesi, örüntü tanıma, biyometrik tanıma ve sinyal analizi, bilgisayar bilimlerinde görüntü segmentasyonu ve veri analizi, tıbbi bilimlerde gen ve protein tanımlanması, hastalık teşhis ve tedavisinde, yer bilimleri ve astronomi bilimlerinin gezegen ve yıldız keşiflerinde, finans ve ekonomi bilimlerinde müşteri ve firma profillerinin çıkarılması gibi alanlara ek olarak sosyoloji, arkeoloji ve psikoloji gibi alanlarda da yaygın bir şekilde kullanılmaktadır [6]. Kümeleme alanında başlangıçta biri birinden bağımsız çalışmaların olması ve biri birinden habersiz benzer algoritmalar geliştirilmesine rağmen son zamanlarda bu çalışma alanı oldukça popüler olmaktadır.

Kümeleme teknikleri içinde kullanılan yöntemlerden biri de K-Ortalamalar (KO) yöntemidir. Etkili bir algoritma olup düşük boyutlu ve büyük veri kümelerinde çalışan birçok uygulamada kullanılır [7].Mac Queen [11] tarafından geliştirilen en çok kullanılan kümeleme algoritmalar içinde klasik olanlardan biridir [8]. KO, kümelemenin gözetimsiz öğrenme işlevini yerine getirmenin yanı sıra sıkça kullanılmaktadır [9][10]. Zaman karmaşıklığı düşük olduğundan dolayı programlanması oldukça kolay olan ve temel olarak kümeleme içi varyansların düşük tutulduğu etkili kümelemeler oluşturabilmektedir [11]. Karmaşıklık kavram olarak, yönteme ait algoritmanın çalışma süresi ve üzerinde çalıştırılan bilgisayarda kullandığı hafıza kaynağı olarak ifade edilmektedir. KO yöntemi de, zaman karmaşıklığın düşük olduğu algoritmalar arasında yer almaktadır. Farklı kümeleme algoritmaları arasında basitliği nedeniyle popüler olarak kabul edilir ancak algoritmadan elde edilen sonuç, ilk ağırlık merkezi seçimine oldukça duyarlıdır. Sonuç olarak, KO algoritmasında başlangıç ağırlık merkezlerinin seçimi, doğruluk ve verimlilikte önemli bir rol oynar [1]. Bu nedenle en iyi sonucun elde edilmesi için 20 defa rastgele başlangıç ile çalıştırması önerilir [12]. Temel bileşenler analizi kullanılarak veri kümeleri üzerinde yapılan boyut azaltma işlemi yöntemdeki başarıyı olumlu yönde etkileyebilmektedir [9].Sonuç olarak KO algoritması doğruluğu garanti etmese de hız açısından oldukça tatmin edicidir [13]. KO algoritmasının doğruluğunu arttırmaya yönelik çalışmalardan biri de Arthur ve Vassilvitskii (2006) [13] tarafından önerilen KO++ yöntemidir. KO++'da, KO yönteminin başlangıç ağırlık merkezi seçimi için olasılık temelli bir yöntem kullanılmıştır [39]. Bu çalışmada, istenen kümeleme sayısının iki katı ile işlem yapıldığından dolayı 'Çift K' (ÇK) olarak adlandırdığımız ve sentetik veri kümesi üzerinde k-ortalama++ yöntemini de kullanan bir dizi işlemden sonra nihai ağırlık merkezleri elde edilmiştir. Bu nihai ağırlık merkezleri ile KO yöntemi çalıştırılmış ve yöntemin KO (rasgele) ve KO++ yöntemlerine göre etiketleme ve toplam karesel hata açısından başarısı incelenmiştir. Kümeleme ölçümleri ile ÇK yönteminin daha başarılı olduğu görülmüş ve elde edilen bulgular çalışmada sunulmuştur.

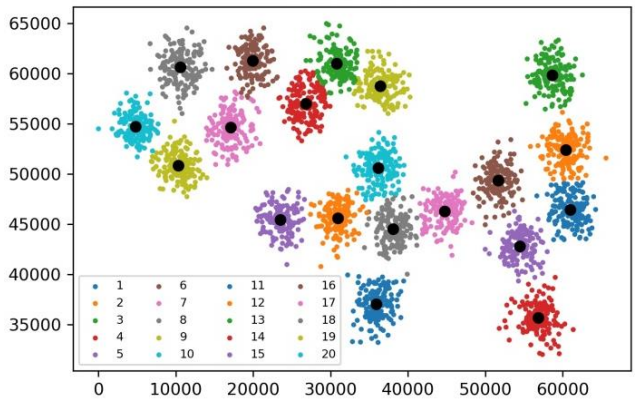
2. Materyal ve Metot

KO yöntemi, Jain'e [34] göre biribirinden bağımsız olarak farklı zamanlarda çok benzer yöntemler kullanılarak Lloyd [14] ve MacQueen [11] tarafından keşfedilmiş olup, günümüzde en yaygın kullanılan bir yöntemdir. Astrahan [2] konuşma dalga formu verilerindeki bir ifadeyi bölümlere ayırmak için her bir kümelemeye ait başlangıç ağırlık merkezi seçiminin rasgele olması yerine en yakın komşuluk yoğunluğuna dayalı olan bir KO yöntemi önermiştir. Lloyd [14] en küçük kareler ile kuantalama işleminde kullanılmak üzere bir ilklendirme yöntemi önermiştir. Bu yönteme göre ilklendirme, verinin rasgele bir alt kümesi üzerinde belirli bir "K" parametresi için KO yönteminin çalıştırılarak elde edilir. Bu algoritma N kez rasgele alt küme üzerinde KO uygulanarak çalıştırılır ve sonuç olarak N*K tane veri son giriş verisi olarak kullanılarak nihai kümeleme merkezleri elde edilmiş olunur. Katsavounidis ve ark. [15] genelleştirilmiş Lloyd [14] algoritmasından daha iyi bir yerel minimum elde edebilecek iyi bir başlangıç kümeleme elde etmek için bir ilklendirme yöntemi önermişlerdir. Bu yönteme göre N tane başlangıç vektörü elde etmek üzere ilk vektör veri kümesinin kenarından seçilir, ardından bu vektöre en uzak mesafedeki vektör ikinci vektör olur. Bu şekilde istenilen N vektör elde edilene kadar iterasyonel olarak devam edilir. Bradley ve Fayyad [16] KO benzeri tekniklerin başlangıç koşullarına olan hassasiyeti ortadan kaldırmayı hedeflemişlerdir. Buna yönelik olarak, verilen bir başlangıç koşulundan rafine bir başlangıç koşulunu hesaplayan bir yöntem önermişlerdir. Bu önerilen yöntem ile elde edilen rafine başlangıç şartları, algoritmalarının yerel minimumların en iyisine yakınsamasını sağlayarak KO kümeleme yöntemine ait çözümleri iyileştirdiğini bildirmişlerdir. Likas ve ark. [17] k-ortalamalar yönteminin kronik başlangıç noktalarına bağımlı olma problemini ortadan kaldırmak için başlangıç noktalarından bağımsız olan ve "global k-means (GKM)" olarak adlandırdıkları yöntemlerini önermişlerdir. Gan ve ark. [18], Xie ve ark. [19], Beigi [20], Agrawal ve ark. [21] belirlenimci olmayan GKM yönteminin başlangıç şartlarından bağımsız bir yöntem olarak kümeleme

problemine bir çözüm olduğunu ancak zaman karmaşıklığından dolayı orta ve büyük ölçekli veriler için uygulanabilir olmadığını bildirmişlerdir. Redmond ve Heneghan [5] KO kümeleme algoritmasının başlangıç merkez noktalarını belirlemek için verilerin yoğunluk tahminini gerçekleştirmek üzere bir kd-ağacının kullanıldığı bir yöntem önermişlerdir. Rani ve Parthipan [22] KO algoritmasının başlangıç ağırlık merkezi noktalarının belirlenmesinde, geliştirilmiş parçacık sürü optimizasyonu (IPSO - Improved Particle Swarm Optimization) olarak adlandırdıkları yöntemlerini önermişlerdir. Önerilen bu yöntem ile, bilinen KO algoritmasına göre daha doğru sonuçlar üreten çözümler sağlandığını bildirmişlerdir.Li ve ark. [23] verileri sınıflandırmak için kısaca AKM (Adaptive KMeans) olarak adlandırdıkları ve kümeleme sayısının otomatik belirlendiği KO temelli yeni bir yöntem önermişlerdir. Bu yöntemde KO yönteminin başlangıç şartları probleminden dolayı kısaca KMRIC (K-Means Refined Initial Centers) olarak adlandırdıkları ve KO algoritmasının başlangıç merkezini seçimini güçlendirecek, Bradley ve Fayyad'ın [16] çalışmasında kullandıkları yönteme benzer bir yöntem kullanmışlardır. Zahra ve ark. [8] KO kümeleme tabanlı bir öneri algoritması önermişlerdir. Yöntemin ağırlık merkezlerinin rasgele seçildiği, bilinen ağırlık merkezi seçim yöntemlerine göre daha iyi doğruluk ve performans gösterdiğini ve bu yöntem ile var olan yaklaşımlardan daha iyi kümeleme ve hızlı yakınsaklık elde edilmesine ek olarak sağlanan tavsiyenin doğruluğunun arttığını bildirmişlerdir. KO işlem olarak öklid uzaklığının karesini kullandığından dolayı her bir kümeleme merkezlerine uzak olan gürültülü verilerin, yöntemin ürettiği sonuç üzerinde etkisi oldukça fazladır. Bu da yöntemin gürültülü verilere karşı yüksek bir hassasiyeti olduğunu göstermektedir [24]. Her ne kadar KO++ yöntemi KO yöntemini güçlendirse de istenilen seviyede değildir.

2.1. Veri Kümesi

Önerilen yöntemde Şekil 1.'de gösterilen, kümeleme problemine yönelik Kärkkäinen ve Fränti'nin [25] geliştirdikleri algoritmanın testi için kullandıkları sentetik 'A' veri kümesi grubu içinden iki boyutlu (d=2) ve 3000 (N) veri noktasından oluşan 'A1' veri kümesi kullanılmıştır. Bu veri kümesi, 20 kümelemeli (k=20) olup her bir kümelemede 150 veri noktası bulunmaktadır.



Şekil 1. A1 veri kümesi ve hesaplanmış ağırlık merkez noktaları

2.2. İstatistiksel Kümeleme Başarı Ölçümleri

Her bir sınıflandırma yönteminde, algoritma çıktılarının değerlendirilmesi için sınıflandırmanın doğruluğunun ölçülmesi gerekmektedir. Böylece harici kıstaslar ile sonuçlar arasındaki karşılaştırmanın bir ölçümü yapılabilir [29].Kümeleme bölümlerinin doğruluğu, oluşan kümelerin karşılaştırma veri kümesindeki sınıflarla ne derece eşleştiğinin ölçülmesiyle değerlendirilebilir [27].Geliştirilen yöntemler literatürde yerini alsa da her bir yöntemin diğer yöntemlere göre avantaj ve dezavantajları bulunmaktadır. Karşılıklı bilgi yöntemi (MI,) iki rasgele değişken arasındaki karşılıklı bağımlılığın olup olmadığının bir ölçüsünü verir. Buna göre eğer MI(X,Y)=0 ise X ve Y istatistiksel olarak bağımsızdır yani Y'de X'e dair herhangi bir bilgi yoktur [30][31]. Makine öğrenmesinde ise kümeleme ve sınıflama alanında, veri kümelerindeki farklı bölümleme veya etiketlemelerin benzerliğinin ölçümünde oldukça sık kullanılan bir yöntemdir [32]. Rand [26], iki farklı kümelemenin benzerliğin ölçümüne dayalı ve Rand indeksi olarak adlandırılan bir yöntem önermiştir. Campello'ya [27] göre Rand indeksi(RI), aynı veri kümesine bir kümeleme veya sınıflandırma yöntemi uygulanmasıyla elde edilen farklı etiketlemelerin başarı karşılaştırma ve değerlendirilmesi için genel bir kıstas tanımlamaktadır. Düzeltilmiş Rand İndeksi (ARI), bir kümeleme ve dışarıdan bir yöntem ile belirlenen kıstaslar tarafından oluşturulan iki bölümleme arasındaki benzerliğin ölçüsü olarak kümeleme doğrulama işlemlerinde sıklıkla kullanılır [28]. Bu çalışmada, önerilen yöntemin ne derece başarılı olduğunun tespiti için kümeleme başarı ölçümünde kullanılan bilinen bu üç yöntem kullanılmıştır.

2.2.1. Karşılıklı Bilgi (Mutual Information - MI)

Newman ve ark.'a [32] göre MI'nin bazı eksiklikleri olsa da en yaygın kullanılan bir yöntemdir. Kökeni Shannon'nun [41] çalışmasına dayanan bu yöntem için X ve Y ayrık rasgele değişkenleri arasındaki karşılıklı bilgi, aşağıdaki şekilde tanımlanabilir [40].

Tanım: X, Y $x_i \in U_X$ ve $y_j \in V_Y$ olacak şekilde iki rasgele değişken, $P(X = x_i)$ ve $P(Y = y_j)$ marjinal olasılık yoğunluk fonksiyonu ve $P(X = x_i, Y = y_j)$ ortak olasılık yoğunluk fonksiyonu olmak üzere, X, Y rasgele değişkenleri ile gösterilen iki örüntü arasındaki Karşılıklı Bilgi (MI) ölçüsü;

$$MI(X,Y) = \sum_{i}^{|U_X|} \sum_{j}^{|V_Y|} P(X = x_i, Y = y_j) \log \frac{P(X = x_i, Y = y_j)}{P(X = x_i)P(Y = y_j)}$$

olarak tanımlanır.

2.2.2. Rand İndeksi (RI)

Rand indeksi [26], sınıflandırılmış veriler için bir değerlendirme kriteriteridir ve sınıflandırmanın performansını ölçer [27]. Bu doğrultuda RI aşağıdaki şekilde tanımlanabilir [26][27][28].

Tanım: $H = \{h_1, h_2, \ldots h_n\}$, n elemanlı bir küme verilsin ve H kümesi, p ve q alt kümeye bölümlensin. Bunlardan ilki $M_i = \{\exists h_k\}$ $(1 \le i \le p \; ve \; 1 \le k \le p \; için \; \bigcup_k^p M_k = H$ ve $\bigcap_k^p M_k = \emptyset)$ şeklindeki $M = \{M_1, M_2, \ldots, M_p\}$ bölümlemesi ve sonuncusu $R_i = \{\exists h_k\}$ $(1 \le i \le q \; ve \; 1 \le k \le q \; için \; \bigcup_k^q R_k = H$ ve $\bigcap_k^q R_k = \emptyset)$ şeklindeki $R = \{R_1, R_2, \ldots, R_q\}$ bölümlemesi olsun. $\forall i \neq j$ için $1 \le i, j \le n$, $\forall u \neq v$ için $1 \le u, v \le p$ ve $\forall z \neq s$ için $1 \le z, s \le q$ olmak üzere,

$\breve{H} = \{(h_i, h_j) | h_i, h_j \in M_u \wedge h_i, h_j \in R_z\}$ olmak üzere $a = |\breve{H}|$

$\breve{H} = \{(h_i, h_j) | h_i \in M_u, h_j \in M_v \wedge h_i \in R_z, h_j \in R_s\}$ olmak üzere $b = |\breve{H}|$

$\breve{H} = \{(h_i, h_j) | h_i, h_j \in M_u \wedge h_i \in R_z, h_j \in R_s\}$ olmak üzere $c = |\breve{H}|$

$\breve{H} = \{(h_i, h_j) | h_i \in M_u, h_j \in M_v \wedge h_i, h_j \in R_z\}$ olmak üzere $d = |\breve{H}|$

şeklindeki matematiksel a, b, c, d ifadeleri ile RI,

$$RI = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} = \frac{a+b}{n(n-1)/2}$$

olarak tanımlanır.

Tanıma göre 'a' ifadesi, eleman çiftlerinden her bir elemanın her iki farklı bölümlemedeki aynı alt kümede bulunanların sayısını ve 'b' ifadesi, eleman çiftlerinden her bir elemanın farklı bölümlemelerdeki farklı alt kümelerde bulunanların sayısını ifade etmektedir. Ayrıca 'c' ifadesi, eleman çiftlerinden her bir elemanın ilk bölümlemede aynı ancak ikinci bölümlemede farklı alt kümelerde bulunanların sayısını ve son olarak 'd' ifadesi, eleman çiftlerinden her bir elemanın ilk bölümlemede farklı ancak ikinci bölümlemede aynı alt kümede bulunanların sayısını ifade etmaktedir.

2.2.3. Düzeltilmiş Rand İndeksi (ARI)

Santos ve Embrechts'e [28] göre, Rand indeksinin geliştirilmiş bir versiyonu olan ve Hubert ve Arabie [33] tarafından önerilen Düzeltilmiş Rand İndeksi (ARI), bölümlemeler arasındaki uyumu ölçer ve RI'nın sabit değerler almayan beklenen değerini [0,1] kapalı aralığına indirger. Santos ve Embrechts [28] ARI yöntemini aşağıdaki şekilde ifade etmiştir.

Tanım: $H = \{h_1, h_2, \ldots . h_n\}$, n elemanlı bir küme verilsin. H kümesini r alt kümeye bölümleyen $U = \{U_1, U, \ldots, U_r\}$ ve c alt kümeye bölümleyen $V = \{V_1, V_2, \ldots . V_c\}$ kümeleri $\cup_{i=1}^{r} U_i = H = \cup_{j=1}^{c} V_j$ ve $\cap_{i=1}^{r} U_i = \emptyset = \cap_{j=1}^{c} V_j$, $1 \le i \le r$ ve $1 \le j \le c$ özellikleri sağlayan iki bölümleme olsun.

Tablo 1. U ve V bölümlenmelerinin karşılaştırmalarına ait uygunluk tablosu

U/V	v_1	v_2	...	v_c	Toplam
u_1	$y_{1\,1}$	$y_{1\,2}$	...	$y_{1\,c}$	p_1
u_2	$y_{2\,1}$	$y_{2\,2}$	...	$y_{2\,c}$	p_2
⋮	⋮	⋮	⋱	⋮	⋮
u_r	$y_{r\,1}$	$y_{r\,2}$	...	$y_{r\,c}$	p_r
Toplam	q_1	q_2	...	q_c	n

$y_{t,k}$ ifadesi H kümesindeki her bir elemandan, U bölümlemesinin hem t alt kümesinde ve hem de V bölümlemesinin k alt kümesinde bulunan elemanların sayısını göstersin. Tablo 1. deki uygunluk tablosuna göre ARI,

$$ARI = \frac{\sum_{ij}\binom{y_{ij}}{2} - \left[\sum_i \binom{p_i}{2} \cdot \sum_j \binom{q_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{p_i}{2} + \sum_j \binom{q_j}{2}\right] - \left(\left[\sum_i \binom{p_i}{2} + \sum_j \binom{q_j}{2}\right] / \binom{n}{2}\right)}$$

denklemiyle tanımlanır.

2.3. K-Ortalamalar Yöntemi

$\forall x_i \in \mathbb{R}^n$, $i = 1 \ldots . n$ için $X = \{x_1, x_2, x_3, \ldots . x_m\}$ bir küme ve $k \in \mathbb{Z}^+$ kümeleme sayısını göstersin. $C = \{C_1, C_2, C_3, \ldots . C_k\}$, X kümesinin bir k kümelemesi, c_k ise C_k kümesinin denklem (1) ile ifade edilen ağırlık merkezi olsun. Denklem (2)'deki $\|.\|$ ifadesi bu yöntemde en çok tercih edilen öklid normudur [38]. Öklid uzaklığının kullanıldığı toplam karesel hata fonksiyonu denklem (2)'de gösterildiği üzere ifade edilir [35].

$$c_k = \frac{\sum_{x_p \in C_k} x_p}{|C_k|} \quad (1)$$

$$TKH = \sum_{i=1}^{K} \sum_{x_p \in X} \|x_p - c_i\|^2 \quad (2)$$

Aggarwal ve Reddy [38] ve Han ve ark.'a [37] göre denklem (2)'yi minimize eden çözümler, denklem (1) de ifade edilen ağırlık merkezlerinin kendisi, yani kümedeki elemanların aritmetik ortalaması olup, her bir iterasyon monoton azalandır ve nihayetinde yerel minimuma yakınsayacaktır. Yönteme ait algoritma Aggarwal ve Reddy[38] ve Bramer [36] tarafından şu şekilde özetlenmiştir.

1. X kümesi içinden rasgele k tane ağırlık merkez noktaları seçilir.
2. X kümesinin herbir elemanı, en yakın olduğu kümeleme merkezine göre, ilgili kümelemeye atanarak k tane kümeleme elde edilir.
3. Oluşan her bir kümelemenin yeni ağırlık merkezi hesaplanır.
4. Adım 2 ve 3 işlemleri ağırlık merkezleri değişmeyene kadar tekrarlanır.

2.4. K-Ortalamalar++ Yöntemi

Arthur ve Vassilvitskii (2006) [13] tarafından önerilen KO++ yöntemi, KO yönteminin kullanıldığı ve olasılık temelli bir yöntemdir [39]. $D(x)$, bir $x \in X$ noktasının en yakın ağırlık merkezine olan uzaklığını göstermek üzere Arthur ve Vassilvitskii (2006) [13] tarafından önerilen KO++ yöntemi aşağıdaki adımlardan oluşur;

1. "k" tane seçilecek ağırlık merkezlerinden ilki olan c_1, X kümesi içinden rasgele seçilir.

2. Geri kalan "k-1" ağırlık merkezi c_i $(1 \le i \le k-1)$olmak üzere, $\frac{D(\hat{x})^2}{\sum_{x \in X} D(x)^2}$ olasılıklı $\hat{x} \in X/\{c_i\}$ için $c_i = \hat{x}$ şeklinde seçilir.

3. Adım 2, $'k-1'$ kez tekrarlanarak Adım 1'deki ağırlık merkezi ile birlikte toplamda $'k'$ adet başlangıç ağırlık merkezi noktası elde edilmiş olunur.

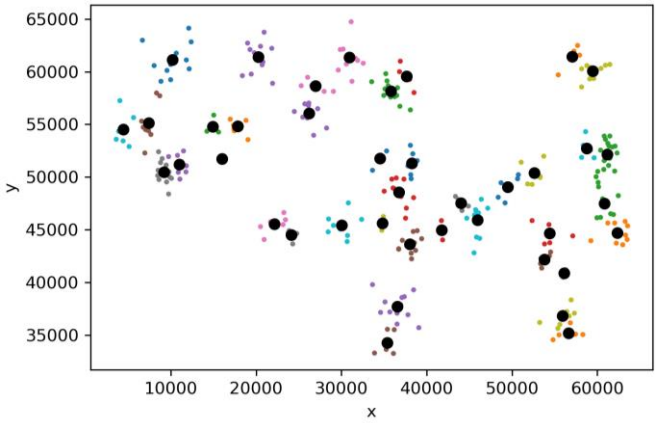
4. KO yöntemi, Adım 3 te elde edilen başlangıç ağırlık merkezleri ile başlatılarak nihai kümeleme ve kümeleme merkezlerine ulaşılır.

2.5. Önerilen Yöntem

k $\in \mathbb{Z}^+$ önceden seçilen kümeleme sayısı olmak üzere $X = \{x_1, x_2, x_3, \ldots . x_m\}$, $|X| \geq 30 * k$ koşulunu sağlayan bir küme ve $\forall x_i \in \mathbb{R}^n$ olsun. Buna göre KO yöntemini başlatacak önerilen ağırlık merkezi elde etme algoritması aşağıdaki adımlardan oluşmaktadır.

1. $Z \subset X$ ve $|Z| = |X|/10$ olacak şekilde rasgele Z örneklem kümesi seçilir.
2. Alınan Z örneklem kümesi üzerinde, kümeleme sayısı $2 * k$ seçilerek KO++ çalıştırılır ve $|Y| = 2 * k$ elemanlı nihai kümeleme merkez noktalarının kümesi olan $Y = \{y_1, y_2, y_3, \ldots . y_{2*k}\}$ elde edilir.
3. $\forall y_i \in Y$ için $Y/\{y_i\}$ kümesi üzerinde $\underset{j}{\operatorname{argmin}} \left(d(y_i, y_j)\right)$ indeksli eleman olan y_j bulunur ve yeni kümeleme merkez noktalarının kümesi olan $\breve{Y} = Y/\{y_j\}$ elde edilir. Bu adım $|\breve{Y}| = k$ olana kadar tekrarlanır.
4. Adım (3) teki kümeleme merkez noktalarının kümesi olan Y yerine yeni $Y = \breve{Y}$ alınarak adım (3) işlemi, $|Y| =$ k olana kadar çalıştırılır.
5. Nihai kümeleme merkez noktalarının kümesi olan Y ile KO yöntemi başlatılarak kümeleme işlemi tamamlanır.

Çift k (ÇK) olarak adlandırdığımız önerilen yöntemin ilk adımında alınan örneklem ile amacı, yöntem içinde KO ve KO++ birer kez başlatıldığından dolayı işlem içerisindeki zaman karmaşıklığının azaltmaktır. Böylece KO ve KO++ yöntemleri Şekil 1 'deki veri kümesi üzerinde çalışmak yerine Şekil 2'deki seyreltilmiş veriler ile çalışarak daha kısa zamanda işlemi tamamlayabilmektedir. Bu çalışmaya ait algoritma, python programlama diliyle geliştirilmiştir. Python programlama dilinin numpy kütüphanesi, rasgele sayı üreteç algoritması olarak PCG64 üreteci kullanmaktadır. Önerilen yöntemin ilk adımında alınan örneklem kümesi, bu kütüphanenin üreteç algoritması kullanılarak elde edilmiştir.



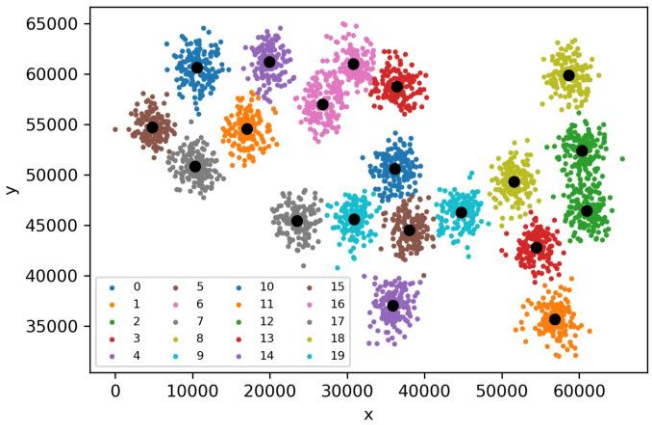
Şekil 2. Deneme 1'e ait 1. adım ve 2. adım çıktıları. (siyah noktalar üretilen ağırlık merkezlerini ve renkli noktalar ise örneklem verilerini göstermektedir)

Örneklem kümesi üzerinde kümeleme sayısı parametresinin 'k' yerine '$2 * k$' olarak seçilmesiyle KO++ yöntemi başlatılmış ve böylece seçilecek ağırlık merkezi noktalarının dağılımı arttırılması hedeflenmiştir.

3. Araştırma Sonuçları ve Tartışma

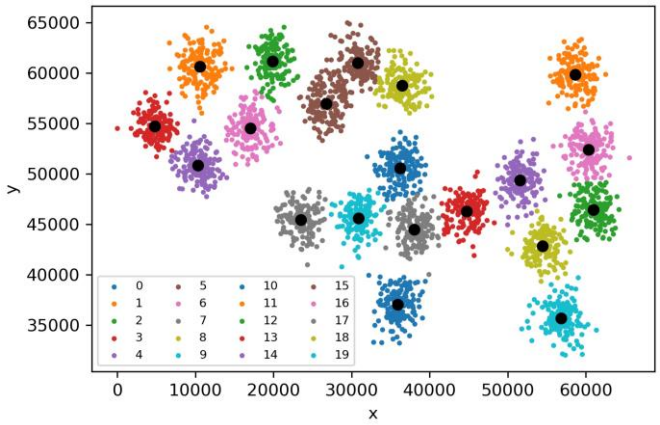
3.1. Bulgular

Şekil 1'deki A1 veri kümesine önceden hesaplanmış merkezler ile başlatılan KO algoritması uygulanmış, nihai TKH ve bölümleme etiketleri, diğer yöntemler ve önerilen ÇK yöntem sonuçlarıyla karşılaştırmak üzere referans olarak alınmıştır. Buradaki "diğer yöntemler" ifadesi ile kastedilen, aynı veri kümesi üzerinde KO (rasgele) ve KO++ yöntemlerinin çalıştırılmasıdır. Referans yöntem dışındaki bu üç yöntem, aynı veri kümesine 5 kez uygulanmış, oluşan nihai bölümleme etiketlerinin referans sonuçlara göre ölçümleri yapılmış ve buna bağlı olarak önerilen yöntem ile birlikte diğer yöntemlerin başarıları hesaplanmıştır. Bu ölçümlere ek olarak üç yöntemin TKH'larının, referans TKH ile olan yüzde bağıl hatası belirlenmiş ve tüm bu sonuçlar Tablo 2.'de gösterilmiştir. ÇK yönteminde KO++ ve KO yöntemleri birer kez olmak üzere KO yöntemi iki kez çalıştırıldığından dolayı, diğer yöntemler de iki kez farklı merkezler ile başlatılmış ve buna göre kıyaslama yapılmıştır.



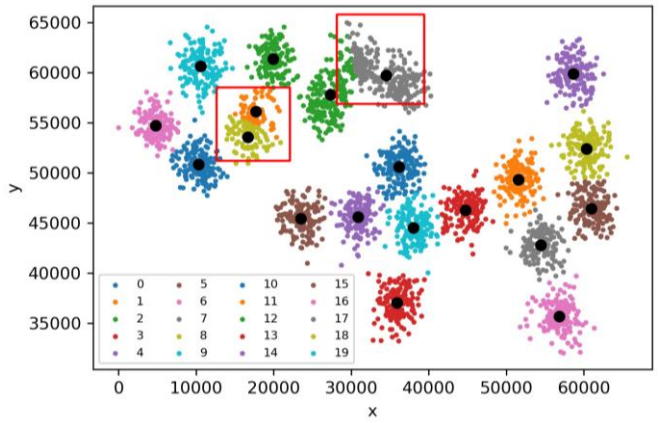
Şekil 3. Önerilen Yöntemin 1. Denemesi (%$\boldsymbol{\delta}$ =0,036, MI=2,990,RI=1,000, ARI=0,998)

Şekil 3. ve Şekil 4.'te, önerilen yöntemin sırasıyla 1. ve 2. denemelerine ait sonuçları gösterilmektedir. ÇK yönteminin Tablo 2.'deki ölçüleri dikkate alındığında diğer yöntemlere göre, TKH anlamında yüzde bağıl hata düşük çıkarak ve etiketleme anlamında ise MI, RI ve ARI değerleri yüksek çıkarak, başarısının oldukça yüksek olduğu görülebilir.



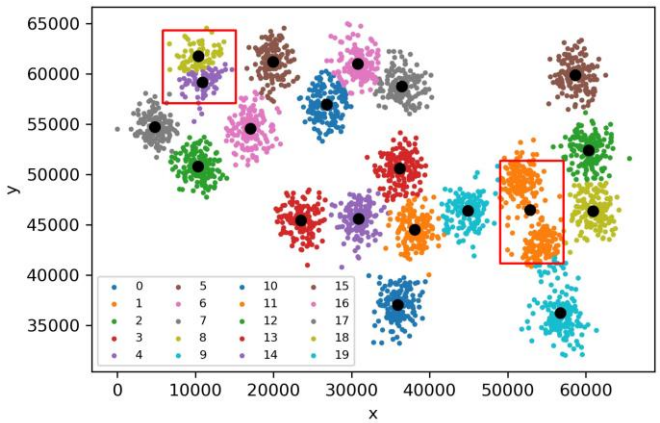
Şekil 4. Önerilen Yöntemin 2. Denemesi (%δ =0,030, MI=2,988,RI=1,000, ARI=0,997)

ÇK yöntemi 3. denemede Şekil 5.'te görüldüğü gibi bir ağırlık merkezini hatalı üreterek iki hatalı etiketleme yapmasına rağmen başarısı, Tablo 2.'den de görüleceği üzere diğer yöntemlerden daha kötü değildir.



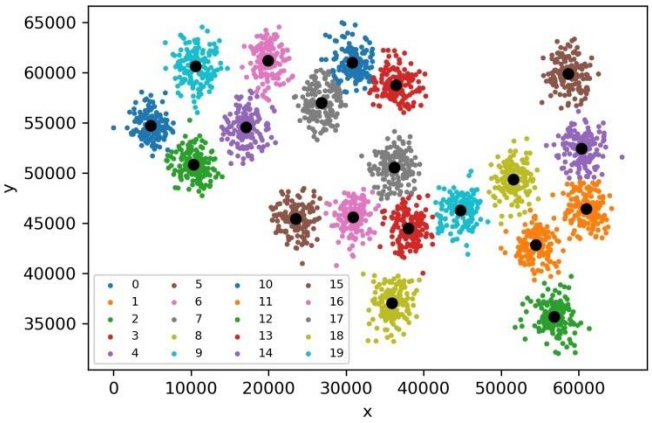
Şekil 5. Önerilen Yöntemin 3. Denemesi (%δ =16,117, MI=2,896,RI=0,993, ARI=0,924)

5 denemenin sadece 4'üncüsünde sadece bir ağırlık merkezini hatalı üreterek başarısız olmuştur. Ancak diğer yöntemlerin başarısı ile kıyaslandığında, THK anlamında yüzde bağıl hata biribirine yakın olup etiketleme anlamında ise MI, RI ve ARI değerleri biribirine çok yakın olduğu Tablo 2.'de görülmektedir.



Şekil 6. Önerilen Yöntemin 4. Denemesi (%δ =26,411, MI=2,890,RI=0,993, ARI=0,924)

Şekil 7.'de gösterilen son deneme kümelemesi Tablo 2.'den de anlaşılacağı üzere önerilen yöntemin diğer yöntemlere göre başarılı olduğunu ortaya koymuştur.



Şekil 7. Önerilen Yöntemin 5. Denemesi (%δ =0,029, MI=2,990,RI=1,000, ARI=0,998)

4. Sonuçlar ve Öneriler

Bu çalışmada A1 verikümesine ÇK yöntemi ile başlatılan KO algoritması, KO (rasgele) ve KO++ algoritmaları uygulanmıştır. Bu yöntemlere ait nihai bölümlemelerin başarıları ve TKH'larının sonuçları Tablo 2.'de gösterilmiştir. Tüm yöntem ve başarı kriterlerine ait kodlama Python programlama dilinde yazılmış olup bu programlama dilinin kütüphaneleri kullanılmıştır.

Bu çalışmaya ait algoritma, python programlama diliyle geliştirilmiştir. Python programlama dilinin "sklearn.cluster" kütüphanesi, KO algoritmasının farklı merkezler ile kaç kez başlatılacağını belirten n_init parametresi kullanmaktadır. Önerilen yöntemin içinde KO ve KO++ yöntemleri birer kez çalıştırıldığından dolayı, kıyaslama için kullanılan KO (rasgele) ve KO++ yöntemlerinin farklı merkezler ile başlatma sayısı (n_init) olarak "2" değeri seçilmiştir. Ayrıca TKH'ların kıyaslanması için yüzde bağıl hata *(%δ)* kullanılmıştır. 5 denemeden oluşan deneyde, önerilen yöntemin KO (rasgele) ve KO++ yöntemlerine göre daha başarılı olduğu, Tablo 2.'den gözlemlenmiştir. Bu açıdan önerilen ÇK ile başlatılan KO yöntemi, KO (rasgele) ve KO++ yöntemine göre daha başarılı sonuçlar vermekte olup, başarısız olunan durumlardaki başarısızlık derecesi ise hemen hemen kıyaslanan yöntemler kadardır.

Başarının arttırılması için önerilen yöntemdeki kümeleme sayısı veya örneklem seçimi gibi parametreler tekrar düzenlenerek başarının artmasına katkı sağlanabilir. Bu anlamda yapılan çalışma gelecek çalışmalara zemin hazırlamaktadır.

Tablo 2. Önerilen ve Diğer Yöntemlerin Başarı Ölçümleri.

Deneme	Yöntem	TKH	%δ	MI	RI	ARI
1	KO (Orijinal Merkez)	12150760265,000	referans	2,996	0,000	1,000
	ÇK	**12146338010,547**	**0,036**	**2,990**	**1,000**	**0,998**
	KO (Rasgele)	14491251869,920	19,262	2,895	0,993	0,928
	KO++	14082208713,901	15,896	2,893	0,993	0,926
2	**ÇK**	**12147128466,414**	**0,030**	**2,988**	**1,000**	**0,997**
	KO (Rasgele)	14491251869,920	19,262	2,895	0,993	0,928
	KO++	14082208713,902	15,896	2,893	0,993	0,926
3	**ÇK**	**14109129146,570**	**16,117**	**2,896**	**0,993**	**0,924**
	KO (Rasgele)	14491251869,920	19,262	2,895	0,993	0,928
	KO++	14082208713,902	15,896	2,893	0,993	0,926
4	**ÇK**	**15359846639,327**	**26,411**	**2,890**	**0,993**	**0,924**
	KO (Rasgele)	14491251869,920	19,262	2,895	0,993	0,928
	KO++	14082208713,902	15.896	2,893	0,993	0,926
5	**ÇK**	**12147254122,73**	**0,029**	**2,990**	**1,000**	**0,998**
	KO (Rasgele)	14491251869,92	19,262	2,895	0,993	0,928
	KO++	14082208713,90	15,896	2,893	0,993	0,926

Kaynakça

[1] Rahim, M. S., & Ahmed, T. (2017). An initial centroid selection method based on radial and angular coordinates for K-means algorithm. In 2017 20th International Conference of Computer and Information Technology (ICCIT) (pp. 1-6). IEEE.

[2] Astrahan, M. M. (1970). Speech analysis by clustering, or the hyperphoneme method (No. AIM-124). STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.

[3] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM computing surveys (CSUR), 31(3), 264-323.

[4] Pena, J. M., Lozano, J. A., & Larranaga, P. (1999). An empirical comparison of four initialization methods for the k-means algorithm. Pattern recognition letters, 20(10), 1027-1040.

[5] Redmond, S. J., & Heneghan, C. (2007). A method for initialising the K-means clustering algorithm using kd-trees. Pattern recognition letters, 28(8), 965-973.

[6] Xu, R., & Wunsch, D. (2008). Clustering (Vol. 10). John Wiley & Sons.

[7] Singhal, M., & Shukla, S. (2018, February). Centroid Selection in Kernel Extreme Learning Machine Using K-Means. In 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 708-711). IEEE.

[8] Zahra, S., Ghazanfar, M. A., Khalid, A., Azam, M. A., Naeem, U., & Prugel-Bennett, A. (2015). Novel centroid selection approaches for KMeans-clustering based recommender systems. Information sciences, 320, 156-189.

[9] Ding, C., & He, X. (2004, July). K-means clustering via principal component analysis. In Proceedings of the twenty-first international conference on Machine learning (p. 29).

[10] Selim, S. Z., & Ismail, M. A. (1984). K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. IEEE Transactions on pattern analysis and machine intelligence, (1), 81-87.

[11] MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).

[12] Hand, D. J., & Krzanowski, W. J. (2005). Optimising k-means clustering results with standard software packages. Computational Statistics & Data Analysis, 49(4), 969-973.

[13] Arthur, D., & Vassilvitskii, S. (2006). k-means++: The advantages of careful seeding. Stanford.

[14] Lloyd, S. (1982). Least squares quantization in PCM. IEEE transactions on information theory, 28(2), 129-137.

[15] Katsavounidis, I., Kuo, C. C. J., & Zhang, Z. (1994). A new initialization technique for generalized Lloyd iteration. IEEE Signal processing letters, 1(10), 144-146.

[16] Bradley, P. S., & Fayyad, U. M. (1998, July). Refining initial points for k-means clustering. In ICML (Vol. 98, pp. 91-99).

[17] Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. Pattern recognition, 36(2), 451-461.

[18] Gan, G., Chaoqun, M., & Wu, J. (2007). Data Clustering: Theory, Algorithms and Applications; 20 of Series on Statistics and Applied Probability. Philadelphia, PA.

[19] Xie, J., Jiang, S., Xie, W., & Gao, X. (2011). An Efficient Global K-means Clustering Algorithm. JCP, 6(2), 271-279.

[20] Beigi, H. (2011). Speaker recognition. In Fundamentals of Speaker Recognition (pp. 543-559). Springer, Boston, MA.

[21] Agrawal, A., & Gupta, H. (2013). Global K-means (GKM) clustering algorithm: a survey. International journal of computer applications, 79(2).

[22] Rani, A. J. M., & Parthipan, L. (2012). Clustering Analysis by Improved Particle Swarm Optimization and KMeans Algorithm.

[23] Li, H., Yang, X., & Wei, W. (2014). The application of pattern recognition in electrofacies analysis. Journal of Applied Mathematics, 2014.

[24] Hennig, C. (2015). Clustering strategy and method selection. arXiv preprint arXiv:1503.02059.

[25]Kärkkäinen, I., & Fränti, P. (2002). Dynamic local search algorithm for the clustering problem. Joensuu, Finland: University of Joensuu.

[26] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association, 66(336), 846-850.

[27] Campello, R. J. (2007). A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. Pattern Recognition Letters, 28(7), 833-841.

[28] Santos, J. M., & Embrechts, M. (2009, September). On the use of the adjusted rand index as a metric for evaluating supervised classification. In International conference on artificial neural networks (pp. 175-184). Springer, Berlin, Heidelberg.

[29]Yeung, K. Y., & Ruzzo, W. L. (2001). Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. Bioinformatics, 17(9), 763-774.

[30] Priness, I., Maimon, O., & Ben-Gal, I. (2007). Evaluation of gene-expression clustering via mutual information distance measure. BMC bioinformatics, 8(1), 111.

[31] Kraskov, A., & Grassberger, P. (2009). MIC: Mutual information based hierarchical clustering. In Information theory and statistical learning (pp. 101-123). Springer, Boston, MA.

[32] Newman, M. E., Cantwell, G. T., & Young, J. G. (2020). Improved mutual information measure for clustering, classification, and community detection. Physical Review E, 101(4), 042304.

[33] Hubert, L., & Arabie, P. (1985). Comparing partitions. Journal of classification, 2(1), 193-218.

[34] Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern recognition letters, 31(8), 651-666.

[35] Jenssen, R., & Eltoft, T. (2008). A new information theoretic analysis of sum-of-squared-error kernel clustering. Neurocomputing, 72(1-3), 23-31.

[36] Bramer, M. (2007). Principles of data mining (Vol. 180). London: Springer.

[37] Han, J., Kamber, M., & Pei, J. (2011). Data mining concepts and techniques third edition. The Morgan Kaufmann Series in Data Management Systems, 5(4), 83-124.

[38] Aggarwal, C. C., & Reddy, C. K. (2014). Data clustering. Algorithms and applications. Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra.

[39] Aubaidan, B., Mohd, M., & Albared, M. (2014). Comparative study of k-means and k-means++ clustering algorithms on crime domain.

[40] Kvålseth, T. O. (2017). On normalized mutual information: measure derivations and properties. Entropy, 19(11), 631.

[41] Shannon, C. E. (1948). A mathematical theory of communication. The Bell system technical journal, 27(3), 379-423.