Avrupa Bilim ve Teknoloji Dergisi
Özel Sayı 29, S. 58-64, Aralık 2021

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
Special Issue 29, pp. 58-64, December 2021

Research Article

Jensen Shannon Mesafesi Temelli Uyarlanmış Bulanık C Ortalamalar Kümeleme Yöntemi

Naciye Aydin[1*], Gökhan Kayhan[2]

[1*] Ondokuz Mayıs Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Samsun, Türkiye, (ORCID: 0000-0002-6261-6121), naciye.aydin@bil.omu.edu.tr

[2] Ondokuz Mayıs Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Samsun, Türkiye (ORCID: 0000-0003-3391-0097), gkayhan@omu.edu.tr

(International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) 2021 – 21-23 October 2021)

(**DOI:** 10.31590/ejosat.1021473)

ATIF/REFERENCE: Aydin, N. & Kayhan, G. (2021). Jensen Shannon Mesafesi Temelli Uyarlanmış Bulanık C Ortalamalar Kümeleme Yöntemi. *Avrupa Bilim ve Teknoloji Dergisi*, (29), 58-64.

Öz

Denetimsiz öğrenmenin önemli bir dalı olan kümeleme yöntemleri, bilgisayar bilimlerinin popüler araştırma alanlarından biridir. Kümeleme yöntemlerinin birçoğunda, küme sayısının tahmin edilememesi önemli bir problem olarak ortaya çıkmaktadır. Bu çalışmada küme sayısını tahmin etmek için Jensen Shannon (JS) mesafesi, Bulanık C Ortalamalar (BCO) algoritmasına uyarlanarak yeni bir Jensen Shannon Bulanık C Ortalamalar (JSBCO) algoritması önerilmiştir. Bu çalışma, BCO algoritmasını temel alan yeni bir algoritma önerisiyle doğru küme sayısını belirleme başarımını artırmayı hedeflemektedir. Bu amaçla, önerilen JSBCO algoritması, Uyarlanmış Bölüm Entropisi (MPE) ile kullanılan BCO yöntemi ve saf BCO algoritması ile karşılaştırılmıştır. BCO algoritması 6 farklı veri seti için, veri tabanında tanımlanan sahip oldukları gerçek küme sayıları ile çalıştırılmıştır. Aynı veri setleri MPE–BCO ve JSBCO yöntemleri için de çalıştırılarak verilere ait küme sayıları tahmin edilmiştir. Elde edilen sonuçlar ile JSBCO, MPE-BCO ve BCO yöntemlerinin karşılaştırması yapılmıştır. Yapılan bu karşılaştırma ile JSBCO algoritmasının küme sayısını tahmin etmede ve amaç fonksiyonunu minimize etmede daha başarılı olduğu sonucuna varılmıştır. JSBCO algoritmasının MPE-BCO yöntemine göre, küme sayısı tahmin etme üstünlüğünün yanı sıra, küme sayısı tahmininde daha kararlı davrandığı sonucuna ulaşılmıştır. JSBCO algoritmasının küme sayısı tahmin etmede daha kararlı davrandığını göstermek için Aggregation veri seti esas alınarak hem MPE-BCO algoritması hem JSBCO algoritması ile 10 farklı çalışmasının sonuçları gösterilmiştir. Bu sonuçlara göre MPE-BCO yöntemi, 10 farklı çalışma içinde toplamda 2 kez doğru tahmin ederek %20 doğruluk elde ederken, JSBCO algoritması 10 farklı çalışma içinde 8 kez doğru tahminde bulunarak %80 doğruluk elde etmiştir. Ayrıca tüm veri setlerinin 10 farklı çalışması sonucu elde edilen küme sayısı tahminleri her iki yöntemde karşılaştırılarak, JSBCO algoritmasının artan küme sayısı ve özellik sayısında da kararlı davranışlarını sürdürdüğü gösterilmiştir. Son olarak JSBCO algoritmasının, BCO algoritması kısmından kaynaklanan dezavantajlı durumlarının giderilmesi için gelecek çalışmalara yol gösteren önerilerde bulunulmuştur.

Anahtar Kelimeler: Denetimsiz Öğrenme, Kümeleme, Bulanık C Ortalamalar, Jensen Shannon Mesafesi, Uyarlanmış Bölüm Entropisi.

Modified FCM Clustering Method based on Jensen Shannon Distance

Abstract

Clustering methods, which is an important branch of unsupervised learning is one of the popular research areas of computer science. The inability to predict the number of clusters is an important problem in many clustering methods. In this study, a new Jensen Shannon Fuzzy C Means (JSFCM) algorithm have been proposed by modifying the Jensen Shannon (JS) distance to the Fuzzy C Means (FCM) algorithm to estimate the number of clusters. The goal of the study is to increase the performance of determining the correct number of clusters with a new algorithm proposal based on the FCM algorithm. For this purpose, the suggested JSFCM algorithm is compared with the FCM method used with Modified Partition Entropy (MPE-FCM) and the pure FCM algorithm. The FCM algorithm was run for 6 different data sets with the real number of clusters defined in the database. The number of clusters of datasets was predicted by running the same datasets for the JSFCM and MPE-FCM methods. The obtained results are compared with the JSFCM, MPE-FCM and pure FCM methods. With this comparison, it is concluded that the JSFCM algorithm is more successful

[*] Naciye Aydin: naciye.aydin@bil.omu.edu.tr

in estimating the number of clusters and minimizing the objective function. It has been concluded that the JSFCM algorithm, in addition to its superiority in estimating the number of clusters is more stable in estimating the number of clusters compared to the MPE-FCM method. Based on the aggregation dataset, when the results of 10 different runs with both JSFCM and MPE-FCM algorithms are examined, it has been demonstrated that the JSFCM algorithm is more stable in estimating the number of clusters. According to these results, the MPE-FCM method achieved 20% accuracy by making 2 correct predictions in 10 different runs while the JSFCM method achieved 80% accuracy by making 8 correct predictions in 10 different runs. In addition, the cluster number predictions of all data sets obtained in 10 different runs were compared with both methods, and it was shown that the JSFCM algorithm maintains its stability when the number of clusters and features increases. Finally, suggestions are made to guide future research to eliminate the disadvantageous situations of the JSFCM algorithm arising from the FCM algorithm.

Keywords: Unsupervised Learning, Clustering, Fuzzy C Means, Jensen Shannon Distance, Modified Partition Entropy

1. Giriş

Denetimsiz makine öğreniminin bir alt dalı olan kümeleme yöntemlerinde, doğru küme sayısını saptayabilmek kümeleme algoritmalarının önemli problemlerinden biri olarak görülmektedir. Yüksek boyutlu gerçek dünya veri setlerinde verinin kaç kümeden oluştuğu çoğu zaman bilinmez. Bu bilinmezliği gidermek için küme sayısını tahmin eden yöntemler geliştirilmiştir.

Literatürde küme sayısının önceden bilinmesini gerektirmeyen algoritmalardan en bilinenleri hiyerarşik kümeleme yöntemleridir. Bu yöntem, bir ağaç yapısı şeklinde iç içe kümelerden oluşur. Genel olarak birleştirici ve bölücü olmak üzere iki tür yaklaşımı mevcuttur. Birleştirici hiyerarşik kümeleme yönteminde, her bir veri başlangıçta bir küme olarak belirlenir. Her bir küme arasındaki mesafe hesaplanarak birbirine en yakın noktaların birleştirilmesi ile hiyerarşik yapı oluşturulur. Bölücü hiyerarşik kümeleme yönteminde ise tüm veri elemanları başlangıçta tek bir küme olarak belirlenir. Her adımda benzerlik oranı düşük olan kümeler bölünerek daha küçük kümelerin oluşması ile elde edilir. Ancak hiyerarşik kümeleme yöntemlerinde, kümeler ardışık bir şekilde oluştuğundan statiktir. Yani bir kümeye atanan veri elemanları başka bir kümeye ait olması durumunda yeniden ayarlama yapılamaz. Ayrıca yüksek karmaşıklık ve hesaplama yavaşlığı, hiyerarşik kümeleme yöntemlerinde ayrı bir dezavantaj oluşturur (Ezugwu et al. 2021 ; Govender & Sivakumar 2020).

K-ortalamalar ve Bulanık C Ortalamalar (BCO) gibi klasik kümeleme yöntemlerinde küme sayısı kullanıcı tarafından belirlenir. Bu kümeleme algoritmaları, farklı yöntemlerin bir araya getirilmesiyle oluşturulan hibrit algoritmalar ile kullanıcıdan küme sayısı girdisi almadan kümeleme yapabilmeyi sağlamıştır. Hibrit yöntemlerle oluşturulan algoritmaların birçoğunda küme sayısının algoritma tarafından belirlendiği meta-sezgisel yöntemlerin bir türü olan evrimsel algoritmalardan yararlanılmıştır (Hruschka et al. 2009). Bu çalışmalardan biri, K-ortalamalar algoritması ile birlikte genetik algoritma kullanarak, küme sayısının kullanıcı tarafından girilmesini önlemek için önerilen GenClust algoritmasıdır (Rahman & Islam 2014). Başka bir çalışmada, kuantumdan ilham alan genetik algoritmaya dayalı k-ortalama kümeleme algoritması (KMQGA) (Xiao et al. 2010) önerilmiştir. BCO ile yapılan bir çalışmada, uzaktan algılama görüntülerinde küme sayısının belirlenmesi problemini çözmek için, değiştirilmiş diferansiyel evrim algoritması kullanılarak bir BCO kümeleme yöntemi olan MoDEAFC algoritması önerilmiştir (Maulik, Member, and Saha 2010). Başka bir araştırma, çok amaçlı genetik algoritma tabanlı bulanık kümeleme algoritması (FCM-NSGA) sunarak küme sayısını belirleme problemini çözmüştür (Wikaisuksakul 2014).

BCO algoritması görüntü bölütleme problemi gibi özel sorunların çözümü için otomatikleştirilmiştir. Görüntü bölütleme problemi için geliştirilen bir çalışmada, küme sayısı bilinmeyen bir görselde, önerilen Otomatik Bulanık C Ortalamalar (*AFCM*) algoritması ile görüntü pikselleri homojen gruplara ayrılmıştır. Bölütleme kalitesini iyileştirmek için AFC algoritması tekrar geliştirilerek, Otomatik Değiştirilmiş Bulanık C Ortalamalar (*AMFCM*) algoritması ikinci bir otomatik kümeleme yöntemi olarak önerilmiştir (Li & Shen 2010).

Bu çalışmada BCO kümeleme algoritmasının dezavantajlarından biri olan küme sayısının belirlenmesi problemi üzerinde durulmuştur. Bu problemi gidermek için bilgi entropisinde önemli bir yeri olan Kullback Leibler sapmasının genişletilmiş hali olan Jensen Shannon mesafesi kullanılarak yeni bir algoritma olan JSBCO algoritması önerilmiştir.

2. Materyal ve Metot

2.1. Bulanık C Ortalamalar

BCO algoritması, bulanık mantık ilkesi esas alınarak J.Bezdek tarafından geliştirilmiş (Bezdek 1984) bir kümeleme algoritmasıdır. Her bir veri örneği için birden fazla kümeye ait olabilen bir üyelik değeri bulunur. Veri örneklerinin farklı kümelere olan üyelik değerlerinin toplamı 1 olmalıdır. BCO amaç fonksiyonu temelli bir bulanık kümeleme algoritmasıdır. Denklem (1)'de gösterilen amaç fonksiyonu en küçük kareler yönteminin genelleştirilmesi ile oluşturulmuştur. BCO algoritması amaç fonksiyonunu olabilecek en küçük değerde tutmaya çalışır. Bu değer küçüldükçe, küme içi benzerlik ve kümeler arası benzersizliğin yükseldiği bir kümeleme gerçekleştirilmiş olur.

$$J_m(U,v) = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^{m} \cdot d_{ik}^{2} \quad (1)$$

Denklem (1)'de U değeri üyelik matrisini, v değeri küme merkezini, n parametresi veri sayısını, c parametresi $2 \leq c < n$ aralığındaki küme sayısını, u_{ik} değeri k. verinin i. kümeye ait olma olasılığını, m değeri ($1 \leq m < \infty$) bulanıklık parametresini, d_{ik} ise i. küme ile veri noktası arasındaki mesafeyi temsil etmektedir.

Denklem (2) ile küme merkezlerinin hesaplanması verilmiştir.

$$V_{ij} = \frac{\sum_{i=1}^{n} u_{ik}^{m} x_{kj}}{\sum_{k=1}^{n} u_{ik}^{m}} \quad (2)$$

Burada u_{ik} *i*. veri örneğinin *k* kümesine üyelik değeridir. *i*. veri örneğinin tüm kümelere olan üyelik toplamı *1* 'e eşittir. x_{kj} veri noktasını temsil eder.

N sayıda veri örneği ve k sayıda küme için üyelik matrisinin güncellenmesi Denklem (3) 'te gösterilmiştir.

$$U_{ik} = \frac{1}{\sum_{j=1}^{k} (\frac{d_{ik}}{d_{jk}})^{\frac{2}{m-1}}} \quad (3)$$

2.2. Uyarlanmış Bölüm Entropisi (MPE)

Uyarlanmış Bölüm Entropisi (Modified Partition Entropy, MPE) yöntemi BCO algoritmasının küme tahmini için uygulanan (Schenatto et al. 2017) en yaygın yöntemlerden biridir. MPE, [0-1] aralığında değer alır. 0'a yaklaşan değerler için kümelemenin daha doğru olduğunu gösterir. 1'e yaklaşan değerler için düzensizliğin arttığını gösterir (Boydell & McBratney 2002).

$$MPE = -(\sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^{m} \, . \log(u_{ik})/n) / \log c \quad (4)$$

MPE ile BCO yönteminde maksimum küme sayısı belirlenir. Denklem (4)'te verilen ifade ile MPE hesaplanır. Burada C maksimum küme sayısını belirtmek üzere; $MPE_{C-1} < MPE_{C}$ ve $MPE_{C-1} < MPE_{C-2}$ şartı sağlandığında, bölüm doğrulama kriterini minimum eden küme sayısı c=c-1 uygun küme sayısı olarak seçilir.

2.3. Jensen Shannon (JS) mesafesi

Jensen Shannon ıraksaması (JSD) iki olasılık arasındaki farkı ölçmek için kullanılır (Lin 1991). JSD, Denklem (5)'te gösterilen Kullback Leibler (KL) sapmasını kullanarak simetrik bir hesaplama yapar.

$$KL\ (X||Y) = \sum_{i=1}^{N} x_i \log \frac{x_i}{y_i} \quad (5)$$

$$JSD(P||Q) = \frac{1}{2} KL(P||M) + \frac{1}{2} KL(Q||M) \quad (6)$$

Denklem (6) 'da *P* ve *Q* JSD hesaplamasında kullanılan iki olasılık dağılımıdır. Burada logaritma 2 tabanında kullanılır. *M* ise Denklem (7)' de gösterilen *P* ve *Q* değerlerinin ortalamasını ifade eder.

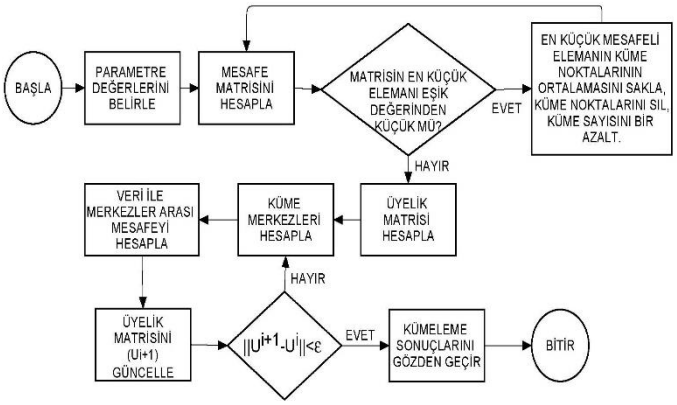
$$M = \frac{1}{2} (Q+P) \quad (7)$$

$$JSD(P||Q) = JSD(Q||P) \quad (8)$$

Denklem (8) *P* ve *Q* olasılığı ile *Q* ve *P* olasılığının JSD ile hesaplandığında aynı sonucu verdiğini gösteren simetrik bir benzerlik ölçüsüdür. *P* ve *Q* değerlerinin olasılıkları toplamı 1 olmalıdır. Denklem (6)'te gösterilen JSD değerinin karekökünün hesaplanması ile Jensen Shanon mesafesi bulunur. Jensen Shannon mesafesi 0 ile 1 arasında değer alır. Değerin 0'a yakın olması iki olasılık arasındaki (*P* ve *Q*) yakınlığının yüksek olduğunu gösterir.

2.4. Jensen Shannon Bulanık C Ortalamalar (JSBCO) Algoritması

Bu çalışmada BCO algoritmasının küme sayısındaki bilinmezliğini gidermek için, Denklem (6)'da gösterilen Jensen Shannon mesafesi kullanılarak yeni bir Jensen Shannon Bulanık C Ortalamalar (JSBCO) kümeleme algoritması önerilmiştir.



Şekil 1. JSBCO Algoritmasının Akış Şeması

JSBCO algoritmasının ilk adımında bulanıklık parametresi, maksimum küme sayısı, maksimum iterasyon sayısı, Jensen Shannon mesafesi için belirlenen eşik değeri gibi parametre değerleri atanır. Algoritmanın maksimum küme sayısı belirlenir. Küme noktaları ilk etapta rastgele seçilir. Seçilen tüm küme noktaları arasındaki Jensen Shannon mesafe matrisi hesaplanır. Hesaplanan matristeki minimum değerli Jensen Shannon mesafesi belirlenen eşik değerinden küçük olduğu sürece, iki küme noktasının ortalama değeri yeni küme noktası olarak atanır. Minimum değer ilişkili iki küme noktası kaldırılarak küme sayısı bir azaltılır. Eşik değeri, minimum değerli Jensen Shannon mesafesinden küçük değilse uygun küme sayısı bulunmuş olur. Ardından üyelik matrisleri rastgele oluşturularak BCO algoritması başlatılmış olur. Böylece BCO algoritması başlatılmış olur. Sonraki adımda küme merkezleri Denklem (2)'deki gibi, üyelik matrislerinin güncellenmesi Denklem (3)'teki gibi hesaplanır. Hesaplama sonucunda JSBCO kümeleme algoritması bulunan uygun değerli küme sayısı ile veri elemanlarının hangi kümeye ne kadar olasılıkla bağlandığı sonucunu verir. Şekil 1'de JSBCO algoritmasının akış şeması gösterilmiştir.

3. Araştırma Sonuçları ve Tartışma

Bu çalışmada JSBCO algoritmasının sonuçlarını değerlendirmek için Path-based1 (Chang and Yeung 2008), Compound (Zahn 1971), Aggregation (Gionis, Mannila, & Tsaparas 2007), R15 (Veenman, Reinders, & Backer 2002) yapay veri setleri ve UCI Makine Öğrenmesi veri seti deposunda bulunan (Dheeru & Taniskidou) İris ve Glass gerçek veri setleri olmak üzere 6 farklı veri seti kullanılmıştır. Tablo 1'de her bir veri setinin içerdiği örnek sayısı, küme sayısı ve veri setlerinin özellik sayısı bilgileri gösterilmiştir.

Tablo 1. Çalışmada Kullanılan Veri Setleri

Veri Seti	Örnek Sayısı	Küme Sayısı	Özellik Sayısı
Path-based1	300	3	2
Compound	788	6	2
Aggregation	399	7	2
R15	600	15	2
İris	150	3	3
Glass	214	6	9

JSBCO algoritmasının doğruluğunu kıyaslamak amacıyla, saf BCO algoritması ve MPE yöntemi kullanılmıştır. Tablo 1'de açıklanan veri setlerinin gerçek küme sayıları kullanılarak BCO algoritması çalıştırılmış ve her bir veri seti için Denklem 1'de gösterilen amaç fonksiyonu (J_m) hesaplanmıştır. Tablo 2'de BCO, JSBCO ve MPE ile BCO yöntemlerinin amaç fonksiyonu üzerinden kümeleme başarımlarının karşılaştırması verilmiştir. Veri setlerinin bilinen gerçek küme sayıları ile BCO algoritması ile kümeleme işlemi gerçekleştirilmiş ve J_m değeri hesaplanmıştır. Sonrasında MPE-BCO ve JSBCO ile bulduğu küme sayılarına göre Jm değeri hesaplanarak benzer durumlardaki başarımları hesaplanmıştır.

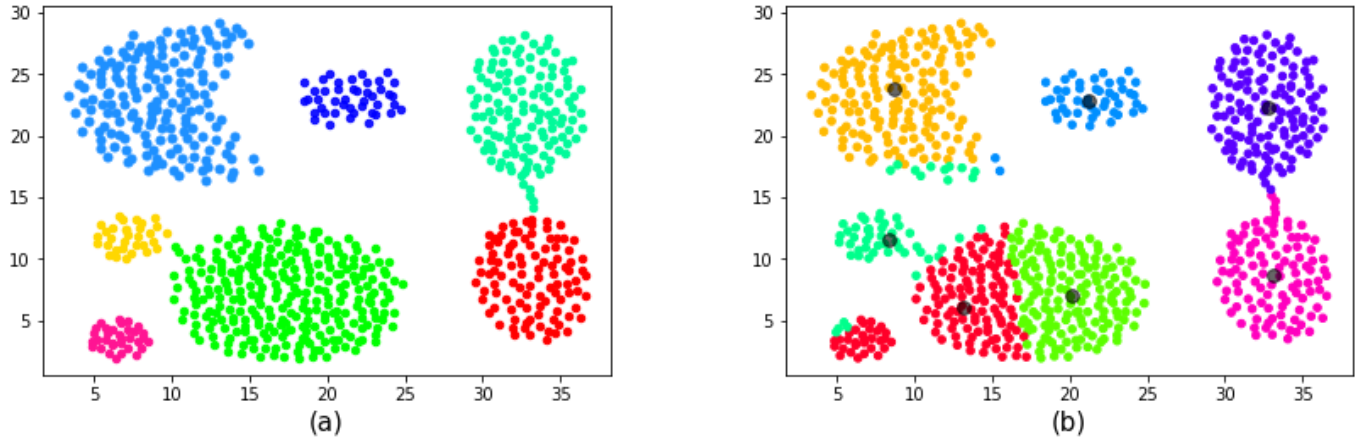
Tablo 2'nin oluşturulmasında Jm değerlerinin uygunluğu ve Tablo 3'te verilen tüm veri setleri için 10 farklı çalışma sonucunda küme sayılarının tekrar sıklığı esas alınmıştır. Tablo 2'de görüldüğü üzere örneğin Path-based1 verisi için JSBCO yöntemi gerçek küme sayısını yakalamış ve BCO yöntemine göre daha düşük bir Jm değeriyle daha başarılı bir kümeleme gerçekleştirmiştir. Ayrıca yine JSBCO yönteminin MPE-BCO yöntemine göre de daha başarılı bir kümeleme yaptığı görülmektedir. Tablodaki veri setlerine göre genel olarak JSBCO yöntemi BCO kadar başarılı olmasının yanı sıra küme sayısı belirsizliği problemini ortadan kaldıran bir yöntem olması avantajına sahiptir.

Şekil 2'de aggregation veri setinin, gerçek küme gösterimi (a) ve BCO algoritmasının gerçek küme sayısı ile çalıştırılması sonucu veri noktalarının kümelenmesi ve küme merkezlerinin dağılımı (b) gösterilmiştir. BCO algoritmasının eliptik şekillerdeki başarımının (b) düşük olması dezavantajı bu veri setinde görülmektedir.

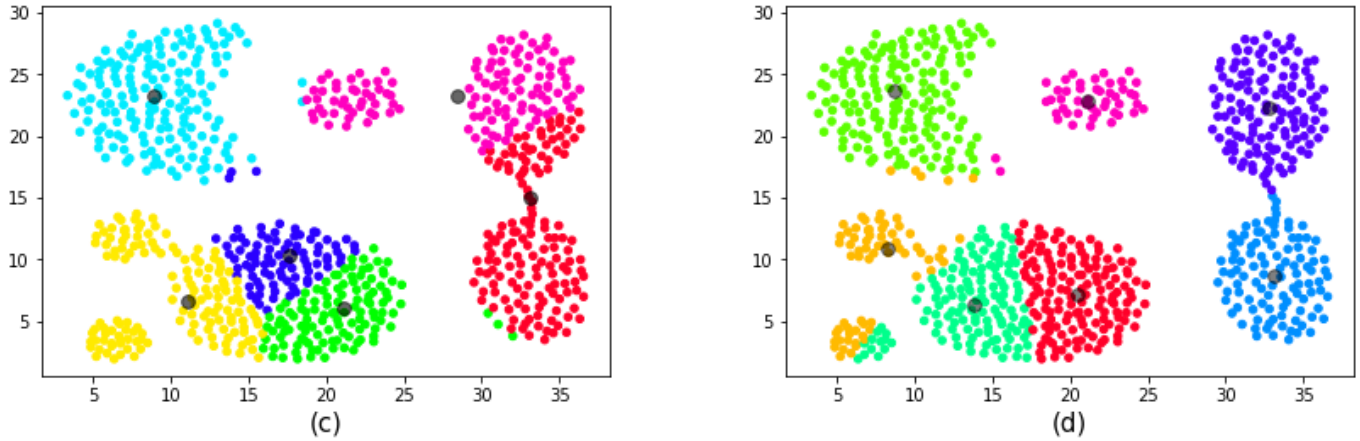
Tablo 2'de gösterildiği üzere, MPE-BCO yöntemi ile aggregation veri seti kullanıldığında tahmini küme sayısı 6 olarak bulunmuştur. Aynı veri seti JSBCO algoritması ile çalıştırıldığında tahmin edilen küme sayısı 7 olarak bulunmuş ve algoritma gerçek küme sayısını bu veri setinde tahmin edebilmiştir. Şekil 3 (c)' de, bu veri setinin MPE-BCO yöntemi ile Şekil 3 (d) JSBCO yöntemi kullanılarak tahmin edilen küme sayısı ile hesaplama sonucu veri noktalarının dağılımı ve küme merkezlerinin bulunduğu konumlar gösterilmiştir. JSBCO yöntemi ile kümeleme görünümü, Şekil 2 (b) ile karşılaştırıldığında BCO ile kümelemedeki gibi bir davranış sergilediği görülmektedir. Bunun nedeni yöntemin BCO algoritmasını esas alan bir yöntem olmasıdır. Ancak JSBCO algoritması, küme sayısını tahmin edebilmesi yönüyle incelendiğinde Şekil 3(c)'deki MPE-BCO yöntemine göre daha başarılı olduğu görülmektedir.

Tablo 2. BCO, JSBCO ve MPE ile BCO Yöntemlerinin Karşılaştırması

Veri Seti	BCO		MPE - BCO		JSBCO	
	Gerçek Küme Sayısı	Amaç Fonksiyonu (J_m) değeri	MPE ile Tahmin Edilen Küme Sayısı	Amaç Fonksiyonu (J_m) Değeri	JSBCO ile Tahmin Edilen Küme Sayısı	Amaç Fonksiyonu (J_m) Değeri
Path-based1	3	962.5024	6	526.6789	3	512.5141
Compound	6	588.9485	6	587.5898	6	523.1621
Aggregation	7	1502.8511	6	1502.2875	7	1465.8436
R15	15	167.9876	11	225.3542	13	187.3340
İris	3	71.5194	5	48.6291	2	108.9791
Glass	6	68.9137	4	104.3339	6	63.7615



Şekil 2. Aggregation Veri Seti için; (a) Gerçek Küme Görünümü ve (b) BCO ile Kümeleme Görünümü



Şekil 3. Aggregation Veri Seti için; (c) MPE-BCO ile Kümeleme Görünümü ve (d) JSBCO ile Kümeleme Görünümü

Ayrıca JSBCO yöntemi, MPE-BCO yöntemine göre kararlı sonuçlar üreten bir yöntemdir. Çünkü JSBCO algoritması, MPE yönteminden farklı olarak üyelik matrisini küme tahmininden sonra sadece bir kez hesaplamaktadır. MPE yöntemi kullanılarak uygulanan BCO algoritması uygun küme sayısına ulaşıncaya kadar, her yeni küme sayısı için rastgele bir üyelik matrisi oluşturarak BCO algoritmasını çalıştırmaktadır. JSBCO algoritmasında ise maksimum küme sayısı kadar küme merkezi, verinin içinden rastgele seçilmektedir. Bu algoritma, belirlenen eşik değeri ile karşılaştırma yaparak küme sayısını tahmin etmektedir. Küme sayısını tahmin ettikten sonra bir kez üyelik matrisini oluşturmakta olup sadece bir kez BCO algoritmasını çalıştırmaktadır. Bu durumun karşılaştırılması amacıyla veri setleri için MPE-BCO ve JSBCO yöntemlerini ürettiği küme sayıları Tablo 3 ile verilmiştir. 3 kümeden oluşan Pathbased1 veri seti için sonuçlar incelendiğinde, 10 farklı çalışmada JSBCO yönteminin 5 kez doğru kümelemeyi yakaladığı ve diğer tahminlerinin gerçek küme sayısına yakın sonuçlar verdiği görülmektedir. MPE-BCO yöntemi ise aynı veri setinde hiç doğru tahminde bulunamamış ve küme sayısı tahminleri gerçek küme sayısından uzak kalmıştır. Küme sayısı diğer veri setlerinden fazla olan 15 kümeli R15 veri seti incelendiğinde, JSBCO algoritmasının 2 kez gerçek küme sayısını verdiği ve diğer çalıştırmalarında 11 ve 16 arasında tahminler yaparak gerçek küme sayısına yakınsadığı görülmektedir. Ancak MPE-BCO yönteminin R15 veri seti için 1 kez doğru tahminde bulunduğu ve diğer tahminlerinin 5 ve 18 gibi gerçek küme sayısından uzakta ve birbiri ile alakasız olduğu fark edilmektedir. Pathbased1 gibi düşük küme sayısında, R15 gibi yüksek küme sayısında ve İris ve Glass gibi özellik sayısının fazla olduğu veri setlerinde JSBCO yönteminin küme sayısını tahmin etmede veya gerçek küme sayısı tahminine yakınsamada MPE-BCO yöntemine göre daha başarılı olduğu görülmektedir.

Ayrıca Aggregation veri seti kullanılarak iki algoritmanın küme sayısı tahminleri ve amaç fonksiyonlarındaki değişim Tablo 4'te gösterilmiştir. Aynı veri seti için MPE-BCO ve JSBCO algoritmaları 10 kez çalıştırılmıştır. 10 farkı çalışma sonucunda, MPE-BCO yöntemi verinin gerçek küme sayısı olan 7 küme için 2 kez doğru tahmin ederek %20 doğruluk elde ederken, JSBCO algoritması 8 kez tahmin ederek %80 oranında doğru tahminde bulunmuştur.

Tablo 3. Veri Setleri için MPE-BCO ve JSBCO Yöntemlerinin Kümeleme Sonuçları

Çalıştırma İndeksi	Veri Setleri için Elde Edilen Küme Sayıları											
	Pathbased1		Compound		Aggregation		R15		İris		Glass	
	MPE - BCO	JSBCO	MPE - BCO	JSBCO	MPE - BCO	JSBCO	MPE - BCO	JSBCO	MPE - BCO	JSBCO	MPE - BCO	JSBCO
1	6	4	5	4	6	7	9	12	5	5	8	6
2	7	3	7	6	5	7	11	12	9	2	5	7
3	6	4	6	6	3	7	8	13	5	2	6	6
4	6	3	6	6	7	7	5	16	5	2	4	7
5	6	3	6	8	6	8	5	15	8	3	5	6
6	5	3	6	7	5	7	11	15	9	3	4	6
7	6	3	5	4	7	7	14	13	4	2	4	7
8	6	2	6	6	6	7	18	14	6	3	4	7
9	6	4	5	6	5	6	15	11	10	3	6	7
10	6	5	7	3	6	7	11	13	7	2	4	6

Tablo 4. Aggregation Veri Seti için MPE-BCO ve JSBCO Yöntemlerinin Karşılaştırılması

Çalıştırma İndeksi	MPE – BCO		JSBCO	
	MPE ile Bulunan Küme Sayısı	Amaç Fonksiyonu (J_m) Değeri	JSBCO ile Bulunan Küme Sayısı	Amaç Fonksiyonu (J_m) Değeri
1	6	1502.2617	7	1502.9480
2	5	1731.6663	7	1466.2350
3	3	2427.3602	7	1506.2454
4	7	1332.8481	7	1486.4336
5	6	1507.0792	8	1288.6918
6	5	1737.9658	7	1545.7768
7	7	1331.4982	7	1465.4474
8	6	1486.6588	7	1465.8436
9	5	1732.0810	6	1731.6657
10	6	1506.5658	7	1465.9470

4. Sonuç

Bu çalışmada, BCO kümeleme algoritması temel alınarak, küme sayısını tahmin edebilen yeni bir JSBCO algoritması önerilmiştir. JSBCO algoritması BCO için kullanılan bir küme tahmin yöntemi olan MPE ile 6 farklı veri seti kullanılarak karşılaştırılmıştır. Bu karşılaştırmaya göre elde edilen bulgular JSBCO algoritmasının küme tahmininde daha iyi sonuç verdiğini göstermiştir. JSBCO algoritmasının küme sayısını tahmin etmede daha kararlı davrandığı, gerçek küme sayısına yakınsayan tahminler yaptığı sonucuna varılmıştır. Ayrıca küme sayısı veya özellik sayısının daha fazla olduğu veri setlerinde, JSBCO algoritması küme sayısını tahmin etmede ve yakınsama özelliğini devam ettirmede başarılı bulunmuştur. Küme sonucunu belirleyen üyelik matrisinin rastgele başlatılması ve veri büyüdükçe yüksek boyutlu verinin doğru kümeleme sonucunu bulmakta zorlanması gibi BCO algoritmasında görülen dezavantajlar JSBCO algoritmasında da mevcuttur. Ancak önerilen yöntemin BCO algoritmanın küme sayısına göre kümeleme yapma zorunluluğunu gideren başarılı bir sistematik ortaya koyduğu açıktır. JSBCO yönteminin BCO algoritmasının diğer dezavantajlarını gideren farklı önerileri gelecek çalışması kapsamındadır.

Kaynakça

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & geosciences*, *10*, 191-203. doi: 10.1016/0098-3004(84)90020-7.

Boydell, B., & McBratney, A.B. (2002). Identifying potential within-field management zones from cotton-yield estimates. *Precision Agriculture*, *3*(1), 9-23. doi: 10.1023/A:1013318002609.

Chang, H., & Yeung, D.Y. (2008). Robust path-based spectral clustering. *Pattern Recognition*, *41*(1), 191-203. doi: 10.1016/j.patcog.2007.04.010.

Ezugwu, A. E., Shukla, A. K., Agbaje, M.B., Oyelade, O. N.,

José-García, A., & Agushaka, J. O. (2021). Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature. *Neural Computing and Applications*, *33*(11), 6247-6306.

Gionis, A., Mannila, H., & Tsaparas, P. (2007). Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data*, *1*(1) 4-es. doi: 10.1145/1217299.1217303.

Govender, P., & Sivakumar, V. (2020). Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019). *Atmospheric Pollution Research*, *11*(1), 40-56. doi: 10.1016/j.apr.2019.09.009.

Hruschka, E. R., Campello, R. J., & Freitas, A. A (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, *39*(2), 133-155. doi: 10.1109/TSMCC.2008.2007252.

Li, Y. L., & Shen. Y. (2010). An automatic fuzzy c-means algorithm for image segmentation. *Soft Computing*, *14*(2), 123-128. doi: 10.1007/s00500-009-0442-0.

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, *37*(1), 145-151.

Maulik, U., & Saha, I. (2010). Automatic fuzzy clustering using modified differential evolution for image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *48*(9), 3503-3510.

Rahman, M. A., & Islam. M. Z. (2014). A hybrid clustering technique combining a novel genetic algorithm with K-Means. *Knowledge-Based Systems*, *71*, 345-365. doi: 10.1016/j.knosys.2014.08.011.

Schenatto, K., de Souza, E. G., Bazzi, C. L., Gavioli, A., Betzek, N. M., & Beneduzzi, H. M. (2017). Normalization of data for delineating management zones. *Computers and Electronics in Agriculture*, *143*, 238-248. doi: 10.1016/j.compag.2017.10.017.

Veenman, C. J., Reinders, M. J. T., & Backer, E. (2002). A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(9), 1273-1280. doi: 10.1109/TPAMI.2002.1033218.

Wikaisuksakul, S. (2014). A multi-objective genetic algorithm with fuzzy c-means for automatic data clustering. *Applied Soft Computing Journal*, *24*, 679-691. doi: 10.1016/j.asoc.2014.08.036.

Xiao, J., Yan, Y., Zhang, J., & Tang, Y. (2010). A quantum-inspired genetic algorithm for k -means clustering. *Expert Systems with Applications*, *37*(7), 4966-4973. doi: 10.1016/j.eswa.2009.12.017.

Zahn, C. T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, *100*(1), 68-86.

Dheeru, D., & Taniskidou, E. K. (2017). UCI machine learning repository.