Görsel Kümelenme Eğilimi Değerlendirmesi ve R'de Uygulaması

Zeynel CEBECİ [(1)] Figen YILDIZ[(2)]

Özet

Bölümleyici kümeleme algoritmalarında girdi olarak küme sayısı parametresi kullanılmakta ve kümelemenin başarısı büyük ölçüde analiz öncesi seçilen bu değere bağlı olmaktadır. Optimal küme sayısını bulmak için analiz sonrasında küme geçerliliği kontrolleri yapılsa da hem hesaplama zamanı maliyeti hem de kullanılan ölçütlerin veri yapılarına duyarlılığından etkilenmektedir. Bu nedenle küme sayısını analiz öncesi tahmin eden yöntemlere ihtiyaç duyulmaktadır. Görsel kümelenme eğilimi değerlendirmesi (GÖKED), küme sayısını bulmak için kullanılan öncü algoritmalardan biridir. Bu çalışmada görsel kümelenme eğilimi algoritmasının tanıtımı yapılarak R ortamında geliştirilen bir GÖKED fonksiyonu ile test edilmektedir.

Anahtar kelimeler: *Küme analizi, kümelenme eğilimi değerlendirmesi, veri görüntüleme*

A Programmatic Implementation of the Visual Assessment of Cluster Tendency Algorithm in R

Abstract

In cluster analysis, the partitioning algorithms require a priori estimate of number of clusters (*c*) as an input parameter, and thus the success of partitioning depends mostly on this parameter. In order to find an optimal *c*, the obtained results are checked by the various cluster validity indices at the end of each run of successive cluster analyses. Cluster validation is time consuming, and also depends on the clustering indices which may not guarantee the quality of clustering since their performances vary with complexity in data structures. In order to find an optimal number of clusters in data sets, one may benefit from the pre-analysis approaches before going to clustering. The visual assessment of clustering tendency (VAT) is a frontier algorithm which produces a grey-level image of reordered distance matrix showing existing clusters with dark blocks. This paper aims to introduce VAT algorithm and demonstrate it with a user-defined function developed in R statistical computing environment.

Keywords: *Cluster analysis, clustering tendency assessment, data visualization*

Giriş

Veri madenciliğinde ana konulardan biri gözlem verilerindeki anlamlı yapıları keşfetmektir. Bu iş genellikle kümeleme analizi (bölümleme, segmentasyon ya da taksonomi analizi olarak da bilinir) gibi açıklayıcı veri analiz yöntemleriyle gerçekleştirilmektedir (Prabhu ve Duraiswamy, 2013). Kümelemede amaç bir nesneler setini gözlenen özellik verilerinden hesaplanan bazı iyi tanımlanmış benzerlik ölçülerini kullanarak *c* adet alt sete (kümeye) bölümlemektir (Hu, 2012). Hiyerarşik kümelemede ihtiyaç duyulmasa da bölümleyici kümeleme algoritmalarında girdi olarak küme sayısı da kullanılmakta ve bu nedenle önceden bilinmesi gerekmektedir. Yapılan kümelemenin kalitesi önemli düzeyde bu girdi değerine bağımlı olmaktadır. Kümeleme kalitesi genellikle kümeleme sonrasında gerçekleştirilen küme geçerlilik ölçüleriyle belirlenmektedir. Bu durumda ise en uygun (optimal) kümeleme sonucuna ulaşmak için kullanılan algoritmaların ardı ardına çok sayıda çalıştırılması gerekmektedir (Krishnamoorthi, 2011). Böylece hem işlem zamanı açısından yüksek maliyet (Pakhira, 2012) hem de kullanılacak geçerlilik indekslerinin farklı veri yapılarına duyarlılıkları

Yayın Kuruluna Geliş Tarihi: 09.06.2015
Prof.Dr., Çukurova Üniversitesi Ziraat Fakültesi Zootekni Bölümü Biyometri ve Genetik Anabilim Dalı, Sarıçam- Adana. zcebeci@cu.edu.tr
Ar.Gör., Çukurova Üniversitesi Fen Bilimleri Enstitüsü Zootekni Anabilim Dalı, Sarıçam- Adana. yildizf@cu.edu.tr

itibariyle birtakım dezavantajlar ortaya çıkmaktadır. Dahası hangi bölümleme algoritması kullanılırsa kullanılsın, veri yapısında anlamlı bir kümelenme olmadığında bile yapılan analizler mutlaka $2 \leq c \leq n$ arasında bir kümeleme sonucu verecektir. Bu nedenle kullanılacak kümeleme algoritmasına karar vermeden çok daha önce veri yapısında kümeler olup olmadığını anlamak ve eğer varsa kaç tane olduğu belirlemek önemlidir. Bu amaçla gerçekleştirilen işlemlere *kümelenme eğilimi değerlendirmesi (clustering tendency assessment)* denilmektedir (Hu, 2012).

Veri setindeki küme sayısını tahmin etmek için geçerlilik indekslerine dayanan birçok algoritma bulunmakla birlikte *Görsel Kümelenme Eğilimi Değerlendirmesi (VAT: Visual Assessment of Cluster Tendency)* algoritması görsel teknikleri öncülerindendir. İlk olarak Bezdek ve Hathaway (2002) tarafından önerilen Görsel Kümelenme Eğilimi Değerlendirmesi (GÖKED) algoritması, aslında Prim algoritmasının (Prim, 1957) bir uyarlaması olup benzeşmezlik matrisinin (dissimilarity matrix) yeniden düzenlenmesine dayanmaktadır. GÖKED algoritması ile benzeşmezlik matrisi yeniden düzenlenmekte ve elde edilen *Yeniden Düzenlenmiş Benzeşmezlik Görüntüsünün (RDI: Reordered Dissimilarity Image)* incelemesi yapılmaktadır. *Kümelenme Eğilimi Görüntüsü (KEG)* olarak adlandırılan bu görüntüde köşegen boyunca yer alan karanlık dikdörtgen blokların sayısı verideki optimal küme sayısını vermektedir. Sayma işlemi gözle incelenerek yapılabileceği gibi *Karanlık Blok Çıkarma (DBE: Dark Block Extraction)* algoritmaları ile otomatikleştirilebilmektedir.

GÖKED, veri yapısındaki kümelerin iyi ayrılmış ve kompakt olduğu hallerde başarıyla kullanılabilmektedir. Ancak veri yapısında çakışan veya düzensiz geometrik şekilli kümeler olduğunda iyi sonuçlar vermeyebilmektedir (Malarvizhi ve Jayanthi, 2013). Ayrıca küçük veri setlerinde efektif iken büyük veri setlerinde zaman alıcı ve maliyetli olduğu da görülmüştür (Krishnamoorthi, 2011). Bu sorunları çözmek üzere sonraki yıllarda yeni algoritmalar geliştirilmiştir. Örneğin benzeşmezlik matrisini doğrusal olmayan özellik çıkarımından (non-linear feature extraction) sonra kompakt bir özellik uzayında hesaplamak daha iyi sonuçlar verebilmektedir (Havens ve ark., 2009).

GÖKED algoritması, büyük veri setlerinde KEG görüntüleri için bigVAT (Huband, Bezdek ve Hathaway, 2005) ve sVAT (Hathaway, Bezdek ve Huband, 2006) ile iyileştirilmiş ve en son olarak eşanlı kümeleme yaparak eğilim belirleyebilen coVAT geliştirilmiştir. Bunlar arasında öGÖKED (sVAT) uyarlamasında veri setinden bir alt veri seti seçilerek orijinal benzeşmezlik matrisinin boyutu indirgenmektedir. Yine GÖKED algoritması bulanık kümeleme için de revize edilerek (reVAT) (Bezdek ve ark., 2007) ve iyileştirilmiş bir sürüm olarak iVAT algoritması da önerilmiştir (Havens ve Bezdek, 2012).

Orijinal GÖKED çıktılarını kullanan Köşegen İzlemeli Görsel Kümelenme Eğilimi Değerlendirmesi (VATdt: Visual Assessment of Cluster Tendency Using diagonal tracing) adı verilen bir başka algoritma daha önerilmiştir (Hu, 2012). Puniethaa'nın Geliştirilmiş Görsel Kümeleme Eğilimi (E-VAT: Enhanced Visual Assessment of cluster Tendency) algoritması ve bunun kompakt olmayan karmaşık veri yapıları için iyileştirilmiş sürümü GE-VAT algoritmasının da başarılı sonuçlar verdikleri bildirilmektedir (Prabhu ve Duraiswamy, 2012; 2013). Son olarak Malarvizhi ve Jayanthi (2013)'nin scoiVat yaklaşımı ile iVAT, coVAT ve specVAT algoritmalarını birlikte kullanan bir teknik önerilmiştir.

KEG görüntülerini iyileştirmek ve küme sayılarını otomatik saptamak üzere yukarıda sözü edilen algoritmalar üzerinde çalışmalar sürmekle birlikte temel GÖKED algoritmasının girdi olarak sadece benzeşmezlik matrisini kullanması algoritmayı hesaplama maliyeti açsından avantajlı kılmaktadır. Diğer yandan GÖKED uyarlamalarının çoğu girdi olarak GÖKED çıktısı yeniden düzenlenmiş

matrislerini kullandığından GÖKED algoritması veri yapısındaki kümelenme eğilimlerinin incelenmesinde temel algoritma durumundadır. Buna karşın Ferraro ve Giordani (2015)'in `fclust` isimli R paketindeki `VAT` fonksiyonu; Hahsler, Hornik ve Buchta (2008)'nın `seriation` paketinde henüz geliştirme altındaki VAT ve iVAT fonksiyonları ile Havens ve Bezdek (2012)'in yayınladığı bazı Matlab kodları dışında GÖKED algoritmasına ait kod ve uygulama bulunmamaktadır. Bu nedenle GÖKED algoritması kullanan yazılım veya uygulama araçlarına ihtiyaç bulunmaktadır. Bu çalışmada, GÖKED algoritmasının tanıtımı yapılarak R istatistik ortamı için geliştirilen bir fonksiyon ve bir örnek veri setinde kullanımı örneklenmektedir.

GÖKED Algoritması

Bir nesneler seti ($\boldsymbol{O}$), o_i i. fiziksel nesneyi göstermek üzere:

$$\boldsymbol{O} = \{o_1, \dots, o_n\} \quad (1)$$

şeklinde yazılabilir. Eşitlik 1'deki $\boldsymbol{O}$ nesne seti genelde iki farklı şekilde temsil edilir. o_i nesnesi $\mathrm{x_i} \in \mathbb{R}^p$ ile temsil edildiğinde:

$$\boldsymbol{X} = \{\mathbf{x_1}, \dots, \mathbf{x_n}\} \subset \mathbb{R}^p \quad (2)$$

$\boldsymbol{O}$'nun bir *nesne verisini* ifade eder. Eşitlik 2'deki $\boldsymbol{X}$'de $\mathbf{x_i}$'nin p bileşeni vardır ve bunlar o_i nesnesinin p özelliğini temsil etmektedirler. Buna göre i. nesne verisi için özellikler vektörü:

$$\mathbf{x_i} = \{x_{1i,} \dots, x_{pi}\} \quad (3)$$

olup burada x_{ji}, $\mathbf{x_i}$ 'de j. özelliğin değerini göstermektedir $(1 \leq j \leq p)$. Eşitlik 1'deki $\boldsymbol{O}$'nun veri çiftleri *ilişkisel veri* olarak adlandırılan bir ilişki ile de temsil edilebilirler. o_i ve o_j gibi herhangi iki nesne arasındaki ilişki r_{ij} ile gösterildiğinde tüm nesneler arasındaki ilişkisel veriler $n \times n$ elemanlı:

$$\boldsymbol{R} = [r_{ij}]_{nxn} \quad (4)$$

bir simetrik matris ile gösterilirler. Eşitlik 4'deki R matrisinde ilişkiler ya da yakınlıkların ölçüsü olarak veri çiftlerinin benzerlikleri (s_{ij}) kullanılabilir. Veri çiftleri arasındaki benzerlikler farklı yöntemlerle hesaplandığında elde edilen benzerlikler matrisi:

$$\boldsymbol{S} = [s_{ij}]_{nxn} \quad (5)$$

şeklinde gösterilebilir. Ancak GÖKED algoritmasında benzeşmezlikler matrisi kullanıldığından Eşitlik 5'deki benzerliklerin Eşitlik 6'daki gibi bir dönüşümle ilişkiler matrisinde depolanması gerekir.

$$\boldsymbol{R} = \boldsymbol{S} = [s_{max} - s_{ij}]_{nxn} \quad (6)$$

Eşitlik 6'daki s_{max} değeri $\boldsymbol{S}$ matrisindeki en büyük benzerlik değeridir. Benzeşmezlikler doğrudan doğruya Eşitlik 2'deki $\boldsymbol{X}$ matrisi kullanılarak da hesaplanabilir. Bu amaçla $\mathbf{x_i}$ ve $\mathbf{x_j}$ arasındaki r_{ij} değerlerini hesaplamak için $\mathbb{R}^p$ özellik uzayında birtakım norm veya metrikler kullanılabilir. Nesne verileri benzeşmezlik değerleriyle temsil edildiğinde ($\boldsymbol{X} \rightarrow \boldsymbol{D}$) benzeşmezlikler d_{ij} gösterilebilirler. Dolayısıyla GÖKED algoritmasında kullanılacak ilişkiler örneğin Öklid normu ile $r_{ij} = d_{ij} = \|x_i - x_j\|$ şeklinde hesaplanabilirler.

$$\boldsymbol{R} = \boldsymbol{D} = [\, d_{ij}]_{nxn} \quad (7)$$

Eşitlik 7'deki $\boldsymbol{D}$ matrisinin görüntüsü genellikle bilgi sağlayıcı değildir ve bu nedenle yorumlanamaz. Ancak birbirine yakın nesneler gruplandırılarak $\boldsymbol{D}$ yeniden düzenlendiğinde bilgi verici olabilmektedir. GÖKED algoritması da özellik uzayında birbirine yakın noktaların genellikle benzer indislere sahip olacakları gerçeğinden hareketle $\boldsymbol{D}$ matrisinin yeniden düzenlenmesine dayanmaktadır. Bunun için $\boldsymbol{D}$ GÖKED algoritması ile yeniden düzenleme işlemine tabi tutularak $\boldsymbol{D}^*$ olarak gösterilen *Yeniden/Düzenlenmiş Benzeşmezlik Matrisi (R/ODM: Re/Ordered Dissimilarity Matrix)* elde edilmektedir. $\boldsymbol{D}^*$ matrisinde:

- $0 \leq d_{ij}^* \leq 1$ (8)

- $d_{ij}^* = d_{ji}^*$ (9)
- $d_{ii}^* = 0$ (10)

olup Eşitlik 8 benzeşmezlik değerlerinin [0,1] aralığında olduğunu, Eşitlik 9 matrisin simetrik olduğunu ve Eşitlik 10 ise bir veri noktasının kendisine uzaklığının yani kendine benzeşmezliğinin 0 olduğunu gösterir.

$\boldsymbol{D}^*$ matrisindeki değerler 0 ve 255 arasındaki gri ton renklerine çevrilerek bir $\boldsymbol{G}$ gri ton matrisi elde edilir. Görsel değerlendirmeye olanak sağlayan $\boldsymbol{G}$ matrisi görüntüsü *Kümelenme Eğilimi Görüntüsü (KEG)* olarak adlandırılır. $\boldsymbol{G}$'de piksel (i,j)'nin gri ton değeri (g_{ij}), $\boldsymbol{D}^*$ matrisindeki d_{ij}^* değerlerinden aşağıdaki gibi hesaplanır:

$$d_{ij}^* = 0 \Rightarrow g_{ij} = 0 \Rightarrow siyah\ görüntü\ elemanı$$
$$d_{ij}^* = 1 \Rightarrow g_{ij} = 255 \Rightarrow beyaz\ görüntü\ elemanı$$

Buna göre 0'a eşit d_{ij}^* değerleri 0 değerli g_{ij} değerlerine dönüştürülerek siyah; en yüksek 1 değerli d_{ij} değeri en yüksek değerli $g_{ij} = 255$ değerine dönüştürülerek beyaz olarak görüntülenir. Bu durumda diğer değerlerin [0,255] aralığında gri ton değerleri olduğu anlaşılacaktır. Benzeşmezlik değerleri veri setlerine göre değişkenlik gösterebileceğinden d_{max}^* değerine göre ölçeklendirme yapılmalıdır. Bunun için $g_{ij} = int(255 * (\frac{d_{ij}^*}{d_{max}^*}))$ şeklinde bir dönüştürme formülü kullanılabilir.

$\boldsymbol{D}^*$ değerlerinin $\boldsymbol{G}$ değerlerine dönüştürülmesiyle n nesne arasındaki benzeşmezlik ölçüsü çiftleri $n \times n$ piksellik bir görüntü ile temsil edilmiş olurlar. Bu $\boldsymbol{G}$ matrisi görüntüsünde köşegen boyunca görünen karanlık (siyah) blokların her biri ayrı bir kümeyi gösterir ve görüntü kümelenme eğilimini açıklar. GÖKED algoritması aşağıda görülen adımlardan oluşmaktadır (Pakhira, 2012).

GÖKED algoritması

Girdi: $n \times n$ boyutlu $\boldsymbol{D}$ matrisi
Veri: $\boldsymbol{K} = \{1,2,\dots,n\}$;
$\boldsymbol{I} = \boldsymbol{J} = 0$;
P=(0, 0, …, 0)
Adım 1:
Seç $(i,j) \in argmax\{D_{pq}\}; p,q \in \boldsymbol{K}$;
$P(1) = i$; $\boldsymbol{I} = \{i\}$ ve $\boldsymbol{J} = \boldsymbol{K} - \{i\}$ ata
Adım 2:
$t = 2,3,\dots,n$ için yap
Seç $(i,j) \in argmin\{D_{pq}\}; p \in \boldsymbol{I}, q \in \boldsymbol{J}$;
$P(t) = j$ ata
$\boldsymbol{I} \leftarrow \boldsymbol{I} \cup \{j\}$ ve $\boldsymbol{J} \leftarrow \boldsymbol{J} - \{j\}$ olacak şekilde değiştir
Diğer t 'ye geç
Adım 3:
P dizisini kullanarak yeniden düzenlenmiş benzeşmezlik matrisini oluştur
$\boldsymbol{D}^* = [D_{p(i)q(j)}]; 1 \le i,j \le n$
Adım 4:
$\boldsymbol{D}^*$ matrisini $\boldsymbol{G}$ matrisine dönüştür
$\boldsymbol{G}$ matrisi görüntüsünü göster

Materyal ve Metot

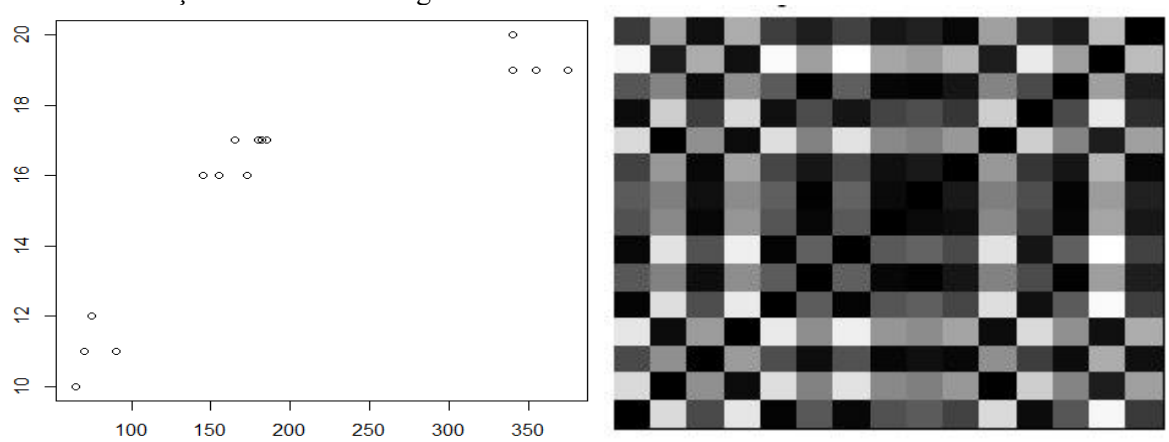
GÖKED algoritmasının yazılım implementasyonu için R'de `gokeda` fonksiyonu geliştirilmiştir (Ek 1). GÖKED algoritması çok boyutlu uzaylar için de kullanılabilmesine karşın bu çalışmada gözlemlerin serpilme durumlarını anlaşılır şekilde gösterebilmek amacıyla iki özellikli ($\boldsymbol{X} \subset \mathbb{R}^2$) bir test veri seti kullanılmıştır. Test veri setinde rastgele seçilen 15 gıda maddesinin enerji (kcal/100g) ve protein (g/100g) değerleri olmak üzere iki özellik bulunmaktadır. Test veri setini içeren `gokedatest.txt` dosyasının (Ek 3) özelliklere ait sütunların sekmelerle ayrıldıkları ve yerel diskin `rdata` isimli bir klasöründe bulunduğu varsayılmıştır. Analiz işlemlerini bir bütün halinde yürütmek üzere bir R programı yazılmış (Ek 2) ve 8GB RAM bellek kapasiteli i7 işlemcili bir bilgisayarda R ver 3.2.0 (R Core Team, 2015) altında test edilmiştir.

Bu çalışmada R'de kullanılmak üzere geliştirilen `gokeda` fonksiyonu Ferraro ve Giordani (2015)'in VAT fonksiyonunun iyileştirilmiş bir uyarlamasıdır. Ancak geliştirilen fonksiyon yalnız $\boldsymbol{X}$ matrisini değil aynı zamanda $\boldsymbol{D}$ matrisini de doğrudan girdi olarak kabul etmektedir. Bezdek ve Hathaway (2002)'nin GÖKED algoritması kodunu içeren fonksiyon çıktı olarak bir liste üretmekte olup listenin `odm` özelliği düzenlenmiş $\boldsymbol{D}$ matrisini,

`P` özelliği ise P vektörünü döndürmektedir. Çalışmada geliştirilen `gokeda` fonksiyonu böylece yukarıda sözü edilen fonksiyonlardan farklı olarak sadece KEG görüntüsü vermemekte aynı zamanda çıktıları diğer analizler için kullanışlı hale getirmektedir.

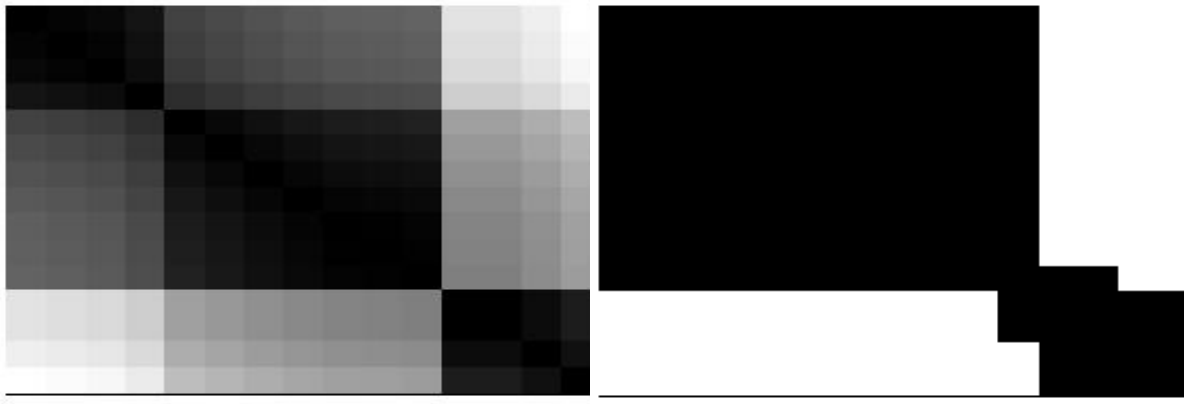
Bulgular ve Tartışma

İstatistik analizlerde veri setindeki dağılışları ve yapıları görebilmek için birçok grafikten yararlanılabilir. Serpilme diyagramları iki özellikli (değişkenli) bir veri setinde bulunan alt veri setlerinin (kümelerin) uzaydaki konumları, yönelimleri ve hacimlerini yani kümelenme eğilimlerini açıklamak için kullanılabilen grafiklerdir. Şekil 1a'da ***X*** veri setindeki noktaların bir serpilme diyagramı görülmektedir. Şekil 1a incelendiğinde test verisinde yer alan gıdaların enerji ve protein bileşenleri açısından birbirinden belirgin ya da iyi şekilde ayrılmış 3 alt kümeden oluştuğu anlaşılmaktadır. Ancak birçok uygulamada özellik sayısı 2'den fazla olabileceği gibi kümeler de birbirinden belirgin şekilde ayrılmamış olabilirler. Böyle durumlarda 2 boyutlu serpilme diyagramları açıklayıcı olamazlar. Gerçi özellik sayısı 3 olduğunda 3 boyutlu (3B) serpilme grafikleri kullanılabilse de daha fazla boyutlarda başka yöntemlere ihtiyaç duyulmaktadır. Aslında GÖKED algoritması ile amaçlanan hususlardan birisi de bu olup çok boyutlu uzayda dağılan veri setini iki boyutta yorumlanabilir hale getirmeye çalışılmaktadır.



(a) Serpilme diyagramı (b) ***D*** matrisi görüntüsü

Şekil 1. ***X*** serpilme grafiği ve ***D*** uzaklık matrisinin görüntüsü



(a) KEG (***D****) görüntüsü (b) İki tonlu KEG (binary) görüntüsü

Şekil 2. KEG görüntüsü ve iki tonlu KEG görüntüsü

X matrisinden hesaplanan uzaklıklar matrisine (***D***) ait Şekil 1b'deki görüntü veri çiftleri arasındaki uzaklıkların orijinal düzenini göstermektedir. Bu görüntü incelendiğinde ***X*** veri yapısı hakkında yorumlanabilir ya da kullanılabilir bir bilgi elde edilememektedir.
Buna karşın ***D*** matrisi GÖKED algoritması ile yeniden düzenlendiğinde ***X***'de birbirine yakın noktalar bir araya toplandıklarından kümelenme eğilimlerini gösterecek duruma gelirler. Şekil 2a'da yeniden düzenlenmiş ***D*** matrisinin yani ***D***** matrisinin görüntüsü verilmiştir. Kümelenme eğilimi görüntüsü (KEG) olarak adlandırılan bu görüntünün diyagonali boyunca 3 adet karanlık blok (dörtgen) olduğu görülmektedir. KEG görüntüsünün diyagonalinde yerleşen bu karanlık bloklar veri yapısında mevcut kümeleri temsil etmektedir. Nitekim Şekil 1a'daki serpilme diyagramında görülen üç kümenin varlığı Şekil 2a'daki KEG'de çok net şekilde seçilebilmektedir. KEG'lerde her bir blokun büyüklüğü ise söz konusu kümenin hacmini temsil etmektedir. Nitekim Şekil 2a'ya göre en büyük kümenin ortadaki küme olduğunu Şekil 1a'daki serpilme grafiğinden doğrulamak mümkündür.

Sonuç olarak KEG'ler veri yapısında kümelenme olup olmadığı ve varsa kümelerin sayısı ve büyüklükleri hakkında bilgi sağlayıcı olmaktadır (Bezdek, 2002; Havens ve ark., 2009). KEG'ler her durum için Şekil 2a'daki belirgin olmadığında Şekil 2b'deki gibi iki tonlu KEG'lerden de yararlanılabilmektedir. Diğer yandan iki tonlu görüntüler küme sayısını otomatik hesaplayan algoritmalarda girdi olarak da kullanılmaktadır.

Sonuç

Kümelenme eğilimi görüntüleri, nesnelerin özellik veri setlerinden hesaplanan yeniden düzenlenmiş benzeşmezlik ölçülerinin gri tonlu görüntüleri olan KEG'lerde diyagonal boyunca görülen karanlık blokların sayısı ve büyüklükleri veri setindeki kümeler ve kümelenme eğilimleri hakkında bilgi sağlayıcı olmaktadır. Bölümleyici kümeleme analizlerinde KEG'leri inceleyerek saptanan küme sayısını parametre olarak kullanmak kümeleme analizindeki başarıyı arttıracak ve maliyeti düşürecektir. Ancak bununla ilgili deneysel çalışmalara ihtiyaç bulunmaktadır.

Bu çalışma kapsamında geliştirilen `gokeda` fonksiyonu R ortamında yapılacak analizlerde doğrudan kullanılabilir durumdadır. Ancak program kodları kolaylıkla diğer istatistik analiz ortamları ve dillerine de uyarlama için örnek olarak kullanılabilir. GÖKED algoritması kümeleme analizinde gerek duyulan optimal küme sayısını önceden belirlemek için bir yeni bir seçenek olarak veri yapısındaki kümelerin ayrık olması ve küçük hacimli olması halinde çok başarılı sonuçlar vermektedir. Büyük ve karmaşık veri yapıları için ileri GÖKED sürümleri ya da uyarlamalarının geliştirilmesi; otomatik sayma araçlarının tasarlanması kümeleme analizi ve dolayısıyla veri madenciliğinde önemli kolaylıklar sağlayabilecektir.

Kaynaklar

Bezdek, J.C. and R.J. Hathaway (2002). VAT: A tool for Visual Assessment of (Cluster) Tendency. *Proc. of IEEE Int. Joint Conference on Neural Networks* (IJCNN 02), 12-17 May 2002, vol. 21, pp. 2225-2230.

Bezdek, J.C., Hathaway, R.J. and J. M. Huband (2007). Visual Assessment of Fuzzy Clustering Tendency for Rectangular Dissimilarity Matrices., *IEEE Transactions on Fuzzy Systems*, 15(5): 890-903.

Ferraro, M.B. and P. Giordani (2015). A toolbox for fuzzy clustering using the R programming language. *Fuzzy Sets and Systems*. http://dx.doi.org/10.1016/j.fss. 2015.05.001).

Hahsler, M., Hornik, K. and C. Buchta (2008). Getting things in order: An introduction to the R package seriation. *J Statistical Software*, 25(3): 1-34.

Hathaway, R.J., Bezdek, J.C. and J. M. Huband (2006). Scalable Visual Assessment of

Cluster Tendency. *Pattern Recognition*, 39(6):1315-1324.

Havens, T.C., Bezdek, J.C., Keller, J.M. and M. Popescu (2009). Clustering in Ordered Dissimilarity Data. *Int. J. of Intelligent Systems*, 24, 504–528.

Havens, T.C. and Bezdek, J.C. (2012). "An Efficient Formulation of the Improved Visual Assessment of Cluster Tendency (iVAT) Algorithm", *IEEE Transactions on Knowledge & Data Engineering*, 24 (5) 5:813-822.

Hu, Y. (2012). VATdt: Visual Assessment of Cluster Tendency Using Diagonal Tracing, *American J of Computational Mathematics*, 2: 27-41**.**

Hu, Y, R. and J. Hathaway (2008). An Algorithm for Clustering Tendency Assessment, *WSEAS Transactions on Mathematics*, 7(7): 441-450.

Huband, J.M., Bezdek, J.C and R.J. Hathaway (2005). BigVAT: Visual Assessment of Cluster Tendency for Large Data Set. *Pattern Recognition*, 38(11):1875-1886.

Krishnamoorthi (2011). Automatic Evaluation of Cluster in Unlabeled Datasets. *Proc. of Int.Conf. on Information and Network Technology*. IACSIT Press, Singapore. pp 120-124.

Malarvizhi, M and S. Jayanthi (2013). Visualization of Clusters Using scoiVat Algorithm. *International Journal of Educational Science and Research*, 3(3): 35-40.

Pakhira, M.K (2012). Finding Number of Clusters before Finding Clusters. *Procedia Technology*, 4: 27-37.

Prabhu, P. and K. Duraiswamy (2012). Enhanced VAT for Cluster Quality Assessment in Unlabeled Datasets. *J. of Circuits, Systems and Computers*, 21(1): 1-19.

Prabhu, P., K. Duraiswamy (2013). An Efficient Visual Analysis Method for Cluster Tendency Evaluation, Data Partitioning and Internal Cluster Validation . *Computing and Informatics*, 32: 1013-1037.

Prim, R. (1957). Shortest connection networks and some generalisations. *Bell System Tech J*, 36:1389–1401.

R Core Team (2015). R: A language and environment for statistical computing R Foundation for Statistical Computing, Vienna, Austria. URL:http://www.R-project.org

Ek 1. R'de gokeda fonksiyonu kodu

```
gokeda <- function(X, img=F, dist=F)
{
       if (missing(X))
              stop("Veri seti
       girilmedi.")
       if (is.null(X))
              stop("Veri seti boş,
       elemanı yok.")
       n = nrow(X)
       X = as.matrix(X)
       if (any(is.na(X)))
              stop("Veri setinde NA
       değerler olamaz.")
       if (!is.numeric(X))
              stop("Veri seti sayısal
       bir matris veya veri tablosu
       olmalıdır.")
       if (is.null(rownames(X)))
              rn = paste("Nesne", 1:n,
       sep = " ")
       else rn = rownames(X)
       if(!dist)
              D = as.matrix(dist(X))
else{
       D = X
       if(typeof(D)!= "dist")
              stop("Girilen
       veri seti uzaklıklar
       matrisi değil.")}
I = rep(0, n)
P = c()
mx = which(D == max(D),
arr.ind = TRUE)[2]
I[mx] = 1
P[1] = mx
for (i in 2:n){
       Dr = matrix(D[I > 0, ],
nrow = sum(I))
       Dr[, I == 1] = max(Dr)
       mn = (which(Dr ==
min(Dr), arr.ind = TRUE))[2]
       I[mn] = 1
       P[i] = mn
}
Ds = D[P, rev(P)]
if(img)
```

```
        image(Ds, col =
    grey(seq(0, 1, length = 256)),
              xlab = "", ylab =
        "",
              xaxt = "rn", yaxt
        = "rn",
              main = "VAT
        Grafiği")
    return (list(odm=Ds, P=P))
}
```

Ek 2. R'de GÖKED Analizi uygulama kodu

```
## gokedatest.R
#
# Çalıştırmadan önce gokeda
fonksiyonunu buraya yapıştırınız.

par(mfrow=c(2,2),  oma = c(0.5, 0.5,
0.5, 0.5), mar = c(2, 2, 2, 2),
ask=T)

# Çalışma klasörünü seç
setwd("c:/rdata")

# Veriyi oku ve ds verisetine ata
ds <- read.table("gokedatest.txt",
head=T)

# Tablo olarak okunan veriyi X
matrisine ata
X <- as.matrix(ds)

#Grafik sonuçlarını kaydetmek
isterseniz aşağıdaki satır başındaki
# kaldırınız
#pdf("gokedaout.pdf")
```

Ek 3. Test veri seti dosyası (gokedatest.txt)

```
Enerji Protein
75     12
340    20
165    17
355    19
70     11
182    17
65     10
173    16
185    17
155    16
340    19
90     11
180    17
375    19
145    16
```

```
plot(X, xlab="", ylab="", main="X
Serpilimi")

# Uzaklıkları D matrisine ata
dst <- dist(X, diag=T, upper=T)
D <- as.matrix(dst)

# plot(D, xlab="", ylab="", main="D
Serpilimi")

# Uzaklıklar için G Matrisini oluştur
ve serpilme diyagramını çiz
G <- round(D/max(D)*255, 0)

gri256 <- grey(seq(0, 1, length =
256))
gri2  <- grey(seq(0, 1, length = 2))

# Orijinal uzaklıklar (D) matrisini
görüntüle
image(G, col = gri256, xlab = "",
ylab = "", xaxt = "n", yaxt = "n")
title(main="Orijinal D Görüntüsü",
font.main=15)

# Kümelenme eğilimi (KEG) görüntüsü
rdm <- gokeda(X, img=F)
DR <- rdm$odm
image(DR, col = gri256, xlab = "",
ylab = "", xaxt = "n", yaxt = "n")
title(main="KEG", font.main=15)

# Kümelenme eğilimi (KEG) ikili
görüntüsü
BDR <- round(DR/max(DR),0)
image(BDR, col = gri2, xlab = "",
ylab = "", xaxt = "n", yaxt = "n")
title(main="Binary KEG",
font.main=15)

## program sonu
```

Açıklama

İngilizcesi "cluster tendency" olan terim Türkçe'ye ilk kez bu makalede "kümelenme eğilimi" olarak çevrilmiştir. "Kümeleme" yerine "kümelenme" teriminin seçilmesinin nedeni terimin veri yapısında mevcut olan durumu ifade etmesidir. Bu nedenle kümelere ayırma işlemi anlamında kullanılan "kümeleme" teriminden farklıdır. Yani "kümelenme" nesnelerle ilgili mevcut eğilimi anlamayı, "kümeleme" ise nesneleri belli bir yöntemle ayırmayı ifade etmektedir.