İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN EDEBİYAT FAKÜLTESİ

MATEMATİK MÜHENDİSLİĞİ PROGRAMI

ÖĞRENCİ HARF NOTLARININ K-MEANS KÜMELEME ALGORİTMASI İLE BELİRLENMESİ

BİTİRME ÖDEVİ

Ece FIRAT 090070026

Tez Danışmanı: Yar. Doç. Dr. Ahmet KIRIŞ

MAYIS 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN EDEBİYAT FAKÜLTESİ
MATEMATİK MÜHENDİSLİĞİ PROGRAMI

ÖĞRENCİ HARF NOTLARININ K-MEANS KÜMELEME ALGORİTMASI İLE BELİRLENMESİ

BİTİRME ÖDEVİ

Ece FIRAT 090070026

Teslim Tarihi: 21.05.2012

Tez Danışmanı: Yar. Doç. Dr. Ahmet KIRIŞ

MAYIS 2012

ÖNSÖZ

Bu çalışmayı hazırlarken her türlü yardımı ve desteği fazlasıyla sağlayan, hiçbir şekilde esirgemeyen saygıdeğer hocam Sayın Yar. Doç. Dr. Ahmet KIRIŞ'a, okul hayatım boyunca bana destek olan tüm arkadaşlarıma ve yaşantım boyunca her daim yanımda olan, bana sevgi, güven ve her türlü desteği veren aileme en içten teşekkürlerimi sunarım.

Mayıs, 2012 **Ece FIRAT**

İÇİNDEKİLER

	Sayfa
ÖZET	**vi**
1. GİRİŞ	**1**
2. KÜMELEME ALGORİTMALARI	**2**
2.1. Kümeleme Analizinin Tanımı	2
2.2. Kümeleme Analizinin Özellikleri	3
2.3. Kümeleme Analizi Veri Türleri	4
2.3.1. Aralık Öncelikli Değişkenler	4
2.3.2. İkili değişkenler	5
2.3.3. Nominal, Ordinal ve Oran Değişkenleri	6
2.3.3.1. Nominal Değişkenler	6
2.3.3.2. Ordinal Değişkenler	6
2.3.3.3. Oran Ölçekli Değişkenler	6
2.3.4. Karışık Tür Değişkenler	7
2.4. Kümeleme Yöntemleri	7
2.4.1. Bölümleme Yöntemleri	7
2.4.1.1. K-medoids Algoritması	8
2.4.1.2. CLARA ve CLARANS Algoritmaları	8
2.4.2. Hiyerarşik Yöntemler	9
2.4.3. Yoğunluk Tabanlı Yöntemler	10
2.4.4. Izgara Tabanlı Yöntemler	10
2.4.5. Model Tabanlı Yöntemler	10
3. K – MEANS ALGORİTMASI	**11**
3.1. Tanım ve Tarihçe	11
3.2. K-means Algoritmasının Adımları	13
3.3. *k* Sayısının Kümelemeye Etkisi	15
3.3.1. Geometrik Hesaplama	17
3.3.2. Aritmetik Hesaplama	18
4. UYGULAMA: ÖĞRENCİ HARF NOTLARININ BELİRLENMESİ	**19**
KAYNAKLAR	**25**
EKLER	**26**

ÖZET

Kümeleme algoritmalarından bölünmeli kümeleme tekniği, nesneleri giriş parametre sayısı kadar kümeye bölmektedir. Bölünmeli kümeleme algoritmaları, merkez tabanlı kümeleri tespit etmede başarılıdır. Bu çalışmada başlıca kümeleme algoritmalarından K-means algoritması anlatılmış ve K-means algoritmasının bir veri üzerindeki uygulaması Mathematica programı ile gerçekleştirilmiştir.

1. GİRİŞ

İnsan beyni veriler üzerinde benzerlikler bulma konusunda eğilimlidir. Bu benzerlikleri açıklama ve uygulama konusunda geliştirilen yöntemlerden birisi de birbirine benzer özelliklere sahip verileri ortak gruplara, kategorilere yerleştirmektir. Bu gruplama işlemi bazen verilerin özelliklerine göre yapılmış, bazen de özellikleri göz ardı edilerek rastgele yapılmıştır. Özelliklerine göre yapılan gruplamalarda, aynı grupta olan verilerin benzer özelliklerinin maksimum, farklı grupta olan verilerin ise benzer özelliklerinin minimum olması hatta benzer özelliklerinin bulunmaması amaçlanmıştır. Örneğin, Yunan filozof Aristoteles, canlıları yaşama ortamlarını (hava, su, kara) temel alarak sınıflandırmıştır [1].

Bir veri kümesindeki bilgileri belirli yakınlık ölçütlerine göre gruplara ayırma işlemine kümeleme analizi denir. Kümeleme işleminde küme içindeki elemanların benzerliği fazla, kümeler arası benzerlik ise az olmalıdır [2]. Kümeleme analizi bireylerin ya da nesnelerin sınıflandırılmasını ayrıntılı bir şekilde açıklamak amacıyla geliştirilmiştir. Bu algoritmaların verimliliği algoritmanın uygulandığı veriye göre değişir fakat her tür veri için kabul edilebilir sonuçlar veren algoritma k-means kümeleme algoritmasıdır.

Bu bitirme projesinde, k-means kümeleme algoritmasının bir uygulaması yapılmıştır. Algoritmanın sonuçlarını görmek için öğrencilerin yılsonu başarı puanlarına göre kümeleme yapılarak harf notları belirlenmiştir.

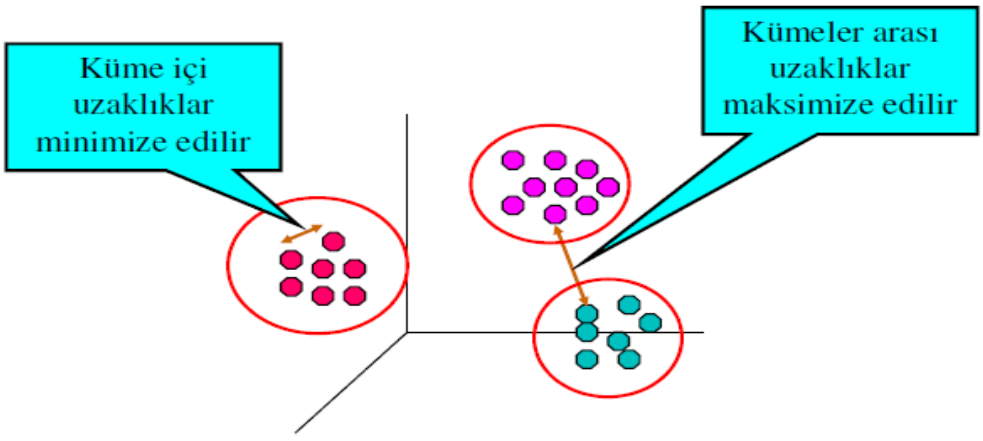
2. KÜMELEME ALGORİTMALARI

2.1 Kümeleme Analizinin Tanımı

Kümeleme psikolojide, doğrusal olmayan beyin fırtınası yardımıyla ussal(sol beyin) ve sezgisel(sağ beyin) düşünce şemaları arasında karşılıklı etkileşimle ortaya çıkan uyumu ifade etmektedir ve ilk defa Gabriele Rico tarafından kullanılmıştır. Kümeleme, düşüncelerin zihinsel olarak haritalanması demektir. Bu sözcük düşünce veya zihin haritalama şeklinde de isimlendirilmektedir. Kavramlar bir ağ modeli içinde gerçekleşir. Kümeleme de biçimsel olarak bir kavramlar ağından oluşmaktadır [3].

Kümeleme analizi, bir veri kümesindeki bilgileri belirli yakınlık kriterlerine göre gruplara ayırma işlemidir. Bu grupların her birine 'küme' denir. Kümeleme analizine ise kısaca kümeleme adı verilir [2].

Kümeleme en basit tanımıyla benzer özellik gösteren veri elemanlarının kendi aralarında gruplara ayrılmasıdır. Aynı küme içindeki elemanların benzerliği fazla, kümeler arası benzerlik ise az olmalıdır. Literatürde kümeleme analizini açıklayan birçok tanım bulunmaktadır [2]. Bu tanımlara göre her küme temsil ettiği nesneleri en iyi şekilde ifade edecek şekilde düzenlenir [4].



Şekil 2.1 : Kümelemenin amacı [10]

Kümeleme, gözetimsiz sınıflama yöntemidir. Gözetimli sınıflandırma işleminde veriler önceden sınıflandırılmış örüntülerdir. Burada temel amaç, yeni gelecek ve henüz hangi sınıfta olduğu bilinmeyen verilerin var olan sınıflardan en uygun olanına yerleştirilmesidir. Gözetimsiz sınıflamada ise amaç, başlangıçta verilen ve henüz sınıflandırılmamış bir küme veriyi anlamlı alt kümeler oluşturacak şekilde öbeklemektir. Kümeleme işlemi tamamen gelen verinin özelliklerine göre yapılır.

Kümeleme analizi istatistik, biyoloji, uzaysal veri madenciliği ve makine öğrenmesi, örüntü tanıma ve resim tanıma alanlarında kullanılmaktadır. İstatistik dünyasında k-means ve k-medoids kümeleme yöntemlerini kullanan S-Plus, SPSS ve SAS gibi paket programlar yoğunlukla kullanılmaktadır. Biyolojide genetik yapıların sınıflandırılması ve yeni yapıların keşfinde, uzaysal/düzlemsel veri madenciliğinde coğrafi konuma göre yerleşim yerlerine götürülecek mal ve hizmetler için ideal yer belirlemede, yapay zekâ alanında makine öğrenmesi için gözetimsiz öğrenme metodu olarak kullanılmaktadır [2].

2.2 Kümeleme Analizinin Özellikleri

İyi bir kümeleme analizi aşağıdaki özelliklere sahip olmalıdır [2]:

- Ölçeklenebilir olmalıdır. Birkaç yüz kayıttan oluşan veri kümesine de milyonlarca kayıt içeren kümeye de uygulanabilmelidir.
- Farklı veri türleri ile kullanılabilmelidir. Hem sayısal hem kategorik veriler içeren veri tabanlarında kullanılabilmelidir.
- Düzgün şekilli olmayan kümeleri de bulabilmelidir.
- En az sayıda giriş değişkeni gerektirmelidir. Bir yöntem ne kadar az giriş değişkeni gerektiriyorsa o ölçüde kullanıcının kararlarından bağımsızdır.
- Gürültü içeren veriler ile de kullanılabilmelidir.
- Veri kümesindeki kayıtların sıralanmasından bağımsız olmalıdır. Kümenin hangi elemanından başlanırsa başlansın sonuç değişmemelidir.
- Çok boyutlu veri tabanlarına uygulanabilmelidir.
- Veri kümesinin sahip olduğu sınırlıkları dikkate alabilmelidir.
- Kolay yorumlanabilir sonuçlar üretebilmeli ve işlevsel olmalıdır.

2.3 Kümeleme Analizi Veri Türleri

Kümeleme analizinde veri yapısı matris formundadır. Kümeleme işleminde kullanılan matrisler iki temel gruba ayrılır [2]:

- Veri Matrisi
- Farklılık Matrisi

2.3.1 Aralık Ölçekli Değişkenler

Aralık ölçekli değişkenler doğrusal bir ölçek üzerinde temsil edilebilen ölçeklerdir. En sık kullanılan aralık ölçekli değişkenler boy, ağırlık, genişlik, uzunluk ve hava sıcaklığı verileridir. Dikkat edilmesi gereken nokta işlem öncesi verilerin standartlaştırılmasıdır. Bir niteliği tanımlayan tüm ölçüm değerleri aynı tür ölçüm birimi ile temsil edilmelidir.

Aralık ölçekli veriler için uzaklık ya da komşuluk mesafesi hesaplamada üç çeşit uzaklık formülü kullanılır:

i. Öklid Uzaklığı:

En sık kullanılan yöntemdir. İki ya da daha çok boyutlu düzlemde kolaylıkla kullanılabilir ve

$$d(i,j)=\sqrt{\left(x_{i1}-x_{j1}\right)^2+\left(x_{i2}-x_{j2}\right)^2+\ldots+\left(x_{ip}-x_{jp}\right)^2} \tag{2.1}$$

ifadesi ile verilmektedir. Burada i ve j ifadeleri p boyutlu veri nesnelerini temsil etmektedir.

ii. Manhattan Uzaklığı

p boyutlu uzayda herhangi iki noktanın karşılıklı her bir koordinat değerinin farkı alınarak bulunur ve bu ifade

$$d(i,j)=\left|x_{i1}-x_{j1}\right|+\left|x_{i2}-x_{j2}\right|+\ldots+\left|x_{ip}-x_{jp}\right| \tag{2.2}$$

olarak verilmektedir.

iii. Minkowski Uzaklığı

Öklid ve Manhattan uzaklığının genelleştirilmiş hali olarak,

$$d(i,j)=(\left|x_{i1}-x_{j1}\right|^{q}+\left|x_{i2}-x_{j2}\right|^{q}+\ldots+\left|x_{ip}-x_{jp}\right|^{q})^{1/q} \quad (2.3)$$

şeklinde ifade edilir. q bir pozitif tam sayı olmak üzere bu ifade $q=1$ için Manhattan uzaklığını, $q=2$ için Öklid uzaklığını belirtir. q değişkeninin değeri artırıldıkça daha hassas uzaklık ölçüm ifadeleri elde edilir [2].

2.3.2 İkili Değişkenler

İkili değişkenler yalnızca 'var' ya da 'yok', diğer bir deyişle '0' ya da '1' değerini alabilen değişkenlerdir. Bu tür değişkenler tanımladıkları niteliğin ne kadar olduğunu değil, olup olmadığını belirtirler.

İkili değişkenler içeren kayıtlar arasındaki uzaklık hesabı için Şekil 2. 'deki bir tablo geliştirilmiştir.

		Nesne j		
		1	0	*toplam*
Nesne i	1	a	b	$a+b$
	0	c	d	$c+d$
	toplam	$a+c$	$b+d$	p

Şekil 2.2 : İkili değişkenler arası uzaklık hesabı tablosu

Bu tabloyu oluşturmak için ikili değişkenler içeren i ve j nesneleri seçilir. Her iki nesnede de aynı anda 1 değerini almış olan özelliklerin sayısı a, nesne i'de 1 ve nesne j'de 0 değerini almış olan değişkenlerin sayısı b şeklinde devam ederek a,b,c,d sayıları bulunur. Bu sayılar kullanılarak aşağıda verilen formül ile i ve j nesneleri arası uzaklık hesaplanır [2].

$$d(i,j)=\frac{b+c}{a+b+c+d} \quad (2.4)$$

2.3.3 Nominal, Ordinal ve Oran Değişkenleri

2.3.3.1 Nominal Değişkenler

Nominal değişkenler ikili değişkenlerin genelleştirilmiş hali olarak ifade edilebilir. İkili değişkenler yalnızca iki farklı değer alabilmelerine karşın Nominal değişkenler ikiden fazla, fakat sonlu sayıda değer alabilen değişkenlerdir. Bu tür değişkenler arasında uzaklık hesabı için;

$$d(i, j) = \frac{p - m}{p} \tag{2.5}$$

formülü kullanılır. Formülde m değişkeni i ve j değişkenlerinde aynı anda aynı değeri almış olan özellik sayısı, p değişkeni ise i ve j nesnelerinin sahip olduğu toplan özellik sayısını belirtir [2].

2.3.3.2 Ordinal Değişkenler

Bu değişkenler de Nominal değişkenlerde olduğu gibi sonlu sayıda farklı durum içerirler fakat Ordinal değişkenler anlamlı bir sıralama takip ederler. Sıralamada daha üstte olan değişken bir altta olan değişkenden daha değerlidir.

Ordinal değişkenler arası uzaklık tespiti için farklı yöntemler geliştirilmiş olsa da en kolay kullanılabilecek yöntem Ordinal değişkenin alabileceği değerleri [0,1] aralığında sayı değerler alabilecek şekilde standartlaştırıp aralık ölçekli değişkenlerde kullanılan mesafe yöntemlerini kullanmaktır [2].

2.3.3.3 Oran Ölçekli Değişkenler

Oran ölçekli değişkenler doğrusal olmayan ölçek üzerinde yapılan ölçümlerin sonuçlarıdır. En bilinen oran ölçekli değişkenler bakteri popülasyonlarının büyüme grafiği ve bir radyoaktif elementin yarı ömrünün ölçüm sonuçlarıdır. Oran ölçekli değişkenlerin genel yapısı aşağıdaki gibidir:

$$Ae^{Bt} \text{ ya da } Ae^{-Bt} \tag{2.6}$$

Bu ifadelerde A ve B pozitif sabitlerdir [2].

2.3.4 Karışık Tür Değişkenler

Karışık tür değişkenler şimdiye kadar açıklanan veri türlerinden iki ya da daha fazlasını içeren değişkenlerdir. Karışık tür değişkenlerin hesaplanmasında iki temel yaklaşım bulunmaktadır [2].

1. Tüm değişkenleri türlerine göre gruplandırıp, her gruba kendi içinde işlem yapılabilir. Bu yaklaşım karmaşık olduğu kadar yoğun işlem gücü de gerektirdiği için tercih edilmemektedir.

2. Bütün veri türleri için genel bir uzaklık hesaplama yöntemi kullanmaktır.

2.4 Kümeleme Yöntemleri

Kümeleme metodu seçimi kullanılacak veri türüne ve uygulamanın amacına göre farklılık gösterir. Kümeleme yöntemleri arasında benzerlik fazladır. Bu nedenle bilimsel literatürde en çok kabul gören yöntemler bu bölümde açıklanmıştır [2].

2.4.1 Bölümleme Yöntemleri

Bölümleme yöntemleri, n adet nesneden oluşan veri tabanını giriş parametresi olarak belirlenen k adet bölüme ($k \leq n$) ayırma temeline dayanır. Veri tabanındaki her bir eleman farklılık fonksiyonuna göre k adet bölümden birine dâhil edilir. Bu bölümlerden her biri bir küme olarak adlandırılır.

Bölümleme yöntemleri k sayısı doğru tahmin edilebilirse benzer şekilli dışbükey kümeleri bulmakta oldukça başarılı sonuçlar vermektedir. Eğer k sayısı hakkında önceden bir fikir belirlenemezse algoritmayı farklı k değerleri için tekrar tekrar uygulayarak en uygun k değeri bulunabilir.

Bölümleme yöntemleri k-means, k-medoids ve CLARA-CLARANS olarak bilinen algoritmaları kullanır.

k-means algoritması verilen nesneleri nitelik veya özelliklerine göre k adet sınıfa ayırmak amacıyla kullanılmıştır. Sınıflandırma, verilerin en yakın veya en benzer oldukları küme merkezleri etrafına yerleştirilmesi ile gerçekleştirilir [2]. k-means algoritması üçüncü bölümde ayrıntılarıyla açıklanmıştır.

2.4.1.1 K-medoids Algoritması

k-medoids algoritması k-means algoritmasının gürültü ve istisna verilere aşırı duyarlılığını gidermek amacıyla Kaufman ve Rousseeuw tarafından 1987 yılında geliştirilmiştir.

k-medoids algoritması kümeyi temsil edecek noktayı bulmak için küme elemanlarının ortalamasını almak yerine kümenin en merkez noktasındaki elemanı yeni küme merkezi olarak alır. Böylece istisna verilerin küme merkezini kenarlara doğru kaydırması problemi giderilmiş olur.

k-medoids algoritmasının birçok farklı türevi bulunmaktadır. PAM (Partitioning Around Medoids) ilk ortaya atılan k-medoids algoritmasıdır. PAM, öncelikle k-means algoritmasında olduğu gibi rastgele seçtiği k adet sayıyı küme merkezi olarak alır. Kümeye her yeni eleman katıldığında kümenin elemanlarını deneyerek kümenin gelişmesine en fazla katkıda bulunabilecek noktayı tespit edince bulduğu noktayı yeni merkez, eski merkezi ise sıradan küme elemanı olacak şekilde yer değiştirme işlemi yapar.

PAM küçük veri tabanlarında çok iyi sonuçlar vermesine rağmen hesaplanabilir karmaşıklığı yüksek olduğu için çok eleman içeren veri tabanlarında zayıf performans gösterir [2].

2.4.1.2 CLARA ve CLARANS Algoritmaları

PAM, k-medoids algoritmalarının başarısını kanıtlamasına rağmen büyük veri tabanlarında başarılı olamayınca Kaufman ve Rousseeuw tarafından 1990 yılında CLARA ortaya atılmıştır.

CLARA, veri tabanının tümünü almak yerine küçük bir örneklem kümesini temsilci olarak alıp örneklem üzerinde PAM algoritmasını uygular. CLARA'nın avantajı PAM'dan daha büyük veri yığınlarına uygulanabilmesi, dezavantajı ise performansının örneklemin boyuna göre değişmesi ve örneklem seçimi yeterince bağımsız değilse seçilen örneklem veri tabanını yeterince temsil edemeyeceği için yanlış sonuçlara ulaşmasıdır.

CLARANS algoritması ise CLARA'nın sonuçlarını örneklem seçimine bağlı olmaktan kurtarmak amacı ile 1994 yılında bilim dünyasına sunulmuştur.

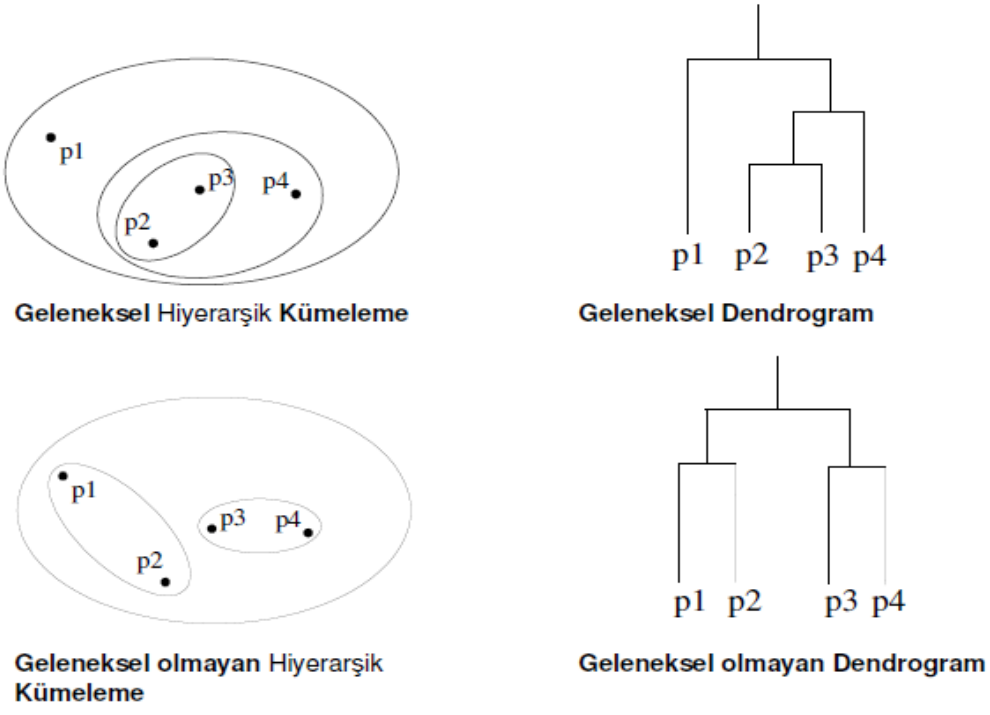
CLARANS örneklem seçimindeki ön yargıyı gidermek için sabit bir örneklem yerine her aşamada değişen örneklem kavramını ortaya atmıştır. Rastgele seçilen noktalar çevresi dikkate alınarak örneklem oluşturulur [2].

2.4.2 Hiyerarşik Yöntemler

Hiyerarşik yöntemler nesneleri Dendrogram denilen ağaç yapısı şeklinde gruplandırma temeline dayanır. Yapının inşa edilme yönüne göre yöntemler iki bölümde incelenir:

- Birleştirici kümeleme
- Ayrıştırıcı kümeleme

Hiyerarşik yöntemler k değerine ihtiyaç duymazlar fakat ağaç yapısı oluşturma işleminin ne zaman durdurulacağını belirten eşik değeri parametresine ihtiyaç duyarlar [2].



Şekil 2.3: Hiyerarşik Kümeleme[10]

Hiyerarşik yöntemlerin kullanıldığı algoritmalar ise aşağıda sıralanmıştır.

1. Birleştirici ve ayrıştırıcı algoritmalar
   1.1. Birleştirici kümeleme AGNES
   1.2. Ayrıştırıcı kümeleme DIANA

2. BIRCH
3. CURE
4. CHAMELEON

2.4.3 Yoğunluk Tabanlı Yöntemler

Yoğunluk tabanlı yöntemler, nesnelerin doğal dağılımını bir yoğunluk fonksiyonu aracılığı ile tespit ederek bir eşik yoğunluğunu aşan bölgeleri küme olarak adlandırırlar. Düzgün şekilli olmayan kümeleri bulma başarısı, gürültü ve istisnalardan etkilenmeme ve tek tarama ile sonuca ulaşma avantajları ile en başarılı kümeleme yöntemleri arasındadır [2].

2.4.4 Izgara Tabanlı Yöntemler

Veri uzayını incelemek için sonlu sayıda kare şeklinde hücrelerden oluşan ızgara yapıları kullanırlar. Kullandıkları ızgara yapısından ötürü veri tabanındaki nesne sayısından bağımsızdırlar [2].

2.4.5 Model Tabanlı Yöntemler

Eldeki verileri bir matematiksel model ile ifade etmeye çalışırlar. Model tabanlı yöntemler iki temel yaklaşımı kullanırlar; istatistik yaklaşım ve yapay zekâ yaklaşımı [2].

3. K – MEANS ALGORİTMASI

3.1 Tanım ve Tarihçe

En eski kümeleme yöntemlerinden biri olan k-means 1957 yılında ilk kez Hugo Steinhaus'un öne sürdüğü bir fikir olmasına rağmen 1967 yılında J.B. MacQueen tarafından geliştirilmiştir [5]. k-means algoritmasının genel mantığı n adet veri nesnesinden oluşan bir veri setini, giriş parametresi olarak verilen k adet kümeye bölümlemektir. Amaç, gerçekleştirilen bölümleme işlemi sonunda elde edilen kümelerin, küme içi benzerliklerinin maksimum ve kümeler arası benzerliklerinin minimum olmasını sağlamaktır. Küme benzerliği, kümenin ağırlık merkezi olarak kabul edilen bir nesne ile kümedeki diğer nesneler arasındaki uzaklıkların ortalama değeri ile ölçülmektedir [6-7].

En yaygın kullanılan gözetimsiz öğrenme yöntemlerinden birisi olan k-means' in atama mekanizması, her verinin sadece bir kümeye ait olabilmesine izin verir. Bu nedenle, keskin bir kümeleme algoritmasıdır. Merkez noktanın kümeyi temsil etmesi ana fikrine dayalı bir yöntemdir [6]. Eşit büyüklükte küresel kümeleri bulmaya eğilimlidir [8].

Algoritmaya k-means adı verilmesinin nedeni, algoritmanın çalışmasından önce sabit bir küme sayısına ihtiyaç duyulmasıdır. Küme sayısı k ile gösterilir ve elemanlarının birbirlerine olan yakınlıklarına göre oluşacak grup sayısını ifade eder. Buna göre k önceden bilinen ve kümeleme işlemi bitene kadar değeri değişmeyen sabit bir pozitif tam sayıdır [2].

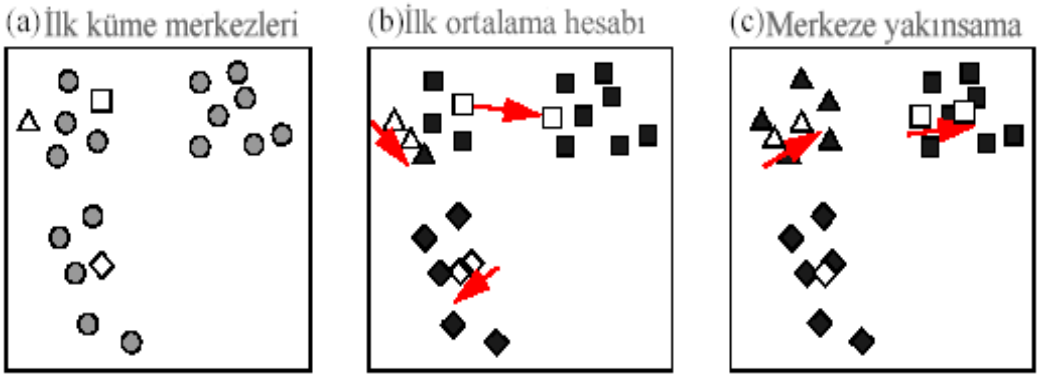
Bazı kümeleme algoritmaları bazı verilerde daha iyi sonuçlar vermesine rağmen k-means kümeleme algoritması her çeşit veride kabul edilebilir sonuçlar verir. Algoritmanın en büyük dezavantajı yerel optimumlarda kalarak genel optimumlara ulaşamamasıdır [1].

Çok yaygın kullanımı olan bu algoritmanın aşağıda belirtildiği gibi birtakım zayıf yanları da bulunmaktadır [2]:

- Algoritmanın başlangıcında giriş parametresi olarak bir k sayısına ihtiyaç vardır. Elde edilecek olan sonuçlar k sayısına göre değişkenlik gösterebilir. Eğer küme sayısı belirli değil ise deneme yoluyla en uygun sayı bulunur.
- Aşırı gürültü ve istisna veriler algoritmayla hesaplanan ortalamayı değiştirdiği için k-means algoritması gürültü ve istisnaya karşı çok duyarlıdır. Algoritma uygulanmadan önce veriler gürültü veya istisnadan temizlenebilir.
- Çakışan kümelerde iyi sonuç vermez.
- Her eleman aynı anda verilen bir kümenin içindedir veya dışındadır.
- k-means algoritması sadece sayısal veriler ile kullanılabilmektedir. Kategorik verilerin kümelenmesi için k-means algoritması bir çözüm sunmaz [7].

Açıklamada kolaylık olması açısından, bu bölümde algoritma iki boyutlu diyagramlar kullanılarak örneklenmiştir. Ancak uygulamada çok boyutlu elemanlarla, diğer bir deyişle çok eleman vektörü ile çalışılabilmektedir. Boyut sayısının artması algoritmada değişiklik yapılmasına neden olmamaktadır [9].

k-means algoritmasının çalışma şekline bir örnek şekil 3.1'de görülmektedir. Bu örnekte $k = 3$ olarak seçilmiş ve beyaz simgelerle şekil 3.1 (a)'da rastgele seçilen küme merkezlerini temsil etmektedir. Şekil 3.1 (b)'de geri kalan noktalar (bu kez aynı simgelerin siyah olanları) aynı şekilli ve beyaz renk olan küme merkezlerine dahil edilerek ilk kümeler oluşturulur. Bu işlem sonunda küme merkezleri her kümedeki elemanların ortalaması dikkate alınarak tekrar hesaplanır. Değişen küme merkezleri Şekil 3.1 (b)'de oklar ile gösterilmiştir. Sekil 3.1 (c)'de aynı işlem tekrar edildiğinde küme merkezlerinin değişimi görülmektedir. Bu şekilde başlangıç durumunda rastgele seçilen küme merkezleri, sürekli yinelemeler ile gerçek kümelenme alanlarının ortasına doğru yaklaşır. Bu işleme merkeze yakınsama denir. Merkeze yakınsama minimum seviyeye geldiğinde veya durduğunda kümeleme işlemi sona erer [2].



Şekil 3.1 : *k* -means kümeleme algoritması

3.2 k-means Algoritmasının Adımları

Algoritmanın ilk adımında öncelikle küme merkezlerini veya diğer bir deyişle küme ortalamasını temsil edecek *k* adet eleman belirlenir. MacQueen algoritmasında küme merkezleri ilk *k* adet elemandan seçilir. Ancak elemanların değerleri birbirine çok yakınsa seçim rastgele yapılabilir, birbirinden uzak elemanlar seçilebilir ya da bu seçim için belirli yöntemler kullanılabilir. Belirlenen bu elemanlar tek elemanlı başlangıç kümeleridir ve ilk küme merkezlerini oluştururlar. Kümenin ağırlıklı ortalama değerine sahip olan ya da bu değere en yakın olan elemanı küme merkezi olarak adlandırılır.

İkinci adımda, okunan elemanlar kendilerine en yakın *k* adet küme merkezinden birine dâhil edilir. Elemanların küme merkezine olan yakınlık derecesini bulmak amacıyla çeşitli geometrik yöntemler kullanılır. Bunlardan bir tanesi, kümeler arasındaki sınırları belirleyerek, elemanların hangi küme merkezine daha yakın olduklarını tespit eder. Bunun için önce iki küme merkezi bir doğruyla birleştirilir. Bu doğrunun orta noktasından geçen ve doğruyu dik kesen başka bir doğru daha geçirildiğinde, bu doğru iki kümenin sınırı olarak kabul edilir. Bulunan sınır çizgisi dikkate alınarak, elemanların hangi kümeye dâhil edileceği belirlenir.

Noktalar arası uzaklığın hesaplanmasında en çok kullanılan yöntem Öklid bağıntısıdır. İki boyutlu bilgilerde, iki küme merkezinin birleştirilmesinde doğru kullanılırken, boyut sayısı arttığında doğru yerine düzlem kullanılır. Çok boyutlu bilgilerde çok boyutlu düzlemler kullanılır. Algoritmanın geometrik gösteriminde küme sınırları yukarıda anlatıldığı şekilde belirlenmektedir. Bilgisayar programları

ile geliştirilen k-means algoritmalarında ise, düzlemler yerine noktalar arasındaki uzaklıklar hesaplanarak, noktaların merkeze yakınlığı dikkate alınmaktadır [9].

Üçüncü adımda her bir kümeye eklenen yeni eleman ile küme elemanlarının ağırlıklı ortalaması tekrar hesaplanarak yeni bir küme merkezi bulunur. Ağırlıklı ortalama kümenin her bir boyutundaki bütün elemanların ortalama değerlerinin alınması ile hesaplanır. Algoritmanın başında seçilen elemanlar küme merkezini oluştururken, ikinci döngü sonucunda bulunan yeni küme merkezleri artık bir küme elemanı değil, sadece bir ortalama değerdir. Bundan sonraki seçim işlemlerinde küme merkezini bu yeni eleman temsil eder. Her bir döngüde elemanlar farklı bir kümeye dâhil edilebilirler.

Kümeleme işlemi, tüm elemanların tekrar aynı veya farklı bir kümeye dâhil edilmesiyle devam eder. Elemanların bir kümeye dâhil edilmesi ve küme merkezlerinin tekrar hesaplanması işlemlerine ait döngü, küme sınırlarının değişimi bitene kadar devam eder. k-means algoritması ile uygulamalarda genellikle birkaç düzine döngü sonrası kararlı bir küme grubu ortaya çıkar.

k-means algoritması bilgisayar programına uygulanırken aşağıdaki adımlar izlenir [2]:

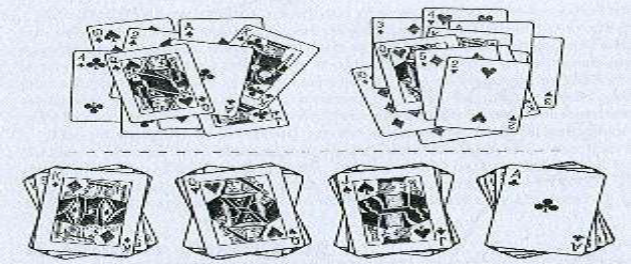
1. Küme sayısı (k) okunur. Bu değer algoritmaya dışarıdan verilir.
2. k adet rastgele veya belirli bir yöntemle küme merkezleri belirlenir.
3. Tüm elemanların merkezlere olan uzaklıkları hesaplanır.
4. Elemanlar yakın oldukları merkezlere göre kümelenir.
5. Dördüncü adımda oluşan kümelerin ortalamaları hesaplanarak yeni küme merkezleri belirlenir.
6. Son bulunan küme merkezleri bir önceki küme merkezlerine eşit oluncaya ya da belirlenen döngü sayısına ulaşılana kadar işlem tekrarlanır.

Algoritmanın akış diyagramı Şekil 3.2'de görülmektedir.

Şekil 3.2 : k -means algoritmasının adımları

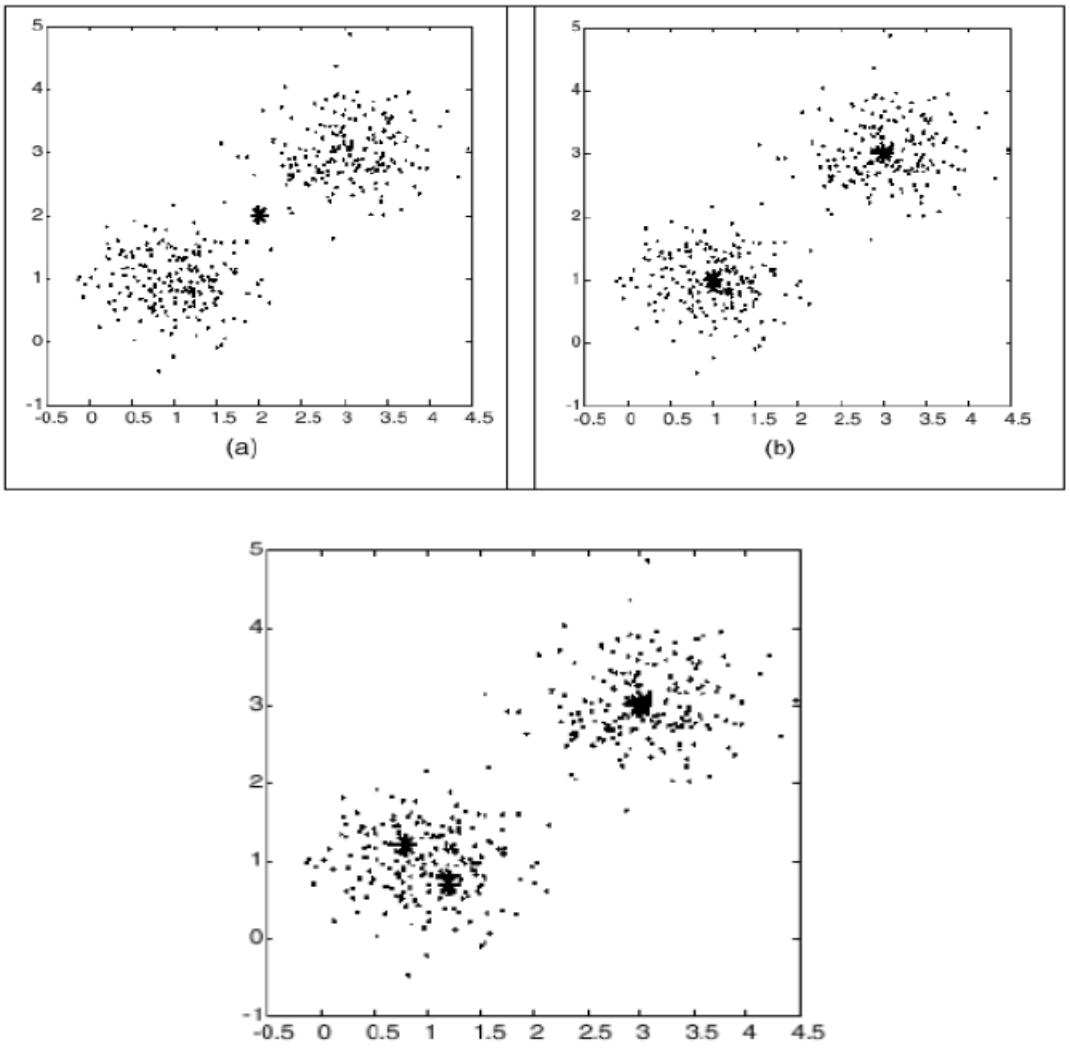
3.3 k Sayısının Kümelemeye Etkisi

Kümeleme algoritmalarında, elemanların birbirlerine olan yakınlıklarına göre oluşturulan kümelerin sayısı k ile gösterilir. k, işlem öncesinde bilinen ve kümeleme işlemi bitene kadar değeri değişmeyen sabit bir pozitif tamsayıdır. Sekil 3.3'te bir deste oyun kağıdının $k=2$ ve $k=4$ için kümeleme sonuçları gösterilmektedir. Şekilden görüleceği gibi k'nın farklı değerler alması, her biri geçerli olan çok farklı kümeler oluşturmaktadır. Hangisinin daha etkili olduğu, hangi kümelemenin kullanılacağına bağlıdır.



Şekil 3.3 : Oyun kağıtlarının $k=2$ ve $k=4$ için kümelenmesi

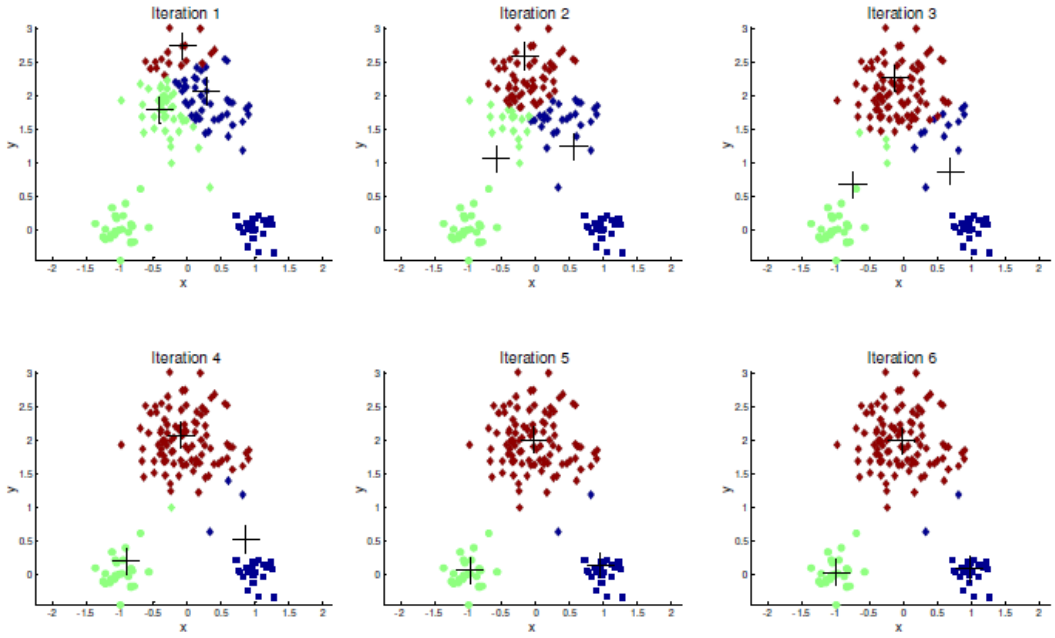
k-means ve benzeri kümeleme algoritmaları k sayısının belirlenmesi konusunda bir çözüm sunmazlar. Ancak birçok durumda, özel bir k değerinin belirlenmesi gerekli olmaz. Analiz aşamasında k değerinin tespiti için ön çalışma yapılır. Tahmini bir değer kullanılarak kümeleme algoritması çalıştırılır ve alınan sonuçlar değerlendirilir. Değerlendirme sonucunda beklenen kümeleme görülmez ise, başka bir k değeri kullanılarak tekrar kümeleme algoritması çalıştırılır veya veriler üzerinde değişiklik yapılabilir. Algoritmanın her çalıştırılması sonrasında, ortaya çıkan kümelerin etkinliğini hesaplamak için, küme içindeki kayıtların arasındaki ortalama uzaklık ile kümeler arası ortalama uzaklık karşılaştırılır. Hesaplamada başka yöntemler de kullanılabilir. Bu yöntemler algoritmaya dâhil edilebilir. Ancak ele alınan uygulama açısından sonucun yararlılığının belirlenmesi için kümeler mutlaka daha öznel temelde değerlendirilmelidir.



Şekil 3.4 : Küme sayısına göre k-means algoritmasının sonuçları

Aynı veri grubuna ait farklı k değerleri ile k-means algoritması uygulandığında ortaya çıkan sonuç Şekil 3.4'te yer almaktadır. Şekil 3.4 (a)'da $k=1$ için bütün elemanlar tek bir küme oluşturmuştur. Daha gerçekçi bir kümeleme, Şekil 3.4 (b)'de $k=2$ için ortaya çıkmıştır. Şekil 3.4 (c)'de $k=3$ için birbirine daha yakın elemanların yer aldığı grupta üçüncü bir küme oluşmuştur [2].

Benzer şekilde aşağıdaki şekilde de bir başka örnek üzerinden bu kez, ilk küme merkezlerinin seçilmesinin k-means algoritmasının uygulamasındaki önemi gösterilmektedir [10].



Şekil 3.5 : İlk başta seçilecek küme merkezlerinin k-means algoritmasındaki önemi

k-means algoritması ile verileri kümelere ayırmak için geometrik veya aritmetik hesaplama yöntemleri kullanılmaktadır. Bu yöntemler aşağıda örneklerle tanıtılmıştır.

3.3.1 Geometrik Hesaplama

Bu yöntemde, kümelenecek veriler koordinat sisteminde birer nokta olarak ele alınır. Noktalar arası uzaklığın hesaplanmasında en çok kullanılan yöntem Öklid bağıntısıdır. Küme sınırlarının belirlenmesi için iki boyutlu sistemde doğru, boyut sayısı arttığında ise doğru yerine düzlem kullanılır. Çok boyutlu bilgilerde çok boyutlu düzlemler kullanılır [2].

3.3.2 Aritmetik Hesaplama

Bilgisayar programları ile geliştirilen k-means algoritmalarında aritmetik hesaplama yöntemlerine başvurulmaktadır. Aritmetik hesaplamada, doğrular veya düzlemler yerine eleman değerleri arasındaki uzaklıklar hesaplanarak, elemanların merkezlere yakınlığı bulunmaktadır [2].

4. UYGULAMA: ÖĞRENCİ HARF NOTLARININ BELİRLENMESİ

Bu bölümde geçmiş yıllarda bir dersi alan tüm öğrencilerin yılsonu başarı notlarından k-means algoritması yardımıyla harf notları belirlenmiştir. Dördüncü bölümde ayrıntıları ile anlatılan k-means algoritmasının tüm adımları incelene probleme uygun bir örnek üzerinde aşağıda açıklanmıştır.

İlk adım olarak öğrenci notlarının

$$notlar = \{82.8, 81.85, 81.76, 79.07, 77.6, 77.46, 77.13, 75.92, 72.58, 71.28, 69.53, 67.46, 66.96, 65.4, 64.91, 62.78, 61.89, 61.83, 60.12, 59.17, 58.23, 57.38, 57.29, 57.08, 56.88, 55.31, 55.29, 53.75, 53.62, 53.48, 52.68, 51.98, 51.47, 50.46, 50.1, 49.6, 49.45, 48.85, 48.37, 48.25, 47.14, 46.24, 44.93, 43.98, 43.88, 43.74, 42.69, 42.17, 41.29, 39.99, 39.69, 39.16, 39.03, 38.27, 37.44, 37.44, 37.07, 36.38, 36.19, 36.18, 34.93, 34.33, 33.74, 33.74, 33.16, 33.14, 32.99, 32.43, 32.32, 31.96, 30.63, 30.18, 29.78, 29.55, 29.51, 29.48, 29.2, 28.76, 27.42, 25.51, 25.5, 25.18, 25.11, 24.93, 24.73, 24.47, 23.85, 23.73, 23.5, 23.4, 22.73, 22.51, 22.45, 22.11, 20.68, 18.94, 16.65, 14.75, 14.43, 14.2, 12.88, 12.39, 11.91, 7.7, 6.667, 3.933, 1.6, 1.575\}$$

olarak verildiği kabul edilmiştir.

Öncelikle not listesinin uzunluğu hesaplanmış ve ardından kümelenecek eleman sayısı olan $k = 8$ (AA, BA, BB, CB, CC, DC, DD, FF) 'e bölünmüştür. Bu bölüm sonucunda çıkan sayı tam sayıya yuvarlanmış ve notlar listesinde 7 adet 16'lık ve 1 adet 10'luk olmak üzere 8 küme oluşturulmuştur. Küme merkezlerinin başlangıç değerleri için oluşturulan her bir kümenin aritmetik ortalamaları alınmıştır.

1. BASLANGIC ORTALAMALARI = {10.5865, 23.4, 28.6662, 34.3538, 40.6746, 49.1938, 56.8792, 72.7282}

Bu ortalamalara -yani ilk küme merkezlerine- göre, notlar listesinin tüm elemanlarının 8 ayrı küme merkezine olan uzaklıkları hesaplanmış ve notlar listesinin her bir elemanı, uzaklığının minimum olduğu kümeye dâhil edilmiştir. Daha sonra oluşturulan her bir kümenin aritmetik ortalaması alınmış ve bu ortalamalar yeni küme merkezleri olarak atanmıştır. Bu adımların sonuçları aşağıda verilmiştir;

1. adim KUMELEME
FF = {1.575,1.6,3.933,6.667,7.7,11.91,12.39,12.88,14.2,14.43,14.75,16.65},
ort = 9.89042

DD = {18.94,20.68,22.11,22.45,22.51,22.73,23.4,23.5,23.73,23.85,24.47,24.73,
24.93,25.11,25.18,25.5,25.51}
ort = 23.49

DC = {27.42,28.76,29.2,29.48,29.51,29.55,29.78,30.18,30.63}
ort = 29.39

CC = {31.96,32.32,32.43,32.99,33.14,33.16,33.74,33.74,34.33,34.93,36.18,36.19,
36.38,37.07,37.44,37.44}
ort = 34.59

CB = {38.27,39.03,39.16,39.69,39.99,41.29,42.17,42.69,43.74,43.88,43.98,44.93}
ort = 41.5683

BB = {46.24,47.14,48.25,48.37,48.85,49.45,49.6,50.1,50.46,51.47,51.98,52.68}
ort = 49.5492

BA = {53.48,53.62,53.75,55.29,55.31,56.88,57.08,57.29,57.38,58.23,59.17,60.12,
61.83,61.89,62.78}
ort = 57.6067

AA = {64.91,65.4,66.96,67.46,69.53,71.28,72.58,75.92,77.13,77.46,77.6,79.07,
81.76,81.85,82.8}
ort = 74.114

Bu aşamada her kümenin içi boşaltılmış, her elemanın yeni küme merkezlerine olan uzaklıkları tekrar hesaplanarak, uzaklığının minimum olduğu kümeye dâhil edilmiş ve tekrar yeni küme merkezleri oluşturulmuştur. Bu işlem bir önceki ve bir sonraki küme sınıflarının her biri birbirine eşit olana kadar tekrarlanmıştır.

1. Başlangıç Ortalamaları'na göre uygulanmış program sonucunda 6. ve 7. adımın sonuçları aynı ve

FF = {1.575,1.6,3.933,6.667,7.7,11.91,12.39,12.88,14.2,14.43,14.75,16.65}

DD = {18.94,20.68,22.11,22.45,22.51,22.73,23.4,23.5,23.73,23.85,24.47,
24.73,24.93,25.11,25.18,25.5,25.51}

DC = {27.42,28.76,29.2,29.48,29.51,29.55,29.78,30.18,30.63,31.96}

CC = {32.32,32.43,32.99,33.14,33.16,33.74,33.74,34.33,34.93,36.18,36.19,
36.38,37.07,37.44,37.44,38.27}

CB = {39.03,39.16,39.69,39.99,41.29,42.17,42.69,43.74,43.88,43.98,44.93,
46.24}

BB = {47.14,48.25,48.37,48.85,49.45,49.6,50.1,50.46,51.47,51.98,52.68,
53.48,53.62,53.75,55.29,55.31}

BA = {56.88,57.08,57.29,57.38,58.23,59.17,60.12,61.83,61.89,62.78,
64.91,65.4,66.96,67.46}

AA = {69.53,71.28,72.58,75.92,77.13,77.46,77.6,79.07,81.76,81.85,82.8}

olarak elde edilmiştir. Burada bulunan standart sapma değeri ise 81.82855938519566 olarak hesaplanmıştır.

Ancak k-means algoritması başlangıç ortalamalarına bağlı olarak farklı sonuçlar verebilmektedir. Daha iyi sonuçların elde edilebilmesi için başlangıç ortalamalarına kullanıcının belirlediği bir aralık dâhilinde keyfi ekleme-çıkarma yapılmıştır. Bu işlem 10 kez tekrarlanarak birbirinden farklı 10 başlangıç ortalaması elde edilmiştir. Bu işlemler sonrasında hangi başlangıç ortalamasının daha iyi sonuç verdiğini görebilmek adına her bir adımda standart sapmalar hesaplanmış ve standart sapma değeri en küçük olan kümeleme en iyi sonuç kabul edilmiştir.

En iyi sonucun elde edildiği aşama için keyfi seçilen başlangıç koşulları

{7.65558,21.3168,23.3632,31.4932,42.0708,43.924,52.8628,74.1334}

olarak bulunmuştur. Bu başlangıç ortalamalarına göre yapılan iterasyonlar neticesinde en iyi sonuç ise

FF = {1.575,1.6,3.933,6.667,7.7}

DD = {11.91,12.39,12.88,14.2,14.43,14.75,16.65,18.94}

DC = {20.68,22.11,22.45,22.51,22.73,23.4,23.5,23.73,23.85,
24.47,24.73,24.93,25.11,25.18,25.5,25.51,27.42}

CC = {28.76,29.2,29.48,29.51,29.55,29.78,30.18,30.63,
31.96,32.32,32.43,32.99,33.14,33.16,33.74,33.74,34.33,34.93}

CB = {36.18,36.19,36.38,37.07,37.44,37.44,38.27,39.03,
39.16,39.69,39.99,41.29,42.17,42.69,43.74,43.88,43.98}

BB = {44.93,46.24,47.14,48.25,48.37,48.85,49.45,49.6,50.1,
50.46,51.47,51.98,52.68,53.48,53.62,53.75}

BA = {55.29,55.31,56.88,57.08,57.29,57.38,58.23,59.17,60.12,
61.83,61.89,62.78,64.91,65.4,66.96,67.46}

AA = {69.53,71.28,72.58,75.92,77.13,77.46,77.6,79.07,81.76,
81.85,82.8}

ve standart sapması 77.9190889540689 olarak elde edilmiştir.

Son olarak elde edilen bu sonuç ile Mathematica Programlama paketindeki kümeleme komutu, *FindClusters* , kullanılarak ikinci bir kümeleme sonucu olan

FF = {1.575, 1.6, 3.933, 6.667, 7.7, 11.91, 12.39, 12.88, 14.2, 14.43, 14.75, 16.65}

DD = {18.94, 20.68, 22.11, 22.45, 22.51, 22.73, 23.4, 23.5, 23.73, 23.85, 24.47,
24.73, 24.93, 25.11, 25.18, 25.5, 25.51, 27.42}

DC = {28.76, 29.2, 29.48, 29.51, 29.55, 29.78, 30.18, 30.63, 31.96, 32.32, 32.43,
32.99, 33.14, 33.16, 33.74, 33.74, 34.33}

CC = {34.93, 36.18, 36.19, 36.38, 37.07, 37.44, 37.44, 38.27, 39.03, 39.16,
39.69, 39.99}

CB = {41.29, 42.17, 42.69, 43.74, 43.88, 43.98, 44.93, 46.24, 47.14}

BB = {48.25, 48.37, 48.85, 49.45, 49.6, 50.1, 50.46, 51.47, 51.98, 52.68, 53.48,
53.62, 53.75, 55.29, 55.31}

BA = {56.88, 57.08, 57.29, 57.38, 58.23, 59.17, 60.12, 61.83, 61.89, 62.78, 64.91,
65.4, 66.96, 67.46}

AA = {69.53, 71.28, 72.58, 75.92, 77.13, 77.46, 77.6, 79.07, 81.76, 81.85, 82.8}

ve standart sapma değeri 80.7558 olan bir sonuç elde edilmiştir.

Mathematica paketinin *FindClusters* komutuyla elde edilen kümeleme ile bu tez kapsamında geliştirilen algoritmadan elde edilen kümeleme sonuçlarının standart sapmaları karşılaştırılmış ve geliştirilen algoritmanın sonucunun daha iyi olduğu görülmüştür.

Son olarak not verme işleminde gerçeğe uygun olarak FF notları için kullanıcının belirleyeceği bir üst sınır kabul edilerek, diğer notlar harf sistemine uygun olarak 7

kümeye ayrılmıştır. Geliştirilen algoritma notu 30 dan küçük olan öğrenciler FF kümesi içinde kalacak şekilde yukarıdaki örneğe tekrar uygulanarak yeni bir kümeleme elde edilmiştir. Bu kümelemeler içindeki en iyi sonuç

FF = {1.575, 1.6, 3.933, 6.667, 7.7, 11.91, 12.39, 12.88, 14.2, 14.43, 14.75, 16.65, 18.94, 20.68, 22.11, 22.45, 22.51, 22.73, 23.4, 23.5, 23.73, 23.85, 24.47, 24.73, 24.93, 25.11, 25.18, 25.5, 25.51, 27.42, 28.76, 29.2, 29.48, 29.51, 29.55, 29.78}

DD = {30.18, 30.63, 31.96,32.32, 32.43, 32.99, 33.14, 33.16, 33.74, 33.74, 34.33, 34.93}

DC = {36.18, 36.19, 36.38, 37.07, 37.44, 37.44, 38.27, 39.03,39.16, 39.69, 39.99}

CC = {41.29, 42.17, 42.69, 43.74, 43.88, 43.98, 44.93, 46.24}

CB = {47.14, 48.25, 48.37, 48.85, 49.45, 49.6, 50.1,50.46, 51.47, 51.98, 52.68,53.48, 53.62, 53.75}

BB = {55.29, 55.31, 56.88, 57.08, 57.29,57.38, 58.23, 59.17, 60.12,61.83, 61.89, 62.78}

BA = {64.91, 65.4, 66.96, 67.46, 69.53, 71.28,72.58}

AA = {75.92, 77.13, 77.46, 77.6, 79.07, 81.76, 81.85, 82.8}

ve standart sapma değeri ise 43.637742618602076 olarak elde edilmiştir. Ancak burada standart sapma değeri hesaplanırken FF kümesi işleme dâhil edilmemiştir.

KAYNAKLAR

[1] Yünel Y., 2010. K-means Kümeleme Algoritmasının Genetik Algoritma Kullanılarak Geliştirilmesi, p. 1,2.

[2] Dinçer E., 2006. Veri Madenciliğinde K-means Algoritması ve Tıp Alanında Uygulanması, p. 24-64.

[3] Makaroğlu B., 2010. Kümeleme(Cluster) Yöntemi ile Öğrencilerin Kavram Üretimi Sürecinin İncelenmesi, p. 54.

[4] Demiralay M., Çamurcu A.Y., 2005. Cure, Agnes ve K-means Algoritmalarındaki Kümeleme Yeteneklerinin Karşılaştırılması. p. 2,4.

[5] http://en.wikipedia.org/wiki/K-means_clustering

[6] Han J., Kamber M., 2001. Data Mining Concepts and Techniques, Morgan Kauffmann Publishers Inc.

[7] Berkhin P., 2002. Survey of Clustering Data Mining Techniques, San Jose, California, USA, Accrue Software Inc.

[8] Işık M., Çamurcu A.Y., 2007. K-means, K-medoids ve Bulanık C-means Algoritmalarının Uygulamalı Olarak Performanslarının Tespiti.

[9] Berry M., Linoff D., 2004. Data Mining Techniques, Wiley Publishing Inc.

[10] Takçı H., 2008. Sekizinci Ders Kümeleme Analizi: Temel Kavramlar ve Algoritmalar .

EKLER

K-means Algoritmasının Kodları

```
ortilk[list_,kumesay_] := Block[{el1,el2,el3,el4},
el1 = Length[list];
el2 = Sort[list];
el3 = Floor[el1 / kumesay];
Do[
el4[i] = Sum[el2[[ j]]],{ j,(i −1)el3 +1,iel3}] /
Max[{el3,10 ^ (−10)}],{i,1,kumesay −1}];
el4[kumesay] = Sum[el2[[ j]]],{ j,(kumesay −1)el3 +1,el1}] /
Max[{(el1 − (kumesay −1)el3),10 ^ (−10)}];
Return[Table[N[el4[i]],{i,1,kumesay}]]]

ayir[list_,ortlist_] := Block[{kumesay,el0,el1,el2,orts,
el3,el4,el5,el6,ortn},
kumesay = Length[ortlist];
el0 = Sort[list];
el1 = Length[el0];
Do[el2[i] = {},{i,1,kumesay}];
orts = ortlist;
Do[el3 = el0[[i]];
el4 = Abs[el3 − orts];
el5 = Min[el4];
el6 = Position[el4,el5][[1,1]];
AppendTo[el2[el6],el3]
,{i,1,el1}];
ortn = Table[N[Sum[el2[i][[ j]]],{ j,1,Length[el2[i]]}] /
Max[{Length[el2[i]]},10 ^ (−10)}],4],{i,1,kumesay}];
Return[Table[{el2[i],ortn[[i]]},{i,1,kumesay}]]]
```

```
sapma[list_,kont_] := Block[{el0,uz0,el1,el2,el3,sap,topsap},
el0 = list;
uz0 = Length[el0];
Do[el1[i] = If[kont == 1,el0[[i]][[1]],el0[[i]]];
el2[i] = Length[el1[i]];
el3[i] = Sum[el1[i][[j]],{j,1,el2[i]}] / Max[{el2[i],10^(-10)}];
sap[i] = Sqrt[Sum[(el1[i][[j]] - el3[i])^2,{j,1,el2[i]}]],{i,1,uz0}];
topsap = N[Sum[sap[i],{i,1,uz0}]];
Return[topsap]]

kumeleme[list_,kumesay_,ort_,maxit_] :=
Block[{el0,el1,sart,kumsap,kume,i,orts},
el0 = Sort[list];
el1 = Length[el0];
sart = True;
kume[0] = ayir[el0,ort];
i = 0;
While[sart,
kumsap[i] = sapma[kume[i],1];Print[StringForm["``. adim
KUMELEME = `` \n SAPMA = `",i,kume[i],kumsap[i]]];
orts[i] = Table[kume[i][[j,2]],{j,1,kumesay}];
kume[i+1] = ayir[el0,orts[i]];
sart = kume[i+1] ≠ kume[i] || i < maxit;
i = i+1];
Return[{kume[i],kumsap[i-1]}]]

RANDKUM[list_,kumesay_,randsay_,randara_,maxit_] :=
Block[{el0,el1,el2,el3,hata,minh,sonuc},
el0 = list;hata = 10^(10);
el1 = ortilk[el0,kumesay];
el2 = Join[{el1},Table[Table[Random[Real,{Max[{Min[el0],
el1[[i]] - randara}],Min[{Max[el0],el1[[i]] + randara}]}],
{i,1,kumesay}],{randsay}]];
Do[
Print[StringForm["``. BASLANGIC ORTALAMALARI =
`` \n",i,el2[[i]]]];
el3[i] = kumeleme[el0,kumesay,el2[[i]],maxit];
If[el3[i][[2]] < hata,hata = el3[i][[2]];minh = i;sonuc =
el3[i]],{i,1,randsay}];
Return[sonuc]]
```

```
NOTKUME[list_,Fsinir_,kumesay_,randsay_,randara_,maxit_]:=
Block[{el0,el1,el2,el3,el4},
el0=Sort[list];
el1=Length[el0];
el2={};el3={};
Do[
el4=el0[[i]];
If[el4>Fsinir,AppendTo[el2,el4],AppendTo[el3,el4]],{i,1,el1}];
Return[{el3,RANDKUM[el2,kumesay-1,randsay,randara,maxit]}]]
```