

Veri Kümeleme Algoritmalarının Performansları Üzerine

Karşılaştırmalı Bir Çalışma

Mustafa Seçkin Durmuş, Serdar İplikçi

Pamukkale Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü Kınıklı, Denizli
msdurmus@pau.edu.tr, iplikci@pau.edu.tr

Özet: Kümeleme, nesnelerin öğreticisiz olarak farklı gruplara (kümelere) ayrıldığı bir çeşit sınıflandırmadır. Aynı kümede bulunan nesnelere diğer kümelere bulunan nesnelere göre birbirlerine daha benzerdirler. Bu karşılaştırmalı çalışmada, farklı kümeleme algoritmalarının performansları incelenmiştir, incelenen bu algoritmaların ortak özelliği işlemler sonunda kaç küme oluşacağı ve hangi nesnenin hangi kümeye yerleştirileceği bilgilerinin önceden bilinmemesidir.

Anahtar Kelimeler: Veri Kümeleme, Kümeleme Algoritmaları, En Yakın Komşu, Karşılıklı Komşuluk, Minimum Örtün Ağaç, Destek Vektörleri.

A Comparative Study on Performances of Data Clustering Algorithms

Abstract: Clustering is a kind of classification which is unsupervised classification of objects (observations, features and data) into different groups (clusters). A cluster is a set of entities which are alike, and entities from different clusters which are not alike. In this comparative study, performances of different data clustering algorithms, in which resulting number of clusters are not known before clustering are considered.

Keywords: Data Clustering, Clustering Algorithms, Nearest Neighbor, Mutual Neighborhood, Minimum Spanning Tree, Support Vectors.

1. Giriş

Günümüzde, örnek analizi, makine öğrenmesi, örnek sınıflandırma ve veri madenciliği gibi çeşitli uygulama alanlarına sahip kümeleme işlemi farklı araştırma topluluklarına göre, (istatistikçilere göre sınıflandırma, pazarlamacılar için bölümlendirme, psikologlara göre sıralama) imlenmemiş verilerin gruplandırma metodlarının tanımlanması olarak bilinmektedir [1,2]. Bu uygulama alanlarından en önemlisi olan Veri Madenciliği ise büyük miktarda veri içerisinde, önceden bilinmeyen fakat potansiyel olarak kullanışlı bilginin bilgisayar programları kullanılarak aranması olarak tanımlanmaktadır [3]. Kümeleme işlemi gerçekleştirilmek amacıyla kullanılmakta olan birçok algoritma bulunmak-

tadır [1,2,6,10,12]. Kümeleme algoritmalarında amaç, elemanların birbirlerine çok benzediği, ancak özellikleri birbirlerinden çok farklı olan kümelerin bulunması ve veri tabanındaki kayıtların bu farklı kümelere bölünmesidir. Kısaca, aynı kümede bulunan veriler diğer kümelere bulunan verilere göre birbirlerine daha benzerdirler. Kümeleme algoritmalarından istenilenler ise uyarlanabilirlik (hem zaman hem de alan açısından), farklı veri tiplerine uygulanabilirlik, gürlüğe dayanıklılık, giriş değerlerinin sırasının önemsenmemesi ve hız olarak tanımlanabilir.

2. Veriler Arası İlişkiler

Örnek, kümeleme algoritmaları tarafından kullanılan veri ögeleridir ve genellikle yapılan

ölçümlerin sonuçlarını içermektedir. Örnek vektörünün her bir sayısal elemanı da (x_i) , yani verilerin yakınlık bileşenleri, öznitelik olarak tanımlanmaktadır (1). Buradaki d örnek uzayının boyutunu, n örnek sayısını ifade etmektedir. Örnek seti (2)'de görülmektedir.

$$x_i = [x_1 \quad x_2 \quad \dots \quad x_n]^T, (i = 1, 2, \dots, n) \quad (1)$$

$$X = [x_1 \quad x_2 \quad \dots \quad x_n]; X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix} \quad (2)$$

Örnek dizisi $n \times d$ örnek matrisi olarak da ifade edilebilmektedir. Bu matrisin her satırı örnekleri ve her sütunu da öznitelikleri veya ölçümleri ifade etmektedir. Veriler arasındaki ilişkiler, satır ve sütunları verilerden oluşan yakınlık matrisleri (3) ile ifade edilmektedir. Bu yakınlıklar, Minkowski ölçüleri olarak da bilinen, Öklit, Manhattan, Supremum, Hamming, Mahalanobis gibi yakınlık ölçüm yöntemleri ile hesaplanmaktadır [1,2]. Bu çalışmada kullanılan yakınlık matrisleri Öklit uzaklık ölçümüne (4) göre hesaplanmıştır.

$$[D(i, j)] = \begin{bmatrix} 0 & d_{12} & \dots & d_{1m} \\ d_{21} & 0 & \dots & d_{2m} \\ \dots & \dots & \dots & \dots \\ d_{n1} & d_{n2} & \dots & 0 \end{bmatrix} \quad (3)$$

$$d_{ik} = d(i, k) = \sqrt{(x_i - x_k)^T (x_i - x_k)} \quad (4)$$

3. Kümeleme İşleminde İzlenecek Yol

Klasik bir örnek kümeleme işleminde takip edilmesi gereken adımlar şunlardır:

1. Örneklerin sunulması,
2. Örneklerin uzaklık ölçümlerinin veri tabanına uygun olarak tanımlanması,
3. Kümeleme veya gruplama,
4. Veri ayıklama (gerekli olduğu durumlarda yapılmaktadır),
5. Çıktının değerlendirilmesi (gerekli olduğu durumlarda yapılmaktadır) [2,10].

Yukarıda bulunan ilk üç adım şekil 1'de görülmektedir. Geri besleme, kümeleme sonucunda elde edilen çıkışın, örnek uzaklık ölçümlerine ve özniteliklerin çıkarılmasına etki etmektedir.

4. Benzetimlerde Kullanılan Algoritmalar

4.1. En Yakın Komşu Algoritması:

1. Her nokta kendisine en yakın (nearest neighbour) kümeye yerleştirilir.
2. Eşik değeri (threshold - t), yeni bir komşuyu veya yeni bir kümeyi belirler.
3. Tüm noktalar herhangi bir kümeye yerleştirilinceye kadar işlemlere devam edilir [1,4].

4.2. Karşılıklı Komşuluk Değeri Algoritması:

1. Tüm noktalar için karşılıklı en yakın komşuluk değerleri (Mutual Neighbourhood Value - MNV) belirlenir.
2. Eşik değeri yerine en yakın komşu sayısı (k) belirlenir.
3. $MNV=2, 3, \dots, 2k$ için kümeler oluşturulur [1,2].

4.3. En Küçük Örtün Ağaç Algoritması:

1. İki nokta arasındaki uzaklıklar "ağırlık" olarak tanımlanır.
2. Olası ağaçlar arasından ağırlıklar toplamı en küçük ağaç seçilir.
3. Seçilen eşik değerinden büyük ağırlığa sahip dallar ağaçtan kaldırılır.

Bu algoritma için eşik değeri yerine uyuşmayan kenar (inconsistent edge) seçimi ile de kümeler belirlenir. Kendisine yakın olan ağırlıkların ortalamasından daha büyük ağırlığa sahip kenar "uyuşmayan kenar" olarak adlandırılır [5,1].

4.4. Delaunay Üçgen Metodu:

1. x_i ve x_j 'yi birleştiren kenar eğer x_i ve x_j 'yi de içeren Dirichlet mozağının (Şekil 2) iki hücresi ortak sınırı paylaşıyorsa oluşturulan çizgede birbirine bağlıdır. Uygulamaların çoğu sadece iki boyutlu veriler için yapılmıştır.
2. Sınır-Kenar ilişkileri göz önünde bulundurulur ve oluşturulacak olan çizge yapısı bu ilişkilere göre belirlenir [1].

Uygulamaların çoğu biyoloji ve coğrafya da-
lında gerçekleştirilmiştir.

4.5. Gabriel Çizgeleri:

x_i ve x_j noktaları dışında hiçbir nokta
DISK(x_i, x_j)'de bulunmuyorsa, x_i ve x_j noktala-
rı oluşturulan çizgede birbirine bağlıdır (Şekil
2). Yani; (5) şartı sağlandığı takdirde noktalar
oluşturulan çizgeye dahil edilir (tüm k değerle-
ri için $k \neq i, k \neq j$).

$$d^2(x_i, x_j) < d^2(x_i, x_k) + d^2(x_j, x_k) \quad (5)$$

DISK, $d(x_i, x_j)$ çaplı dairedir ve Gabriel Çiz-
gelerinin etki bölgesidir (Şekil 3). [1].

4.6. Bağlı Komşuluk Çizgesi:

x_i ve x_j noktaları dışında diğer hiçbir nokta
LUNE(x_i, x_j)'de bulunmuyorsa, x_i ve x_j noktala-
rı oluşturulan çizgede birbirine bağlıdır. Yani;
(6) şartı sağlandığı takdirde noktalar oluşturu-
lan çizgeye dahil edilir (tüm k değerleri için
 $k \neq i, k \neq j$).

$$d^2(x_i, x_j) \leq \max \{d^2(x_i, x_k), d^2(x_j, x_k)\} \quad (6)$$

LUNE, $d(x_i, x_j)$ yarıçaplı iki dairenin kesi-
şimidir ve Bağlı Komşuluk Çizgelerinin etki
bölgesidir (Şekil 4) [1].

4.7. Destek Vektörleri:

1. Veriler lineer ayrılabilir ise amaç; sınırı
maksimize eden düzlemin bulunmasıdır
(Optimal Separating Hyperplane).
2. Lineer olarak ayrılamayan verileri uygun
bir non-Linear dönüşüm kullanarak lineer
ayrılabilir hale getiren ve optimizasyon
tabanlı bir eğitim algoritması kullanarak
öğrenebilen sistemlerdir.
3. Verimizi çevreleyecek en küçük küreye
bakılır [7].

Amaç Fonksiyonu :

$$R^2 \quad (7)$$

Kısıtlamalar:

$$\|\Phi(x_i) - \mathbf{a}\|^2 \leq R^2 + \xi_i \quad \forall i; \xi_i \geq 0 \quad (8)$$

$$\|x\| = \sqrt{\langle x, x \rangle} = \sqrt{(x^T \cdot x)} = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \quad (9)$$

$\|\cdot\|$ Öklid Normu;
 \mathbf{a} Kürenin Merkezi;
 R Kürenin yarıçapı

Problemin çözümü için Lagrange ifadesi
yazılır (β ve μ Lagrange katsayılarını ifade
etmektedir).

Sırasıyla R, \mathbf{a} ve ξ 'a göre türevler;

$$L = R^2 - \sum_i (R^2 + \xi_i - \|\Phi(x_i) - \mathbf{a}\|^2) \beta_i - \sum \xi_i \mu_i + C \sum \xi_i \quad (10)$$

$$\begin{aligned} \frac{\partial L}{\partial R} = 0 &\Rightarrow 2R - \sum_{i=1}^l 2R\beta_i = 0 \\ &\Rightarrow 2R \left(1 - \sum_{i=1}^l \beta_i\right) = 0 \Rightarrow \sum_{i=1}^l \beta_i = 1 \end{aligned} \quad (11)$$

$$\frac{\partial L}{\partial \mathbf{a}} = 0 \Rightarrow \mathbf{a} = \sum_{i=1}^l \beta_i \Phi(x_i) \quad (12)$$

$$\frac{\partial L}{\partial \xi_j} = 0 \Rightarrow \beta_j = C - \mu_j \quad (13)$$

KKT tamamlayıcı koşulları;

$$\xi_j \mu_j = 0 \quad (14)$$

$$(R^2 + \xi_j - \|\Phi(x_j) - \mathbf{a}\|^2) \beta_j = 0 \quad (15)$$

$$0 \leq \beta_j \leq C; j = 1, 2, \dots, N$$

Buna göre denklemler tekrar düzenlenirse sırasıyla R , \mathbf{a} ve μ , yok edilebilir ve Lagrange ifadesi tekrar yazılabilir;

$$\mathbf{W} = \sum_j \Phi(\mathbf{x}_i)^2 \beta_j - \sum_{i,j} \beta_i \beta_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j) \quad (16)$$

Çekirdek (Kernel) fonksiyonu tanımından faydalanarak (16) denklemi tekrar yazılabilir;

$$\mathbf{W} = \sum_j \beta_j \mathbf{K}(\mathbf{x}_j, \mathbf{x}_j) - \sum_{i,j} \beta_i \beta_j \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \quad (17)$$

Çekirdek fonksiyonu öznilelik uzayında iç çarpım işlemini gerçekleştirdiğinden $\Phi(\mathbf{x}_i)$ 'in analitik formunun bilinmesine gerek yoktur. Burada gaussian kernel fonksiyonu kullanılmıştır.

$$\mathbf{K}_{ij} = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = e^{-q \|\mathbf{x}_i - \mathbf{x}_j\|^2} \quad (18)$$

buradaki q genişlik parametresidir. Veri kümelerini çevreleyen sınırlar bu parametreye ve (10) ifadesindeki C parametresine bağlıdır (benzetimlerde $C = 1$ olarak seçilmiştir). Tüm bu bilgilerin ışığında, her noktanın kürenin merkezine olan uzaklığı;

$$R^2 = \|\Phi(\mathbf{x}_i) - \mathbf{a}\|^2 \quad (19)$$

(18)'de \mathbf{a} 'nın değeri yerine yazılırsa;

$$\mathbf{a} = \sum_j \beta_j \Phi(\mathbf{x}_j) \quad (20)$$

$$R^2(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{x}) - 2 \sum_j \beta_j \mathbf{K}(\mathbf{x}_j, \mathbf{x}) + \sum_{i,j} \beta_i \beta_j \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) \quad (21)$$

(20) denklemi elde edilir. Veri uzayını çevreleyen sınırlar (21) ile verilen kümeye göre elde edilmektedir (Şekil 5).

$$\{\mathbf{x} \mid R(\mathbf{x}) = R\} \quad (22)$$

Kümelendirme algoritması farklı kümelere ait olan noktaları ayırt edememektedir. Bu ayırımı yapabilmek amacıyla (22) ifadesini de kapsayan bir yaklaşım kullanılmıştır. Bu yaklaşıma göre, "farklı kümelere ait veri çiftlerini birleştiren herhangi bir yol (path), öznilelik uzayında

kürenin dışında kalmalıdır" [7]. Bu nedenle, yol, $R(\mathbf{y}) > R$ gibi \mathbf{y} veri setinin bir bölümünü içermektedir. Bu tanımlamalar ışığında, öznilelik uzayındaki görüntüleri kürenin içinde veya üzerinde bulunan \mathbf{x}_i ve \mathbf{x}_j veri çiftleri arasında \mathbf{A}_{ij} komşuluk matrisi tanımlanmıştır [7].

$$\mathbf{A}_{ij} = \begin{cases} 1 & ; y \text{ değerleri } \mathbf{x}_i \text{ ve } \mathbf{x}_j \text{ çiftlerini birleştiren} \\ & \text{doğru parçası üzerinde ise} \\ 0 & ; \text{aksi halde} \end{cases} \quad (23)$$

5. Benzetimlerde Kullanılan Veri Setleri

5.1. Iris Veri Seti:

Üç farklı türde (Setosa, Versicolor, Virginica) Iris çiçeklerinin çanak yaprak uzunluğu, çanak yaprak genişliği, taç yaprak uzunluğu ve taç yaprak genişliği ölçümlerinden oluşan dört boyutlu bir veri setidir [8].

5.2. Avustralya Yengeçlerinden Oluşan Veri Seti:

Avustralya kaya yengeçlerinin (Leptograpsus), Ön lob uzunluğu, kabuk uzunluğu, kabuk genişliği, vücut derinliği ve arka genişliklerinin ölçümlerinden oluşan beş boyutlu veri setidir [9].

5.3. Rasgele Oluşturulan Veri Seti:

Bu iki değerlendirme (benchmark) veri seti dışında rasgele oluşturulmuş iki boyutlu bir veri setidir (Şekil 6).

Veri setleri benzetimlerden önce normalize edilmiştir. Ayrıca; oluşturulan bu veri setlerine uygulanan kümeleme algoritmalarının karşılaştırılmasında kullanılmak üzere veri setlerine belirli oranlarda gürültü (24) eklenmiştir.

$$\text{SNR} = 10 \log_{10} \left(\frac{\sigma_v^2}{\sigma_n^2} \right) \quad (24)$$

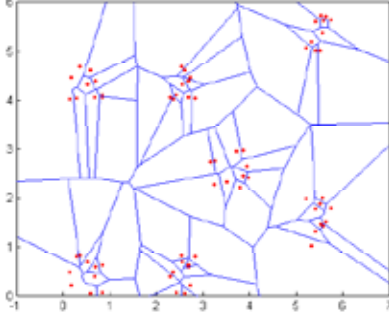
Bu ifadede, s_v^2 veri setlerinin bileşenlerinin,

s_h^2 ise eklenen gürültünün değişkesidir.

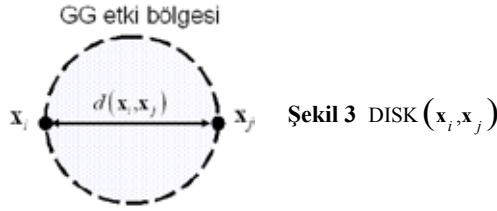
6. Tablo ve Şekiller



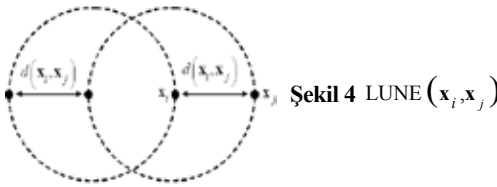
Şekil 1 Kümeleme İşleminin Adımları



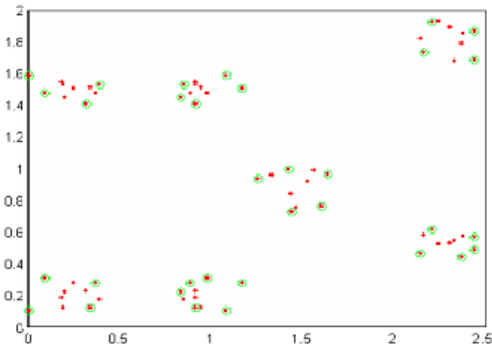
Şekil 2 Dirichlet Mozaigi (Voronoi Diyagramı)



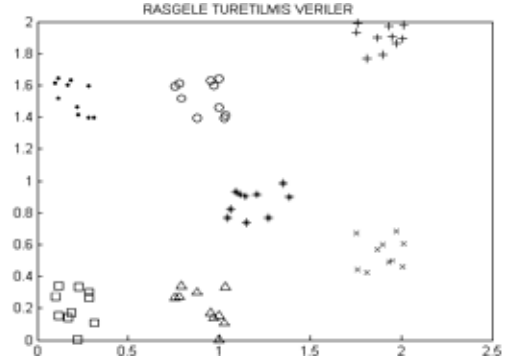
Şekil 3 DISK (x_i, x_j)



Şekil 4 LUNE (x_i, x_j)



Şekil 5 Rasgele Oluşturulmuş Verilerin Destek Vektörleri Yardımı ile Kümelenmesi



Şekil 6 Rasgele Oluşturulmuş Veriler

İris eşik değerleri	Gürültüsüz			Gürültülü	
	Süre (sn.)	Flops	Küme Sayısı	Küme Sayısı 45Db	Küme Sayısı 24Db
0.1	0.735	89867	1	1	1
0.05	0.735	89867	1	1	1
0.02	0.734	89868	2	2	1
0.015	0.733	89869	3	3	3
0.01	0.720	89875	9	9	7
0.008	0.719	89881	15	15	14
Yengeç eşik değerleri	Süre (sn.)	Flops	Küme Sayısı	Küme Sayısı	
				45Db	24Db
0.05	1.688	159817	1	1	1
0.02	1.688	159817	1	1	1
0.01	1.1689	159819	3	3	3
0.007	1.69	159821	5	4	6
0.006	1.7	159824	8	8	9
0.005	1.703	159830	14	14	15
Data eşik değerleri	Süre (sn.)	Flops	Küme Sayısı	Küme Sayısı	
				45Db	24Db
0.5	0.109	19547	1	1	1
0.1	0.109	19548	2	2	2
0.08	0.109	19549	3	3	2
0.065	0.109	19550	4	4	5
0.04	0.109	19553	7	7	7
0.02	0.11	19557	11	11	11

Tablo 1 En Yakın Komşu Algoritması Sonuçları

İris	Gürültüsüz			Gürültülü	
	Süre (sn.)	Flops	Küme Sayısı	45Db	24Db
en yakın komşu sayısı					
20	8.797	354038	14	10	10
30	18.64	361436	16	12	11
50	55.422	374634	16	12	11
70	107.219	387930	16	12	12
90	185.203	401199	16	12	12
120	357.875	543064	16	12	12
Yengeç					
en yakın komşu sayısı					
10	30.469	855295	7	7	9
20	47.031	870898	8	10	13
40	131.359	901365	10	10	13
60	259.328	933051	10	10	13
90	439.484	979642	10	10	13
130	754.797	1041699	10	10	13
Data					
en yakın komşu sayısı					
5	0.672	62195	4	4	3
10	1.344	65439	6	6	5
15	2.344	68437	6	6	5
20	3.766	71342	6	6	6
30	7.812	77207	7	6	6
50	20.563	88959	7	7	6

Tablo 2 Karşılıklı Komşuluk Değeri Algoritması Sonuçları

İris	Gürültüsüz			Gürültülü	
	Süre (sn.)	Flops	Küme Sayısı	45Db	24Db
eşik değerleri					
0.1	10.969	367879	1	1	1
0.05	5.64	360979	1	1	1
0.02	2.141	355836	3	3	4
0.015	1.437	355570	6	6	7
0.01	0.812	35772	29	25	30
0.008	0.672	362490	49	50	60
Yengeç					
eşik değerleri					
0.05	20.516	770996	1	1	1
0.02	7.578	754420	1	1	1
0.01	2.484	747614	2	2	3
0.007	1.485	751257	10	9	13

Data	Süre (sn.)	Flops	Küme Sayısı	Küme Sayısı	
				45Db	24Db
0.006	1.281	751213	24	23	20
0.005	1.141	753814	44	45	57
eşik değerleri					
0.5	1.562	52673	1	1	1
0.1	0.547	49665	1	1	1
0.08	0.391	49139	1	1	1
0.065	0.281	48490	2	2	2
0.04	0.219	48699	5	4	4
0.02	0.188	48451	7	7	8

Tablo 3 En Küçük Örtünme Ağaç Algoritması Sonuçları

İris	Gürültüsüz			Gürültülü	
	Süre (sn.)	Flops	Küme Sayısı	45Db	24Db
eşik değerleri					
0.1	14.89	1660445	1	1	1
0.05	7.875	1128141	1	1	1
0.02	2.734	797918	2	2	2
0.015	2.031	659366	3	3	3
0.01	1.328	557880	6	6	7
0.008	1	513515	8	8	11
Yengeç					
eşik değerleri					
0.05	25.969	3751299	1	1	1
0.02	11.079	2260337	1	1	1
0.01	5.313	1544031	2	2	2
0.007	3.532	1296177	2	2	2
0.006	2.984	1220427	2	2	2
0.005	2.5	1148111	2	2	2
Data					
eşik değerleri					
0.5	2.094	232125	1	1	1
0.1	0.532	129309	1	1	1
0.08	0.39	111453	1	1	1
0.065	0.281	95896	2	2	2
0.04	0.204	81867	3	3	3
0.02	0.141	77829	5	5	5

Tablo 4 Delaunay Üçgen Metodu Sonuçları

İris	Gürültüsüz			Gürültülü	
	Süre (sn.)	Flops	Küme Sayısı	Küme Sayısı	
eşik değerleri				45Db	24Db
0.1	10.875	1610816	1	1	1
0.05	5.594	1082920	1	1	1
0.02	2.328	679865	2	2	2
0.015	1.563	575622	4	4	5
0.01	0.875	475158	14	13	15
0.008	0.625	434425	18	18	23
Yengeç	Süre (sn.)	Flops	Küme Sayısı	Küme Sayısı	
				45Db	24Db
eşik değerleri					
0.05	24.75	3602350	1	1	1
0.02	9.375	2020084	1	1	1
0.01	3.5	1254490	2	2	2
0.007	2.016	1037340	2	2	3
0.006	1.609	971593	4	4	3
0.005	1.219	913042	7	7	9
Data	Süre (sn.)	Flops	Küme Sayısı	Küme Sayısı	
eşik değerleri				45Db	24Db
0.5	1.594	218424	1	1	1
0.1	0.5	112440	1	1	1
0.08	0.359	94080	1	1	1
0.065	0.234	79315	2	2	2
0.04	0.156	68020	5	5	3
0.02	0.125	61248	7	7	7

Tablo 5 Gabriel Çizgeleri Sonuçları

İris	Gürültüsüz			Gürültülü	
	Süre (sn.)	Flops	Küme Sayısı	Küme Sayısı	
eşik değerleri				45Db	24Db
0.1	10.719	1592424	1	1	1
0.05	5.5	1065896	1	1	1
0.02	2.032	639225	3	3	3
0.015	1.313	537739	6	6	7
0.01	0.625	434425	18	18	23
0.008	0.437	408551	35	34	41
Yengeç	Süre (sn.)	Flops	Küme Sayısı	Küme Sayısı	
				45Db	24Db
eşik değerleri					
0.05	23.375	3517510	1	1	1
0.02	8.203	1877270	1	1	1
0.01	2.484	1106424	2	2	2
0.007	1.219	913042	7	7	9
0.006	0.922	868583	10	9	13
0.005	0.687	829939	24	23	20

Data	Süre (sn.)	Flops	Küme Sayısı	Küme Sayısı	
				45Db	24Db
eşik değerleri					
0.5	1.593	218424	1	1	1
0.1	0.485	110856	1	1	1
0.08	0.344	92064	1	1	1
0.065	0.218	77299	2	2	2
0.04	0.156	67872	7	7	4
0.02	0.109	59808	7	7	7

Tablo 6 Bağlı Komşuluk Çizgesi Sonuçları

İris	Gürültüsüz			Gürültülü	
	Süre (sn.)	Flops (x 10 ⁹)	Küme Sayısı	45Db	24Db
q değerleri					
500	191.656	1.1799	1	1	1
750	285.718	1.2488	2	2	2
1500	278.609	1.3095	2	2	2
5000	300.812	1.3755	3	3	2
7000	315	1.483	4	5	6
10000	361.203	1.6023	7	8	8
Yengeç	Süre (sn.)	Flops (x 10 ⁹)	Küme Sayısı	Küme Sayısı	
				45Db	24Db
q değerleri					
750	67.672	3.2894	1	1	1
8000	124.406	3.2411	2	2	2
25000	540.766	2.7744	3	3	2
30000	712.562	2.6208	4	4	4
45000	1.152e+003	2.3451	16	14	15
50000	12784e+003	2.348	18	16	18
Data	Süre (sn.)	Flops (x 10 ⁹)	Küme Sayısı	Küme Sayısı	
q değerleri				45Db	24Db
500	4.047	53658278	1	1	1
750	38.313	75121638	2	2	2
1500	52.016	79146050	6	6	7
5000	52.204	73746657	7	7	7
7000	54.109	73715098	7	7	7
10000	54.422	71681937	7	7	7

Tablo 7 Destek Vektörleri Sonuçları

7. Sonuçlar

Kullanılan tüm veri setleri için en iyi kümelemeyi yapabilen bir algoritma bulunmamaktadır. Çünkü tüm kümeleme algoritmalarının performansları verilerin dağılımına bağlıdır, Iris veri seti için başarıyı yüksek (düşük flop sayısı ve işlem süresi, daha belirgin kümeler) kümeleme yapabilen bir algoritma diğer veri setleri için anlamlı kümeler oluşturamamaktadır. Bu nedenle amacımıza uygun bir kümeleme algoritması önceden belirlenmelidir. Bu belirleme işleminde, uzmanın önemi unutulmamalıdır. Bu çalışmada gerçekleştirilen algoritmalar arasında tüm özellikler göz önünde bulundurulduğunda, en yakın komşuluk algoritması en iyi algoritma olarak belirlenmiştir. Algoritmaların seçimi dışında kümeleme işlemlerine önemli oranda etki eden diğer bir önemli husus da uygun eşik değerlerinin belirlenmesidir. Halen üzerinde çalışılan bir konu olmakla birlikte, kümelenecek olan veri setinin yakınlık matrisinde bulunan en büyük, en küçük ve ortalama değerlere göre de eşik değeri belirlenebilmektedir veya bir noktanın diğer noktalara olan uzaklık değerleri arasında ortalama değerinin üstünde olan yakınlık değer(ler)i uyuşmayan kenar (inconsistent edge) olarak belirlenip kaldırılmaktadır. İç içe girmiş veriler için bulanık kümeleme kullanılarak daha iyi sonuçlar elde edilebilir. Günümüzde veri tabanlarının terabayt'lar cinsinden ifade edilmektedir. Mesela, uydular vasıtası ile alınan bir görüntüyü işlemek amacıyla hem hızlı, hem de verimli kümeleme algoritmalarına ihtiyaç duyulmaktadır. Bu gibi büyük boyutlu verileri kümeleyebilmek için bu amaca uygun hazırlanmış bilgisayarlar (paket programlar v.b.) ve algoritmalar kullanmak daha elverişli olacaktır.

8. Kaynaklar

[1] Jain A. K., Dubes R. C., *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[2] Jain A. K., Murty M. N., Flynn P. J., *Data Clustering: A Review*, ACM Computing Surveys, Vol. 31, No. 3, 1999.

[3] Mannila H., *Data mining: Machine Learning, Statistics, and Databases. Eight International Conference on Scientific and Statistical Database Management*, Stockholm, June 18-20, p. 1-8, 1996.

[4] İplikçi S. and Denizhan Y., *Kaotik Sistemler İçin Yapay Sinir Ağı Tabanlı Bir Hedef Bölgesine Götürme Yöntemi*, TOK'2002 Bildiriler Kitabı, s.281-291, Ankara, 2002.

[5] Zahn C. T., *Graph Theoretical Methods for Detecting and Describing Gestalt Clusters*, IEEE Trans. on Computers, SLAC-PUB-672, 1970.

[6] Hartigan J. A., *Clustering Algorithms*, John Wiley & Sons Inc., ISBN 0-471-35645-X, New York, 1975.

[7] Ben-Hur A., Horn D., Siegelmann H. T., Vapnik V., *Support Vector Clustering*, Journal of Machine Learning Research, 125-137, 2001.

[8] Anderson E., *The Irises of the Gaspé Peninsula*, Bulletin of the American Iris Society, 59, 2-5, 1935.

[9] Ripley B. D., *Pattern Recognition and Neural Networks*, Cambridge University Press, ISBN 0-521-460-867, Cambridge, 1996.

[10] Hösel V., Walcher S., *Clustering Techniques: A Brief Survey*, AMS Subject Classification, 62H30, 68T10, 62-07, Germany, 2000.

[11] Venkataraman P., *Applied Optimization with MATLAB Programming*, John Wiley & Sons Inc., ISBN 0-471-34958-5, U.S.A., 2002.

[12] http://members.tripod.com/asim_saeed/paper.htm