KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

KÜME MERKEZLERİNİN BELİRLENMESİNDE YENİ BİR YÖNTEM (IFART)

DOKTORA TEZİ

Sevinç İLHAN

Anabilim Dalı: Elektronik ve Haberleşme Mühendisliği

Danışman: Doç. Dr. Nevcihan DURU

KOCAELİ, 2009

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

KÜME MERKEZLERİNİN BELİRLENMESİNDE YENİ BİR YÖNTEM (IFART)

DOKTORA TEZİ

Sevinç İLHAN

Tezin Enstitüye Verildiği Tarih: 09 Haziran 2009

Tezin Savunulduğu Tarih: 16 Eylül 2009

Tez Danışmanı
Doç.Dr. Nevcihan DURU
(.....)

Üye
Prof.Dr. Eşref ADALI
(.....)

Üye
Prof.Dr. Kadir ERKAN
(.....)

Üye
Yrd.Doç.Dr. M.Kemal GÜLLÜ
(.....)

Üye
Yrd.Doç.Dr. İbrahim ÖZÇELİK
(.....)

KOCAELİ, 2009

ÖNSÖZ ve TEŞEKKÜR

Kümeleme, çok boyutlu verilerde herhangi bir önbilgiye gerek olmaksızın doğal örüntülerin keşfedilmesini sağlayan ve bu nedenle çok sık başvurulan bir veri madenciliği yöntemidir. Veri tabanlarında toplanan veri miktarındaki artışla, kümeleme analizi son zamanlarda veri madenciliği araştırmalarında aktif bir konu haline gelmiştir. Kümeleme denince akla ilk gelen ve en sık kullanılan algoritma ise k-means algoritmasıdır. K-means algoritması başlangıçta seçilen rastgele küme merkezleri ile en iyi kümelemeyi garanti etmemektedir. Bu nedenle algoritma, birden çok kez farklı başlangıç noktaları ile çalıştırılmaktadır. Elde edilen küme sonuçları yeniden analiz edilmektedir. Başlangıç noktalarının farklı kümeleme algoritmaları ile belirlenmesi algoritmayı daha kararlı çalışan bir algoritma haline getirmekte, kümeleme hatalarını azaltmakta ve kümeleme hızını artırabilmektedir. Sadece verilerin kümelenmesi hedeflemek yeterli değildir. Bunun yerine elde edilen kümelerin yüksek kalitede kümeler olmasını garanti edecek yöntemler tercih edilmelidir ve bu tür yöntemlerin geliştirilmesine ve farklılaştırılmasına çalışılmalıdır.

Çalışmam süresince gösterdiği emek, destek ve ilgiyle çalışmanın gerçekleşmesi ve ilerlemesini sağlayan danışman hocam Sayın Doç. Dr. Nevcihan DURU' ya,

Çalışmam sırasında fikirleriyle bana yol gösteren ve çalışmanın gerçekleşmesini sağlayan hocam Sayın Prof. Dr. Eşref ADALI' ya,

Çalışmam sırasında fikirleriyle bana yol gösteren ve destekleyen hocam Sayın Prof. Dr. Kadir ERKAN' a,

Çalışmam sırasında yardımını hiçbir zaman esirgemeyen arkadaşım Arş. Gör. Dr. Gülşen AYDIN KESKİN' e ve K.Savaş OMURCA' ya,

Eğitim ve kariyer hayatım süresince sevgi ve emeği ile her zaman yanımda olan anneme ve babama,

Sonsuz teşekkürlerimi sunuyorum.

Bilgisayar Yük. Müh. Sevinç İLHAN

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SEMBOLLER

K	: küme sayısı
CM	: kümelerin birleşimi
r	: algoritmanın çalıştırılma sayısı
C	: kümeleme sonucu oluşan herhangi bir küme temsili
n	: veri tabanındaki nesne sayısı
e	: kümeleme hata oranı
d	: iki nokta arasındaki uzaklık
p	: veriye ait toplam nitelik sayısı
X	: Y.S.A. giriş vektörü
Y	: Y.S.A. çıkış vektörü
W	: Y.S.A. bağlantı ağırlık vektörü
i	: yarışmacı öğrenme ağı giriş birimleri
o	: yarışmacı öğrenme ağı çıkış birimleri
F1	: giriş katmanı
F2	: çıktı katmanı
I	: A.R.T. ağı için normalize edilmiş giriş vektörü
O	: men edici işaret
S	: F1 katmanı çıktısı (çıktı örüntüsü)
T	: F2 katmanı için girdi örüntüsü
Y	: F2 katmanı çıktı örüntüsü
X*	: F1 katmanında K.D.H. örüntüsü
Y*	: F2 katmanında bir örüntü
V	: A.R.T. 1 ağıdan geri yöndeki ağırlıklar
α	: seçim parametresi
ρ	: uygunluk parametresi
ß	: öğrenme oranı parametresi
M	: eşleşme fonksiyonu
T	: seçme fonksiyonu
N	: veri tabanındaki verilere ait nitelik temsili
V	: küme merkezi
p	: veriye ait toplam nitelik sayısı
$\cap$	: mantıksal VE operatörü (kesişim)
$\wedge$	: Bulanık VE operatörü (minimum)
$x \wedge y$	: minimum (x, y)
U	: üyelik derecesi matrisi
Σ	: toplam
v	: değişinti
m	: ortalama
ε	: küçük değerli bir sabit
k'	: kazanan hücre

Alt indisler

i	: giriş

j	: küme
s	: sınıf sayısı
θ	: giriş ile eşleşmeyen küme
(yeni)	: güncellenmiş ağırlık değeri
(eski)	: bir önceki ağırlık değeri
iç	: elemanların küme merkezine uzaklığı
dış	: kümeler arası uzaklık
p	: nitelik değeri indisi
k	: küme indisi
r	: rastgele sayı
o	: çıkış birimi
n1	: nitelik 1
n2	: nitelik 2

Kısaltmalar

V.T.B.K.	: Veri Tabanlarından Bilgi Keşfi
W.W.W.	: World Wide Web
A.G.N.E.S.	: AGlomerative NESting
D.I.A.N.A.	: DIvisive ANAlysis
R-SEL	: Rastgele Seçim Algoritması
R-MEAN	: Rastgele Ortalama Algoritması
S.C.S.	: Basit Küme Arama (Simple Cluster Seeking)
K.K.Z.	: Katsavaounidis Kuo Zhang
K.R.	: Kauffman Rousseuw
S.O.M.	: Kendi Kendini Organize Eden Model
A.R.T.	: Adaptif Rezonans Teorisi
F.A.R.T.	: Bulanık Adaptif Rezonans Teorisi
Y.S.A	: Yapay Sinir Ağları
L.V.Q.	: Vektör Kuantizasyon Modelleri
K.D.H.	: Kısa Dönemli Hafıza
U.D.H.	: Uzun Dönemli Hafıza
Y.Y.M.	: Yeniden Yerleştirme Modülü
A.R.T.M.A.P.	: Adaptif Rezonans Teorisi Bilişsel Haritaları
Bulanık A.R.T.M.A.P.	: Bulanık Adaptif Rezonans Teorisi Bilişsel Haritaları
İ.F.A.R.T.	: İyileştirilmiş Bulanık Adaptif Rezonans Teorisi
U.C.I.	: California, Irvire Üniversitesi
P.C.A-Part	: Temel Bileşenler Analizi (Principal Component Analysis-Part)
C.C.I.A	: Küme Merkezi Başlatma Algoritması (Cluster Center Initialization Algorithm)
R.B.F	: Radyal Tabanlı Ağlar
P.N.N	: Olasılıksal Sinir Ağları
G.R.N.N	: Regresyonlu Sinir Ağları
V.L.S.I.	: Büyük Ölçekli Entegre Devre

KÜME MERKEZLERİNİN BELİRLENMESİNDE YENİ BİR YÖNTEM (IFART)

Sevinç İLHAN

Anahtar kelimeler: Kümeleme, K-means, Başlangıç Küme Merkezlerinin Belirlenmesi, İyileştirilmiş Bulanık Adaptif Rezonans Teorisi.

Özet: İnsanoğlu sürekli olarak çevresinde gördüklerini sınıflama ya da kümeleme eğilimindedir. Bu nedenle kümeleme, veri madenciliği yöntemleri içerisinde en sık başvurulan ve en yaygın olarak kullanılanlardan bir tanesidir. K-means, büyük veri yığınlarını hızlı kümeleyebilen bir algoritma olması nedeni ile kümeleme algoritmaları içerisinde en yaygın olarak kullanılan algoritmadır. Ancak algoritmaya getirilen en büyük eleştiri, başlangıç parametrelerine aşırı duyarlı olmasıdır. Başlangıç parametreleri küme sayısı ve başlangıç küme merkezleridir. Dolayısı ile başlangıç küme merkezleri ne kadar iyi seçilebilir ise kümeleme de o kadar etkin ve doğru şekilde gerçekleştirilebilir. Elde edilen sonuç ağırlıklı olarak başlangıç küme merkezlerinin seçimine bağlı olarak değişmektedir. Genelde algoritma küme merkezlerine ait farklı başlangıç değerleri ile çalıştırılmakta ve en iyi kümelemenin belirlenebilmesi için sonuçlar birbiri ile karşılaştırılmaktadır.

Adaptif Rezonans Teorisi (A.R.T.) yapay sinir ağları, sınıflandırma problemleri için geliştirilmiş denetimsiz öğrenme algoritmalarıdır.

Tez kapsamında, bulanık adaptif rezonans teorisi ağlarının kümelemedeki başarısızlıkları, değerlendirilip giderilerek; iyileştirilmiş bulanık adaptif rezonans teorisi adı verilen bir algoritma önerilmiştir. Önerilen bu yöntem k-means algoritmasının başlangıç küme merkezlerinin belirlenmesinde kullanılmıştır. İyileştirilmiş bulanık A.R.T. ile başlatılan k-means kümeleme sonuçları, rastgele örnekler ile başlatılan k-means sonuçları ile karşılaştırılmıştır. Sonuç olarak hem hata payı hem de kümeleme hızı açısından k-means algoritmasının performansının başarılı şekilde artırıldığı gözlenmiştir. Ayrıca k-means algoritması daha kararlı bir algoritma haline gelmiştir.

A NEW METHOD FOR DETERMINING CLUSTER CENTERS (IFART)

Sevinç İLHAN

Key Words: Clustering, Improved Fuzzy Adaptive Resonance Theory, K-means, Initialization Cluster Centers.

Abstract: People always tend to classify or cluster the things seen around. Because of this, clustering is one of the frequently used data mining methods. The k-means algorithm is most commonly used algorithm among the clustering algorithms because of its ability to cluster the huge data quickly. However, the most important review about the algorithm is that, it is very sensitive to initial parameters. The initial parameters are the cluster number and the initial cluster centers. So, how much the initial cluster centers can be selected fairly, the clustering can be done more accurate and valid. The obtained result mostly depends on the selection of the initial cluster centers. Usually, k-means algorithm runs with different cluster centers and then the clustering results are compared to determine the best clustering situation.

Adaptive Resonance Theory (A.R.T.) is one of the learning algorithms without consultants which are developed for clustering problems in artificial neural networks.

Within the thesis the fuzzy art clustering results are evaluated. In order to make these results better, an algorithm called Improved Fuzzy Art is proposed. This proposed algorithm is used for determining the cluster centers for k-means algorithm. K-means clustering results which have been initialized with Improved Fuzzy Art method are compared with the results which have been initialized with random selection. Consequently, in terms of error rate and also execution time the performance of k-means algorithm increased successfully. Additionally, the k-means algorithm became more stable.

1. GİRİŞ

Bilgi yönetimi, farklı alanlarda faaliyet gösteren birçok organizasyon için, rekabette avantaj kazanmak ve sermayelerini doğru yönlendirebilmek konusunda ticari bir ihtiyaç haline gelmiştir. Bilgi yönetimi sistemi yapılandırılmadan önce organizasyonlara ait bilgi tanımlanmalı, düzenlenmeli ve yeniden gözden geçirilmelidir. Bu bağlamda verilerin kümelenmesi, organizasyonların hitap ettikleri müşteri gruplarına ait profillerin çıkarılması ya da ürettikleri ürün gruplarının özelliklerinin saptanması gibi konular açısından önemli bir analiz aşamasıdır. Benzer şekilde deneysel verilerin kümelenmesi, bilimsel verilerin işlenmesi ve yorumlanması; endüstriyel verilerin kümelenmesi, üretim planlama, strateji yönetimi gibi konular açısından önemli bir analiz aşamasıdır. Faaliyet alanı ne kadar farklı olursa olsun, toplanan verilerin doğru kümelenmesi bilgi yönteminin en önemli aşamalarından bir tanesidir.

Veri madenciliği büyük miktardaki veri içinden kullanışlı ve yararlı bilginin otomatik olarak keşfedilmesi işlemidir. Başka bir ifade ile büyük miktardaki veri arasında önceden bilinmeyen örüntülerin keşfedilmesi ya da gelecek ile ilgili tahminlerde bulunulmasını sağlayacak bağıntıların bilgisayar programları ile aranması işlemidir. Veri madenciliği, geçerli tahminler yapabilmek için, veri örüntülerini ve veriler arasındaki ilişkileri keşfetmek üzere bir takım veri analiz tekniği araçlarını kullanmaktadır. Veri madenciliği algoritmaları, market analizi, risk analizi, hata tespiti, metin madenciliği, internet madenciliği, bilgi yönetimi gibi birçok alanda kullanılmaktadır [1].

En önemli ve en sık başvurulan veri madenciliği yöntemlerinden bir tanesi sınıflandırmadır. Sınıflandırma, denetimli sınıflandırma (supervised classification) ve denetimsiz sınıflandırma (unsupervised classification) olmak üzere iki grupta incelenmektedir. Denetimli sınıflandırma tekniklerinde, sınıf modelleri önceden

belirlenmekte ve sınıflandırma bu modellere göre gerçekleştirilmektedir. Denetimsiz sınıflandırmada, benzer özellikler gösteren veriler herhangi bir ön bilgi olmadan gruplandırılmaktadır. Veri madenciliği yöntemlerinden olan kümeleme de bir denetimsiz sınıflandırma yöntemidir. Kümeleme yönteminde amaç, elemanları kendi içinde birbirlerine çok benzeyen, ancak birbirinden farklı özelliklerdeki kümelerin bulunması ve veri kümesindeki kayıtların bu farklı kümelere ayrılmasıdır. Verilerin hangi kümelere hatta kaç değişik kümeye ayrılacağı eldeki verilerin birbirlerine olan benzerliğine göre belirlenmektedir. Benzer verilerin farklı denetimsiz algoritmalar ile gruplandırıldığı kümeleme yöntemi bilgi keşfindeki önemli araçlardan bir tanesidir.

Sınıflandırma ve kümeleme algoritmalarından bazıları karar ağaçları [2], Bayesian sınıflandırıcılar [3], k-means kümeleme algoritması [4], yapay sinir ağı yöntemleri, S.O.M. (Self Oragnizing Maps) [5], bulanık A.R.T. [6]' dir.

Kümelemede en yaygın olarak kullanılan algoritma MacQueen (1967) tarafından geliştirilmiş olan k-means algoritmasıdır. K-means algoritması sürekli kümelerin yenilendiği ve en uygun çözüme ulaşana kadar devam eden döngüsel bir algoritmadır. Algoritmada her küme kendi küme merkezi ile temsil edilebilmektedir. Bu nedenle döngüsel işlemlerden önce K adet başlangıç küme merkezi belirlenmek zorundadır. Sonrasında döngüsel işlemler boyunca bu küme merkezleri sürekli olarak güncellenmektedir. K-means algoritması büyük ölçekli veri kümelerini hızlı şekilde kümelendirme özelliğine sahiptir.

Ancak k-means algoritması uygulamalarında başlıca iki sorun bulunmaktadır. Birincisi, çözüm ağırlıklı olarak başlangıç küme merkezlerine bağlı olarak değişmektedir. İkincisi, yalnızca doğrusal ayrılabilir kümeler bulunabilmektedir [7]. K-means' in başlangıç noktalarına göre çok farklı kümeler oluşabildiği için bu noktalarının iyi seçilmiş olması çok önemli bir etken haline gelmektedir [8]. Diğer bir deyişle, k-means algoritmasının geçerliliği ve performansı, çok büyük oranda seçilen başlangıç küme merkezlerine bağlıdır.

K-means algoritmasında parametre olarak belirlenmesi ve sunulması gereken değişken, K küme sayısıdır. Gerçek bir veri kümesinde K genelde önceden

bilinmemektedir. Uygulamalarda birçok *K* değeri denenmekte ve kümeleme geçerlilik teknikleri, küme sonuçlarının sınanması ve en iyi *K* değerinin tespit edilmesi için kullanılmaktadır. Mark Junjie Li [9] ve Hamerly [10], k-means algoritmalarında *K* değerinin belirlenmesi için istatistiksel metotları kullanmışlardır.

K-means algoritması seçilen başlangıç noktalarına göre çok farklı küme sonuçları ile sonlanmaktadır. Genellikle algoritma, küme merkezlerine ait farklı başlangıç tahminleri ile çalıştırılmakta ve elde edilen kümelerden en iyi kümeleme sonuçlarının belirlenebilmesi için sonuçlar birbirleri ile karşılaştırılmaktadırlar. Kaynaklarda, k-means tipindeki algoritmalarda asgari hedef fonksiyonu ile kümeleme sonuçlarının seçilmesi yöntemi uygulanmıştır [11]. Buna ek olarak, kümeleme geçerlilik(doğruluk) teknikleri en iyi kümeleme sonuçlarının seçilmesi için uygulanmışlardır [12]. Diğer yaklaşımlar, genetik algoritmaların yardımı ile bu sorunun çözülmesi için önerilmiş olan yaklaşımlardır [13-16]. Arthur ve Vassilvitskii (2007) kümeleme sonuçlarının kalitesinin yükseltilmesi için bir dikkatli arama yöntemi önermişlerdir [17].

Kaynaklarda, birbirinden farklı birçok yöntem k-means algoritmasında başlangıç noktalarının belirlenmesi için önerilmiş ve uygulanmıştır. Yöntemlerden başlıcaları bu bölümde belirtilmektedir.

Tez kapsamında, k-means algoritması için başlangıç küme merkezlerini belirleyen bir yöntem önerilmektedir. Önerilen bu yöntem bulanık A.R.T. algoritmasına dayananmaktadır. Bu bölümde, kaynaklarda kümelemeye k-means ile birlikte melez bir çözüm olarak sunulmuş olan yöntemlerden başlıcaları ve bulanık A.R.T. ile kümelemenin gerçekleştirildiği çalışmalardan bazıları incelenmektedir.

Başlangıç noktalarının belirlenmesi için yinelemeli bir yöntem Duda ve Hart [18] (1973) tarafından sunulmuştur. Bu yöntemin, mevcut veriyi alma daha sonra *K* kez rastgele harmanlama şeklindeki bir başka biçimi Thiesson ve diğerleri [19] tarafından 1997' de sunulmuştur.

Bradley ve Fayyad [20] 1998' de çok geniş veri kümeleri ile başa çıkabilmek için

rastgele örnekleme yönteminin alt-örnekleme biçimindeki şeklini önermişlerdir. Bu yöntemde algoritma tüm veri kümesinden J tane küçük rastgele alt-veri örnekleri seçmektedir $S_i = 1,...,J$. Alt-veri örnekleri k-means algoritması ile kümelendirilmektedir ($CM_i = 1,...,J$). Bu kümelerin birleşimi CM kümesini oluşturmaktadır. CM kümesi k-means algoritması ile kümelenmekte ve en son adım olarak k-means algoritması CM_i kümelerinden üretilen çözüm ile başlatılmaktadır.

Ting Su ve Jennifer Dy [21] 2004' de, k-means için bölümlemeli hiyerarşik yaklaşıma dayanan belirleyici bir başlangıç yöntemi önermişlerdir. Önerdikleri yöntemde, k-means algoritması için iyi seçilmiş başlangıç noktalarının düzenli olarak dağıtılmış olan noktalar olduklarını söyleyerek örnek uzayı hiyerarşik olarak bölümlemişlerdir. Bir küme ile başlayıp, sonra onu ikiye bölerek, bunlardan bir tanesi ile bölümleme işlemine devam etmek şeklinde bir yol izlemişlerdir. Bu işlem K tane küme kalana kadar devam etmektedir. Yönteme P.C.A.-Part (Principal Component Analysis-Part) adını vermişlerdir. Elde edilen sonuçlar iki adet iki boyutlu yapay veri kümesi ve üç adet gerçek veri kümesi ile değerlendirilmiştir. Rastgele örnekleme yöntemine göre çok daha etkin bir kümeleme geçekleştirildiği gözlenmiştir.

Pen~a ve diğerleri [22] 1999'da, k-means algoritmasında uygulanan dört farklı başlangıç yöntemi için üç gerçek veri kümesi üzerinde karşılaştırmalı bir çalışma sunmuştur ve rastgele örnekleme yönteminin k-means algoritmasını daha etkin yaptığı gözlenmiştir.

Kohei ve Barakbah [23] 2007' de, başlangıç noktalarının tespiti için bir hiyerarşik k-means algoritması önermişlerdir. K-means algoritması ile en iyi küme sonuçlarının elde edilebilmesi için algoritmanın tekrar tekrar çalıştırılması gerekmektedir. Algoritmanın kaç kez yeniden çalıştırılacağına karar vermek zor bir işlemdir. Bu tip belirsizlikler k-means algoritmasını, gerçek kümeleme problemleri için uygulanması zor bir algoritma haline getirmektedir. Bu çalışmada, veri kümesi birden çok kez k-means algoritması ile kümelenmekte ve küme sonuçları kaydedilmektedir. Her farklı işletim sonucunda elde edilen küme merkezleri belirlenmektedir. Bu noktalara

hiyerarşik kümeleme algoritmaları uygulanmaktadır. Hiyerarşik kümeleme algoritmasından sonra elde edilen en son merkez noktaları k-means algoritmasına başlangıç noktası olarak işaretlenmektedir. Altı adet gerçek veri kümesine uygulandıktan sonra elde edilen yeni küme sonuçlarının rastgele örnekleme yönteminden elde edilenlere oranla daha yüksek kalitede oldukları gözlenmiştir.

Shehroz S. Khan ve Amir Ahmad [24] 2004' de, C.C.I.A. olarak adlandırdıkları, k-means algoritması için başlangıç küme merkezlerini belirleyen bir algoritma önermişlerdir. Algoritma iki kısımdan oluşmaktadır. Birinci kısım K' küme merkezlerinin oluşmasını sağlamaktadır. Eğer K'>K ise o zaman algoritmanın ikinci kısmı işletilmektedir. *K* adet küme elde etmek için benzer birleştirilmektedir. Bu *K* adet nokta başlangıç küme merkezi olarak alınmaktadır. Algoritmanın ilk adımı birbirinden ayrı nitelik değerleri için küme merkezlerinin hesaplanması adımıdır. Bunu başarabilmek için k-means algoritması nitelik alanı üzerinden uygulanmaktadır. K-means için başlangıç noktaları belirlenirken sıra dışı veriler dışarıda bırakılmıştır. Benzer işlemler tüm nitelik alanları için uygulanmaktadır. Rastgele örnekleme yöntemine göre daha iyi sonuçlar alındığı gözlenmiştir.

Moth'd Belal ve Al-Daoud [25] 2005' de, adımları: maksimum değişintiye sahip olan boyutu (nitelik değerini) bulmak, bu boyuttaki değerleri sıralamak, gruplara ayırmak ve her grup için ortanca bularak ve bunları k-means algoritmasına başlangıç noktaları olarak sunmak şeklinde işleyen bir algoritma önermişlerdir. Sonuçlar iki gerçek veri kümesinde sınanmıştır. Önerdikleri yöntemin rastgele örnekleme yöntemlerine göre daha kaliteli kümeler oluşturduğu gözlenmiştir.

Fuyuang Cao ve diğerleri [26] tarafından 2009' da komşuluk temelli bir kaba küme modeli kullanılarak, nesnelerin komşulukları arasındaki bağlantı derecesi bu modele göre tanımlanmıştır. Komşuluklara en yüksek uyum derecesi olan noktalar başlangıç noktaları olarak belirlenmiştir. Önerdikleri yöntemin rastgele örnekleme yöntemlerine göre daha kaliteli kümeler oluşturduğu gözlenmiştir.

Yukarıda da bahsedildiği üzere kaynaklarda k-means algoritması için birçok rastgele

başlatma yöntemi geliştirilmiştir. Bunlardan klasik olan ve en çok uygulanan iki tanesi, rastgele örnekleme ve rastgele bölümleme yöntemleridir. Rastgele örnekleme yöntemi, verilerden rastgele *K* tane örneği başlangıç küme merkezi olarak seçmekte ve diğer örnekleri en yakınlarındaki başlangıç noktasına göre bir kümeye dahil etmektedir. Rastgele bölümleme yöntemi, her veri örneğini rastgele seçtiği *K* tane kümeden birisine dahil etmektedir. Rastgele başlangıç noktalarına göre elde edilen birbirinden farklı kümeleme sonuçları arasından en doğru ve geçerli olanı seçebilmek için algoritma r kez çalıştırılmaktadır.

Bu yöntemlerdeki temel problem, r kez çalıştırılsalar dahi en iyi çözümü garanti etmiyor olmalarıdır, aynı zamanda algoritma için zaman karmaşıklığı da iyice artmaya başlamaktadır [8].

K-means algoritmasının başlangıç noktalarına duyarlılığına rağmen, eğer başlangıç noktaları çözüm kümelerine yakın noktalar olarak seçilir ise k-means algoritması yüksek olasılık ile doğru ve geçerli kümeleri bulabilecektir, aksi takdirde yanlış küme sonuçlarına doğru bir yönelim izleyecektir.

Kaynaklarda, bulanık Adaptif Rezonans Teorisi (Bulanık A.R.T.) çok farklı alanlarda veri kümeleme problemine çözüm olarak önerilmiş ve uygulanmıştır. Yapılan bu çalışmalardan bazılarına aşağıda değinilmektedir.

Kondadadi ve Kozma [27] 2002'de yazılı belgelerin kümelenmesinde bulanık A.R.T. algoritmasını kullanmışlardır. L. Cinque ve diğerleri [28] 2004'de görüntü işlemede bulanık A.R.T. algoritmasının değiştirilmiş bir çözümünü önermişlerdir. C. Chen ve L. Wang [29] 2006'da, bulanık A.R.T yöntemini kullanarak daha etkin bir kümeleme aracı önermişlerdir. Xiang ve diğerleri [30] 2006'da bilgisayar ağlarında beklenmeyen saldırıların tespitinde bulanık A.R.T. yöntemini kullanan bir sistem önermişlerdir. Xu ve diğerlerinin [31] 2007'de önerdikleri kanserli hücrelerin tespitinde kullanılan bir sistem içerisinde, kanser örneklerinin bölünmesi için bulanık A.R.T. yöntemi uygulanmıştır. Kumar ve diğerleri [32] 2008'de bulanık A.R.T. yaklaşımını algılayıcı ağlarına ait verilerin kümelenmesinde kullanmışlarıdır. Isawa ve diğerleri [33] 2008'de üstü üste çakışan kümeleri bir araya getiren yeni bir bulanık

A.R.T. algoritması önermişlerdir. Gu ve diğerleri [34] 2008'de bulanık A.R.T temelli bir yüz tanıma algoritması önermişlerdir.

1.1. Tezin Katkısı

Birçok kümeleme algoritması, başka kümeleme algoritmaları için başlangıç algoritması şeklinde melez kümeleme çözümü olarak uygulanabilmektedir [20]. Tez kapsamında da melez bir kümeleme gerçekleştirilmektedir. Bu tez kapsamında önerilen İyileştirilmiş Bulanık A.R.T. (Improved Fuzzy A.R.T. - İ.F.A.R.T.) algoritması k-means kümeleme algoritmasının başlangıç küme merkezlerini belirleyen bir algoritma olarak önerilmiş ve uygulanmıştır.

İ.F.A.R.T. algoritması, Carpenter, Grossberg ve Rosen tarafından [6] 1991' de geliştirilmiş olan Bulanık Adaptif Rezonans Teorisi (Bulanık A.R.T.) algoritmasına dayanan bir kümeleme çözümü sunmaktadır.

Veri kümeleri bulanık A.R.T. (Fuzzy Adaptive Resonance Theory - F.A.R.T.) ile kümelendirildikten sonra kümeler incelendiğinde etkin ve geçerli bir kümeleme geçekleştirilemediği, kümelerin sınırlarının birbirlerinin içine geçmiş olduğu, iyi ayrılmış kümeler olmadıkları gözlenmiştir. Bu nedenle, kümeleme F.A.R.T. ile gerçekleştirildikten sonra kümeler üzerinde bir iyileştirme işlemi gerçekleştirilmiştir. Önerilen bu yöntemde, her giriş verisinin F.A.R.T. sonucu oluşan her kümeye üyelik derecesi hesaplanmaktadır. Üyelik dereceleri hesaplanırken kümeyi temsil eden eleman olarak küme merkezi seçilmektedir. Sonrasında, üyelik dereceleri incelenerek her giriş verisi maksimum üyelik derecesi ile bağlı olduğu kümeye taşınmaktadır. Böylelikle F.A.R.T. sonucu oluşan kümelerin elemanları üzerinde bir yer değiştirme işlemi gerçekleştirilmiş olmaktadır. Yer değiştirme sonucu elde edilen yeni kümeler ile F.A.R.T. ile oluşturulmuş olan eski kümeler karşılaştırıldığında İ.F.A.R.T. yönteminin F.A.R.T.'a göre çok daha geçerli bir kümeleme gerçekleştirildiği gözlenmiştir.

Yapılan çalışmanın ikinci aşamasında İ.F.A.R.T. algoritmasının oluşturduğu küme merkezleri k-means algoritmasının başlangıç küme merkezleri olarak algoritmaya

sunulmaktadır. Bu şekilde çalıştırılan k-means algoritması ile standart k-means algoritması adım sayısı, kümelemedeki hata oranı ve kararlığı açısından değerlendirilmektedir.

İ.F.A.R.T. algoritması ile küme merkezleri sonuçta elde edilecek olan kümelere daha yakın seçilebildiği için k-means algoritmasının adım sayısı, rastgele başlatıldığında elde edilen adım sayısına oranla azaltılmıştır.

Standart k-means algoritması ve İ.F.A.R.T. ile başlatılan k-means algoritması ile elde edilen hata oranları karşılaştırıldıklarında; önerilen yöntem ile daha düşük hata oranlarında kümeleme gerçekleştirildiği gözlenmiştir.

En son ölçüt olarak mevcut şu durum değerlendirilmiştir. K-means algoritması rastgele küme merkezleri ile başlatıldığında çok değişik küme sonuçları oluşturmaktadır. Bunlar arasından en iyi kümelemenin seçilebilmesi için algoritmanın defalarca çalıştırılması ve bunlar arasında en iyi kümelemenin seçilmesi gerekmektedir. Buna karşın başlangıç küme merkezleri İ.F.A.R.T. ile belirlendikten sonra k-means algoritmasının sadece bir kez çalıştırılması yeterli olmaktadır. Bu durumda İ.F.A.R.T. ile başlatılan k-means algoritması daha kararlı yapıda çalışmaktadır.

1.2. Tezin Düzenlenmesi

Bu tez, yukarıdaki çözüm aşamaları paralelinde, yedi bölüm halinde yazılmıştır. Birinci bölümde, problemin tanımı, kaynak incelemesi ve çözüm aşamaları ana hatları ile verilmekte, tezin genel bir tanımı yapılmaktadır.

İkinci bölümde veri madenciliğine ait genel tanım ve kavramlara, veri madenciliğinin önemi, gelişen teknolojiler ile ortaya çıkan problmelere nasıl çözüm üretebildiği gibi konulara, uygulama alanlarına ve veri madenciliği yöntemlerine yer verilmektedir.

Üçüncü bölümde veri madenciliği yöntemlerinden kümeleme yöntemi ve bir kümeleme algoritması olan k-means algoritmasından bahsedilmektedir. Bu bölümde

kümeleme algoritmalarının özellikleri, kümeleme analizi, kümeleme geçerlilik ölçütleri gibi konulara yer verilmektedir.

Dördüncü bölümde yapay sinir ağları ve bir yapay sinir ağı algoritması olan bulanık A.R.T. algoritması anlatılmaktadır. Yapay sinir ağlarının genel özelliklerinden, denetimsiz ve yarışmacı öğrenmenin kurallarından, denetimsiz öğrenme gerçekleştiren yapay sinir ağı algoritmalarıdan bahsedilmektedir. Bulanık A.R.T. algoritması ayrıntılı olarak incelenmektedir.

Beşinci bölümde İ.F.A.R.T. algoritmasına ait genel kuramsal tanımlar, algoritmanın bulanık A.R.T. algoritmasından farkı, çalışma şekli verilmektedir. Örnek veri kümelerinden İ.F.A.R.T. algoritması ile edilen kümeleme sonuçları, bulanık A.R.T. ve S.O.M. algoritmasından elde edilenler ile karşılaştırılmaktadır.

Altıncı bölümde, beşinci bölümde anlatılmış olan İ.F.A.R.T. algoritması k-means algoritmasının başlangıç küme merkezlerini belirleyen yöntem olarak önerilmiştir. İ.F.A.R.T. ile başlatılan k-means ve standart k-means algoritmasına ait deneysel sonuçlar verilmektedir.

Tezin son bölümünde ise, genel olarak elde edilen sonuçlar ve ileriki çalışmalar için yararlı olabileceği düşünülen bazı saptamalara yer verilmektedir.

2. VERİ MADENCİLİĞİ

2.1. Giriş

Bu bölümde, veri madenciliğinin tanımı, önemi, bilgi teknolojilerinde neden bir ihtiyaç haline geldiği, veri madenciliği yöntemleri ve uygulama alanları ile ilgili genel bilgilere yer verilmektedir.

2.2. Veri Madenciliğine Neden İhtiyaç Duyulmuştur?

Son zamanlarda bilgi teknolojilerinde dikkati çekecek ölçülerde yaşanan gelişmeler ile depolanan veriler çok büyük boyutlara ulaşmaya başlamıştır. Bilgisayar sistemleri her geçen gün ucuzlamakta, bununla birlikte işlemciler gittikçe hızlanmakta, disklerin veri depolama kapasiteleri artmaktadır. Buna bağlı olarak daha büyük miktarlardaki veri saklanabilmekte ve daha kısa sürelerde işlenebilmektedirler. Bilgisayar sistemleri sayesinde verinin sayısal olarak toplanması ve saklanabilmesi sağlanmakta, bunların sonucu olarak da ayrıntılı ve doğru bilgiye erişilebilmektedir.

Örneğin eskiden süpermarketteki kasalar basit bir toplama makinesinden ibaretti ve yalnızca müşterinin o anda satın almış oldukları malların toplam tutarını hesaplamak için kullanılmaktaydılar. Günümüzde ise kasa yerine kullanılan satış noktası terminalleri sayesinde müşteri hareketlerinin bütün detayları saklanabilmektedir. Depolanan binlerce malın ve müşterinin hareket bilgileri sayesinde, her malın zaman içindeki hareketi, müşterilerin zaman içindeki hareketleri, satın almış oldukları ürünler ile ilgili ayrıntılı analizler gerçekleştirilebilmektedir. Bunun dışında banka ve kredi kartı işlemleri, bilimsel veriler, uydu ve radarlardaki algılayıcılardan gelen veriler, web verileri gibi veriler de depolanmakta ve veriler üzerinde ayrıntılı analizler gerçekleştirilmektedir.

Bilgi, bir amaca yönelik olarak işlenmiş veridir. Veri kendi başına değersizdir ancak bir hedef doğrultusunda bilgiye dönüştürülürse değer kazanmaktadır. Verinin bilgiye çevrilmesi işlemine ise "veri analizi" denmektedir.

Yukarıda verilen süper market örneğinde, veri analizi yapılarak her mal için sonraki ayın satış tahminleri çıkarılabilmekte, müşteriler satın aldıkları ürünlere göre gruplanabilmekte, yeni bir ürün için potansiyel müşteriler belirlenebilmekte ve müşterilerin zaman içindeki hareketleri incelenerek onların davranışları ile ilgili tahminler yapılabilmektedir. Binlerce ürün ve müşterinin olacağı düşünüldüğünde bu analizlerin otomatik olarak gerçekleştirilmesi gerektiği kaçınılmazdır.

Geniş ölçekli veri tabanları arasından yararlı veriye ya da bilgiye erişim ihtiyacı," veri madenciliği tekniklerini güncel araştırma konularından birisi haline getirmiştir. Veri madenciliği iş yönetimi, ürün kontrol sistemleri, market analizi, finans yönetimi, risk analizi, mühendislik ile ilgili analizler gibi konularda kullanılmaya başlanmıştır. Veri madenciliği aslında, bilgi teknolojilerinin doğal gelişim sürecinin sonucu olarak da değerlendirilebilir.

Çok büyük ölçekli veriler farklı alanlardaki büyük ölçekli veri tabanları içlerinde değerli verileri bulunduran bir veri madeni gibi düşünülebilir. Bu büyüklükteki verilerin analizi, bu analiz sonucunda daha anlamlı bilgi elde etme ve elde edilen bilgiyi yorumlama işi insan yeteneğini ve ilişkisel veri tabanlarının yapabileceklerini aşmaktadır. Bu ihtiyaçların sonucunda otomatik ve akıllı veri tabanı analizi için yeni kuşak teknikler doğmuştur. Veri madenciliği teknikleri veriyi akıllı ve otomatik şekilde yararlı bilgiye dönüştürebilen teknikler şeklinde cevap olarak sunulmuşlardır. Veri madenciliği ile keşfedilen bilgi, bilgi yönetimi, karar mekanizmaları, kontrol sistemleri ve sürekli veri takibi gibi birçok farklı uygulama alanında kullanılabilmektedir [1]

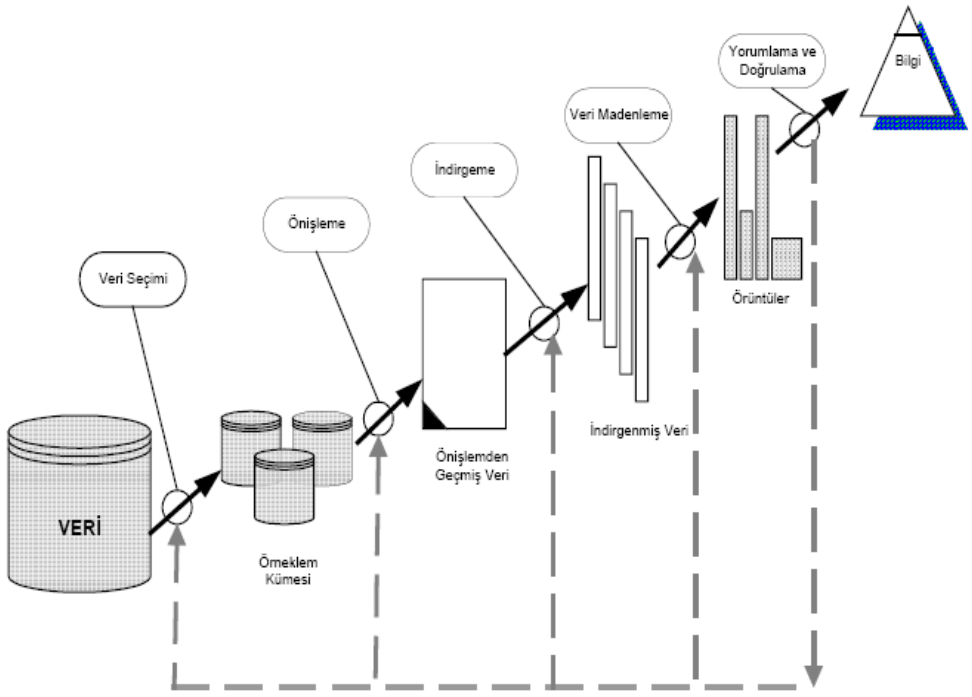
2.3. Veri Madenciliği Nedir?

Veri madenciliği geniş ölçekli veriler içinden kullanışlı ve yararlı bilginin otomatik olarak keşfedilmesi işlemidir. Başka bir ifade ile büyük ölçekli veriler arasında

önceden bilinmeyen örüntülerin keşfedilmesi ya da gelecek ile ilgili tahminlerde bulunulmasını sağlayacak olan bağıntıların bilgisayar programları ile aranması işlemidir [1].

2.3.1. Veri madenciliği ve bilgi keşfi

Kaynaklarda veri içinden faydalı örüntülerin bulunması işlemine pek çok terim karşılık gelmektedir. Bunlardan bir tanesi ve en çok kullanılanı Veri Tabanlarından Bilgi Keşfi (VTBK)' dir. VTBK' nın tanımı ve faaliyet alanının ne olacağı konusunda farklı yaklaşımlar bulunmaktadır. Veri madenciliği, veri tabanlarından bilgi keşfine ait kısımlardan bir tanesidir. Fayyad' a göre VTBK sürecine ait adımlar Şekil 2.1 'de gösterilmektedir [35].



Şekil 2.1: Bilgi keşfi adımları [36]

VTBK süreci adımları aşağıdaki şekilde özetlenebilirler;

- Veri Seçimi: Bu adım birkaç veri kümesini birleştirerek, sorguya uygun örneklem kümesini elde etmeyi gerektirir.

- Veri Temizleme ve Önişleme: Seçilen örneklemde yer alan hatalı verilerin çıkarılması, eksik ve gürültülü niteliklerin değiştirilmesi aşamasıdır.
- Veri İndirgeme: Seçilen örneklemden ilgisiz niteliklerin atıldığı ve tekrarlı tutanakların ayıklandığı adımdır. Bu aşama seçilen veri madenciliği sorgusunun çalışma zamanını iyileştirir.
- Veri Madenciliği: Bir veri madenciliği tekniğinin işletilmesi aşamasıdır.
- Değerlendirme: Keşfedilen bilginin geçerlilik, yenilik, yararlılık ve basitlik ölçütlerine göre değerlendirilmesi aşamasıdır.

2.4. Veri Madenciliği Uygulama Alanları

Veri madenciliği bankacılık, pazarlama, sigortacılık, sağlık gibi değişik alanlarda uygulanmaktadır. Veri madenciliğinin uygulanmasında sektör farkı gözetilmemekle beraber, geniş veri ambarlarının oluşturulmasına olanak veren, perakende satış, sigortacılık, sağlık gibi alanlarda kullanılması daha yaygın ve doğrudur. Uygulama alanları ana başlıklar halinde aşağıda incelenmektedir;

2.4.1. Pazarlama yönetimi

Pazarlama alanıyla ilgili olarak bu güne kadar yapılmış ve yapılmakta olan uygulamaların bazıları şunlardır:

- Müşterilerin satın alma örüntülerinin belirlenmesi: Müşterileri herhangi bir ürünü aldıktan sonra anlamlı bir sıklıkla başka bir ürünü alıyor mu? Sorusunun cevabı gibi satın alınan ürünler arasındaki örüntüler yakalanmaya çalışılmaktadır.

- Müşterilerin demografik özellikleri arasındaki bağlantıların bulunması: Müşterilerin, yaşı, eğitim durumu, medeni hali gibi özellikleri ile satın alınan ürünler arasında herhangi bir bağıntı var mı? Bu sorunun yanıtı seçilecek uygun bir veri madenciliği teknik veya yöntemi ile verilebilmektedir.

- Posta kampanyalarında cevap verme oranının artırılması: Gerek tanıtım promosyon için yapılan, gerekse belirli bir ürüne ilgi gösteren potansiyel müşteri

grubu hakkında bilgi sahibi olmak için yapılan posta kampanyalarına katılımın artırılabilmesi için bu kampanyaya sadece katılması en olası kişileri dahil etmek bir çözüm olabilir. İşletmenin mevcut müşterilerinin hangilerinin yapılacak posta kampanyalarına katılmasının olası olduğu veri madenciliği ile belirlenmektedir.

- Mevcut müşterilerin elde tutulması, yeni müşterilerin kazanılması: Mevcut müşterilerin bağlılığının sınanıp kaybedilmeye en yakın müşterilerin ve yine kazanılmaya en yakın müşterilerin belirlenmesi veri madenciliği yöntemleri ile yapılmaktadır. Bu her sektörde kullanılabilir; örneğin telekomünikasyon şirketleri tarafından sıkça kullanılmaktadır.

- Pazar sepeti analizi: Özellikle süper market gibi alışveriş merkezlerinde, müşteriler birden fazla ürün alırlar. Acaba A ürünü alan müşteri yanında başka bir B ve C gibi ürün alıyor mu? B ürünü alan kişinin aynı gün C ürünü alma olasılığı nedir? Bu sorulara cevaplar, uygun veri madenciliği yöntemleri kullanılarak belirlenebilmektedir.

Bunların dışında pazarlama yöntemleri açısından veri madenciliği şu konularda da kullanılabilmektedir:

- Müşteri ilişkileri yöntemi
- Müşteri değerlendirme
- Satış tahmini
- Birlikte satış

2.4.2. Risk yönetimi ve dolandırıcılık saptama

Dolandırıcılık başlıca şu başlıklar altında değerlendirilebilir.

- Kredi kartı dolandırıcılığı
- Internet dolandırıcılığı
- Sigorta dolandırıcılığı
- Bilgisayar sistemleri ve bilgisayar ağlarına girme

- Telefon dolandırıcılığı
- Üyelik abonelik dolandırıcılığı

Listede de görüldüğü gibi teknolojinin ilerlemesi ile birlikte dolandırıcılık türlerinde de bir artma olmuştur. Kredi kartı veren finans kuruluşları daha dolandırıcılık meydana gelmeden dolandırıcılığı tespit etmektedirler. Bunun için de bilgi keşfi, yapay zeka ve veri madenciliği gibi yöntemler kullanmaktadırlar.

2.4.3. Diğer uygulamalar

Pazarlama ve risk yöntemi dışında veri madenciliği şu alanlarda da kullanılmaktadır.

- İşaret işleme: Telefon hatlarında parazitlenmeden dolayı oluşacak kayıpları ve buna bağlı olarak konuşmada ortaya çıkan gürültüyü yok etme gibi konularda kullanılmaktadır.

- Biyoloji: DNA sıra (veri) analizinde kullanılmaktadır. İnsanda yaklaşık yüz bin gen vardır. Hastalıklara yol açan gen sıralama örneklerini binlerce gen arasından bulmak, tanımlamak oldukça zor bir iştir. Veri madenciliği ile geliştirilen sıralama örnek analizi ve benzerlik arama yöntemleri DNA verisi üzerinde analiz yapmayı kolaylaştırmaktadır.

- Tıp: Daha önceden işlem uygulanmış, dış bulguları ve operasyon sonucu kaydedilmiş hasta adaylarına ait veritabanı, veri madenciliği algoritmaları tarafından incelenerek, bir makine öğrenmesi, sınıflama, karar ağacı vs. tekniği gerçekleştirilmektedir [37].

2.4.4. Metin madenciliği

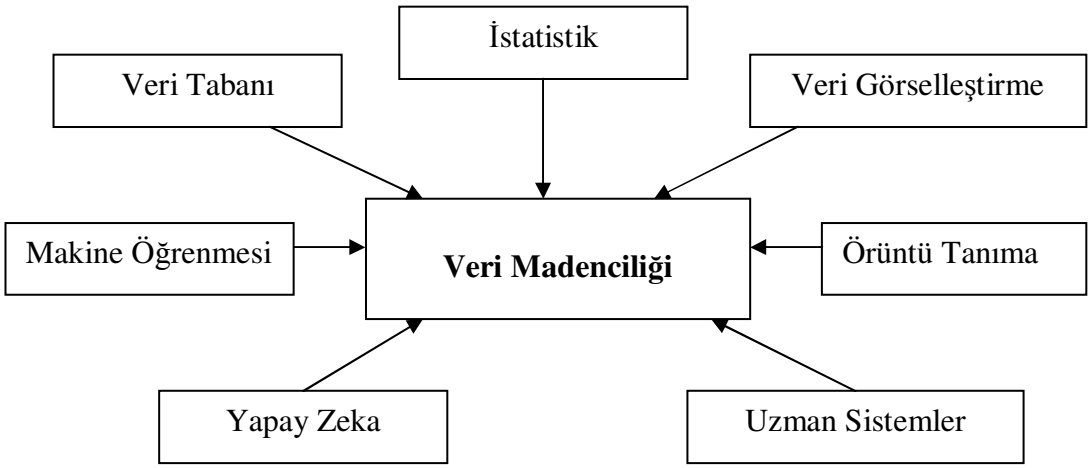
Çok büyük ve anlamsız metin yığınları arasından anlamlı ilişkiler elde etmekte kullanılmaktadır. Metinlerin sınıflandırılması, metin için anahtar sözcüklerin tespit edilmesi, otomatik soru-cevap sistemleri gibi uygulamalar gerçekleştirilebilmektedir.

2.4.5. İnternet madenciliği

İnternet üzerindeki veriler hem boyut, hem de karmaşıklık olarak hızla artmaktadır. İnternetin belirli sınıflara ayrılarak veriye ulaşım süresinin azaltılması web madenciliğinin temel hedefidir.

2.5. Veri Madenciliği ve Diğer Disiplinler

Veri madenciliği farklı disiplinlerin bir kesişim noktası olarak doğmuştur ve bu bağlamda gelişmesini sürdürmektedir. Veri madenciliği, makine öğrenimi, istatistik, veri tabanı yönetim sistemleri, veri ambarlama gibi farklı disiplinlerde kullanılan yaklaşımları birleştirmektedir [38]. Bahsedilen bu yapı temel olarak Şekil 2.2'de görüldüğü gibi ifade edilebilir.



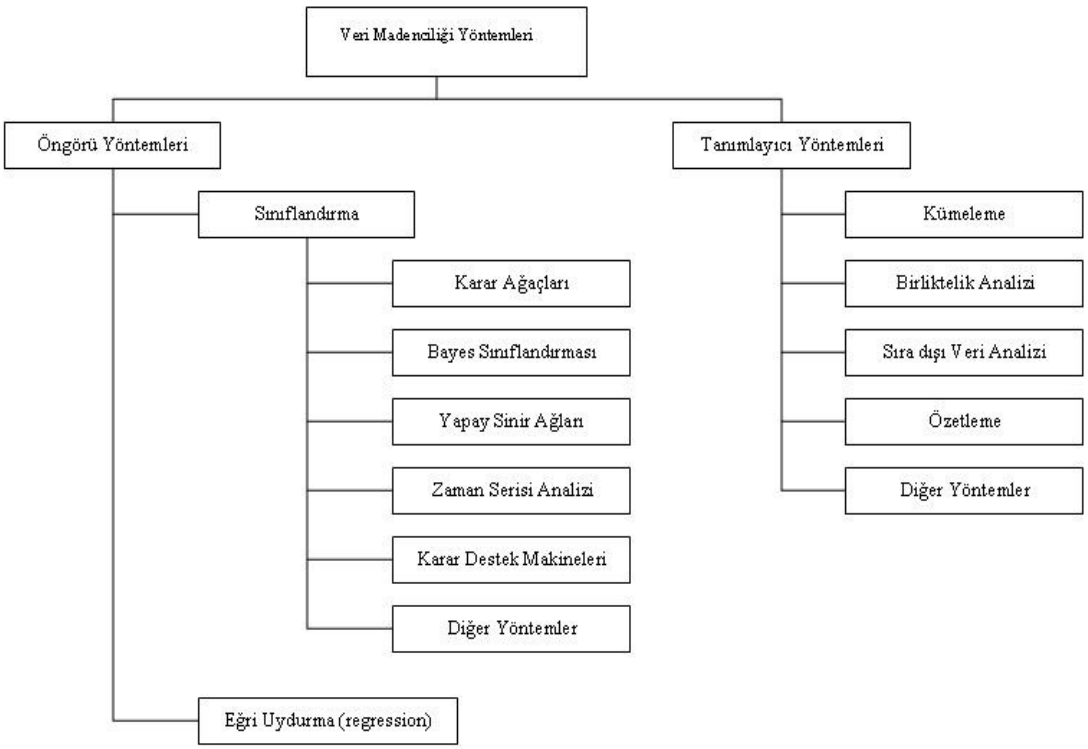
Şekil 2.2: Veri madenciliğinin diğer disiplinlerle ilişkisi

Makine öğrenmesi, örüntü tanıma ve istatistik alanları, veri madenciliğinde örüntü keşfetme aşamasında; yapay zeka teknolojileri, bulunan örüntüleri yorumlama aşamasında; veritabanı teknolojileri eldeki verileri depolama, süzme, temizleme, sorgulama işlemi aşamasında; veri görselleştirme ise, raporlama ve insan beyni için anlamlı sembollere çevirme aşamasında veri madenciliğine yardımcı olmaktadır.

2.6. Veri Madenciliği Yöntemleri

Genel olarak veri madenciliği yöntemleri temelde iki sınıfa ayrılmaktadır [4]. Bu sınıflama Şekil 2.3' de daha ayrıntılı olarak görülmektedir.

- Tanımlayıcı yöntemler: Veriyi tanımlayan yorumlanabilir örüntülerin keşfedilmesini sağlayan yöntemler.
- Öngörü yöntemleri: Öngörü amacı ile var olan verilerden yorum çıkarılmasını sağlayan yöntemler.



Şekil 2.3: Veri madenciliği yöntemleri

Veri madenciliği yöntemleri kullanıldıkları veri yapılarına ve keşfedebildikleri örüntü biçimlerine göre sınıflara ayrılmaktadır. Farklı kaynaklarda veri madenciliği yöntemleri için farklı gruplandırmalar görülmektedir. Bunların arasında en yaygın olarak kabul göreni J.Han'ın [1] ortaya sürdüğü sınıflardır ve bu bölümde de bunlar incelenmektedir.

Veri Madenciliği yöntemleri şu şekilde sıralanmaktadır:

- Tanımlama ve Ayrımlama (Characterization and Discrimination)
- Birliktelik Analizi (Association Analysis)
- Sınıflama ve Öngörü (Classification and Prediction)
- Kümeleme Analizi (Cluster Analysis)
- Sıra dışılık Analizi (Outlier Analysis)
- Gelişimsel Analiz (Evolution Analysis)

2.6.1. Tanımlama ve ayrımlama

Veriler gösterdikleri ortak özelliklere göre genelleştirilmiş sınıflara ayrılabilmektedirler. Bir firma müşteri profilini, alışveriş ortalaması belirli bir miktardan daha yüksek olan müşterileri "zengin", diğerlerini ise "orta halli" ya da "fakir" şeklinde tanımlayarak belirleyebilmektedir. Bu tür genellemeler veri kümesinin elemanlarının ortak özelliklerini belirlemekte ve diğer veri kümelerinden de farklılıklarını ortaya koymaktadır.

Bu iki tür veri madenciliği yöntemi birbirine çok benzer teknikler kullanmaktadırlar. Ayrıca her iki yöntemle elde edilen sonuçlar pasta grafiği, sütun grafiği, eğriler ve çok boyutlu küpler ile sunulmaktadır.

1-) Tanımlama

Bir veri kümesinin elemanlarının genel özelliklerini özetlemek için kullanılmaktadır. Örneğin bir alışveriş merkezinde "bu yıl satışı oranı %25'in üzerinde artan mallar" ifadesi bir tanımlama işlemidir.

2-) Ayrımlama

Bir veri kümesinin diğer bir veri kümesinden farklarını ortaya çıkarma işlemidir. Örneğin "bu yıl satış oranı %10 artan mallar ile satış oranı %15 azalan mallar" ın karşılaştırılması ayrımlama tabanlı veri madenciliğidir.

2.6.2. Birliktelik analizi

Birliktelik analizi, bir veri kümesinde kendiliğinden, sıklıkla gerçekleşen, birlikte ya da aynı süre içinde alınma, yapılma, oluşma gibi etkileri keşfetme temeline dayanmaktadır. Bankacılık işlemlerinin analizinde ya da pazar sepeti analizinde yaygın olarak kullanılan bir yöntemdir. Pazar sepeti analizi, bir alışveriş sırasında veya birbirini izleyen alışverişlerde müşterinin hangi mal veya hizmetleri satın alma eğiliminde olduğunun belirlenmesiyle müşteriye daha fazla ürün satılması yollarının aranmasıdır [39].

Birliktelik analizi yalnızca mal ve hizmetlerin birlikte satın alınması için değil aynı zamanda hangi koşulları sağlayan müşterilerin hangi ürünleri alacağı hakkında da çözümler getirmektedir. Örneğin bir banka kredi kartı kayıtları incelendiğinde, "yaşları 20 ile 29 arasında değişen müşterilerden, gelirleri 700 TL ile 900 TL arasında değişen müşterilerin bilgisayar satın aldıkları görülmüştür" gibi ilişkiler yakalanmaktadır.

2.6.3. Sınıflama ve öngörü

Sınıflama işlemi insan düşünce yapısına en uygun veri madenciliği yöntemidir. İnsanoğlu dünyayı, çevresindeki nesneleri ve olayları daha iyi anlamak ve başkalarına anlatabilmek için hemen her şeyi sınıflandırma eğilimindedir. Örneğin, insanları davranışlarına göre, hayvanları türlerine göre, evleri görünüşlerine göre sınıflandırmaktadır. Bu nedenle en sık kullanılan yöntemlerdendir.

Veri madenciliğinde sınıflama, eldeki mevcut verileri önceden belirlenen bir özelliğe göre sınıflara ayırma ve yeni eklenecek verilerin hangi sınıfa dahil olacağının belirlenmesi işlemdir. Sınıflama işlemi denetimli ya da denetimsiz olarak gerçekleştirilmektedir. Denetimli sınıflamada, hangi veri nesnesinin hangi sınıfa dahil edileceği ve sınıfların sayısı önceden bilinmektedir. Denetimsiz sınıflamada ise hangi veri nesnesinin hangi sınıfta olduğu ve sınıf sayısı önceden bilinmemektedir. Bu tür sınıflamaya kümeleme de denmektedir. Bankaların kredi başvurularını düşük, orta ve yüksek riskli olarak sınıflandırması bu yönteme örnek olarak verilebilir.

Öngörü işlemi sınıflama işlemine çok benzemektedir. Ancak öngörü işleminde sınıflanma, gelecek için tahmin edilen belirli bir davranışa ya da belirli bir değere göre yapılmaktadır. Öngörü işleminde yapılan sınıflamanın doğru olup olmadığını sınamanın tek yolu "bekle ve gör" ilkesidir [40]. Öngörü işlemine örnek olarak deprem tahmini, bir turizm şirketi müşterilerinden hangilerinin bu yaz yurtdışında tatil yapmak isteyeceğinin belirlenmesi verilebilir.

Bir veri madenciliği uygulamasında ayrık nitelik değerlerini tahmin etmek sınıflama iken, sürekli nitelik değerlerini tahmin etmek öngörüdür. Örneğin hangi topun hangi sepete koyulabileceği sınıflama iken, topun ağırlığı öngörüdür [41].

Sınıflama ve öngörü işleminde temel olarak karar ağaçları, yapay sinir ağları, Bayesian sınıflama, genetik algoritmalar gibi teknikler kullanılmaktadır.

2.6.4. Kümeleme analizi

Kümeleme işleminin sınıflama işleminden en önemli farkı önceden belirlenmiş sınıflar ya da sınıf tanımlarının olmamasıdır. Bu yüzden kümeleme işlemi bir denetimsiz veri madenciliği yöntemidir. Sonuçta kaç adet küme oluşturulması gerektiği genelde veri elemanlarının birbirlerine olan benzerlikleri düşünülerek önceden belirlenmektedir. Bu anlamda, kümeleme işlemi sonunda elde edilen kümeler kullanılan yöntemin giriş parametrelerine bağımlı olsa da, giriş parametrelerinden bağımsız kümeleme teknikleri geliştirme çalışmaları sürmektedir [42].

Kümeleme işleminde amaç, küme içi benzerliği en yüksek, kümeler arası benzerliği en düşük yapmaktır. Bir kümeleme yönteminin geçerliliği ve doğruluğu bu ilkeyi sağlaması ile doğru orantılıdır. Kümeleme analizi sadece veri madenciliğinde değil, örüntü tanıma, görüntü işleme, coğrafi bilgi sistemleri gibi birçok alanda yoğun olarak kullanılmaktadır.

Tez kapsamında da bir kümeleme işlemi gerçekleştirildiği için kümeleme analizi ayrıntılı olarak bölüm 3'de incelenmektedir.

2.6.5. Sıra dışılık analizi

Veri kümesinde, verilerin genel davranışından veya veri dağılım modelinden farklılık gösteren veri nesnelerine sıra dışı (istisna) denir. Birçok veri madenciliği yöntemi sıra dışı noktaları gürültü veya aşırı durumlar olarak görmekte, bu yüzden dikkate almamaktadır. Fakat bazı durumlarda sıra dışı noktalar diğerlerine göre çok daha fazla bilgi içermektedir. Örneğin kredi kartı veya sigorta dolandırıcılıklarının tespitinde, tıp biliminde yeni bir hastalığın başlangıcını belirlemede sıra dışı veriler analiz edilmektedir. Sıra dışılık analizinde iki yöntem bulunmaktadır [4]:

1-) İstatistik tabanlı yöntemler:

Dağılım analizi ya da standart sapma hesabı gibi istatistik yöntemlerle sıra dışı olabilecek noktalar tespit edilmektedir. Fakat çok büyük veri yığınlarında yoğun hesaplama gücü gerektirdikleri için performansları sınırlıdır.

2-) Yoğunluk tabanlı yöntemler:

Bu yöntemde her noktanın çevresindeki komşuları ile olan yakınlığı hesaplanmaktadır. Yakınlık hesaplamada genelde Öklid uzaklığı kullanılsa da veri türüne göre yakınlık hesaplama yöntemi farklılık gösterebilir. Bu yöntemin temel ilkesi "yeterince komşusu olmayan noktaları" tespit etmektir.

2.6.6. Evrimsel analiz

Evrimsel analiz, zamanla davranışları değişen nesnelerin düzenlilik ya da eğilimlerini ortaya çıkarmayı amaçlamaktadır [4]. Evrimsel analiz tanımlama, ayrımlama, birliktelik analizi, sınıflama ve kümeleme yöntemlerini içerse de asıl amacı verinin zaman ile olan ilişkisini ortaya çıkarmaktır. Bunun için zaman serileri, ardışıklık ve periyodiklik örüntüsü bulma, benzerlik analizi gibi yöntemleri kullanmaktadır.

3. KÜMELEME ve K-MEANS ALGORİTMASI

3.1. Giriş

Bu bölümde, veri madenciliği yöntemlerinden olan kümeleme yöntemi ayrıntılı olarak anlatılmakta; kümeleme algoritmalarından olan k-means algoritması ile ilgili temel tanımlara, problemlere ve algoritmanın çalışma ilkesine yer verilmektedir. Bölümün amacı tez kapsamında önerilen ve beşinci bölümde anlatılan İ.F.A.R.T. algoritması ile başlatılacak olan k-means algoritmasına ait alt yapının oluşturulmasıdır. Kümeleme veri madenciliğinde çok geniş bir alt başlık olduğundan bu bölümde sadece tez kapsamında uygulamaya dahil edilmiş olan k-means algoritmasına ayrıntılı olarak yer verilmektedir.

3.2. Kümeleme

Veri madenciliği tekniklerinden olan kümeleme, nesneleri, kayıtları, durumları, verileri benzer gruplara dahil etmeyi hedeflemektedir. Heterojen yapıya sahip büyük veri yığınlarının daha kolay anlaşılabilir, yönetilebilir ve işlenebilir daha küçük homojen alt kümelere ayrılması işlemidir. Bir küme, yer aldığı kümedeki kayıtlara benzer özellikler taşıyan, diğer kümelerdeki kayıtlardan ise farklı özelliklerde olan kayıtlardan oluşmaktadır. Kümeleme, hedeflenen, beklenen ya da daha önceden bilinen bir sonuç olmaması noktasında sınıflandırmadan ayrılmaktadır. Bunun yerine kümeleme, kayıtları homojen kümelere bölmeyi hedeflemektedir. Bu işlem gerçekleştirilirken, kümeler içindeki benzerlik oranının en yüksek derecede; kümeler arasındaki benzerlik oranının ise en düşük derecede olması önemli bir noktadır [43].

Kümeleme, gizli kalmış örüntülerin keşfedilmesini ve büyük boyutlu veri yığınları içerisinden en hızlı şekilde bilgiye erişilmesini sağlayan bir teknik olması nedeni ile veri madenciliğinde çok sık başvurulan tekniklerden bir tanesidir [44].

Sınıflandırma işleminde sınıflar önceden belirli iken kümelemede sınıflar önceden belirli değillerdir. Verilerin hangi gruplara/kümelere, hatta kaç değişik gruba ayrılacağı eldeki verilerin birbirlerine olan benzerliğine göre belirlenmektedir. Belirlenen her bir gruba küme ismi verilmektedir. Küme analizi biyoloji, tıp, antropoloji, pazarlama, ekonomi ve telekomünikasyon gibi birçok ve farklı alanlarda kullanılmaktadır [45].

Kümeleme işlemini anlamı açısından değerlendirmek gerekirse; ortak karakteristik özellikleri taşıyan sınıflar, anlamlı gruplar, insanoğlunun dünyayı anlamasında ve analiz etmesinde önemli bir rol oynamaktadırlar. İnsanoğlu sürekli olarak, nesneleri gruplara bölme (kümeleme) ve belirli özelliklerdeki nesneleri bu gruplara ayırma (sınıflama) eğilimindedirler. Örneğin, bir çocuk bir fotoğrafta yer alan nesneleri binalar, otomobiller, insanlar, hayvanlar ve bitkiler olarak hızlıca etiketleyebilmektedir. Veriyi anlamada, kümeler potansiyel nesne sınıflarıdır ve küme analizi sınıfların otomatik olarak bulunması için olan tekniklere ait olan çalışmadır. Bazı örnekler aşağıda sunulmaktadır [46].

- Biyoloji: Biyologlar, yaşayan canlıları sınıflandırma bilimi için çok uzun seneler harcamışlardır. Yapılmış olan çalışmalar, bu canlıların sınıflandırılması için yaratılan matematiksel modellerin oluşturulmasına ışık tutmuştur. Daha yakın geçmişte, biyologlar kümelemeyi, şu an mevcut olmayan çok miktardaki genetik bilgiyi analiz etmek için uygulamışlardır.

- Bilgi keşfi: W.W.W., milyarlarca web sayfası içermektedir ve bir arama motoruna yapılan bir sorgu binlerce sayfa döndürebilmektedir. Kümeleme bu arama sonuçlarının küçük gruplara ayrılması işleminde kullanılabilmektedir. Örnek olarak, bir "film" sorgusu, şu sınıflara ayrılmış web sayfaları döndürebilmektedir: eleştiriler**,** fragmanlar, yıldızlar, gösterildiği salonlar. Her sınıf alt sınıflara ayrılabilmektedirler. Sorgu sonuçları için hiyerarşik bir yapı oluşturmak kullanıcı için kullanıcının sonuçları anlamasında yardımcı olabilmektedir.
- İklim: Yeryüzünün iklimini anlamak atmosferde ve okyanustaki örüntüleri bulmayı gerektirmektedir. Bu amaçla, küme analizi, kutup bölgelerinin

atmosferik basınçtaki ve iklim için önemli olan okyanus alanlarındaki örüntüleri bulmak için uygulanmaktadır.

- Psikoloji ve ilaç: Bir hastalık çok çeşitli varyasyonlara sahip olabilmektedir ve küme analizi bu farklı alt sınıfların tanımlanmasında kullanılmaktadır. Örneğin, kümeleme, depresyonun farklı tiplerinin tanımlanmasında kullanılmaktadır.

- İş: İş, mevcut ve potansiyel müşteriler üzerinde çok büyük miktarda bilgi toplamaktadır. Kümeleme, müşterileri analiz işlemleri ve market aktivitelerine göre bölümlemede kullanılmaktadır.

Denetimsiz sınıflama olarak da bilinen kümeleme, doküman kümeleme [47], protein dizilerinin kümelenmesi [48], içerik temelli görüntü tanıma [49], görüntü parçalama [50], DNA analizi [51] gibi çok çeşitli alanlarda birçok uygulaması olan başlıca veri madenciliği araçlarından bir tanesidir [52].

Kümeleme işlemini yararlılığı açısından değerlendirmek gerekirse; küme analizi, örgün veri nesnelerinden bu nesnelerin ait olduğu kümelere soyutlama sağlamaktadır. Bazı kümeleme teknikleri her kümeyi belirli bir küme örneğine göre karakterize etmektedir; örneğin bir veri nesnesi kümeyi temsil edebilmektedir. Bu küme örnekleri bir grup veri analizi ve veri işleme tekniği için kaynak olarak kullanılabilmektedir. Bu nedenle küme analizi, kümeleri en iyi şeklide temsil edebilecek olan örnekleri bulma tekniklerine ait olan çalışmadır.

- Özetleme: Birçok veri analiz tekniği, regresyon gibi zaman talebi ve algoritma karmaşıklığı fazla olan tekniklerdir. Bu nedenle, algoritma veri kümesinin bütününe uygulanmak yerine sadece kümelerin prototiplerinden oluşan azaltılmış bir veri kümesine uygulanabilmektedir.

- Sıkıştırma: Küme prototipleri veri sıkıştırma için de kullanılabilmektedirler. Her kümeye ait prototiplerden oluşan bir tablo yaratılır; örneğin her prototip tablodaki pozisyonunu(indeksi) belirtir bir tamsayı ile işaretlenir. Her nesne, küme ile ilişkilendirilmiş olan prototipin indeksi ile temsil edilir. Bu tür sıkıştırmaya

"vektör nicemleme" (vektör kuantizasyonu) denir ve genelde görüntü, ses ve video verilerine uygulanır.

- En yakın komşuyu bulma: En yakın komşuların bulunması yöntemi ile kümeler ve prototipleri çok daha etkin şekilde bulunabilmektedir [46].

3.2.1. Kümelemenin temel adımları

Bir kümeleme işleminde gerçeklenmesi gereken adımlar bulunmaktadır. Bunlar aşağıda özetlenmektedir [12].

- Örüntü seçimi
- Veriler arası benzerliğinin ölçümünde kullanılacak uygun yöntemin seçilmesi
- Kümeleme işlemi
- Sonuçların özetlenmesi ve saklanması (gerekli ise)

1-) Örüntü seçimi

Örüntü seçimi sürecinde, küme sayısının belirlenmesi, örüntü kümesi büyüklüğü, kümeleme algoritmasında kullanılabilecek kayıt niteliklerinin sayıları, tipleri gibi bilgilerin belirlenmesi işlemleri gerçekleştirilmektedir.

2-) Benzerlik yöntemi seçimi

Veri kümelemede örüntü içerisindeki çiftlerin birbirlerine olan benzerliklerinin ya da aykırılıklarının belirlenmesi için bir uzaklık fonksiyonu tanımlanmaktadır. Kaynaklarda farklı uzaklık fonksiyonları kullanılmaktadır [53, 12, 54]. İki nokta arasındaki uzaklığın bulunması için en sık kullanılan yöntem olan Öklid uzaklığı fonksiyonu kullanılabileceği gibi örüntü elemanları üzerinde benzerlikleri bulan başka yöntemler de kullanılabilmektedir [55].

3-) Kümeleme işlemi

Kümeleme temelde iki farklı şekilde gerçekleştirilmektedir. Giriş verisi kesin sınırlarla kümelere ayrılacak şekilde keskin olarak kümelendirilmektedir. Ya da her örüntü elemanının her kümeye ne kadar yakın olduğu belirlenerek bulanık olarak kümelendirilmektedir. Bu süreç kapsamında kümelemede uygulanacak olan algoritma belirlenmekte ve işletilmektedir.

4-) Sonuçların özetlenmesi ve saklanması

Kümeleme sonuçlarının basit ve anlaşılır bir şekilde sunulması aşamasıdır. Kümeleme sonuçları uzman kişiler tarafından özetlenecek ya da bu sonuçlar başka bir algoritma tarafından giriş verisi olarak kullanılmak üzere saklanacaktır. Her kümeyi karakterize eden kuralların bir özeti hazırlanmaktadır. Bunun için örneğin her küme, oluşan kümenin merkezinin özellikleri ile özetlenebilmektedir [54] Ya da kural türetme algoritmaları yardımı ile kümeleri özetleyen kurallar türetilebilmektedir

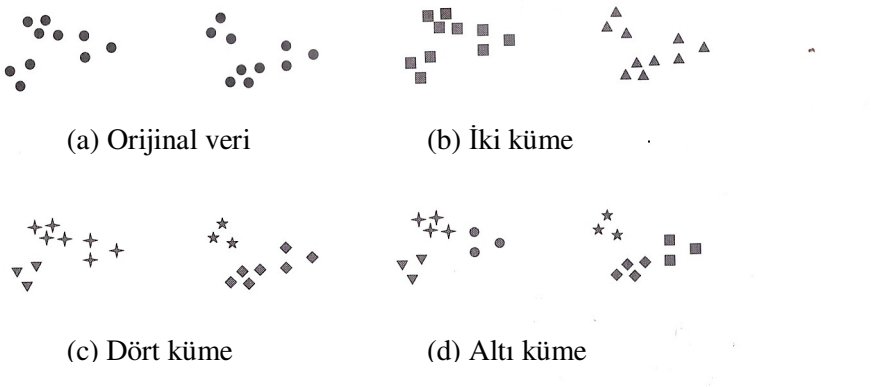
3.3. Kümeleme Analizi

3.3.1. Kümeleme analizi nedir?

Küme analizi, veriyi sadece veri içinde bulunan, veri nesneleri ve aralarındaki ilişkileri tanımlayan bilgiye dayalı olarak gruplamaktadır. Amaç bir grup içindeki nesnelerin birbirlerine yüksek oranda benzerlik göstermeleri, diğer gruplardaki nesnelerden ise mümkün olduğunca farklı olmalarıdır. Küme içindeki benzerlik ya da başka bir deyişle homojenlik ve gruplar arasındaki farklılık arttıkça daha iyi ve belirgin bir kümeleme gerçekleştirilmiş demektir.

Birçok uygulamada, küme kavramı iyi tanımlanmış değildir. Kümeleri belirlemedeki zorluğu anlamak açısından yirmi tane nokta ve bunlar üzerinde gerçekleştirilen üç farklı kümeleme durumunu içeren Şekil 3.1 incelenebilir. İşaretlerin şekilleri kümeye üyelikleri göstermektedir. Şekil 3.1.(b), 3.1.(c) ve 3.1.(d)' de veriler sırasıyla 2, 4 ve 6 kümeye bölünmüştür. Kümeler incelendiğinde, bu şekil, küme tanımının belirsiz

olduğunu ve en iyi küme tanımının da verinin doğasına ve beklenen sonuçlara göre değiştiğini ifade etmektedir [46].



Şekil 3.1: Farklı kümeleme durumları [46]

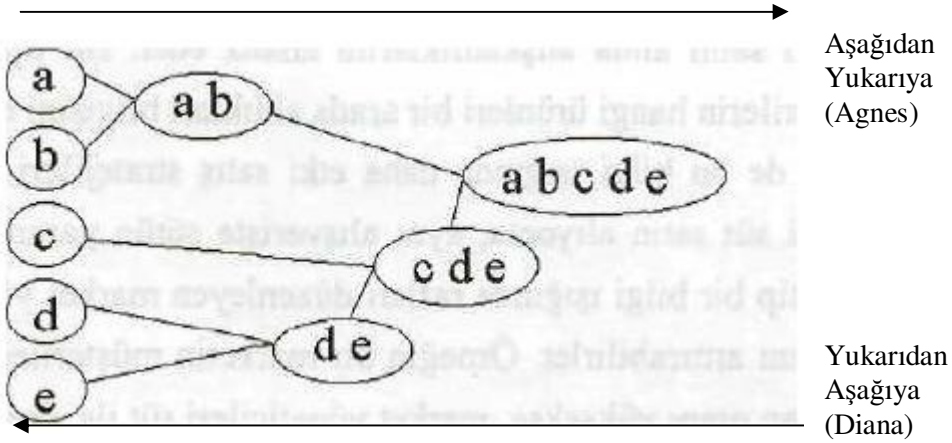
3.3.2. Kümeleme analizinin sınıflandırılması

Kaynaklarda, birçok kümeleme algoritmasının adı geçmektedir. Algoritmalar birbirlerinden, kümelemenin oluşturuluş şekline göre ayrıldıkları gibi, kullanılan veri türüne, yapılacak olan çalışmanın amacına göre de farklılık göstermektedirler. Kümeleme algoritmaları genel olarak, hiyerarşik ve bölümleyici olarak ikiye ayrılırken bu konuda yapılmış bir kaynak taraması bu algoritmaların daha alt bölümlere ayrılabileceğini göstermektedir [56].

- Hiyerarşik yöntemler
    - Toplaşım (agglomerative) kümeleme algoritmaları
    - Bölünür (divisive) kümeleme algoritmaları
- Bölümleyici(partitioning) yöntemler
    - Yer değiştiren algoritmalar
    - Olasılıksal Algoritmalar
    - K-medoid yöntemler
    - K-means yöntemler
    - Yoğunluğa dayalı algoritmalar
    - Yoğunluğa dayalı bağlantılı kümeleme
    - Yoğunluk fonksiyonlu kümeleme

- Grid temelli yöntemler
- Kategorik verinin yinelenmesine dayanan yöntemler
- Kısıtlara dayanan yöntemler
- Makine öğrenmesi alanında kullanılan yöntemler
    - Yapay sinir ağları
    - Ölçeklenebilir kümeleme yöntemleri [37]

Hiyerarşik yöntemler, veri nesnelerini kümeler ağacı seklinde gruplara ayırma esasına dayanmaktadır. Hiyerarşik kümeleme yöntemleri, hiyerarşik ayrışmanın aşağıdan yukarıya veya yukarıdan aşağıya doğru olmasına göre toplaşım (agglomerative) ve bölünür (divisive) hiyerarşik kümeleme olarak sınıflandırılmaktadırlar. Toplaşım hiyerarşik kümeleme Kaufmann ve Rousseuw tarafından 1990 yılında önerilmiştir. Şekil 3.2' de görüldüğü üzere hiyerarşik ayrışma aşağıdan yukarıya doğru olmaktadır. İlk olarak her nesne kendi kümesini oluşturmakta ve ardından bu atomik kümeler birleşerek, tüm nesneler bir kümede toplanıncaya ya da istenen sayıda küme oluşturuluncaya dek daha büyük kümeler oluşturmaktadırlar. Aralarında en az uzaklık bulunan kümeler her adımda birleştirilmektedirler. Bölünür hiyerarşik kümeleme Kaufmann ve Rousseuw tarafından 1990 yılında önerilmiştir. Şekil 3.2' de görüldüğü üzere hiyerarşik ayrışma yukarıdan aşağıya doğru olmaktadır. İlk olarak tüm nesneler bir kümededir ve her nesne tek basına bir küme oluşturana ya da istenen küme sayısı elde edilene dek, kümeler daha küçük parçalara bölünmektedirler [57].



Şekil 3.2: Hiyerarşik kümeleme [57]

Sekil 3.2, bir toplaşım hiyerarşik kümeleme yöntemi olan A.G.N.E.S. (AGlomerative NESting) ve bir bölünür hiyerarşik kümeleme yöntemi olan D.I.A.N.A. (DIvise ANAlysis) uygulaması göstermektedir [13]. Bu yöntemler beş nesneli (a,b,c,d,e) bir veri kümesine uygulanmaktadır. Başlangıçta A.G.N.E.S. her nesneyi bir kümeye yerleştirir. Kümeler, bazı ölçütlere göre basamak-basamak birleşirler. Örneğin C1 ve C2 kümeleri, eğer C1 kümesindeki bir nesne ve C2 kümesindeki bir nesne ile diğer kümelerdeki herhangi iki nesne arasında belirlenen uzaklık mesafesini karşılayacak bir mesafe varsa birleşebilirler. Bu birleşme işlemi tüm nesneler bir kümede toplanıncaya kadar devam etmektedir [1]. D.I.A.N.A.' da ise tüm nesnelerin içinde toplandığı küme, her küme bir nesne içerecek duruma gelene kadar bölünmektedir [13].

Bölme yöntemlerinde, *n* veri tabanındaki nesne sayısı ve *K* oluşturulacak küme sayısı olarak kabul edilmektedir. Bölme algoritması *n* adet nesneyi, *K* adet kümeye böler. Kümeler tarafsız bölme ölçütü olarak nitelendirilen bir ölçüte uygun oluşturulduğu için aynı kümedeki nesneler birbirlerine benzerken, farklı kümedeki nesnelerden farklıdırlar [1]. En iyi bilinen ve en çok kullanılan bölme yöntemleri k-means yöntemi, k-medoids yöntemi ve bunların çeşitli varyasyonlarıdır [58].

3.4. Kümeleme Geçerlilik Analizi

Bulanık kümeleme algoritmaları yaygın bir şekilde sınıflandırma problemlerinde kullanılmaktadırlar. Kümeleme algoritması ile ilişkili olarak sorulabilecek en önemli soru, algoritmanın verinin içinde var olan yapıyı nasıl ve ne ölçüde tanımladığı olacaktır. Bu durum "kümeleme doğruluk problemi" ya da "kümeleme geçerlilik problemi" olarak adlandırılmaktadır [59].

Genel olarak, kümeleme geçerliliği sorunu, belirgin bir sorundur ve birçok soruyu kapsamaktadır. Bu sorular araştırıldığında, çoğunluğun kabul ettiği gibi kümeler arasındaki uzaklık ölçümü ve küme içindeki yakınlık (benzerlik) ölçümü cevap olarak sunulmaktadır [60].

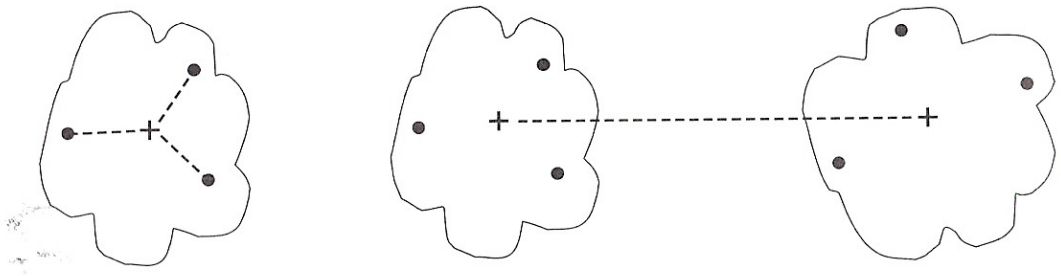
- Aynı küme içinde yer alan veriler mümkün olduğunca birbirlerine yakın veriler olmalıdırlar.
- Ayrı kümelerde yer alan veriler mümkün olduğunca birbirlerinden uzak veriler olmalıdırlar [44].

Bir başka deyişle kümeler, kendi içlerinde yüksek oranda benzerlik gösteren verilerden oluşurlarken, aynı zamanda birbirlerinden uzak oluşturulmalıdırlar. Bu iki kavrama dayanan çeşitli kümeleme geçerlilik göstergeleri önerilmiştir [61]. Ayrıca kümeleme geçerliliğini ölçebilmek için birçok ölçüt geliştirilmiştir [61-66]. Tüm bu ölçütlerin ortak bir hedefi bulunmaktadır: birbirlerinden iyi ayrılmış ve sıkıştırılmış kümeler saptayabilmek [67].

Tez kapsamında, kümeleme geçerliliğini ölçebilmek için bağıntı 3.1' de tanımlanmış olan kümeleme hata oranı *"e"* kullanılmaktadır.

$$e = \frac{D_{İÇ}}{D_{DIŞ}} \tag{3.1}$$

Burada, $D_{İÇ}$ küme içindeki ortalama uzaklık göstergesini; $D_{DIŞ}$ ise kümeler arasındaki ortalama uzaklık göstergesini temsil etmektedir. Parametreler için sırası ile küme içindeki bağlılık ve kümeler arası ayrım ifadeleri kullanılmaktadır ve bu ifadeler Şekil 3.3' de belirtilmektedir.



(a) küme içi bağlılık (b) kümeler arası ayrım

Şekil 3.3: Kümeleme geçerlilik ölçütleri [46]

$D_{İÇ}$ (küme içi bağlılık) ve $D_{DIŞ}$ (kümeler arası ayrım) parametreleri bağıntı 3.2 ve (3.3)' de tanımlanmaktadırlar [68].

$$D_{İÇ} = \frac{1}{K}\sum_{k=1}^{K}\left(\sum_{x_i \in C_k}\frac{d(x_i, C_k)}{elsay(C_k)}\right) \tag{3.2}$$

$$D_{DIŞ} = \frac{2}{K(K-1)}\sum_{\substack{k=1 \\ k'=k+1}}^{K}\left(d(C_k, C_{k'})\right) \tag{3.3}$$

Burada *K*, oluşmuş olan küme sayısını, elsay (C_k) k kümesine ayrılan verilerin sayısını, C_k, k kümesinin merkez noktasını ve $C_{k'}$, k' kümesinin merkez noktasını temsil etmektedir.

$D_{İÇ}$ ile her kümenin elemanlarının küme merkezine olan ortalama uzaklıkları hesaplanmaktadır. $D_{DIŞ}$ ile oluşan her kümenin merkez noktalarının birbirlerine uzaklıkları hesaplanmaktadır. İki nokta arasındaki uzaklık, en yaygın kullanılan uzaklık ölçümü olan Öklid bağıntısı ile hesaplanmaktadır. Bu bağıntı 6.4' deki gibi tanımlanmaktadır.

$$d(x_i, C_k) = \sqrt{\sum_{p=1}^{niteliksayısı}\left[N_p(x_i) - N_p(V_k)\right]^2} \tag{3.4}$$

N_p : örnek veriye ait p. nitelik değerini temsil etmektedir. (p=1,2…, nitelik sayısı)

V_k : k kümesinin merkezidir.

Daha iyi bir kümeleme, daha düşük $D_{İÇ}$ daha yüksek $D_{DIŞ}$ değeri elde edildiğinde, dolayısı ile *"e"* azaldığında gerçekleşmektedir.

3.5. K-Means Algoritması

J. MacQueen tarafından 1967 yılında tanıtılmış olan k-means algoritması, kümeleme algoritmaları içinde en sık kullanılan algoritmalardan bir tanesidir. Algoritma adında yer alan *K*, algoritma çalışmaya başlamadan önce ihtiyaç duyulan sabit küme sayısını ifade etmektedir. Buna göre *K* önceden bilinen, kümeleme işlemi bitene kadar değeri değişmeyen sabit bir pozitif tamsayı olarak küme sayısını göstermekte ve elemanlarının birbirlerine olan yakınlıklarına göre oluşacak grup sayısını ifade etmektedir [39].

K-means algoritmasının uygulamasının kolay olması birçok alanda algoritmayı en sık kullanılan kümeleme algoritması haline getirmiştir. Algoritma bir bölümleyici kümeleme algoritmasıdır. Tekrarlı bölümleyici yapısı ile k-means algoritması, her veri nesnesinin ait olduğu kümeye olan uzaklıkları toplamını küçültmektedir. Sık kullanılıyor olmasının bir başka nedeni ise büyük ölçekli verileri hızlı ve etkin şekilde kümeleyebilmesidir [23].

Açıklamada kolaylık olması açısından, algoritma iki boyutlu diyagramlar kullanılarak örneklenmektedir. Ancak uygulamada çok boyutlu elemanlarla, diğer bir deyişle çok eleman vektörü ile çalışılabilmektedir. Boyut sayısının artması algoritmada değişiklik yapılmasına neden olmamaktadır [39].

Kaynaklarda birçok uygulamada kullanılıyor olmasına rağmen k-means algoritmasına ait dezavantajlar bulunmaktadır. Bunlardan en önemli olanları şu şekilde özetlenmektedir.

- Kümeleme yöntemlerinin birçoğunda olduğu gibi, k-means algoritması da *K* küme sayısının önceden bilindiğini varsaymaktadır. Çok açık olarak gerçek yaşamdaki uygulamalar bu duruma uymamaktadır.
- Tekrarlı bir teknik olarak, k-means algoritması başlangıç noktalarına çok duyarlı bir algoritmadır.
- K-means algoritması yerel minimuma yakalanmaktadır. Algoritmanın çalışması, başlangıç noktasından bitişe kadar olan belirleyici bir plan belirtmektedir [69].

3.5.1. K-Means algoritmasının adımları

İlk adımda küme merkezlerini temsil edecek olan *K* adet rastgele seçilmiş nokta belirlenir. *K* kullanıcı tanımlı bir parametredir. MacQueen [4] algoritmasında küme merkezleri ilk *K* adet elemandan seçilir. Elemanların değerlerinin birbirine çok yakın olduğu durumlarda seçim rastgele yapılır veya birbirinden uzak elemanlar seçilir. Belirlenen bu elemanlar tek elemanlı başlangıç kümeleridir ve ilk küme merkezlerini oluştururlar. Başlangıçta her küme tek elemandan oluşmaktadır.

İkinci adımda, okunan elemanlar kendilerine en yakın *K* adet küme merkezinden birine dahil edilir. Bunu gerçekleştirmenin bir yolu, kümeler arasındaki sınırların belirlenerek elemanların hangi küme merkezlerine daha yakın olduklarını tespit etmektir. 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 dahil edileceği ortaya çıkar.

Noktalar arası uzaklığın hesaplanmasında yaygın olarak çok kullanılan yöntem Öklid bağıntısıdır. İki boyutlu bilgilerde, iki küme merkezinin birleştirilmesinde doğru kullanılırken, boyut sayısı arttığında doğru yerine düzlem kullanılır. Bilgisayar programları ile geliştirilen k-means algoritmalarında ise, düzlemler yerine noktalar arasındaki uzaklıklar hesaplanarak, noktaların merkeze yakınlığı dikkate alınmaktadır.

Üçüncü adımda kümelere eklenen her yeni eleman ile yeni bir küme merkezi bulunmaktadır. Yeni küme merkezleri küme elemanlarının ağırlıklı ortalaması hesaplanarak bulunmaktadır. 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 edecektir. Her bir döngüde elemanlar farklı bir kümeye dahil olabilmektedirler.

Elemanların bir kümeye dahil edilmesi ve küme merkezlerinin tekrar hesaplanması işlemlerine ait döngü, küme sınırlarının değişimi bitene kadar devam etmektedir. K-means algoritması ile uygulamalarda genellikle birkaç düzine döngü sonrası kararlı bir küme grubu ortaya çıkmaktadır. Şekil 3.4' de algoritmaya ait akış şeması görülmektedir.

Şekil 3.4: K-means akış şeması

Algoritmanın akışı aşağıda özetlenmektedir.

Giriş:

- K: oluşturulacak küme sayısı
- C: n elemanlı veri kümesi

Çıkış:

- K adet küme

Yöntem:

1-) Başlangıç küme merkezleri olarak *K* adet elemanı, C kümesinden rastgele seç;

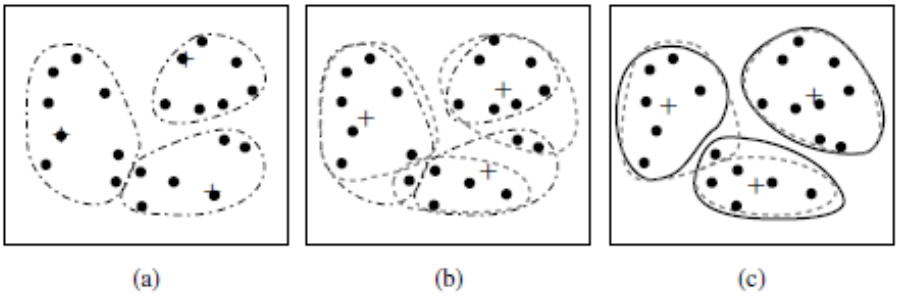
2-) Tekrarla

3-) Her giriş elemanını, kümelerin tüm elemanlarının ortalama değerine göre en çok benzerlik gösterdiği kümeye dahil et;

Her kümenin yeni ortalama değerini (merkez noktasını) hesapla;

4-) Kümeler değişmeyene kadar [1].

Şekil 3.5.(a)' da görülen uzaydaki örnek veriler üzerinde k-means algoritmasının adımları gösterilmektedir. Veriler k=3 adet kümeye ayrılmak istenmektedir. Algoritmaya göre 3 tane rastgele nokta başlangıç küme merkezi olarak seçilmiştir ve "+" sembolü ile işaretlenmiştir. Şekil 3.5. (b)' de görüldüğü gibi uzaydaki her nokta kendisine yakın olan küme merkezine göre bir kümeye dağıtılmıştır. Sonraki adımda oluşan yeni kümelerin merkezleri güncellenmektedir. Her kümenin elemanlarının ortalama değeri hesaplanarak yeni küme merkezleri tespit edilmektedir. Oluşan yeni küme merkezlerine göre veriler yeniden kümelere dağıtılmaktadır. Dağılımdan sonraki veri kümeleri Şekil 3.5.(c)' de görülmektedir. Küme merkezlerinin güncellenmesi işlemi tekrarlı olarak bu noktalar değişmez olana kadar devam etmektedir. Sonuç kümeleri kümeleme işleminden elde edilen çıktılar olmaktadır.



Şekil 3.5: K-means kümeleme örneği [1]

K-means algoritması, karesel hatayı en küçük yaprak olan *K* adet kümeyi tespit etmeye çalışmaktadır. Karesel hata bağıntı 3.5' deki gibi tanımlanmaktadır.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \tag{3.5}$$

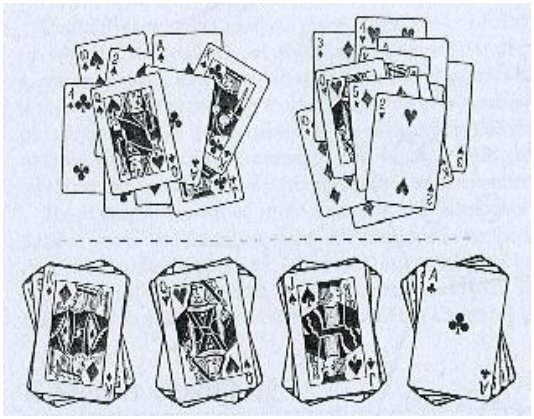
E: veri kümesindeki elemanlara ait toplam karesel hatayı ifade etmektedir. p: veri kümesindeki herhangi bir veri örneği olarak uzaydaki bir noktayı ifade etmektedir. m_i : C_i kümesinin merkezini (orta noktasını) ifade etmektedir (hem p hem de m_i çok boyutludur). Özetle, her kümedeki her eleman için elemanın küme merkezine uzaklığı karesel olarak hesaplanmakta ve bu uzaklıklar toplanmaktadır.

3.5.2. K sabitinin kümelemeye etkisi

K-means algoritmasında K oluşacak olan küme sayısını temsil eden, önceden bilinen ve kümeleme işlemi bitene kadar değeri değişmeyen sabit bir pozitif tamsayıdır. K-means tipi algoritmalarda en önemli parametre küme sayısını temsil eden parametredir. *K* kullanıcı tanımlı bir parametredir ve belirlenmesi zor bir parametredir. Uygulamalarda farklı *K* değerleri ile algoritma çalıştırılmakta ve sonuçlar kümeleme doğruluk analizi yöntemleri ile sınanmaktadır. Bu amaçla çok farklı doğruluk ve geçerlilik analizi indisleri önerilmiştir [70].

Şekil 3.6' da 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 oluşmasını sağlamaktadır. Hangisinin daha etkili olduğu hangi kümelemenin kullanılacağına bağlıdır.



Şekil 3.6: Oyun kağıtlarının k=2 ve k=4 için kümelenmesi [39]

K-means algoritması *K* sayısının belirlenmesi konusunda bir çözüm sunmamaktadır. Birçok durumda, özel bir *K* değerinin belirlenmesi gerekmemektedir. *K* sayısı için tahmini bir değer kullanılarak kümeleme algoritması çalıştırılmakta ve alınan sonuçlar değerlendirilmektedir. Değerlendirme sonucunda beklenen kümeleme görülmez ise, başka bir *K* değeri ile algoritma çalıştırılmakta veya veriler üzerinde değişiklik yapılabilmektedir. Algoritmanın her yeni işletimi sonrasında, ortaya çıkan kümelerin etkinliğini hesaplamak için, küme içindeki elemanların arasındaki ortalama uzaklık ile küme merkezleri arasındaki ortalama uzaklık değeri hesaplanarak karşılaştırılmaktadır.

K adet başlangıç küme merkezlerinin rastgele seçimi karşısında algoritma çok farklı küme sonuçları oluşturabilmektedir. K-means algoritması, *K* sayısına ve *K* adet seçilen başlangıç küme merkezine bağlı olarak çok farklı küme sonuçları oluşturabildiği için kararlı olmayan yapıda çalışan bir algoritmadır.

3.5.3. Biçimsel benzerlik ölçümleri

İki kayıt arasındaki benzerlik oranını hesaplamak için geliştirilmiş olan birçok farklı yaklaşım bulunmaktadır. Bunlardan bazıları, metin içindeki paragrafların

karşılaştırılması gibi özel amaçlı uygulamalar içindir. Diğerleri ise özellikle ikili ya da kategorik veriler için tasarlanmıştır [39].

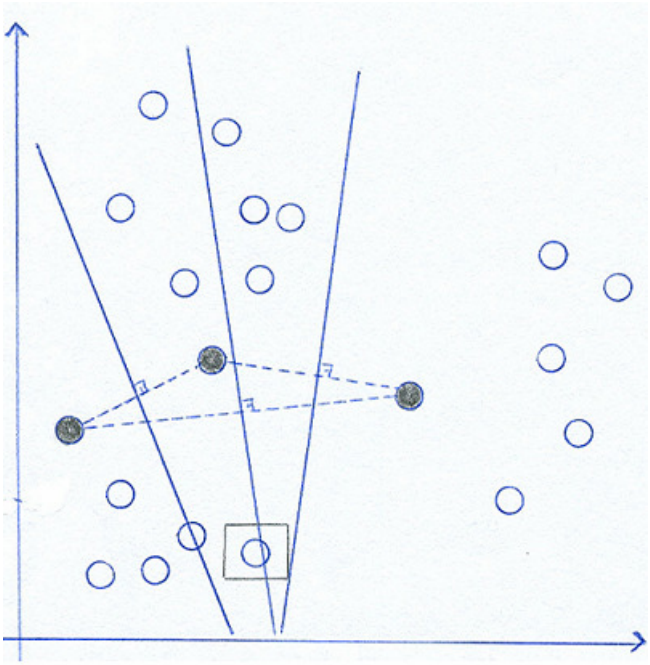
3.5.3.1. İki nokta arasındaki geometrik uzaklık

Örnek veri kayıtlarına ait nitelik alanları sayısal değerleri olduklarında, kayıtlar n boyutlu uzayda bir nokta ile temsil edilirler. Bu noktalardan iki tanesi arasındaki uzaklık, iki veri kaydı arasındaki benzerlik oranını vermektedir. Eğer noktalar birbirlerine yakın ise, bu durumda kayıtlar birbirlerine benzer kayıtlardır. Noktalar arası uzaklığın hesaplanmasında en sık kullanılan yöntem Öklid bağıntısıdır. p adet niteliği olan i ve j nesneleri arasındaki Öklid uzaklığı aşağıdaki bağıntıya göre hesaplanmaktadır.

$$d(i,j) = \sqrt{\left|x_{i1} - x_{j1}\right|^2 + \left|x_{i2} - x_{j2}\right|^2 + ... + \left|x_{ip} - x_{jp}\right|^2} \tag{3.6}$$

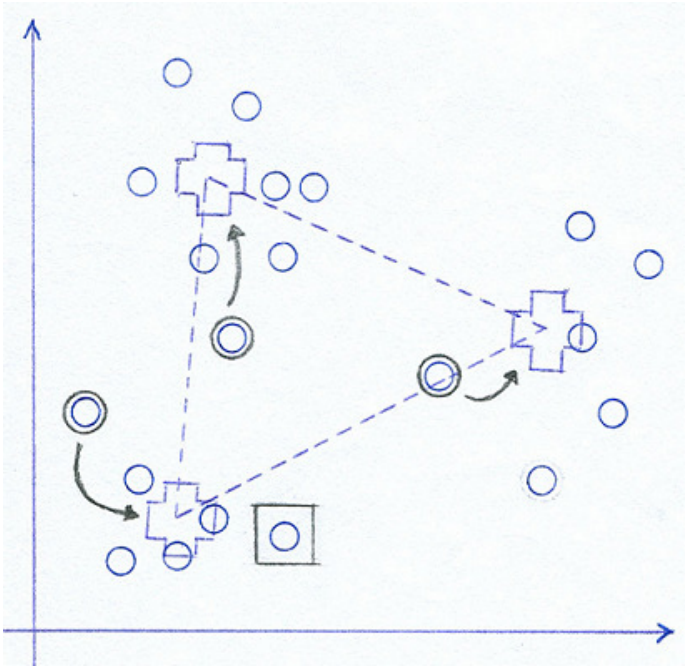
Geometrik hesaplama yöntemi, 20 elemanın üç kümeye bölünmesi örneği üzerinden aşağıdaki şekiller ile anlatılmaktadır.

İlk adımda başlangıç küme merkezlerini temsil edecek 3 nokta seçilmektedir. Şekil 3.7' de içi dolu çember sembolleri ile işaretlenen noktalar ilk küme merkezleridir. İkinci adımda, diğer bütün elemanları yakın oldukları bir küme merkezine dahil etmek için her bir kümenin sınırı belirlenmektedir. Küme merkezlerinden eşit uzaklıkta bulunan noktalar küme sınırını oluşturmaktadır. Bu noktaların belirlenebilmesi için küme merkezleri birer doğruyla birleştirilmektedir. Doğruların orta noktalarından geçen ve doğruları dik kesen başka doğrular çizildiğinde, kümelerin sınırları ortaya çıkarılmaktadır. Şekil 3.7' de ilk küme merkezleri noktalı çizgiler ile birleştirilmiş durumdadır. Bunları dik kesen düz çizgiler küme sınırlarıdır. Kümelerin sınır çizgilerine bakılarak diğer elemanların hangi kümeye dahil edileceği görülmektedir.



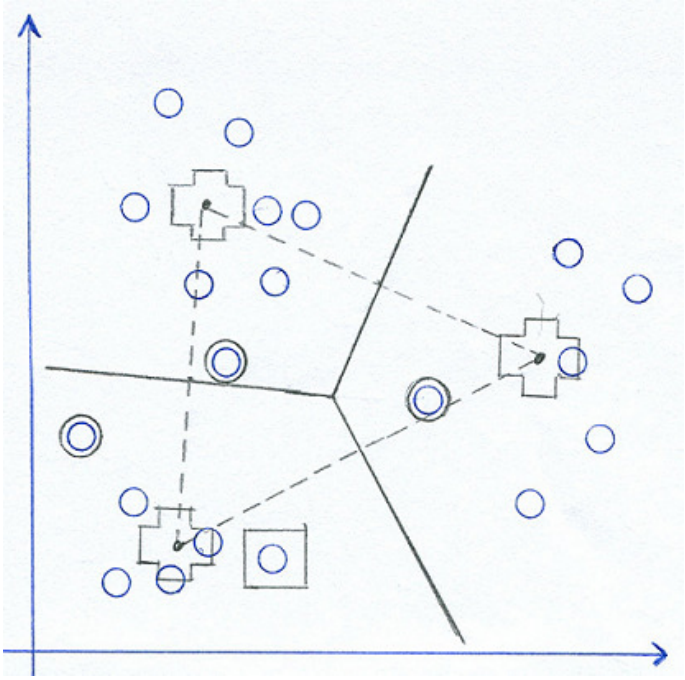
Şekil 3.7: Geometrik hesaplama yöntemiyle ilk kümelerin belirlenmesi [39]

Üçüncü adımda, her bir kümedeki elemanların ortalama değerleri alınarak yeni küme merkezleri hesaplanmaktadır. Şekil 3.8' de yeni küme merkezleri artı sembolünün içinde yer almaktadır. Önceki küme merkezlerinin çevresine çember çizilerek, nereden nereye taşındıkları gösterilmektedir. Şekil 3.8' de kare içine alınarak işaretlenmiş olan eleman, ilk döngü sonrasında ikinci kümeye dahil olmuştur. İkinci döngü sonrasında ise Şekil 3.8'de görüldüğü gibi birinci kümeye dahil edilmiştir.



Şekil 3.8: Noktaların kümelere dahil edilmesi sonrasında yeni küme merkezleri [39]

Döngü içinde ikinci adım tekrarlandıktan sonra, bütün elemanlar en yakın oldukları küme merkezlerine yeniden dahil edilmektedirler. Her döngüde küme merkezlerinin değişimiyle küme sınırları da değişmektedir. Şekil 3.9' da yeni kümelerin sınırları gösterilmektedir. Bahsedilen bu döngü, kümelerin sınırları değişmez olana kadar sürmektedir.



Şekil 3.9: Her döngü sonrasında küme sınırları değişmektedir [39]

3.5.3.2. Manhattan uzaklığı

Manhattan uzaklığı boyutlar arasındaki ortalama farka eşittir. Bu ölçüt kullanıldığında farkın karesi alınmadığı için sıra dışılıkların etkisi azalmaktadır [71]. p adet niteliği olan i ve j nesneleri arasındaki Manhattan uzaklığı aşağıdaki bağıntıya göre hesaplanmaktadır.

$$d(i,j) = q\sqrt{\left|x_{i1} - x_{j1}\right|^q + \left|x_{i2} - x_{j2}\right|^q + \ldots + \left|x_{ip} - x_{jp}\right|^q} \tag{3.7}$$

Burada q=1 dir.

3.5.3.3. Chebychev uzaklığı

Chebychev uzaklığı iki nesne arasındaki mutlak maksimum uzaklığa eşittir. Chebychev uzaklığının bağıntısı aşağıda görülmektedir [72].

$$\text{d (x,y)= } \max\left|x_i - y_i\right| \tag{3.8}$$

3.5.4. K-means algoritması için başlangıç noktalarını belirlemek

Giriş bölümünde de bahsedildiği üzere, k-means algoritmasına başlangıç noktalarının atanmasında birçok farklı yöntem önerilmiştir. Kaynaklarda, geliştirilmiş olan k-means algoritması başlangıç yöntemleri esas olarak üç grupta toplanmaktadırlar. Bunlar, rastgele örnekleme, uzaklık optimizasyonu ve yoğunluk kestirim yöntemleri olarak isimlendirilmektedirler [44].

3.5.4.1. Rastgele örnekleme yöntemleri:

Rastgele örnekleme yöntemleri başlangıç küme merkezlerini tüm veri içinden rastgele belirleyen ve kaynaklarda en sık kullanılan yöntemlerdir [8]. Bu yöntemlerde, örnek veri kümesi içinden rastgele seçilmiş olan veri çiftleri, başlangıç kümelerinin merkezleri olarak belirlenmektedir. Yöntemlerden R-SEL (Random Selection) ve R-MEAN (Random Mean) [72] aşağıda açıklanmaktadır.

R-SEL:

i=1,…,K için $c_i = x_r$; r =Rand(1, N)

N: toplam giriş sayısı

Rand (min, max): r∈ [min, max] olacak şekilde rastgele bir r değeri üreteci.

R-MEAN:

i=1,…,K için $c_i = gaussRand\left(\bar{x}, \varepsilon\right)$

gaussRand(m,v): m ortalaması ve v değişintisi ile rastgele değer üreteci.

$\bar{x} = \frac{1}{N}\sum_{j=1}^{N} x_j$ girişlerin ortalama değeri, ε : küçük bir sabit.

3.5.4.2. Uzaklık optimizasyonu yöntemleri:

Kümeleme algoritmaları küme merkezleri arasındaki olası maksimum uzaklığı hedeflerken, küme içinde ise merkezlere minimum uzaklığı hedeflemektedirler. O halde doğaldır ki küme merkezleri arasındaki uzaklığı optimize etmek düşünülecek bir çözümdür. Optimizasyon işlemi kümeleme etkinliğini artıran bir etkendir.

İlk yöntem, S.C.S. (Simple Cluster Seeking) [73] olarak adlandırılmıştır.

S.C.S.:

1-) İlk küme merkezini, ilk giriş verisi olarak belirle: $c_1 = x_1$;

2-) j=2,…,N, tüm başlangıç c_k kümeleri için eğer $\|x_j - c_k\| > \rho$ ise x_j'yi yeni başlangıç kümesi olarak ekle. *K* tane küme başlangıç kümeleri olarak belirlendiğinde dur. ρ :eşik değeri.

3-) Eğer tüm giriş çiftlerini taradıktan sonra, *K*'dan daha az küme oluşturulmuş ise eşik değerini (ρ) düşür 1 ve 2. adımları tekrarla.

İkinci yöntem K.K.Z. (Katsavaounidis Kuo Zhang) [74] olarak adlandırılmıştır.

K.K.Z.:

1-) İlk başlangıç kümesini maksimum normunu kullanarak oluştur. $c_1 \equiv \arg\max\{\|x_j\|\}$;

2-) i=2,…,K için her c_i şu şekilde oluşturulmuştur: her giriş x_j verisi için en yakın başlangıç kümesine olan uzaklığını hesapla. $d_j = \min\{\|x_j - c_k\| : \text{tüm mevcut } c_k \text{ lar için}\}$. Eğer en büyük d_j değerine sahip giriş verisi j. giriş ise o zaman $c_i = x_j$.

3.5.4.3. Yoğunluk kestirim yöntemleri:

Yoğunluk kestirim yöntemleri, genelde giriş verilerinin Gauss dağılımında olduğunu göz önüne almaktadır. O halde başlangıç küme merkezleri olarak giriş verilerinin en yoğun olduğu alanlardaki noktalardan seçilebilmektedir. Bu seçilen noktalar daha yoğun küme yaratma konusunda kümeleme algoritmasına yardım etmektedirler. K.R. (Kauffman Rousseuw) [75] olarak adlandırılmış bir yönteme ait adımlar aşağıda özetlenmektedir.

K.R.:

1-) En merkezde konumlanmış olan nokta ile ilk küme merkezi c_1'i başlat.

2-) i=2,…, K için her c_i şu şekilde oluşturulmaktadır: her bir seçilmemiş x_j giriş verisi için diğer seçilmemiş girişlere (x_i) ortalama uzaklığını hesapla. $s_j = \sum_{l \neq j} \max\left(\min\left\{\|x_l - c_k\|\right\} - \|x_l - x_j\|, 0\right)$, Eğer en büyük s_j değerine sahip giriş verisi j. giriş ise $c_i = x_j$.

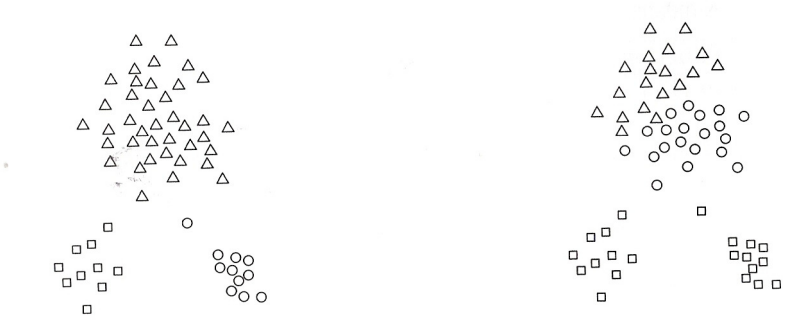
Bu yöntemlere ek olarak, birçok kümeleme algoritması, başka kümeleme algoritmaları için başlangıç konumlarını belirleyen bir algoritma olabilmektedir. Tüm bu başlangıç algoritmaları kümelemenin kalitesini ve performansını artırmaktadırlar [8].

Önerilmiş olan bu yöntemlere rağmen, k-means algoritması sadece başlangıç kümeleri sonuçta elde edilecek olan kümelere yakın olduklarında daha iyi sonuç verebilmektedir [12].

3.5.5. K-means algoritması için başlangıç noktalarını rastgele belirlemek

K-means algoritması küme merkezlerinin rastgele seçimi ile başladığında, farklı işletimler farklı kümeleme hataları üretmektedirler. Bu durum doğal olarak üç

kümeden oluşan noktalar ile Şekil 3.10' da gösterilmektedir. Şekil 3.10(a) en iyi kümeleme durumunu, Şekil 3.10(b) ise yerel minimuma sahip olan bir idealin altında olan bir kümeleme durumunu göstermektedir.

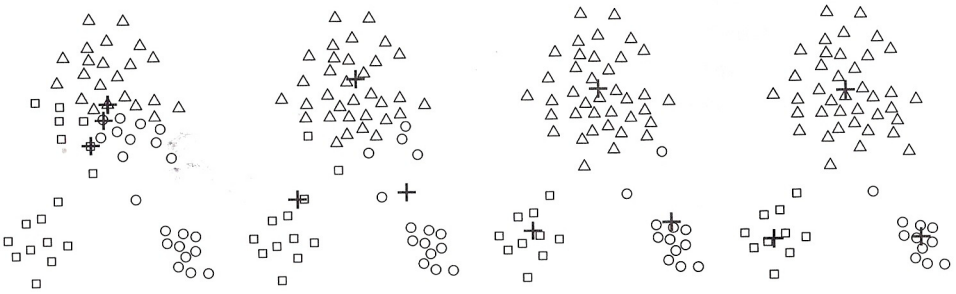


(a) En uygun kümeleme (b) En uygun olmayan kümeleme

Şekil 3.10: Üç adet ideal ve ideal olmayan küme [46]

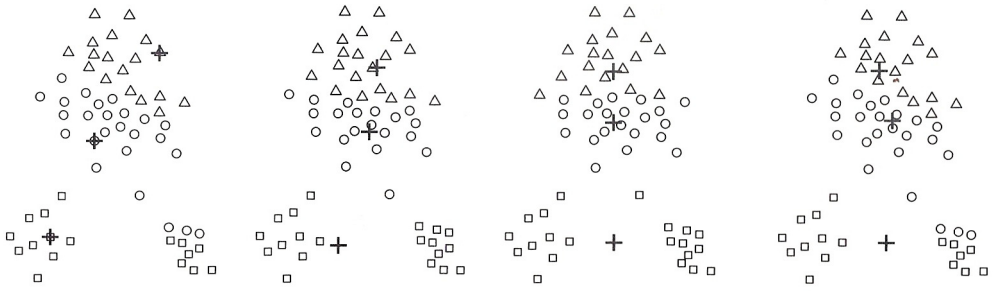
Uygun başlangıç noktalarını seçmek k-means algoritması için anahtar adımdır. En yaygın yaklaşım küme merkezlerinin rastgele seçilmesidir. Ancak rastgele başlangıç ile sonuç kümeleri genelde düşük kalitede oluşmaktadırlar [46].

1-) Örnek (Düşük kaliteli küme merkezleri): Rastgele seçilmiş olan küme merkezleri düşük kalitede olabilirler. Şekil 3.11 ve 3.12'de küme merkezlerinin iki farklı seçim durumu ile başlatılan algoritmalardan elde edilen kümeler gösterilmektedir. Şekil 3.12' de küme merkezleri daha iyi dağıtılmış şekilde görünse de, Şekil 3.11'de gerçekleştirilmiş olan kümeleme daha yüksek kalitededir.



(a) Tekrar 1 (b) Tekrar 2 (c) Tekrar 3 (d) Tekrar 4

Şekil 3.11: Örnek veri kümesinden üç kümeyi bulmak üzere k-means algoritmasının adımları [46]

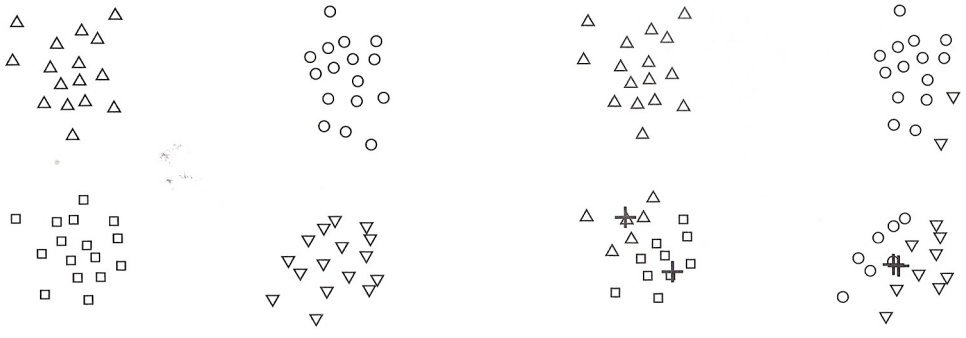


(a) Tekrar 1 (b) Tekrar 2 (c) Tekrar 3 (d) Tekrar 4

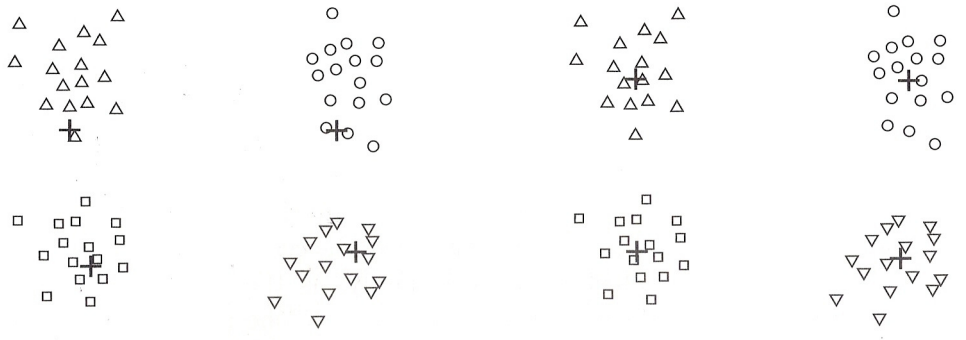
Şekil 3.12: Yetkin olmayan başlangıç noktaları ile başlatılan k-means algoritması adımları [46]

2-) Örnek (Rastgele başlangıcın sınırları): Rastgele seçilen başlangıç noktaları ile çalıştırılan k-means algoritmasına ait problemleri çözmede yaygın olarak kullanılan teknik, algoritmayı birçok kez çalıştırmaktır. Her çalışma sonucunda farklı kümeler elde edilmektedir. Bunlar arasında hata oranı en düşük olan kümeleme seçilmektedir.

Bu yöntem Şekil 3.13 (a)' da gösterilen veri kümesi üzerinde uygulanmaktadır. Veri kümelerin iki çiftinden oluşmaktadır. Her çiftteki kümeler (üstte ve alttaki) birbirlerine diğerlerine olduklarından daha yakındırlar. Her kümede bir başlangıç noktası ile başlamak yerine, her küme çiftindeki iki başlangıç noktası ile başlandığında; küme merkezlerinin kendilerini yeniden dağıtacağı ve daha doğru kümeler elde edileceği Şekil 3.13 (b-d)'de gösterilmektedir. Bununla birlikte, Şekil 3.14' de görüldüğü üzere, eğer bir çift küme tek bir başlangıç noktasına sahip olursa ve diğer kalan kümeler üç taneye sahip olursa; o zaman doğru kümelerden ikisi birleşecek ve doğru olan bir küme de bölünecektir.

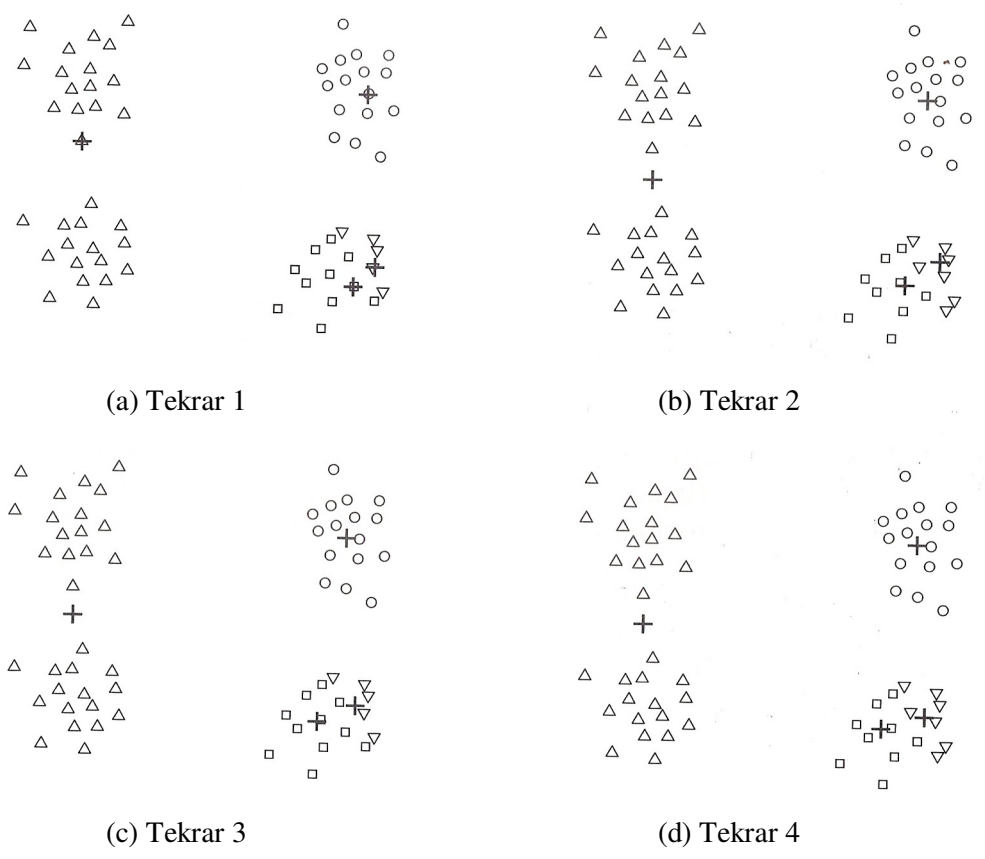


(a) Başlangıç noktaları (b) Tekrar 1



(c) Tekrar 3 (d) Tekrar 4

Şekil 3.13: Bir çift başlangıç noktasının iki ayrı kümede yer alması [46]



(a) Tekrar 1 (b) Tekrar 2

(c) Tekrar 3 (d) Tekrar 4

Şekil 3.14: Bir çift ya da daha az başlangıç noktasının farklı kümelerde yer alması [46]

Şekil 3.13 ve Şekil 3.14' de oluşan kümelemede de görüldüğü üzere en uygun kümeleme, iki başlangıç noktası bir küme çiftinde herhangi bir yere düştüğü sürece sağlanmış olacaktır. Ne yazık ki, küme sayısı arttıkça, her küme çifti yalnızca bir başlangıç noktasına sahip olacaktır. Bu durumda, k-means algoritması başlangıç noktalarını diğer küme çiftleri arasında yeniden dağıtamayacak ve böylece bir yerel minimuma ulaşılacaktır.

Rastgele seçilmiş başlangıç noktaları nedeniyle ortaya çıkan ve algoritmanın çok kez çalıştırılmasına rağmen üstesinden gelinemeyen problemler yüzünden küme merkezlerinin seçilmesi işleminde başka teknikler uygulanmaya başlamıştır. Etkin bir yöntem, örnek noktaları seçip onları hiyerarşik kümeleme tekniği ile kümelemektir. *K* adet küme hiyerarşik kümelemeden elde edilmektedir ve bu kümelerin merkezleri başlangıç küme merkezleri olarak belirlenmektedir. Bu yöntem ancak veri kümesinin küçük olduğu durumlarda kabul edilebilir bir çözüm yöntemidir.

Başlangıç noktalarının belirlenmesinde bir başka çözüm yöntemi de şu şekildedir; ilk başlangıç noktası rastgele seçilmekte ya da tüm noktaların orta noktası bulunmaktadır ve bu nokta ilk başlangıç noktası olarak seçilmektedir. Sonra seçilen her başlangıç noktasına en uzak olan nokta seçilmektedir. Bu şekilde bir grup başlangıç noktası elde edilmektedir. Bu başlangıç noktaları sadece rastgele seçilen başlangıç noktaları olmayacaklar aynı zamanda iyi ayrılmış noktalar olacaklardır. Yalnız bu yönteme ait bir dezavantaj sıra dışı verilerin başlangıç noktası olarak seçilebilecekleri ihtimalidir. Aynı zamanda en uzak noktanın hesaplanmasının maliyeti de yüksektir. Bu nedenden dolayı, yöntem, sıra dışı verilerin dışında bırakıldığı bir grup alt veri üzerinde uygulanmaktadır [46].

4. YAPAY SİNİR AĞLARI

4.1. Giriş

Bu bölümde, yapay sinir ağlarının genel özelliklerinden, çalışma ilkesinden bahsedilmektedir. Ayrıca tez kapsamında önerilen algoritma ile ilgili olarak adaptif rezonans teorisi (Adaptive Resonance Theory-A.R.T.), bulanık A.R.T. ve önerilen yöntemle karşılaştırılan bir başka yapay sinir ağı modeli olan S.O.M. (Self Organizing Map) modelinden ve algoritmaların çalışma şekillerinden bahsedilmektedir.

4.2 Yapay Sinir Ağları

Yapay sinir ağları (Y.S.A.) üzerinde ilk çalışmanın 1943 yılında başladığı kabul edilir. McCulloch ve Pitts, 1943 [76] yılında ilk olarak yapay sinir tanımını yaparak hücre modelini geliştirmişlerdir.

Y.S.A., sahip olduğu özelliklerden dolayı alışılagelmiş bilgi işleme yöntemlerinden farklılıklar göstermektedir. Hatta sahip olduğu bazı özellikler bakımından birçok yönteme göre daha sağlıklı sonuçlar vermektedir. Bu özelliklerden bazıları paralellik, hata toleransı, öğrenebilirlik ve gerçekleme kolaylığı olarak tanımlanabilir [77].

Bir yapay sinir ağı girişlere bağlı olarak çıkış bilgisi veren, bir graf topolojisi ile çalışan, paralel ve dinamik bir sistemi temsil etmektedir. Ağa ait giriş ve çıkış birimleri ile bu birimleri birbirine bağlayan kanallar yapay sinir ağına ait düğümler olarak düşünülmektedir.

Genel anlamda Y.S.A., beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanmaktadır. Y.S.A., yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşmaktadır ve genellikle katmanlar

şeklinde düzenlenmektedir. Y.S.A., bir öğrenme sürecinden sonra bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış bir işlemci gibi çalışmaktadır.

Yapay sinir ağları, insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, oluşturabilme ve keşfedebilme gibi yetenekleri her hangi bir yardım almadan otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar sistemleridir. Bu yetenekleri geleneksel programlama yöntemleri ile gerçekleştirmek oldukça zor veya mümkün değildir. O nedenle, yapay sinir ağlarının, programlanması çok zor veya mümkün olmayan olaylar için geliştirilmiş, adaptif bilgi işleme ile ilgilenen bir bilgisayar bilim dalı olduğu söylenebilir [78]. İnsan beyninin fonksiyonel özelliklerine benzer şekilde öğrenme, sınıflandırma, kümeleme, genelleme ve optimizasyon konularında bir çok farklı algoritma ile başarılı şekilde uygulanmaktadır.

Kaynaklarda temel Y.S.A. yapıları ile ilgili olarak gerçekleştirilen çalışmalar kronolojik olarak şu şekilde listelenebilir:

1940'lar: Warren McCulloch ve Walter Pitts ilk yapay sinir ağı olarak kabul edilen sistemi tasarlamışlardır.
1943 – Yapay sinir hücrelerine dayalı hesaplama teorisinin ortaya atılması ve eşik değerli mantıksal devrelerin geliştirilmesi
1949 –Biyolojik olarak mümkün olabilen öğrenme prosedürünün bilgisayarlar tarafından gerçekleştirilecek biçimde geliştirilmesi gerçekleştirilmiştir. Donald Hebb yapay sinir ağları için ilk öğrenme kuralını öne sürmüştür.
1956 – 1962 – ADALINE ve Widrow öğrenme algoritmasının geliştirilmesi
1957 – 1962 – Tek katmanlı algılayıcının geliştirilmesi
1965 – İlk makine öğrenmesi kitabının yayınlanması
1967 – 1969 – Bazı gelişmiş öğrenme algoritmalarının (Grosberg öğrenme algoritması gibi) geliştirilmesi
1969 – Tek katmanlı algılayıcıların problemleri çözme yeteneklerinin olmadığının gösterilmesi

1969 – DARPA' nın yapay sinir ağlarını desteklemeyi durdurup diğer yapay zeka çalışmalarına destek vermesi
1969 – 1972 – Doğrusal ilişkilendiricilerin geliştirilmesi
1972 – Korelasyon matris belleğinin geliştirilmesi
1974 – Geriye yayılım modelinin (çok katmanlı algılayıcıların ilk çalışmalarının) geliştirilmesi
1978 – 1982 – Öğretmensiz öğrenmenin geliştirilmesi
1978 – A.R.T. modelinin geliştirilmesi
1982 – Kohonen öğrenmesi ve S.O.M. modellemenin geliştirilmesi
1982 – Hopfield ağlarının geliştirilmesi
1982 – Çok katmanlı algılayıcının geliştirilmesi
1984 – Boltzman makinesinin geliştirilmesi
1985 – Çok katmanlı algılayıcıların (genelleştirilmiş Delta öğrenme kuralı ile)
1988 – R.B.F. modelinin geliştirilmesi
1988 – P.N.N. modelinin geliştirilmesi
1991 – G.R.N.N. modelinin geliştirilmesi
1991 – Bulanık A.R.T. modelinin geliştirilmesi
1991 yılından sonra geliştirilen modeller değişik alanlarda uygulanmış ve bu modeller için yazılımlar geliştirilmiştir [78].

4.2.1. Yapay sinir ağlarının özellikleri

Paralel dağılmış yapısı, öğrenebilme ve genelleme yapma yeteneği gibi özellikleri nedeni ile Y.S.A. karmaşık ve matematiksel olarak modellenemeyen problemleri çözebilme yeteneğine sahiptir. Örüntü tanıma, sınıflandırma, işaret işleme, sistem tanımlama, sistem kontrolü gibi birçok alanda başarılı şekilde uygulanmıştır ve uygulanmaktadır. Yapay sinir ağı özellikleri aşağıdaki gibi özetlenmektedir.

Doğrusal olmaması: Y.S.A.' nın benzetildiği sinir ağının temel işlem elemanı olan hücre doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen yapay sinir ağı da doğrusal değildir. Bu özelliği ile Y.S.A., doğrusal olmayan karmaşık problemlere çözüm getiren bir araç haline gelmiştir.

Öğrenme: üzerinde çalışılan problem için giriş-çıkış ilişkisini tanımlayacak en uygun ağırlık değerlerinin bulunması süreci ağın öğrenmesi olarak ifade edilmektedir.
Genelleme: Y.S.A., ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı giriş verileri için de doğru çıkış değeri üretebilmektedir. Veri Y.S.A.' ya eksik, bozuk verilse bile ağ kabul edilebilir en uygun çıkışı üretecektir.

Uyarlanabilirlik: Belirli bir problemi çözmek amacıyla eğitilen Y.S.A., problemdeki değişimlere göre tekrar eğitilebilir, değişimler devamlı ise gerçek zamanda da eğitime devam edilebilmektedir.

Hata Toleransı: Eğitilmiş bir Y.S.A.'nın bilgisi ağın tüm ağırlıkları üzerine dağılmış durumdadır. Ağın bazı ağırlıklarının hatta bazı hücrelerinin etkisiz hale gelmesi, doğru bilgi üretmesini önemli ölçüde etkilememektedir. Bu nedenle de, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir.

Donanım ve Hız: Y.S.A., paralel yapısı nedeniyle büyük ölçekli entegre devre (V.L.S.I.) teknolojisi ile gerçeklenebilmektedir. Bu özelliği, Y.S.A.' nın hızlı bilgi işleme yeteneğini artırmakta ve gerçek zamanlı uygulamalarda tercih edilmesini sağlamaktadır.

4.2.2. Yapay sinir ağlarının uygulama alanları

Y.S.A. modelleri çözümü karmaşık ve güç olan birçok alanda uygulanmış ve genelde başarılı sonuçlar alınabilmiştir. Uygulama alanları çok geniş bir yelpazede değişmektedir. Burada 6 grup ile özetlenmektedir.

- Arıza analizi ve tespiti: Bir sistemin, cihazın ya da elemanın düzenli (doğru) çalışma şeklini öğrenen bir Y.S.A. yardımıyla bu sistemlerde meydana gelebilecek arızaların tanımlanma olanağı vardır. Bu amaçla Y.S.A.; elektrik makinelerinin, uçakların yada bileşenlerinin, entegre devrelerin arıza analizinde kullanılmaktadır.

- Tıp alanında: EEG ve ECG gibi tıbbi sinyallerin analizi, kanserli hücrelerin analizi, protez tasarımı ve hastanelerde giderlerin optimizasyonu gibi konularda uygulama yeri bulmuştur.
- Savunma sanayi: Silahların otomasyonu ve hedef izleme, nesneleri/görüntüleri ayırma ve tanıma, yeni algılayıcı tasarımı ve gürültü önleme gibi alanlara uygulanmaktadır.

- Haberleşme: Görüntü ve veri sıkıştırma, otomatik bilgi sunma servisleri, konuşmaların gerçek zamanda çevirisi v.s gibi alanlarda uygulama örnekleri vardır.

- Üretim: Üretim sistemlerinin optimizasyonu, ürün analizi ve tasarımı, ürünlerin (entegre, kağıt, kaynak v.s.) kalite analizi ve kontrolü, planlama ve yönetim analizi gibi alanlarına uygulanmaktadır.

- Otomasyon ve kontrol: Uçaklarda otomatik pilot sistemi otomasyonu, ulaşım araçlarında otomatik yol bulma/gösterme, robot sistemlerinin kontrolü, doğrusal olmayan sistemlerin modelleme ve kontrolü gibi alanlarda uygulanmaktadır [36].

4.2.3. Yapay sinir ağlarının temel çalışma ilkesi

Y.S.A.'da genel olarak 3 katmanlı bir çalışma şeklinden bahsedilmektedir.

- Giriş katmanında, giriş elemanları bilgileri alarak ara katmanlara iletmekten sorumludurlar.

- Ara katmanlarda, giriş katmanından gelen bilgi işlenerek çıkış katmanına iletilmektedir. Bilgi işleme işi ara katmanlarda gerçekleştirilmektedir.

- Çıkış katmanında, çıkış elemanları ara katmanlardan gelen bilgileri işleyerek, ağa giriş olarak sunulan veriler için üretmesi gereken çıkış değerlerini üretmektedirler.

Yapay sinir ağlarının çalıma şekli en genel hali ile Şekil 4.1' deki gibi ifade edilebilir. Y.S.A., kendisine sunulan giriş verisini, beklenen ya da önceden bilinmeyen çıkış verisine dönüştürmektedir. Bunun için ağın kendisine sunulan girdiler için eğitilmesi gerekmektedir. Ağ giriş ve çıkış verileri olarak sayısal verileri beklemektedir. Ağa sunulacak örnekler öncelikle bir vektör haline getirilmektedir. Bu vektör ağa sunulmakta ve ağ bu vektör için gerekli çıktı vektörünü üretmektedir. Y.S.A.' nın parametre değerleri doğru çıkışı üretecek şekilde düzenlenmektedir. Giriş ve çıkış vektörlerinin tasarımı ağı geliştiren kişi tarafından belirlenmekte ve örnekler belirlenen formata dönüştürülerek eğitim sırasında ağa gösterilmektedirler.

Girdi Vektörü: $X = (x_1, x_2, x_3, \ldots, x_n)$

Çıktı Vektörü: $Y = (y_1, y_2, y_3, \ldots, y_m)$

Şekil 4.1: Yapay sinir ağı girdi, çıktı ilişkisi

Bir Y.S.A.' nın eğitilmesi ağa ait elemanların bağlantı ağırlıklarının belirlenmesi anlamına gelmektedir. Başlangıçta ağırlık değerleri rastgele atanmaktadır. Y.S.A.'ya girişler sunuldukça ağın ağırlık değerleri güncellenmektedir. Amaç ağa gösterilen örnekler için doğru çıkışları üretecek ağırlık değerlerini bulmaktır. Ağın doğru ağırlık değerlerine ulaşması örneklerin temsil ettiği olay hakkında genellemeler yapabilme yeteneğine kavuşması demektir. Bu genelleştirme özelliğine kavuşması işlemine ağın öğrenmesi denir. Ağırlıkların değerlerinin değişmesi belirli kurallara göre yürütülmektedir. Bu kurallara Y.S.A.' nın öğrenme kuralları denir.

Ağın eğitimi tamamlandıktan sonraki aşama, ağın sınanması aşamasıdır. Sınanma sürecinde ağa daha önceden gösterilmemiş olan sınama verisi sunulmaktadır. Bu aşamada ağırlık güncellemeleri yani ağın öğrenmesi işlemleri gerçekleştirilmemektedir. Eğitim sırasında belirlenen bağlantı ağırlıkları yardımı ile daha önce karşılaşmadığı örnekler için çıkışlar üretilir. Beklenen çıkış değerleri elde edildiği oranda ağın öğrenmesi başarılıdır. Eğitim için kullanılan veri kümesine

eğitim kümesi, sınama için kullanılan veri kümesine sınama kümesi denir. Y.S.A.' ların bu şekilde bilinen örneklerden belirli bilgileri çıkartarak bilinmeyen örnekler hakkında yorum yapabilme yeteneğine adaptif öğrenme denir.

Geliştirilen Y.S.A. modelleri arasında en yaygın kullanılanları, tek ve çok katmanlı algılayıcılar, L.V.Q., A.R.T. ve S.O.M. ağlarıdır.

4.3. Öğrenme Algoritmalarına Göre Yapay Sinir Ağları Sınıflaması

Y.S.A' larda genel olarak üç öğrenme yönteminden ve bunların uygulandığı farklı öğrenme kurallarından söz edilmektedir. Bu öğrenme kuralları aşağıda özetlenmektedir.

- Denetimli öğrenme: bu öğrenme şeklinde, Y.S.A.' ya beklenen örnek çıkışlar verilmektedir. Beklenen ve elde edilen gerçek çıkış arasındaki farka (hataya) göre, bağlantı ağrılıkları en uygun çıkışı elde etmek üzere düzenlenirler. Bu nedenle, denetimli öğrenme bir "danışmana" ya da "eğiticiye" ihtiyaç duymaktadır. Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı veya geri beslemeli Y.S.A. algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir.

- Denetimsiz öğrenme: Giriş verisinden elde edilen çıkış bilgisine göre Y.S.A. sınıflandırma kuralını kendi kendine öğrenmektedir. Beklenen bir çıkış değeri bulunmamaktadır. Yalnızca giriş verileri ağa sunulmaktadır. Öğrenme sonucunda, bağlantı ağırlıkları kümeleri (benzer verileri) oluşturmak üzere düzenlenmiş durumdadır.

- Takviyeli öğrenme: Denetimli öğrenme modeline yakın bir yöntemdir. Denetimsiz öğrenme algoritması, beklenen bir çıkış değerine ihtiyaç duymamaktadır. Hedef çıkışı vermek için bir "danışman" yerine, burada Y.S.A.'ya bir çıkış verilmemekte fakat elde edilen çıkışın verilen girişe karşı iyiliğini değerlendiren bir ölçüt kullanılmaktadır vektör nicemleme modelleri

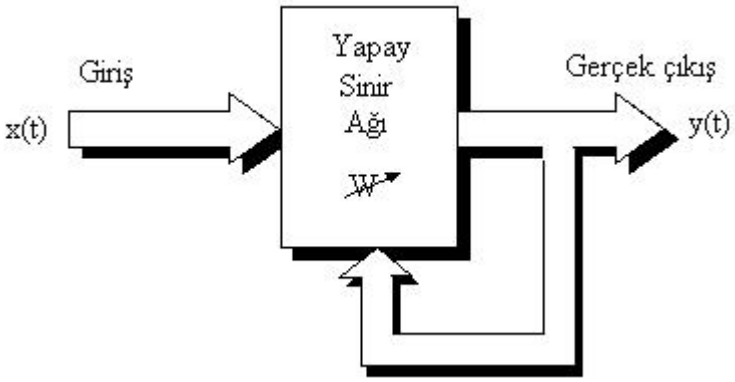
(L.V.Q.), optimizasyon problemlerini çözmek için Hinton ve Sejnowski'nin geliştirdiği Boltzmann kuralı takviyeli öğrenmeye örnek olarak verilebilirler [78, 79,80].

4.4. Denetimsiz Öğrenme için Yapay Sinir Ağları

Tez kapsamında yapılan çalışmada bir denetimsiz öğrenme algoritması kullanıldığı için denetimsiz öğrenme için yapay sinir ağları incelenmiştir.

Yapay sinir ağlarında, bazı problemlerin çözümünde elde sadece giriş verileri bulunmaktadır. Ağdan beklenen çıkış değerleri önceden bilinmeyebilir. Bu durumda çıkış değerleri hakkındaki bilgiye sadece giriş verilerinden ulaşılabilmektedir.

Giriş vektörleri belirlenmektedir fakat bunların karşılığında beklenen çıkış vektörleri bulunmamaktadır. Denetimsiz öğrenme modeli Şekil 4.2' de gösterilmektedir. Yapay sinir ağı oluşturulan her küme için kümeyi temsil eden örnek(temsilci) bir vektör üretmektedir [81] .



Şekil 4.2: Denetimsiz öğrenme modeli [78]

Bu denetimsiz öğrenme problemleri örnekleri şu şekilde özetlenebilir.

- Kümeleme: giriş verisi "kümelerde" gruplanmaktadır. Veri işleme sistemi bu kümelere giriş verileri içerisinden karar verebilir. Ağın çıkışı olarak kümelerin etiketleri düşünülmektedir.

- Vektör nicemleme: bu problem sürekli uzaydaki veri ayrıklaştırılacağı zaman oluşmaktadır. Sistemin girişi n boyutlu bir x vektörüdür, çıkışı giriş uzayının ayrık bir temsilidir.

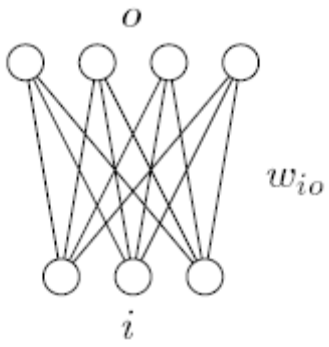
- Boyut indirgeme: Giriş verisi, verinin asıl boyutundan daha küçük boyutlu alt uzayda gruplanmaktadır. Sistem, giriş verisindeki boyut indirgemenin çıkış verilerini değiştirmeyeceği şekilde en uygun boyut indirgemeyi gerçekleştirebilmelidir.

Bu bölümde, bu tür problemler için önerilmiş ve tez kapsamında kullanılmış olan bazı yapay sinir ağı yaklaşımlarından bahsedilmektedir. Öğrenme bir dış öğretici olmadan gerçekleştirilmektedir. Denetimsiz ya da öğreticisiz olarak adlandırılan bu yaklaşımlar genelde nöronlar arasındaki yarışmaya dayalı adaptif ağırlık güncelleştirmeleri ile gerçekleşmektedir.

Bunlar çok geniş alanlarda kullanılmakta olan kendi kendini organize eden yapay sinir ağlarıdır. En temel ve en sık kullanılan yaklaşımlardan bir tanesi Kohonen tarafından önerilmiş olan Self Orginizing Maps (S.O.M.) algoritmasıdır. Bir başka yaklaşım Carpenter ve Grosberg [82, 83] tarafından önerilmiş olan Adaptive Resonance Theory (A.R.T.) algoritmasıdır. Bunlardan başka, Rumelhart ve Zipser [84] tarafından önerilmiş olan yarışmacı öğrenme algoritması sayılabilir.

4.4.1. Yarışmacı öğrenme

Yarışmacı öğrenme, giriş verisini, kendi doğasına uygun olarak kümelere bölen bir öğrenme şeklidir. Bir yarışmacı öğrenme ağı, sadece x giriş vektörü ile sağlanabilmektedir, böylece bir denetimsiz öğrenme geçekleştirilebilir.



Şekil 4.3: Yarışmacı öğrenme ağı [85]

Bir yarışmacı öğrenme ağı Şekil 4.3' de gösterilmektedir. Tüm çıkış birimleri (o) tüm giriş birimlerine (i) w_{io} ağırlıkları ile bağlanmıştır. Bir x giriş vektörü ağa sunulduğunda, sadece ağın tek bir çıkış birimi (kazanan hücre) aktif olmaktadır. Kazanan hücre tespitinde iki yöntem uygulanmaktadır.

4.4.1.1. Kazanan hücre seçimi: nokta toplamı

Giriş vektörü X ve ağırlık vektörü W_o normalize edilmektedir. Her çıkış birimi (o) kendi aktivasyon değerini (y_o) giriş ve ağırlık vektörünün nokta toplamına göre hesaplamaktadır.

$$y_O = \sum_i w_{io} x_i = W_o^T X \tag{4.1}$$

Bir sonraki geçişte, en yüksek aktivasyonlu çıkış hücresi k' seçilmektedir.

$$\forall_O \neq k' : \qquad y_O \leq y_{k'} \tag{4.2}$$

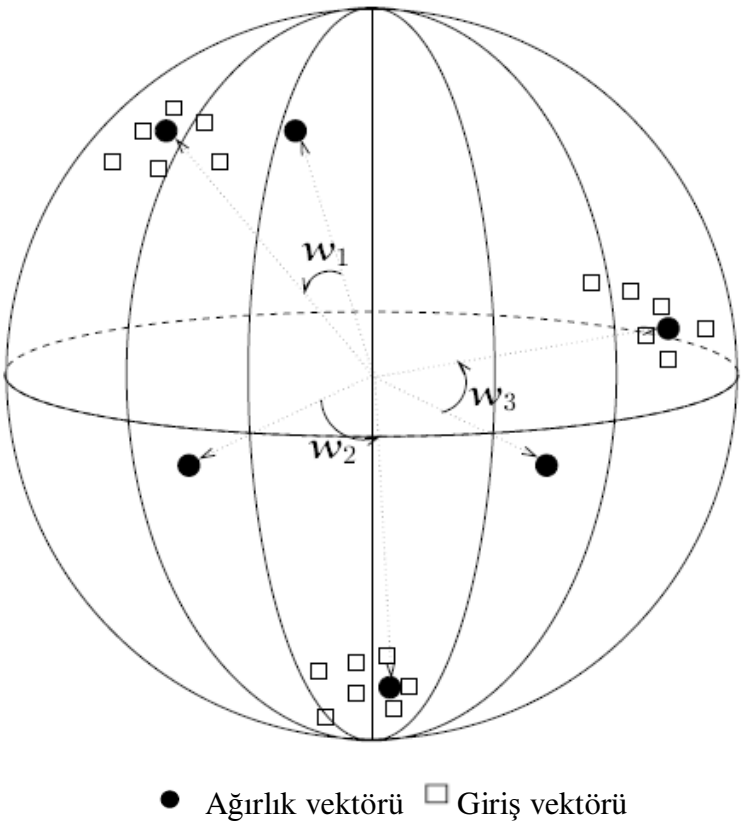
Bu durumda ağın çıkış katmanı "kazanan hepsini alır" katmanıdır.

Kazanan k' hücresi bir kez seçildikten sonra, ağırlıklar aşağıdaki bağıntıya göre güncellenmektedir.

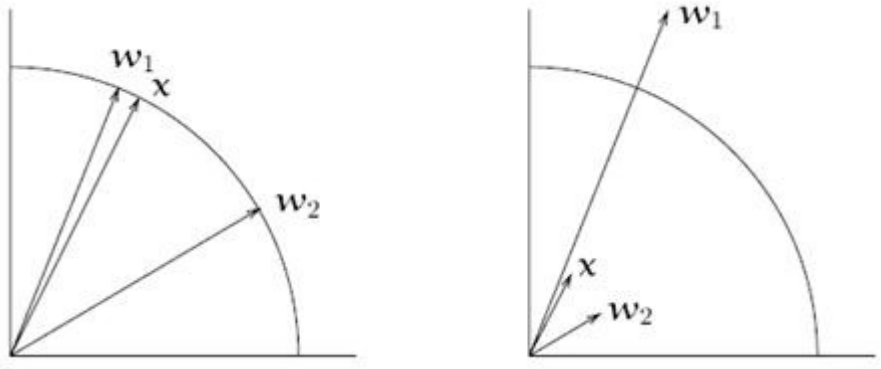
$$W_{k'}(t+1)=\frac{W_{k'}(t)+\gamma\left(X(t)-W_{k'}(t)\right)}{\left\|W_{k'}(t)+\gamma\left(X(t)-W_{k'}(t)\right)\right\|} \tag{4.3}$$

Sadece kazanan k' hücresine ait ağırlıklar güncellenmektedir.

Bağıntı 4.4' de verilen ağırlık güncelleme bağıntısı W_O ağırlık vektörünü X giriş vektörüne doğru döndürmektedir. X vektöründen her bir eleman ağa sunulduğunda, ağırlık vektörü giriş vektörüne yaklaşmaktadır. Sonuç olarak, ağırlık vektörleri giriş vektörlerinde kümelerin bulunduğu elemanların yoğunlaştığı alanlara döndürülmektedir. Bu durum Şekil 4.4' de ifade edilmektedir.



Şekil 4.4: Üç ağırlık vektörü farklı küme merkezlerine doğru döndürülmüşlerdir [85].



(a) (b)

Şekil 4.5: Yarışmacı öğrenme ağında kazanan hücreyi belirlemek. (a) Üç normalize edilmiş vektör. (b) Üç vektör a'daki ile aynı yönde fakat farklı uzunluklarda [85].

Şekil 4.5.(a)' da x ve W_1 vektörleri birbirlerine yakındırlar ve nokta toplamları da $X^T W_1 = |X||W_1|\cos\alpha$ X ve W_2'nin nokta ağırlık toplamlarından daha büyüktür. Bununla birlikte, Şekil 4.4.(b)' de giriş ve ağırlık vektörleri normalize edilmemiştir ve bu durumda X ağa sunulduğunda W_2 kazanan olmaktadır. Fakat $X^T W_1$ hala $X^T W_2$'den daha büyüktür.

4.4.1.2. Kazanan hücre seçimi: Öklid uzaklığı

Bir önceki yaklaşımda olduğu gibi bunda da giriş ve ağırlık vektörü normalize edilmektedir. Şekil 4.5' de normalize edilmemiş vektörlerin ağa sunulması halinde algoritmanın nasıl yanlış sonuçlar üreteceği ifade edilmektedir. Bunu sonlandırmak için, kazanan hücre k' kendi W_k ağırlık vektörü ile seçilmektedir. X giriş vektörüne en yakın olan Öklid uzaklığı ile belirlenmektedir.

Öklid uzaklık ölçümü:

$$k' : \left\| W_{k'} - X \right\| \leq \left\| W_O - X \right\| \qquad \forall_O \tag{4.4}$$

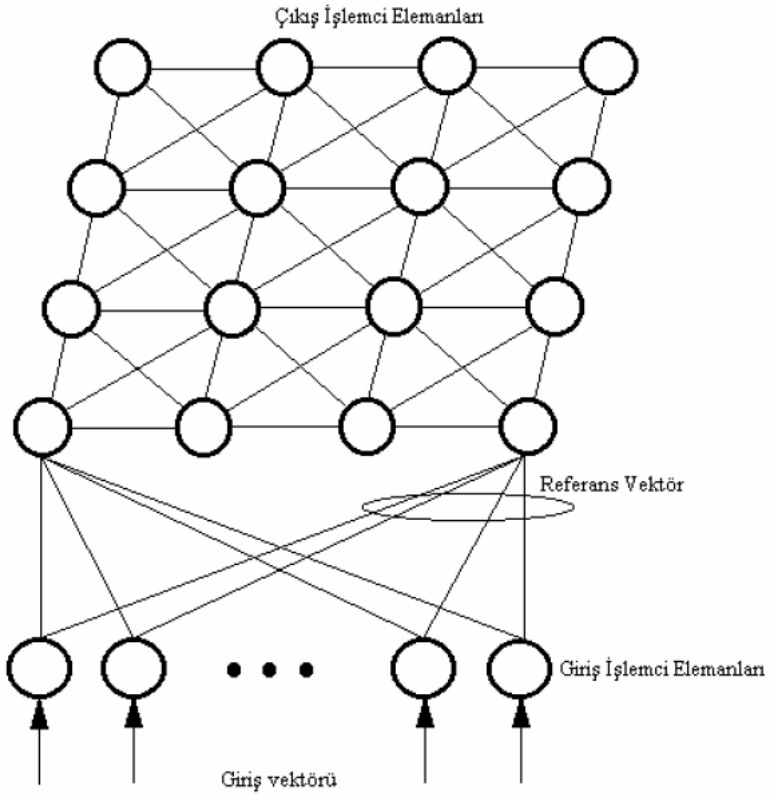
Ağırlık güncellemeleri bağıntı 4.5'e göre gerçekleştirilmektedir.

$$W_{k'}(t+1) = W_{k'}(t) + \gamma\left(X(t) - W_{k'}(t)\right) \tag{4.5}$$

Yine sadece, kazanan hücreye ait ağırlıklar güncellenmektedir.

4.5. S.O.M. Ağı

S.O.M. ağı [86, 87] yarışmacı öğrenmenin bir uzantısı olarak görülebilir. Genel olarak sınıflandırma yapmak için kullanılmaktadırlar. Bu ağların girdi vektörlerini sınıflandırmak ve girdi vektörlerinin dağılımlarını öğrenebilme yetenekleri çok fazladır. Bu ağların en temel özelliği olayları öğrenmek için bir öğretmen veya ağın üretmesi gereken çıktıların ağa söylenme zorunluluğunun olmamasıdır. Bir giriş bir de çıkış olmak üzere iki katmandan oluşmaktadır. Bu ağ Şekil 4.6' da gösterilmektedir. Çıkış katmanındaki her nöron bütün giriş düğümlerine bağlıdır. Bağlantıların ağırlıkları, verilen çıkış elemanı ile ilgili olan referans vektörünün elemanlarını oluşturmaktadır.



Şekil 4.6: S.O.M. ağı [88]

S.O.M. ağına ait öğrenme modeli şu şekildedir:

- Çıkış nöronlarının bütün referans vektörlerini küçük rastgele değerlere çekilmesi
- Bir giriş verisinin ağa sunulması
- Giriş verisine en yakın referans vektörüne sahip nöronun belirlenmesi
- Belirlenen nöronun ve onun komşularının referans vektörlerinin güncelleştirilmesi. Bahsedilen bu referans vektörleri güncelleme işleminden sonra giriş vektörüne yaklaştırılmış olmaktadır. Bu yaklaştırma, belirlenen nöron için en fazla ve bu nörondan uzaklaştıkça daha azdır. Öğrenme ilerledikçe komşuların sayısı azalmakta ve öğrenme sonunda belirlenen nörona ait referans vektörü son halini almış olmaktadır.

4.6. A.R.T. Ağı

Stephen Grossberg tarafından 1976 yılında önerilen A.R.T. ağları denetimsiz öğrenme algoritmalarıdır. Grosberg'in biyolojik beynin fonksiyonlarına yönelik yaptığı çalışmalar sonucunda ortaya çıkmıştır. Yapısal olarak insan beyninin davranışları ve sinir sistemi hakkında bilinen bulgular üzerine kurulmuşlardır [78]. A.R.T. ağları insanlardaki öğrenme sürecini andıran otomatik adımlar kapsamaktadır [89]. Ağ, sınıflandırma problemleri için geliştirilmiştir. İlk örnek ağa sunulduğunda, örnek model olarak saklanmaktadır. İkinci örnek daha sonra örnek model ile karşılaştırılmakta ve eğer benzerlik varsa aynı kümede, benzerlik yoksa yeni bir kümede saklanmaktadır [88, 90].

A.R.T. ağlarının en temel özelliği sınıflandırma problemleri için geliştirilmiş olmalarıdır. Sınıflandırma günümüzde en çok karşılaşılan problemlerin başında gelmektedir. Çevremizdeki birçok nesneyi de bizler sınıflandırmış durumdayız. Benzer şekilde endüstriyel kuruluşlarda makineler üzerinde oluşacak olan hataları sınıflandırmakta ve her sınıfa giren hatalar için o sınıfa özel çözümler üretilmektedir. Meslekler değerlendirilirken de sınıflandırılmaktadır. Mühendislik problemlerinin çoğu da sınıflama problemi haline getirilerek çözülmektedir. Sınıflandırma hayatımızda bu kadar önemli bir yer tuttuğundan sınıflandırma yapabilen yapay sinir

ağları da önemli bir yer tutmaktadır. A.R.T. ağları bu amaçla geliştirilmiş ve başarılı bir şekilde kullanılabilen bir yöntem olarak bilim dünyasında kabul görmüştür [78].

A.R.T. ve diğer sınıflandırma yöntemleri arasındaki en büyük farklılık ağın; uygunluk parametresi olarak adlandırılan kullanıcı tanımlı bir sabit vasıtasıyla, aynı kümenin üyeleri arasındaki benzerlik derecesinin kontrolüne izin vermesidir. A.R.T. ağlarında yapılacak olan sınıflandırma ile ilgili olarak ağa herhangi bir bilgi verilmemektedir. A.R.T. ağları bu sınıflandırmayı belirlenen uygunluk parametresine göre kendi başına yapmaktadır.

Bu ağlarda öğrenme doğru bilgilerin belleğe alınması anlamına gelmektedir. Öğrenme sırasında kullanılan örneklerden öğrenilen bilgilere dayanarak daha sonra görülmemiş örnekler hakkında yorumlar yapılabilmektedir. Bilgilerin bellekte saklanması ve bellekte tutulması iki şekilde olmaktadır [78]:

1. Kısa dönemli hafıza (K.D.H.): Bilgilerin geçici olarak tutulduğu ve zaman içerisinde yok olduğu ve yerlerine başka bilgilerin saklandığı bellektir. Bu durum insanlar için de benzer şekildedir. Bir takım bilgiler kısa dönemli bellekte tutularak başka olayların etkisi ile unutulmaktadırlar.

2. Uzun dönemli hafıza (U.D.H.): Bilgilerin sürekli tutulduğu ve kolay unutulmadığı bellektir. Bilginin silinmesi için çok uzun zamanın geçmesi gerekebilmektedir. Bilginin uzun dönemli tutulabilmesi için o bilginin bizim için öneminin olması gerekmektedir.

A.R.T. ağlarında da tıpkı insanlarda olduğu gibi bilgiler hem kısa dönemli hem de uzun dönemli bellekte saklanmaktadır.

A.R.T., standart ileri besleme ağlarında görülen öğrenmedeki istikrarsızlık problemini çözmek amacıyla geliştirilmiştir. Geçmişteki bilgilerle elde edilen ağırlıklar yeni bilgiler geldikçe değişmektedir. Bu nedenle, zamanla eski bilgiyi kaybetme tehlikesi meydana gelmektedir. Ağırlıklar yeni bilgiyi barındıracak kadar esnek ama eski bilgiyi kaybetmeyecek kadar eski olmalıdır.

Bu sabitlik-esneklik ikilemi olarak adlandırılır ve yapay sinir ağları paradigmasının gelişmesinde ana etkenlerden biridir. A.R.T. bu problemi, yukarıdan aşağı (çıktı-girdi) öğrenmeyle aşağıdan yukarı (girdi-çıktı) rekabetçi öğrenme birleşimini ortaya koyarak çözümlemektedir. A.R.T., bilgiyi önceden öğrenilen örneklerle sabit ve adaptif (esnek) olarak korumaktadır [91].

A.R.T. ağlarının ilk sürümü Carpenter ve Grossberg tarafından 1988 yılında geliştirilen A.R.T. 1 ağıdır. Küme keşfinde kullanılan A.R.T 1 ağı, kümeleri saptama işlemini, önceden öğrenilmiş olan kümeleme bilgileri ile alakalı herhangi bir geri çağırma işlemi olmadan gerçekleştirmektedir [88]. Sonradan, bu algoritmaya birçok iyileştirme uygulanarak farklı ağ türleri geliştirilmiştir.

4.6.1. A.R.T. modelinin temel özellikleri

Grossberg'in insan beyninin biyolojik fonksiyonlarından yola çıkarak yaptığı çalışmalar sonucunda beynin çalışmasını açıklayacak bir model önermiştir. Bu modelin 3 temel özelliği vardır:

1-) Normalizasyon: Bu, özellikle biyolojik sistemlerin çevredeki büyük değişikliklere karşı adaptif oldukları durumu göstermektedir. Örneğin insanın çok fazla gürültülü bir ortamda bir süre sonra gürültüden rahatsız olması sisteme adapte olduğunu ve çevredeki olayların normalize edildiğini göstermektedir.

2-) Ayrıştırabilme: İnsanın karar verebilmesi ve olayları yorumlayabilmesinde çevredeki olaylar arasında var olmasına karşın görülmesi zor farklılıkları ayrıştırmak çok önemlidir. Bazen küçük ayrıntılar hayati öneme sahip olabilir. Biyolojik sistemlerin ayrıntıları fark etmeleri çok önemli bir özelliktir.

3-) Ayrıntıların saklandığı kısa dönemli bellek: Belirlenen farklılıklar ve çevresel olaylar davranışlara neden olmadan önce bellekte saklanmakta ve daha sonra eyleme dönüşmektedir. Bu uzun dönemli bellekte değişikliklere neden olmaktadır. Bellekteki her olay uzun süre etkili olmamakla beraber, sürekli aynı şeyleri tekrar etmek sonucu olaylar unutulmaz hale gelebilmektedir [78]. Giriş örneği

kodlanmadan önce, kısa dönemli bellekte saklanması gerekmektedir. Uzun dönemli bellek, harekete geçiren bir mekanizma uygular (örneğin kümelendirme), oysa kısa dönemli bellek, uzun dönemli bellekte kademeli değişikliklere neden olması açısından kullanılmaktadır [85].

Bu özelliklerden yola çıkılarak A.R.T. adı verilen yapay sinir ağları oluşturmuştur.

4.6.2. A.R.T. ağlarının diğer yapay sinir ağlarından farkları

Farklar Grossberg tarafından şu şekilde özetlemektedir:

- A.R.T. ağları gerçek zamanlı olarak oldukça hızlı ve kararlı bir şekilde öğrenme yeteneklerine sahiptirler. Bu yetenek birçok ağda yoktur. A.R.T. ağları bu özellikleri ile gerçek zamanlı olarak kullanılabilen donanımla da desteklenerek gerçek zamanlı öğrenebilen bilgisayarların oluşmasına yardımcı olmaktadırlar.

- Gerçek zamanda ortam genel olarak durağan değildir. Durumların oluşumu her an beklenmedik olaylar ile değişebilmektedir. Bunun da ötesinde gerçek zamanlı olaylar sürekli devam etmektedir. A.R.T. ağları bu durağan olmayan dünyada sınırsız karmaşıklık altında çalışabilme yeteneğine sahiptirler. Diğer ağların çoğu ise durağan olarak çevrimdışı öğrenip çalışırlar. Esneklikleri yoktur. Ortama anında uyum sağlamaları çok sınırlıdır.

- A.R.T. ağları beklenen çıktıları bir öğretmenden almak yerine kendi kendine öğrenmeye çalışır.

- A.R.T. ağları ağa sunulan farklı nitelikteki ve değişik durumlardaki örnekler karşısında kendi kendilerine kararlı bir yapı oluşturabilirler. Ağa sunulan, yeni bir girdi geldiği zaman ya bilinen kümelerin kodlarına ulaşabilecek şekilde ağda iyileştirmeler yapılır ya da yeni kod (küme) oluşturulur. Bu ağın büyümesine neden olabilir ve ağın bütün kapasitesini kullanana kadar devam eder.

- A.R.T. ağları çevredeki olayları sürekli öğrenmeye devam eder. Uzun dönemli

bellekte bulunan ağırlıklar sürekli olarak gelen girdi değerlerine göre değişmeye devam ederler.

- A.R.T. ağları girdi değerlerini otomatik olarak normalize ederler. Çok fazla ve oldukça düşük orandaki gürültülerin girdi işaretindeki etkileri ortadan kaldırılmış olur.

- A.R.T. ağlarında hem aşağıdan yukarı hem de yukarıdan aşağıya ağırlık değerleri vardır. Özellikle yukarıdan aşağıya ağırlıklar kümeleri temsil etmektedirler. Bunları ağın kendisi girdilere bağlı olarak otomatik olarak belirlemektedir. Bu ağırlıklar aynı kümeden olan bütün örneklerin ortak yönlerini içermektedir. Bu ağırlıklardan oluşan örüntülere kritik özellik örüntüleri denmektedir. Yukarıdan aşağı ağırlıklar ağın öğrendiği beklentileri (beklenen girdi temsilcileri) göstermektedir. Bu değerler aşağıdan yukarı gelen bilgiler ile karşılaştırılarak eşleme yapılır. Aşağıdan yukarı gelen bilgiler ile karşılaştırma kısa zamanlı bellekte oluşmaktadır. Aşağıdan yukarı ve yukarıdan aşağı ilişkiler bir A.R.T. ağında kapalı çevrimi tanımlamaktadır.

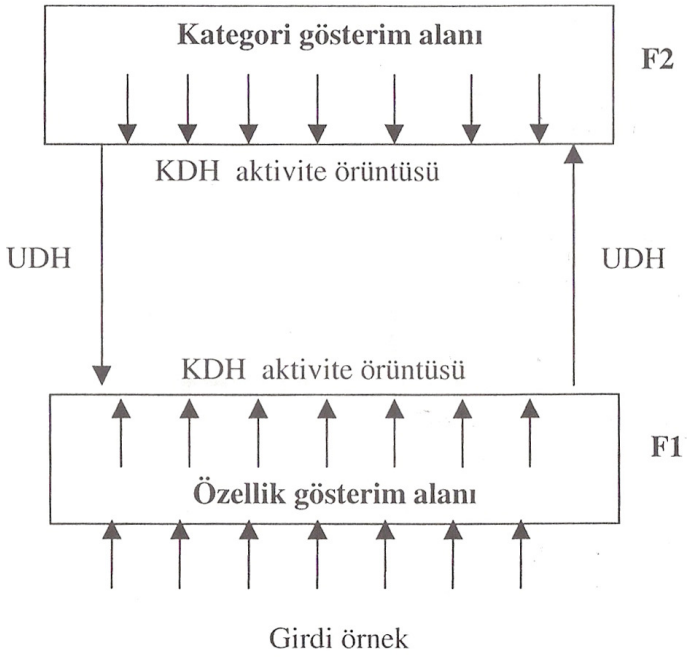
- Bu kapalı çevrimden dolayı yukarıdan aşağı ağırlıklar K.D.H.' da yapılan karşılaştırma ile kazanç faktörünü kullanarak aynı kümede olmayan girdilerin o kümeye girmesini önlemektedir. Böylece kümeyi gösteren ağırlıkların gerçek zamanlı gelen farklı bir girdiden etkilenmeleri önlenmektedir. Böyle bir kontrol, yapılmaması gereken her girdi değerinin ağırlıklarını değiştirerek önceden öğrenilen bilgilerin kayıp olmasına neden olacaktır. A.R.T. bu özelliği ile sürekli öğrenmeyi desteklemekte ve önceden öğrenilenler ancak aynı gruptaki başka örneklerin yeni özellikleri olunca değiştirilmektedir. Bu özellik ise yakın eşleşme olarak bilinmektedir.

- A.R.T. ağlarının yakın eşleşme özelliğinden dolayı hem hızlı hem de yavaş öğrenebilme yetenekleri vardır. Hızlı öğrenme U.D.H. bir denemede yeni bir dengenin oluşturulması ile gerçekleştirilir. Yavaş öğrenme ile bir dengenin oluşması için birden çok denemenin yapılması durumu kastedilmektedir. Halbuki

çok katmanlı algılayıcılar gibi ağlarda salınımları önlemek için özellikle yavaş öğrenme zorunluluğu vardır [78].

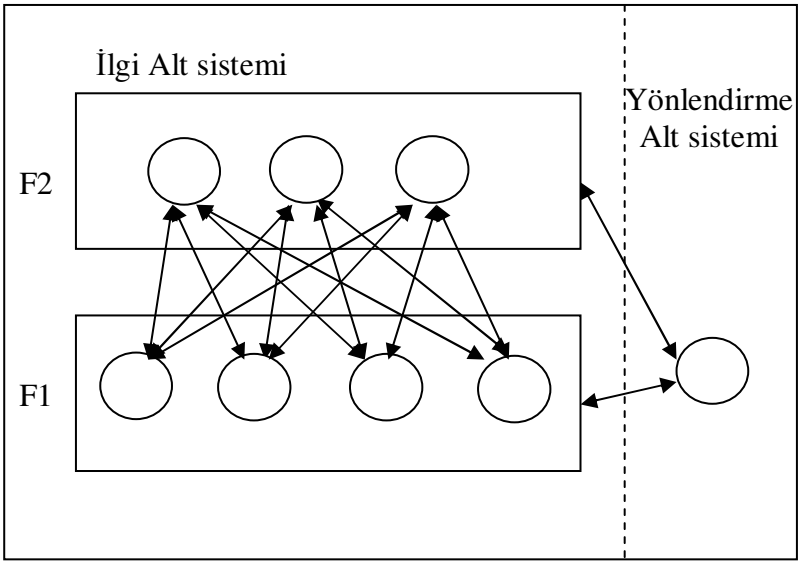
4.6.3. A.R.T. ağlarının yapısı

A.R.T. ağları genel olarak iki katmandan oluşmaktadır. Bu katmanlar F1 (giriş katmanı) ve F2 (çıktı katmanı) olarak isimlendirilmiştir. Diğer ağlardaki gibi gizli katmanlar bulunamamaktadır. F1 katmanı girdinin özelliklerini gösterirken F2 katmanı elde edilen kümeleri göstermektedir. F2 katmanındaki düğüm sayılarına dinamik olarak karar verilmektedir. Bu iki katman birbirlerine uzun dönemli bellek bağlanmaktadırlar. Giriş verileri F1 katmanından alınmakta ve kümeleme F2 katmanında gerçekleştirilmektedir. A.R.T. ağlarının genel yapısı Şekil 4.7 ve 4.8'de gösterilmektedir.



Şekil 4.7: A.R.T. ağının genel yapısı [78]

A.R.T. ağı, ilgi alt sistemi ve yönlendirme alt sistemi olmak üzere iki alt-sistem aracılığıyla idare edilmektedir: İlgi alt sistemi, küme üretmekte; yönlendirme alt sistemi, kümeyi kabul edip etmeyeceğine karar vermektedir. Bu yapı Şekil 4.8'de verilmektedir [27, 28].



Şekil 4.8: İlgi ve yönlendirme alt sistemi [27]

A.R.T. ağlarında girdiler doğrudan sınıflandırılmamaktadırlar. İlk olarak girdilerin özellikleri incelenerek F1 katmanının aktivasyonu belirlenmekte; U.D.H.' daki bağlantı değerleri ile gelen bilgiler kümelere ayrılarak F2 katmanına gönderilmektedirler. F2 katmanındaki sınıflandırma ile F1 katmanından gelen sınıflandırma birbirleri ile eşleştirilmekte, eğer örnek belirlenmiş bir sınıfa uyuyorsa o kümede gösterilmektedir. Aksi takdirde yeni bir küme oluşturulmaktadır [28].

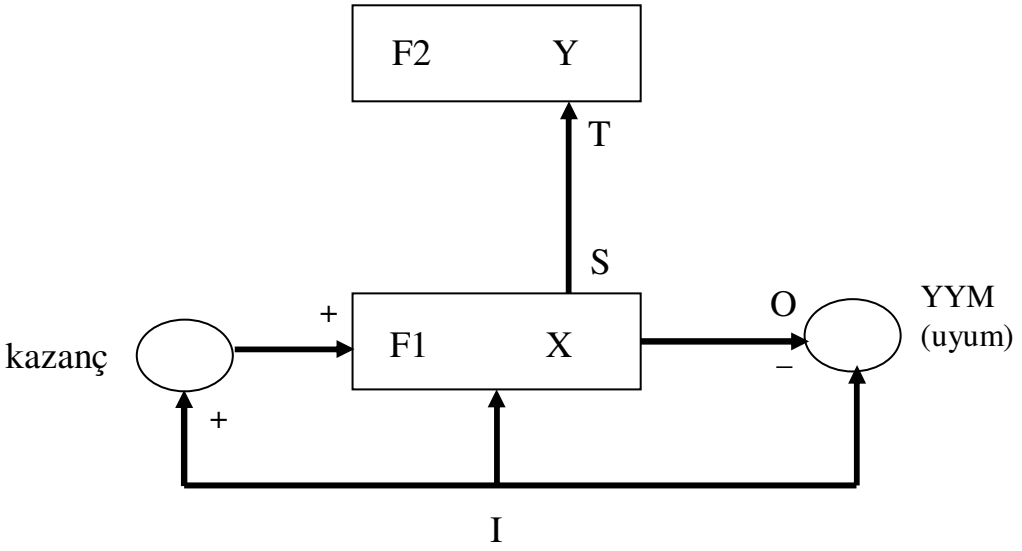
4.6.4. A.R.T. ağlarının çalışma ilkesi

A.R.T. ağları F1 katmanından gelen bilgileri F2 katmanındaki kümeler ile eşleştirmektedir. Bu eşleşme sağlanamaz ise yeni bir küme oluşturmaktadır. A.R.T. ağlarının çalışması iki yönlü olmaktadır:

- Aşağıdan yukarı (F1 den F2' ye) bilgi işleme
- Yukarıdan aşağı (F2 den F1' e) bilgi işleme

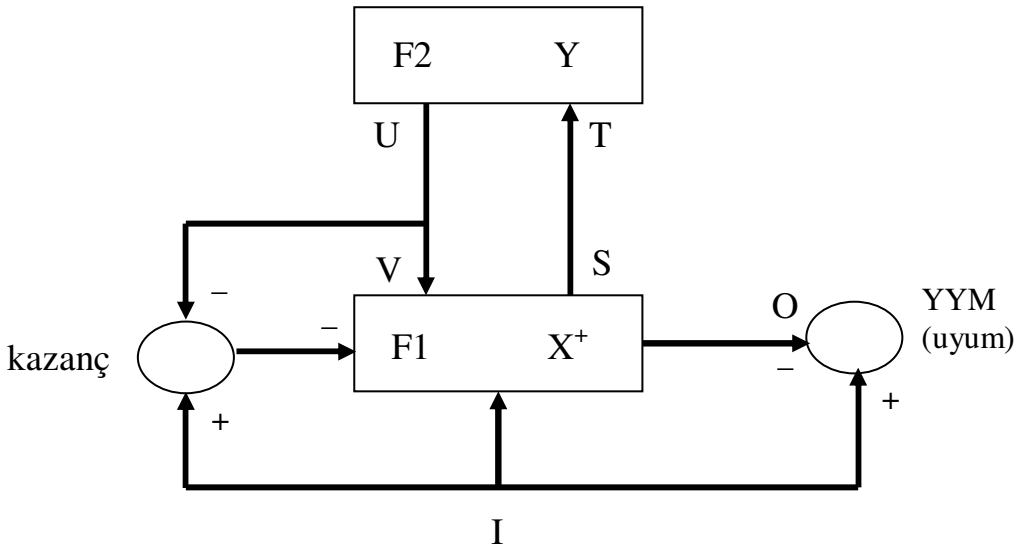
A.R.T. ağlarında aşağıdan yukarı bilgi işleme ilkesi Şekil 4.9' da gösterilmektedir. Şekilde, bir girdi örüntüsü (I) ağa gösterilmektedir. Bu örüntü hem F1 katmanında K.D.H. da X aktivite örüntüsünü oluşturmakta hem de uyum sistemini veya diğer bir deyişle yeniden yerleştirme modülünü (Y.Y.M.) aktif etmek üzere bir işaret

göndermektedir. Benzer şekilde oluşturulan X örüntüsü hem Y.Y.M.' ne bir men-edici işaret (O) göndermekte hem de F1 katmanından bir çıktı örüntüsü (S) oluşturmaktadır. S sinyali F2 katmanına giden bir girdi örüntüsüne (T) dönüştürülmektedir. Bu girdi örüntüsü ise F2 katmanının çıktısı olan örüntüyü (Y) oluşturmaktadır. Bu aynı zamanda ağın da çıktısıdır. Bu şekilde aşağıdan yukarı (F1 katmanından F2 katmanına) bilgi işleme tamamlanmış olmaktadır [78].



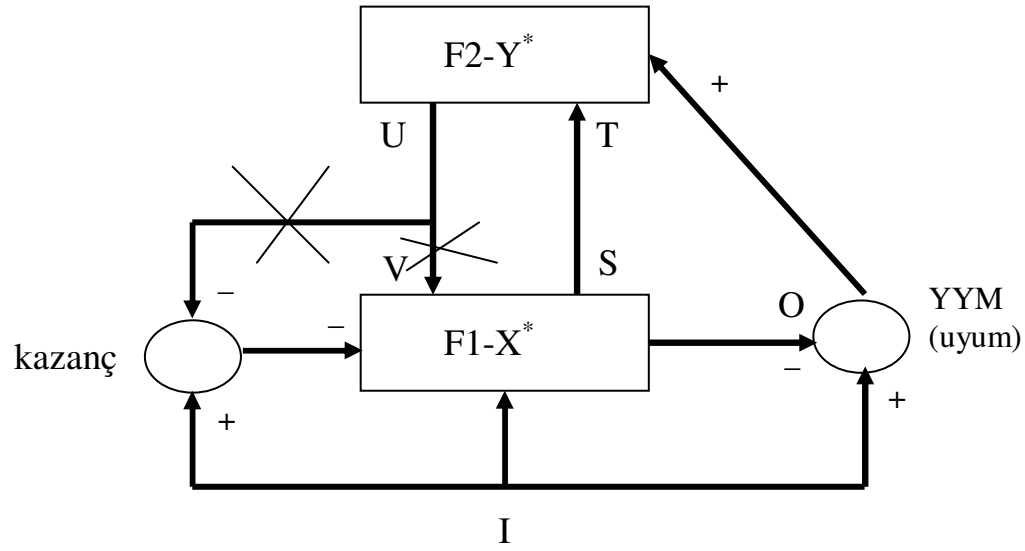
Şekil 4.9: A.R.T. ağında çıktı oluşturma süreci (aşağıdan yukarı) [78]

Yukarıdan aşağı bilgi işleme de, benzer şekilde, Şekil 4.10' da gösterildiği gibi yapılmaktadır. Bu durumda, F2 katmanında oluşturulan çıktı örüntüsü yukarıdan aşağıya bir sinyal (U) göndermektedir. Bu sinyal daha sonra beklenen şablon örüntüye (V) dönüştürülmektedir. Aynı zamanda kontrol faktörü (kazanç) için men-edici bir işaret üretmektedir. Bundan sonra şablon örüntünün girdi örüntüsü ile eşlenip eşlenemeyeceği sınanmaktadır. Eğer böyle bir eşleşme mümkün değilse o zaman F1 katmanında yeni bir K.D.H. örüntüsü (X*) oluşturulmaktadır. Bu örüntü uyum sistemindeki men-edici işaretin etkisini azaltmaktadır.



Şekil 4.10: A.R.T. ağında çıktı oluşturma süreci (yukarıdan aşağı) [78]

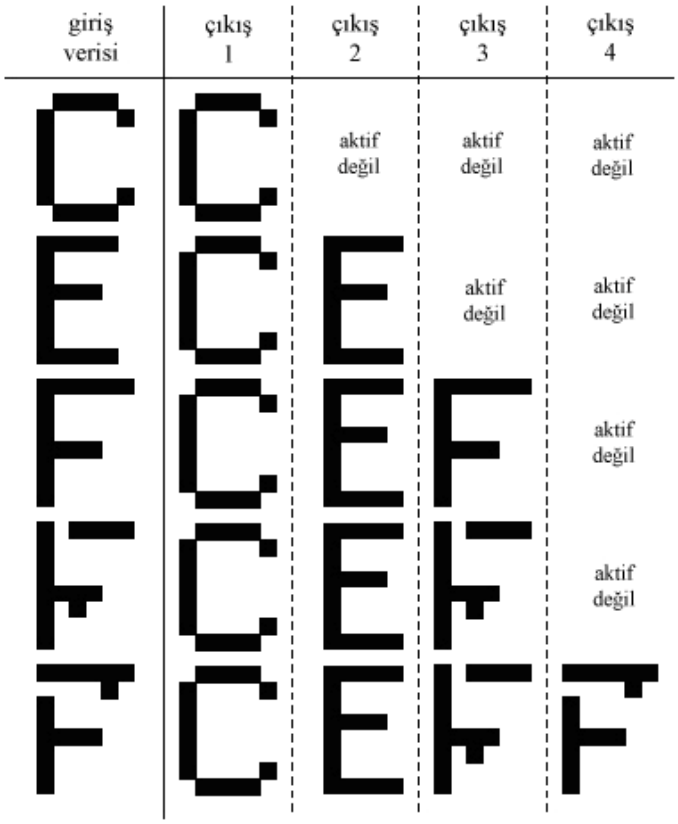
Oluşturulan X* sinyali uyum sisteminde O işaretinin men edici etkisini azaltarak YYM' nün (uyum modülünün) F2 katmanına bir sinyal göndermesini sağlamaktadır. Bu işaret F2 katmanında Y* örüntüsünü oluşturmaktadır. Böylece, Şekil 4.11'de gösterildiği gibi girilen I örüntüsü için doğru sınıfı gösteren Y* çıktısı üretilmektedir.



Şekil 4.11: A.R.T. ağında yeni bir sınıf oluşturma [78]

Eğer üretilen V şablon örüntüsü ile girdi örüntüsü eşleşir ise o zaman sadece yukarıdan aşağı o girdinin sınıfını gösteren ağırlıklar değiştirilir. Bu değiştirme öğrenme kuralına göre gerçekleştirilir. Her A.R.T. modelinin öğrenme kuralı ayrıdır [78]

A.R.T. ağının örnek davranış modeli Şekil 4.12'de gösterildiği gibidir.



Şekil 4.12: Harf verileri için A.R.T. ağının çalışması [85]

Şekil 4.12' nin sol tarafında harflere ait ikilik veri çiftleri sırası ile ağa sunulmaktadır. Sağ tarafta ise kümeler görülmektedir.

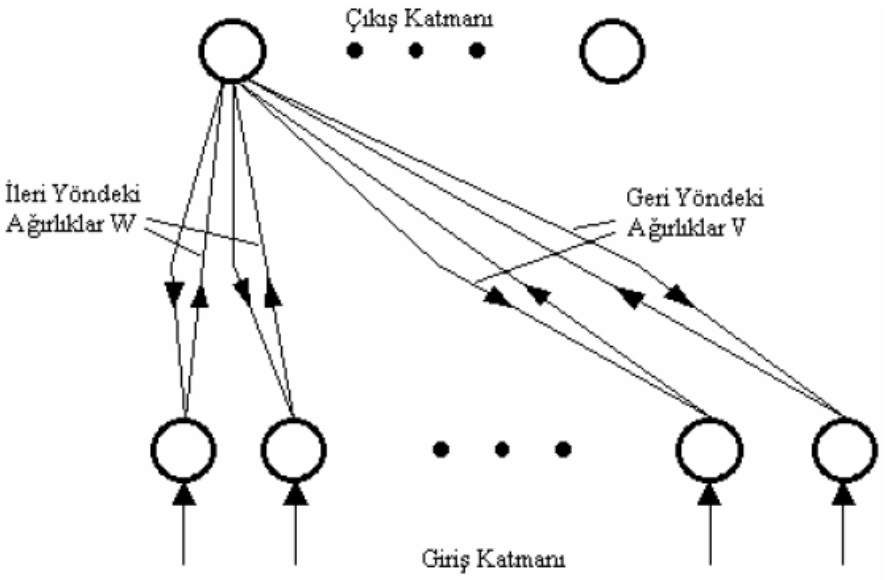
4.6.5. A.R.T. ağlarındaki farklı modeller

1976 yılından bugüne kadar farklı A.R.T. ağları tanımlanmıştır. Bunlar arasında A.R.T.1, A.R.T.2, A.R.T.3, A.R.T.M.A.P., Bulanık A.R.T. gibi ağları saymak mümkündür. Bu ağların hepsi aslında aynı temel felsefeye dayanmakta ve çok az farklılıklar göstermektedir. En yaygın olarak kullanılan A.R.T.1 ve A.R.T.2 ağlarıdır.

4.6.5.1. Adaptif rezonans teorisi 1

A.R.T.1 ağı Carpenter ve Grosberg tarafından [82, 92], geliştirilmiş kümeleme algoritması olarak da kaynaklarda bahsedilen bir A.R.T. modelidir. Ağ kümeleri

denetimsiz öğrenme modeli ile öğrenmektedir. Ağ kümeleri herhangi bir önbilgi olmadan kendisi üretmektedir. İlk giriş verisi ilk kümeyi çizmeye başladıktan sonra bir lider gibi algoritma bu giriş verisini takip ederek başlamaktadır. Peşinden gelen girişler eşik benzerlik değerini geçerlerse aynı kümeye dahil edilirler, geçemezler ise yeni bir küme oluştururlar. Bu şekilde kümeleme işlemi devam etmektedir [88]. Ağ sadece ikili girişler ile çalışmaktadır. Bu nedenle giriş verileri [0,1] aralığında normalize edilmektedirler. F1 ve F2 katmanlarından oluşmaktadır. F1 katmanındaki tüm elemanlar F2 katmanındaki tüm tüm elemanlara ağırlık değerleri ile bağlanmışlardır. Ağırlık değerlerini ayarlayarak yeni girişleri kodlamaktadır. Aşağıdan–yukarı ağırlıklar sınıflandırma sağlamaktadır. Yukarıdan–aşağı ağırlıklar sınıflandırmanın doğru olup olmadığını sınamak için kullanılmaktadır. İşlemleri açıklanırken geleneksel küme teorisini ve mantığını kullanılmaktadır. Şekil 4.13'de aşağıdan - yukarı ağırlıklar sınıflandırma sağlamaktadır.



Şekil 4.13: Küme keşfi için yapay sinir ağı (A.R.T.1) [88]

i. çıkış nöronunun ileri yöndeki bağlantılarının ağırlıklarının oluşturduğu W_i vektörü, temsil ettiği sınıfın bir örneğini oluşturur. İleri bağlantılarının W_i vektörlerinin toplamı A.R.T. 1 ağının uzun dönem belleğini oluşturur. Bu vektörler giriş verisine benzerlikleri oranında çıkış nöronlarını belirlerler. Çıkış nöronlarına ait geri bağlantıların W_i vektörleri ise, giriş verisinin bellekteki veriye yeterince benzeyip

benzemediğini (aynı kümeye dahil olup olmadıklarını) kesin olarak belirlemek için uygunluk sınaması amacı ile kullanılmaktadır. Sınama vektörleri de denen V_i vektörleri A.R.T.1 ağının kısa dönem belleğini oluşturmaktadırlar.

Başlangıç durumunda X giriş ve W_m ağırlık vektörüne ait ilk eşleşme sonuçları bağıntı 4.6 ile ifade edilmektedir.

$$W_i = \frac{V_i}{\varepsilon + \sum V_{ji}} \tag{4.6}$$

Burada ε küçük bir sabittir ve V_{ji} : V_i 'nin j. elemanıdır.

4.6.5.2. Adaptif rezonans teorisi 2

- 1986 ve 1987 yıllarında Carpenter ve Grosberg [82, 93] tarafından geliştirilmiştir.
- A.R.T. 1 algoritmasının geliştirilmiş halidir. Hem ikili hem de sürekli değerli giriş örnekleriyle çalışmaktadır.

4.6.5.3 Adaptif rezonans teorisi 3

- Biyolojik olarak esinlenilmiş ve teori geliştirmek için kullanılmaktadır.

4.6.5.4. Bulanık adaptif rezonans teorisi

- A.R.T.1' e benzemektedir ancak geleneksel mantıksal operasyonları bulanık mantık operasyonları ile yer değiştirmiştir.
- Danışmasız öğrenme (kümeleme) gerçekleştirmektedir.
- Sürekli değerli veriler ile çalışmaktadır.
- İki katmanlı eğiticisiz öğrenme modelidir.
- Önceden belirlenen uygunluk parametresi kaç kümenin oluşturulacağını kontrol etmektedir. Yüksek uygunluk değeri seçilerek az elemanlı daha fazla küme

oluşturulabilmekte. Düşük uygunluk değeri ile çok elemanlı daha az küme oluşturulabilmektedir.

- Küme merkezlerinin giriş örneklerinin (ağırlık kümesi) temsilini çabucak oluşturduğu durumlarda öğrenme hızlı olabilmektedir. Ağırlıkların küme merkezlerini oluşturmada yapılan kademeli değişiklikler alternatif yavaş öğrenmedir.
- Prototip değerlerini kodlayan kümelere ağırlıklar liderlik etmektedir.

4.6.5.5. A.R.T.M.A.P. ve bulanık A.R.T.M.A.P.

Eğiticili öğrenme sistemini yaratmak için birleşmeli bellek ile, iki eğiticisiz öğrenme sınıflandırma ağını birleştirir. (A.R.T.1 ve Bulanık A.R.T.) [78, 94, 95].

4.7. Bulanık A.R.T.

Bulanık adaptif rezonans teorisi (Bulanık A.R.T.) Carpenter, Grossberg ve Rosen tarafından [6] 1991'de geliştirilmiş bir A.R.T. ağı modelidir. Diğer A.R.T. ağlarında olduğu gibi "kazanan hücre hepsini alır" yarışmacı öğrenme mantığı ile işlemektedir. Bulanık A.R.T. yönteminde çıkış düğümleri hem ikili hem de analog örnekler şeklinde bulanık mantık teorisi ile ele alınmaktadır. İki katmanlı sinir ağından oluşan Bulanık A.R.T ağı, adaptiftir, öğrenmeyi eğiticisiz gerçekleştirmektedir ve eğitime ihtiyacı yoktur. Ağa ait iki katman F1, giriş veya karşılaştırma katmanı ile F2, çıktı veya sınıflandırma katmanı olarak adlandırılmaktadır. Bu iki katmana ait hücrelerin(nöronların) tümü birbirleri ile w_{ji} ağırlık vektörü ile bağlanmış durumdadır. Ağın ağırlık değeri w_{ji}, [0,1] aralığında değer almaktadır. i ve j indisleri sırayla F1 ve F2 katmanlarına ait olan hücreleri göstermektedir [90, 99].

Bulanık A.R.T. algoritması, A.R.T.1 algoritmasına göre bazı farklılıklar göstermektedir. A.R.T.1 algoritmasında, sadece ikili giriş değerlerini işlenebilirken, bulanık A.R.T. ikili olmayan giriş değerlerini de işleyebilmektedir. Bulanık A.R.T. algoritmasına ait parametreler, uygunluk eşik değeri (ρ), seçim parametresi (α) ve öğrenme oranıdır (β) [7].

Tablo 4.1: A.R.T. 1 ve bulanık A.R.T. karşılaştırması

	A.R.T. 1 (İkili)	Bulanık A.R.T. (Sürekli)
Küme seçimi	$T_j = \frac{\lvert I \cap W_j \rvert}{\alpha + \lvert W_j \rvert}$	$T_j = \frac{\lvert I \wedge W_j \rvert}{\alpha + \lvert W_j \rvert}$
Eşleşme kriteri	$\frac{\lvert I \cap W_j \rvert}{\lvert I \rvert} \geq \rho$	$\frac{\lvert I \wedge W_j \rvert}{\lvert I \rvert} \geq \rho$
Hızlı öğrenme	$W_j^{yeni} = I \cap W_j^{eski}$	$W_j^{yeni} = I \wedge W_j^{eski}$
	$\cap$ **:** mantıksal VE (AND)	$\wedge$ **:** Bulanık VE (AND) (min işlemi)

A.R.T. 1 ve bulanık A.R.T. algoritmalarının karşılaştırması Tablo 4.1' de görülmektedir. Tablodan da anlaşılacağı üzere, bulanık A.R.T. ağlarındaki bulanık kelimesi ağın kullanmış olduğu fonksiyonlardan türemiştir ancak tam anlamıyla bulanık değildir. Verileri kümelendirirken bulanık A.R.T. tam bir lider algoritma gibi davranmaktadır, örneğin, ilk giriş ilk küme ile karşılaştırılmakta ve eğer gerekirse yeni kümeler oluşturulmaktadır [96, 97]. Bulanık A.R.T. ağı kümeleri sabit değillerdir, yeni girişlere adapte olabilmektedirler [28].

Bulanık A.R.T. ağı, adaptif rezonans teorisine dayanmaktadır ve bulanık küme teorisi işlemleri üzerine kurulmuştur. Bundan dolayı girdi değerleri, ağ bağlantılarının ağırlıkları gibi [0,1] arasında yayılmaktadır. Kazanan hepsini alır kuramsal yapısındadır. Tam olarak bulanık yapıda olmamasına karşın bulanık ön adı kullandığı fonksiyonlardan türemiştir [98].

4.7.1. Bulanık A.R.T. özellikleri

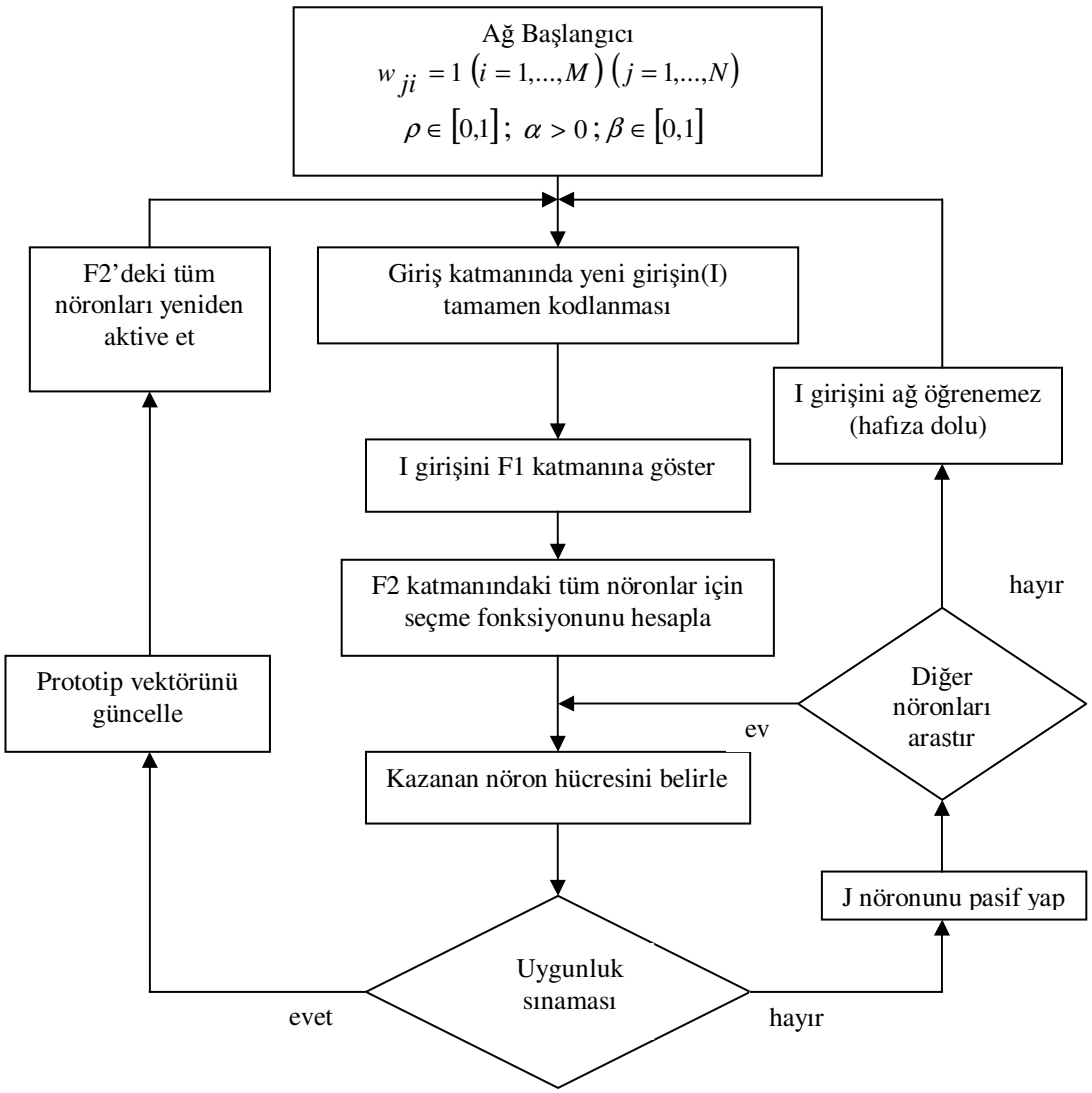
Bulanık A.R.T. ağlarının özellikleri şu şekilde özetlenebilir:

- A.R.T 1 ağına benzemektedir ancak [0,1] arasında bulanık üyelik değerlerini ifade eden sürekli girişler kullanılmaktadır. Ağrılıklar sürekli ve [0,1] arasındadır.
- İki çeşit öğrenme vardır, birincisi $\beta = 1$ olan hızlı öğrenmedir. İkincisi $\beta < 1$ olan yavaş öğrenmedir.

- Eğitilebilir ağırlıkların *(w)* bir katmanı vardır. Tüm ağırlıkların başlangıç değerleri 1'e eşittir.
- Giriş değerlerinin saklanan küme örneğine ne kadar yakın olması istendiğini belirten bir uygunluk parametresi (ρ) vardır. Bu değer ile küme içindeki elemanların birbirlerine ne ölçüde benzerlik göstereceği belirlenmiş olmaktadır.
- Kümelendirme düğümlerinde giriş değerinin paydasında sabit bir α bulunmaktadır.

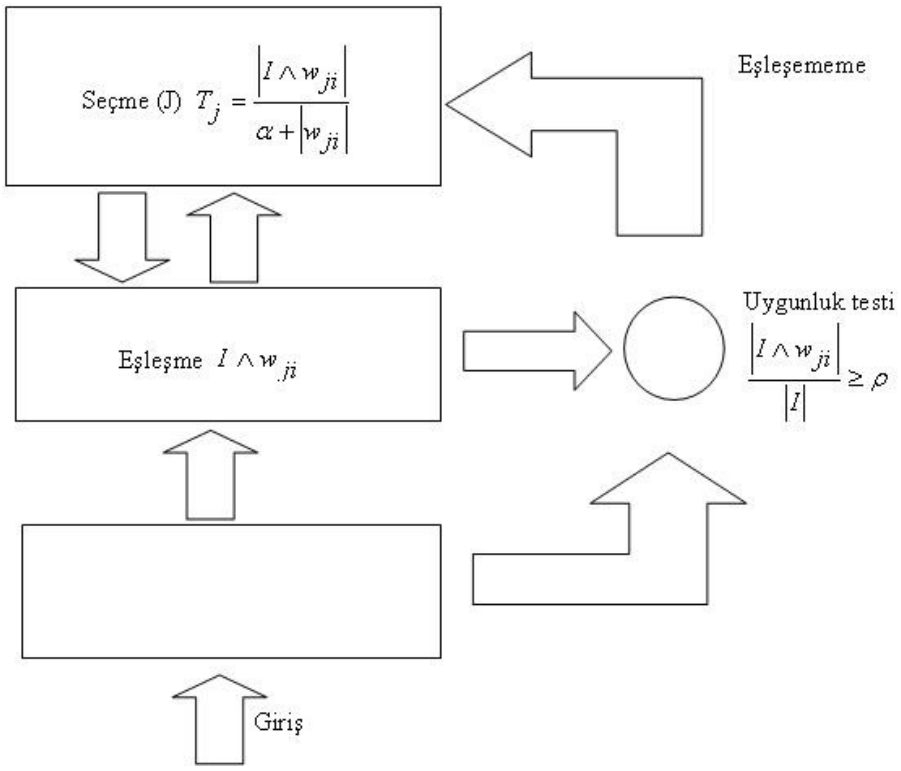
4.7.2. Bulanık A.R.T. akış şeması ve algoritması

Bulanık A.R.T.' ye ait akış şeması Şekil 4.14'de gösterilmektedir.



Şekil 4.14: Bulanık A.R.T. akış şeması [99]

Yukarıda algoritması verilen bulanık A.R.T. algoritmasına ait olan yapının basit bir temsili Şekil 4.15'de gösterilmektedir.



Şekil 4.15: Bulanık A.R.T. mimarisi [100]

Her bir giriş vektörü (I) m-boyutludur, (I_1, I_2,…,I_m) temsil etmektedir ve her bir I_i [0,1] aralığında değer almaktadır.

Her bir küme (j), W_j= w_{j1}, w_{j2},…,w_{jm} ağırlıklarını temsil etmektedir. Potansiyel küme sayısı başlangıçta 1'dir daha sonra giriş vektörü ve algoritmanın parametrelerine göre artış göstermektedir (j=1, 2,…,N). Bulanık A.R.T. ağırlık vektörü, hem aşağıdan yukarı ağırlık vektörlerini hem de yukarıdan-aşağı ağırlık vektörlerini göstermektedir [34].

Bu durumda algoritmanın adımları aşağıdaki gibi özetlenmektedir.

Adım 1: Normalizasyon: I giriş vektörü bağıntı 4.7' ye göre normalize edilmektedir.

$$I_{ij} = \frac{I(i,j) - \min(j)}{\max(j) - \min(j)} \tag{4.7}$$

min(j): j. nitelik değerinin tüm veri kümesinde aldığı en küçük değeri; max(j), j. nitelik değerinin tüm veri kümesinde aldığı en büyük değeri ifade etmektedir. Normalizasyon işleminden sonra tüm giriş değerleri 0-1 arasındaki sayılar ile ifade edilebilmektedir.

Adım 2: Başlangıç: Algoritmaya ait tüm parametrelere başlangıç değerlere atanmaktadır.

- Uygunluk parametresi ρ $(0 \leq \rho \leq 1)$: aynı kümede yer alacak olan kayıtların birbirlerine olan benzerlik derecelerini belirlemektedir. F2' de oluşturulan küme sayısından sorumludur. Uygunluk parametresi 1'e yaklaştıkça daha az eleman içeren daha çok küme oluşmaktadır. Uygunluk parametresi daha küçük değerler aldığında ise oluşacak olan küme sayısı azalmakta, kümelerin eleman sayıları ise artmaktadır.
- Seçme parametresi α ($\alpha > 0$): küme seçiminde etkilidir.
- Öğrenme oranı $(\beta \in [0,1])$: ağ adaptasyonunun hızını kontrol eder. Eğer β = 1 olursa, hızlı öğrenme gerçekleştirilmektedir. β' nın 0' a yaklaşması öğrenme oranını azaltır. Eğer veri karmaşıksa veya kümelerin hızlı seçimi yanlışa neden olacaksa yavaş öğrenme tercih sebebidir.

Adım 3: Ağırlık vektörüne başlangıç değerleri atanır:

$W_j = (w_{j1},...,w_{jM})$, başlangıçta tüm ağırlıklar bağıntı 4.8' de ifade edildiği gibi 1 olarak seçilmektedir.

$$w_{j1} = ... = w_{jM} = 1. \tag{4.8}$$

Ağırlık vektörü uzun dönemli belleği ortaya çıkartmaktadır. Potansiyel kümelerin sayısı N (j=1,…,N) rastgele seçilmiştir. Başlangıçta, her bir kümenin bağımsız

olduğu söylenir. Alternatif olarak, başlangıç ağırlıkları w_{ji} 1'den daha büyük alınabilir. Daha büyük ağırlıklar bağımsız düğümlerin seçiminde sisteme karşıt eğilim gösterebilmektedirler. Ağırlık vektörü $W_j = (w_{j1},...,w_{jM})$ hem aşağıdan yukarı hem de yukarıdan aşağı ağırlık vektörlerini kapsamaktadır. Küme seçildikten sonra ağırlık vektörü bağımlı hale gelir.

Adım 4: Giriş vektörünün ağa gösterimi: [0,1] aralığında normalize edilmiş giriş değerlerinden oluşan giriş vektörü (I) ağa gösterilmektedir.

$$I = i_1,...,i_M ;\ i_i \in [0,1]$$

Adım 5: Küme seçimi gerçekleştirilir. Her küme için, seçme fonksiyonu (Tj) bağıntı 4.9 ile tanımlanmaktadır.

$$T_j(I) = \frac{|I \wedge w_j|}{\alpha + |w_j|} \tag{4.9}$$

Burada, Λ bulanık AND (ve) operatörüdür, $(x \wedge y) = \min(x_i, y_i)$ olarak belirlenmiştir.

Adım 6: Tüm kümeler için hesaplanmış olan seçme fonksiyonu değerleri arasından en büyük değer seçilerek; bu seçme fonksiyonuna sahip olan küme belirlenmektedir.

$$T_j = \max\{T_j : j = 1,...,N\} \tag{4.10}$$

Adım 7: Rezonans sınaması (Uygunluk sınaması) gerçekleştirilir. En yüksek seçme fonksiyonuna sahip olan küme için uygunluk sınaması gerçekleştirilir. Bağıntı 4.11 sağlanıyorsa rezonans oluşmaktadır ve adım 9' a gidilir.

$$\frac{|I \wedge w_j|}{|I|} \geq \rho \tag{4.11}$$

Bir önceki adımda seçilmiş olan prototip, uygunluk sınamasına tabi tutulur. Uygunluk sınaması, kazanan prototip ile mevcut giriş çifti arasındaki benzerliği karşılaştırmaktadır. Uygunluk parametresi kullanıcı tanımlıdır ve önceden belirlenmiştir. Eğer prototip uygunluk sınamasını geçerse verilmiş olan giriş örneğine adapte edilmektedir.

Adım 8: Eşleşememe durumunda:

$$\frac{\left|I \wedge w_j\right|}{\left|I\right|} < \rho \tag{4.12}$$

T_θ= -1 olarak güncellenir ve bağıntı 4.10 ile yeni bir J kümesi seçilir. Seçme işlemi 4.11'i sağlayan bir J kümesi bulunana kadar devam eder.

Eğer prototip uygunluk sınamasını geçemezse, mevcut prototip mevcut giriş örneği için pasif yapılmakta ve F2 katmanındaki diğer prototiplerden biri sınamayı geçene kadar uygunluk sınamasına tabi tutulmaktadırlar. Eğer hiç biri sınamayı geçemez ise eşleşememe (reset) meydana gelmekte; mevcut giriş örneği için yeni bir prototip yaratılmaktadır.

Adım 9: Öğrenme: ağın öğrenmesi ağırlık güncellemesi ile gerçekleştirilir. Ağrılık vektörü W_j bağıntı 4.13' e göre güncellenir.

$$W_j^{(yeni)} = \beta(I \wedge W_j^{(eski)}) + (1-\beta)W_j^{(eski)} \tag{4.13}$$

$\beta\,(0 \leq \beta \leq 1)$ kümeleme hızını kontrol etmektedir. Hızlı öğrenme seçeneği için $\beta = 1$ seçilmelidir. β ' nın 0' a yaklaşması öğrenme oranını azaltmaktadır. Eğer giriş verilerinin karmaşık veya kümelerin seçim hızının yüksek olması sınıflama hatasına neden olacaksa öğrenmenin yavaş olması tercih edilmektedir. Ancak öğrenmenin yavaş olması programın çözüm süresini uzatmaktadır.

Adım 10: Tekrarla: Adım 4’ e gidilir: Güncellemeden sonra, tüm prototipler yeniden aktive edilir ve algoritma bir sonraki girişle devam eder [27, 28, 94].

5. İYİLEŞTİRİLMİŞ BULANIK A.R.T.

5.1. Giriş

Bu bölümde, tez kapsamında bulanık A.R.T. (F.A.R.T.) algoritmasına alternatif olarak önerilmiş olan İyileştirilmiş Bulanık A.R.T. (İ.F.A.R.T.) algoritmasının çalışma şekli anlatılmaktadır. Önerilen yöntem ile elde edilen küme sonuçları, F.A.R.T. ve S.O.M. algoritmasından elde edilen küme sonuçları ile karşılaştırılmaktadır.

5.2. İyileştirilmiş Bulanık A.R.T. (İ.F.A.R.T.)

Üçüncü bölümde de anlatıldığı üzere, kümeleme işleminde asıl önemli olan nokta doğru ve geçerli kümelemenin gerçekleştirilebilmesidir. Kümelemenin geçerli olabilmesi için farklı kümelerdeki elemanlar birbirlerine benzemezken, aynı kümedeki elemanların birbirlerine yüksek oranlarda benzemeleri gerekmektedir. Her kümeleme algoritmasından sonra elde edilen kümelerin doğru ve geçerli kümeler oldukları söylenemez. Bu bağlamda, tez kapsamında F.A.R.T. ile elde edilen kümeleme sonuçlarının iyileştirilmesi adına, kümeler üzerinde bir analiz işlemi gerçekleştirilmiştir. Bu amaca yönelik olarak, farklı uygunluk parametreleri ile başlatılan F.A.R.T. algoritması işletildikten sonra elde edilen kümeler incelenmiştir. Analizler sonucunda, farklı verilerden elde edilen sonuç kümelerinin birbirlerinden uzak oluşturulamadıkları, küme sınırlarının iç içe geçmiş oldukları gözlenmiştir.

F.A.R.T. algoritması, dördüncü bölümde de ayrıntılı olarak anlatıldığı gibi, başlangıçta belirlenen uygunluk parametresi oranında birbirine benzeyen girişleri aynı kümeye dahil eden bir algoritmadır. F.A.R.T. algoritmasının başlangıçta seçilen benzerlik oranına ve verilerin sisteme sunuluş sırasına çok duyarlı olmasından kaynaklı doğru ve geçerli bir kümeleme garanti edilememektedir.

Bu sorunun giderilmesi ve gerçekte ait olmaları gereken kümelerden farklı kümelere

yerleştirilmiş giriş verilerinin tespit edilebilmesi için; her giriş verisinin oluşan her kümeye üyelik derecesi hesaplanmaktadır. Hangi kümeye gerçekte ne kadar ait oldukları üyelik dereceleri temel alınarak incelenmektedir. Üyelik derecesi hesaplanırken kümeyi temsil eden nokta küme merkezi olarak kabul edilmiştir. Üyelik dereceleri incelendikten sonra, giriş verilerinin maksimum üyelik derecesi ile bağlı olduğu kümede yer almadıkları yani yanlış kümelendikleri saptanmıştır. Bu durumda olan giriş verileri belirlenerek maksimum üyelik gösterdikleri doğru kümelere taşınmışlardır. Böylelikle F.A.R.T. algoritması ile elde edilmiş olan kümeler değişmeye başlamıştır.

İ.F.A.R.T. algoritmasının adımları şu şekildedir:

Adım 1: Normalizasyon: Giriş verileri (I vektörü) bağıntı 5.1' e göre normalize edilmektedir.

$$NI_{i,j} = \frac{I_{i,j} - \min(j)}{\max(j) - \min(j)} \tag{5.1}$$

min(j): j niteliğinin gösterdiği sütundaki tüm değerler arasındaki en küçük değeri;
max(j): j niteliğinin gösterdiği sütundaki tüm değerler arasındaki en büyük değeri ifade etmektedir. Normalizasyon adımından sonra tüm giriş değerleri [0,1] arasında yer almaktadır.

$NI_{i,j}$: normalize edilmiş giriş değeri

i: aday giriş vektörünün indisi
j: nitelik indisi
n: nitelik sayısı
s: küme indisi

Adım 2: Algoritmaya ait tüm parametrelere başlangıç değerleri atanmaktadır.

- Uygunluk parametresi ρ $(0 \leq \rho \leq 1)$: aynı kümede yer alacak olan kayıtların birbirlerine olan benzerlik derecelerini belirlemektedir. Uygunluk parametresi 1'e yaklaştıkça daha az eleman içeren daha çok küme oluşmaktadır. Uygunluk

parametresi daha küçük değerler aldığında ise oluşacak olan küme sayısı azalmakta, kümelerin eleman sayıları ise artmaktadır.

- Seçme parametresi α ($\alpha > 0$): küme seçiminde etkilidir.
- Öğrenme oranı $(\beta \in [0,1])$: ağ adaptasyonunun hızını kontrol eder. eğer β = 1 olursa, hızlı öğrenme gerçekleştirilmektedir. β' nın 0' a yaklaşması öğrenme oranını azaltır.

Adım 3: Ağırlık vektörüne başlangıç değerleri atanmaktadır:
Başlangıçta tüm ağırlıklar bağıntı 5.2' de ifade edildiği gibi 1 olarak seçilmektedir.

$$w_{i,j,s}(0) = 1 \text{ ve } s=1 \tag{5.2}$$

Adım 4: Giriş vektörünün ağa gösterimi: [0,1] aralığında normalize edilmiş giriş değerlerinden oluşan giriş vektörünün (NI) sıradaki elemanı ağa gösterilmektedir.

Adım 5: Küme seçimi gerçekleştirilir. Her küme için, seçme fonksiyonu ($T_{i,s}$) bağıntı 5.3 ile tanımlanmaktadır.

$$T_{i,s}(NI) = \frac{\sum_{j=1}^{n} NI_{i,j} \wedge w_{i,j,s}}{\alpha + \sum_{j=1}^{n} w_{i,j,s}} \tag{5.3}$$

Adım 6: Tüm kümeler için hesaplanmış olan seçme fonksiyonu değerleri arasından en büyük değer seçilerek; bu seçme fonksiyonuna sahip olan küme belirlenmektedir.

$$T^{*} = \max\{T_{i,s}, s = 1,2,...,m\} \tag{5.4}$$

Adım 7: Rezonans sınaması (Uygunluk sınaması) gerçekleştirilir. En yüksek seçme fonksiyonuna sahip olan küme için uygunluk sınaması gerçekleştirilir. Bağıntı 5.5 sağlanıyorsa rezonans oluşur ve adım 9' a gidilir.

$$M_{i,s}\left(T^*\right)=\frac{\sum_{j=1}^{n}\left(NI_{i,j}\wedge w_{i,j,s}\right)}{\sum_{j=1}^{n}NI_{i,j}}$$

$$M_{i,s}\left(T^*\right)\geq\rho \tag{5.5}$$

Adım 8: Eşleşememe durumunda:

$$M_{i,s}\left(T^*\right)<\rho \tag{5.6}$$

$M_{i,s}$ = -1 olarak güncellenir ve bağıntı 5.4 ile yeni bir küme seçilir. Seçme işlemi bağıntı 5.5' i sağlayan bir küme bulunana kadar devam etmektedir.

Eğer prototip uygunluk sınamasını geçemez ise, mevcut prototip mevcut giriş örneği için pasif yapılmakta diğer prototiplerden biri sınamayı geçene kadar uygunluk sınamasına tabi tutulmaktadırlar. Eğer hiç biri sınamayı geçemez ise eşleşememe (reset) meydana gelmekte; mevcut giriş örneği için yeni bir prototip yaratılmaktadır.

Adım 9: Öğrenme: ağın öğrenmesi ağırlık güncellemesi ile gerçekleştirilmektedir. Ağrılık vektörü (W) bağıntı 5.7' ye göre güncellenmektedir.

$$W_{i,j,s}^{(yeni)}=\beta(NI_{i,j}\wedge W_{i,j,s}^{(eski)})+(1-\beta)W_{i,j,s}^{(eski)} \tag{5.7}$$

Adım 10: Tekrarla: Adım 4' e gidilir: Güncellemeden sonra, tüm prototipler yeniden aktive edilir ve algoritma bir sonraki girişle devam etmektedir.

Adım 11: Oluşan kümelerin merkez noktalarının bulunması bağıntı 5.8 ile gerçekleştirilmektedir [29].

$$V_s=\frac{\sum_{i=1}^{elsay}I_{i,s}}{elsay} \tag{5.8}$$

burada, $I_{i,s}$: (i=1, 2,…, elsay) s kümesinin elemanlarını ifade etmektedir.

Adım 12: Giriş verilerinin oluşan kümelere üyelik derecelerinin hesaplanması. i. giriş verisinin s. kümeye üyeliği bağıntı 5.9'a göre hesaplanmaktadır [65].

$$\mu_{i,s} = \frac{\left[\frac{1}{\|I_i, C_s\|}\right]^{1/(q-1)}}{\sum_{k=2}^{K}\left[\frac{1}{\|I_i, C_k\|}\right]^{1/(q-1)}} \tag{5.9}$$

Burada *K* küme sayısını ifade etmektedir, q bir sabittir ve 2 olarak seçilmiştir.

	1	2	3
1	0.93427	0.039055	0.026677
2	0.81785	0.10893	0.073222
3	0.83415	0.098465	0.067388
4	0.794	0.12316	0.082838
5	0.91512	0.050321	0.034561
6	0.73231	0.1594	0.10829
7	0.83284	0.099283	0.067877
8	0.9675	0.01941	0.013091
9	0.72291	0.16498	0.11211
10	0.84083	0.095309	0.063859
11	0.80162	0.11772	0.080658
12	0.88539	0.068601	0.046009
13	0.80339	0.11735	0.079263
14	0.71379	0.16837	0.11785
15	0.67415	0.1895	0.13635
16	0.62513	0.21772	0.15715
17	0.75651	0.14306	0.10043
18	0.93249	0.040154	0.027359
19	0.68934	0.18494	0.12572
20	0.83574	0.097376	0.066889
21	0.79762	0.12184	0.080532
22	0.85584	0.085742	0.058418
23	0.7806	0.12824	0.091162
24	0.82489	0.10602	0.069091

Şekil 5.1: İris veri kümesi için üyelik derecesi matrisi (U_{155x3})

Adım 13: Giriş verileri, Şekil 5.1' deki U matrisindeki gibi maksimum üyelik gösterdikleri kümelere taşınmaktadırlar.

5.3. F.A.R.T., İ.F.A.R.T. ve S.O.M. Algoritmalarının Karşılaştırılması

Algoritmalar, oluşan kümelerin doğruluk dereceleri ve çalışma süreleri açısından karşılaştırılmaktadırlar. Birinci karşılaştırma ölçütü olarak ayrıt 3.4' de anlatılan kümeleme hata payı kullanılmaktadır. Bu hata ölçümü ifadesini kısaca yinelemek gerekirse, küme içindeki benzerlik oranı arttıkça ve aynı zamanda, kümeler arasındaki uzaklık da arttıkça azalan bir orandır.

İkinci karşılaştırma ölçütü olarak ise, algoritmaların çalışmaya başladıkları zaman ile tamamlandıkları zaman arasındaki süre saniye cinsinden ölçülerek, ne kadar sürede tamamlandıkları hesaplanmaktadır.

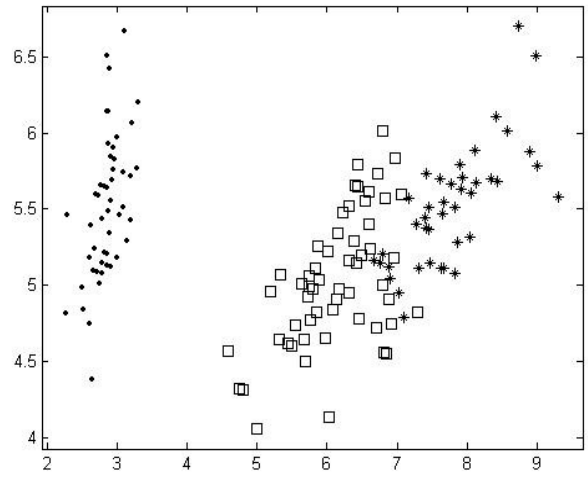
5.4. F.A.R.T., İ.F.A.R.T. ve S.O.M. Algoritmalarından Elde Edilen Kümeler

İ.F.A.R.T. algoritmasının çalışması gerçek veri kümeleri üzerinde incelenmiştir. U.C.I. veri tabanından [101] seçilmiş olan veri kümeleri, sınıflama veya kümeleme problemlerinde sık kullanılan veri kümeleridir. Ayrıca kategorik nitelik değerleri içermeyen sayısal nitelik değerleri olan veri kümeleridir.

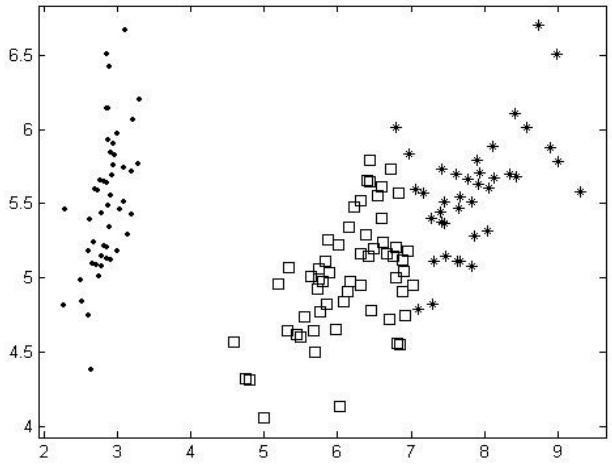
Tez kapsamında gerçekleştirilen tüm çizimler için "*n*" boyutlu uzay, temel bileşenler analizi yöntemi ile 2 boyutta gösterilmektedir. Boyut indirgeme yönteminden sonra elde edilen boyutlar nitelik değerlerinden herhangi birini doğrudan temsil etmediğinden "bileşen 1" ve bileşen 2" gibi düşünülmektedir.

İris veri kümesi süsen çiçeğine ait "petal boyu", "petal eni" ve "tür tipi" özelliklerinin bulunduğu 155 adet giriş verisinden oluşmaktadır. İris veri kümesi yani süsen çiçeği verileri "iris setosa", "iris versicolor" ve "iris virginica" olmak üzere 3 sınıfa ayrılmışlardır. Bu nedenle, bu veri kümesi üzerinde kümeleme gerçekleştirirken küme sayısı 3 olarak belirlenmiştir. F.A.R.T. ve İ.F.A.R.T. için başlangıç ρ=0.6; α=0.001; β=0.1 olarak seçilmiştir. Şekil 5.2 ve 5.3' de sırası ile

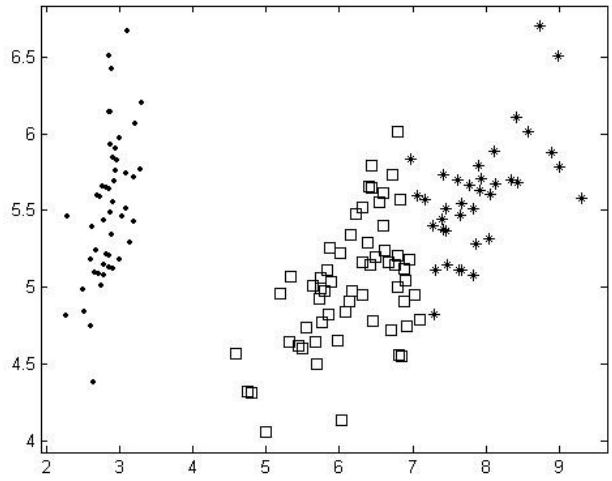
F.A.R.T. ve İ.F.A.R.T. sonucu elde edilen kümeler gözlenmektedir. Buna ek olarak kümeleme konusunda yapay sinir ağı algoritmaları içerisinde en sık kullanılan algoritma olan S.O.M. algoritmasından elde edilen kümeler de Şekil 5.4' de karşılaştırılmak üzere gösterilmektedir.



Şekil 5.2: Iris için F.A.R.T. ile elde edilen kümeler



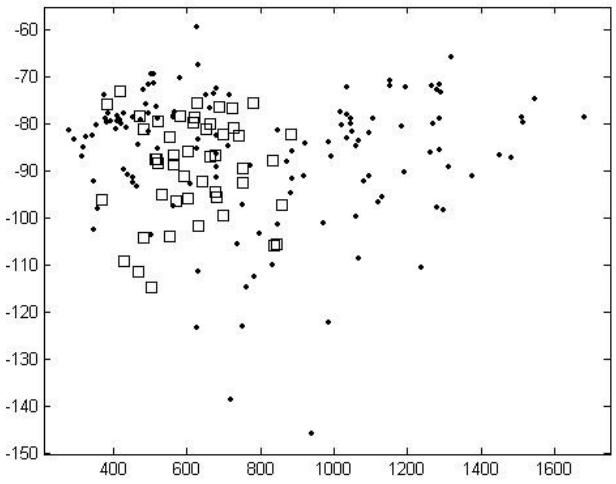
Şekil 5.3: Iris için İ.F.A.R.T. ile elde edilen kümeler



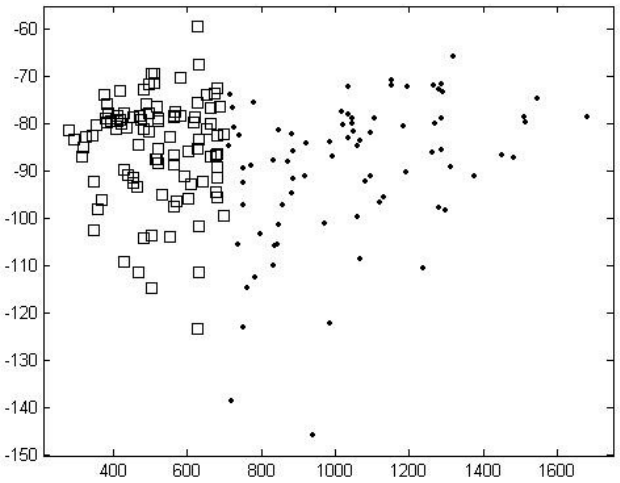
Şekil 5.4: Iris için S.O.M. ile elde edilen kümeler

Şekil 5.2 incelendiğinde F.A.R.T. ile elde edilen kümelerin iyi ayrılmış kümeler olmadıkları gözlenmektedir. Kümeler birbirlerinin sınırlarına geçmiş şekildedir. Şekil 5.3 incelendiğinde İ.F.A.R.T. algoritması ile verilerin maksimum üye oldukları kümelere taşınması sonucunda elde edilen kümelerin, F.A.R.T. ile elde edilen kümelerden çok daha yüksek nitelikli kümeler oldukları gözlenmektedir. İ.F.A.R.T. algoritması Şekil 5.4' de S.O.M. ile karşılaştırıldığında çok benzer bir kümeleme sonucu elde edildiği gözlenmektedir.

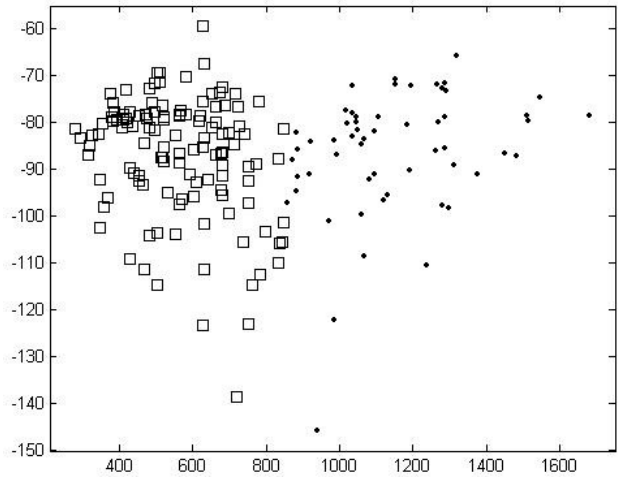
Wine veri kümesinde 13 nitelik değeri ve 178 veri örneği bulunmaktadır. F.A.R.T. ve İ.F.A.R.T. algoritmaları için başlangıç parametreleri p=0.5; alfa=0.001; beta=0.05 seçilmiştir. F.A.R.T., İ.F.A.R.T. ve S.O.M. algoritmaları sonucunda elde edilen kümeler sırası ile Şekil 5.5, 5.6 ve 5.7'de verilmektedir.



Şekil 5.5: Wine için F.A.R.T. ile elde edilen kümeler

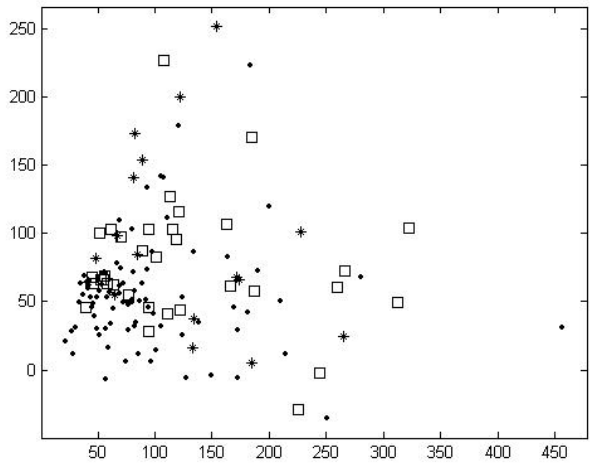


Şekil 5.6: Wine için İ.F.A.R.T. ile elde edilen kümeler

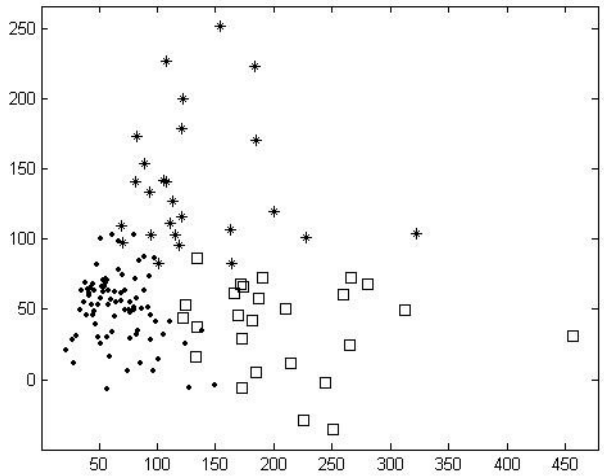


Şekil 5.7: Wine için S.O.M. ile elde edilen kümeler

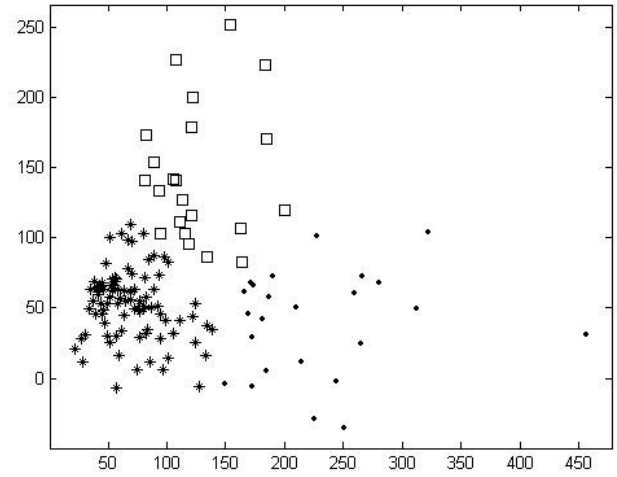
Hepatitis veri kümesinde 19 nitelik değeri ve 155 veri örneği bulunmaktadır. F.A.R.T. ve İ.F.A.R.T. algoritmaları için başlangıç parametreleri p=0.4; alfa=0.001; beta=0.08 seçilmiştir. F.A.R.T., İ.F.A.R.T. ve S.O.M. algoritmaları sonucunda elde edilen kümeler sırası ile Şekil 5.8, 5.9 ve 5.10'da verilmektedir.



Şekil 5.8: Hepatitis için F.A.R.T. ile elde edilen kümeler

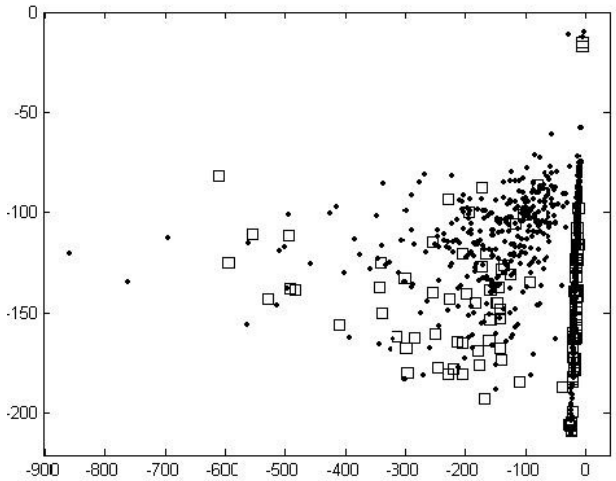


Şekil 5.9: Hepatitis için İ.F.A.R.T. ile elde edilen kümeler

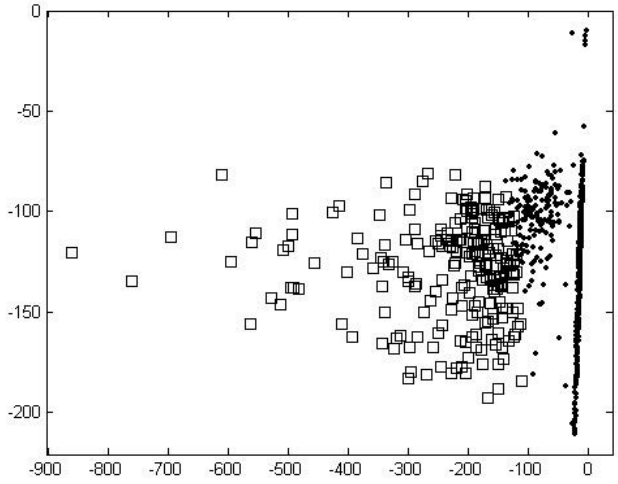


Şekil 5.10: Hepatitis için S.O.M. ile elde edilen kümeler

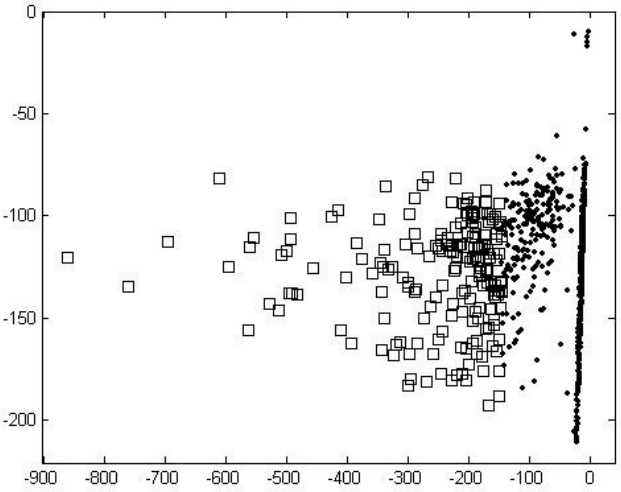
Pima Indians Diabetes veri kümesinde 8 nitelik değeri ve 768 veri örneği bulunmaktadır. F.A.R.T. ve İ.F.A.R.T. algoritmaları için başlangıç parametreleri p=0.3; alfa=0.001; beta=0.03 seçilmiştir. F.A.R.T., İ.F.A.R.T. ve S.O.M. algoritmaları sonucunda elde edilen kümeler sırası ile Şekil 5.11, 5.12 ve 5.13'de verilmektedir.



Şekil 5.11: Pima Indians Diabetes için F.A.R.T. ile elde edilen kümeler

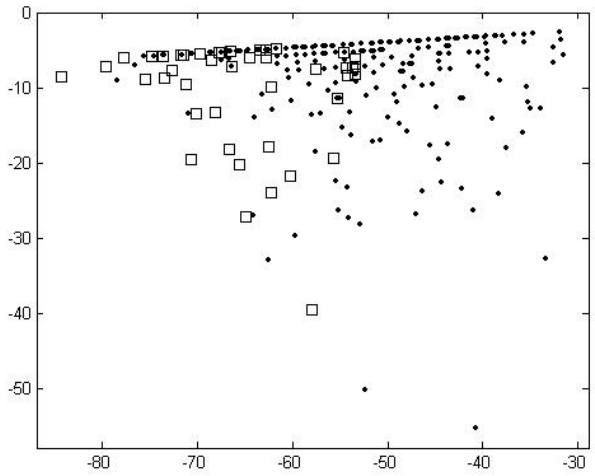


Şekil 5.12: Pima Indians Diabetes için İ.F.A.R.T. ile elde edilen kümeler

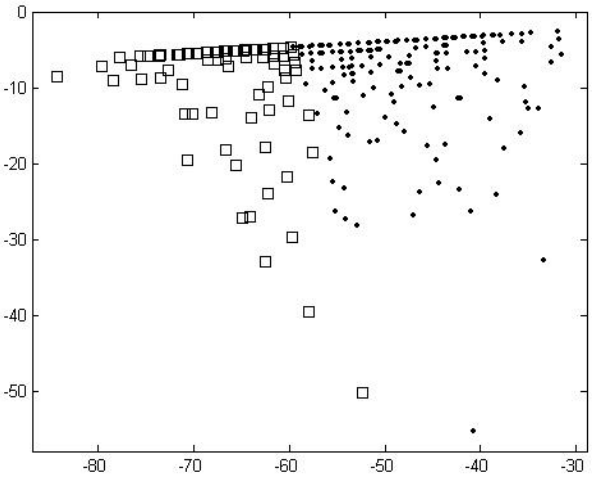


Şekil 5.13: Pima Indians Diabetes için S.O.M. ile elde edilen kümeler

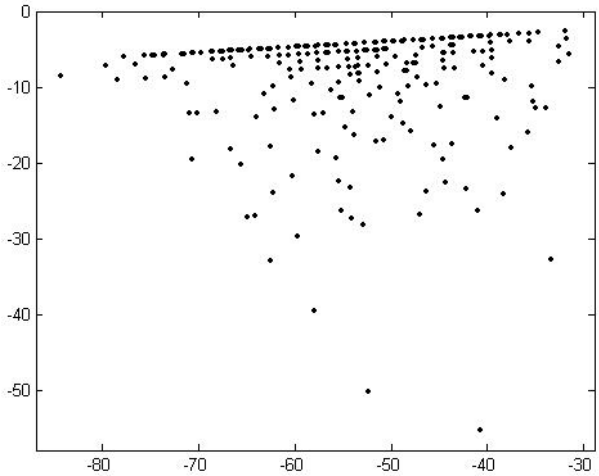
Haberman's Survival veri kümesinde 4 nitelik değeri ve 306 veri örneği vardır. F.A.R.T. ve İ.F.A.R.T. algoritmaları için başlangıç parametreleri p=0.55; alfa=0.001; beta=0.01 seçilmiştir. F.A.R.T., İ.F.A.R.T. ve S.O.M. algoritmaları sonucunda elde edilen kümeler sırası ile Şekil 5.14, 5.15 ve 5.16'da verilmektedir.



Şekil 5.14: Haberman's Survival için F.A.R.T. ile elde edilen kümeler

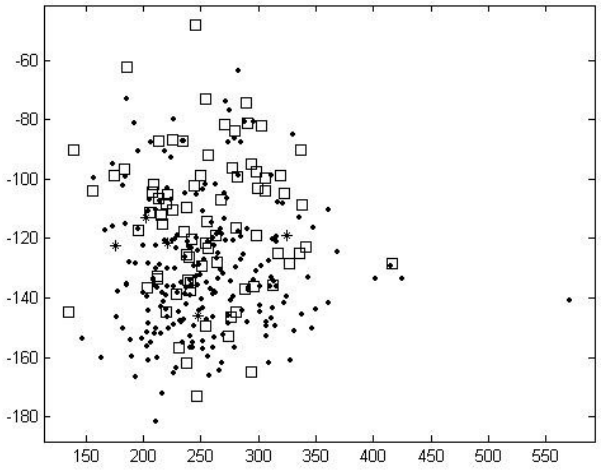


Şekil 5.15: Haberman's Survival için İ.F.A.R.T. ile elde edilen kümeler

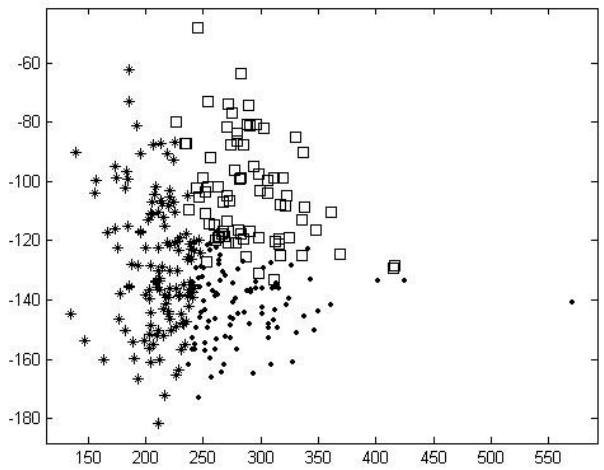


Şekil 5.16: Haberman's Survival için SOM ile elde edilen kümeler

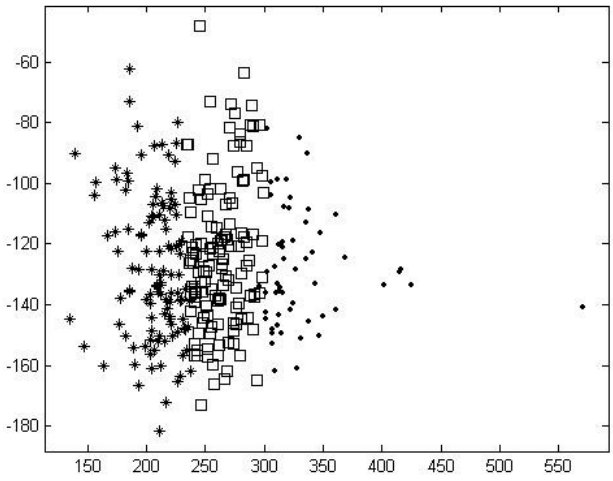
Heart-Disease-Cleveland veri kümesinde 14 nitelik değeri ve 303 veri örneği bulunmaktadır. F.A.R.T. ve İ.F.A.R.T. algoritmaları için başlangıç parametreleri p=0.7; alfa=0.001; beta=0.01 seçilmiştir. F.A.R.T., İ.F.A.R.T. ve S.O.M. algoritmaları sonucunda elde edilen kümeler sırası ile Şekil 5.17, 5.18 ve 5.19'da verilmektedir.



Şekil 5.17: Heart-Disease-Cleveland için F.A.R.T. ile elde edilen kümeler



Şekil 5.18: Heart-Disease-Cleveland için İ.F.A.R.T. ile elde edilen kümeler



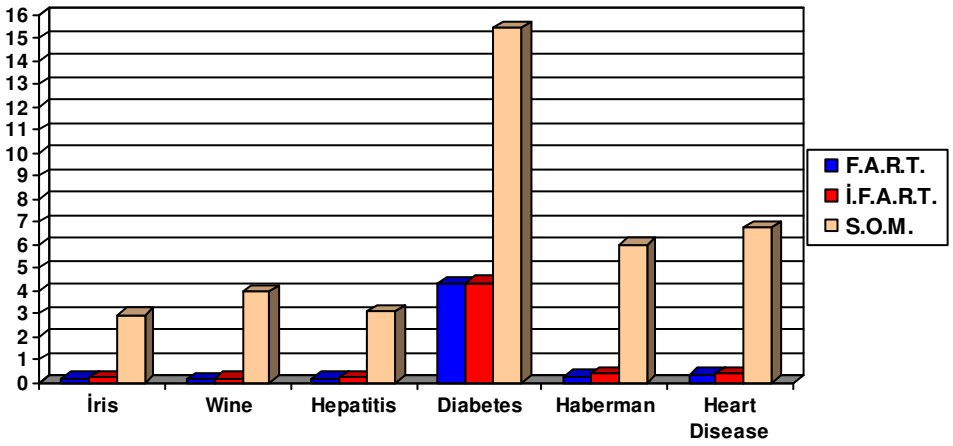
Şekil 5.19: Heart-Disease-Cleveland için SOM ile elde edilen kümeler

5.5. F.A.R.T., İ.F.A.R.T. ve S.O.M. Algoritmalarına Ait Hata Payları ve Kümeleme Hızları

Yukarıdaki örneklerde F.A.R.T. ve İ.F.A.R.T. algoritmaları 1 tekrarda, S.O.M. algoritması ise 10 tekrarda gerçekleştirilmiştir. Bu örnek veri kümelemelerinin gerçekleşebilmesi için gerekli olan algoritma çalışma süreleri Tablo 5.1'de ve Şekil 5.20'deki grafikte gösterilmektedir.

Tablo 5.1: F.A.R.T., İ.F.A.R.T., S.O.M. algoritmalarının çalışma süreleri

Veri Kümeleri/ Alg.Çalışma Zamanları (saniye)	F.A.R.T	İ.F.A.R.T.	S.O.M.
Iris	0.1720	0.2350	2.9840
Wine	0.1410	0.1720	3.9530
Hepatitis	0.2109	0.2350	3.0940
Pima Indians Diabetes	4.2960	4.3430	15.4220
Haberman's Survival	0.3000	0.3910	6
Heart-Disease-Cleveland	0.3900	0.4220	6.7350



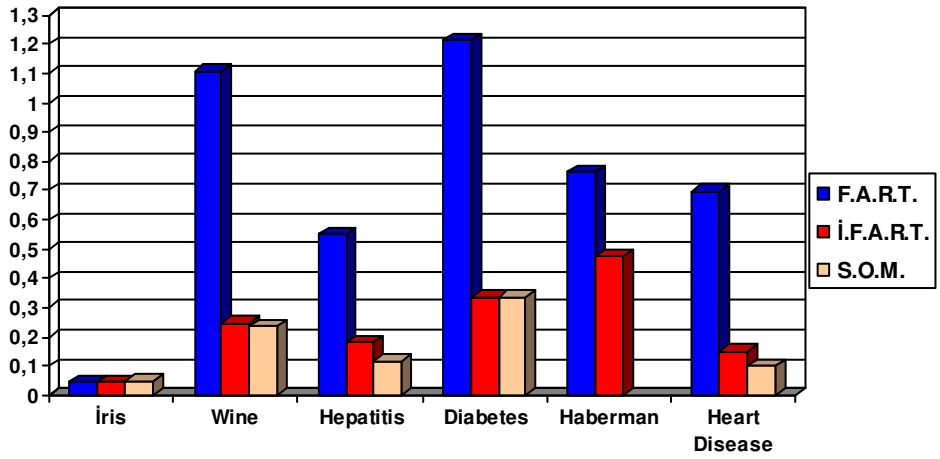
Şekil 5.20: Çalışma sürelerinin grafik gösterimi

Tablo 5.1' deki veriler değerlendirildiğinde, İ.F.A.R.T. ve F.A.R.T. algoritmalarının S.O.M. algoritmasından çok daha kısa bir çalışma zamanında tamamlandıkları görülmektedir. İ.F.A.R.T. algoritması ve F.A.R.T. algoritmasının çalışma zamanları arasında ise çok az farklılık olduğu gözlenmektedir. İ.F.A.R.T., kümelemede en sık kullanılan yapay sinir ağı algoritması S.O.M.' a göre çok daha hızlı kümeleme gerçekleştirebilmektedir.

Örnek verilerin kümelenmesinde oluşan kümeleme hata payları Tablo 5.2'de ve Şekil 5.21'deki grafikte gösterilmektedir.

Tablo 5.2: Hata kestirim indeksi

Veri Kümeleri / Hata oranı	F.A.R.T.	İ.F.A.R.T.	S.O.M.
İris	0.0470	0.0457	0.0497
Wine	1.1086	0.2457	0.2365
Hepatitis	0.5498	0.1815	0.1156
Pima Indians Diabetes	1.2123	0.3373	0.3354
Haberman's Survival	0.7632	0.4746	-
Heart-Disease-Cleveland	0.6997	0.1527	0.1001



Şekil 5.21: Kümeleme hata oranlarının grafik gösterimi

Tablo 5.2' deki veriler değerlendirildiğinde, İ.F.A.R.T. algoritmasının F.A.R.T. algoritmasına göre daha doğru ve geçerli kümeler oluşturduğu gözlenmektedir. Kümelemeyi daha az hata payı ile tamamlamaktadır. S.O.M. algoritmasına göre daha yüksek hata payı ile kümeleme yapmaktadır ancak bu hata payları arasında çok az farklılık bulunmaktadır.

Yapılan deneylerde, F.A.R.T. algoritmasından sonra oluşan kümelerdeki veri nesnelerinden kaçar tanesinin İ.F.A.R.T. sonucu kümeler arasında yer değiştirdiği Tablo 5.3' de görülmektedir.

Tablo 5.3: Yanlış kümelenen veri nesnesi sayısı

Veri Kümesi / Veri Nesnesi Sayısı	Toplam	Yanlış kümelenen
İris	150	6
Wine	178	90
Hepatitis	155	85
Pima Indians Diabetes	768	615
Haberman's Survival	306	232
Heart-Disease-Cleveland	303	136

6. KÜMELEME DENEYLERİ

6.1. Giriş

Bu bölümde, farklı veri kümeleri üzerinde standart k-means algoritmasının uygulanması sonucunda elde edilen kümeler ve İyileştirilmiş Bulanık A.R.T (İ.F.A.R.T.) ile başlangıç küme merkezleri belirlenen k-means algoritmasının uygulanması sonucunda elde edilen kümeler karşılaştırmalı olarak sunulmaktadır.

6.2. Deneylerde Kullanılan Veri Kümeleri

Tez kapsamında yapılan deneylerde UCI (California, Irvire Üniversitesi, Makine öğrenimi veri tabanı) kütüphanesinden [101] alınan gerçek veri kümeleri ve Kanada Alberta Üniversitesi Bilgisayar Bilimleri bölümünde Osmar Zaïane ve Yaling Pei tarafından hazırlanmış veri tabanlarından [102] alınan Web Logs, Documents_Similarity, Mars, Image Extraction ve bu amaçla üretilen yapay veri kümeleri kullanılarak kümeleme işlemi gerçekleştirilmiştir. Deneylerde kullanılan gerçek veri kümeleri ve yapay veri kümelerine ait özellikler sırası ile Tablo 6.1 ve Tablo 6.2'de gösterilmektedir.

Tablo 6.1: Deneylerde kullanılan gerçek veri kümeleri

Veri Kümeleri	Örnek Sayısı	Nitelik Sayısı
Iris	150	4
Wine	178	13
Hepatitis	155	19
Pima Indians Diabetes	768	8
Haberman's Survival	306	3
Heart-Disease-Cleveland	303	14
Ruspini	75	2
Letter Recognition	20000	16

Tablo 6.2: Deneylerde kullanılan yapay veri kümeleri

Veri Kümeleri	Örnek Sayısı	Nitelik Sayısı
Web Logs	250	2
Documents Similarity	250	2
Mars	250	2
Image Extraction	250	2
Yapay Veri 1	1000	2
Yapay Veri 2	2000	2
Yapay Veri 3	10000	2
Yapay Veri 4	20000	2

Deneylerde kullanılan gerçek veri kümelerinin karakteristik özellikleri Ek A' da verilmektedir. Yapay veri kümeleri ise, Veri kümesi (n1, n2) şeklinde düşünüldüğünde $n1 \in (0,450)$ ve $n2 \in (0,250)$ aralığında rastgele değerler alan iki nitelikli kümelerdir.

6.3. Standart K-Means ve Yeni K-means Sonucu Oluşan Kümeler

Örnek veri kümeleri üzerinde standart k-means algoritması ve İ.F.A.R.T. ile başlangıç noktaları belirlenen k-means algoritması sonucu elde edilen kümeler değerlendirilmektedir. Standart k-means algoritması rastgele küme merkezleri ile başlatıldığından her işletimde birbirinden farklı kümelerin oluşmasına neden olmaktadır. Yani algoritma kararlı bir yapıda çalışmamaktadır. Bu nedenden dolayı önerilen yöntem ile karşılaştırılabilmesi için, 100 farklı rastgele seçilmiş başlangıç küme merkezi ile yeniden çalıştırılmaktadır. Önerilen yöntemde ise, küme merkezleri İ.F.A.R.T. algoritmasının bir kez işletilmesi ile belirlenmektedir. Sonrasında k-means algoritması, bu belirlenmiş olan kümelere üyelik derecesi en yüksek olan noktalar ile yani küme merkezleri ile başlatılmaktadır.

İki farklı kümeleme yöntemi ile elde edilen sonuçlar değerlendirilirken iki farklı ölçütten yararlanılmaktadır. Bu ölçütler:

- Oluşan kümelerin geçerlilik ve doğruluk oranı
- Kümeleme hızı

Oluşan kümelerin doğru ve geçerli kümeler olup olmadıklarının tespitinde ayrıt 3.4'

de anlatıldığı gibi kaynaklarda farklı yaklaşımlar önerilmiştir. Ancak tüm bu yaklaşımlar arasında en yaygın olarak kullanılanı, küme içindeki elemanların birbirlerine yüksek oranda benzerlik göstermeleri, farklı kümelerdeki elemanların ise mümkün olduğunca az benzerlik göstermelerinin ölçüt olarak kullanılmasıdır. Bu iki durum sağlanabildiği ölçüde yapılan kümeleme doğru ve başarılı bir kümelemedir denir. Bu bölümde, kümeleme hata payı olarak adlandırılan ölçümün nasıl hesaplandığı ayrıt 3.4' de ayrıntılı olarak anlatılmaktadır. Kısaca tez kapsamında değerlendirilen kümeleme hata payı, küme içindeki elemanların ortalama uzaklıkları ile kümeler arası ortalama uzaklığın oranı olarak tanımlanmaktadır.

Kümeleme hızı ise şu şekilde hesaplanmaktadır. K-means algoritmasının ilk adımında öncelikle rastgele K adet küme merkezi belirlenmektedir. Belirlenen bu elemanlar tek elemanlı başlangıç kümeleridir ve ilk küme merkezlerini oluşturmaktadırlar. Kümenin ağırlıklı ortalama değerine sahip olan ya da bu değere en yakın olan eleman küme merkezi olarak adlandırılmaktadırlar.

İkinci adımda, okunan giriş elemanları kendilerine en yakın K adet küme merkezinden birine dahil edilmektedir. Noktalar arasındaki uzaklık Öklid bağıntısı ile hesaplanmaktadır.

Üçüncü adımda, her bir kümeye eklenen yeni eleman ile küme elemanlarının ağırlıklı ortalaması tekrar hesaplanarak yeni bir küme merkezi bulunmaktadır. Ağırlıklı ortalama kümenin her bir boyutundaki bütün elemanların ortalama değerlerinin alınması şeklinde hesaplanmaktadır. 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 dahil edilebilmektedir.

Kümeleme işlemi, tüm elemanların tekrar aynı veya farklı bir kümeye ayrılması ile devam etmektedir. Elemanların bir kümeye ayrılması ve küme merkezlerinin tekrar hesaplanması işlemlerine ait döngü, küme sınırlarının değişimi bitene kadar devam

etmektedir. K-means algoritması ile uygulamalarda genellikle birkaç düzine döngü sonrası kararlı bir küme grubu ortaya çıkmaktadır.

Tez kapsamında, k-means algoritmasına ait kümeleme hızı, başlangıç küme merkezlerinin güncelleme sayısı ile temsil edilmektedir. Algoritma başlamadan önce adım sayısı değeri sıfırlanır. Algoritmada gerçekleştirilen her güncelleştirme işlemi için algoritmanın adım sayısı bir artırılır. Adım sayısı ne kadar küçük değerde ise algoritma o kadar hızlı çalışmaktadır denir.

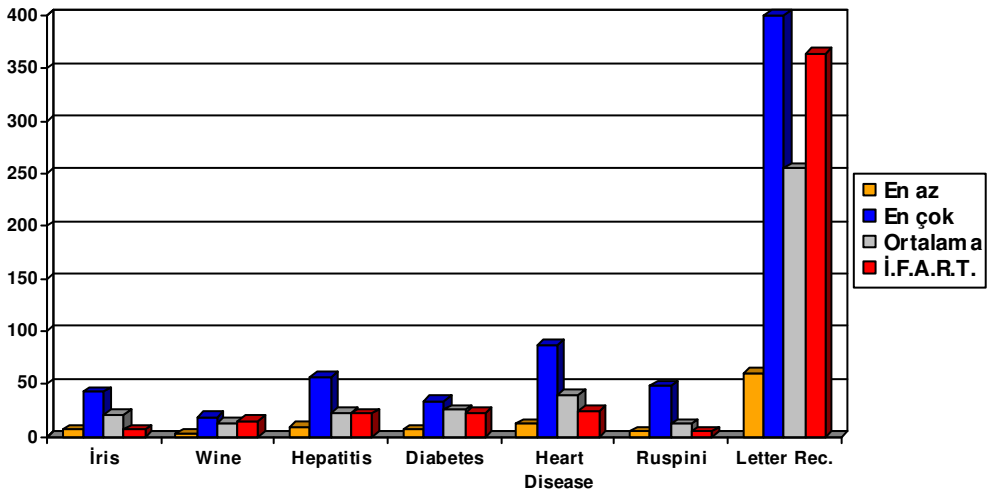
6.4. Deney Sonuçları

Tez kapsamında yapılan deneylerde, standart k-means algoritması Tablo 6.1 ve Tablo 6.2'de gösterilen tüm veri kümeleri için, 100 farklı rastgele başlangıç noktası ile çalıştırılmıştır. Rastgele örnekler ile başlatılan ve önerilen yöntem ile başlatılan k-means algoritmasına ait iki farklı işletim şekli aşağıdaki tablolarda adım sayısı ve hata oranı açısından karşılaştırılmaktadır.

Algoritmalar gerçek veri kümeleri ile çalıştırıldıklarında elde edilen adım sayıları ve hata oranları sırasıyla Tablo 6.3 ve Tablo 6.4 ile Şekil 6.1 ve Şekil 6.2'deki grafiklerde gösterilmektedir. Standart k-means algoritması 100 farklı başlangıç noktası ile 100 kez işletildiği için, algoritmaya ait adım sayıları ve hata oranları bu işletimlerde oluşan en az, en çok ve ortalama adım sayıları ve hata oranları şeklinde ifade edilmektedir. Tablolardaki gösterimler de buna göre yapılmıştır.

Tablo 6.3: Gerçek veri kümelerinin adım sayıları

Veri Kümesi	Standart K-Means En Az Adım	Standart K-Means En Çok Adım	Standart K-Means Ort. Adım (≈)	İ.F.A.R.T K-Means Adım
Iris	6	42	20	9
Wine	2	18	13	14
Hepatitis	9	57	22	21
Pima Indian Diabetes	6	34	25	22
Haberman's Survival	2	44	15	14
Heart-Disease-Cleveland	12	87	39	24
Ruspini	4	48	12	4
Letter Recognation	60	564	255	364



Şekil 6.1: Gerçek veri kümelerinde adım sayılarına ait grafik

İris veri kümesinde, standart k-means [6, 42] aralığında değişen adımlarda; ortalama olarak ise 20 adımda tamamlanmaktadır. İ.F.A.R.T. yöntemi ile başlatılan k-means ise 9 adımda tamamlanmaktadır. Kümeleme, seçilen rastgele başlangıç noktalarına göre 6 adım da sürebilirken 42 adım da sürebilmektedir. Ancak yeni k-means'de, başlangıç küme merkezleri bir yönteme dayandırılarak belirlendiğinden algoritma daha kararlı bir yapıda çalışmakta ve 9 adımda tamamlanmaktadır.

Wine veri kümesinde, standart k-means [2,18] aralığında değişen adımlarda; ortalama olarak ise 13 adımda tamamlanmaktadır. Yeni k-means ise 14 adımda tamamlanmaktadır yani standart k-means'in ortalama süresinden daha uzun

sürmektedir. Ancak göz önünde bulundurulması gereken bir başka durum, standart k-means'in rastgele seçilmiş olan başlangıç noktasına göre 18 adımda da tamamlanabileceğidir.

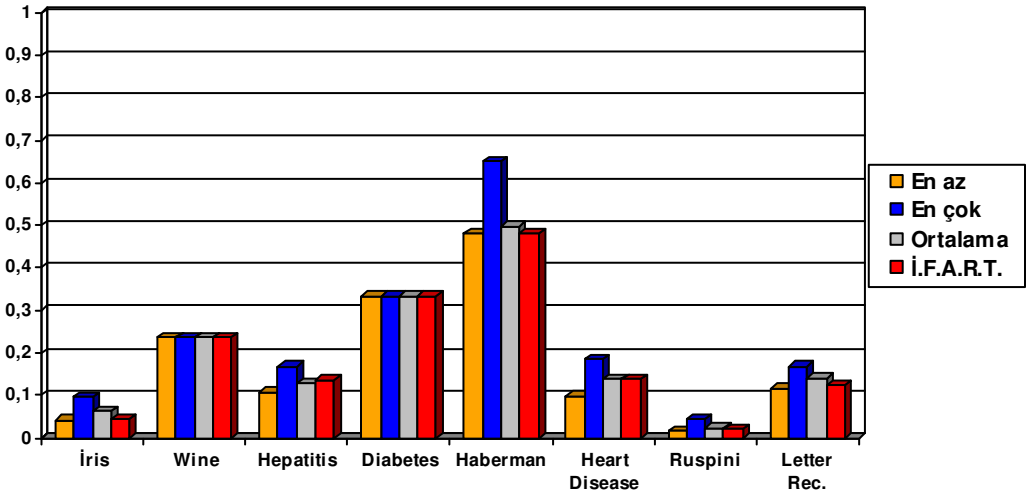
Hepatitis, Pima Indian Diabetes, Haberman's Survival ve Heart-Desease-Cleveland ve Ruspini veri kümelerinde de İris veri kümesinde olduğu gibi yeni k-means, standart k-means'in ortalama adım sayılarından daha düşük adımlarda işletimini tamamlamıştır.

Letter Recognition veri kümesinde ise standart k-means'in ortalama adım sayısı 255 iken yeni k-means'in ortalama adım sayısı 364 dür. Ancak tablodan da görüldüğü gibi, standart k-means algoritması 564 adımda da tamamlanmaktadır.

Tablo 6.4' de ve Şekil 6.2'deki grafikte algoritmalara ait kümeleme hata oranları gösterilmektedir.

Tablo 6.4: Gerçek veri kümelerinin hata oranları

Veri Kümesi	Standart K-Means En Az Hata	Standart K-Means En Çok Hata	Standart K-means Ort. Hata	İ.F.A.R.T. K-means Hata
Iris	0.0410	0.0964	0.0613	0.0451
Wine	0.2358	0.2365	0.2363	0.2365
Hepatitis	0.1077	0.1677	0.1290	0.1352
Pima Indians Diabetes	0.3321	0.3321	0.3321	0.3321
Haberman's Survival	0.4809	0.6502	0.4968	0.4809
Heart-Desease-Cleveland	0.0975	0.1850	0.1387	0.1381
Ruspini	0.0151	0.0450	0.0224	0.0192
Letter Recognation	0.1172	0.1685	0.1394	0.1227



Şekil 6.2: Gerçek veri kümelerinde hata oranlarına ait grafik

Gerçek veri kümeleri üzerinde, standart k-means ve yeni k-means işletildiğinde hesaplanan ve Tablo 6.4' de gösterilmiş olan hata oranları değerlendirildiğinde, yeni k-means ile standart k-means'in ürettiği ortalama hata değerine yakın değerlerin elde edildiği gözlenmektedir.

İris veri kümesinde, standart k-means algoritması işletildiğinde [0.0410, 0.0964] aralığında değişen hata oranları elde edilmiştir. 100 farklı işletim sonucunda elde edilen hata paylarının ortalama değeri ise 0.0613 olarak hesaplanmıştır. Buna karşılık yeni k-means algoritması 0.0451 hata payı ile kümelemeyi gerçekleştirmiştir.

Wine ve Hepatitis veri kümelerinde standart k-means'in ortalama hatasına yakın ama biraz daha büyük bir hata payı ile kümeleme yapılmıştır.

Pima Indians Diabetes veri kümesinde standart k-means ile aynı hata payı ile kümeleme yapılmıştır.

Haberman's Survival veri kümesinde yeni k-means algoritması, standart k-means'in üretebildiği en düşük hata payı ile kümelemeyi gerçekleştirmiştir.

Heart-Disease-Cleveland veri kümesinde ise standart k-means'in ortalama hata payından biraz daha düşük ancak üretebileceği en yüksek hata payından çok daha düşük bir hata payı ile kümeleme gerçekleştirilmiştir.

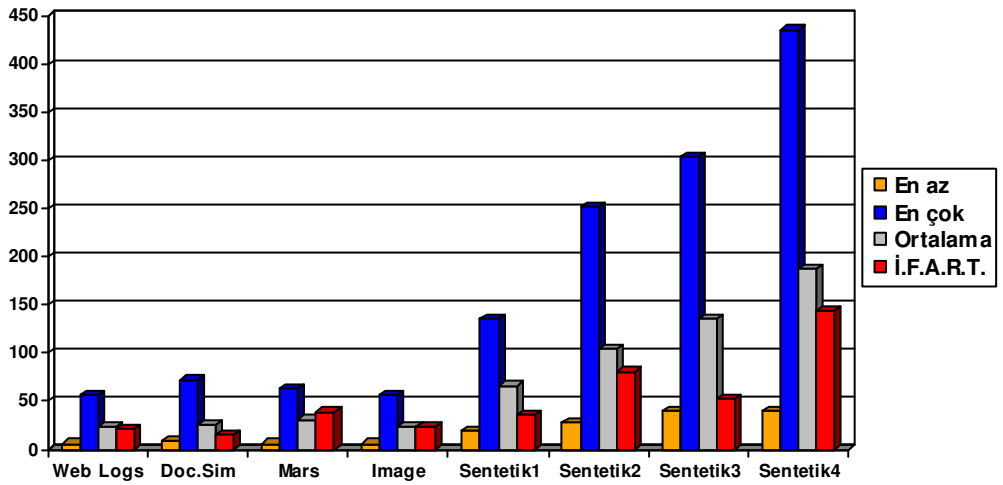
Ruspini ve Letter Recognition veri kümelerinde, standart k-means'in ortalama hata payından daha düşük hata payı ile kümeleme yapılmıştır.

Buraya kadar sonuçları yorumlanan veri kümeleri, genellikle sınıflandırma algoritmalarının değerlendirilmesinde kullanılan veri kümeleri olduklarından ve sınıflandırma sonucu oluşması gereken küme sayıları belirlenmiş olduğundan, kümeleme algoritmaları için üretilmiş olan yapay veri kümeleri üzerinde de iki yöntem karşılaştırılmıştır.

Yapay veri kümeleri üzerinde algoritmaların adım sayıları ve hata oranları sırasıyla Tablo 6.5, Tablo 6.6 ile Şekil 6.3, Şekil 6.4'deki grafikler ile gösterilmektedir.

Tablo 6.5: Yapay veri kümelerinin adım sayıları

Veri Kümesi	Standart K-Means En Az adım	Standart K-Means En Çok adım	Standart K-Means Ort. adım (≈)	İ.F.A.R.T. K-Means Adım
Web Logs	6	57	24	21
Documents Similarity	9	72	26	15
Mars	6	63	31	39
Image Extraction	6	57	24	24
Yapay Veri 1	20	136	66	36
Yapay Veri 2	28	252	104	81
Yapay Veri 3	40	304	136	52
Yapay Veri 4	40	436	188	144



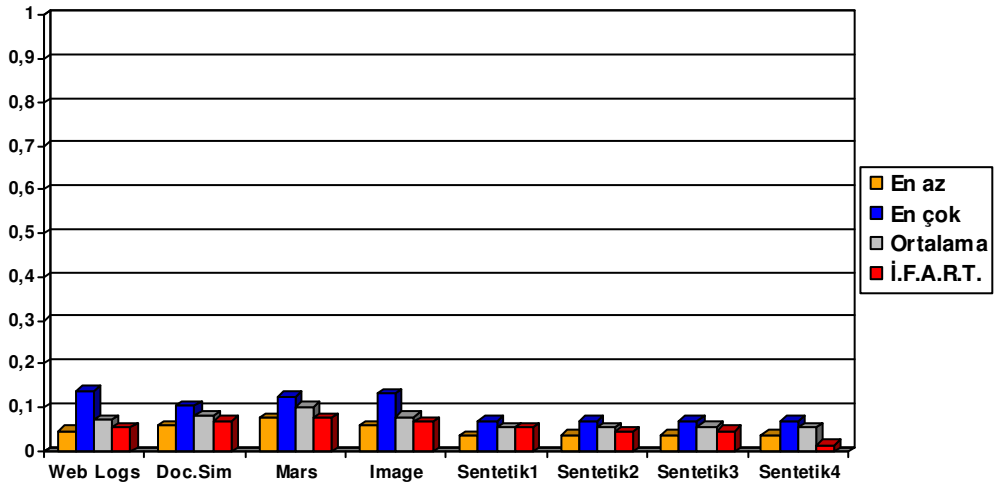
Şekil 6.3: Yapay veri kümelerinde adım sayılarına ait grafik

Yukarıdaki tablo değerlendirildiğinde, Web Logs, Documents Similarity, Image Extraction veri kümelerinde standart k-means'in ortalama adım sayısından daha az adımda kümeleme tamamlandığı görülmektedir. Mars veri kümesinde ise, ortalama adımdan (31) daha çok adımda (39) tamamlanmaktadır. Ancak oluşabilecek en yüksek adım sayısının 63 olduğu göz önüne alındığında daha az adımda da tamamlanabileceği de görülmektedir.

Yapay veri 1, 2, 3 ve 4 diye adlandırılan veri kümelerine bakıldığında, oluşan adım sayılarının standart yöntemin oluşturduğu ortalama adım sayılarından çok daha düşük oldukları gözlenmektedir.

Tablo 6.6: Yapay veri kümelerinin hata oranları

Veri Kümesi	Standart K-Means En Az Hata	Standart K-Means En Çok Hata	Standart K-Means Ort. Hata	İ.F.A.R.T. K-Means Hata
Web Logs	0.0471	0.1400	0.0713	0.0538
Documents Similarity	0.0593	0.1053	0.0832	0.0702
Mars	0.0778	0.1259	0.1020	0.0786
Image Extraction	0.0578	0.1318	0.0798	0.0672
Yapay Veri 1	0.0368	0.0700	0.0537	0.0535
Yapay Veri 2	0.0379	0.0712	0.0554	0.0449
Yapay Veri 3	0.0379	0.0707	0.0559	0.0467
Yapay Veri 4	0.0379	0.0704	0.0555	0.0149



Şekil 6.4: Yapay veri kümelerinde hata oranlarına ait grafik

Tablo 6.6 incelendiğinde kümeleme işleminin, tüm yapay veri kümelerinde yeni k-means ile standart k-means algoritması ile olduğundan daha az hata payı ile gerçekleştirildiği gözlenmektedir.

Deneylerde kullanılan veri kümeleri için, İ.F.A.R.T. ile başlatılan k-means algoritmasının süresi ve standart k-means algoritmasının 100 iterasyon işletimi için gerekli olan süre, Intel 1.66 GHz işlemci, 1 GB Ram özellikli Windows XP işletim sistemi olan bir bilgisayar üzerinde ölçülmüştür. Sonuçlar saniye olarak Tablo 6.7'de görülmektedir.

Tablo 6.7: Toplam çalışma süreleri

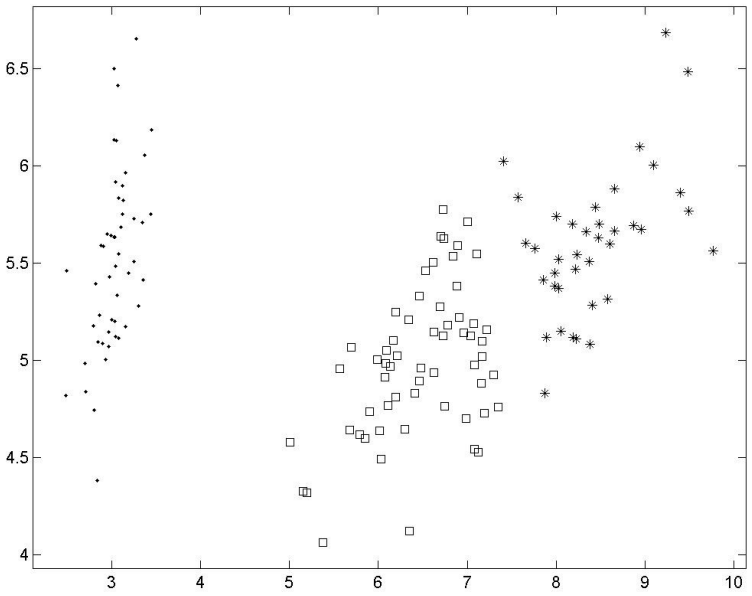
Veri kümesi / Alg. Süresi (saniye)	IFART ve K-means	K-means 100 iterasyon
Iris	0.282	15.453
Wine	0.125	6.328
Hepatitis	0.157	15.954
Pima Indians Diabetes	0.625	48.5
Haberman's Survival	0.141	15.407
Heart-Disease-Cleveland	0.531	34.047
Ruspini	0.078	3.765
Letter Recognition	292.719	14097
Web Logs	0.39	15.14
Document Similarity	0.203	21.11
Mars	0.25	14.297
Image Extraction	0.219	10.235

6.5. Deney Sonuçlarının İki Boyutlu Uzayda Gösterimi

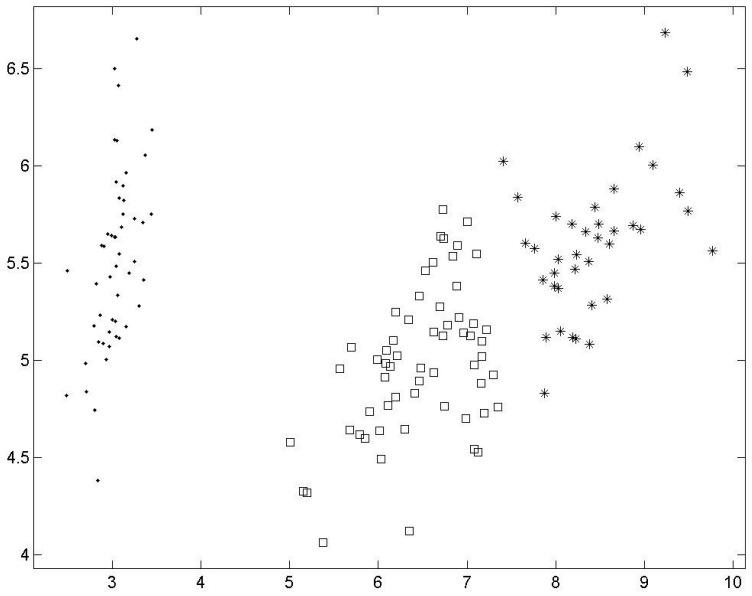
Rastgele başlangıç noktaları ile başlatılan standart k-means algoritması ve İ. F.A.R.T. ile başlatılan k-means algoritmasının işletiminden sonra elde edilen kümeler 2 boyutlu uzayda gösterilmektedir. Şekiller, "n" boyutlu uzayın temel bileşenler analizi [103] yardımı ile iki boyuta indirgenmesi ile iki boyutlu olarak gösterilmektedirler.

6.5.1. Gerçek veri kümelerinden elde edilen sonuçlar

1-) İris verisine ait kümeler: Şekil 6.3 ve 6.4'de iki farklı k-means algoritmasının İris veri kümesine uygulandığında elde edilen sonuçlar gözlenmektedir.

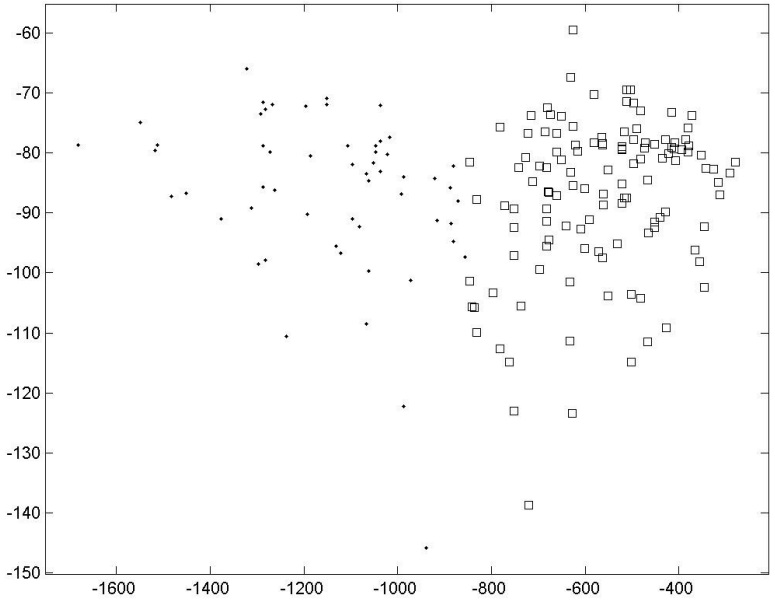


Şekil 6.5: İris için İ.F.A.R.T. ile başlatılan k-means kümeleri

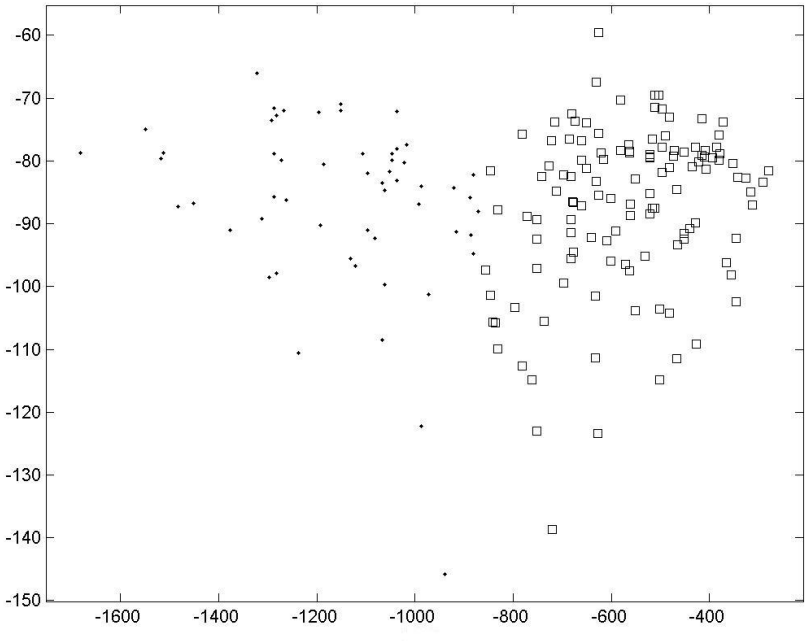


Şekil 6.6: İris için standart k-means kümeleri

2-) Wine verisine ait kümeler: Şekil 6.5 ve 6.6'da iki farklı k-means algoritmasının Wine veri kümesine uygulandığında elde edilen sonuçlar gözlenmektedir.

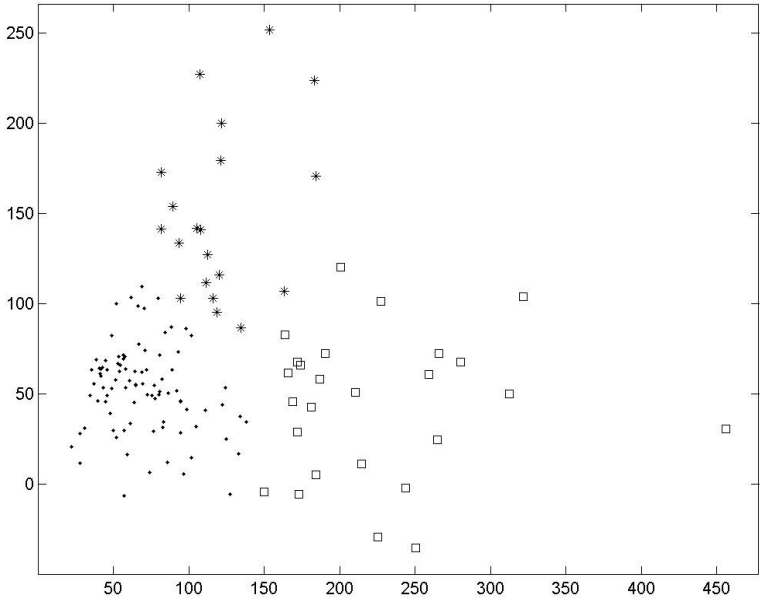


Şekil 6.7: Wine için İ.F.A.R.T. ile başlatılan k-means kümeleri

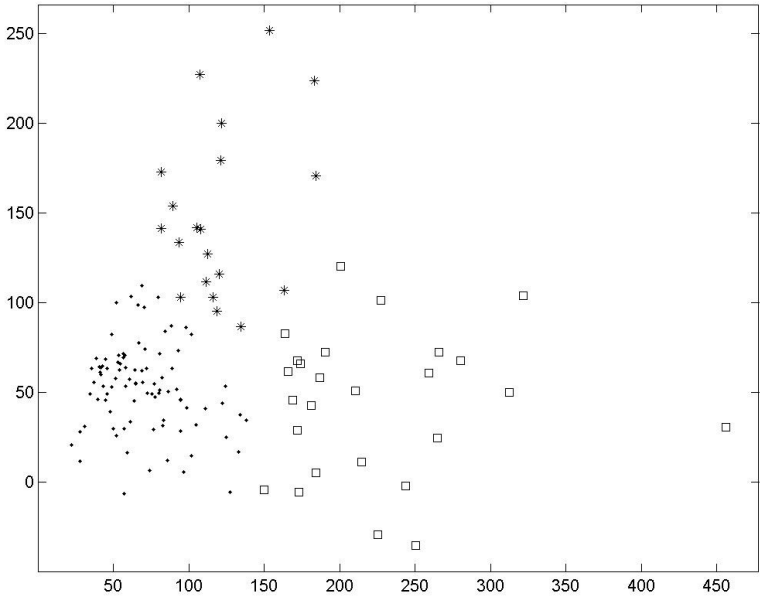


Şekil 6.8: Wine için standart k-means kümeleri

3-) Hepatitis verisine ait kümeler: Şekil 6.7 ve 6.8'de iki farklı k-means algoritmasının Hepatitis veri kümesine uygulandığında elde edilen sonuçlar gözlenmektedir.

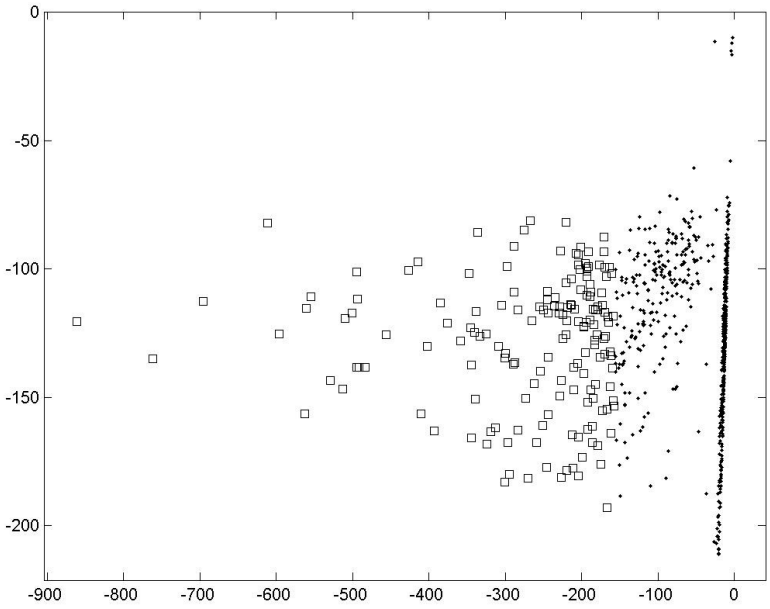


Şekil 6.9: Hepatitis için İ.F.A.R.T. ile başlatılan k-means kümeleri

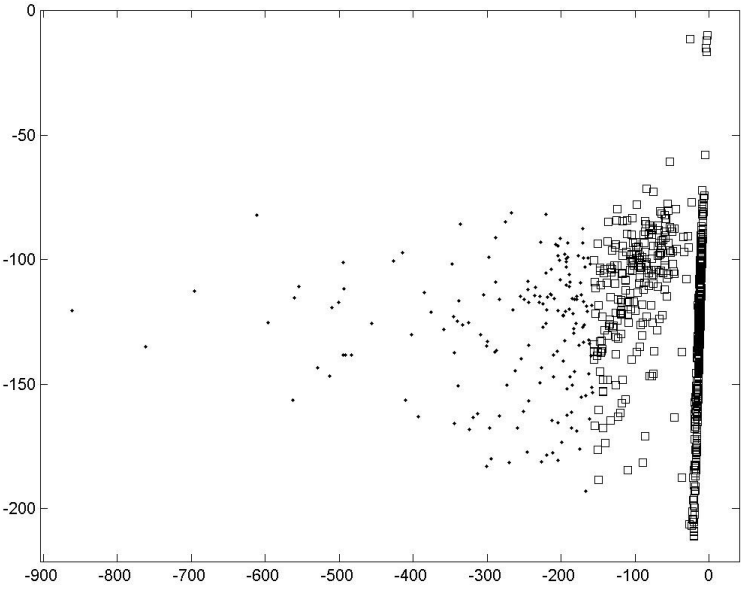


Şekil 6.10: Hepatitis için standart k-means kümeleri

4-) Pima Indians Diabetes verisine ait kümeler: Şekil 6.9 ve 6.10'da iki farklı k-means algoritmasının Pima Indians Diabetes veri kümesine uygulandığında elde edilen sonuçlar gözlenmektedir.

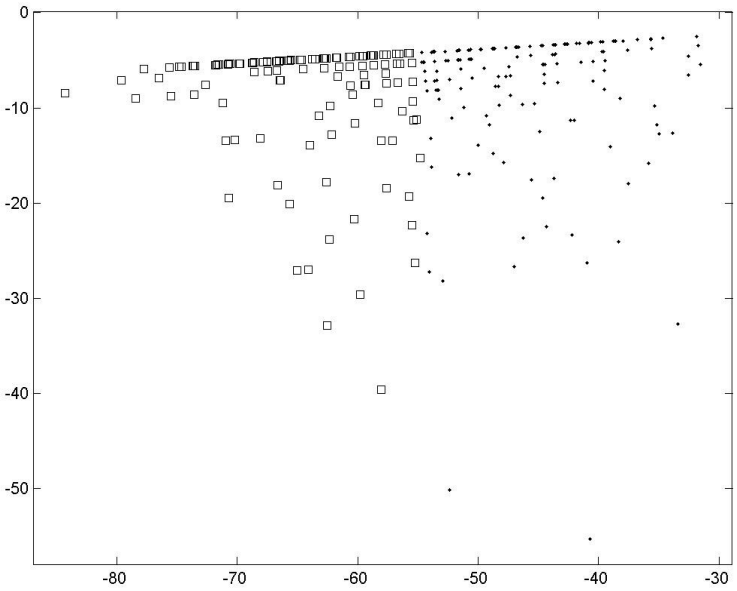


Şekil 6.11: Pima Indians Diabetes için İ.F.A.R.T. ile başlatılan k-means kümeleri

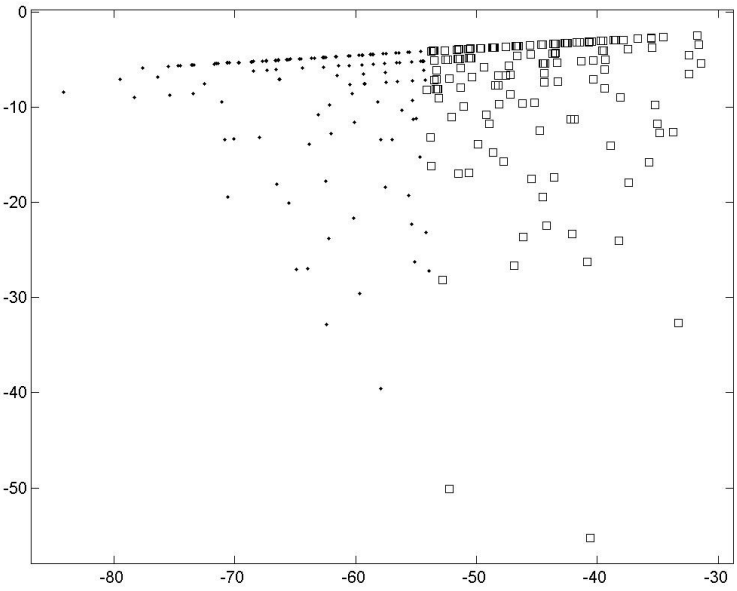


Şekil 6.12: Pima Indians Diabetes için standart k-means kümeleri

5-) Haberman's Survival verisine ait kümeler: Şekil 6.11 ve 6.12'de iki farklı k-means algoritmasının Haberman's Survival veri kümesine uygulandığında elde edilen sonuçlar gözlenmektedir.

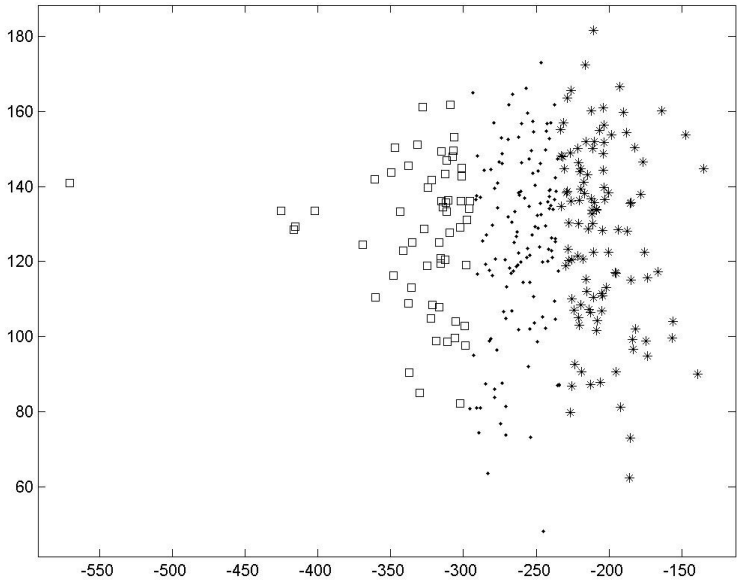


Şekil 6.13: Haberman's Survival için İ.F.A.R.T. ile başlatılan k-means kümeleri

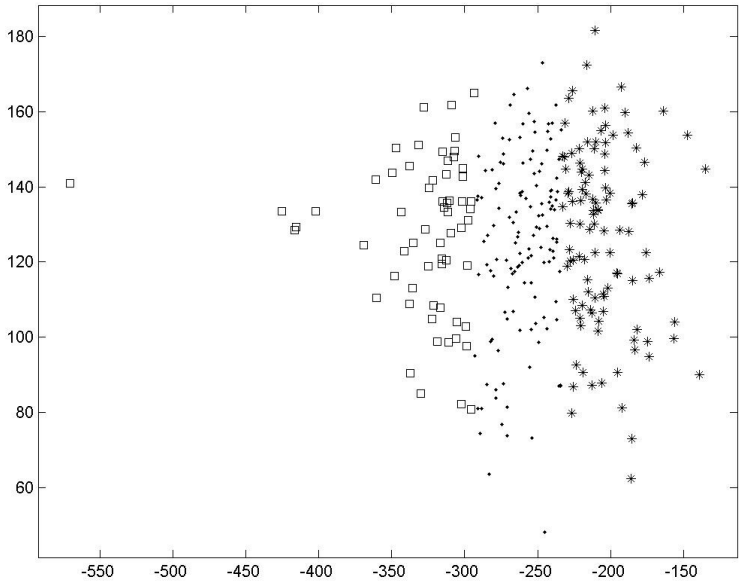


Şekil 6.14: Haberman's Survival için standart k-means kümeleri

6-) Heart-Disease-Cleveland verisine ait kümeler: Şekil 6.13 ve 6.14'de iki farklı k-means algoritmasının Heart-Disease-Cleveland veri kümesine uygulandığında elde edilen sonuçlar gözlenmektedir.

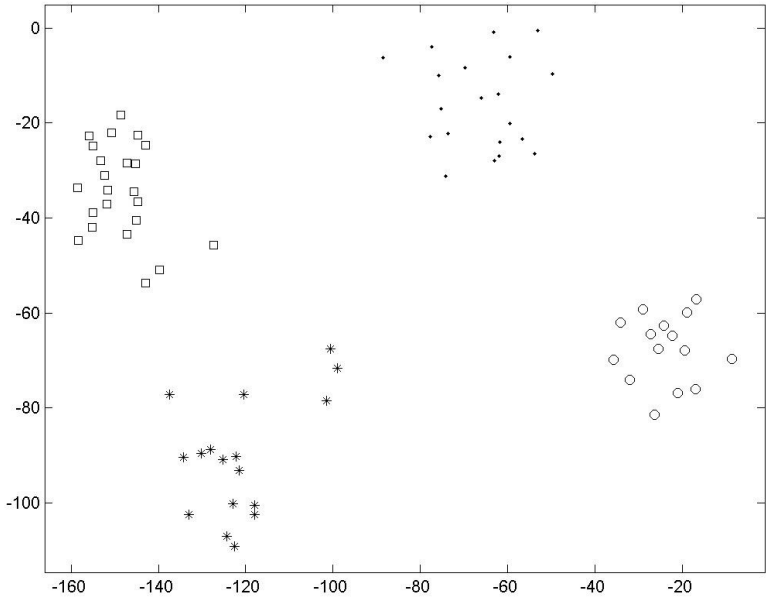


Şekil 6.15: Heart-Disease-Cleveland için İ.F.A.R.T. ile başlatılan k-means kümeleri

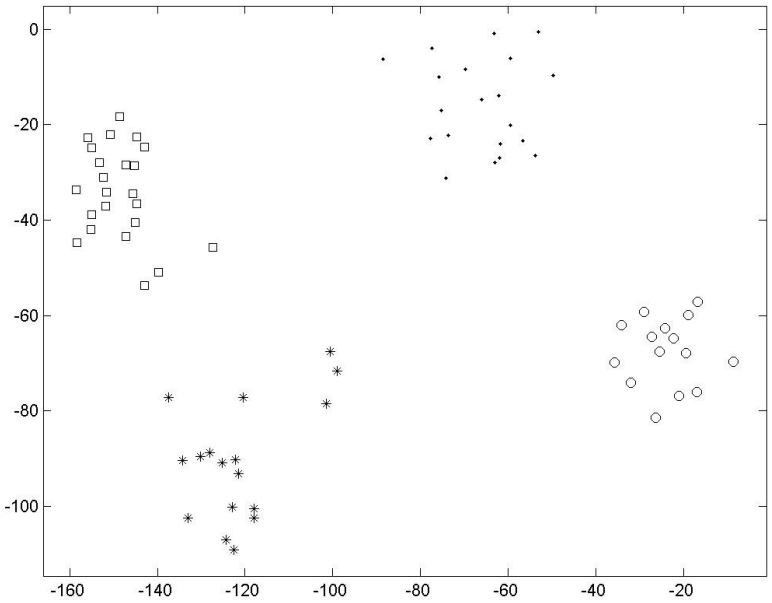


Şekil 6.16: Heart-Disease-Cleveland için standart k-means kümeleri

7-) Ruspini verisine ait kümeler: İki farklı k-means algoritmasının Ruspini veri kümesine uygulandığında elde edilen sonuçlar Şekil 6.15 ve 6.16'da gözlenmektedir.



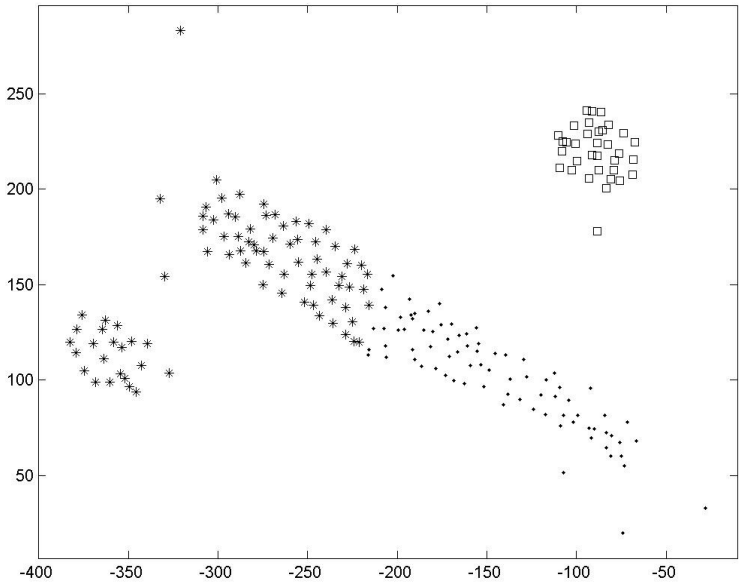
Şekil 6.17: Ruspini için İ.F.A.R.T. ile başlatılan k-means kümeleri



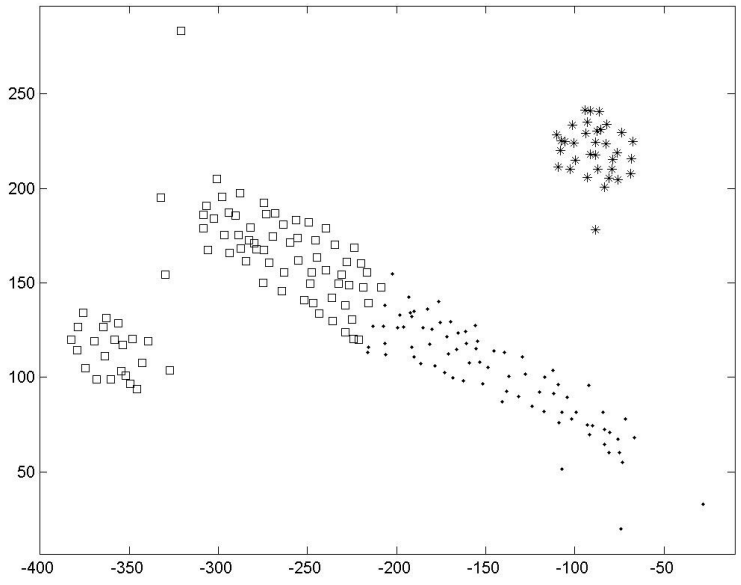
Şekil 6.18: Ruspini için standart k-means kümeleri

6.5.2. Yapay veri kümelerinden elde edilen sonuçlar

1-) Web logs verisine ait kümeler: İki farklı k-means algoritması web logs veri kümesine uygulandığında elde edilen sonuçlar Şekil 6.17 ve 6.18'de gözlenmektedir.

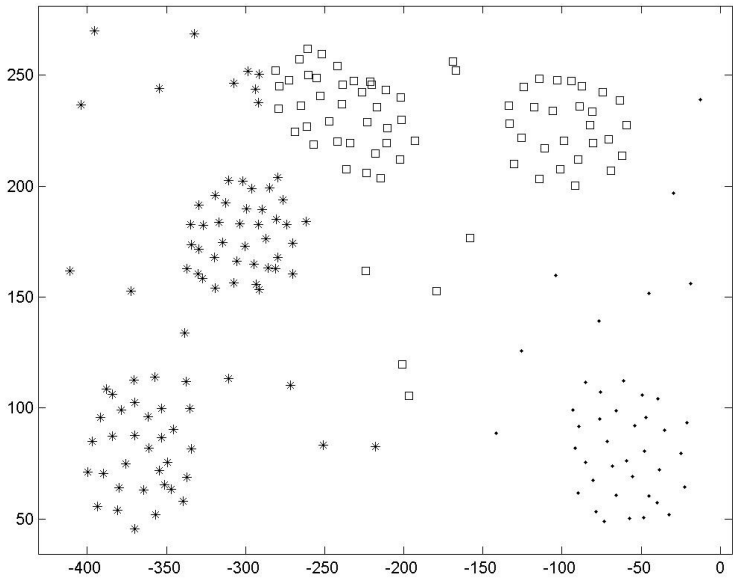


Şekil 6.19: Web logs için İ.F.A.R.T. ile başlatılan k-means kümeleri

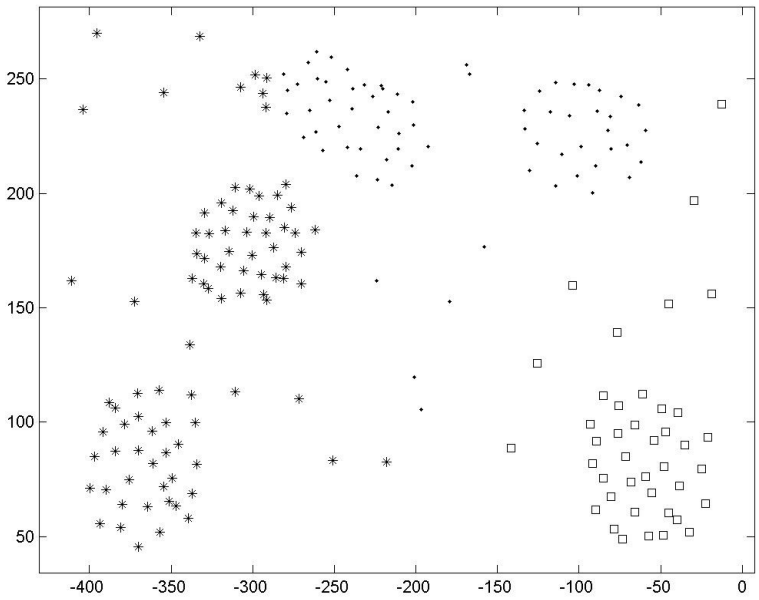


Şekil 6.20: Web logs için standart k-means kümeleri

2-) Document Similarity verisine ait kümeler: İki farklı k-means algoritması document similarity veri kümesine uygulandığında elde edilen sonuçlar Şekil 6.19 ve 6.20'de gözlenmektedir.

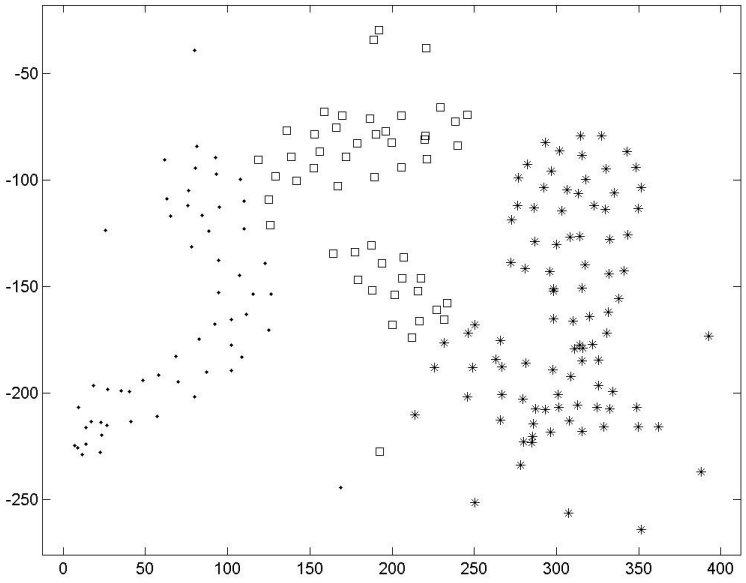


Şekil 6.21: Document similarity için İ.F.A.R.T. ile başlatılan k-means kümeleri

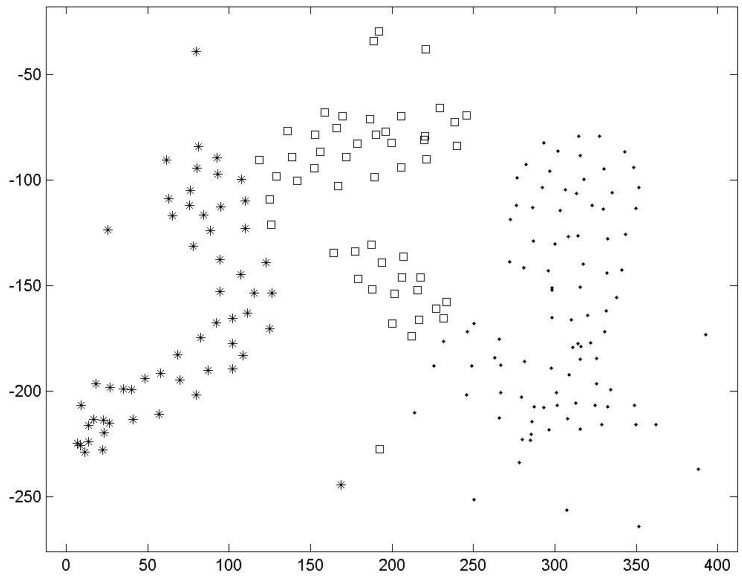


Şekil 6.22: Document similarity için standart k-means kümeleri

3-) Mars verisine ait kümeler: İki farklı k-means algoritması mars veri kümesine uygulandığında elde edilen sonuçlar Şekil 6.21 ve 6.22'de gözlenmektedir.

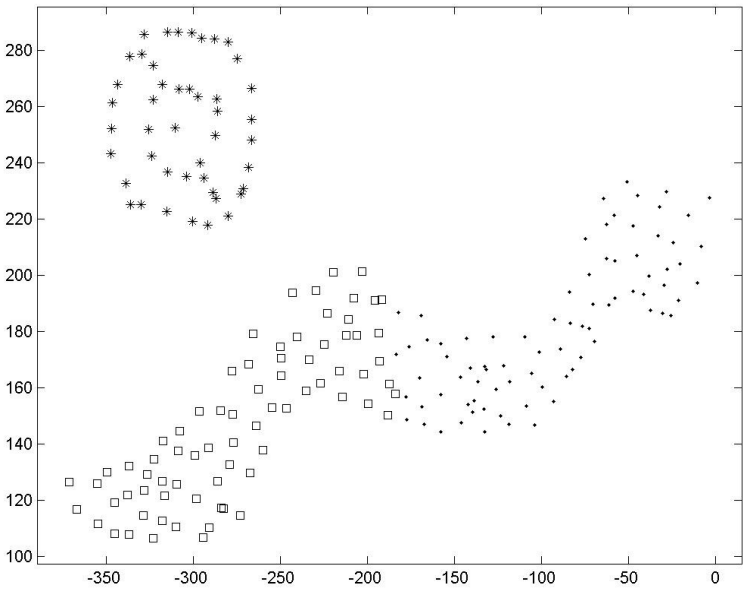


Şekil 6.23: Mars için İ.F.A.R.T. ile başlatılan k-means kümeleri

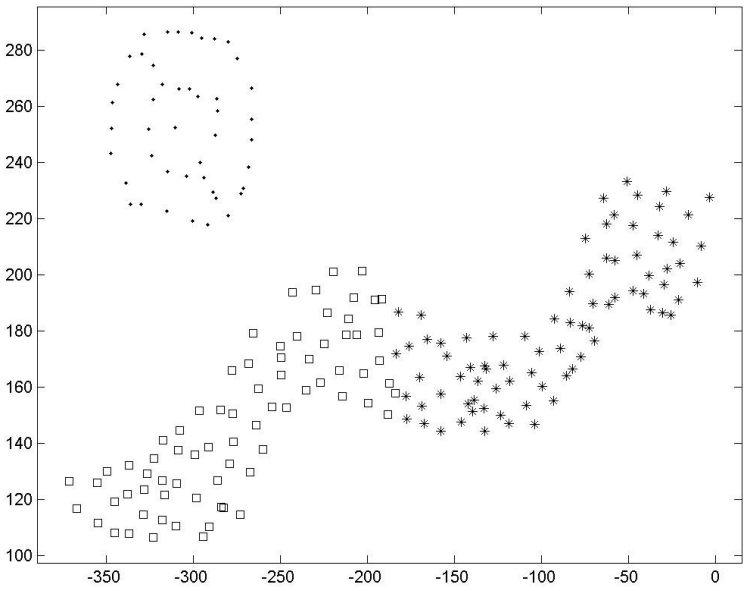


Şekil 6.24: Mars için standart k-means kümeleri

4-) Image Extraction verisine ait kümeler: İki farklı k-means algoritması image extraction veri kümesine uygulandığında elde edilen sonuçlar Şekil 6.23 ve 6.24'de gözlenmektedir.



Şekil 6.25: Image extraction için İ.F.A.R.T. ile başlatılan k-means kümeleri



Şekil 6.26: Image extraction için standart k-means kümeleri

Bu şekilde, İ.F.A.R.T. algoritması ile başlangıç noktaları belirlenen k-means algoritmasından elde edilen kümeler ve standart k-means algoritması ile elde edilen kümeler örnekler ile gösterilmekte ve analiz edilmektedir. Yöntemin belirgin üstünlükleri ve rastgele başlangıç noktaları ile başlatılan standart k-means algoritması ile karşılaştırılması sonuçlar ve tartışmalar bölümünde verilecektir.

7. SONUÇLAR ve ÖNERİLER

Bu bölümde tez çalışmasının genel bir değerlendirilmesi yapılarak elde edilen önemli sonuçlar verilecektir. Ayrıt 5.2' de verilen iyileştirilmiş bulanık A.R.T. algoritması ile belirlenmiş olan küme merkezleri ile başlatılan k-means kümeleme algoritmasının, rastgele başlangıç noktaları ile başlatılan k-means algoritması ile karşılaştırılarak üstünlükleri değerlendirilecek ve sonraki çalışmalar için bazı öneriler tartışılacaktır.

Tezde ele alınan temel problem, k-means algoritmasının temel sorunu olan başlangıç noktalarına duyarlılığın tez kapsamında önerilmiş olan iyileştirilmiş bulanık A.R.T. algoritması ile azaltılmasıdır. Bu amaçla, üçüncü bölümden başlayarak sistematik bir yaklaşımla mevcut yöntemler kuramsal olarak incelenmiş ve önerilen yöntem için altyapı oluşturulmaya çalışılmıştır.

İlk yaklaşım olarak kümeleme işlemi bulanık A.R.T. algoritması ile bilgisayar ortamında farklı veri kümeleri üzerinde gerçekleştirilmiştir. Bu çalışmadan şu şekilde bir sonuç elde edilmiştir.

Bulanık A.R.T. algoritması sonucunda oluşan kümelerin, iyi ayrılmış kümeler olmadıkları gözlenmiştir. Birbirine benzer kayıtların aynı kümede, birbirinden farklı kayıtların ayrı kümelerde olmaları beklenirken; yüksek oranda benzerlik gösteren kayıtların bile ayrı kümelerde yer aldıkları gözlenmiştir. Farklı veri kümelerinden elde edilen bu sonuçlar ayrıt 5.3' de verilmektedir.

Bu problemden yola çıkarak, hangi kayıtların yanlış kümelerde yer aldıkları her giriş verisinin her kümeye üyelik derecesi hesaplanarak belirlenmiştir. Giriş verilerinin kümelere olan üyeliklerinde kümeyi temsil eden eleman olarak küme merkezi seçilmiştir. Kümelemenin daha geçerli ve doğru gerçekleştirilmesi adına, elemanlar

en yüksek üyelik derecesi ile bağlı oldukları kümelere dahil edilmişlerdir. Yer değiştirme işlemi sonucunda elde edilen yeni kümeler bulanık A.R.T. algoritması ve kümelemede sık kullanılan bir başka yapay sinir ağı olan S.O.M. algoritması sonucunda elde edilen kümeler ile karşılaştırılmışlardır. Ayrıt 5.3' de bu üç algoritma sonucu oluşan kümeler karşılaştırmalı olarak gösterilmektedir.

Önerilen yöntem, bulanık A.R.T. algoritması sonucu oluşturulan kümelerin daha geçerli kümeler haline gelmesini sağlamıştır. Bu durum, kümelemeden elde edilen hata payındaki azalma ile de doğrulanmıştır. Örnek veri kümelerinde iyileştirilmiş bulanık A.R.T: algoritması standart bulanık A.R.T. algoritmasından çok daha düşük hata payları ile kümelemeyi gerçekleştirmektedir. Bunun yanında örnek verilerin, SOM algoritması ile gerçekleştirilen kümelemeye yakın hata payları ile kümelendiği de gözlenmektedir. Başka bir ölçüt olan kümeleme hızı açısından ise, kümeleme deneylerinin SOM algoritmasından çok daha kısa sürelerde tamamlandığı gözlenmiştir.

Kaynaklarda birçok kümeleme algoritması, başka kümeleme algoritmaları için başlangıç konumlarını belirleyen bir algoritma şeklinde uygulanmıştır. Bu başlangıç algoritmaları, k-means ile yapılan kümelemenin kalitesini ve performansını artırmaktadırlar. Yapılan çalışmada, bölüm 6'da iyileştirilmiş bulanık A.R.T. sonucu elde edilen kümeler k-means algoritmasının başlangıç küme merkezlerinin belirlenmesinde kullanılmıştır.

K-means algoritması ve algoritmanın zorlukları bölüm 3'de incelenmiştir. K-means algoritması başlangıç küme merkezlerinin seçimine göre çok farklı kümelemeler gerçekleştirebilen bir algoritmadır. Doğru başlangıç noktalarının seçilmesi algoritmanın etkinliği açısından anahtar adımdır. En yaygın olarak kullanılan yöntem, başlangıç küme merkezlerinin rastgele seçilmesidir. Ayrıt 3.4.4'de kaynaklarda uygulanan rastgele başlangıç yöntemlerinden ve rastgele belirlenen küme merkezleri ile başlatılan algoritma sonucu oluşan kümelerin düşük kalitede olduklarından bahsedilmiştir. K-means algoritmasına ait bu problemin üstesinden gelmek üzere algoritma farklı başlangıç noktaları ile çok kez çalıştırılmaktadır, ve en geçerli kümeleme sonucu bu şekilde saptanmaktadır. Bu nedenle algoritma kararlı bir

algoritma değildir ve en uygun sonucun elde edilebilmesi için defalarca çalıştırılması gerekmektedir.

Yöntem diğer yöntemler ile karşıl aştırıldığında göze çarpan en önemli üstünlükleri aşağıdaki başlıklar ile sıralanmaktadır.

1- Önerilen yöntem, basit bir mantıkla ve karmaşık matematiksel temellere gerek olmaksızın geliştirilmiştir.

2- Standart k-means algoritmasından daha etkin bir kümeleme gerçekleştirmektedir.

3- Standart k-means algoritmasına göre daha az hata payı ile kümelemeyi gerçekleştirmektedir.

4- Standart k-means algoritması kararlı yapıda çalışmamakta, farklı başlangıç noktaları ile oluşan kümeler arasından en iyi kümeleme durumu seçilmeye çalışılmaktadır. Önerilen yöntem ise, başlangıç noktalarını iyileştirilmiş bulanık A.R.T. yöntemi ile belirlediğinden daha kararlı sonuç üretmektedir.

5- İyileştirilmiş bulanık A.R.T. algoritmasının tamamlanması için harcanan süre, k-means algoritmasının yeniden defalarca çalıştırılması için gereken süreden daha kabul edilebilir bir süredir.

6- Önerilen yöntem ile k-means başlangıç küme merkezleri hızlı çalışan, akıllı bir bulanık kümeleme algoritması ile belirlenmiş olmaktadır.

7- Bahsedilen bu sonuçlar 8 adet gerçek veri kümesi ve 8 adet yapay veri kümesi üzerinde uygulanarak sınanmıştır.

Önerilen yöntemin şu aşamadaki en önemli sakıncası, başlangıç küme merkezlerinin belirlenmesi için ek süre harcanmasıdır. Ancak bu sakınca, standart k-means algoritmasının defalarca çalıştırılarak elde edilen sonuçların saklandığı ve bunların arasından en iyi sonucun arandığı düşünüldüğünde ortadan kalkmaktadır.

Kümeleme veri madenciliği alanında en sık başvurulan yöntemlerdendir. Kümeleme konusunda ise oluşan kümelerin doğru ve geçerli olmaları, yapılan işin başarısından söz edebilmek için en önemli olan noktadır. Bu konuda çok farklı yöntem ve algoritma çözüm olarak önerilmiştir. Tez kapsamında, daha önce bu alanda kullanılmamış olan bulanık A.R.T. yöntemine dayanan yeni bir algoritma tarafımızdan önerilmiştir. Bu konu için bir çözüm yöntemi düşünülürken açık olan bir nokta vardır. Büyük veri kümelerini hızlı ve etkin şekilde kümeleyen algoritmalar başlangıç küme merkezlerinin belirlenmesi konusunda daha doğru ve uygun çözümler olacaklardır. Bu bağlamda büyük verilerde hızlı kümeleme gerçekleştiren bulanık A.R.T. algoritması daha etkin hale getirilerek çözüm olarak sunulmuş ve uygulanmıştır. Yukarıda da bahsedildiği gibi, yöntem, standart k-means algoritmasına göre daha geçerli kümeleme yapabilmektedir.

Tez kapsamında önerilmiş olan İ.F.A.R.T. algoritması, sonuçların başlangıç noktalarına bağlı olarak değiştiği tüm kümeleme algoritmalarında, başlangıç noktalarının belirlenmesinde kullanılabilmektedir. Örnek olarak bulanık c-means (Fuzzy c-means) algoritmasında kümeleme durumunu iyileştirebilir. K-medoids algoritmasında ise, başlangıç kümelerine ait noktalar, İ.F.A.R.T. sonucu elde edilen kümelerdeki elemanlardan, bağlı oldukları kümeye en yüksek üyelik derecesine sahip olandan azalana doğru istenen sayı kadarının seçilmesi yolu ile belirlenebilir. Önerilen yöntem, k-means algoritmasında olduğu gibi farklı kümeleme algoritmalarını da daha geçerli kümeleme yapabilir hale getirebilir.

KAYNAKLAR

[1] Han J., Kamber M., "Data Mining: Concepts and Techniques", ***Morgan Kaufmann Publishers,*** United States of America, (2006).

[2] Quinlan, Jr., "Induction of decision trees", ***Machine Learning,*** 1(1), 81-106, (1986)]

[3] Friedman, N., Geiger, D., Goldszmidt, M., "Bayesian Network Classifiers", ***Machine Learning,*** 29(2-3), 131-163, (1997).

[4] MacQueen, J., "Some methods for classification and analysis of multi-variate observations", ***In: Proc. of the Fifth Berkeley Symp. on Math., Statistics and Probability,*** LeCam, L.M., and Neyman, J., (eds.), Berkeley: U. California Press, 281 – 297 , (1967).

[5] Kohonen, T., "A Simple Paradigm for the Self-organized Formation of Structured Feature Maps", Competition and Cooperation in Neural Nets, (1982).

[6] Carpenter, G.A., Grossberg, S., Rosen, D.B., "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System", ***Neural Networks,*** 4, 759 – 771, (1991).

[7] Tzortzis, G., Likas, A., "The Global Kernel k-Means Clustering Algorithm", ***International Joint Conference on Neural Networks,*** 1977 - 1984, (2008).

[8] Yang, S-Z., Luo, S-W., "A novel algorithm for initializing clustering centers" ***Proceedings of the Fourth International Conference on Machine Learning and Cybernetics,*** Guangzhou, 5579 – 5583 , (2005).

[9] Li, M.J., Ng, M.K., Cheung, Y-M., "Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters", ***IEEE Transactions On Knowledge And Data Engineering,*** 20(11), 1519 - 1534, (2008).

[10] Hamerly, G., Elkan, C., "Learning the k in k-Means" ***Proc. 17th Ann. Conf. Neural Information Processing Systems,*** (2003)

[11] Huang, Z., Ng, M., Rong, H., Li, "Automated Variable Weighting in k-Means Type Clustering," ***IEEE Trans. Pattern Analysis and Machine Intelligence,*** 27(5), 657 - 668, (2005).

[12] Jain, A.K., Dubes, R.C., "Algorithms for Clustering Data", ***Prentice Hall,*** New Jersey, (1988).

[13] Babu, G.P., Murty, M.N., ***"A Near-optimal Initial Seed Value Selection for K-Means Algorithm Using Genetic Algorithm",*** Pattern Recognition Letters, 14, 763-769, (1993).

[14] Krishna, K., Murty, M.N., "Genetic K-Means Algorithm" ***IEEE Trans. Systems, Man, and Cybernetics,*** 29(3), (1999).

[15] Laszlo, M., Mukherjee, S. "A Genetic Algorithm Using Hyper- Quadtrees for Low-Dimensional K-Means Clustering", ***IEEE Trans. Pattern Analysis and Machine Intelligence,*** 28(4), 533 - 543, (2006).

[16] Laszlo, M., Mukherjee, S., "A Genetic Algorithm That Exchanges Neighboring Centers for K-Means Clustering", ***Pattern Recognition Letters,*** 28(16), 2359 - 2366, (2007).

[17] Arthur, D., Vassilvitskii, S., "K-Means++: The Advantages of Careful Seeding," ***Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms,*** 1027-1035, (2007).

[18] Duda, R.O., Hart, P.E., "Pattern classification and scene analysis", ***John Wiley and Sons,*** New York, (1973).

[19] Thiesson, B., Meck, C., Chickering, D., Heckerman, D., "Learning mixtures of Bayesian networks", ***Microsoft Research Technical Report,*** Redmond, WA., 97-30, (1997).

[20] Bradley, P. S., Fayyad, U. M., "Refining initial points for K-Means clustering", ***In Proceedings of the Fifteenth International Conference on Machine Learning,*** San Francisco, CA, Morgan Kaufmann, 91–99, (1998).

[21] Ting, S., Dy., J., "A Deterministic Method for Initializing K-means Algorithm", ***Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence,*** 784 – 786, (2004).

[22] Pen~a, J.M., Lozano, J.A., Larra~naga, P., "An empirical comparison of four initialization methods for the k-means algorithm", ***Pattern Recognition Lett.,*** 20, 1027 – 1040, (1999).

[23] Kohei, A., Barakbah, A.R., "Hierarchical K-means: an algorithm for centroids initialization for K-means", ***Reports of the Faculty of Science and Engineering,*** Sga University, 36(1), 25 – 31, (2007).

[24] Khan, S.S, Ahmad, A., "Cluster Center Inialization Algorithm for K-means Clustering", ***Pattern Recognation Letters,*** 25, 1293 – 1302, (2004).

[25] Al-Daoud, M.B., "A New Algorithm for Cluster Initialization", ***Proceedings of World Academy of Science, Engineering and Technology,*** 4, 74 – 76, (2005).

[26] Caoa, F., Liang, J., Jiang, G., "An initialization method for the K-Means algorithm using neighborhood model" ***Computers and Mathematics with Applications***, 58, 474-483, (2009).

[27] Kondadadi, R., Kozma, R., "A Modified Fuzzy ART for Soft Document Clustering", ***Neural Networks***, 3, 2545-2549, (2002).
[28] Cinque, L., Foresti, G., Lombardi, L., "A clustering fuzzy approach for image segmentation", ***Pattern Recognition***, 37, 1797-1807, (2004).

[29] Chen, C., Wang, L., "An Efficient and Applicable Clustering Algorithm Using Fuzzy ART", ***Proceeding s of the 6th World Congress on Intelligent Controland Automation***, 3178-3182, (2006).

[30] Xiang, G., Min, W., Rongchun Z., "Application of Fuzzy ART for Unsupervised Anomaly Detection System", International Conference on Computational Intelligence and Security, 2, 621-624, (2006).

[31] Xu, R., Damelin, S., Wunsch D.C., "Applications of Diffusion Maps in Gene Expression Data-Based Cancer Diagnosis Analysis", ***Proceedings of the 29th Annual International Conference of the IEEE***, 4613-4616, (2007).

[32] Kumar, M., Verma, S., Singh P.P, "Data Clustering in Sensor Networks using ART", ***Wireless Communication and Sensor Networks***, 51-56, (2008).

[33] Isawa, H., Matsushita, H., Nishio Y., "Fuzzy Adaptive Resonance Theory Combining Overlapped Category in Consideration of Connections", ***IJCNN IEEE***, 3595-3600, (2008).

[34] Gu, M., Zhou, J-Z., Li, J_Z., "Onlıne Face Recognıtıon Algorıthm Based On Fuzzy Art", ***Proceedings of the Seventh International Conference on Machine Learning and Cybernetics***, 556-560, (2008).

[35] Fayyad, U. M., Piatetsky-Shapiro, G., Uthurusamy, R., "Advances in Knowdedge Discovery and Data Mining", ***Cambridge, MA: MIT Press.,*** (1996a).

[36] http://herseyekitap.googlepages.com/YapaySinirAglari.doc, ***(Ziyaret tarihi: 02.06.2009)***

[37] Silahtaroğlu, G., "Kavram ve Algoritmalarıyla Temel Veri Madenciliği", ***Papatya Yayıncılık Eğitim A.Ş.,*** İstanbul, 2008.

[38] Deogun, J. S., Raghavan, V. V., Sever, H., "Exploiting upper approximations in the rough set methodology", Fayyad, U., Uthurusamy, R., (eds.), ***The First International Conference on Knowledge Discovery and Data Mining,*** Montreal, Quebec, Canada, 69 – 74, (1995).

[39] Berry, M.J.A., Linoff, G.S., "Data Mining Techniques, For Marketing, Sales and Customer Relationship Management", ***Wiley Publishing Inc.,*** Second Edition, Canada, (2004).

[40] Fakih, S., Das, T., "A methodology for learning efficient approaches to medical diagnosis", ***Information Technology in Biomedicine IEEE Transactions on***, 10(2), 220 - 228, (2006).

[41] http://www3.itu.edu.tr/~sgunduz/courses/verimaden/slides/d3.pdf, ***(Ziyaret tarihi: 03.05.2009)***

[42] Morana, H., Camara, F., Arboleda-Florez J., "Cluster analysis of a forensic population with antisocial personality disorder", ***Forensic Science International***, (2006).

[43] Larose, D.T., Discovering Knowledge in Data, ***Wiley,*** 978-0-471-66657-8, (2005).

[44] He, J., Lan, M., Tan, C-L., Sung, S-Y., Low, H-B., "Initialization of Cluster Refinement Algorithms: AReview and Comparative Study", ***Neural Networks IEEE International Joint Conference,*** 1, 297 – 302, (2004).

[45] Dunham, M. H., "Data Mining Introductory and Advanced Topics", ***Prentice Hall,*** New Jersey, (2003).

[46] Tan, P.N., Steinbach, M., Kumar, V., "Inroduction to Data Mining", ***Pearson Education,*** (2006).

[47] Boley, D. L., "Principal direction divisive partitioning", ***Data Mining and Knowledge Discovery,*** 2(4), 324 – 344, 1998.

[48] Paccanaro, A., Casbon, J. A., Saqi, M. A. S., "Spectral clustering of protein sequences", ***Nucleic Acids Research,*** 34(5), 1571 – 1580, (2006).

[49] Saux, B. L., Boujemaa, N., "Unsupervised robust clustering for image database categorization", ***International Conference on Pattern Recognition,***1, 10259, (2002).

[50] Rosenberger, C., Chehdi, K. "Unsupervised clustering method with optimal estimation of the number of clusters: Application to image segmentation", ***International Conference on Pattern Recognition,*** 1, 1656, (2000).

[51] Tritchler, D., Fallah, S., Beyene, J., "A spectral clustering method for microarray data", ***Computer. Statistics & Data Analysis,*** 49, 63 – 76, (2005).

[52] Fang, H-R., "Farthest Centroids Divisive Clustering", ***Seventh International Conference on Machine Learning and Applications,*** ,232-238, (2008).

[53] Anderberg, M.R., **"**Cluster Analysis for Applications", ***Academic Press Inc.,*** New York, (1973).

[54] Diday, E., Simon, J.C., **"**Clustering analysis in Digital Pattern Recognition", ***Springer-Verlag,*** Secaucus, NJ, 47 - 94, (1976).

[55] Michalski, R., Stepp, R. E., Diday, E., "Automated construction of classifications:conceptual clustering versus numerical taxonomy", **IEEE Trans. Pattern Anal Mach. Intell.,** PAMI-5, 396 - 409, (1983).

[56] Berkhin P., "Survey of Clustering Data Mining Techniques", ***Acurate Software Inc.,*** (2002).

[57] http://www.cs.sfu.ca/~han/bk/8clst.ppt, ***(Ziyaret tarihi: 25.05.2009)***

[58] Fayyad U., "Mining Databases: Towards Algorithms for Knowledge Discovery", ***IEEE Bulletin of the Technical Committee on Data Engineering,*** 21(1),41 – 48, (1998).

[59] Rhee, H-S., Oh, K-W., "A Performance Measure for the Fuzzy Cluster Validity", ***Soft Computing in Intelligent Systems and Information Processing,*** 364-369, (1996).

[60] Xie, X.L., Beni, G., "A Validity Measure for Fuzzy Clustering", ***IEEE Trans. Pattern Anal. Machine Intelligence,*** 13(8), 841 - 847, (1991).

[61] Sun, H., Wang, S., Jiang, Q., "FCM-Based Model Selection Algorithms for Determining the Number of Clusters", ***Pattern Recognition,*** 37, 2027 - 2037, (2004).

[62] Wu., K-L., Yang, M-S., "A cluster validity index for fuzzy clustering", ***Pattern Recognition Letters,*** 26(9), 1275 – 1291, (2005).

[63] Millligan, G.W., Cooper, M.C., "An Examination of Procedures for Determining the Number of Clusters in a Data Set", ***Psychometrika,*** 50, 159 – 179, (1985).

[64] Nikhil, R., James, C., "On Cluster Validity for the Fuzzy c-Means Model", ***IEEE Trans. Fuzzy Systems,*** 3(3), 370 – 379, (1995).

[65] Gath, I., Geva, A., "Unsupervised Optimal Fuzzy Clustering", ***IEEE Trans. Pattern Analysis and Machine Intelligence,*** 11(7), 773 – 781, (1989).

[66] Rezaee, M., Lelieveldt, B., Reiber, J., "A New Cluster Validity Index for the Fuzzy c-Mean", ***Pattern Recognition Letters,*** 19, 237 – 246, (1998).

[67] El-Melegy, M. Zanaty, E.A., Abd-Elhafiez, W.M., Farag, A., "On Cluster Validity Indexes in Fuzzy and Hard Clustering Algorithms for Image Segmentation", ***IEEE International Conference on Image Processing,*** 6, 5-8, (2007).

[68] Chen, C., Wang, L., "An Efficient and Applicable Clustering Algorithm Using Fuzzy Art", ***Proceedings of the 6th World Congress on Intelligent Control and Automation,*** 3178 – 3182, (2006).

[69] Pena, J.M, Lozano, J.A., Larranaga, P., "An empirical comparison of four initialization methods for the K-means Algorithm", ***Pattern Recognation Letters,*** 1293 – 1302, (2004).

[70] Li, M.J., Michael K. Ng, Cheung, Y-M., "Agglomerative Fuzzy K-means Clustering Algorithm with Selection of Number of Clusters", ***IEEE Transactions on Knowledge and Data Engineering,*** 20(11), 1519 – 1534, (2008).

[71] Demiralay, M., Çamurcu, Y., Cure, "Agnes ve K-means Algoritmalarındaki Kümeleme Yeteneklerinin Karşılaştırılması", ***İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi,*** 4(8), 1 – 18, (2005).

[72] Forgy, E., "Cluster Analysis of Multivariative Data: Efficiency vs. Interpretability of Classifications", ***In Wnar Meetings, Univ of Calif Riverside,*** 21, 768 – 769, (1965).

[73] Tou, J.T., Gonzalez, R.C., "Pattern Recognition Principles", ***Addison-Wesley,*** Massachusetts, (1974).

[74] Katsavounidis, I., Kuo, C., Zhang, Z., "A New Initialization Technique for Generalized Lloyd Iteration", **IEEE Signal Processing Letters,** 1, 144 – 146, (1994).

[75] Kaufman, L., Rousseeuw, P.J., "Finding Groups in Data: An Introduction to Cluster Analysis", ***Wiley,*** New York, (1990).

[76] McCulloch, W. S., Pitts, W. H., "A logical calculus of the ideas immanent in nervous activity", ***Bulletin of Mathematical Biophysics,*** 5, 115 – 133, (1943).

[77] Şen, Z., "Yapay Sinir Ağları İlkeleri", ***Su Vakfı Yayınları***, İstanbul, (2004).

[78] Öztemel, E., "Yapay Sinir Ağları", ***Papatya Yayıncılık***, İstanbul, (2003)

[79] Elmas, Ç., "Yapay Sinir Ağları", ***Seçkin Yayıncılık A.Ş.***, Ankara, (2003).

[80] Efe, M.Ö., Kaynak, O., "Yapay Sinir Ağları ve Uygulamaları", ***Boğaziçi Üniversitesi Yayınları***, (2000).

[81] Fausett, L., "Fundementals of Neural Networks Architectures, Algorithms and Applications", ***Prentice Hall,*** (1994).

[82] Carpenter G.A., Grosberg S., "ART2: Self-organisation of stable category recognation codes for analog input patterns", ***Applied Optics,*** 26, 4919 – 4930, (1987).

[83] Grossberg, S., "Classical and Instrumental Learning by Neural Networks", ***Progress in Theoretical Biology,*** 3, 51 – 141, (1977).

[84] Rumelhart, D. E., Zipser, D., "Feature discovery by competitive learning", ***Cognitive Science,*** 9, 75 – 112, (1985).

[85] Kröse, B., Van Der Smagt, P., "An Introduction to Neural Networks", ***University of Amsterdam,*** Eight Edition, (1996).

[86] Kohonen, T., "A Simple Paradigm for the Self-organized Formation of Structured Feature Maps", Competition and Cooperation in Neural Nets, (1982).

[87] Kohonen, T., Self-organization and Associative Memory, ***Springer-Verlag,*** Berlin, (1984).

[88] Zurada, J.M., "An Instroduction to Artificial Neural Systems", ***West Publishing Company, St. Paul,*** New York, USA, (1992).

[89] Rojas, R., "Neural Networks A Systematic Introduction", ***Springer-Verlag,*** Berlin, (1996).

[90] Hopgood, A.A., "Intelligent Systems For Engineers and Scientists", ***Crc Press, Boca Raton,*** Washington D.C., (2001).

[91] Kasabow, N.K., "Fundementals of Neural Networks, fuzzy systems and Knowledge engineering", ***The MIT press,*** Cambridge, England., (1998).

[92] Carpenter, G. A,, Grossberg, S., "Neural Dynamics of Category Learning and Recognition: Attention, Memory Consolidation and Amnesia", ***Brain Structure, Learning and Memory,*** ed. J. Davis, R. Newburgh, I. Wegman. AAAS Symp. Series. Westview Press, Boulder, Colo., (1988).

[93] Carpenter G.A., Grosberg S., "Category Learning and adaptive pattern recognation: A neural network model", ***Proc. of the third Army Conference on Applied Mathematics and Computing,*** 81 (1), 37 – 56, (1986).

[94] Kuan, M.M., Lim, C.P., Harrison, R.F., "On Operating Strategies of The Fuzzy ARTMAP Neural Network: A Comparative Study", ***International Journal of Computational Intelligence and Applications,*** 3 (1), 23 – 43, (2003).

[95] Yeo, N.C., Lee, K.H., Venkatesh, Y.V., Ong, S.H., "Colour Image Segmentation Using The Self – Organizing Map and Adaptive Resonance Theory", ***Image and Vision Computing,*** 1 – 20, (2005).

[96] Cheng, C.H., "A Comparative Examination of Selected Cellular Manufacturing Clustering Algorithms", ***International Journal of Operations and Production Management,*** Vol. 15, No. 12, 86-97, (1995).

[97] Singer, S., Venetsky,L., Lynch, M.L., "Virtual Test Automation Generator (VTAG)", ***Navair Lakehurst,*** 1 – 10, (2000).

[98] Simpson, P.K., "Fuzzy Min-Max neural Networks—Part 2: clustering", **IEEE Trans. Fuzzy Systems,** 1(1), 32 – 45, (1993).

[99] Granger, E., Blaquiere, Y.S., Cantin, M.A., Lavoie, P., "A VLSI Architecture for Fast Clustering With Fuzzy ART", ***Neural Networks,*** 0-8186-7456-3, IEEE, (1996).

[100] Mun-Hwa, K., Dong-Sik, J., Young-Kyu, Y., "A Robust-Invariant Pattern Recognition Model Using Bulanık ART", ***Pattern Recognition,*** 34, 1685 – 1696, (2001).

[101] Murphy, P.M., Alia, D. W., 1994, UCI repository of machine leaning databases, University of California-Irvine, Department of Information and Computer Science, http://www.ics.uci.edu/~mlearn/MLRepository.html, ***(Ziyaret tarihi: 15.04.2009)***.

[102] Zaïane, O., and Pei, Y., Sentetik Veri Seti Documents_Sim, Mars ve Image Extraction'ın kaynağı: http://www.cs.ualberta.ca/~yaling/Cluster/Php/index.php, ***(Ziyaret tarihi: 15.04.2009).***

[103] Smith L.I., "A Tutorial on Principal Component Analysis" (2002).

EKLER

İris Veri Kümesi

Karakteristiği:	Çok değişkenli
Nitelik karakteristiği:	Reel sayı
İlgili alanlar:	Sınıflandırma
Veri sayısı:	150
Nitelik sayısı:	4
Kayıp değerler?	Yok
Veri tabanına eklenme yılı:	1988
Başvurulma sayısı:	74613

Wine Veri Kümesi

Karakteristiği:	Çok değişkenli
Nitelik karakteristiği:	Tamsayı, reel sayı
İlgili alanlar:	Sınıflandırma
Veri sayısı:	178
Nitelik sayısı:	13
Kayıp değerler?	Yok
Veri tabanına eklenme yılı:	1991
Başvurulma sayısı:	50561

Hepatitis Veri Kümesi

Karakteristiği:	Çok değişkenli
Nitelik karakteristiği:	Tamsayı, reel sayı
İlgili alanlar:	Sınıflandırma
Veri sayısı:	155
Nitelik sayısı:	19
Kayıp değerler?	Var
Veri tabanına eklenme yılı:	1988
Başvurulma sayısı:	8785

Pima Indians Veri Kümesi

Karakteristiği:	Çok değişkenli
Nitelik karakteristiği:	Tamsayı, reel sayı
İlgili alanlar:	Sınıflandırma
Veri sayısı:	768
Nitelik sayısı:	8
Kayıp değerler?	Yok
Veri tabanına eklenme yılı:	1990
Başvurulma sayısı:	13321

Haberman's Survival Veri Kümesi

Karakteristiği:	Çok değişkenli
Nitelik karakteristiği:	Tamsayı
İlgili alanlar:	Sınıflandırma
Veri sayısı:	306
Nitelik sayısı:	3
Kayıp değerler?	Yok
Veri tabanına eklenme yılı:	1999
Başvurulma sayısı:	7619

Heart-Disease-Cleveland Veri Kümesi

Karakteristiği:	Çok değişkenli
Nitelik karakteristiği:	Tamsayı, reel sayı
İlgili alanlar:	Sınıflandırma
Veri sayısı:	303
Nitelik sayısı:	14
Kayıp değerler?	Var
Veri tabanına eklenme yılı:	1988
Başvurulma sayısı:	15540

Ruspini Veri Kümesi

Karakteristiği:	Çok değişkenli
Nitelik karakteristiği:	Tamsayı
İlgili alanlar:	Kümeleme
Veri sayısı:	75
Nitelik sayısı:	2
Kayıp değerler?	Yok

Letter Recognition Veri Kümesi

Karakteristiği:	Çok değişkenli
Nitelik Karakteristiği:	Tamsayı
İlgili alanlar:	Sınıflandırma
Veri sayısı:	20000
Nitelik sayısı:	16
Kayıp değerler?	Yok
Veri tabanına eklenme yılı:	1991
Başvurulma sayısı:	13813

KİŞİSEL YAYINLAR

İlhan, S., Duru, N., "Improved Fuzzy Art and Rough Sets: A Hybrid Solution to Characterize Clustered Data", ***International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)***, Trabzon-Turkey, 2009.

İlhan, S., Duru, N., "Improved Fuzzy Art for Clustering Patterns", ***International Workshop on Applications of Wavelets to Real World Problems,*** Kocaeli-Turkey, 2009

İlhan, S., Duru, N., "An Improved Method for Fuzzy Clustering", ***Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control,*** Famagusta-NortCyprus, 2009.

ÖZGEÇMİŞ

1979 yılında Burdur' da doğdu. İlk, orta ve lise öğrenimini Burdur' da tamamladı. 1997 yılında girdiği Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü' nden 2001 yılında Bilgisayar Mühendisi olarak mezun oldu. 2001-2004 yılları arasında Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı' nda Yüksek Lisans Öğrenimini tamamladı. 2004 yılında Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Mühendisliği Anabilim Dalı' nda Doktora programına başladı. 2001-2004 yılları arasında Kocaeli Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü' nde Araştırma Görevilisi olarak görev yapmıştı. 2004 yılından beri aynı bölümde Öğretim Görevlisi olarak görev yapmaktadır.