SEZGİSEL YÖNTEMLERLE K-HARMONİK ORTALAMA VERİ KÜMELEME ENİYİLEMESİ

Alper ÜNLER

DOKTORA TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ARALIK 2006
ANKARA

Alper ÜNLER tarafından hazırlanan SEZGİSEL YÖNTEMLERLE K-HARMONİK ORTALAMA VERİ KÜMELEME ENİYİLEMESİ adlı bu tezin Doktora tezi olarak uygun olduğunu onaylarım.

Prof.Dr.Zülal GÜNGÖR
Tez Yöneticisi

Bu çalışma jürimiz tarafından oybirliğiyle Endüstri Mühendisliği Anabilim Dalında Doktora tezi olarak kabul edilmiştir.

Başkan : Prof.Dr.Fevzi KUTAY

Üye : Prof.Dr.Zülal GÜNGÖR

Üye : Prof.Dr.Berna DENGİZ

Üye : Prof.Dr.Semra ORAL ERBAŞ

Üye : Prof.Dr.Nizami AKTÜRK

Üye :

Tarih : 13.12.2006

Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orjinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Alper ÜNLER

SEZGİSEL YÖNTEMLERLE K HARMONİK ORTALAMA VERİ KÜMELEME ENİYİLEME

(Doktora Tezi)

Alper ÜNLER

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Aralık 2006

ÖZET

Veri madenciliği ve vektör nicemleme alanlarındaki en önemli problemlerden birisi veri kümeleme problemidir. Veri kümeleme analizi temelde hiyerarşik ve hiyerarşik olmayan veri kümeleme olarak ikiye ayrılmaktadır. Hiyerarşik olmayan veri kümelemede merkez tabanlı kümeleme en çok kullanılan yöntemdir ve merkez tabanlı kümelemenin temelini K-Ortalama kümeleme algoritması oluşturmaktadır. Kullanımı ve anlaşılması çok kolay bir algoritma olmasına rağmen K-Ortalama algoritmasının ilk seçilen merkezlere duyarlılık ve yerel eniyi noktasına takılma gibi iki temel problemi vardır. Diğer bir merkez tabanlı kümeleme algoritması olarak önerilen K-Harmonik Ortalama kümeleme yöntemi ilk merkezlere karşı duyarlılık problemine çözüm olarak önerilmiştir. Ancak amaç fonksiyonunun birden fazla yerel en küçük noktaya sahip olması nedeniyle yerel eniyiye takılma problemi halen devam etmektedir. Bu çalışmada K-Harmonik Ortalama veri kümelemede yerel eniyiye takılmamak ve en iyiye yakın çözümler bulmak amacıyla sezgisel yöntemler geliştirilmiştir. Tabu arama, tavlama benzetimi ve parçacık sürü en iyilemesi gibi üç temel sezgisel yöntem ayrı ayrı K-Harmonik Ortalama kümeleme algoritması ile birleştirilerek melez yöntemler oluşturulmuştur (TABAKHO, TAVBEKHO, PARSEKHO).

Yapılan test ve analiz çalışmalarında önerilen yöntemlerin diğer yöntemlerden daha iyi çözümler ürettiği gözlemlenmiştir.

Bilim Kodu : **906.1.148**
Anahtar Kelimeler :**Veri kümeleme, K-Ortalama, Bulanık K-Ortalama, K-Harmonik Ortalama, tabu arama, tavlama benzetimi, Parçacık sürü en iyilemesi**
Sayfa Adedi :**132**
Tez Yöneticisi :**Prof.Dr.Zülal GÜNGÖR**

OPTIMIZATION OF K HARMONIC MEANS DATA CLUSTERING WITH METAHEURISTICS

(Ph.D. Thesis)

Alper ÜNLER

GAZİ UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

December 2006

ABSTRACT

Data clustering analysis is the basic problem of data mining and vector quantization. There are basically two types of clustering: Hierarchical and partitional. Center based clustering methods are the most common ones in partitional clustering. K-Means clustering algorithm forms the basis of the center based clustering. Although it is very easy to implement and understand, K-Means clustering algorithm, it suffers form two major drawbacks; Firstly it is sensitive to initial conditions and secondly it converges to a local optimum. K-Harmonic Means data clustering method considerably solves the sensitivity problem. Since the objective function is nonconvex and there exists many local minima, converging to a local optimum is still a problem of K-Harmonic Means data clustering. In this study, some heuristics like tabu search, simulated annealing and particle swarm optimization are integrated with K-Harmonic Means method to solve the local optimum problem. Three hybrid algorithms are developed and implemented during this PhD study. The names of the algorithms are TABAKHO, TAVBEKHO and PARSEKHO.

The hybrids developped in this study are tested on the well known datasets and in most cases they overperformed the alternative methods.

Science Code : 906.1.148
Key Words :Data clustering, K-Means, Fuzzy K-Means, K-Harmonic Means, tabu search, simulated annealing, particle swarm optimization
Page Number :132
Advisor :Prof.Dr.Zülal GÜNGÖR

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren hocam Prof.Dr. Zülal GÜNGÖR'e, yine kıymetli tecrübelerinden faydalandığım hocalarım Prof.Dr. Fevzi KUTAY'a, jüri üyeleri Prof.Dr. Berna DENGİZ'e, Prof.Dr. Semra ORAL ERBAŞ'a ve Prof.Dr. Nizami AKTÜRK'e, işyerimde bana sınırsız destek olan Alb. Levent ALTUNCU'ya ve diğer çalışma arkadaşlarıma ve varlıklarıyla bana her zaman büyük güç katan annem Asiye ÜNLER, kardeşim İsmail ÜNLER ve değerli eşim Av. Kadriye ÜNLER'e, canımdan çok sevdiğim yavrularım Ahmet ve Eralp'e teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklama**
$\boldsymbol{C}$	Pozisyon vektörleri
c_1	Bilişsel öğrenme hızı
c_2	Sosyal öğrenme hızı
$\boldsymbol{d}$	Uzaklık
$\boldsymbol{gbest}$	Tümel eniyi vektör
$\boldsymbol{K}$	Küme sayısı
$\boldsymbol{m}$	Bulanıklaştırıcı
$\boldsymbol{N}$	Veri sayısı
$\boldsymbol{P}$	Olasılık eşiği
$\boldsymbol{p}$	KHO kümelede üstel değişken
$\boldsymbol{pbest}$	Yerel eniyi vektör
$\boldsymbol{r}$	Rasgele sayı
$\boldsymbol{Tmax}$	Başlangıç sıcaklığı
$\boldsymbol{Tmin}$	Durma sıcaklığı
$\boldsymbol{U}$	Üyelik değerleri matrisi
$\boldsymbol{V}$	Hız vektörleri
$\boldsymbol{X}$	Veri matrisi
ω	Atalet ağırlığı

Kısaltmalar	**Açıklama**
ALÇS	Alternatif Çözüm Sayısı
BIC	Bayes Information Criterion
BKO	Bulanık K-Ortalama

Kısaltmalar	Açıklama
CLIQUE	Clustering In QUEst
DENCLUE	DENsity-based CLUstEring
EBTLU	En Büyük Tabu Listesi Uzunluğu
EM	Expectation Maximization
FA	Forgy Approach
GRASP	Greedy Randomized Search Procedure
ISODATA	Iterative Self Organizing Data Analysis
YİNESA	İterasyon Sayısı
KA	Kaufman Approach
KHO	K-Harmonik Ortalama
KO	K-Ortalama
LBG	Linde Buzo Gray
MA	Macqueen Approach
MC	Mariotte Criterion
MDL	Minimum Description Length
MIN	Minimum
PAM	Partitioning Around Medoids
PC	Partitioning Coefficient
PE	Partitioning Entropy
PSE	Parçacık Sürü Eniyileme
SOM	Self Organizing Maps
STING	Statistical Information Grid
TA	Tabu Arama
TB	Tavlama Benzetimi
TSGO	Tepe Sinyal Gürültü Oranı
VN	Vektör Nicemleme
VNS	Variable Neighborhood Search
VRC	Variance Ratio Criterion

1. GİRİŞ

Teknolojik gelişmelerin başdöndürücü hızla ilerlediği günümüz iş ve bilim dünyasında veri toplama, veriden bilgi elde etme, veriyi depolama ve transfer etme konularına gereksinim her geçen gün daha da artmaktadır. Bilgi, sermayenin en önemli bileşenlerinden birisi haline gelmiştir. Organizasyonlar günlük operasyonları yürütebilmek ve geleceğe yönelik kararlar alabilmek için bilgi yönetimine çok büyük boyutlarda yatırım yapmaktadırlar. Veritabanlarında depolanan verinin boyutu terrabaytlar seviyesine gelmiştir. Bu şekilde depolanan veri, ham şekliyle karar alma süreçlerinde kullanılamadığından, veri madenciliği adı verilen analiz yöntemleriyle ham veriden bilgiye ulaşılmaya çalışılmaktadır. Bunun yanında özellikle ses ve görüntü verilerinin etkin olarak depolanması ve transfer edilmesi konuları da günümüzün en önemli bilgi sistem gereksinimlerinden birisi haline gelmiştir.

Veri kümeleme analizi, hem veri madenciliğinin, hem de veri depolama ve veri transferinin en önemli yöntemlerinden birisidir. Eldeki verinin durumu, teknik altyapı ve uygulama alanına bağlı olarak bir çok farklı veri kümeleme yöntemi mevcuttur.

Veri kümeleme yöntemleri en genel olarak hiyerarşik ve hiyerarşik olmayan kümeleme olarak ikiye ayrılmakla birlikte, üzerinde en çok çalışma yapılan yöntem hiyerarşik olmayan kümeleme yöntemidir. Hiyerarşik olmayan kümelemenin temel modeli merkez tabanlı kümeleme modelidir. Yazında bu yaklaşıma prototip bazlı kümeleme, ortalama kümeleme isimleri de verilmektedir. Merkez tabanlı kümeleme, küme temsilcisi olarak elde edilen kümelerin merkez koordinatlarının seçildiği kümeleme yöntemidir.

Merkez tabanlı veri kümeleme analizi üzerindeki çalışmalar yaklaşık 40 yıldır devam etmektedir [Forgy, 1965, Ball ve Hall, 1967, King, 1967, MacQueen, 1967]. Bu konudaki en genel yöntem K-Ortalama veri kümeleme yöntemidir. Diğer bütün merkez tabanlı kümeleme yöntemleri K-Ortalama yöntemini temel almaktadır.

Merkez tabanlı kümeleme analizi probleminin np-zor problem olması nedeniyle analitik yöntemlerle çözülmesi mümkün görünmemektedir [Garey, 1982]. K-Ortalama kümeleme modelinin bu kadar geniş kabul görmesinin nedeni, kümeleme analizi problemine uygulanması, anlaşılması ve yorumlanması oldukça kolay bir çözüm getirmesidir.

K-Ortalama kümeleme yöntemi başka bir çok yönteme temel teşkil etmesine ve en çok kullanılan yöntem olmasına rağmen zayıf tarafları da mevcuttur;

- Küme sayısı (K) değerini kullanıcıdan istemesi,
- Aykırı elamanlara karşı duyarlı olması,
- Başlangıçta belirlenen merkezlere duyarlı olması,
- Yerel en küçük tuzağına takılması, ve
- Sadece sayısal veri üzerinde çalışması,

K-Ortalama kümeleme yönteminin zayıf taraflarının en önemlileridir.

Harmonik ortalama fonksiyonunun K-Ortalama kümelemede kullanılan en küçük fonksiyonuna çok benzer olması nedeniyle, K-Ortalamadaki en küçük fonksiyonu harmonik ortalama ile değiştirilerek, K-Harmonik Ortalama (KHO) veri kümeleme yöntemi önerilmiş ve önerilen yöntemin özellikle seçilen ilk merkezlere karşı duyarlılığı büyük ölçüde azalttığı yapılan testlerle ortaya konulmuştur [Zhang, 1999, 2000]. K-Harmonik Ortalama kümeleme amaç fonksiyonunda, K-Ortalamaya benzer şekilde arama uzayında birden fazla yerel en küçük noktası bulunmakta ve önerilen yöntemin yerel en küçük tuzağına takılma riski hala devam etmektedir.

Birden fazla yerel en küçük noktaya sahip amaç fonksiyonlu eniyileme problemlerinin yerel minimuma takılmadan çözümü ve/veya tümel eniyi çözüme yakın çözümler bulmak için doğadaki canlıların yaşama, üreme, beslenme gibi davranışlarını modelleyerek geliştirilmiş sezgisel yöntemler bulunmaktadır. İnsan hafızasının hatırlama ve unutma prensiplerinden yola çıkan tabu arama yöntemi, katıların ısıtılıp soğutulması yoluyla tavlanması prensibinden yola çıkan tavlama

benzetimi, evrim teorisinin esaslarından yola çıkan genetik algoritmalar, karınca, kuş vb. gibi sürü halinde yaşayan hayvanların yaşam felsefelerinden yola çıkan sürü algoritmaları (karınca-koloni, parça-sürü vb. gibi) bu sezgisel yöntemlerden en çok kullanılanlardır.

Fritzke (1997) ve Patane ve Russo (2001), hiyerarşik olmayan kümelemede iyi bir kümeleme analizi sonucu oluşan küme yapısında her bir kümenin toplam dağınıklığa eşit oranda katkıda bulunması gerektiğini belirtmektedir. Vektör nicemleme alanında önerilen bu yaklaşım bu çalışmada da en iyi merkez noktaları arama stratejisi olarak kullanılmaktadır.

Bu çalışmada K-Harmonik Ortalama veri kümeleme probleminin çözümünde tümel eniyi çözüme yakın çözümler elde edebilmek için, tabu arama, tavlama benzetimi ve parçacık sürü en iyilemesi yöntemleri, K-Harmonik Ortalama kümeleme yöntemi ile birleştirilerek melez algoritmalar geliştirilmiştir. En iyi küme yapısını oluşturan merkez koordinatları aranırken arama stratejisi olarak fayda fonksiyonu [Fritzke, 1997 - Patane and Russo, 2001] adı verilen fonksiyondan yararlanılmış, ayrı ayrı kümelerin dağınıkları dengelenmeye çalışılmıştır.

Geliştirilen sezgisel yöntemlere tabu arama ile K-Harmonik Ortalama veri kümeleme (*TABAKHO*), tavlama benzetimi ile K-Harmonik Ortalama veri kümeleme (*TAVBEKHO*), parçacık sürü en iyilemesi ile K-Harmonik Ortalama veri kümeleme (*PARSEKHO*) isimleri verilmiştir. Geliştirilen sezgisel yöntemler literatürde çok kullanılan veri yığınları üzerinde test edilmiş ve genellikle alternatiflerinden daha iyi sonuçlar ürettikleri tespit edilmiştir.

Çalışmanın ikinci bölümünde veri kümeleme analizi, veri kümeleme analizinin diğer disiplinlerle ilişkisi ve kullanılan parametreler hakkında detaylı bilgiler verilmektedir. Üçüncü bölümde temel kümeleme yaklaşımları olan hiyerarşik kümeleme ve hiyerarşik olmayan kümeleme ve bu yaklaşımlara ait yöntemler anlatılmaktadır. Dördüncü bölümde merkez tabanlı kümeleme ile ilgili problem sahaları ve literatürde yapılan çalışmalar detaylı olarak anlatılmaktadır. Beşinci

bölümde sezgisel arama yöntemlerinden tabu arama tekniği, tavlama benzetimi tekniği ve parçacık sürü en iyilemesi tekniği hakkında bilgiler verilmekte ve bu tekniklerin K-Harmonik Ortalama kümeleme ile entegrasyonu anlatılmakta, geliştirilen sezgisel yöntemlerde kullanılacak uygun parametre kombinasyonları belirlenmekte ve önerilen sezgisel yöntemlerin algoritmaları verilmekte, ayrıca yapılan test çalışmaları anlatılmaktadır. Altıncı ve son bölümde ise önerilen yöntemlerin yorumlanması ve ilerde yapılacak olan çalışmalar hakkında bilgiler verilmektedir.

2. KÜMELEME ANALİZİ

Veri toplama ve analiz işlemleri gerçek hayat problemlerini tespit etmek ve çözüm üretmek için üzerinde uzun zamandan beri çalışılan bir konudur. 1854 yılında Londra'daki bir kolera salgını sırasında John Snow adında bir görevli kendisine rapor edilen vakaları üzerinde işaretlemek için özel bir harita oluşturmuş ve kullanmıştır [Gilbert, 1958]. Harita üzerinde işaretlenen vakaları inceleyen görevli, merkezdeki bir cadde üzerindeki yerleşim yerlerinde hastalığın yoğunlaştığını ve diğer bölgelere buradan dağıldığı gerçeğini tespit etmiştir. Gilbert (1958), bu çalışmayı literatürdeki ilk veri kümeleme analizi çalışması olarak değerlendirmektedir.

Fisher (1936) tarafından önerilen doğrusal ayırma analizi yöntemi veri kümeleme analizi literatüründeki temel aşamalardan birisidir. Önerilen yöntem bireylerin veya nesnelerin çeşitli özelliklerine ait ölçümlerden yararlanarak mevcut grupları ayırmada kullanılacak uygun fonksiyonları belirlemede kullanılmaktadır.

Veri kümeleme analizi kavram olarak ilk kez Tyron (1939) tarafından kullanılmıştır. Clements (1954), antropolojik verinin kümelenmesi çalışmasında veri kümeleme analizi kavramını kullanmıştır. Cox (1957) ve Fisher (1958), kümeleme analizi kavramını gruplama adını verdikleri çalışmalarında kullanmışlardır. Sokal ve Sneath (1963), sayısal sınıflandırmanın prensipleri adını verdikleri kitapta kümeleme analizini anlatmışlardır. 1960'lı yılların sonundan itibaren de kümeleme analizi hakkındaki çalışmalar günümüze kadar artarak devam etmiştir.

2.1. Kümeleme Problemi Tanımı ve Kullanım Alanları

Kümeleme analizi; veri madenciliği ve istatistik alanından bakıldığında veri analizi ve desen tanıma işlemleri, vektör nicemleme alanından bakıldığında ise depolama ve transfer işlemleri için kullanılan bir veri işleme yöntemidir. Bu bölümde farklı kaynaklardan alınan veri kümeleme analizi tanımları verilmektedir.

Kümeleme analizi bir yığını oluşturan bireylerin birden fazla özelliklerini karşılaştırmak suretiyle farklı alt gruplara bölünmesini sağlayan istatistiksel sınıflandırma tekniği olarak tanımlanmaktadır (Webster sözlüğü, 2005).

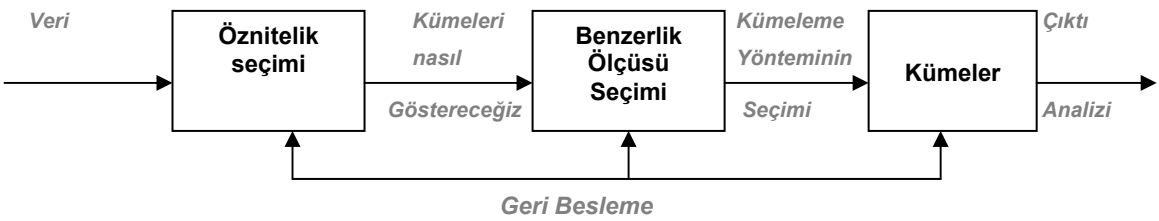
Mac Quenn (1967), Webster tanımına benzer şekilde kümeleme analizini bireylere ait çok boyutlu gözlem değerlerinin analizi ve sınıflandırılması olarak tanımlamaktadır.

Kümeleme analizi bir araştırmada incelenen birimleri aralarındaki benzerliklerine göre belirli gruplar içinde toplayarak sınıflandırma yapmayı, birimlerin ortak özelliklerini ortaya koymayı ve bu sınıflar ile ilgili genel tanımlar yapmayı sağlayan bir yöntemdir [Kauffman, 1990].

Kümeleme analizi *X* veri matrisinde yer alan ve doğal gruplamaları kesin olarak bilinmeyen birimleri, değişkenleri ya da birim ve değişkenleri birbirleri ile benzer olan alt kümelere ayırmaya yardımcı olan yöntemler topluluğudur [Ye, 2003].

Kümeleme analizi elde bulunan veri yığınının belirlenen yöntemlerle analiz edilerek daha önceden etiketleri belli olmayan gruplara ayrılması işlemidir.Bu işlem sonucunda elde edilen kümeler yüksek düzeyde küme içi homojenlik ve kümeler arası heterojenlik gösterirler [Kantardzic, 2003].

Kümeleme analizi etkileşimli ve geri beslemeli bir süreçtir. Temel olarak ham veri üzerinde değişim ve dönüşüm işleminin yapılması, uygun benzerlik ölçüsü ve kümeleme kriterinin seçimi ve çıktıların analiz edilmesi aşamalarından oluşmaktadır. Şekil 2.1'de kümeleme analize ait akış şeması görülmektedir.



Şekil 2.1. Kümeleme probleminin çözüm aşamaları [Jain ve ark. 1999]

Anderberg (1973), kümeleme problemini dokuz aşamalı bir problem olarak tanımlamaktadır

1) Nesnelerin seçimi
2) Nesnelere ait özniteliklerin seçilmesi
3) Neyin kümelemeleneceğinin belirlenmesi (verinin tamamı mı yoksa özniteliklerden bazıları mı?)
4) Verinin normalize edilmesi
5) Benzerlik/benzemezlik ölçümünün tespiti
6) Kümeleme kriterinin seçimi
7) Algoritma ve uygulama yönteminin seçimi
8) Küme sayısının belirlenmesi
9) Sonuçların yorumlanması

Kümeleme analizinde kullanılan verinin oluşturulma amacı analiz yapmak olmadığı için bu veri çok büyük boyutta, eksik ve/veya hatalı olabilir. Sağlıklı sonuçlar elde edebilmek için bu ve buna benzer veri ile ilgili problemlerin kümeleme analizi öncesinde mutlaka çözülmesi gerekmektedir. Famili ve ark. (1997) zeki veri analizinde veri önişleme problemleri ve çözüm yöntemlerini göstermektedir.

Veri kümeleme analizinde ele alınan problemlerin amacı, içeriği ve boyutu değişkenlik gösterdiğinden her probleme uygulanacak genel geçerli bir yöntem bulunmamaktadır. Yaygın olarak kullanılan kümeleme yöntemleri birimler arasındaki uzaklıklara dayanan benzerlik ya da benzemezlik matrisine göre işlem yaptıklarından, farklı kümeleme yöntemleri farklı uzaklık ölçülerine göre farklı sonuçlar verebilmektedir [Fraley ve Raftery, 1998]. Jain ve ark. (2004), amaç fonksiyonlarına göre kümeleme analizi yöntemlerini inceleyen bir çalışma yapmışlardır.

Kümeleme analizinin kullanım alanlarından bazıları;

- Örüntü tanıma,
- Konuşma tanıma,

- Parmak izi tanıma,
- Coğrafya: Konumsal veriden yararlanılarak bölgeler arasındaki benzerliklere göre bölgeleri gruplandırma,
- Biyoloji: Genlerin analizi ve genleri işlevlerine göre gruplandırma,
- Ekonomi: Müşterileri gruplandırma ve pazar araştırması,
- www: Belge sınıflandırma,
- Diğer veri madenciliği algoritmaları için bir önişleme,
- Pazarlama: Müşterileri sınıflandırarak, bu bilgi pazar büyültme,
- Şehir planlaması: Konumlarına, değerlerine ve türlerine göre binaları gruplandırma,
- Deprem araştırmaları: Sürekli fay hatları belirlenerek deprem merkezlerinin gruplandırılması

2.2. Kümeleme Analizi ve Diğer Veri İşleme Yaklaşımları Arasındaki İlişki

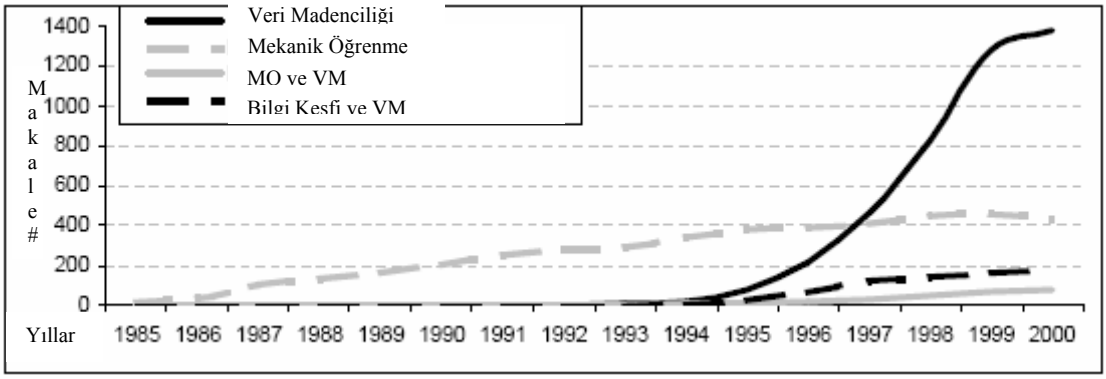
Veri kümeleme analizi kendi başına bir disiplin olmaktan çok istatistik, veri madenciliği, vektör nicemleme, makine öğrenmesi gibi diğer disiplinler içinde veya onlarla birlikte kullanılan bir tekniktir [Jain ve Dubes, 1988 - Zhou, 2003].

Veri kümeleme analizi, veri madenciliğinin en önemli yöntemlerinden birisidir. Veri madenciliğinin yeni gelişen bir disiplin olması nedeniyle veri madenciliği literatüründeki yerini 1990'lı yıllarda almaya başlamıştır. Bunun yanında vektör nicemleme alanında da veri kümeleme analizi önemli bir yere sahiptir. Gray (1998) nicemleme probleminin skalar nicemleme olarak 1948 yılında tanımlandığını, vektörel nicemlemenin ise 1957 yılından itibaren literatüre tanıtıldığını belirtmektedir.

2.2.1. Veri madenciliği ve kümeleme analizi

Veri kümeleme analizi kavram olarak daha eski bir kavram olmakla birlikte veri madenciliği kavramının ortaya çıkmasıyla birlikte onun en temel yöntemlerinden biri

haline gelmiştir. Veri madenciliği kavramı ilk olarak 1989 yılındaki IJCAI veritabanlarında bilgi keşfi atölye çalışmasında ortaya atılmış ve daha sonra kabul görerek literatürdeki yerini almıştır [Frawley ve ark., 1991]. Cios ve Kurgan (2002), veri madenciliği ve ilişkide bulunduğu diğer disiplinlerle ilgili literatürdeki gelişmeyle ilgili bir çalışma yapmışlardır. Şekil 2.2'de bu çalışmanın sonucunu gösteren bir grafik görülmektedir.

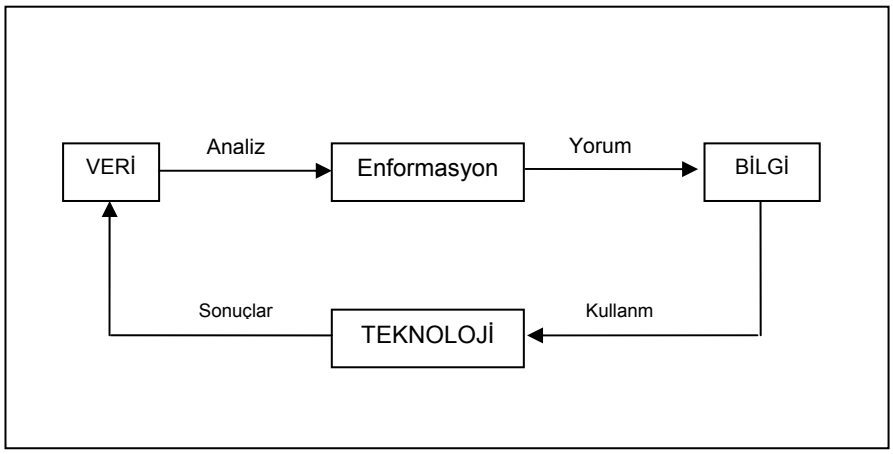


Şekil 2.2. Veri madenciliği ile ilgili literatürdeki gelişmeler [Cios ve Kurgan, 2002]

Han ve Kamber (1998), veri madenciliğini bilgi teknolojisindeki doğal evrimin bir sonucu olarak nitelendirmişlerdir. Bilgi teknolojilerinde gerek veri depolama ve transferi ile ilgili gelişmeler gerekse donanım (işlemci, RAM vs.) alanındaki süratli gelişmeler hayatın her alanına bilgi otomasyon sistemlerini sokmuş ve çok büyük miktarlarda depolanan veriden yararlanabilmek için yeni mekanizmalar üretilerek bu verinin işlenmesi gereksinimini ortaya çıkarmıştır. Çok büyük hacimlerdeki bu verinin otomasyon sistemleri olmadan işlenmesi teknik olarak mümkün değildir [Matheus ve ark., 1993].

Veri madenciliğinde asıl amaç ham veriden bilgiye ulaşmak ve bunları yorumlayarak yönetimde karar destek süreçlerinde kullanmaktır. Veri madenciliği, ham maddesi işlenmemiş veri, son ürünü ise bilgi olan tamamen dinamik bir sistemdir. Sistem veriden bilgiye ulaşan süreçte hem bilimsel hem de teknolojik gelişmelerden yararlanmaktadır. Newman (1997), veri-enformasyon-bilgi-teknoloji modeli adını

verdiği çerçevede veriden bilgiye giden süreci anlatmaktadır. Şekil 2.3'de VEBT modeli gösterilmektedir.



Şekil 2.3.Veri-Enformasyon-Bilgi-Teknoloji (VEBT) modeli [Newman,1997]

Veri madenciliği, veritabanlarında bulunan milyonlarca veri satırı içinde saklı bulunan genel, geçerli, tutarlı ve daha önceden farkında olunmayan yapı ve kuralların çıkartılması sürecidir [Fayyad ve ark.1996]. Veri tabanı teknolojisi, istatistik, mekanik öğrenme, örüntü tanımı, yapay sinir ağları, verilerin görselleştirilmesi ve uzaysal veri analizi gibi farklı disiplinlerde yer alan tekniklerin bir birleşimini içerir [Deogun ve ark., 1997]. Bu disiplinlerin aralarındaki kesin sınırları tanımlamak zor olduğu gibi, bu alanlar ile veri madenciliği arasındaki kesin sınırları tanımlamak da zordur [Hand ve ark. 2001]. Bunların yanında veri madenciliği problemleri matematiksel temellere dayanan eniyileme problemleri olarak da değerlendirilmektedir [Bradley ve ark. 1999].

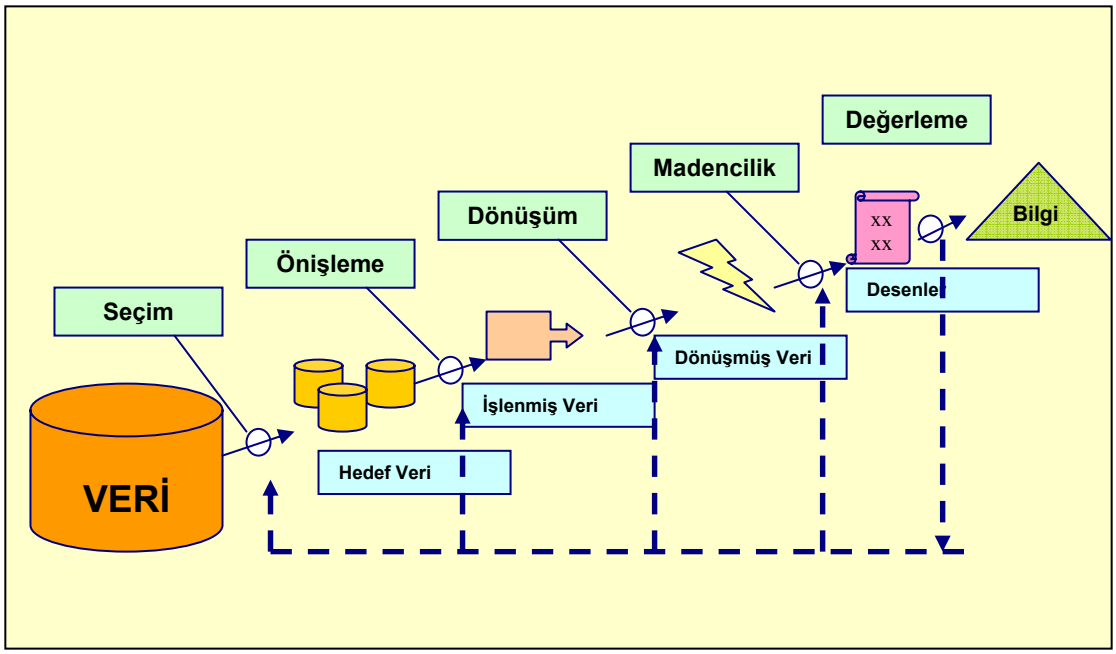
Literatürde veri madenciliği ile ilgili yazılmış olan kitaplarda da yazarların bakış açısı ve altyapılarına göre sözü edilen disiplerin bir veya birkaçı ağırlıklı olarak etkin olmuştur. Witten ve Frank (2005), veri madenciliği problemine makine öğrenmesi bakış açısıyla yaklaşıp bununla ilgili çözümler getirirken, Han ve Kamber (1996, 2001) veritabanı, Hand ve ark. (2001), istatistiksel, Wang ve Fu (2005) ise işlemsel zeka (Sinir ağları, bulanık yöntemler, genetik modeller vb.) bakış açısıyla yaklaşmışlardır. Henüz göreceli olarak yeni bir disiplin olması nedeniyle veri madenciliği alanında hala açık bir çok problem alanı bulunmaktadır. Veri

madenciliği problemlerinin çözümü için genel bir çerçeve oluşturma, ölçeklenebilir algoritmalar geliştirme, sonuçları görsel olarak sunabilme bu problem sahalarından sadece birkaçıdır [Mannila, 1997].

Veri madenciliği bir çok aşamadan oluşan, etkileşimli bir süreçtir [Brachman ve Anand, 1996]. Başka bir deyişle genel sistem tanımından yola çıkarak girdiyi (veri) çıktıya (bilgi) dönüştüren bir sistem olarak da adlandırılabilir [Newman, 1997]. Bütün sistemlerde olduğu gibi de bu dönüşümü sağlayan alt sistemler ve/veya süreçlerden oluşmaktadır [Frawley ve ark. 1991].

Veri madenciliğinde öncelikle çalışmanın amacının ne olduğuna karar verilmeli daha sonra bu amaca uygun olarak çalışmanın öznesinin ne olacağına karar verilmelidir. Özne olarak kastedilen istatistikteki gözleme karşılık gelmektedir. Seçilen özneye ait tüm veriler ortaya konulduktan sonra yapılacak olan analize uygun olan ve içinde en çok bilgiyi barındırdığı düşünülen veri alanları seçilmelidir [Mitra ve ark., 2002]. Burada bahsedilen veri alanları istatistik biliminde tesadüfi değişken, makine öğrenmesi biliminde ise öznitelik kavramı ile karşılanmaktadır. Bu çalışma içerisinde özne için veri nesnesi ve veri vektörü, öznitelikler içinse boyut, alan gibi kavramlar kullanılmıştır.

Çalışmanın nesnesi ve bu nesneyi tanımlayan özellikler belirlendikten sonra bunlara ait ön işleme ve dönüşüm işlemleri gerçekleştirilir. Çünkü veri her zaman saf haliyle kullanıma uygun olmayabilir. Son olarak ise uygulanacak madencilik yöntemi ve sonuçların değerlendirilmesi aşamaları gelmektedir. Şekil 2.4'de veri madenciliğinin aşamaları gösterilmektedir.



Şekil 2.4.Veri madenciliği sürecinde aşamalar [Fayyad, 1996]

Veri madenciliğinde yapılacak olan çalışmanın hedefine göre bir çok model mevcuttur. En genel anlamda bu modeller tanımlayıcı modeller ve tahmin edici modeller olarak ikiye ayrılmaktadır. Tahmin edici modeller, eldeki verilerden istifade ederek gelecekte oluşabilecek durumlar hakkında öngörüde bulunma işlevini üstlenirken, tanımlayıcı modeller ise yine eldeki verilerden yararlanmak suretiyle sistemin yapısı hakkında bilgi sahibi olmaya yaramaktadır [Larose, 2006]. Tahmin edici modeller daha çok istatistik ve makine öğrenmesi yöntemlerinin veri madenciliğine uyarlanması şeklinde de düşünülebilir [Chidand ve Shalom, 1997]. Tahmin edici veri madenciliği modellerine örnek olarak sınıflandırma, regresyon analizi, zaman serileri analizi tanımlayıcı modellere örnek olaraksa kümeleme analizi, birliktelik kurallarının keşfi, dizi keşfi verilebilir [Dunham, 2003].

Kümeleme analizi veri madenciliği modellerinde tanımlayıcı modeller arasında yer almaktadır. Simoudis (1996) ise kümeleme analizini keşfe dayalı veri madenciliği algoritmaları içerisinde saymaktadır. Kullandığı yöntemler ve içerik itibariyle sınıflandırma modeliyle büyük benzerlikler göstermektedir. Sınıflandırma modeli ise veri madenciliğinin tahmin edici yöntemlerindendir. Sınıflandırma ve kümeleme analizi arasındaki en temel fark sınıflandırmada sınıf etiketlerinin başlangıçta belli

olması kümelemede ise veriden otomatik olarak elde edilmesidir. Bu nedenle literatürde sınıflandırma problemi gözetimli öğrenme, kümeleme problemi ise gözetimsiz öğrenme olarak nitelenmektedir. Gözetimsiz öğrenmede, ilgili örneklerin gözlenmesi ve bu örneklerin özellikleri arasındaki benzerliklerden hareket ederek sınıfların tanımlanması amaçlanmaktadır [Akpınar, 2000]. Rubinov ve ark. (2006), sınıfları belli olan bir veri yığını üzerinde yapılacak kümeleme analizi sonucunda oluşan kümeler ve mevcut sınıflar arasındaki ilişkiyi inceleyen bir çalışma yapmışlardır.

2.2.2. Vektör nicemleme ve kümeleme analizi

Günümüz toplumunda bir çok farklı biçimleriyle faydalı ve değerli olan bilgi, büyük bir hızla artmakta bunun sonucu olarak da bilginin verimli olarak saklanabilmesi, ulaşılabilmesi ve iletilmesi önem kazanmaktadır. Bu durum özellikle, multimedya ortamlarında çok miktarda yer kaplayan sayısal sinyaller ve ses sinyalleri için geçerlidir. Sayısal sinyaller ve ses sinyallerinin iletilmesi çok büyük veri hızları, depolanması da çok büyük bellek boyutları gerektirmektedir. Bu nedenle, ses, resim ve video gibi sinyallerinin önemli bir kalite kaybı olmaksızın veri azaltma teknikleri kullanılarak sıkıştırılması konusu önem kazanmaktadır. Teknolojik olarak bellek iletim kanal kapasitelerinin sınırlı olması nedeniyle sinyallerinin düşük veri hızlarında iletilmesi ve depolanması istenmektedir [Vladimir ve Filip, 1997].

Nicemleme bir niceliği toplamları yine aynı niceliği verecek şekilde belli miktarda alt gruplara ayırmak olarak tanımlanmaktadır [Webster, 2005]. Lloyd (1957), ilk defa eniyi nicemlemenin gerek ve yeter şartlarını tanımlamıştır. Bu şartların üzerine tekrarlamalı azalan bir skalar nicemleme algoritması geliştirmiştir. Linde ve ark. (1980), Lloyd'un algoritmasını vektör nicemleme tasarımı olarak genişletmişler ve önerdikleri yöntem literatüre LBG yöntemi (genelleştirilmiş Lloyd algoritması - GLA- olarak da bilinir) olarak geçmiştir.

Veri sıkıştırmada, yüksek sıkıştırma oranından dolayı önemli bir yeri olan vektör nicemleme yöntemi, vektörleri eniyi şekilde temsil edecek kod kitabını oluşturmak

ve bu vektörleri kendisine en yakın kod vektörünün temsil ettiği bölgeye kümeleme işlemidir [Shen ve Hasegawa, 2006]. Bu işlemin kodlamadan kaynaklanan hata miktarını en aza indirmek için sağlanması gereken iki eniyileme şartı vardır. Bunlar; her vektörü kendisine uzaklık olarak en yakın olan kod vektör bölgesine yerleştirmek olan "en yakın komşuluk" şartı ile, kod vektörünü temsil ettiği bölgenin merkezinde seçmek olan "merkezleme" şartıdır. Nicemleme sonunda elde edilen resim ile orijinal resim arasındaki bozulma ölçümü olan TSGO (tepe sinyal- gürültü oranı) değerleri hesaplanmaktadır.

Nicemleme, teorik olarak kodlayıcı (sembol üretici) ve kod çözücü (sembolü sinyale dönüştürücü) gibi iki fonksiyonun birleştirilmesi ile olur. Kodlayıcıda, giriş vektörü bir indeks ile "gösterilir". Hangi indeksin seçileceği ise giriş vektörü ve kodlayıcı içindeki her kod vektörü arasındaki mesafe hesaplanarak en küçük mesafeye sahip kod vektörü seçilerek elde edilir. Kod çözücüde ise, bu indeksin "gösterdiği" sinyal bloğu (vektör) çıktı olarak oluşturulur.

2.2.3. İstatistik ve kümeleme analizi

Bir önceki bölümde veri madenciliği ve istatistik ilişkisi açıklanmıştı. Smyth (2001), birliktelik kuralları analizi dışında tüm veri madenciliği modellerinin temelinde istatistik prensiplerinin ve aksiyomlarının olduğunu belirtmektedir. Elder ve Pregibon (1996), modern istatistiksel yöntemler olarak nitelendirdiği veri analiz yöntemleri ve bunların veri madenciliğindeki uygulamalarından bahsetmektedir.

İstatistik ve veri madenciliği birbirleri ile neredeyse içiçe geçmiş iki farklı disiplindir. Kümeleme analizi açısından ele alınacak olursa, kümeleme analizi özde istatistiğin bir çok varsayımını ve yaklaşımını kullanmakla birlikte esas anlamını veri madenciliğinde bulmaktadır. İstatistiksel bakış açısıyla ele alındığında ise benzer amaçla kullanılan diskriminant analizi ve beklenti maksimizasyonu yöntemleri kümeleme analizine en yakın istatistiki yöntemler olarak görülmektedir. Diskriminant analizi, genel anlamda veriyi ayırma olup, bireylere ait p tane özellikten yararlanarak ait oldukları grupları (populasyon) belirlemede veya mevcut

grupları birbirinden ayıracak eniyi fonksiyonu bulmada kullanılan çok değişkenli istatistik tekniklerinden birisidir.

2.3. Kümeleme Analizinde Karşılaşılan Veri Tipleri ve Benzerlik Ölçüleri

Kümeleme analizinde çalışmanın konusu yani nesneleri insanlar, genler, malzemeler, ürünler gibi farklı özellikleri üzerinde barındıran bir çok şey olabilir. Bu farklı özellikler kümeleme analizinin değişkenlerini oluşturmaktadır ve bunların gerek tipi gerekse aldıkları değerler kümeleme algoritmasının seçimini ve seçilen yöntemin performansını direk olarak etkilemektedir [Jain ve ark., 1999].

2.3.1. Veri tipleri

Veri tipleri genel olarak sayımla belirlenen kategorik veriler ve ölçümle belirlenen sayısal veriler olmak üzere ikiye ayrılır.

Sayımla belirlenen (kategorik) veriler

Nominal (sınıflayıcı,isimsel, kalitatif)

Nominal bir değişkende, ölçüm düzeyleri arasında bir sıralama ya da uzaklık – yakınlık gibi belirli bir mesafe yoktur. Nominal değişken değerleri, rakam olarak değil, ad olarak anlam taşırlar. Genellikle metin tipinde verilerdir. Örnek olarak bir veritabanı tablosunda bulunan ad soyad alanları verilebilir.

Ordinal (sıralayıcı)

Ordinal ölçüm, nominal ölçümün belirli bir biçimde veya belirli bir kritere göre sıralandırılmasıdır. Sıralandırma, iyiden kötüye doğru ya da kötüden iyiye doğru yapılabilir. Ordinal bir değişkende, ölçüm düzeyleri arasında bir sıralama vardır, ancak düzeyler arasındaki uzaklık belli değildir. Örnek olarak rütbe, sınıf bilgileri verilebilir.

Ölçümle belirlenen (sayısal) veriler

Bir değişkenin aldığı değerler, nominal ya da ordinal değişkenlerde olduğu gibi araştırmacı tarafından belirlenmiş kodlar değil de gerçek rakamlarsa, o değişkenin sayısal ölçüm skalasında ölçüldüğü söylenebilir. Sayısal ölçümle belirlenen değişkende, değişken düzeyleri arasında hem sıralama, hem de belirli bir uzaklık vardır. Sayısal değişken değerlerine, reel sayılara uygulanan her türlü matematik işlem uygulanabilir. Sayısal değişkenleri, *sınıflayarak ordinal değişkenlere dönüştürebiliriz*. Sayısal değişkenleri çok gerektirmedikçe, ordinal değişkenlere dönüştürmek uygun değildir. Bu nedenle çalışmalar sırasında verileri toplarken, sayısal değişkenleri sınıflandırmadan, gerçek değerleri ile kaydetmek, daha sonra gerekirse dönüşüm uygulamak yerinde olur.

2.3.2. Benzerlik ölçüleri

Kümeleme analizinde ikinci aşama ise benzerlik ölçütünün seçilmesidir. Bu noktada benzerlik ölçütü kavramına açıklık getirmek gerekmektedir. Benzerlik ölçütü birbirinden farklı veri çiftlerinin birbirine ne kadar benzer olduğunu tespit etmeye yarayan bir kavramdır. Bazı çalışmalarda benzemezlik, uzaklık ölçütü olarak da tanımlanmaktadır.

Özniteliklerin veri çeşitlerinin birbirinden farklı olabilmesi (sayısal, kategorik) benzerlik ölçütünün seçiminde dikkat edilmesini gerektirmektedir. Farklı ölçütler farklı veri tipleri üzerinde doğru ve etkin sonuçlar üretebilmektedir.

Benzerlik ölçüsünün sağlaması gereken bazı koşullar vardır. Eğer x ve y iki farklı veri vektörü ise x ve y'nin benzerliği $d(x,y)$ ile gösterilmek şartıyla benzerlik ölçütünün;

(i) $d(x,y) > 0 \quad \forall x \neq y\,, \qquad d(x,x) = 0\,,$ (2.1)

(ii) $d(x,y) = d(y,x)\,,$ (2.2)

(iii) $d(x,y) \le d(x,z) + d(z,y) \qquad \forall z$. (2.3)

özelliklerini sağlaması gerekmektedir.

Literatürde en kullanılan en genel benzerlik ölçüsü Minkowski uzaklığıdır. Minkowski uzaklığının formülü Eş.2.4'de görülmektedir.

$$d(x,y) = \left(\sum_{i=1}^{m} |x_i - y_i|^r \right)^{1/r} \tag{2.4}$$

Minkowski uzaklığı formülünde r=1 olduğunda Manhattan uzaklığı (L_1), r=2 olduğunda ise (L_2) öklit uzaklığı adı verilmektedir.

Öklit uzaklığı ve Öklit uzaklığının karesi formülleri ile standartlaştırılmış verilerle değil, işlenmemiş verilerle hesaplama yapılır. Öklit uzaklıkları kümeleme analizine sıradışı olabilecek yeni nesnelerin eklenmesinden etkilenmezler. Ancak boyutlar arasındaki ölçek farklılıkları öklit uzaklıklarını önemli ölçüde etkilemektedir. Öklit uzaklık formülü en yaygın olarak kullanılan uzaklık hesaplama formülüdür. Öklit ve Öklit uzaklığının karesinin formülleri aşağıda görülmektedir.

$$d(x,y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{2.5}$$

Eş. 2.5'de formülü verilen Öklit uzaklığı *r=2* olan Minkowski ölçüsünün özel bir durumudur.

Manhattan uzaklığı boyutlar arasındaki ortalama farka eşittir. Bu ölçüt kullanıldığında farkın karesi alınmadığı için sıradışılıkların etkisi azalır. Manhattan uzaklığının formülü aşağıda görülmektedir.

$$d(x,y) = \sum_{i=1}^{m} |x_i - y_i| \tag{2.6}$$

Eş. 2.6'da formülü verilen Manhattan uzaklığı *r=1* olan Minkowski ölçüsünün özel bir durumudur.

Minkowski uzaklığı ve bunun değişik varyasyonları olan Öklit ve Manhattan uzaklıkları veri yapısına karşı duyarlıdır. Şöyleki eğer veri vektöründeki boyutlardan birisi (*i*) 0-100 ölçeğinde diğeri (*j*) 0-10 ölçeğinde ise *i* boyutu *j* boyutuna baskın gelmektedir. Bu duyarlılığı engellemenin bir yolu veriyi standartlaştırmak yani normalize etmek diğer yolu ise duyarlı olmayan uzaklık ölçülerini kullanmaktır. Ölçeğe duyarlı olmayan uzaklık ölçüleri Canberra uzaklığı, Cosine uzaklığı ve Mahalonobis uzaklığıdır.

Canberra uzaklığının formülü aşağıdaki gibidir.

$$d(x,y)=\sum_{i=1}^{m}\left|\frac{x_i-y_i}{x_i+y_i}\right| \tag{2.7}$$

Cosine uzaklığının formülü ise aşağıdaki gibidir.

$$<z_u,z_w>=\frac{\sum_{j=1}^{N_d} z_{u,j}z_{w,j}}{\|z_u\|\|z_w\|} \tag{2.8}$$

Cosine uzaklığı aslında bir uzaklık ölçüsünden çok bir benzerlik ölçüsüdür. İki vektör arasındaki farkı açısal olarak ölçmektedir. Yüksek boyutlardaki veriyi kümelemek için uygun bir ölçüdür.

Mahalonobis uzaklığının formülü ise aşağıdaki gibidir.

$d(x,y)=[\det V]^{1/m}(x-y)^T V^{-1}(x-y)$; V , $A_1....A_m$ in kovaryans matrisidir. (2.9)

Mahalanobis uzaklığı özniteliklerin varyanslarına ve karşılıklı (ikili) korelasyonlarına göre farklı özniteliklere farklı ağırlıklar verir. Diğer bir deyişle bu ölçü kümelerin elemanlarının çok değişkenli gauss dağılımdan geldiğini varsayar.

Literatürde kullanılan diğer uzaklık ölçülerinden başka bir tanesi ise Çebişev uzaklığıdır. Çebişev uzaklığı iki nesne arasındaki mutlak maksimum uzaklığa eşittir. Çebişev uzaklığının formülü aşağıda görülmektedir.

$$d(x,y) = \max_{i=1}^{m} |x_i - y_i| \tag{2.10}$$

Merkez tabanlı veri kümeleme analizinde olduğu gibi belli bir performans kriterini (amaç fonksiyonu) eniyilenerek veri içinde saklı bulunan kümelerin araştırılması problemi seçilen uzaklık fonksiyonu ile doğrudan ilişkilidir. Farklı uzaklık fonksiyonlarının seçilmesi, Öklit uzaklığının küresel kümelerin oluşmasına neden olması gibi farklı geometrik şekillerde kümelerin oluşmasına neden olur. Özellikle bulanık kümeleme alanında Bezdek (1981) tarafından geliştirilen bulanık c doğru, bulanık c elipsoid ve bulanık c değişken algoritmaları farklı geometrik şekillerde kümeleme problemine çözüm olarak önerilmektedir.

2.3.3. Veri standartlaştırma

Veri kümeleme analizinde kullanılacak olan verilerin her biri aynı birim ve/veya ölçekte tanımlanmadığı için tutarlı sonuçlar üretebilmek için kümeleme algoritması çalıştırılmadan verinin standart hale getirilmesi gerekmektedir. Bu standartlaştırma genelde her bir boyuta ait değerin 0-1 aralığında bir değer alması ile neticelenir. En çok kullanılan veri standartlaştırma formülleri aşağıda sıralanmıştır.

$$\dot{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2.11}$$

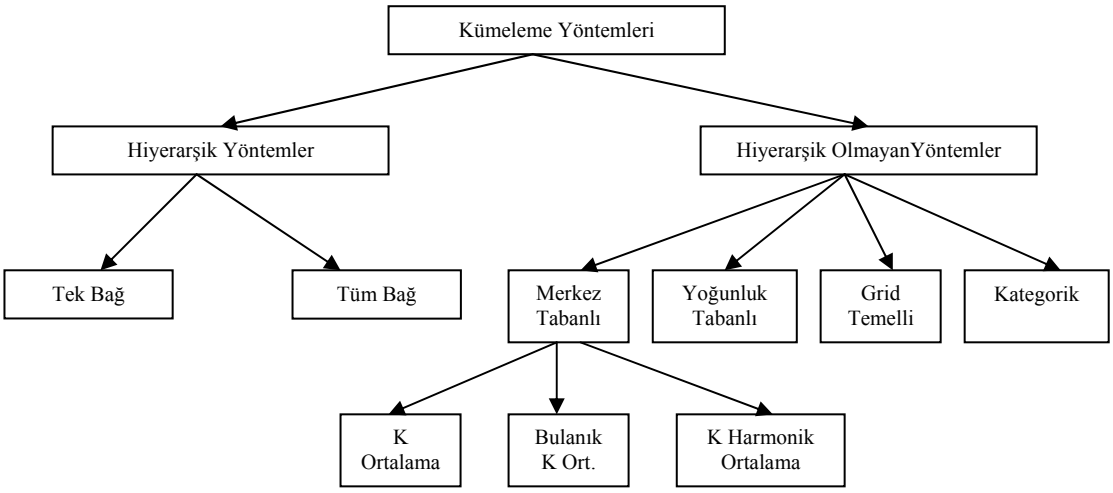
$$\dot{x} = \frac{x - \bar{x}}{s} \qquad s = \frac{1}{N}\sum_{i=1}^{N} |x_i - \bar{x}| \tag{2.12}$$

Bu çalışmada Eş.2.11'deki veri standartlaştırma formülü kullanılmaktadır.

Veri kümeleme analizinde, kullanılacak benzerlik ölçüsü seçildikten sonra yapılacak olan işlem uygun kümeleme yönteminin seçilmesidir. Üçüncü bölümde temel kümeleme analizi yaklaşımları hakkında bilgi verilecektir.

3. TEMEL KÜMELEME YAKLAŞIMLARI

Kümeleme analizi kavramı literatüre ilk olarak 1939 yılında girmiştir [Tryon, 1939]. 1940'lı yılların sonunda ve 1950'li yıllarda verilerin hiyerarşik olarak gruplanması üzerinde çalışılmış, 1960'lı yıllarda ise veriyi hiyerarşik olmayan alt gruplara bölerek kümeleme yöntemleri geliştirilmeye başlanmıştır. Bu çalışmaların bir kısmı istatistik bilimi kavramları ile yürütülürken büyük bir bölümü de veri madenciliği, makine öğrenmesi ve görüntü analizi disiplinlerinde gelişme göstermektedir. Kümeleme analizinde yapılacak olan çalışmanın amacına göre farklı kümeleme yöntemleri mevcuttur. Şekil 3.1'de kümeleme yöntemlerine ait bir sınıflandırma ağacı verilmektedir.

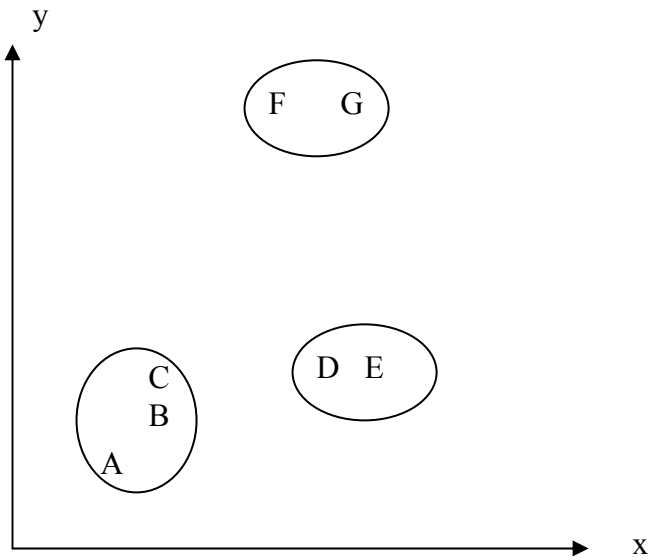


Şekil 3.1. Kümeleme yöntemlerinin sınıflandırılması

3.1. Hiyerarşik Kümeleme

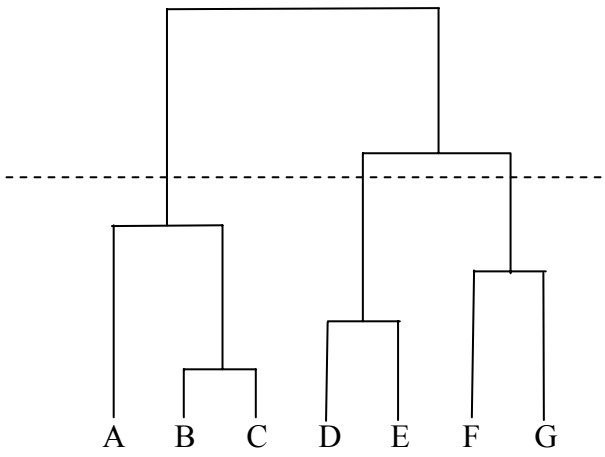
Hiyerarşik kümeleme yöntemlerinin tümü ilk adımda bütün kayıtları, tek kayıttan oluşan ayrı bir küme olarak değerlendirir. Daha sonra bu kümeler arasında uzaklıkları en az olan iki tanesini birleştirerek bir tane iki elemanlı küme haline getirir. Bunu izleyen her adımda, varolan kümeler arasında, birbirlerine uzaklıkları en küçük olan iki adet küme birleştirilerek tek bir küme haline getirilir. Bu birleştirme işlemleri, daha önceden belirlenen sayıda küme kalana kadar devam eder.

Aşağıda iki boyutlu özellik uzayında gerçeklenen hiyerarşik kümeleme algoritması özetlenmiştir. Şekil 3.2'de yer alan kayıtlar hiyerarşik kümeleme yöntemi ile gruplanmışlardır.



Şekil 3.2 Hiyerarşik kümeleme örnek

Hiyerarşik yöntemlerde her adımda hangi kayıtların bir araya getirilerek kümelendiğini gösteren DENDOGRAM (birleştirme ağacı) yapıları oluşturulur. Şekil 3.2'de yer alan kayıtlara ilişkin dendogram Şekil 3.3'de gösterilmektedir.



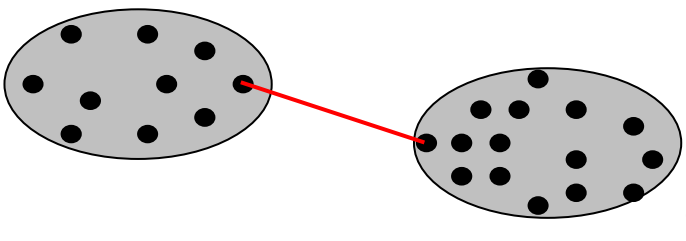
Şekil 3.3. Hiyerarşik kümelemede oluşan birleştirme ağacı

Şekil 3.3'deki birleştirme ağacından da görüleceği üzere ilk adımda B-C, ikinci adımda D-E, üçüncü adımda F-G, dördüncü adımda A-BC birleştirilmiştir. Sonraki adımda DE-FG ile ve en son adımda da ABC-DEFG birleştirilerek bütün kayıtları içeren en büyük küme oluşturulmuştur. Adını bu hiyerarşik birleştirme ağacından alan hiyerarşik yöntemlerin sonlanması, daha önceden belirlenen küme sayısına bağlıdır. Örneğin bu birleştirme ağacında 3 küme oluşacak şekilde bir kümeleme gerçeklenmiştir. Hiyerarşik yöntemlerin en güzel yanlarından bir tanesi de bir kez bu birleştirme ağacının oluşturulmasından sonra ağacın istenilen seviyelerden (istenilen sayıda küme sağlayacak şekilde) ayrılabilmesidir [Jain ve Dubes, 1988].

Çoğu hiyerarşik kümeleme yöntemi, tek-bağ [Sneath ve Sokal, 1973], tüm-bağ [King, 1967] ve en küçük-varyans [Ward, 1963] algoritmalarının değişik türevleridir. Bunlardan tek-bağ ve tüm-bağ algoritmaları en yaygın kullanılan olanlarıdır. Bu iki algoritma arasındaki fark, birleştirme işleminde iki küme için hesapladığı benzerlik (uzaklık) çıkarma yönteminden ileri gelmektedir.

3.1.1. Tek bağ kümeleme

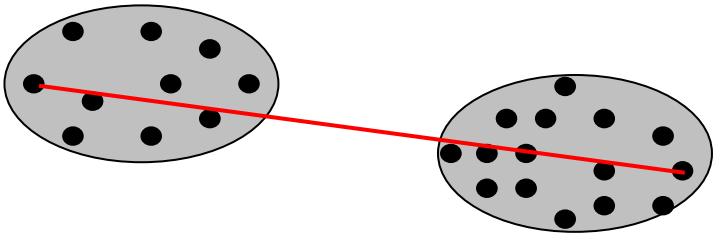
Tek-bağ algoritmasında iki küme arasındaki uzaklık, her iki küme arasında yer alan kayıtlardan birbirlerine en *yakın* olanların uzaklığı olarak değerlendirilmektedir. Bu işlem Şekil 3.4'te gösterilmiştir:



Şekil 3.4. Tek-Bağ kümelemede iki küme arası uzaklık

3.1.2. Tüm bağ kümeleme

Tüm-bağ algoritmasında iki küme arasındaki uzaklık hesap edilirken, bu iki küme içerisinde birbirlerine en *uzak* iki kaydın arasındaki uzaklık, bu iki küme arasındaki uzaklık olarak alınır (Bkz.Şekil 3.5).



Şekil 3.5. Tüm-Bağ kümelemede iki küme arası uzaklık

$$d_{kl} = \max d(\bar{x}_i, \bar{x}_j), \qquad \bar{x}_l = \frac{1}{|C_l|}\sum_{i \in C_l} x_i \tag{3.1}$$

Her iki yöntem sonunda da birbirlerine en yakın (en benzer) iki küme birleştirilerek daha büyük tek bir küme haline getirilir. Tek-bağ algoritması, tüm-bağ algoritmasına oranla daha kullanışlı bir yöntemdir. Örneğin bu yöntem içiçe geçmiş kümeleri bulabilirken, tüm-bağ yöntemi bu kümeleri bulamamaktadır. Ancak her ne kadar teorik olarak tek-bağ yöntemi daha kullanışlı gözükse de bir çok uygulamada tüm-bağ algoritmasının daha işlevsel olduğu ve ürettiği birleştirme ağaçlarının kullanılabilirlik oranının daha yüksek olduğu belirlenmiştir [Jain ve Dubes, 1988].

Bunların dışında, iki küme arası benzerliğin hesaplanması için iki yöntem daha vardır. Bunlardan ilki, iki küme içindeki bütün noktaların birbirleri ile arasındaki uzaklıkların ortalamasını, iki küme arası uzaklık olarak almaktır. Bu uzaklığın formülü Eş.3.2'de verilmiştir:

$$d_{kl} = \frac{1}{|C_l||C_k|}\sum_{i \in C_l, j \in C_k} d(x_i, x_j) \tag{3.2}$$

Son uzaklık formülü ise, iki küme arasındaki benzerliği bulmadan önce bu iki küme içerisindeki verilerden bir ortalama alarak küme merkezlerini bulmaktadır. Daha sonra bu küme merkezleri arasındaki uzaklık, iki küme arası uzaklık olarak kullanılır.

3.2. Hiyerarşik Olmayan Kümeleme

Hiyerarşik olmayan kümeleme yöntemleri, hiyerarşik kümeleme yöntemlerinde kullanılan birleştirme ağacı (dendogram) kullanmak yerine tüm kayıtların belirli bir şekilde bölünmesi ile kümeler oluşturur. Kümelenmesi gereken çok sayıda kaydın bulunduğu uygulamalarda, birleştirme ağacının oluşturulması gibi çok ağır bir iş yükü getiren bir algoritma yerine hiyerarşik olmayan yöntemler daha verimli olmaktadır. Ancak hiyerarşik olmayan yöntemlerin zayıf tarafları da vardır. Bunlardan en önemlisi, kümeleme işlemi sonucunda oluşması istenilen küme adedinin işlemden önce algoritma tarafından bilinmesinin gerekliliğidir.

Hiyerarşik olmayan kümeleme yöntemleri kümeleri oluştururken çoğunlukla yerel ya da tümel olarak tanımlanmış olan bir benzerlik kriterini optimize etmeye çalışır. Bu fonksiyonun muhtemel eniyi değerini bulmak için tüm kayıtlar üzerinde çeşitli kombinasyonlar denemek mümkün değildir. Bunun yerine pratik alanda birkaç başlangıç durumu için algoritma işletilir ve sonuç kümeleri de bu önceki çalışmaların sonuçlarından elde edilerek oluşturulur.

Hiyerarşik olmayan kümeleme yaklaşımında üzerinde en çok çalışılan merkez tabanlı kümeleme yaklaşımıdır. Merkez tabanlı kümeleme yaklaşımının başlangıcı 1960'lı yılların sonuna dayanmaktadır. Literatürde bu yaklaşıma prototip temelli kümeleme, amaç fonksiyonu temelli kümeleme adları da verilmektedir. Verilerin farklı kümeler arasında yinelemeli olarak yerleştirilmesi esasına dayanır. Zhang ve ark. (1999), bu tip kümelemeyi K tane merkezi bulabilmek için $J(X,C)$ amaç fonksiyonunun en iyilemesi problemi olarak tanımlamaktadır. Merkez tabanlı kümelemede en temel kümeleme algoritması K- ortalama kümeleme algoritmasıdır. Bulanık K-Ortalama ve K-Harmonik Ortalama algoritmaları ise K- Ortalama'dan türetilmiş esnek kümeleme algoritmalarıdır.

3.2.1. K-Ortalama kümeleme

Bölünmeli kümeleme yöntemlerinde çözümün kalitesi seçilen kümeleme kriterinin ölçülmesi ile belirlenir. Bu tür kümelemede yakınsama sağlanana kadar her bir iterasyonda kümeleme kriteri azaltılacak şekilde yeniden yerleştirme yapılmaktadır. En yaygın ve sıklıkla kullanılan kümeleme kriteri fonksiyonu, hatanın karesi kriteridir.

Hatanın karesini azaltma kriteri ilkesince çalışan en basit ve yaygın kullanılan algoritma K-Ortalama algoritmasıdır [Forgy, 1965; MacQueen, 1967]. Bu algoritmada öncelikle rasgele olarak başlangıç noktaları seçilir ve her adımda, kümelenmek istenilen örüntü kümesinin elemanları, bu başlangıç noktalarına olan uzaklıklarının değerine göre en yakın olduğu kümeye atanırlar. Her adım sonunda da belirli bir küme içerisine giren kayıtların tüm özelliklerinin ortalaması alınarak bu kümenin yeni merkezi hesap edilir. Bu işlemler, belirli bir durma kriterine yakınsayıncaya dek devam eder. Durma kriteri olarak belirli bir iterasyonda hiç bir küme elemanının başka kümeye atanmaması ya da hesaplanan uzaklıkların tümünün önceden belirlenen belirli bir uzaklığın altında olması verilebilir. Şekil 3-6 da merkez tabanlı kümelemenin algoritması akış şeması verilmektedir. K-Ortalama algoritmasının sık kullanılmasının nedeni, uygulanmasındaki basitlik ve zaman karmaşıklığının $O(n)$ olmasıdır.

K-Ortalama algoritmasının adımları aşağıda verilmektedir.

1) K küme merkezini K adet rasgele seçilen veriye ya da veri topluluğunu içeren uzayda k adet rasgele tanımlanan noktaya uygun şekilde seç.

2) Her bir veriyi en yakın kümeye ata.

3) Mevcut küme üyelik değerlerini kullanarak yeniden küme merkezlerini hesapla.

4) Eğer bir yakınsama koşulu karşılanmamışsa 2. Adıma geri dön. Tipik bir yakınsama ölçütü: toplam hatanın karesinin en küçük olması veya verilerin yeni

kümelere minimal düzeyde atanması veya hiç atanmaması. (Ölçüt fonksiyonunun eniyi değeri)

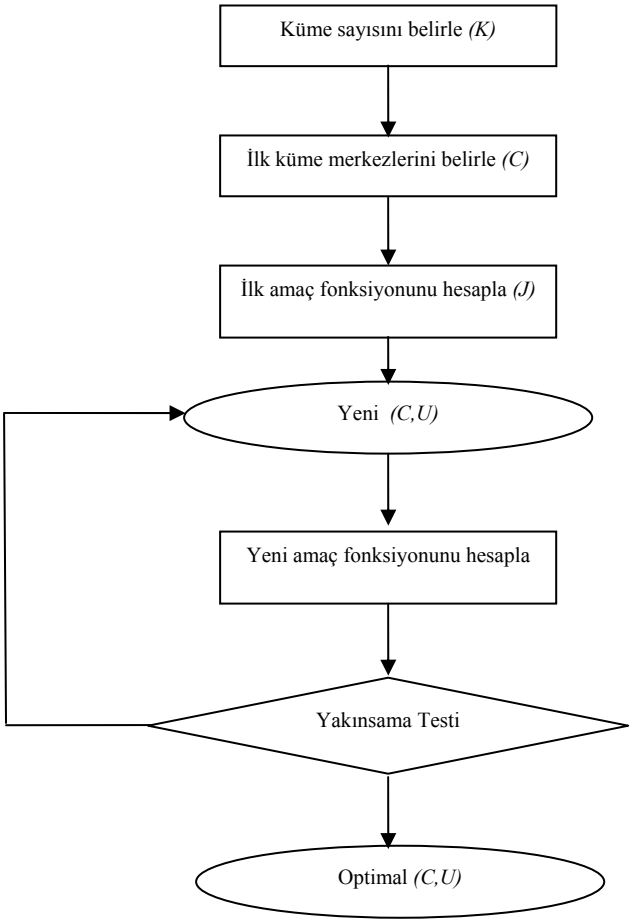
5) Küme sayısını mevcut kümeleri birleştirerek, parçalayarak, küçük veya sınır dışındaki kümeleri silerek küme sayısını düzelt.

Hatanın karesi kriteri, izole ve kompakt kümelerin bulunduğu veri kümeleri üzerinde eniyi sonucu vermektedir [Kaufman and Rousseeuw, 1990]. Bu fonksiyona göre *K* adet küme içinde toplam *N* tane kayıt içeren bir örüntü kümesi üzerinde *J* kümelemesindeki hatanın karesi şu şekilde hesaplanır:

$$\mathrm{J(X,C)} = \sum_{i=1}^{N}\sum_{j=1}^{K}\left\|x_i - c_j\right\|^2 \tag{3.3}$$

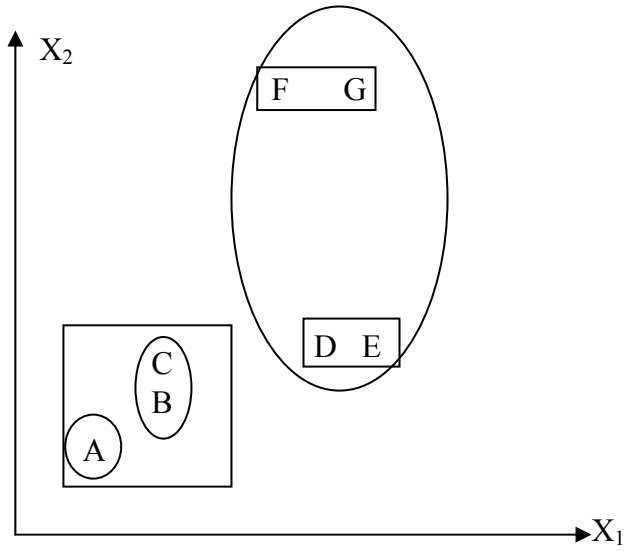
Burada x_i, *j.* küme içerisinde yer alan *i.* kaydı, c_j, ise *j*. kümenin merkez noktasını ifade etmektedir. Şekil 3.7'de K-Ortalama kümelemenin basit akış şeması gösterilmektedir.

K-Ortalamave beklenti maksimizasyonu algoritmaları birbirlerine çok benzer olarak gösterilselerde [He ve ark., 2004], aralarındaki en önemli fark K-Ortalama tekniği dağınıklığı (küme içi varyansı) minimize etmeye çalışırken, beklenti maksimizasyonu yöntemi, olabilirliği maksimize etmeye çalışmaktadır [Kearns ve ark., 1997].



Şekil 3.6. K-Ortalama kümeleme akış şeması

K-Ortalama algoritmasının en önemli zayıf noktası, başlangıçta rasgele olarak seçilen küme merkezlerinin seçimine çok duyarlı olmasıdır. Bu ilk değerlerin uygunsuz olarak seçilmesi durumunda algoritma, toplam hatanın karesi kriterinin yerel minimumunu bulmaktadır. Örnek vermek gerekirse Şekil 3.8'de gösterilen iki boyutlu olarak verilen analizde başlangıç noktaları olarak A, B, ve C seçilirse K-Ortalama algoritmasının ürettiği kümeler elips şekilleri ile gösterilmiştir. Bu durumda üretilen kümeler {A}, {B,C} ve {D, E, F,G} olacak şekilde üretilmiştir.



Şekil 3.7. K-Ortalama kümelemede başlangıç noktası seçimi [Tantuğ, 2002]

Oysa bu verilerin yer aldığı bir nesne kümesinin eniyi bölümlenmesi {A,B,C}, {D,E} ve {F,G} biçimindedir. Bu bölümlenme Şekil 3.8'de dikdörtgenler ile gösterilmiştir ve göz ile de rahatlıkla yapılabilmektedir. Oysa ki ilk durumda üretilen bölümlenmede toplam hatanın karesi fonksiyonunun değeri, ikinci bölümlenmede elde edilen değerden daha büyüktür. Yani ilk kümeleme işlemi sonucunda bu fonksiyonunun yerel bir en küçük değeri elde edilmiştir ancak bu değer tümel en küçük değildir. Tümel en küçük değerini bulmak için seçilmesi gereken başlangıç noktaları A, D ve F olmaktadır. Görüldüğü gibi başlangıç kümelerinin merkez noktalarının seçimi, algoritmanın eniyi kümelemeyi sağlayacak şekilde toplam hatanın karesi fonksiyonunu minimize etmesi üzerinde doğrudan etkilidir.

Merkez tabanlı kümeleme ve vektör nicemleme arasındaki ilişki ikinci bölümde anlatılmıştı. Burada özellkle K-Ortalama kümeleme ve vektör nicemleme benzerliğini anlatmak için tekrar vektör nicemleme kısaca anlatılacaktır.

Vektör nicemleyici tasarımı, sinyal gruplarından oluşan vektörleri eniyi şekilde temsil edecek kod kitabını oluşturma işlemidir. Gray (1998), vektörleri kendisine en yakın kod vektörünün temsil ettiği bölgeye kümeleme işlemini vektör nicemleme (VN) olarak tanımlamaktadır. VN sadece sıkıştırmada değil, aynı zamanda verinin

kümelenmesinde ve dizinlenmesinde de faydalı bir yöntemdir. Bu şekilde veri sınıflandırması ve örüntü tanıma yapılmaktadır. Vektör nicemlemenin eniyi olabilmesi iki koşula bağlıdır:

1. Her vektörü kendisine mesafe olarak en yakın olan kod vektör bölgesine yerleştirmek olan "en yakın komşuluk" şartı,
2. Kod vektörünü temsil ettiği bölgenin merkezinde seçmek olan "merkezleme" şartı.

Görüldüğü gibi K-Ortalama kümeleme ve vektör nicemleme temel olarak aynı süreci izlemektedir.

Klasik K-Ortalama kümeleme modelinde herhangi bir veri satırı (vektör) bir kümeye ya aittir ya değildir. Bu kümeleme modelinin içerdiği stratejiye "kazanan hepsini alır„ denmesinin nedeni de buradan kaynaklanmaktadır. Halbuki gerçek hayatın bu denli kesinlikten uzak olduğunu göz önüne aldığımızda bu derece kesin değerlendirmeler yapmanın sakıncaları görülmektedir. Bulanık K-Ortalama kümeleme kavramı klasik k ortalamanın sözü edilen sakıncasından hareketle ortaya çıkmıştır.

3.2.2. Bulanık K-Ortalama kümeleme

Bulanık kümeleme kavramına girmeden önce bulanık küme teorisi ve bulanık mantıktan özet olarak bahsetmekte fayda olacaktır. Bulanık mantığı oluşturacak temel düşünceyi Plato, "Doğru" ve "Yanlış'ın" iç içe girdiği üçünce bir durumu belirterek oluşturdu. Lotfi A. Zadeh (1965), bu değerleri [0,0, 1,0] aralığındaki sayılarla ifade ettiği teorisini "Bulanık Mantık" adlı çalışmasında tanımlayana dek, sonsuz-değerli mantık uygulamada başarılı olamamıştı.

Bulanık mantığın ardındaki temel fikir, bir önermenin 'doğru', 'yanlış', 'çok doğru', 'çok yanlış', 'çok çok doğru', 'çok çok yanlış', 'yaklaşık olarak doğru', 'yaklaşık olarak yanlış', v.b. gibi, olabileceğidir. Diğer bir deyişle, doğruluk, önermelerle,

klasik yanlış ve doğru arasındaki sonsuz sayıdaki doğruluk değerlerini içeren bir kümedeki değerler, ya da sayısal olarak [0, 1] gerçel sayı aralığıyla ilişkilendiren bir fonksiyondur.

Bulanık küme kuramı, belirsizliğin bir tür biçimlenişi, formüllendirilmesidir. Bir çeşit çok-değerli küme kuramıdır. Fakat işlemleri, diğer küme kuramlarının işlemlerinden farklılıklar gösterir. Kümedeki her birey, çift-değerli küme kuramlarında olduğu gibi 'üye' ya da 'üye değil' olarak değil, bir dereceye kadar üye olarak görülür.

'Bulanık küme' kavramı, hassasiyetin arttırılması açısından, klasik kümelerinkine göre daha uygun olan yeni bir araç sağlıyor olarak görülebilir. Getirdiği yaklaşım, klasik küme kuramlarında kullanılan üyelik kavramını bir kenara bırakıp yerine tamamen yenisini koymak değil, iki-değerli üyeliği çok-değerliliğe taşıyarak genellemesini yapmaktır. Bulanık kümeler doğal dildeki belirsiz ve bulanık kavramları temsil etmemize ve onları matematiksel olarak ifade etmemizi mümkün kılarlar.

Bulanık kümeleme yönteminde her bir veri, belli bir kümeye belli bir üyelik değeri ile dahil olur. Klasik K-Ortalama kümelemede üyelik foksiyonu değeri kesikli iken yani sadece 1 ve 0 değerlerini almakta iken bulanuk K-Ortalama kümelemede bu değer 0-1 aralığında süreklidir ve küme merkezi bizzat veri elemanının kendisi olmadığı sürece hiç bir zaman 1.0 değerini almamaktadır [Klawonn ve Höppner, 2003]. Bulanık kümelemede her bir küme bulanık olarak tanımlanmaktadır. Üyelik değerine belli bir eşik değeri uygulayarak bulanık kümelemeden keskin kümelemeye geçilir ya da Pham ve Prince [1999] tarafından belirtilen maksimum üyeliğe göre gruplama yaklaşımı kullanılabilir. En popüler bulanık kümeleme algoritması Bulanık K-Ortalama (BKO)' dır. BKO, Dunn (1973, 1974), tarafından önerilmiş ve Bezdek tarafından 1981'de BKO algoritmasının genelleştirilmiş hali ortaya konmuştur. Gustafson ve Kessel (1979), bulanık kümeleme yönteminde Öklit uzaklığı yerine Mahalanobis uzaklığı kullanmışlardır. Pedrycz (1998, 2005), veri yığını içerisindeki bilgi taneciklerinin bulanık kümeler yardımıyla tespit edilmesi kavramını ortaya

atmış ve anılan yayınlarda veri madenciliği ve bulanık kümeler arasındaki ilişkileri anlatmıştır.

Aşağıda temel bulanık kümeleme algoritması tanıtılmaktadır.

1) *N* nesneyi *K* kümeye *N x K* üyelik matrisi *U*'yu seçerek ata. ($\mu_{ij} \in [0,1]$)

2) *U*' yu kullanarak bulanık ölçüt fonksiyonunu bul. Örn. Belli bir kümeleme için ağırlıklı toplam hatanın karesi ölçütü. Bir olası bulanık ölçüt fonksiyonu:

$$J(X,U) = \sum_{i=1}^{N}\sum_{k=1}^{K} u_{ij} \left\| x_i - c_k \right\|^2$$

Buradaki $c_k = \sum_{i=1}^{n} u_{ik} x_i$, k. bulanık küme merkezidir. Bu ölçüt fonksiyonunu azaltmak için verileri yeniden ata ve *U*' yu yeniden hesapla.

$$c_i = \frac{\sum_{j=1}^{N} (u_{ij})^q x_j}{\sum_{j=1}^{N} (u_{ij})^q} \qquad u_{ij} = \frac{1}{\sum_{r=1}^{c} \left(\frac{d_{ij}}{d_{rj}} \right)^{\frac{2}{q-1}}} \qquad \text{i=1,.....,c,}$$

j=1,.....,N

3) 2.adımı *U* önemli ölçüde değişmeyinceye kadar tekrarla.

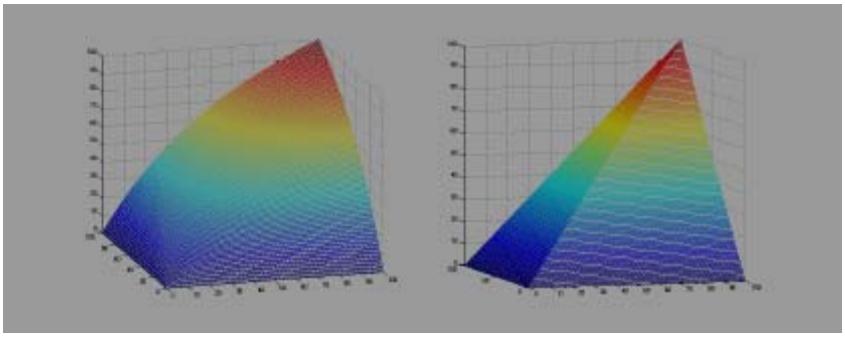
Temel bulanık küme algoritmasına $\sum_{j=1}^{K} u_{ij} = 1$ kısıtı ilave edildiğinde algoritma olasılıklı bulanık kümeleme algoritmasına dönüşmektedir [Masulli ve Rovetta, 2006). Olasılıklı denmesinin nedeni üyelik değerleri toplamının tıpkı olasılık kavramında olduğu gibi 1'e eşit olmasıdır.

3.2.3. K-Harmonik Ortalama kümeleme

K tane sayının harmonik ortalaması $\{a_1.....a_k\}$

$HA(\{a_1.....a_k\}) = \dfrac{K}{\sum_{k=1}^{K} \dfrac{1}{a_k}}$ şeklinde tanımlanmaktadır.

Eğer a_k değerlerinden bir tanesi küçükse harmonik ortalamanın değeri de küçük çıkmaktadır. Bu nedenle harmonik ortalama aritmetik ortalamadan daha çok en küçük fonksiyonuna benzemektedir. Bu da zaten kümeleme analizinde istenen bir durumdur. Şekil 3.10' da [0,10] aralığında değişken değerler alan *x* ve *y* ikilisinin en küçük ve harmonik ortalama davranışları gösterilmektedir. Sol taraftaki grafik harmonik ortalamayı sağ taraftaki grafik ise en küçük fonksiyonunun davranışını göstermektedir.



(a) (b)

Şekil 3.8. Minimum fonksiyonu ve harmonik ortalama benzerliği
a)MIN Fonksiyonu b)Harmonik ortalama

Veri noktaları ile en yakın merkez arasındaki ilişki o kadar güçlüdür ki herhangi bir veri elemanının üyelik derecesi başka bir kümeye yaklaşana kadar değişiklik göstermemektedir. Diğer bir deyişle "kazanan hepsini alır" stratejisi gerçekleşmektedir. Başlangıca karşı duyarlılık ta zaten buradan gelmektedir. K-Ortalama kümeleme algoritmasına düzeltmeler veya yeni başlatma modelleri geliştirmek yerine en küçük belirleme olayına yeni bir bakış açısı getiren harmonik ortalama kavramı kullanılacaktır. K-Harmonik Ortalama başlangıca karşı duyarlılığı ortadan kaldırmaktadır.

K-Harmonik Ortalama kümelemenin getirdiği diğer bir özellik ise verilere ağırlık fonksiyonu vererek hiçbir kümeye yakın olmayan veri elemanına bir sonraki iterasyonda öteleme yapmasıdır.

C= *K* tane küme merkezinden oluşan küme olduğunda, kümeleme algoritmasının performans değeri (amaç fonksiyonu)

$$Perf(X,C) = \sum_{x \in X} d(x,C) \tag{3.4}$$

Yani bütün veri elemanlarının ait oldukları küme merkezlerinden uzaklıklarının toplamıdır.

$d(x,C)$ fonksiyonu ise:

K-Ortalama için; $MIN\left\{\|x-c\|^2 \,|\, c \in C\right\}$ (3.5)

K-Harmonik ortalama için; $HA\left\{\|x-c\|^2 \,|\, c \in C\right\} = \dfrac{|C|}{\sum_{c \in C} \dfrac{1}{\|x-c\|^2}}$ (3.6)

Beklenti Maksimizasyonu içinse; $-\log(\sum_{m \in M} p_m * G_m(x))^2$ (3.7)

K-Ortalama kümeleme için için performans fonksiyonu;

$$Perf_{KM}(\{x_i\}_{i=1}^N, \{c_l\}_{i=1}^K) = \sum_{l=1}^{K} \sum_{x \in S_l} \|x - c_l\|^2 \tag{3.8}$$

$S_l \subset X = \{x_i\}_{i=1}^N$, c_l 'ye $C = \{c_l\}_{l=1}^K$ deki diğer bütün merkezlerden daha yakın olan x'lerin alt kümesidir. Eş. 3.8'deki ikili toplama bütün *x* değerleri üzerinde tekli toplamaya dönüştürülebilir ve toplamalar altındaki kareli uzaklık *MIN()* fonksiyonu ile ifade edilebilir. K-Ortalama kümeleme performans fonksiyonunu buna göre yeniden yazıldığında;

$$Perf_{KM}(X,C)=\sum_{i=1}^{N} MIN\left\{\left\|x_i-c_l\right\|^2 \middle| l=1,.....,K\right\} \tag{3.9}$$

Eş. 3.9'daki $MIN()$ fonksiyonunu $HA()$ ile yer değiştirildiğinde;

$$Perf_{KHMp}(X,C)=\sum_{i=1}^{N} HA\left\{\left\|x_i-c_l\right\|^2 \middle| l=1,.....,K\right\}=\sum_{i=1}^{N}\frac{K}{\sum_{l=1}^{K}\frac{1}{\left\|x_i-c_l\right\|^2}} \tag{3.10}$$

Eş. 3.10'da dıştaki toplamanın içindeki değer K tane kareli uzaklığın- $\left\{\left\|x-c_l\right\|^2 \middle| l=1,.....,K\right\}$- harmonik ortalamasıdır.

KHM_p Performans fonksiyonu ve KHM_p Algoritması

Eş.3.10'un c_k ya göre kısmi türevi alınıp sıfıra eşitlendiğinde;

$$\frac{\partial Perf_{KHM}(X,C)}{\partial c_k}=-K\sum_{i=1}^{N}\frac{p(x_i-c_k)}{d_{i,k}^{p+2}(\sum_{l=1}^{K}\frac{1}{\left\|x_i-c_l\right\|^p})^2}=0$$ ve buradan da m_k ları

çekildiğinde;

$$c_k=\frac{\sum_{i=1}^{N}\frac{1}{d_{i,k}^{p+2}(\sum_{l=1}^{K}\frac{1}{d_{i,l}^{p}})^2}x_i}{\sum_{i=1}^{N}\frac{1}{d_{i,k}^{p+2}(\sum_{l=1}^{K}\frac{1}{d_{i,l}^{p}})^2}}$$ olur ve bu değer her bir iterasyonda hesaplanır. (3.11)

$$\alpha_i=\frac{1}{(\sum_{l=1}^{K}\frac{1}{d_{i,l}^{p}})^2}\ ;q_{i,k}=\frac{\alpha_i}{d_{i,k}^{p+2}}\ ;\ q_k=\sum_{i=1}^{N}q_{i,k}\ ;\ p_{i,k}=\frac{q_{i,k}}{q_k}\ ;\quad c_k=\sum_{i=1}^{N}p_{i,k}x_i \tag{3.12}$$

Bütün değerler eşit olduğunda aritmetik ortalama, harmonik ortalamaya eşittir,diğer durumlarda ise aritmetik ortalama, harmonik ortalamadan büyüktür.

K-Harmonik Ortalama amaç fonksiyonu: $$KHM(X,C)=\sum_{i=1}^{N}\frac{k}{\sum_{j=1}^{k}\frac{1}{\left\|x_i-c_j\right\|^{p}}} \quad (3.13)$$

p: girdi parametresi ve genellikle $p \geq 2$ olarak alınmalıdır. Zhang [2000] önerdiği yöntemde genellikle p değeri olarak 3.5 değerinin uygun çözümler verdiğini belirtmektedir.

$w(x_i)$: ağırlık fonksiyonu. Ağırlık fonksiyonunun o iterasyonda alacağı değer $w(x_i)>0$ olmak üzere x_i veri noktasının bir sonraki iterasyonda merkezlerin hesaplanmasını nasıl etkileyeceğini göstermektedir.

Aşağıda temel K-Harmonik Ortalama kümeleme algoritmasının aşamaları gösterilmektedir.

1) K küme merkezini K adet rasgele seçilen veriye ya da veri topluluğunu içeren uzayda k adet rasgele tanımlanan noktaya uygun şekilde seç.

2) Merkezlere göre küme üyelik değerlerini hesapla.

Üyelik fonksiyonu: $$m_{KHM}(c_j / x_i)=\frac{\left\|x_i-c_j\right\|^{-p-2}}{\sum_{j=1}^{K}\left\|x_i-c_j\right\|^{-p-2}}$$

3) Her bir veri elemanının ağırlığını hesapla.

Ağırlık fonksiyonu: $$w_{KHM}(x_i)=\frac{\sum_{j=1}^{K}\left\|x_i-c_j\right\|^{-p-2}}{(\sum_{j=1}^{K}\left\|x_i-c_j\right\|^{-p})^{2}}$$

4) Yeni hesaplanan üyelik değerleri ve ağırlık değerlerine göre yeni merkezleri hesapla ve bu merkezlere göre amaç fonksiyon değerini hesapla.

$$c_k = \frac{\sum_{i=1}^{N} \frac{1}{d_{i,k}^{p+2}(\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}})^2} x_i}{\sum_{i=1}^{N} \frac{1}{d_{i,k}^{p+2}(\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}})^2}}$$

5) Eğer bir yakınsama koşulu karşılanmamışsa 2. Adıma geri dön.

6) Yakınsama koşulu sağlanmış ise üyelik değerlerini durulaştırarak her bir elemanın hangi kümeye ait olduğunu belirle.

Çalışmanın buraya kadar olan bölümünde merkez tabanlı kümeleme yöntemleri hakkında genel bilgi ve bu yöntemlere ait algoritmalar verilmiştir. Çizelge 3.1' de üç farklı merkez tabanlı kümeleme yöntemine ait hesaplama yöntemleri detaylı olarak gösterilmektedir.

Çizelge 3.1. Kümeleme algoritmasında hesaplama yöntemleri

	KO	***BKO***	***KHO***
Üyelik değeri	$m_{KO}(c_j / x_i) = \begin{cases} 1; l = \arg\min(x_i - c \\ 0; otherwise \end{cases}$	$m_{BKO}(c_j / x_i) = \frac{1}{\sum_{k=1}^{K} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{2/(m-1)}}$	$m_{KHO}(c_j / x_i) = \frac{\|x_i - c_j\|^{-p-2}}{\sum_{j=1}^{K} \|x_i - c_j\|^{-p-2}}$
Merkez	$c_k = \frac{\sum_{i=1}^{N} (m_{KO}(c_k / x_i))^m x_i}{\sum_{i=1}^{N} (m_{KO}(c_k / x_i))^m}$	$c_k = \frac{\sum_{i=1}^{N} (m_{BKO}(c_k / x_i))^m x_i}{\sum_{i=1}^{N} (m_{BKO}(c_k / x_i))^m}$	$c_k = \frac{\sum_{i=1}^{N} \frac{1}{d_{i,k}^{p+2}(\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}})^2} x_i}{\sum_{i=1}^{N} \frac{1}{d_{i,k}^{p+2}(\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}})^2}}$
Perf (X, C)	$KO(X,C) = \sum_{j=1}^{K} \sum_{i=1}^{N} (m_{KM}(c_j / x_i))^$	$BKO(X,C) = \sum_{j=1}^{K} \sum_{i=1}^{N} (m_{BKO}(c_j / x_i))^m d$	$KHO(X,C) = \sum_{i=1}^{N} \frac{k}{\sum_{j=1}^{k} \frac{1}{\|x_i - c_j\|^$

3.2.4. Diğer kümeleme yöntemleri

Model tabanlı kümeleme yöntemleri eldeki veri yığınına eniyi uyacak matematiksel bir model bulmaya çalışan yöntemlerdir. Bu tür yöntemler genelde verinin belirli olasılık dağılımlarının karışımlarından türediğini varsayar. N nesneden oluşan $D = \{x_1, x_2, ..., x_N\}$ şeklindeki bir veri yığınındeki her x_i nesnesi Θ parametre kümesiyle tanımlanan bir olasılık dağılımından oluşturulur. Olasılık dağılımının $c_j \in C = \{c_1, c_2, ..., c_K\}$ şeklinde K adet bileşeni vardır. Her $\Theta_g, g \in [1, ..., G]$ parametre kümesi k bileşeninin olasılık dağılımını belirleyen Θ kümesinin ayrışık bir alt kümesidir. Herhangi bir x_i nesnesi öncelikle, $p(c_k \setminus \Theta) = \tau_k$, $\sum_K \tau_k = 1$ olacak şekilde bileşen katsayısına (ya da bileşenin seçilme olasılığına) göre bir bileşene atanır. Bu bileşen kendi parametrelerine göre $p(x_i \setminus c_k; \Theta)$ olasılık dağılımına göre x_i değişkenini oluşturur. Böylece x_i nesnesinin bu model için olasılığı bütün bileşenlerin olasılıklarının toplamıyla ifade edilebilir. Modelin parametre kümesi, bileşenlerin seçilme olasılığı ve her bileşenin parametre kümesinden oluşur.

Beklenti maksimizasyonu algoritmasında verinin K bileşenli bir modelden geldiği varsayılır. Her bileşen bir kümeye karşılık gelir. Kümeleme problemi hem hangi verinin hangi kümeye ait olduğunu bulmak hem de bu kümelere ilişkin model model parametrelerini bulmaktır. Başlangıçta model parametreleri için bir başlangıç değeri saptanır. Algoritma iki adımdan oluşur. İlk adımda model parametreleri sabit varsayılarak verilerin hangi kümeye ne kadar olasılıkla atandırıldıkları hesaplanır. İkinci adımında

$$p(\Theta_1, ..., \Theta_K; \tau_1, ..., \tau_K \setminus D) = \prod_{i=1}^{N} \sum_{k=1}^{K} \tau_k \, p(x_i \setminus c_k; \Theta_k) \tag{3.14}$$

değeri en büyük olacak şekilde model parametreleri güncellenir. Bu iki adım model parametrelerinde değişiklik olmayıncaya kadar ya da değişiklik tanımlanan bir aralıkta kaldığı sürece devam eder. Algoritmaya başlamak için küme sayısı ve model

başlangıç parametreleri gibi başlangıç koşullarının seçilmesi ve buna ek olarak verinin hangi dağılımdan geldiğinin belirtilmesi gerekmektedir. Burada en önemli adım verinin hangi dağılımdan geldiğini kestirmektir. Özellikle boyut sayısı arttıkça olasılık dağılımını bulmak oldukça güçtür. Bu güçlüğü aşmak için olasılık uygulamalarında kullanılan bir varsayımla her boyutun birbirinden bağımsız olduğu varsayılabilir. Böyle bir varsayım yapıldığında problem her boyut için bir olasılık dağılımı bulmaya indirgenmiş olur.

Hiyerarşik ve hiyerarşik olmayan veri kümeleme yaklaşımlarının dışında bazı özel problem yapıları veya özel veri yapılarına uygun olarak geliştirilmiş olan kümeleme yaklaşımları da bulunmaktadır.

Kaufman ve Rousseeuw [1990] medoidler etrafında bölünme (PAM) adını verdikleri bir yöntem önermişlerdir. Bu yöntemde uzaklık ölçüsünden bağımsız çalışan bir yöntemdir. Öncelikle *K* tane medoid (en merkezdeki eleman) küme temsilcisi olarak belirlenmekte daha sonra seçilen uzaklık ölçüsüne göre oluşan benzerlik/benzemezlik matrisi oluşturulmakta ve K-Ortalama kümelemedekine benzer şekilde süreç iteratif olarak durma koşulu sağlanana kadar devam ettirilmektedir. PAM yönteminin maliyeti yüksek bir yöntem olması nedeniyle bu yöntemden CLARA ve CLARANS adı verilen iki farklı kümeleme yöntemi daha türetilmiştir [Ng ve Han, 1994].

Yoğunluk temelli kümeleme

Yoğunluk temelli kümelemede kümeleme nesnelerin yoğunluğuna göre yapılır. Başlıca özellikleri;

- Rasgele şekillerde kümeler üretilebilir,
- Aykırı nesnelerden etkilenmez,
- Algoritmanın son bulması için yoğunluk parametresinin verilmesi gerekir.

Yoğunluk tabanlı yöntemlerden bazıları;

- DBSCAN (**D**ensity **B**ased **S**patial **C**lustering of **A**pplications with **N**oise) [Ester ve ark., 1996]
- OPTICS (**O**rdering **P**oints **T**o **I**dentify the **C**lustering **S**tructure) [Ankerst, ve ark., 1996]
- DENCLUE (**DEN**sity-based **CLU**st**E**ring) [Hinneburg ve Keim, 1998]
- CLIQUE (**C**lustering **I**n **QUE**st) [Agrawal ve ark., 1998] olarak sayılabilir.

Yoğunluk temelli kümeleme işleyişine bir örnek olarak DBSCAN algoritmasından özet olarak bahsetmek gerekirse bu algoritmanın 2 parametresi vardır. Birincisi en büyük komşuluk yarıçapı (*Eps*) ikincisi ise *Eps* yarıçaplı komşuluk bölgesinde bulunan en az nesne sayısı (*MinPts*) dir.

$$N_{eps}(p) : \{q \in D | d(p,q) \leq Eps\}$$

Yoğunluk temelli kümeleme algoritması aşağıda verimektedir.

1) Veritabanındaki her nesnenin *Eps* yarıçaplı komşuluk bölgesi araştırılır.
2) Bu bölgede *MinPts* den daha fazla nesne bulunan *p* nesnesi çekirdek nesne olacak şekilde kümeler oluşturulur.
3) Çekirdek nesnelerin doğrudan erişilebilir nesneleri bulunur.
4) Yoğunluk bağlantılı kümeler birleştirilir.
5) Hiç bir nesne bir kümeye eklenmezse işlem sona erer.

Doğrudan erişilebilir nesne: Eps veya *minPts* koşulları altında bir *q* nesnesinin doğrudan erişilebilir bir *p* nesnesi şu koşulları sağlar:

- $p \in N_{eps}(q)$
- Q nesnesinin çekirdek nesne koşulunu sağlaması: $N_{eps}(q) \geq MinPts$

Erişilebilir nesne: Eps ve MinPts koşulları altında *q* nesnesinin erişilebilir bir *p* nesnesi olması için ;

- $p_1, p_2, ..., p_n$ nesne zinciri olması,
- $p_1 = q, p_n = p$,
- p nesnesinin doğrudan erişilebilir nesnesi: p_{i+1}

Yoğunluk bağlantılı nesne: Eps veya minPts koşulları altında bir *q* nesnesinin yoğunluk bağlantılı nesnesi *p* şu koşulları sağlar;

- *p* ve *q* nesneleri *Eps* ve *MinPts* koşulları altında o nesnenin erişilebilir nesnesidir.

Grid temelli kümeleme

Grid temelli kümeleme algoritmalarının temel konusu uzaydaki nesnelerin geometrik yapısını, birbirleriyle olan ilişkilerini, özelliklerini ve işlemlerini modelleyen uzaysal veridir. Bu algoritmaların amacı veri yığınını miktarı belli olan hücrelere ayırmak ve daha sonra hangi nesne hangi hücreye aittir sorusunun cevabını bulmaktır [Halkidi ve ark, 2001]. Yapısal olarak hiyerarşik kümeleme yaklaşımına daha yakındır. Asıl farkı uzaklık ölçüsü yerine kullanıcı tarafından tanımlanan daha başka parametreler kullanmasıdır. *STING* (Statistical Information Grid-based method, [Wang ve ark, 1997] ve *WaveCluster* [Sheikholeslami ve ark., 1998] grid temelli kümeleme yaklaşımına verilebilecek iki örnektir.

Kohonen [1995] özörgütlemeli haritalar (SOM) adını verdiği bir yöntem önermektedir. Burada adı geçen öz örgütlemeli haritalar özellikle çok yüksek boyuttaki verileri iki veya üç boyut gibi çok daha küçük boyutta sinir hücrelerinin birleşiminden oluşan bir grid üzerinde göstererek diğer başka işlemlerin yanında sınıflandırma, kümeleme analizi gibi veri analiz yöntemlerinde de kullanılmasını sağlamaktadır.

Kategorik kümeleme

Kümeleme problemi özünde sadece sayısal veri üzerinde çalışmakla birlikte özellikle veritabanı kümelemede sayısal veri dışında kategorik veriler de kapsam dahilinde olduğundan bu tip veriler için de farklı yöntemlere ihtiyaç duyulmaktadır. Her ne

kadar farklı yöntemler denilse de bu yöntemler hem Hiyerarşik olmayan hem de hiyerarşik veri kümeleme için geliştirilen algoritmalardan yararlanmaktadır. Örneğin Guha ve ark., [2000] tarafından önerilen ROCK hiyerarşik kümeleme için geliştirilen yöntemlerden yararlanmakta iken, Huang [1997] tarafından önerilen K-mode algoritması Hiyerarşik olmayan kümeleme mantığını kullanmaktadır. Doring ve ark. [2004] ise hem numerik hem de kategorik veriyi ele alan bulanık bir kümeleme yöntemi önermişlerdir. Ng ve Wong [2002] kategorik verilerin kümelenmesi ile ilgili olarak bulanık k-mod problemini tabu arama sezgiseli ile entegre eden bir yöntem önermişlerdir.

3.3. Kümeleme Analizinde Değerlendirme Teknikleri

Merkez tabanlı kümeleme analizinin değerlendirme işlemi aşağıdaki üç soruya cevap vermek için yapılır.

1) Gerçekte bu veri yığını kaç farklı kümeye ayrılabilir?
2) Uygulanan yöntem veriyi doğru olarak kümelere ayırabiliyor mu?
3) Daha iyi bir kümeleme mümkün mü?

Genel olarak küme değerlendirme analizine 3 farklı yaklaşım vardır. Birincisi kullanıcı tarafından yönlendirilen harici kriter üzerine inşa edilmiştir. İkincisi veri tarafından yönlendirilen dahili kriter, üçüncüsü ise aynı algoritmanın farklı parametrelerle çalıştırılması sonucu oluşan göreceli kriter olarak adlandırılmaktadır. Harici değerlendirme yöntemleri istatistiksel hipotez testleri şeklinde yapılmaktadır ve hesaplama maliyetleri oldukça yüksektir. Bu nedenle genellikle iki ve üçüncü tip değerlendirme analizleri tercih edilmektedir.

Küme değerlendirmenin yapılması ve eniyi kümeleme şemasının belirlenmesi için 2 temel kriter vardır:

1) Sıkılık (Yoğunluk): Her bir kümenin elemanları birbirlerine olabildiğince yakın olmalıdır. Varyans bu kriteri ölçen ve minimize edilmesi gereken en temel ölçülerden birisidir.
2) Ayrıklık: Her bir küme birbirinden olabildiğince ayrık (uzak) olmalıdır.

Bu iki kriteri ölçen ise 3 temel yaklaşım mevcuttur:

- Tek bağ: İki kümenin en yakın elemanları arasındaki uzaklığı ölçer
- Tüm bağ: İki kümenin en uzak elemanları arasındaki uzaklığı ölçer.
- Merkezlerin karşılaştırılması: İki küme merkezleri arasındaki uzaklığı ölçer.

3.3.1. Katı kümelemede değerlendirme

Halkidi ve Vazirgiannis [2000] katı kümeleme için önerilen küme değerlendirme yöntemleri aşağıdaki gibi vermektedir.

- Geliştirilmiş Hubert Γ istatistiği

$$\Gamma = (1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} P(i,j).Q(i,j) \tag{3.15}$$

M=N(N-1)/2,

P:Veri kümesinin benzerlik matrisi.

Q:Her bir elemanın ait olduğu merkeze olan uzaklığını gösteren matris.

Normalize edilmiş Γ değerinin küme sayılarına göre değişimi 2 boyutlu grafik üzerinde görselleştirilirse Γ değerinde en dikkat çekici artışın gözlendiği nokta veriyi eniyi temsil eden küme sayısını verir.

- Dunn indeksi

$$D_{nc} = \min_{i=1,\ldots,nc}\left\{\min_{j=i+1,\ldots,nc}\left(\frac{d(c_i,c_j)}{\max_{k=1,\ldots,nc} diam(c_k)}\right)\right\} \tag{3.16}$$

c_i ve c_j iki farklı küme olmak üzere bu iki kümenin birbirine uzaklığı $d(c_i, c_j) = \min_{x \in c_i, y \in c_j} d(x, y)$, bir kümenin yarıçapı ise $diam(C) = \max_{x,y \in C} d(x, y)$ formülleri ile hesaplanır.

D_{nc} değerinin küme sayısına göre değişimini gösteren iki boyutlu grafikte maksimum D_{nc} değeri veriyi eniyi temsil eden küme sayısını verir.

- Davies Bouldin indeksi

$$DB_{nc} = \frac{1}{nc} \sum_{i=1}^{nc} R_i \qquad R_i = \max_{i=1,\ldots,nc} R_{ij}, \; i \neq j \tag{3.17}$$

DB_{nc} değeri her bir küme ile ona en yakın olan arasındaki ortalama benzerlik değerini vermektedir. DB_{nc} değerinin küme sayısına göre değişimini gösteren iki boyutlu grafikte en küçük DB_{nc} değeri veriyi eniyi temsil eden küme sayısını verir.

3.3.2. Bulanık kümelemede değerlendirme

Bulanık küme değerlendirme analizinde amaç veri elemanlarının kümelere yüksek üyelik değerleriyle atandıkları kümeleme şemasına ulaşmaktır. Bezdek (1984), bölüm katsayısı (*PC*) adını verdiği bir indis tanımlamıştır.

$$PC = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{nc} u_{ij}^2 \tag{3.18}$$

PC değeri [1/nc,1] arasında değişmektedir. Burada nc küme sayısını temsil etmektedir. Bütün üyelik fonksiyonları aynı değeri aldığında yani $u_{ij} = 1/nc$ olduğunda PC en küçük değere sahiptir. *PC* değeri *1/nc* ye ne kadar yakınsa kümeleme o kadar bulanıktır demektir.

Bölümleme entropi katsayısı (*PE*) bu kategoride yine Bezdek (1984), tarafından tanımlanmış başka bir indistir.

$$PE = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{nc} u_{ij}^2 . \log_a(u_{ij}) \tag{3.19}$$

Eşitlik 3.19'da *a* logaritmanın tabanıdır, 1'den büyük olan bütün küme sayıları için hesaplanır ve değeri $[0, \log_a nc]$ aralığında değişir. 0'a yakın *PE* değeri katı bir kümeleme olduğunu gösterir. Üst limite yakın olan *PE* değeri ise veri içerisinde küme yapısının bulunmadığını veya kullanılan algoritmanın küme yapısını çıkarmaya yeterli olmadığını göstermektedir.

Bu kategoride tanımlanan diğer bir değerlendirme indisi ise Xie ve Beni (1991) tarafından önerilen Xie-Beni indisidir. Bu indise göre $\prod = \frac{\sigma_i}{n_i}$ *I* kümesinin ortalama yoğunluğunu göstermektedir. σ_i, *I* kümesi için küme içi varyansların toplamını,

n_i ise I kümesindeki toplam veri elemanı sayısını temsil eder. Ayrıca en yakın iki küme arasındaki uzaklık, yani $d_{\min} = \min\left\|c_i - c_j\right\|$ şeklinde bulunduktan sonra *XB* indeksi;

$$XB = \frac{\Pi}{N.d_{\min}} \tag{3.20}$$

formülüyle hesaplanır. Daha küçük *XB* değerleri daha özlü ve daha iyi ayrıştırılmış kümeleri anlatır. Küme sayısı veri sayısına yaklaştıkça, *XB* değeri doğaldır ki monoton olarak azalmaktadır. Bu sakıncanın önüne geçebilmek için kullanıcı tarafından bir $c_{\max}$ değeri tanımlanmalı ve $[2, c_{\max}]$ aralığında bir *XB* değeri aranmalıdır.

Diğer bir index Fakuyama-Sugeno indeksidir. Pal ve Bezdek [1995] tarafından tanımlanmıştır.

$$FS_m = \sum_{i=1}^{N}\sum_{j=1}^{nc} u_{ij}^m \left(\left\|x_i - c_j\right\|_A^2 - \left\|c_j - c\right\|_A^2 \right) \tag{3.21}$$

Eş. 3.21'de *c, X* 'in ortalama vektörü *A* ise pozitif tanımlı simetrik bir matristir. *A=1* değeri olduğunda uzaklık kareli öklit uzaklığı olur. Yoğun ve iyi ayrıştırılmış kümeler için küçük FS_m değerlerine ihtiyaç vardır. Parantez içindeki ilk ifade kümelemenin yoğunluğunu ikinci ifade ise ayrıklığını temsil etmektedir.

Gath ve Geva (1989) tarafından önerilen diğer bir değerlendirme indisi ise hiperhacim ve yoğunluk kavramları üzerine kurulmuştur. *Jn*. kümenin bulanık kovaryans matrisi Σ_j olmak üzere;

$$\Sigma_j = \frac{\sum_{i=1}^{N} u_{ij}^m (x_i - c_j)(x_i - c_j)^T}{\sum_{i=1}^{N} u_{ij}^m} \tag{3.22}$$

olduğunda *j* nci kümenin bulanık hiper hacmi $C_j = \left|\Sigma_j\right|^{1/2}$ ile hesaplanır.

Burada $\left|\Sigma_j\right|$ değeri Σ_j nin determinantıdır ve aynı zamanda kümenin bir yoğunluk ölçüsüdür. Sonuç olarak toplam bulanık hiper hacim $FH = \sum_{j=1}^{nc} C_j$ formülüyle hesaplanır. Küçük *FH* değerleri daha özlü kümelerin bulunduğunu gösterir.

Ortalama bölüm yoğunluğu (*PA*) da yine bu kategoride tanımlanmış başka bir indistir.

$$PA = \frac{1}{nc}\sum_{j=1}^{nc}\frac{S_j}{C_j} \tag{3.23}$$

şeklinde formüle edilmiştir. Böylece $S_j = \Sigma_{x \in X_j} u_{ij}$, X_j c_j küme merkezi etrafında daha önceden tanımlanan bölgede bulunan veri noktalarının kümesi ve S_j ise j kümesini merkezi üyelerinin toplamıdır. Bölüm yoğunluk indeksinin farklı bir ölçüsü ise $S = \sum_{j=1}^{nc} S_j$ olmak üzere $PD = S / FH$ dir.

4. YAZIN ARAŞTIRMASI

Merkez tabanlı kümeleme problemindeki problem sahaları daha önceki bölümlerde anlatılmıştı. Bu bölümde

- K (küme sayısı) değerini kullanıcıdan istemesi,
- Başlangıçta belirlenen merkezlere duyarlı olması,
- Yerel en küçük tuzağına takılması

problemleri gibi bu çalışmanın içeriğini oluşturan ve üzerinde en çok çalışılan problemler hakkında literatür özeti verilmektedir.

4.1. Küme Sayısının Kullanıcıdan İstenmesi

Ball ve Hall (1967) tarafından önerilen ISODATA (Iterative Self Organizing Data Analysis Technique) tekniği K-Ortalama tekniğinin bir uzantısıdır. K-Ortalama tekniği bazı çalışmalarda Temel ISODATA olarak da adlandırılmaktadır [Duda ve Hart, 1973]. ISODATA, K-Ortalama kümelemede olduğu gibi iteratif olarak her bir elemanı en yakın merkeze atayan bir tekniktir. Bunun yanında eğer iki küme merkezi arasındaki uzaklık kullanıcı tarafından belirlenen bir eşik değerinin altında ise bu kümelerin birleştirilmesi imkanı da mevcuttur. Bunun tersi olarak dağınık bazı kümeleri de yine kullanıcı tarafından belirlenen bir eşik değerine göre iki farklı kümeye ayrıştırabilmektedir.

Tou (1979), yine ISODATA algoritmasına benzer dinamik bir yöntem geliştirmiş ve adına da DYNOC (Dynamic Optimal Cluster Seek) adını vermiştir. Bu algoritma en küçük küme içi uzaklığı ve maksimum kümeler arası uzaklığı arasındaki oranı maksimize etmeye çalışmaktadır. Bu yöntem da yine birleştirme ve ayrıştırma ile yinelenen bir süreç tarafından gerçekleşmektedir. ISODATA da olduğu gibi birleştirme ayrıştırma parametrelerinin yine kullanıcı tarafından belirlenmesi gerekmektedir.

Wallace (1986) ve Wallace ve Dowe (1994), nesnelere kümelere zeki bir şekilde atayabilmek için farklı yöntemler üzerinde çalışmışlardır. Her bir atamadan sonra ismine "Wallace Information Measure" (Minimum Description Length -MDL- olarak da bilinmektedir) denen bir ölçüm hesaplanmakta ve yapılan atama kabul veya red edilmektedir. Yine bu teknikte de birleştirme ve ayrıştırma olanağı vardır.

Bischof ve ark. (1999), K-Ortalama tekniği üzerine inşa edilmiş ve MDL çerçevesini kullanan bir yöntem önermişlerdir. Algoritma büyük bir *K* değeri ile başlamakta tanımlama uzunluğunu azaltmaya yönelik olarak kümelerin elimine edilmesi suretiyle ilerlemektedir.

Gath ve Geva (1989), Bulanık K-Ortalama ve bulanık en yüksek olabilirlik tahmininin bir kombinasyonu olan danışmansız bulanık kümeleme algoritması önermişlerdir. Yöntem kullanıcı tarafından belirlenen bir K alt limiti ile başlar. İlk olarak geliştirilmiş bir K-Ortalama tekniği veriye uygulanır. Bunun sonucunda oluşan merkezleri kullanarak bulanık en yüksek olabilirlik tahmini uygulanır. Bulanık en yüksek olabilirlik tahmini Öklit uzaklığı yerine üstel bir uzaklık fonksiyonu kullanır. Çünkü üstel uzaklık ölçüsü hiper ellipsoid kümeler için daha uygundur. Sonuçta oluşan kümelerin kalitesi küme değerlendirme indeksi kullanarak tespit edilir. Daha sonra *K* değeri artırılır ve aynı süreç kullanıcı tarafından belirlenen üst K limitine ulaşılıncaya kadar tekrar tekrar uygulanır. Gath ve Geva (1989), önerdikleri yöntemin farklı küme şekilleri için oldukça iyi çalıştığını belirtmektedir. Bunun yanında yöntemin karmaşıklığı yüksektir ve bu nedenle de yerel eniyiye takılma olasılığı mevcuttur.

Yager ve Filev (1994), tepe kümeleme adını verdikleri bir yöntem önermişlerdir. Tepe fonksiyonu adı verilen bir yoğunluk ölçüsü ile yaklaşık küme merkezlerini bulmaya yarayan bir yöntemdir. Sonucu yaklaşık olarak verdiği için diğer yöntemlerin öncüsü olarak kullanılabilir. Bu yöntem veri uzayındaki mümkün olan her pozisyonda bir tepe fonksiyonu (yoğunluk fonksiyonu) oluşturarak değerini hesaplar ve en yüksek yoğunluk değerine sahip olan pozisyonu ilk küme merkezi olarak seçer. Daha sonra birinci kümenin etkisini yok sayarak ikinci kümeyi

oluşturur ve bu süreç arzu edilen küme sayısına ulaşılıncaya kadar devam eder. Yöntem tek başına kullanılabileceği gibi diğer karmaşık kümeleme algoritmalarının öncülü olarak da kullanılabilir. Tepe kümelemenin en büyük dezavantajı boyut sayısı arttıkça karmaşıklık üstel olarak artmaktadır.

Chiu (1994) tarafından tepe kümeleme yönteminin çözüm karmaşıklığı problemine çözüm önerisi olarak eksiltici kümeleme adı verilen başka bir yöntem önerilmiştir. Tepe kümeleme yöntemine çok benzemekle birlikte veri uzayındaki her bir pozisyonda yoğunluk fonksiyonu hesaplamak yerine yoğunluk fonksiyonu olarak veri noktalarının değerini kullanır ve bu şekilde hesaplama miktarını dikkate değer bir şekilde azaltır.

Geliştirilmiş Linde-Buzo-Grey (MLBG) tekniği Rosenberger ve Chehdi (2000) tarafından önerilmiş bir yöntem olup süreç içindeki sonuçları kullanarak küme sayısını otomatik olarak belirlemeye çalışan bir tekniktir. Diğer tekniklere benzer şekilde K küme ile başlayan yinelemeli bir sürece dayanır.Her bir iterasyonda kümeler arası uzaklığı maksimize eden bir C_k kümesi ayrıştırmak üzere seçilir. Ayrıştırma işleminden sonra iki yeni merkez ortaya çıkmış olur. İlk merkez c_1 orjinal C_k kümesinin yeni merkezi ve diğer merkez c_2 ise yine C_k kümesi içerisinde fakat c_1 merkezine en uzak eleman olarak seçilir. Bundan sonra K+1 küme üzerinde K-Ortalama uygulanmaya devam edilir. Dağılma ölçüsü üzerine bina edilmiş bir uygulama kriterini sağlıyorsa yeni kümeleme kabul edilir. Bu sürecin dezavantajı kullanıcı tarafından 4 adet parametre girilmesini istemesidir.

Pelleg ve Moore (2000), X-Ortalama adını verdikleri ve K-Ortalamayı temel alan başka bir model önermişlerdir. Önerilen modelde bir küme ile sürece başlanır K-Ortalama işleminin yapılmasını müteakip her bir aşamada Bayes Bilgi Kriteri (BIC) ne göre ayrıştırma sürecine başlanır.

$$BIC(C/Z) = \hat{I}(Z/C) - \frac{K(N_d+1)}{2}\log N_p \qquad (4.1)$$

$\hat{I}(Z/C)$ Z veri yığınının C modeline göre maksimum olabilirliğinin ölçüsüdür. Eğer ayrıştırma süreci BIC değerinde düzelmeye yol açacaksa ayrıştırma yapılır aksi halde vazgeçilir. Bu yöntemde BIC değeri yerine başka değerler (skor fonksiyonları) de kullanılabilmektedir. Bu süreç kullanıcı tarafından belirlenen bir üst K değerine ulaşılıncaya kadar devam etmektedir.

Lorette ve ark. (2000), veri içerisinde saklı kümelerin sayısını dinamik olarak belirleyen bir bulanık kümeleme algoritması önermişlerdir. Önerilen yönteme danışmansız bulanık kümeleme yöntemi adı da verilmektedir. Yöntemde kullanılan amaç fonksiyonu formülü:

$$J_{UFC} = \sum_{k=1}^{K}\sum_{p=1}^{N_p} u_{kp}^{q} d^2(z_p, m_k) - \beta \sum_{k=1}^{K} p_k \log(p_k) \qquad (4.2)$$

q : bulanıklık değeri

u_{kp} : p nci veri elemanının k ncı kümeye üyelik değeri.

β : program ilerledikçe değeri azalan bir parametre

p_k ise $p_k = \frac{1}{N_p}\sum_{p=1}^{N_p} u_{kp}$ şeklinde formüle edilen C_k kümesinin apriori olasılığı.

Algoritma yüksek bir K değeri ile başlar. Daha sonra üyelik fonksiyonları ve merkezleri güncelleyerek devam edilir bu güncellemeler esansında p_k değeri de güncellenir. Eğer $p_k < \varepsilon$ ise k kümesi yok edilir.

Protnoy ve ark. (2001), yine K-Ortalama algoritması üzerine inşa ettikleri yeni bir yöntem önermişleridir. Küme sayısı (K değeri) Protnoy algoritması tarafından otomatik olarak belirlenmektedir. Fakat bu yöntem başlangıç *K* değeri ve küme genişliği için bir üst limit değerini (*W*) kullanıcıdan istemektedir. Kümeleme esnasında eğer bir veri ve ona en yakın olan merkez *W* değerinden daha küçükse o veri o kümeye atanır değilse bu veri elemanı yeni bir küme oluşturur ve *K* değeri bir

artırılır. Bu yöntem başlangıca karşı duyarlılık konusunda bir yenilik getirmemektedir. *K* sayısını otomatik olarak belirlediğini iddia etmesine rağmen kullanıcıdan istenen *W* değeri hakkında herhangi bir standart olmaması yöntemin zayıf noktalarını oluşturmaktadır.

Lin ve ark. (2001), genetik algoritmalar ve en yakın komşu tekniklerini birlikte kullanarak uygun küme sayısını otomatik olarak belirlemek için bir yöntem önermişlerdir. Bu yöntemde yine kullanıcıdan bir kaç parametre değeri istemektedir.

Huang ve ark.(2001), vektör nicemlemede kod çizelgesi tasarımı için tabu arama yöntemi temelli maksimum azalım yöntemi önermektedirler. Önerilen yöntemin daha iyi kod çizelgelerini daha kısa zamanda tasarladığı iddia edilmektedir. Maksimum azalım yönteminde ilk önce tüm veri yığınını tek bir küme olarak kabul etmekte daha sonra bu ilk kümeyi ikiye ayırmak suretiyle ve bu iki kümeden de en az amaç fonksiyon değerini verenleri yine ikiye ayırmak suretiyle hedeflenen küme sayısına ulaşılıncaya kadar süreç devam etmektedir. Bu tanımdan anlaşılacağı üzere maksimum azalım yöntemi oldukça maliyetli bir yöntemdir. Huang ve ark.(2001), bu maliyeti minimuma indirmek ve süreci hızlandırmak için tabu arama tekniğinden yararlanmışlardır. Tabu arama tekniğinin buradaki avantajı mümkün olan tüm çözümleri denemek yerine sadece uygun olan çözümlerden yararlanması ve işlem zamanını kısaltmasıdır.

Veenman ve ark. (2002), merkez tabanlı kümeleme problemi için K-Ortalama ve benzeri yöntemlerdeki *K* küme sayısının kullanıcı tarafından parametre olarak verilmesi yerine probleme maksimum varyans kısıtı adını verdikleri yeni bir kısıt ekleyerek küme sayısının otomatik olarak belirlenmesini sağlayan bir yöntem önermişlerdir. Önerilen yöntemde *K* sayısı kullanıcı tarafından verilmemekle birlikte bunun yerine maksimum varyans değeri bir eşik değeri olarak yine kullanıcı tarafından verilmektedir. Yazarlara göre iki kümenin birleşiminin varyansı belirlenen maksimum varyans değerinden daha büyük olmalıdır.

Huang (2002), SYNERACT adını verdiği ISODATA'ya alternative bir yöntem önermiştir. Bu yöntemde 3 kavram kullanılmaktadır:

- Bir kümeyi 2 ayrı kümeye ayıracak ve merkezlerini hesaplayacak bir hiperdüzlem,
- Herbir pikseli uygun kümeye atayacak yinelemeli bir kümeleme,
- Ayrıştırma sürecinden gelen kümeleri depolamak için ikili bir ağaç yapısı.

Yazara göre bu teknik en az ISODATA kadar doğru ve ondan daha hızlı bir tekniktir. Bunun yanında küme sayısının ve başlangıç merkezlerinin kullanıcı tarafından belirlenmesini gerektirmemektedir.

Veenman ve diğerleri (2002), Eş.4'3'deki küme değerlendirme indeksini minimize ederek optimal küme sayısını bulan hiyerarşik olmayan bir kümeleme yöntemi önermişlerdir.

Küme değerlendirme indeksi : $$IV = \min \sum_{k=1}^{K} n_k Var(C_k) \qquad (4.3)$$

Burada n_k değeri C_k kümesinde bulunan eleman sayısı ve $Var(C_k) = \frac{1}{n_k} \sum_{z_p \in C_k} \left\| x_p - c_k \right\|^2$, $Var(C_k \cup C_{kk}) \geq \sigma^2_{\max}, \forall C_k, C_{kk}, k \neq kk$ $\sigma^2_{\max}$ ise kullanıcı tarafından belirlenen bir parametredir ve sonuç üzerine önemli bir etkisi vardır. Önerilen yöntem küme sayısını veri sayısına eşitleyerek başlar. Müteakiben yukarıdaki indeks değerini minimize edecek şekilde kümeler birleştirilerek ve ayrıştırılarak süreç devam ettirilir. Veenman ve ark.'na göre önerilen teknik K-Ortalama ve beklenti maksimizasyonu yöntemlerinden daha iyi sonuçlar vermektedir. Bunun yanında yöntemin dezavantajları hesaplama karmaşıklığının yüksek olması ve $\sigma^2_{\max}$ parametresinin kullanıcı tarafından belirlenmesidir.

Hamerly ve Elkan (2003), *G* ortalama adını verdikleri K-Ortalama tekniği üzerine inşa edilmiş başka bir yöntem önermişleridir. Önerilen yöntem küçük bir *K* değeri ile başlamakta ve her bir iterasyonda Gauss dağılımına uymayan kümeleri ayrıştırarak devam etmektedir. Yazarlara göre yöntem sadece küresel ve ekliptik kümeler içeren veri üzerinde çalışmasına rağmen *X* ortalama tekniğinden daha üstün performans sergilemektedir.

Klawon ve Höppner (2003), bulanık kümelemedeki bulanıklaştırma parametresinin (*m*) kümeleme üzerindeki etkisini araştıran bir çalışma yapmışlardır. Yazarlara göre bu parametre kaç tane kümenin üstüste binebileceğini göstermektedir. Ayrıca yazarlar bulanıklaştırma parametresinin olumsuz etkilerinden bahsetmekte ve alternatif çözüm önerisi de sunmaktadırlar.

Kanade ve Lawrence (2003), bulanık kümeleme probleminde optimal küme sayısını belirlemek için karınca koloni en iyilemesi tekniğini uygulamışlardır. Bu konuda önerilen diğer yöntemlere benzer şekilde karınca koloni eniyilemenin arama prensiplerini kullanarak kümeleri birleştirmek ve ayrıştırmak suretiyle uygun küme sayısının ne olduğuna karar verilmeye çalışılmaktadır. Önerilen yöntemde elde edilen küme sayıları diğerlerinin aksine sürekli değerler alabilmektedir. Örneğin yazarlar iris veri yığını için farklı parametre değerleri kullanarak 2,16 ile 4,04 arasında, şarap veri yığını içinse 2,2 ile 4,64 arasında değerler elde etmişlerdir. Her iki veri yığını için de literatürde kabul edilen küme sayıları 3 olmasına rağmen veri yığınının özelliği nedeniyle burada elde edilen değerler de oldukça anlamlı görünmektedir.

Guan ve ark. (2004), yarı eniyi küme sayısı bulmak ve başlangıca karşı duyarlılığı elimine etmek için *K*-Means+ algoritması adını verdikleri bir yöntem önermektedirler. Küme sayısını belirlerken verinin istatistiksel özelliklerinden yararlanmaktadırlar. Buna ilave olarak başlangıçta oluşturulan kümeleri birleştir-ayrıştır yöntemiyle sürekli yenileyerek yarı optimal küme sayısına ulaştıklarını belirtmektedirler.

Wang ve Garibaldi (2005), kanser hastalığının teşhisi için uyguladıkları bulanık kümeleme problemindeki hem küme sayısını optimize etmek hem de belirlenen küme sayısında daha iyi performans değeri elde edebilmek için tavlama benzetimi sezgiselini kullanmışlardır. En uygun küme sayısını belirlemek için küme değerlendirme indekslerinden olan Xie-Beni indeksinden yararlanmışlar ve amaç fonksiyon değeri olarak da bu değeri kullanmışlardır. Bezdek (1998) tarafından önerilen en küçük ve maksimum küme sayılarını kullanarak $(2, \sqrt{n})$ kümeleri birleştirmek ve dağıtmak suretiyle mevcut çözümlere komşu çözümler üretmişlerdir.

Omran ve ark.(2005), küme değerlendirme indeksini en iyilemeye çalışan parçacık sürü en iyilemesi destekli bir kümeleme algoritması önermişlerdir. Önerilen yöntem optimal küme sayısını belirlemeye çalışırken aynı zamanda bu küme sayısına uygun yerleşim sistemi yani bölünme yapısını da vermektedir.

4.2. Aykırı Elemanlar ve Küme Şekilleri

Dave (1991), bulanık kümeleme yönteminde gürültüyü ele almak üzere gürültü kümeleme kavramını ortaya atmıştır. Gürültülü veri yığınının genel yapısına, dağılımına uymayan veridir. Kümeleme işlemi sonucunda gürültülü veriler genellikle hiç bir kümeye ait olmayan -hariç elemanlar- olarak kalmaktadırlar. Gürültülü verilerin çeşitli nedenleri olabilir. Özellikle büyük veri hacmine sahip veritabanlarında gürültülü veri girişi ve güncellemeden kaynaklanan bazı hatalardan kaynaklanabilir. Bunun yanında görüntü ve sinyal işleme problemlerinde ise veri bozulmuş olabilir. Bir çok kümeleme yöntemi gürültülü veriyi yok saymakta ya da bazı işlemlerle normalize etmektedir. Gürültü kümelemede önce gürültü kümesi belirlenmekte daha sonra bu kümeye ait benzerlik ölçüsü saptanmaktadır. Gürültü kümesi belirlemenin amacı tüm gürültülü verinin burada toplanmasını sağlamaktır. Fazladan bir merkez de bu küme için belirlenir ve bu merkezin tüm hariç (gürültülü) elemanların temsilcisi olduğu söylenir.

Setnes ve Kaymak (1998), bulanık K-Ortalama yönteminde merkez temsilcilerini nokta olarak göstermek yerine merkezi c_i, yarıçapı r_i olan olan hiperküplerle gösteren bir yöntem önermişlerdir. Önerilen yöntemde hiperkübün sınırları içine giren veriler o kümeye 1.0 üyelik derecesiyle üye olmaktadırlar. Halbuki normal bulanık kümelemede eğer merkez bizzat verinin kendisi değilse 1.0 üyelik değeri hiç bir zaman gerçekleşmemektedir. Yazarlar önerdikleri yöntemin kümeleme sonucunun veri dağılımına karşı duyarlılığı azalttığını belirtmektedir.

Wu ve Yang (2002), hem K-Ortalama, hem de bulanık K-Ortalama kümeleme yöntemini ele alan ve Öklit uzaklığı yerine kendi önerdikleri uzaklığını kullanan alternatif K-Ortalama ve alternatif bulanık K-Ortalama yöntemi önermişlerdir. Önerdikleri uzaklık ölçüsünün formülü $d(x,y) = 1 - \exp(-\beta\|x-y\|^2)$ şeklinde verilmektedir. Buradaki tek bilinmeyen olan β değeri pozitif bir sabit olarak belirtilmekte yığın kovaryansının tersi hesaplanarak elde edilmektedir. Önerilen yöntemde uzaklık ölçüsünün hesaplama şekli değiştiğinden üyelik fonksiyonu ve merkezlerin güncellenme şekilleri de buna göre değişmektedir. Algoritmanın performansına göre iris veri yığını için yapılan değerlendirmede alternatif K-Ortalama, temel K-Ortalama ile aynı sonucu verirken alternatif bulanık K-Ortalama temel bulanık K-Ortalamadan daha iyi sonuç verdiği belirtilmektedir.

Karayiannis ve Randolph (2003), K-Ortalama problemine Öklit uzaklığı kullanmayan bir yöntem önermişlerdir. Önerilen yöntem veri vektöründe her bir boyutun uzaklık hesaplamada farklı miktarda katkıda bulunduğunu ileri sürmekte bunun için w ağırlık değerlerini kullanmaktadır. Yöntem performans değeri olarak K-Ortalama probleminde kullanılan amaç fonksiyonunun yerine bunun yeniden formüle edilmiş şeklini $R_p(C,W_1,....,W_k) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{K}\sum_{l=1}^{K}\left(\|x_i - c_l\|^2 W_l\right)^p\right)^{\frac{1}{p}}$ kullanmaktadır. Klasik K-Ortalama tekniği bu önerilen formülde *p=1* ve *W=I* değerleri alınan özel bir durumdur.

Wu ve Yang (2006), bulanık K-Ortalama problemine temel bulanık K-Ortalama amaç fonksiyonuna ilave olarak bölünme katsayısı (*PC*) ve bölünme entropisi (*PE*) değerlerini de kullanan bir olabilirlikli kümeleme yöntemi önermişlerdir. Önerilen yöntemin özellikle gürültüye ve aykırı elemanlara karşı duyarlılığı bertaraf ettiği belirtilmektedir. Bu çalışmada aynı zamanda küme sayısı (*K*) ve bulanıklık değeri (*m*) ninde performans üzerindeki etkilerinden de bahsedilmektedir.

4.3. İlk Seçilen Merkezlere ve Veri Sırasına Karşı Duyarlılık

K-Ortalama algoritması eğer başlangıç merkezleri sonuçta ortaya çıkacak merkezlere, yani optimal merkezlere yakın yerlerde seçildiyse iyi sonuçlar üretebilmektedir [Jain ve Dubes 1988]. Özellikle aykırı örneklerden başlangıç merkezlerini seçmek kümeleme performansını olumsuz olarak etkilemektedir.

K-Ortalama kümeleme probleminde başlangıç merkezleri seçmek için dört genel yöntem önerilmiştir. Bunlar;

- Rasgele Yöntem : Veri kümesini rasgele k tane gruba bölmek,
- Forgy Yaklaşımı (FA) : Veri kümesinden K tane rasgele merkez seçmek ve geriye kalan elemanları uzaklıklarına göre bu merkezlerin tanımladığı kümelere atamak [Forgy, 1965],
- Macqueen Yaklaşımı (MA) : Veri kümesinden K tane rasgele merkez seçmek ve geriye kalan elemanları veri sırasına ve uzaklıklarına göre bu merkezlerin tanımladığı kümelere atamak. Bu yöntemde her atama sonunda merkezlerin güncellenmesi yapılmaktadır [Mac Queen, 1967].
- Kaufman Yaklaşımı (KA) : İlk önce tek bir merkez bütün veri yığınının en merkezinde bulanan elemanı temsilci yapmak suretiyle seçilir. Daha sonra geriye kalan elemanlardan her bir merkez en çok eleman kapsayacak şekilde kümeler oluşturulur [Kaufman ve Rousseuw, 1990].

Babu ve Murty (1993), başlangıç kümelerinin seçimi problemine genetik algoritma sezgiselini kullandıkları bir çözüm önerisi getirmişlerdir.

Katsavounidis ve ark. (1994), genelleştirilmiş Lloyd algoritmasında başlangıç koşullarına karşı duyarlılığı minimize edecek bir yöntem önermişlerdir. Önerilen yönteme göre başlangıç merkezlerini belirlerken ilk önce tüm vektörlerin normları hesaplanır. En büyük norma sahip olan vektör birinci merkez olarak seçilir. Daha sonra geriye kalan tüm vektörlerin ilk merkeze olan uzaklıkları hesaplanır ve en uzak olan vektör de ikinci merkez olarak seçilir ve bu durumda merkez sayısı 2 olur. Eğer küme sayısı sadece 2 ise bu noktada durulur değilse *K* tane küme için *K*'ya ulaşana kadar geri kalan *K-1* kümeye olan uzaklıklar hesaplanır ve en uzak olan vektör yeni merkez olarak atanır. Amaç o ana kadar hesaplanan merkezlere en uzak olan vektörü yeni merkez olarak atayarak daha ilk aşamada iyi ayrılmış kümelerin oluşmasını sağlamaktır.

Bradley ve Fayyad (1998), K-Ortalama kümelemede başlangıç problemine bir çözüm önerisi olarak kümeleri kümeleme adını verdikleri bir yöntem önermişlerdir. Yazarlar önce veriden rasgele örneklemler alarak bunları kümelemekte, daha sonra burada oluşan kümeleri kümeleyerek sonuca ulaşmaktadır. Çalışmada önerilen yöntemin en çok kullanılan başlatım yöntemi olan rasgele başlatımdan daha iyi sonuçlar ürettiği ve daha tutarlı olduğu belirtilmiştir.

Pena ve ark. (1999), K-Ortalama kümeleme yönteminde başlangıç merkezlerinin seçilmesi ile ilgili bir karşılaştırmalı çalışma yapmışlardır. Bu çalışmanın sonuçlarına göre random ve FA veri sırasından bağımsız olarak başlangıç merkezlerini oluşturmakta, MA ise veri sırasını bağımlı başlangıç merkezlerini oluşturmaktadır. Genel karşılaştırma sonuçlarına bakıldığında ise random ve Kaufman yöntemlerinin diğer iki yönteme göre daha iyi kümeleme sonucu ürettikleri gözlemlenmiştir.

Likas ve ark. (2003), tümel K-Ortalama adını verdikleri bir yöntem önermişlerdir. Yazarlar *K* kümeleme problemini çözerken tek bir kümeden başlamak suretiyle

ardışık olarak önce *K*-1 tane küme oluşturup merkezleri belirledikten sonra *K* ncı merkezi değişik noktalarda denemek suretiyle uygun çözümler aramaktadırlar.

Patane ve Russo (2001), klasik LBG algoritmasına yönelik olarak bir yöntem önermişlerdir. Önerilen yöntem özellikle kötü başlangıç değerleri atanan K-Ortalama kümeleme algoritmasının kötü sonuçlar üretmesine karşı bir tedbir olarak açıklanabilir. Yöntem küme merkezlerinin kaydırılması stratejisini belirlemektedir. Bu yöntemde her bir kod kelimesinin (küme merkezi) yani her bir küme temsilcisinin fayda değeri hesaplanmakta özellikle fayda değeri az olanların daha faydalı olanlara yaklaştırılması yoluyla daha iyi kümeleme sonuçları elde edilmesi önerilmektedir. Fayda değeri ise her bir küme için o kümeye ait çarpıklık değerinin ortalama çarpıklığa bölünmesi yoluyla elde edilmektedir. Hangi merkezler hangi istikamette hareket etmeli sorusuna cevap verilmektedir. Yazarlara göre yüksek çözünürlüklü optimal bir veri kümelemede her küme toplam çarpıklığa (hata kareleri toplamı) eşit katkıda bulunmalıdır.

Cano ve ark. (2002), açgözlü rasgele algoritma ve K-Ortalama kümelemeyi entegre eden (GRASP) bir kümeleme yöntemi önermişlerdir. Açgözlü rasgele arama ve Kaufman başlangıç merkezi seçme yöntemleri ile ilk kümeleme gerçekleşmekte müteakiben K-Ortalama ve yerel arama yöntemleriyle eniyi kümeleme oluşturulmaya çalışılmaktadır. Çalışmanın sonucunda diğer başlangıç seçme yöntemleri ile karşılaştırmalar yapılmış ve önerilen yöntemin daha iyi sonuçlar ürettiği örneklerle açıklanmıştır.

Cheung (2003), K-Ortalama kümeleme algoritmasının bir genellemesi olarak adlandırdığı K*-Ortalama kümeleme algoritmasını önermiştir. Önerilen yöntemde başlangıç merkezleri oluşturulduktan sonra geriye kalan elemalar rasgele olarak kendisine en yakın merkeze atandırılır. Burada bütün kümelerin merkezlerini hesaplamak yerine sadece kazanan küme yani en son veri elemanı hangi kümeye atandı ise onun merkezi hesaplanır. Sadece küresel değil elips şeklinde kümeler oluşturduğu da belirtilmektedir. Algoritmanın ayrıca doğru küme sayısını belirlemede de etkin olduğu bildirilmiştir.

He ve ark. (2004), K-Ortalama kümelemedeki başlangıç problemi ile ilgili olarak rasgele örnekleme, uzaklık en iyilemesi ve yoğunluk tahmini yaklaşımlarını ele alan bir karşılaştırma çalışması yapmışlardır. Bu çalışmada yazarlar performans kriteri olarak kümelerin kompaktlık değerine ilave olarak ayrılmışlık değerlerini de ele almışlardır. Çalışma sonucuna göre uzaklık en iyilemesi kullananan kümeleme yöntemi daha iyi ayrılmış kümeler üretmiştir. Ayrıca yazarlar en hızlı çalışan algoritmaların rasgele seçimle başlayan algoritmalar olduğunu belirtmektedirler.

Khan ve Ahmad (2004), K-Ortalama kümelemede başlangıca karşı duyarlılık problemine çözüm olarak bir yöntem önermişlerdir. Yöntem iki aşamadan oluşmaktadır. Birinci aşamada veri vektörlerinde toplu hesaplama yapmak yerine her bir boyutu tek tek ele alarak önce tekil kümelemeler yapılmakta daha sonra bu kümelemenin sonuçları birleştirilerek burada elde edilen merkezler başlangıç merkezleri olarak kullanılmakta ve bunun üzerine temel K-Ortalama algoritması uyarlanarak küme yapısına ulaşılmaktadır. Algoritmanın performansını değerlendirmek için uygunluk fonksiyonu olarak küme merkezi yakınlık indeksi (*CCPI*) adını verdikleri bir değer kullanmışlardır.

Qin ve Suganthan (2005), harmonik ortalama ve öğrenen vektör nicemleme tekniklerini entegre eden bir kümeleme tekniği önermişlerdir. Önerilen yöntemin başlangıç merkezlerine karşı duyarlılık riskini azalttığı belirtilmektedir.

Cui ve Potok (2005), parçacık sürü optimizayonu ve K-Ortalama kümelemeyi entegre eden bir doküman kümeleme yöntemi önermişlerdir. Yazarlara göre PSE sezgiseli K-Ortalama kümelemeden daha iyi sonuçlar vermekle beraber özellikle geniş veri kümelerinde işlem zamanı olarak K-Ortalama tekniği daha etkilidir. Bu nedenle önerilen yöntemde ilk aşamada PSE sezgiseli ile tümel en iyiye yakın değerler elde edildikten sonra K-Ortalama tekniği ile iyileştirme çalışması yapılmaktadır. Böylece her iki yönteminde avantajlarından yararlanılmaktadır. Aslında burada yapılan işlem K-Ortalama tekniğinin başlangıca karşı duyarlılığını yok etmeye çalışmaktan ibarettir.

4.4. Yerel En Küçük Tuzağına Takılma

Raghavan ve Birchand (1979), merkez tabanlı kümeleme problemi ile genetik algoritma sezgiselini birleştiren bir yöntem önermiştir. Önerilen yöntemde genetik algoritmalardaki yeniden üreme bölümünde bir iyileştirme yapılmıştır. Çalışmanın geleneksel kümeleme yöntemlerinden farkı olarak arama uzayında paralel arama yapması gösterilmiştir.

Al-Sultan (1995), K-Ortalama kümeleme problemine tabu arama tekniğini entegre ederek bir yöntem önermiştir. Yazar burada kullanılan kümeleme yöntemi katı kümeleme olduğundan sadece veri elemanlarının ait oldukları kümeleri değiştirerek yani U üzerinde oynama yaparak çözüm uzayında arama yapmaktadır.

Al-Sultan (1997), bulanık K-Ortalama kümeleme problemine tabu arama tekniğini entegre ederek bir yöntem önermiştir. Yazar mevcut çözüme komşu çözümler üretirken merkez vektörünün koordinatları ile işlem yapmanın üyelik değerleri ile işlem yapmaktan daha iyi çözümler ürettiğini belirtmektedir. Yazar merkez vektör koordinatlarını kaydırırken mevcut çözümdeki vektörü $c_t^i = c_c^i + \alpha d$ şeklinde kaydırmaktadır. Burada α değeri istikamet çarpanı d değeri ise 0,-1 ve 1 değerlerinden herhangi birini rasgele alan istikamet belirtecidir. d değerinin 0 olması herhangi bir kayma işleminin olmaması -1 değerini alması istikamet çarpanı kadar tam aksi istikamette kayma, 1 değerini alması ise istikamet çarpanı kadar aynı istikamette kayma olmasını belirtmektedir.

Delgado ve ark. (1997), bulanık kümeleme poblemine tabu arama tekniğini uygulamışlardır. Yazarlar tabu arama tekniğinin parametreleri olan maksimum iterasyon sayısı, deneme çözüm sayısı ve tabu listesi uzunluğu değerleriyle oynarken, aynı zamanda bulanıklık parametresi (m) değeri ile oynamak suretiyle eniyi çözümü bulmaya çalışmışlardır. Yazarlara göre özellikle iris veri yığını için eniyi amaç fonksiyon değeri EBTLU=850, ALÇS=15, iterasyon sayısı=1000 iken m=1.1 ve m=2,3 iken elde edilmektedir.

Fränti ve ark. (1997), vektör nicemlemede eniyi kod kitabını bulabilmek için genetik algoritma tekniğini kullanan bir yöntem önermişlerdir. Önerilen yöntemin getirdiği en önemli değişiklik genetik algoritmada kullanılan çaprazlama tekniğinde kullandıkları rasgele çaprazlama, merkez uzaklığı, ikili çaprazlama, en büyük kümeler ve ikili en yakın komşu teknikleridir.

Fränti ve ark. (1998), vektör nicemlemede eniyi kod kitabını bulabilmek için tabu arama tekniğini kullanan bir yöntem önermişlerdir. Önerilen yöntemde merkez vektörleri yani kod çizelgesi üzerinde değişiklikler yapılmak suretiyle komşu çözümler aranmaktadır. Yazarlara göre bir vektör herhangi bir kümeye atandığı zaman o kümenin merkezi atama yapılan vektöre doğru yaklaşmaktadır. Sonuçta tüm merkezler veri setinin merkezi etrafında toplanmaktadır. Bu yöntemde yine diğer benzer yöntemlerede olduğu gibi ya mevcut merkezlerin koordinaları rasgele bir vektör ile değiştirilmek suretiyle ya da mevcut çözüme gürültü ilave edilmek yani herhangi bir boyutta kaydırmak suretiyle komşu çözümler aranmaktadır. Tabu arama tekniğini kullanan diğer yöntemlerden farkı tabu listesiyle karşılaştırma yapılırken tam eşitlik değil yaklaşık eşitliğin dikkate alınmasıdır. Aday çözüm tabu listesindeki çözümlerden birine yeteri kadar yakınsa tabu sayılmaktadır.

Trejos ve ark. (1999), K-Ortalama kümeleme problemine yerel en küçük tuzağından kurtulmak için tabu arama, tavlama benzetimi ve genetik algoritma tekniğini uygulamışlardır. Çalışmada tavlama benzetimi ile ilgili bölümün Klein ve Dubes (1989) tarafından önerilen yönteme benzer olduğu belirtilmiştir. Mevcut çözüme komşu çözümler ararken üyelik değerlerini değiştirme stratejisi izlemişleridir. Yazarlar her bir yeni çözümde tüm kümelerin varyansını hesaplamak yerine sadece değişen kümelerin varyansını hesaplayarak işlem zamanını da azaltma yoluna gitmişlerdir.

Zhang ve ark. (1999), K-Ortalama kümeleme tekniğine yerel arama yöntemini entegre ederek yerel K-Ortalama adını verdikleri bir yöntem önermişlerdir. Önerilen yöntem diğer önerilenlerin aksine BIRCH yöntemi [Zhang ve ark., 1996] gibi daha

büyük veri kümeleri üzerinde çalıştırılmış ve hem hız olarak hem de performans değeri olarak daha uygun çözümler elde edilmiştir.

Krishna ve Murti (1999), K-Ortalama kümeleme problemine genetik algoritma tekniğini kullanarak çözüm üretmeye çalışan bir yöntem önermişlerdir. Önerilen yöntem üyelik değerleri üzerinde değişiklikler yaparak çözüm uzayında arama gerçekleştirmektedir.

Sung ve Jin (2000), K-Ortalama kümeleme problemine tabu arama tekniği ile entegre iki prosedür (paketleme ve serbest bırakma) uygulamışlardır. Önerilen algoritma iki aşamadan oluşmaktadır. Birinci aşamada başlangıç çözümü oluşturulmakta, ikinci aşamada ise mevcut çözüm üzerinde iyileştirmeler yapılmaktadır. Paketleme ve serbest bırakma proseürleri başlangıç çözümünün oluşturulmasında kullanılmaktadır. Birinci aşamada adına paketleme ve serbest bırakma adı verilen prosedürler yardımıyla veri elemanları benzerliklerine göre biraraya getirilerek (paketleme) ya da bazıları ayrıştırılarak (serbest bırakılarak) belli kritelere göre başlangıç çözümü oluşturulmaktadır. İkinci aşamada daha önce oluşturulan çözüme tabu arama ile iyileştirmeler yapılmak suretiyle daha iyi çözümler aranmaktadır.

Maulik ve Bandyopadhyay (2000), K-Ortalama kümeleme problemine genetik algoritma tekniğini kullanarak çözüm üretmeye çalışan bir yöntem önermişlerdir. Önerilen yöntem diğer K-Ortalama ve genetik algoritmayı kullanan diğer algoritmaların aksine üyelik değerlerini değil merkez koordinatlarını kullanmaktadır. Algoritma merkez vektörlerinden oluşan aday listeler üzerinde seçim ve çaprazlama işlemlerini gerçekleştirerek yeni çözümler üretmekte ve daha iyi çözümlere ulaşabilmektedir.

Fränti ve Kivijarvi (2000), tarafından önerilen kümeleme problemi için rasgele yerel arama tekniği K-Ortalama tekniğinin yerel en küçük değere takılmasına bir çözüm getirmektedir. Yazarlara göre K-Ortalama yöntemi başlangıç kümesi üzerinde yerel değişiklikler yaparak çözüm aradığından gördüğü ilk en küçük değere takılmaktadır. Başlangıca duyarlı olmasının nedeni de budur. Bu probleme çözüm önerisi olarak

mevcut çözüme komşu aday çözümler yaratılmalı ve bunlar etrafında çözüm arayışına gidilmelidir. Bu komşular yaratılırken üyelik değeri (U) ki burada katı kümeleme kullanıldığından sadece 1 ve 0 değerleri alabilmekte, merkez vektörlerden (C) ya da her ikisinin birleşiminden yararlanılmaktadır. Üyelik değerlerinden yararlanıldığında bazı nesnelerin ait oldukları kümeler değiştirilmekte ve çözüm bu şekilde aranmaktadır. Merkez vektörleri üzerinde iyileştirme yapılırken ise vektörün herhangi bir boyutunu vektör uzayında kaydırma gerçekleşmektedir. Burada mevcut çözüm üzerinde yapılacak iyileştirmeler mevcut çözümü çok fazla bozmayacak kadar küçük ama yerel en küçük tuzağından kaçınacak kadar da büyük olmalıdır.

Kövesi ve ark. (2001), vektör nicemlemede kod çizelgesi tasarımı için stokastik K-Ortalama tekniği önermişlerdir. Önerilen yöntemin temelini stokastik gevşetme yöntemi oluşturmaktadır. Stokastik gevşetme yöntemi, tavlama benzetimini andıran bir yöntem olmakla birlikte ondan farkı tavlama benzetiminde bir denge sağlamaya çalışan ısı azaltım mekanizmasının burada denge sağlamaya çalışmadan her iterasyonda azaltılmasıdır. Hem stokastik gevşetme yönteminde hem de önerilen yöntemde değişken olan konu üstel olasılık yoğunluk değerinin hesaplanma şeklidir.

Huang ve ark. (2001), vektör nicemlemede kod çizelgesi tasarımı için genetik algoritma ve tavlama benzetiminin birleşiminden oluşan melez bir arama yöntemi önermişlerdir. Önerilen yöntemin özelliği merkeze en uzak olan vektörü seçerek bu vektörü rasgele başka bir kümeye atayarak arama stratejisini kullanmasıdır. Algoritmanın temeli genetik yöntem ile elde edilen çözümleri tavlama benzetimi ile iyileştirme esasına dayanmaktadır.

Belacel ve ark. (2002), bulanık K-Ortalama problemi ile ilgili olarak bulanık J-Ortalama adını verdikleri bir yöntem önermişlerdir. Önerilen yöntemde amaç fonksiyonu olarak klasik bulanık kümeleme amaç fonksiyonunun yeniden formüle edilmiş hali kullanılmış böylece bilinmeyen teke indirilmiştir. Ayrıca komşu arama stratejisi olarak değişken komşu arama (VNS) sezgiseli kullanılmıştır. Bu sezgisele göre mevcut çözüme komşu çözümlere doğru hareket mümkün olan tüm merkez nesne eşleştirmeleri üzerinden tanımlanmaktadır.

Chu ve Roddick (2003), K-Ortalama kümeleme problemine tabu arama tekniği ve tavlama benzetimi tekniğini entegre ederek melez bir yöntem önermişlerdir. Yazarlar tabu arama tekniğini yerel olmayan çözümler üretmek başka bir deyişle bazı arama istikametlerini yoksaymak için tavlama benzetimini ise eniyi performans değerini bulmak için kullanmışlardır.

Kanade ve Lawrence (2004), bulanık kümeleme probleminde optimal merkezleri bulabilmek için bulanık karınca kümeleme tekniğini önermişlerdir. Önerilen yöntemde amaç fonksiyon değeri olarak temel bulanık kümeleme amaç fonksiyonunun yeniden formüle edilmiş hali kullanılmıştır. Yazarlar bu yöntemde merkezleri rasgele kaydırarak mevcut çözüm üzerinde daha iyi komşu çözümler arama stratejisini kullanmışlardır.

Shelokar ve ark. (2004), K-Ortalama kümeleme problemine karınca koloni eniyile sezgiselini entegre eden bir yöntem önermişlerdir. Önerilen yöntem üyelik değerleri üzerinde değişiklikler yaparak arama uzayında işlem yapmaktadır. Veri elemanlarının hangi kümelere ait olacağını yani üyelik değerlerini hesaplarken algoritma içinde oluşturulan ve karınca koloni yöntemlerinin mantığını oluşturan feromon matrislerinden yararlanılmaktadır.

Ye ve Chen (2005), K-Ortalama tekniği ve parçacık sürü en iyilemesi sezgiselini entegre eden KPSO adını verdikleri melez bir yöntem önermişlerdir. Önerilen yöntemde uzaklık ölçüsü olarak Wu ve Yang (2002) tarafından önerilen ölçü kullanılmış ve önerilen yöntemin K-Ortalama ve bulanık K-Ortalamadan daha iyi sonuçlar verdiği belirtilmiştir.

Pan ve Cheng (2006), vektör nicemlemede kod çizelgesi tasarımı için evrimsel tabanlı bir tabu arama yöntemi önermişlerdir. Yazarlara göre K-Ortalama algoritması her bir deneme çözümünde düzeltme yöntemi olarak K-Ortalamanın basit tepe tırmanma tekniğini kullandığından birbirine benzer bir çok deneme çözümünün ortaya çıkmasına ve erken yakınsamaya neden olmaktadır. Önerilen yöntemde çözüm uzayında daha iyi çözümler aranırken her birey ve o ana kadar ziyaret edilen

eniyi birey arasındaki uzaklık (tabu uzaklığı) her bir bireyin uygunluğunu hesaplamada önemli bir özelliktir. Bu uzaklığın 0 değerini alması demek aynı çözümün tekrar denemesi demektir. Dolayısıyla bu yöntemde bilinen hata kareleri toplamı amaç fonksiyonuna ilave olarak tabu uzaklığı kullanılmaktadır.

Bagirov ve Yearwood (2006), hata karelerinin en küçüğüne dayalı kümeleme problemi adını verdikleri bilinen adıyla K-Ortalama kümeleme problemine düzgün olmayan (nonsmooth) bir eniyileme algoritması önermişlerdir. Önerilen yöntem hem optimal küme sayısını bulmada hem de hata kareleri toplamını en küçüklemede etkili sonuçlar üretmektedir.

4.5. Diğer Katkılar

Linde ve ark. (1980), vektör nicemlemede kod kitabı tasarımı için LBG adını verdikleri bir yöntem önermişlerdir. Önerilen yönteme göre kod vektörlerinin ikiye bölünmesi sırasında orjinal kod vektöre küçük bir vektör eklenip ve çıkarılırak iki ayrı vektör elde edilir. Bu küçük vektör için, bölge içi kovaryans matrisinin en büyük özdeğerine karşılık gelen özvektörün doğrultusunu seçmek uygundur. Böylece eklenen ve çıkartılan vektör kod vektörün temsil ettiği bölgedeki en büyük varyasyonun bulunduğu doğrultuya karşılık gelen vektör seçilmiş olur. Bu şekilde kod vektör, resimleri eskisinden daha kötü temsil etmeyecek iki ayrı vektöre bölünür. LBG algoritmasının orijinal Lloyd-Max iterasyonunun sabit bir seviye sayısı ile iterasyonundan farkı, iterasyon sonucunda oluşabilecek olan “boş hücre” problemine sahip olmaması, ve yerel en küçük problemini daha iyi çözebilmesidir.

Fraley ve Rafteri (1998), kümeleme problemini beklenti maksimizasyonu (EM) ve Bayes bilgi kriteri (BIC) kullanarak çözen bir yöntem önermişlerdir. Önerilen yöntemin küme sayısını ve bu sayıya göre oluşan kümeleri ortaya koyduğu bunun yanında belirsizliği de ele aldığı yazarlar tarafından ileri sürülmektedir.

Kümeleme analizi aşamalarından birisi olarak benzerlik ölçütünün seçimi gösterilmesine rağmen Ramkumar ve Swami (1998), uzaklık fonksiyonu kullanmayan bir kümeleme yöntemi önermektedir.

Cogwill ve ark. (1999), K-Ortalama kümeleme problemine genetik algoritma tekniğini kullanarak çözüm üretmeye çalışan bir yöntem önermişlerdir. Bu yöntemde amaç fonksiyon değeri yani uygunluk fonksiyonu olarak varyans-rasyo kriteri kullanmışlardır. Yazarlar varyans–oran kriterini özet olarak kümeler arası varyans matrisinin izinin, küme içi varyanslar toplamı matrisinin izine oranı olarak ifade etmektedirler. Formül olarak vermek gerekirse $VRC = \frac{traceB/(k-1)}{traceW(n-k)}$ şeklinde ifade edilebilir.

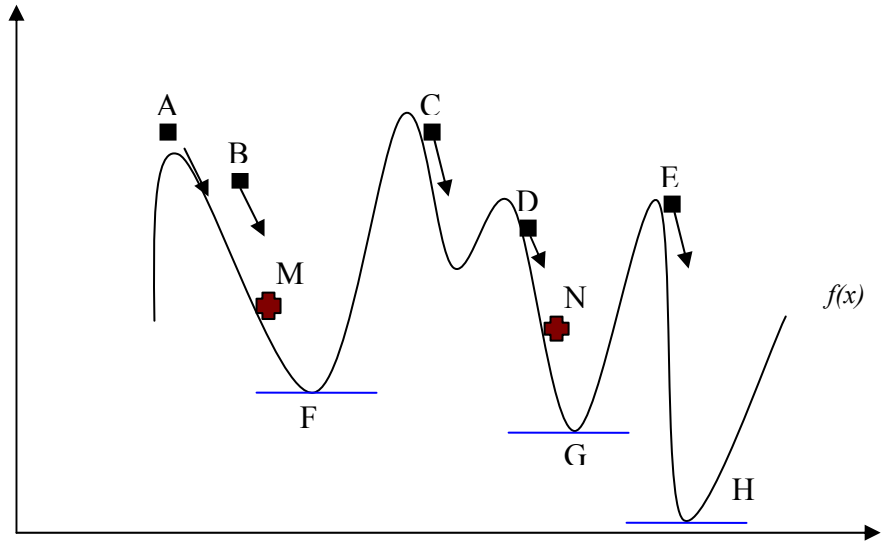
Pedrycz ve Vukovich (2004), gözetimli bulanık kümeleme adını verdikleri bir yöntem önermişlerdir. Önerilen yöntemde klasik kümeleme problemi bir nevi sınıflandırma problemi olarak ele alınmaktadır. Bu yöntemin amacı mevcut veri etiketlerinin veri içinde saklı bulunan yapısal özellikleri gerçekten temsil edip etmediğini belirlemek olarak açıklanabilir. Yöntemde orjinal bulanık kümeleme amaç fonksiyonuna gözetim bileşeni adı verilen yeni bir ilave yapılmaktadır.

Paterlini ve Krink (2006), merkez tabanlı katı kümeleme problemine rasgele arama, genetik algoritma, diferansiyel evrim, Parçacık sürü en iyilemesi sezgisellerini uygulayarak sonuçlarını değerlendiren bir karşılaştırma çalışması yapmışlardır. Performans fonksiyonu olarak varyans-oran kriteri (*VRC*), Mariott kriteri ($MC=\frac{\det W}{\det T}$) ve klasik hata kareleri toplamı kriteri (*W*) kullanmışlardır. Arama stratejileri üyelik değerleri üzerine yoğunlaşmaktadır. Çalışmanın sonucunda probleme uygulanan sezgisellerin sonuçları karşılaştırmalı olarak değerlendirilmiştir.

5. KÜMELEME ANALİZİ İÇİN GELİŞTİRİLEN SEZGİSEL YÖNTEMLER VE ANALİZLERİ

5.1. Sezgisel Yaklaşımlar

Bütün merkez tabanlı kümeleme algoritmalarında iki tane durma koşulu vardır. Birincisi belli bir iterasyon sayısı kadar ilerleyip durma, ikincisi ise eğer ardışık iki sonuç arasındaki fark belirlenen bir eşik değerinden (ε) küçükse durmaktır.



Şekil 5.1. Örnek yakınsama grafiği

Şekil 5.1'de iki boyutlu bir uzayda amaç fonksiyonu eğrisi ve rasgele belirlenmiş $\{A, B, C, D, E, F, G, H, M, N\}$ noktaları gösterilmektedir. Eğer başlangıç noktaları A,B,C,M,N noktalarından herhangi birisi olursa algoritma en iyimser ihtimalle yerel en küçük noktaları olan F ve G noktalarında duracaktır. Bazı durumlarda iterasyon miktarı nedeniyle belki de A ve B de başlayan M noktasında D de başlayan ise N noktasında durabilir. Halbuki Şekil 5.1'deki amaç fonksiyonuna göre tümel eniyi nokta H noktasıdır. K-Ortalama gibi tepe tırmanma algoritmasını kullanan yöntemler bu tür problemlerde genellikle ilk gördükleri yerel eniyi noktaya takılmaktadır.

Merkez tabanlı kümeleme probleminin *J(X,C)* performans kriterini optimize etmeye çalışan bir eniyileme problemi olduğu daha önceki bölümlerde anlatılmıştı. *J(X,C)* performans kriteri fonksiyonunun birden fazla yerel en küçük noktası olması nedeniyle problem ancak sezgisel yöntemler yardımıyla çözülebilmekte ve tümel eniyi olmasa bile buna yakın çözümlere ulaşılabilmektedir. En iyileme yazını incelendiğinde bu tür problemlerin çözümüne yönelik olarak geliştirilmiş bir çok sezgisel yöntem bulunmaktadır. Bu çalışmada bunlardan tabu arama, tavlama benzetimi ve parçacık sürü en iyilemesi sezgisellerinden yararlanılmış ve bu sezgisel yöntemler K-Harmonik Ortalama sezgiseli ile birleştirilerek yeni yöntemler önerilmiştir. Bundan sonraki bölümlerde öncelikle adı geçen sezgisel yöntemler hakkında genel bilgi verilmekte ve daha sonra önerilen sezgisel yöntemler anlatılmaktadır.

5.1.1. Tabu arama tekniği

Tabu arama algoritması ilk defa Glover (1989) tarafından insan hafızasının çalışmasından esinlenerek önerilmiş bir yerel arama yöntemidir. Genel olarak Basit Tepe Tırmanma (BTT) yönteminin zaaflarını gidermek üzere düşünülmüştür. BTT eldeki aday çözüme, bir komşuluk işleci uygulayarak, yeni adaylar üretir.Yeni adaylar bir değerlendirmeye tabii tutulur. Değerlendirme çözümün sonuca yakınlığını ölçer. Yeni adaylar ile eldeki eski aday içerisinden çözüme en yakın olan eskinin yerine geçer. BTT bu haliyle bir kısır döngüye sebep olabilir. Komşuluklar kapsamında birbirine eşit değerde iki ya da daha fazla komşu arasında BTT takılıp kalabilir. Arama yapılan alanın özellikleri ya da BTT yönteminin iyi seçilmemesi, herhangi bir denetim mekanizması kullanılmadan yapılan aramayı yerel eniyi noktalara götürebilir. Fakat gerçek hayattaki problemler yerel en iyilerin bol olduğu, fakat tümel en iyinin az, hatta tek olabildiği problemlerdir. Kısacası kısır döngü kaçınılmazdır.

Tabu arama tekniği kısır döngüden kaçınabilmek için hafıza kullanılmasını yani hatırlamayı önerir. Daha önce ziyaret edilmiş ya da herhangi bir nedenle ziyaret edilmesi istenilmeyen aday çözümlerle ilgili özellikler tabu listesi adı verilen, kısa

dönem hafızaya benzer bir yapıda tutulur. Yöntem bu listedeki hamlelerin belirli bir süre yapılmasını yasaklar. Böylece arama yerel eniyi noktadan kurtularak asıl sonuca yakınsayabilir. Bu temel altyapının yanısıra tabu arama yöntemlerinde aramayı belirli bir noktaya yönlendirebilecek uzun dönem hafıza yapıları kullanılmıştır. Tabu arama sezgiselin bileşenleri;

- *Başlangıç çözümünün oluşturulması***:** En genel şekilde başlangıç çözümü rassal olarak elde edilebilir. Ancak ilgilenilen problem için geliştirilmiş olan bir sezgisel algoritmadan yararlanılarak da başlangıç çözümünün elde edilmesi mümkündür.

- *Hareket mekanizması:* Mevcut bir çözümde yapılan bir değişiklikle yeni çözümün elde edilmesi hareket mekanizmasıyla gerçekleştirilir. Hareket mekanizmasındaki olası hareketler mevcut çözümün komşularını oluşturur. Hareket mekanizması algoritmanın etkinliği açısından önemli olduğu için problemin yapısına bağlı olarak uygun bir şekilde seçilmelidir.

- *Aday liste stratejileri:* Tabu arama algoritması yapılması mümkün olan, tabu olmayan amaç fonksiyonunun değeri açısından eniyi sonucu veren hareketlerin seçilmesi kuralına dayalı olarak çalışır. Aday liste stratejileri mümkün olan hareket listeleridir.Bu listelerden hareketler belli stratejilere göre seçilir.

- *Hafıza:* Tabu arama algoritmasının temel elemanlarından birisi de hafızadır. Arama boyunca ortaya çıkan durumlar hafızaya kaydedilir. Bu hafıza kısa dönemli hafıza olarak adlandırılır. Yapılmasına izin verilmeyen hareketler "tabu" olarak adlandırılır ve esnek hafıza içinde "tabu listesi" adı altında kaydedilirler. Bu hareketler belli süre sonra tabu listesinden çıkarılır ve yapılmasına izin verilir.

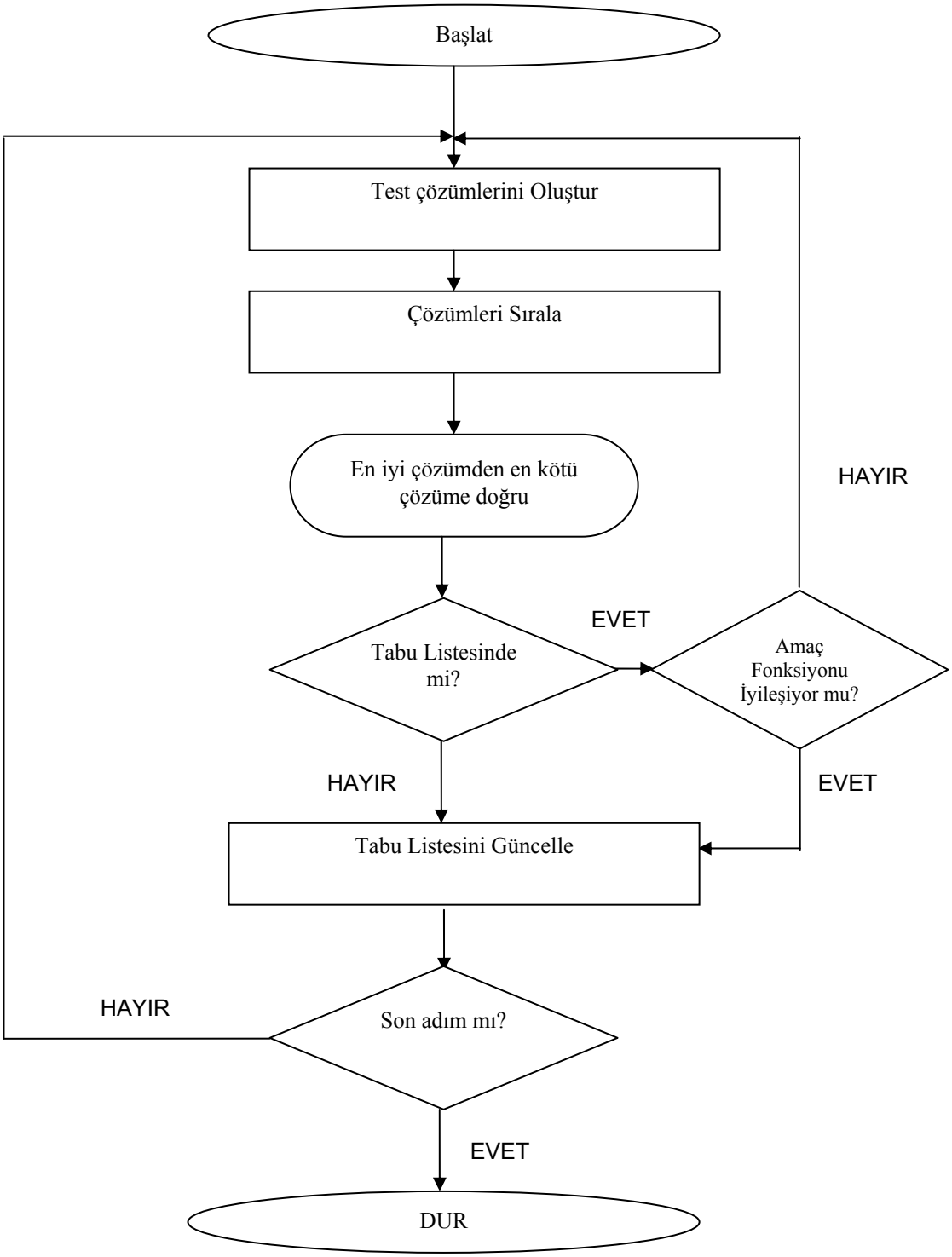
- *Tabu yıkma kriterleri:* Tabunun ortadan kalkabileceği durumları ifade etmektedir. En genel tabu yıkma kriteri, mevcut durumdan daha iyi bir sonuç verebilecek tabu hareketinin yapılmasına izin verilmesidir. Bu kriterin kullanılması tabu arama algoritmasının etkinliğini artırmaktadır. Ayrıca eğer tüm mümkün hareketler tabu ise

bu hareketlerden tabu süresinin bitmesine en yakın olan bir tabu hareketine izin verilir.

- *Durdurma koşulu:* Tabu arama algoritması bir veya birden fazla durdurma koşulunu sağlayıncaya kadar aramasını sürdürmektedir. Bu koşullardan bazıları aşağıda verilmiştir.

- Seçilen bir komşu çözümün komşusunun olmaması,
- Belirli bir iterasyon sayısına ulaşılması,
- Belirli bir çözüm değerine ulaşılması
- Algoritmanın bir yerde tıkanması ve daha iyi sonuç üretememesi.

Tabu arama algoritması bir başlangıç çözümü üreterek aramaya başlar. Algoritmanın her bir yinelemesinde tabu olmayan bir hareket ile mevcut çözümün komşuları arasından bir tanesi seçilerek değerlendirilir. Eğer amaç fonksiyonunun değerinde bir iyileşme sağlanmışsa komşu çözüm mevcut çözüm olarak dikkate alınır. Seçilen bir hareket tabu olmasına rağmen tabu yıkma kriterlerini sağlıyorsa mevcut çözüm oluşturmak için kullanılabilir. Geriye dönüşleri önlemek için bir takım hareketler tabu listesine kaydedilerek tekrar yapılması belli bir sure için yasaklanabilir. Belirlenen bir durdurma koşuluna göre algoritmanın çalışması sonlandırılır. Şekil 5.2'de tabu arama sezgiselinin akış şeması gösterilmektedir.



Şekil 5.2. Tabu arama algoritması akış şeması

5.1.2. Tavlama benzetimi tekniği

İlk olarak Kirkpatrick ve ark.(1983), tarafından önerilmiş olasılık tabanlı sezgisel bir algoritmadır. Katıların fiziksel tavlama işlemi ile tümleşik eniyileme problemlerinin çözümü arasındaki benzerlik üzerine dayalıdır. Günümüze kadar, bilgisayar tasarımı, görüntü işleme, moleküler fizik ve kimya, çizelgeleme gibi farklı alanlardaki bir çok eniyileme problemine uygulanmıştır Tavlama Benzetimi'nde hedef yerel en iyiden kurtulup tümel en iyiye ulaşmaktır.

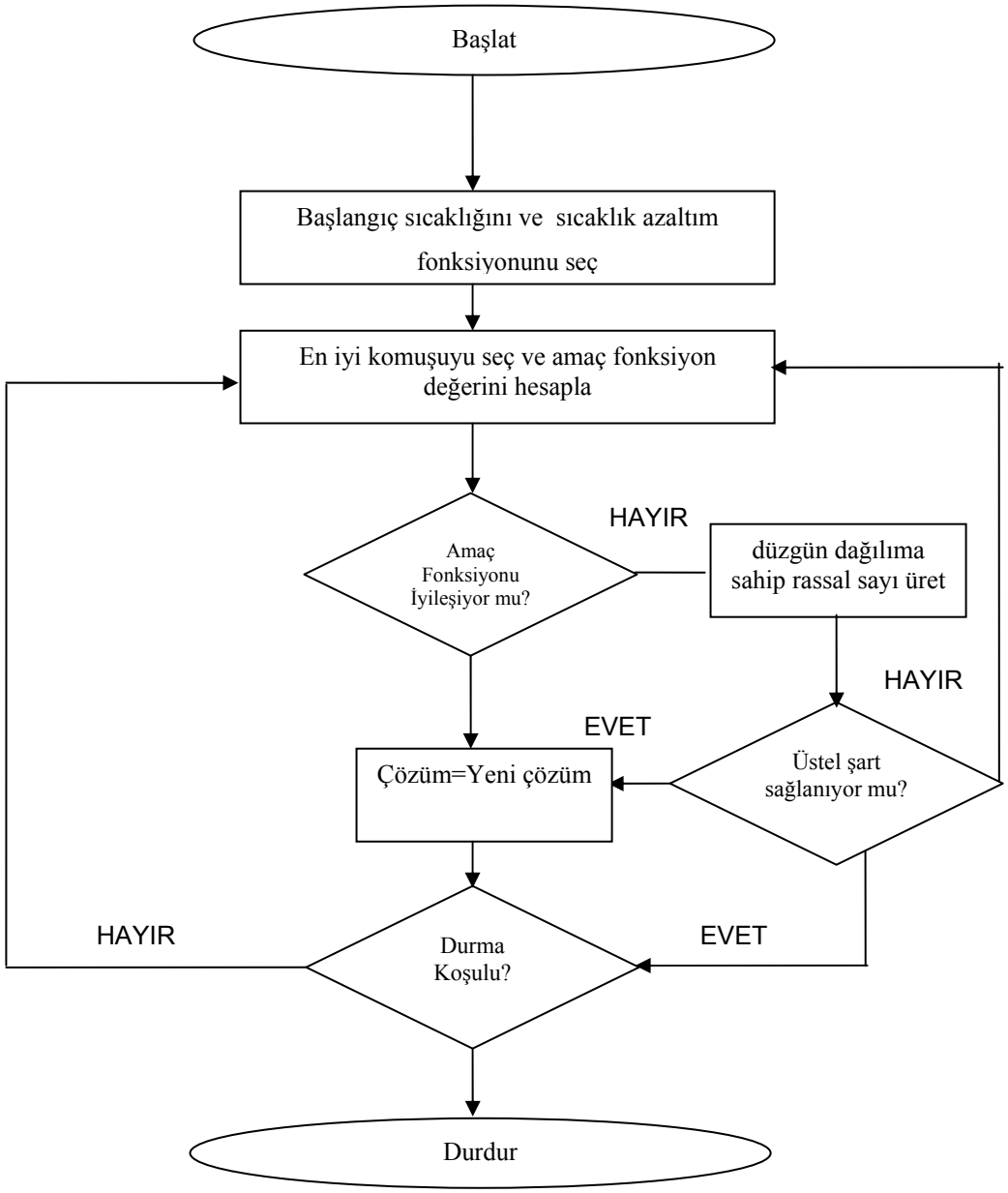
Tavlama, katının ergime noktasına kadar ısıtılması ve devamında kristalize olana kadar yavaşça soğutulması işlemidir. Malzemelerin atomları yüksek sıcaklıklarda yüksek enerji seviyelerindedir ve düzgün yerleşimler için daha fazla hareket serbestisine sahiptirler. Düzgün yapılı bir kristal sağlandığında sistem en küçük enerjiye sahiptir. Sıcaklık azaldıkça atomik enerji düşer. Eğer soğutma işlemi çok hızlı gerçekleşirse kristal yapıda bozukluklar ve düzensizlikler ortaya çıkacaktır. Bu nedenle soğutma işleminin özenle yapılması gerekir. Fiziksel tavlama işlemi Monte Karlo tekniği üzerine dayalı olarak Metropolis ve ark. (1953) tarafından modellenmiştir. Verilen bir T sıcaklığında sistem enejilerinin olasılık dağılımı $P(E) = e^{-E/(kT)}$ ile belirlenir. Burada E sistem enerjisi, k ise Boltzmann sabitidir. Küçük bir karışıklıkla sistem durumunda değişiklik yaratılması halinde Metropolis algoritmasına göre sistemin yeni enerjisi hesaplanır. Eğer enerji azalmış ise sistem bu yeni duruma geçer, eğer enerji artmış ise yeni durumun kabul edilip edilmeyeceğine yukardaki eşitlik kullanılarak şu şekilde karar verilir. Düzgün dağılımdan rasgele bir p sayısı üretilir eğer $p \leq e^{\Delta E/T}$ ise yeni durum kabul edilir. Aksi takdirde mevcut durum değiştirilmez.

Tavlama Benzetimi (TB) tekrarlamalı bir algoritmadır, yani algoritma çözüm uzayında sayıların vektörü formunda ifade edilen bir çözümü sürekli olarak geliştirmeye çalışır. Performansı büyük oranda seçilen soğutma planına bağlıdır.

Tavlama Benzetimi (TB), karesel eniyileme problemleri için iyi çözümler veren olasılıklı karar verme yöntemi olarak görülebilir. Kontrol parametresi "sıcaklık"tır ve minimizasyon problemleri için çözümün daha iyi bir çözüme ulaşmasının kabulu olasılığını değerlendirir.

Kabul fonksiyonuna göre, çözümde meydana gelen küçük artışların kabul edilme olasılığı, büyük artışların kabul edilme olasılığından daha fazladır. Ayrıca, *T* yüksek olduğunda hareketlerin çoğu kabul edilecektir. *T* sıfıra yaklaştığında ise, çözümde artışa yol açan hareketlerin çoğu reddedilecektir. Bu nedenle TB algoritmasında, yerel eniyi tuzaklarına düşülmesini engellemek için göreceli olarak yüksek bir *T* değeri ile aramaya başlanır. TB algoritması, bir taraftan sıcaklık yavaş yavaş azaltılırken, her sıcaklık değerinde belli sayıda hareket deneyerek arama işlemini sürdürür. Algoritma önceden tanımlanmış iterasyon sayısına bağlıdır. İlk once rassal olarak başlangıç çözüm kümeleri üretilmekte ve kaç iterasyonda sonlandırılacağı belirlenmektedir. Daha sonra TB belirlenen en yüksek ısı (*Tmax*) ile başlamakta ve *t* fonksiyonu ile ısı *Tmin*'den küçük olana kadar soğutulmaktadır. Böylece, seçilen ve işlenen çözüm kümesi, tekrardan çözüm kümeleri içine diğerleriyle kıyaslanmak için konmaktadır. Daha sonra çözüm kümeleri içindeki diğer herhangi bir çözüm rassal olarak seçilmekte ve işlenmektedir.

Şekil 5.3'de temel tavlama benzetimi sezgiselinin akış şeması verilmektedir.



Şekil 5.3. Tavlama Benzetimi algoritması akış şeması

5.1.3. Parçacık sürü en iyilemesi tekniği

Geleneksel arama algoritmalarından farklı olarak evrimsel hesaplama teknikleri arama uzayındaki potansiyel çözüm noktalarından oluşan popülasyonlar üzerinde çalışırlar. Bu yöntemler potansiyel çözümler arasında bir dayanışma ve yarış ortamı yaratarak eniyi veya en iyiye yakın çözüme daha çabuk ulaşırlar.

Parçacık Sürü En iyilemesi (PSE), genetik algoritma gibi popülasyon temelli sezgisel bir sürü algoritmasıdır. Kuş ve balık sürülerinin iki boyutlu hareketlerinden esinlenerek Kennedy ve Eberhart (1995) tarafından geliştirilmiş, Shi ve Eberhart (1998) tarafından modele atalet ağırlığı ilave edilmiştir. Bireyler arasındaki sosyal bilgi paylaşımını geliştirmeyi amaç edinmiştir. Evrimsel algoritmalara benzer, her bir aday çözüm parçacıkla, popülasyon ise sürü ile ifade edilir. Her bir aday çözüm (parçacık) bir sonraki pozisyonunu hız vektörü, kendi tecrübesi (yerel en iyi) ve sürü tecrübesine (tümel en iyi) göre ayarlar.

PSE eniyi ya da en iyiye yakın çözüm bulmak için, önce her biri aday çözümü sunan bireyler (parçacıklar) oluşturur. Bireylerin bir araya gelmesinden çözüm için gerçekleştirilen popülasyon meydana gelir. PSE, bireyler arasında bilgi paylaşımını esas alır. Her bir parçacık kendi pozisyonunu sürüdeki eniyi pozisyona doğru ayarlarken bir önceki tecrübesinden (*pbest*) yararlanır. Sürünün tecrübesi ise tüm parçacıklar arasındaki eniyi değer (*gbest*) olarak alınır ve algoritma sonlandığındaki *gbest* problemin çözümünü vermektedir.

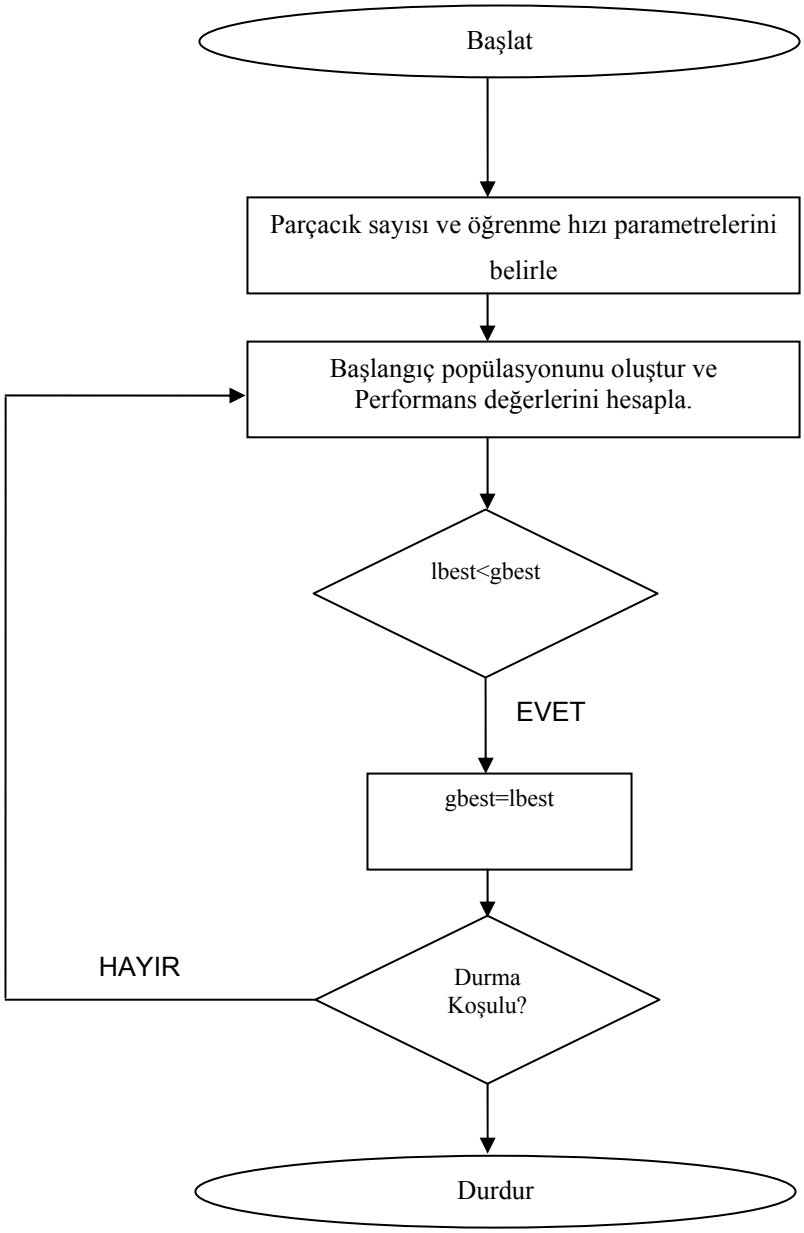
PSE algoritmasının parametreleri:

G:Parçacık sayısı	$YİNESA$: Maksimum iterasyon sayısı
i: popülasyondaki i *nci* parça	Pozisyon vektörleri : C
c_1 : bilişsel öğrenme hızı	Hız vektörleri: V
c_2 : sosyal öğrenme hızı	Yerel eniyi vektörler: *pbest*
t: t nci iterasyon (zaman)	Tümel eniyi vektörler :*gbest*
r: 0-1 aralığında rasgele bir sayı	Atalet ağırlığı : ω

Bu parametrelerden öğrenme hızlarının (c_1, c_2) yakınsama üzerinde kritik bir etkisi yoktur. Yakınsama üzerinde en etkili olan parametre atalet ağırlığı (ω) parametresidir. Bu parametrenin değeri gereğinden büyük alınırsa arama süreci daha geniş bir alanda yapılırken gereğinden küçük alınması ise daha yerel arama yapmaya yol açmaktadır. Bu nedenle en uygun atalet ağırlığı değeri yerel eniyiye takılmayacak kadar büyük ama gereksiz aramalara yolaçmayacak şekilde küçük bir değer olmalıdır [Abraham ve ark., 2006].

Sousa ve ark. (2004), parçacık sürü en iyilemesi tekniğinden veri madenciliğinde birliktelik kurallarının keşfi ve sınıflandırma analizinde yararlanmışlardır. Mahamed (2005), görüntü tanımlama ve renkli görüntü nicemleme problemine parçacık sürü en iyilemesi tekniği ile iyileştirme yöntemleri önermiştir. Feng ve ark (2006), PSE tekniği ile evrimsel bulanık teknikleri entegre eden bir vektör nicemleme yöntemi önermişlerdir.

Şekil 5.4'de parçacık sürü en iyilemesi algoritmasının akış şeması gösterilmektedir.



Şekil 5.4. Parçacık sürü en iyilemesi algoritması akış şeması

5.2. Geliştirilen Yöntemler

K-Harmonik Ortalama kümeleme K-Ortalama kümelemenin başlangıca karşı duyarlılığını azaltmasına rağmen bu yöntemin yerel minimuma yakalanma olasılığı hala devam etmektedir. Bu çalışmanın amacı K-Harmonik Ortalama veri kümeleme analizinde yerel en küçük tuzağına takılmadan globale en yakın çözümler bulabilmektir. Bu nedenle bir önceki bölümde detayları verilen tabu arama, tavlama

benzetimi ve parçacık sürü en iyilemesi sezgisellerinden yararlanılarak üç tane sezgisel yöntem geliştirilmiştir. Bundan sonraki bölümlerde geliştirilen sezgisel yöntemler anlatılmaktadır.

5.2.1. Tabu arama tekniği ile KHO kümeleme (TABAKHO)

Tabu arama tekniği yerel en küçük tuzağına yakalanma riski olan algoritmaları iyileştirmede bir başka deyişle eniyi veya eniyiye yakın sonuç bulmada kullanılan etkili ve uygulanması kolay bir yöntemdir. Bunun yanında yapılan testlerde bu tekniğin harmonik ortalamaya göre işlem zamanını dikkate değer bir şekilde azalttığı gözlemlenmiştir.

Önerilen yöntemde çözüm olarak ele alınan değerler küme merkez değerleridir. Merkez tabanlı kümeleme analizinde amaç elde edilen merkezlerin oluşturduğu kümelerdeki elemanların küme içi varyansı en küçük ve kümeler arası uzaklık değerleri ise maksimum olacak şekilde k tane merkez bulmaktır. Tabu listesini dolduracak olan ve arama uzayında hareket etmeyi sağlayacak olan bulunacak küme merkez değerleridir ve geliştirilen sezgisel yöntemin amacı bunları eniyi veya eniyiye yakın bir noktaya taşıyarak problemi çözüme kavuşturmaktır.

Aşağıda tabu arama tekniği ile K-Harmonik Ortalama kümeleme algoritması verilmektedir.

1) K küme merkezini k adet rasgele seçilen veriye ya da veri topluluğunu içeren uzayda k adet rasgele tanımlanan noktaya uygun şekilde seç.

2) K-Harmonik Ortalama tekniğini kullanarak deneme çözüm sayısı kadar çözüm üret.

3) Üretilen çözümleri amaç fonksiyon değerine göre küçükten büyüğe sırala. En baştaki çözüme karşılık gelen merkezler eğer tabu listesinde yoksa veya tabu listesinde var fakat bu amaç fonksiyon değeri eniyi amaç fonksiyon değerinden küçükse 4'e git. Eğer bütün sonuçlar için bu koşullar sağlanmıyorsa 2 'ye git.

4) Yeni bulunan merkez değerlerini tabu listesinin sonuna koy. Eğer tabu listesi dolduysa yani maksimum limitine ulaştıysa en üstteki elemanı sil.

5) Eğer bir yakınsama koşulu karşılanmamışsa 2. Adıma geri dön.

6) Yakınsama koşulu sağlanmış ise üyelik değerlerini durulaştırarak her bir elemanın hangi kümeye ait olduğunu belirle.

TABAKHO sezgiselinin detayları aşağıda verilmektedir.

EBTLU: En büyük tabu listesi uzunluğu
AL*ÇS*: Deneme çözüm sayısı
P: Olasılık eşiği
ITERSA: Maksimum İterasyon sayısı
TLU: Anlık tabu listesi uzunluğu

1) S_c geçici bir çözüm ve Z_c bu çözüme ait amaç fonksiyon değeri olsun. $S_b = S_c$ ve $Z_b = Z_c$ olsun. *EBTLU*, *P*, *ALÇS*, ve *YİNESA* için başlangıç değerlerini belirlenir. *k*=1 *TLU*=1 yapılır.

2) S_c yi kullanarak *ALÇS* tane deneme çözümü oluşturulur (S_t^n) ve bu çözümler için amaç fonksiyon değerlerini hesaplanır (Z_t^n). 3. aşamaya geçilir.

- Belirlenen arama stratejilerine göre deneme çözümlerini oluşturulur. Burada deneme çözümlerini oluşturmak için başlangıç noktası olarak S_c yi kullanılmakta ve asıl problem S_c den yola çıkarak gidilecek en uygun yeri bulmaktır. 2 adet farklı strateji uygulanabilir;

a) *Rasgele değiştirme*: Rasgele seçilen bir küme merkezi yok edilir ve bunun yerine yine rasgele seçilen bir veri elemanı (vektör) x_j yeni merkez olarak belirlenir.

b) *Mantıksal değiştirme*: Merkezleri dağınıklık dengeleme ve fayda fonksiyonuna göre kaydırma esasına dayanır [Fritzke, 1997 - Patane and Russo, 2001]. Burada önce kümeleme sürecine ait ortalama dağınıklık D_{avg} ve her bir kümeye ait dağınıklık D_i hesaplanır. Buradan elde edilen bilgi ışığında $U_i = \frac{D_i}{D_{avg}}$ formülü ile

her bir kümeye ait bir fayda fonksiyonu değeri U_i hesaplanır ve and U_i<1 değeri olanlar U_i>1 olanlara doğru kaydırılır.

- [0,1] aralığında rasgele bir sayı (***r***) yaratılır. Eğer bu sayı başlangıçta belirlenen olasılık değerinden küçükse r<P mevcut çözümü belirlediğimiz stratejiye göre oluşan çözümle değiştirilir aksi halde mevcut çözümle devam edilir.

3) Z_t^n ler artan bir şekilde sıralanır ve en üsttekini alınır (Z_t^1). Eğer Z_t^1 tabu değilse ya da tabuysa fakat $Z_t^1 < Z_b$ ise $S_c = S_t^1$ ve $Z_c = Z_t^1$ yapılır ve 4'e gidilir. Aksi halde Z_t^n lerin en iyisini al ve Z_t^1 için yapılan testler tekrarlanır. Eğer bütün Z_t^n ler için test başarısız olursa 2'ye gidilir.

4) S_c tabu listesinin sonuna yerleştirilir ve *TLU*=*TLU*+1 yapılır. Eğer *TLU*=*EBTLU*+1 ise tabu listesindeki ilk eleman silinir ve *TLU*=*TLU*-1 yapılır. Eğer $Z_b > Z_c$ ise $S_b = S_c$ ve $Z_b = Z_c$ yapılır. Eğer k= *YİNESA* ise durulur aksi halde k=k+1 yap ve 2 ye gidilir.

5.2.2. Tavlama benzetimi tekniği ile KHO kümeleme (TAVBEKHO)

Tmax: En yüksek ısı

Tmin: En düşük ısı

t: Soğutma derecesi

YİNESA : Maksimum iterasyon sayısı

Tavlama benzetimi ile KHO kümeleme algoritma benzetim kodu aşağıda verilmektedir.

1) S_c geçici bir çözüm ve Z_c bu çözüme ait amaç fonksiyon değeri olsun. $S_b = S_c$ ve $Z_b = Z_c$ yapılır, *Tmax*, *Tmin*, *t*,ve *YİNESA* için başlangıç değerlerini belirlenir. k=1 yapılır.

2) S_c yi kullanarak deneme çözümü oluşturulur (S_t^n) ve bu çözüm için amaç fonksiyon değerlerini hesaplanır (Z_t^n). 3. aşamaya geçilir.

3) Belirlenen arama stratejilerine göre deneme çözümlerini oluştur. Burada deneme çözümlerini oluşturmak için başlangıç noktası olarak S_c kullanılıyor ve asıl problem S_c den yola çıkarak gidilecek en uygun yer neresidir? ($p = e^{-\Delta Z / tn}$) olasılık eşiğini hesapla. 2 adet farklı strateji uygulanabilir;

Rasgele değiştirme: Rasgele seçilen bir küme merkezi yok edilir ve bunun yerine yine rasgele seçilen bir veri elemanı (vektör) x_j yeni merkez olarak belirlenir**.**

Mantıksal değiştirme: Merkezleri dağınıklık dengeleme ve fayda fonksiyonuna göre kaydırma esasına dayanır [Fritzke, 1997 - Patane and Russo, 2001]. Burada önce kümeleme sürecine ait ortalama dağınıklık D_{avg} ve her bir kümeye ait dağınıklık D_i hesaplanır. Buradan elde edilen bilgi ışığında $U_i = \dfrac{D_i}{D_{avg}}$ formülü ile her bir kümeye ait bir fayda fonksiyonu değeri U_i hesaplanır ve and U_i<1 değeri olanlar U_i>1 olanlara doğru kaydırılır.

4) Eğer ($\Delta s = Z_n^t - Z_n$) <0 ise eski çözümü yeni çözümle değiştirilir. Aksi halde [0, 1] aralığında rasgele bir sayı üretilir (***r***). Eğer $e^{-\Delta Z / tn}$ > r ise mevcut çözümü belirlenen stratejiye göre oluşan çözümle değiştirilir, aksi halde mevcut çözümle devam edilir.

5) Eğer k= *YİNESA* ise durulur, aksi halde k=k+1 yapılır ve 2'ye gidilir.

5.2.3. Parçacık sürü eniyileme tekniği ile KHO kümeleme (PARSEKHO)

Parçacık sürü en iyilemesi ile KHO kümeleme benzetim kodu aşağıda verilmektedir.

1. Parçacık sayısı G, *YİNESA* , c_1, ve c_2 parametrelerini belirlenir. t=1 yapılır.

2. G tane çözümü S_c rasgele oluşturulur, bu çözümlere ait amaç fonksiyon değerlerini Z_c hesaplanır ve her bir parçacık için $pbest = Z_c$ yapılır.

3. En iyi Z_c değerini alınır ve $gbest = Z_c$ yapılır.

4. Hız vektörü

$$V_i(t+1) = \omega.V_i(t) + c_1 * r * (pbest_i(t) - C_i(t)) + c_2 * r * (gbest(t) - C_i(t))$$

formülüne göre güncellenir.

5. Pozisyon vektörünü $C_i(t+1) = C_i(t) + V_i(t+1)$ formülüne göre güncellenir.

6. Amaç fonksiyon değerleri pbest ve gbest güncellenir. t=t+1 yapılır.

7. Eğer t=YİNESA ise *gbest* eniyi çözümdür ve durulur . Aksi halde 4 e gidilir.

5.3. Algoritmaların Etkinliklerinin İncelenmesi

Kümeleme analizi problemi için geliştirilen her bir algoritma kendine özgü performans değerlendirme fonksiyonuna sahiptir [Zhang, 2001]. Bunun yanında daha önceden sınıf etiketleri bilinen veri yığınlarının kümelenmesinde performans fonksiyonu olarak doğru kümelere ayrılan elemanların yüzdesi de kullanılmaktadır. Ancak genelde bu tür performans kriteri çok fazla ayırdedici olmadığından amaç fonksiyon değerleri daha çok kullanılmaktadır.

Farklı algoritmaları aynı fonksiyonla değerlendirebilmek için hem uygulamasının diğerlerine oranla daha kolay olması hem de en popüler algoritma olması nedeniyle bu çalışmada K-Ortalama kümeleme algoritmasının performans fonksiyonu seçilmiştir. Ayrıca bazı bölümlerde buna ilave olarak küme değerlendirme indekslerinden bölünme entropisi, bölünme katsayısı, Xie-Beni indeksi ve Fukuyama-Sugeno indeksleri de karşılaştırmalar yapmak maksadıyla kullanılmışlardır.

5.4. Test Problemleri

Bu bölümde veri kümeleme analizinde literatürde en çok kullanılan ve bu çalışmada da kullanılan veri yığınları verilecek ve daha sonra bu veri yığınlarına uygulanan teknikler ve sonuçları karşılaştırmalı olarak anlatılacaktır.

Rasgele veri yığını: İki boyutlu 15 adet veriden oluşan ve aşağıda içeriği verilen veri yığınıdır. Tipik bir rasgele veri yığını Çizelge 5.1'de verilmektedir.

Çizelge 5.1. Rasgele veri yığını

1	8	2	2	8	2	8	4	7	1	9	1	9	5	4
1	4	6	1	0	2	2	5	3	2	3	5	1	6	8

*İris veri yığını***:** Özellikle görüntü tanıma problemlerinde neredeyse bütün çalışmalarda temel alınan veri yığınıdır. Fisher (1936) tarafından ilk defa kullanılmış ve literatürde bir klasik haline gelmiştir. Her biri dört öznitelikten oluşan toplam 150 adet veri satırı bulunmaktadır. Öznitelikler sırasıyla sepal uzunluk, sepal genişlik, petal uzunluk ve petal genişliktir. Ölçümler cm. cinsinden yapılmıştır. Bu zamana kadar tespit edilen optimal küme sayısı üçtür ve her bir kümede 50 eleman bulunmaktadır. Sınıf isimleri

-- Iris Setosa

-- Iris Versicolour

-- Iris Virginica dır.

Eksik ve hatalı veri yoktur.

Şarap veri yığını: Yine MCI laboratuvarından alınmış bir veri yığınıdır. Bu veri İtalyada aynı yörede yetişen üç farklı cins üzümün kimyasal analizi sonucunda elde edilmiştir. Analiz sonucunda üzümlerin şaraba 13 farklı boyutta katkı sağladığı sonucuna varılmıştır. Dolayısıyla öznitelik sayısı 13 tür. 178 adet veri satırı mevcuttur. Eksik ve hatalı veri yoktur. Bu zamana kadar tespit edilen optimal küme sayısı üçtür ve kümelerde sırasıyla 59,71,48 eleman bulunmaktadır. Eksik ve hatalı veri yoktur.

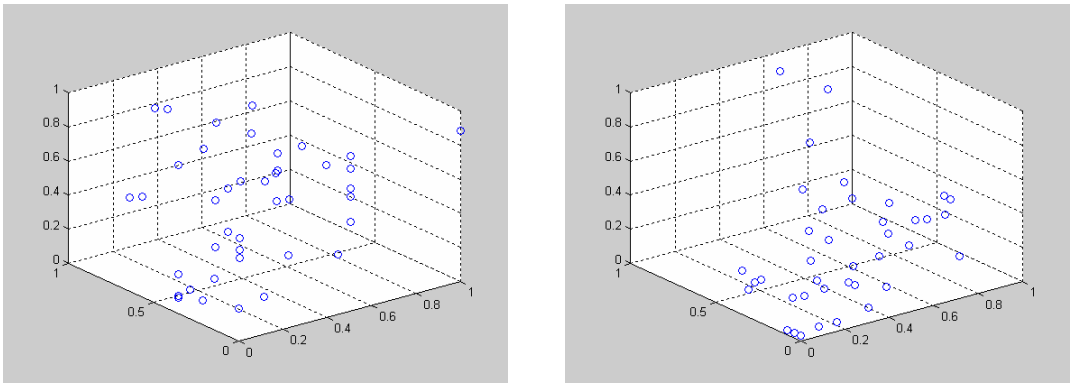
Göğüs kanseri veri yığını: Wisconsin üniversite hastanesinden alınan gerçek veri yığınıdır. Kanser hastalığının tespitinde yapılan tahlil sonuçlarında kanser mikrobunun iyi huylu veya kötü huylu olduğunu tespit etmede kullanılmaktadır. Her biri 10 öznitelikten oluşan 699 adet veri satırı mevcuttur. Birinci öznitelik hastaları tanımlayan numara olduğundan analizde sadece 2-10 arası öznitelikler kullanılmıştır. Küme sayısı ikidir. 16 adet eksik veri mevcuttur.

Bardak veri yığını: Suçluların tespiti amacıyla bardakların delil olarak kullanılması için cam analizi sonucunda elde edilen ve MCI laboratuvarından alınmış bir veri yığınıdır. Her biri 10 öznitelikten oluşan 214 adet veri satırı mevcuttur. Birinci öznitelik malzemeyi tanımlayan numara olduğundan analizde sadece 2-10 arası öznitelikler kullanılmıştır. Küme sayısı ikidir. Eksik ve hatalı veri yoktur.

Yerleşim yerleri ver yığını: Türkiye'de bulunan yerleşim birimlerinin değişik özellikleri ele alınarak gruplanması hedeflenmiştir. Bu nedenle Devlet İstatistik Enstitüsü arşivlerinden alınan aşağıda açıklaması yapılan veri yığını üzerinde çalışılmıştır. Çalışmanın tutarlı ve doğru sonuç verebilmesi için içeriği doğru olduğu varsayılan 41 adet ilin verileri ele alınmıştır. İlk ikisi tanımlayıcı olmak üzere toplam 17 adet öznitelik belirlenmiştir. Bunlar sırasıyla:

F1=İl
F2=İlçe
F3=Nüfus
F4=Rakım
F5=Ocak ayı sıcaklık ortalaması
F6=Temmuz ayı sıcaklık ortalaması
F7=Ankara'ya mesafe
F8=İstanbul'a mesafe
F9=İlkokul sayısı
F10=Lise sayısı
F11=Fakülte sayısı
F12=Dersane sayısı
F13=Sinema sayısı
F14=Kamu Bankası sayısı
F15=Özel Banka sayısı
F16=Hastane sayısı
F17=Diğer sağlık kuruluşu miktarı

Öznitelik tanımlarından anlaşılacağı üzere herbir nitelik değeri birbirinden farklı aralıklara sahiptir. Özellikle nüfus verisi büyüklük olarak diğer alanlardan çok daha fazla değerler içermektedir. Bu husus ise nüfus niteliğinin diğer nitelikler üzerinde baskın olmasına neden olmaktadır. Bu hassasiyeti gidermek için veriler hem bulundukları şekliyle hem de 0-1 aralığında normalize edilerek kullanılmışlar ve her iki sonuç da gösterilmiştir. Şekil 5.9'da örnek olarak ele alınan üç boyut gösterilmekte, Çizelge 5.3'de ise 41 ile ait 17 öznitelik değerleri görülmektedir.



(a) (b)

Şekil 5.5 Yerleşim verisinin 3 boyutlu örnek gösterim
a) F6,F7,F8 boyutları b) F12,F13,F16 boyutları

Okul Veri Yığını: Bir okul veritabanından alınan küçük bir örneklem üzerinde algoritmanın işleyişi anlatılmaktadır. Çizelge 5-2' de bir okuldan örnekleme alınan 9 öğrenciye ait 5 dersten aldıkları notlar görülmektedir.

Çizelge 5.2 Okul veri tabanı örneği

Öğrenci	*Mat*	*Fen*	*İng*	*Türkçe*	*Beden*
Can	6	6	5	5,5	8
Ali	8	8	8	8	9
Gamze	6	7	11	9,5	11
Serap	14,5	14,5	15,5	15	8
Didem	14	14	12	12,5	10
Faruk	11	10	5,5	7	13
Tekin	5,5	7	14	11,5	10
Metin	13	12,5	8,5	9,5	12
Ersin	9	9,5	12,5	12	18

Çizelge 5.3. Yerleşim veri yığını

F1	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17
Adıyaman	178538	675	5	31	530	879	228	5	3	9	1	4	3	3	27
Afyon	128516	1050	1	22	239	285	95	6	8	9	2	5	11	7	29
Amasya	74393	400	3	24	265	579	81	4	4	5	1	4	6	2	17
Artvin	23157	600	3	21	771	1078	34	1	2	3	1	3	2	1	9
Aydın	143267	70	8	29	493	364	79	6	6	13	1	5	15	5	24
Balıkesir	215436	145	5	25	428	179	99	10	4	13	3	4	19	5	28
Batman	246678	570	4	32	752	1100	72	5	1	10	2	3	6	3	15
Bingöl	68876	1150	2	27	669	1011	101	3	1	3	1	3	2	2	16
Bolu	84565	730	1	20	138	223	85	3	5	5	2	4	10	5	16
Burdur	63363	960	3	25	333	383	74	3	2	7	2	3	5	2	16
Çanakkale	75810	6	7	25	551	238	41	4	7	8	4	4	8	3	13
Çorum	161321	820	0	21	191	507	158	7	4	12	1	5	12	5	26
Denizli	275480	400	6	28	406	360	105	9	6	24	2	7	32	6	47
Edirne	119298	60	3	25	566	216	69	3	7	7	2	4	13	5	17
Elazığ	266495	1050	0	27	567	914	178	12	10	13	2	7	10	7	31
Erzincan	107175	1189	2	24	569	904	104	6	3	6	1	4	6	2	20
Giresun	83636	30	7	23	482	792	73	3	3	7	2	5	7	5	14
Hakkari	58145	1750	4	25	983	1329	50	2	1	2	1	3	2	1	8
Isparta	148496	1056	2	24	314	386	66	6	10	14	2	4	10	4	26
Kmaraş	326198	550	5	28	442	785	285	10	7	18	2	4	14	5	25
Kars	78473	1750	9	18	874	1190	81	2	5	4	1	3	5	2	14
Kırıkkale	205078	740	1	25	57	406	54	8	8	15	1	4	5	4	20
Kırşehir	88105	980	0	23	143	489	52	4	4	6	1	4	6	3	19
Malatya	381081	965	1	28	505	854	166	18	8	27	2	7	15	5	32
Manisa	214345	75	7	29	491	298	128	5	4	10	3	4	17	8	31
Muğla	43845	660	6	27	496	426	50	2	8	4	1	3	7	6	13
Nevşehir	67864	1260	0	22	218	559	42	3	1	9	1	3	8	4	20
Niğde	78088	1250	0	23	271	596	88	6	5	9	2	3	6	3	37
Ordu	112525	50	7	23	441	748	64	4	1	11	3	6	9	3	13
Rize	78144	50	7	23	661	970	102	4	4	6	2	4	6	3	19
Siirt	98281	900	3	31	819	1166	50	4	1	4	1	3	2	1	8
Sinop	30502	20	7	23	302	527	48	1	3	4	1	3	1	2	8
Sivas	251776	1255	3	20	357	697	161	6	7	12	2	8	10	7	26
Şanlıurfa	385588	515	6	32	604	951	353	7	7	11	1	4	14	5	23
Tekirdağ	107191	10	5	24	467	123	37	3	1	8	2	3	9	5	15
Tokat	113100	623	2	22	319	645	139	5	5	9	4	4	6	4	21
Trabzon	214949	50	7	23	592	901	94	8	9	28	2	8	23	8	28
Uşak	137001	900	3	24	329	263	122	5	4	9	2	3	12	3	20
Van	284464	1725	3	23	924	1265	147	9	9	10	3	5	10	5	17
Yozgat	73930	1300	2	20	168	512	99	5	4	3	3	4	3	2	16
Zonguldak	104276	10	6	22	191	240	168	6	3	8	1	5	10	4	27

Analiz çalışmalarında kullanılacak olan veri yığınları belirlendikten sonra geliştirilen sezgisellerde kullanılan parametrelerin alacağı değerler belirlenecektir.

5.5. Parametrelerin ayarlanması

Parametre belirlemek için yapılan ön çalışmalarda denemeler farklı büyüklüklerde iki farklı veri yığını (bardak, rasgele) üzerinde yapılmıştır. Kullanılan farklı parametre değerleri sonucunda elde edilen veriler üyelik değerleri durulaştırılarak elde edilen KO kümeleme performans değeri kullanılarak değerlendirilmiştir. Parametrelerin performans kriteri üzerindeki etkisini araştırmak için varyans analizi (ANOVA) kullanılmıştır.

Geliştirilen sezgisel yöntemlerin tümü için başlangıç çözümünün oluşturulmasında (ilk merkezlerin atanmasında), rasgele atama tekniği kullanılmıştır.

Hareket mekanizmasının seçimi ile ilgili olarak kümeleme analizi çalışmalarında kullanılan üç farklı yöntem denenebilir. Bunlar rasgele ikili yer değiştirmeye dayalı hareket, kullanıcı tarafından yönlendirilen hareket ve tüm kümelerin performans değerine eşit oranda katkıda bulunduğu fayda fonksiyonuna dayalı hareket mekanizmasıdır. Bu çalışmada fayda fonksiyonuna dayalı hareket mekanizması kullanılmıştır ve bu hareket mekanizmasına geliştirilen algoritma içinde mantıksal değiştirme ismi verilmiştir.

Tabu arama sezgiselinin diğer parametreleri olan yineleme sayısı, en büyük tabu listesi uzunluğu ve alternatif çözüm sayısını (alternatif liste stratejisi) belirlemek için, yineleme sayısı olarak 100, 300, 500 ve 1000, en büyük tabu listesi uzunluğu değeri olarak 7, 10, 20 ve 50, alternatif liste uzunluğu olarak ise 10, 20, 30, 50 ve 100 seçilerek 4 x 4 x 5= 80 farklı parametre kombinasyonu ile algoritma 5'er koşum çalıştırılmıştır. Varyans analizi sonucunda $\alpha = 0.05$ düzeyinde her bir parametrenin algoritma performansı üzerinde etkili olduğu görülmüştür. İkili ve üçlü ortak etkiler değerlendirildiğinde ise sadece *EBTLU* ve *ALÇS* parametrelerinin ikili ortak etkisi anlamlı bulunmuştur. Çizelge 5.4'de *EBTLU* ve *ALÇS* ikilisi için varyans analizi sonuçları, Çizelge 5.5'de ise farklı parametre değerlerine göre elde edilen ANOVA testi sonuçları gösterilmektedir.

Çizelge 5.4. TA sezgiseli ANOVA sonuçları (*EBTLU* ve *ALÇS* ikilisi)

Kaynak	Kareler Toplamı	Serbestlik derecesi	Ortalama Kare	F-Değeri	p-Değeri
Sütunlar	8,333	3	2,778	8,820	0,002
Satırlar	4,425	4	3,510	3,51	0,040
Hata	3,778	12	0,315		
Toplam	16,536	19			

Yineleme sayısının algoritma performansı üzerinde sadece işlem zamanını artırma yönünde bir etkisi vardır. Alternatif liste çözüm sayısı olarak 20-30 aralığında seçilecek bir değerin ele alınan test problemleri için diğer değerlere oranla daha iyi sonuçlar ürettiği gözlemlenmektedir. Tabu arama sezgiselinin hafıza parametresi olarak da adlandırılan *EBTLU* parametresi içinse 20-50 arasında atanan değerlerin daha iyi sonuçlar ürettiği gözlemlenmektedir.

Çizelge 5.5. TA sezgiseli parametre ayarlama sonuçları

Kaynak	Kareler Toplamı	Serbestlik derecesi	Ortalama Kare	F-Değeri	p-Değeri
YİNESA	8,082	3	2,694	8,88	0,002
ALÇS	4,425	4	3,510	3,51	0,040
EBTLU	8,333	3	2,778	8,820	0,002
YİNESA X ALÇS	21,43	12	10,715	2,11	0,237
YİNESA X EBTLU	15,097	9	7,549	1,15	0,403
ALÇS X EBTLU	11,271	12	1,878	4,77	0,005
YİNESA X ALÇS X EBTLU	16,967	36	8,484	1,41	0,345

Tavlama benzetimi sezgiselinde başlangıç sıcaklığının, sıcaklık azaltma oranının ve durdurma koşulunun belirlenmesi tavlama planı olarak tanımlanmaktadır. Algoritmanın etkinliğini sağlamak için bu planın doğru olarak belirlenmesi gerekmektedir. Başlangıç sıcaklığının uygun şekilde seçilmesi TB algoritmasının performansı üzerinde etkilidir. Kirkpatrick (1983), mevcut değerden daha yüksek değerlerin % 80'ini kabul edebilen başlangıç sıcaklıklarının uygun olduğunu belirtmektedir. Bu nedenle tavlama planı oluşturulurken % 80 üzerindeki kabul olasılığının ilk gözlendiği 1000 değeri (ortalama % 81) alt limit olarak belirlenmiştir. Algoritma parametrelerinin belirlenmesi için yapılan ön çalışmalarda başlangıç

sıcaklığı için 1000, 10000, 100000 değerleri, sıcaklık azaltma oranı için 0,8, 0,9 ve 0,995 değerleri, son sıcaklık değeri içinse 1 ve10 değeri seçilerek, 3 x 3 x 2=18 farklı parametre kombinasyonu ile algoritma 5'er koşum çalıştırılmıştır ve tavlama planı oluşturulmaya çalışılmıştır. Varyans analizi sonucunda $\alpha = 0.05$ düzeyinde başlangıç sıcaklığı ve soğutma oranı parametrelerinin algoritma performansı üzerinde etkili olduğu görülmüş, durma sıcaklığının etkisi tek başına anlamlı bulunmamıştır. İkili ve üçlü ortak etkiler değerlendirildiğinde ise başlangıç sıcaklığı ve soğutma oranı parametrelerinin ikili ortak etkisi anlamlı bulunmuştur.

Çizelge 5.6. TB sezgiseli parametre ayarlama sonuçları

Kaynak	Kareler Toplamı	Serbestlik derecesi	Ortalama Kare	F-Değeri	p-Değeri
Tmax	1,264	2	0,632	9,36	0,031
t	200,686	2	100,343	1485,94	0
Tmin	0,968	1	0,968	2,09	0,174
Tmax X t	6,629	4	1,657	10,24	0,001
Tmax X Tmin	595,04	2	297,52	1,82	0,274
t X Tmin	571,91	2	285,95	1,85	0,270
Tmax X t X Tmin	25,615	4	12,808	3,03	0,158

Parametre belirleme ilgili yapılan çalışmaların sonuçları Çizelge 5.6' da ANOVA testi sonuçları ise Çizelge 5.7'de gösterilmektedir. Bu sonuçlara göre tavlama benzetimi sezgiselinin en kritik parametreleri olarak başlangıç sıcaklığı ve soğutma oranı görülmektedir.

Çizelge 5.7. TB sezgiseli ANOVA testi sonuçları (*Tmax* ve *t* ikilisi)

Kaynak	Kareler Toplamı	Serbestlik derecesi	Ortalama Kare	F-Değeri	p-Değeri
Sütunlar	1,264	2	0,632	9,36	0,031
Satırlar	200,686	2	100,343	1485,94	0
Hata	0,27	4	0,068		
Toplam	202,22	8			

Diğer sezgisel yöntemlerde olduğu gibi, PSE sezgiselininde performansı, seçilen parametre değerleri ile doğrudan ilişkilidir. Algoritma parametrelerinin belirlenmesi için iterasyon sayısı olarak 300, 500 ve 1000, bilişsel ve sosyal hızları olarak 1,5-2,

parçacık sayısı değeri olarak 20, 30, 50, atalet ağırlığı değeri olarak 0,7, 0,9 ve 1,2 değerleri seçilerek, algoritma 3 x 2 x 3 x 3=54 farklı parametre kombinasyonu ile 5'er koşum çalıştırılmıştır. Yapılan denemeler sonucunda bilişsel ve sosyal öğrenme hızlarının algoritmanın performansı üzerinde anlamlı bir etkisi olmadığı, buna karşılık iterasyon sayısı, parçacık sayısı ve atalet ağırlığı parametrelerinin tekli olarak, parçacık sayısı ve atalet ağırlığının ise ikili olarak algoritmanın performansı üzerindeki etkileri anlamlı bulunmuştur. Çizelge 5.8'de parçacık sayısı ve atalet ağırlığının ikili olarak ele alındığında ortaya çıkan ANOVA testi sonuçları gösterilmektedir.

Çizelge 5.8 PSE sezgiseli ANOVA testi sonuçları (*G* ve *w* ikilisi)

Kaynak	Kareler Toplamı	Serbestlik derecesi	Ortalama Kare	F-Değeri	p-Değeri
YİNESA	1,264	2	0,632	9,36	0,031
G	200,686	2	100,343	1485,94	0
w	150,887	2	75,443	100,35	0,001
YİNESA X G	1,32	4	0,660	2,57	0,192
YİNESA X w	0,663	4	0,331	2,73	0,179
G X w	2,477	4	1,238	7,86	0,041
YİNESA X G X w	0,497	8	0,148	1,24	0,380

Geliştirilen sezgisellerin parametre ayarlaması ile ilgili yapılan çalışmalar sonucunda;

- *TABAKHO* sezgiseli için en uygun parametre kombinasyonu; iterasyon sayısı=1000, en büyük tabu listesi uzunluğu=20 ve alternatif liste çözüm sayısı=50 olarak,
- *TAVBEKHO* sezgiseli için en uygun parametre kombinasyonu; başlangıç sıcaklığı=100000, soğutma oranı=0,995 ve durma sıcaklığı=10 olarak,
- *PARSEKHO* sezgiseli için en uygun parametre kombinasyonu; iterasyon sayısı=500, parçacık sayısı=50 ve atalet ağırlığı=0.9 olarak

seçilen parametre kombinasyonlarının en iyi sonuçları ürettiği görülmüştür.

5.6. Karşılaştırmalı Analizler

Algoritmaların etkinlikleri incelenirken performans kriteri olarak sınıf etiketleri önceden bilinen veri yığınlarının kümeleme sonuçlarının karşılaştırılması değerlendirme yöntemlerinden birisidir . Özellikle sınırlı sayıda veriden oluşan veri yığınlarında, geliştirilen yöntemlerin bir çoğu birbirine çok yaklaşık sonuçlar ürettiğinden bu çalışma kapsamında bu şekilde bir değerlendirme tercih edilmemiştir. Ancak örnek olması bakımından irisi veri yığını için farklı yöntemlerin ürettiği sonuçlar aşağıdaki çizelgelerde gösterilmiştir.

Çizelge 5.9. İris veri yığını K-Ortalama kümeleme sonuçları

		KO ile oluşan kümeler (SPSS)			
		1	2	3	Toplam
Gerçek Kümeler	1	50			50
	2		48	2	50
	3		14	36	50
Toplam		50	62	38	150

Çizelge 5.10. İris veri yığını Bulanık K-Ortalama kümeleme sonuçları

		BKO ile oluşan kümeler			
		1	2	3	Toplam
Gerçek Kümeler	1	50			50
	2		47	3	50
	3		13	37	50
Toplam		50	60	40	150

Çizelge 5.11. İris veri yığını K-Harmonik Ortalama kümeleme sonuçları

		KHO ile oluşan kümeler			
		1	2	3	Toplam
Gerçek Kümeler	1	50			50
	2		47	3	50
	3		14	36	50
Toplam		50	61	39	150

Çizelge 5.12. İris veri yığını sezgisel kümeleme sonuçları

		KHO ile oluşan kümeler(st)			Toplam
		1	2	3	
Gerçek Kümeler	1	50			50
	2		46	4	50
	3		14	36	50
Toplam		50	60	40	150

Çizelge 5.3'de iris veri yığını için SPSS programı ile yapılan K-Ortalamaveri kümeleme sonuçları gösterilmektedir. Bu sonuçlara göre oluşan kümelerden birinci kümede 50 eleman, ikinci kümede 62 eleman, üçüncü kümede ise 38 eleman bulunmaktadır.

Çizelge 5.4'de aynı veri yığını için BKO kümeleme sonuçları gösterilmektedir. BKO kümeleme sonuçlarına göre oluşan kümelerden birinci kümede 50 eleman, ikinci kümede 60 eleman, üçüncü kümede ise 40 eleman bulunmaktadır.

Çizelge 5.5'de KHO kümeleme sonuçları gösterilmektedir. BKO kümeleme sonuçlarına göre oluşan kümelerden birinci kümede 50 eleman, ikinci kümede 60 eleman, üçüncü kümede ise 40 eleman bulunmaktadır.

Çizelge 5.6'da geliştirilen sezgisel yöntemlerle KHO kümeleme sonuçları gösterilmektedir. Burada her üç sezgisel de aynı sonuçları ürettiğinden sonuçlar tek çizelge şeklinde verilmiştir. Kümeleme sonuçlarına göre oluşan kümelerden birinci kümede 50 eleman, ikinci kümede 61 eleman, üçüncü kümede ise 39 eleman bulunmaktadır.

Çizelgelerde görüldüğü gibi farklı kümeleme algoritmaları arasında dikkate değer farklar bulunmamaktadır. Burada elde edilen farklı sonuçlar ele alınan veri yığınının özelliğinden kaynaklanmaktadır. İris veri yığını için iki ve üçüncü kümelerin bazı elemanlarının sınırlara çok yakın noktalar veya bizzat sınırlar üzerinde bulunmaları bu sonucu üretmektedir.

Bu çalışmadaki analizlerin tümü Pentium IV 3,2 GHz işlemcili, 512 MB RAM bellekli bilgisayarlar üzerinde MATLAB 7.0 yazılımı ile gerçekleştirilmiştir.

Bu bölümde yapılan analiz çalışmalarında;

- Öncelikle K-Ortalama, bulanık K-Ortalama ve K-Harmonik Ortalama kümeleme yöntemleri K-Ortalama amaç fonksiyon değerlerine göre (katı kümeleme) karşılaştırılarak KHO kümeleme yönteminin başlangıç merkezlerine diğerlerinden daha az duyarlı olduğu ve çoğunlukla daha iyi sonuçlar ürettiği gösterilecek,
- KHO kümeleme ve geliştirilen sezgisel yöntemlerin farklı *p* parametrelerine göre amaç fonksiyonu değeri ve işlemci zamanları karşılaştırılarak en uygun *p* değeri ve en iyi amaç fonksiyon değerini üreten sezgisel yöntem gösterilecek,
- Bulanık K-Ortalama için geliştirilmiş olan küme değerlendirme indekslerinin KHO kümeleme içinde kullanılabileceği farklı performans değerleri ele alınarak sonuçları ile gösterilecektir.

Çizelge 5.13. KO Amaç fonksiyon değerlerine göre sonuçlar (Iris, k=3)

J(U,X)	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	81,317	79,642	80,168	82,196	79,545	80,569
TABAKHO	81,343	79,454	79,061	79,062	79,549	80,400
TAVBEKHO	81,342	79,454	79,061	79,062	79,549	80,327
PARSEKHO	81,296	79,453	79,060	79,062	79,548	80,271

Üç kümeden oluşan iris veri yığını için KO kümeleme analizinde elde edilen sonuçlar 79,097 ile 143,454 arasında yayılmaktadır. Çizelge 5.13'de gösterilen sonuçlara göre *p*=2 değeri için KHO kümelemede 79,642 ile en uygun sonuç elde edilmiş geliştirilen sezgisel yöntemlerde ise *p*=2,3 değeri için 79,061 ve 79,060 ile en iyi sonuçlar elde edilmiştir. Tüm sonuçlar içerisinde ise bütün *p* değerleri için en iyi sonuçlar PARSEKHO kümeleme ile edilmiştir.

Çizelge 5.14. KO Amaç fonksiyon değerlerine göre sonuçlar (Bardak, K =2)

J(U,X)	*p=1.5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	849,136	828,532	824,525	825,587	846,835	922,294
TABAKHO	851,209	828,540	824,523	825,582	846,686	918,330
TAVBEKHO	851,210	828,540	824,523	825,582	846,686	918,330
PARSEKHO	850,411	828,563	824,523	825,582	846,692	917,550

İki kümeden oluşan bardak veri yığını için KO kümeleme analizinde elde edilen sonuçlar 831 ile 936 arasında yayılmaktadır. Çizelge 5.14'de gösterilen sonuçlara göre tüm yöntemlerde p=2.3 değeri için 824,523-824,525 ile en iyi sonuçlar elde edilmiştir. p=1,5 ve p=2 parametreleri için KHO kümeleme, p=2,3 ve p=2,5 için geliştirilen sezgisel yöntemlerin tümü, p=3 değeri için TABAKHO ver TAVBEKHO, p=3,5 değeri içinse PARSEKHO kümeleme sonucunda en iyi değerler elde edilmiştir.

Çizelge 5.15. KO Amaç fonksiyon değerlerine göre sonuçlar (Şarap, K =3)

J(U,X)	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	2686,8e3	2470,2e3	2637,5e3	2676,1e3	2686,8e3	3333,6e3
TABAKHO	2396,8e3	2402,7e3	2438,2e3	2489,1e3	2687,4e3	2733,7e3
TAVBEKHO	2397,9e3	2402,7e3	2438,2e3	2489,2e3	2687,4e3	2705,5e3
PARSEKHO	2445,9e3	2404,5e3	2437,9e3	2487,5e3	2690,5e3	2721,5e3

Üç kümeden oluşan şarap veri yığını için KO kümeleme analizinde elde edilen sonuçlar 2405e3 ile 2636e3 arasında yayılmaktadır. Çizelge 5.15'de gösterilen sonuçlara göre TABAKHO ve TAVBEKHO kümelemede p=1,5 değeri için tüm sonuçlar arasında en iyi sonuçlar elde edilmiştir. p=2 ve p=3 parametresi kullanıldığında TABAKHO ve TAVBEKHO, p=2,3 ve p=2,5 için PSOKHM, p=3,5 içinse TAVBEKHO ile en iyi sonuçlar elde edilmiştir.

Çizelge 5.16. KO Amaç fonksiyon değerlerine göre sonuçlar (Kanser, K =2)

J(U,X)	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	19863	19885	19960	20048	20426	21055
TABAKHO	19864	19885	19960	20048	20426	21053
TAVBEKHO	19864	19885	19960	20048	20426	21053
PARSEKHO	19864	19885	19960	20048	20426	21053

İki kümeden oluşan kanser veri yığını için KO kümeleme analizinde elde edilen sonuçlar 19826 ile 99823 arasında yayılmaktadır. Çizelge 5.16'da görüldüğü gibi geliştirilen sezgisel yöntemler bu veri yığını için p=1,5 ve p=3,5 değerleri dışında KHO ile fark yaratmamaktadır. KHO kümeleme ve geliştirilen diğer sezgisel yöntemler için en iyi sonuçlar p=1,5 değeri kullanıldığında elde edilmiştir. p=1,5 değerinde KHO p=3,5 değerinde ise geliştirilen sezgiseller en iyi sonuçları üretmektedir.

Aynı veri kümeleri normalize edilerek karşılaştırma yapıldığında ise Çizelge 5.17'de gösterilen sonuçlar elde edilmiştir.

Çizelge 5.17. KO Amaç Fonksiyon değerlerine göre karşılaştırma

Veri	*Perf(X, C) (KO)*	*Perf(X, C) (BKO)*	*Perf(X, C) (KHO)*	*Perf(X, C) Sezgiseller*	*cpu (PSE-KHO)*
iris	*7,139*	7,083	7,043	7,042	5,781
şarap	*51,203*	49,955	49,658	49,608	7,251
kanser	*260,294*	218,459	213,368	213,368	40,810
bardak	*34,132*	20,231	16,792	16,729	8,742

Çizelge 5.17'de daha önceden anlatılan yöntemlere bulanık K-Ortalama kümeleme sonuçları da eklenmiştir. Bu analizde KHO ve türevi olan tüm yöntemlerde p=2,3 değeri kullanılmıştır. Elde edilen sonuçlara göre kanser veri kümesi için geliştirilen sezgisel yöntemler KHO kümeleme ile aynı sonucu üretmiş diğer üç veri kümesi için de geliştirilen sezgiseller en iyi sonuçları üretmiştir. Burada dikkat çekilmesi gereken diğer konu KHO kümeleme ve geliştirilen sezgisellerin sadece KO kümelemeye değil aynı zamanda BKO kümelemeye de iyi birer alternatif olduklarıdır. Özellikle şarap ve bardak veri kümelerinde amaç fonksiyon değerlerinde dikkate değer bir düşüş görülmektedir.

Buraya kadar anlatılan dört örnektede görüldüğü gibi KHO kümeleme KO kümelemenin başlangıç merkezlerine karşı duyarlılığını taşımamakla birlikte bir çok noktada aynı performans kriteri ile değerlendirildiğinde K ortalamadan daha iyi sonuçlar üretmiştir.

Bundan sonraki bölümlerde KHO kümeleme ve geliştirilen sezgisel yöntemlerin farklı *p* paramatreleri için KHO performans kriterine ve küme değerlendirme indekslerine göre karşılaştırmaları yapılmakta aynı zamanda işlem zamanları da gösterilmektedir.

Çizelge 5.18. KHO Performans kriterine göre sonuçlar (Iris, K =3)

J(U,X)	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	182,331	181,795	184,629	183,243	186,827	194,182
TABAKHO	182,293	181,728	182,252	183,037	186,827	193,637
TAVBEKHO	182,293	181,728	182,252	183,037	186,827	193,566
PARSEKHO	182,294	181,728	182,253	183,037	186,827	193,591

Çizelge 5.18'de üç kümeden oluşan iris veri yığını için yapılan analiz çalışmasının KHO performans sonuçları görülmektedir. Bu çalışma sonucunda KHO performans kriterine göre en iyi sonuçlar *p=2* değeri ile edilmiştir. *p=2* parametre değeri kullanıldığında KHO yönteminin 181,795 değerine karşılık, geliştirilen sezgisellerin tümü 181,728 değeri ile daha iyi bir değer üretmektedirler. TAVBEKHO yöntemi diğer yöntemlerle karşılaştırıldığında tüm *p* değerleri için en iyi sonucu üretmektedir. *p=3* değerinde KHO ve diğer sezgiseller aynı sonucu üretmektedir.

Çizelge 5.19. İşlemci zamanları (Iris, K =3)

cpu	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	0,047	0,062	0,062	0,078	0,031	0,078
TABAKHO	0,764	0,375	0,764	0,764	0,374	0,378
TAVBEKHO	1,574	1,013	1,543	1,511	0,951	1,544
PARSEKHO	0,592	0,358	0,640	0,654	0,358	0,670

Çizelge 5.19'da üç kümeden oluşan iris veri yığını için yapılan analiz çalışmasının işlem zamanları görülmektedir. KHO kümeleme yöntemi doğal olarak en hızlı

yöntem olmakla birlikte geliştirilen sezgisel yöntemlerin en hızlısı olarak PARSEKHO yöntemi dikkat çekmektedir. TABAKHO yöntemi işlem zamanı olarak PARSEKHO ya yakın zamana sahiptir. En yavaş çalışan yöntem ise TAVBEKHO yöntemi olmuştur.

Çizelge 5.20. Geçerlilik Analizi (Iris, K =3)

METOD		*J(PC)*	*J(PE)*	*J(FS)*	*J(XB)*
KHO	p=1,5	0,931	0,280	-174,287	0,776
	p=2,0	0,914	0,221	-180,438	0,711
	p=2,3	0,896	0,215	-182,739	0,692
	p=2,5	0,910	0,180	-183,956	0,666
	p=3,0	0,917	0,145	-185,034	0,644
	p=3,5	0,919	0,125	-185,598	0,643
TABAKHO	p=1,5	0,930	0,282	-175,371	0,769
	p=2,0	0,914	0,219	-180,666	0,706
	p=2,3	0,914	0,190	-182,868	0,678
	p=2,5	0,915	0,174	-183,880	0,664
	p=3,0	0,917	0,145	-185,257	0,646
	p=3,5	0,919	0,125	-184,804	0,648
TAVBEKHO	p=1,5	0,930	0,281	-175,382	0,769
	p=2,0	0,914	0,219	-180,666	0,706
	p=2,3	0,914	0,190	-182,868	0,678
	p=2,5	0,915	0,174	-183,880	0,664
	p=3,0	0,917	0,145	-185,257	0,646
	p=3,5	0,919	0,125	-185,725	0,642
PARSEKHO	p=1,5	0,930	0,281	-175,455	0,770
	p=2,0	0,914	0,219	-180,651	0,706
	p=2,3	0,914	0,190	-182,872	0,678
	p=2,5	0,915	0,174	-183,880	0,664
	p=3,0	0,917	0,145	-185,256	0,644
	p=3,5	0,919	0,125	-185,864	0,644

Çizelge 5.20'de üç kümeden oluşan iris veri yığını için yapılan analiz çalışmasının farklı p parametre değerleri için değerlendirme indeks değerleri gösterilmektedir. KHO performans kriterine göre bu veri yığını ve *K=3* ve *p=2* değeri için elde edilen sonuçların en iyi sonuçlar olduğu Çizelge 5.18'de gösterilmişti. Geçerlilik indekslerinde *p=2* ye karşılık gelen değerlere bakıldığında PC indeksi için KHO ve geliştirilen sezgiseller 0,914 değeri ile aynı performansı göstermektedirler. PE indeksi için KHO 0,221 değerine sahipken, geliştirilen sezgiseller 0,219 ile, XB

indeksine göre ise yine KHO 0,711 iken geliştirilen sezgiseller 0,706 ile daha iyi bir performans sergilemişlerdir. FS indeksine göre ise KHO -180,438 ile tüm yöntemler arasında en iyi sonucu üretmiştir. KHO performans kriteri dikkate alınmaksızın sadece değerlendirme indeksleri ile yapılacak olan bir karşılaştırmada PC ve FS indeksine göre p=1,5 değeri ve KHO yöntemi en iyi sonucu verirken, XB indeksine göre p=3,5 ve TAVBEKHO yöntemi en iyi sonucu vermekte, PE indeksine göre p=3,5 değeri ile tüm yöntemler aynı sonucu vermektedir.

Çizelge 5.21. KHO Performans kriterine göre sonuçlar (İris, K =2)

J(U,X)	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	221,741	257,847	287,744	311,748	387,983	512,650
TABAKHO	221,740	257,847	287,744	311,748	387,983	495,511
TAVBEKHO	221,740	257,847	287,744	311,748	387,983	490,842
PARSEKHO	221,740	257,847	287,744	311,748	387,983	490,847

Çizelge 5.21'de iki kümeden oluşan iris veri yığını için yapılan analiz çalışmasının KHO performans sonuçları görülmektedir. Bu çalışma sonucunda KHO performans kriterine göre en iyi sonuçlar *p=1,5* değeri ile edilmiştir. *p=1,5* parametre değeri kullanıldığında KHO yönteminin 221,741 değerine karşılık, geliştirilen sezgisellerin tümü 221,740 değeri ile daha iyi bir değer üretmektedirler. TAVBEKHO yöntemi diğer yöntemlerle karşılaştırıldığında tüm *p* değerleri için en iyi sonucu üretmektedir. *p=3* değerinde KHO ve diğer sezgiseller aynı sonucu üretmektedir. p parametresinin 2 ve 3 aralığındaki değerleri için tüm yöntemler aynı sonucu vermektedir.

Çizelge 5.22. İşlemci zamanları (İris, K =2)

cpu	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	0,484	0,156	0,125	0,125	0,109	0,125
TABAKHO	0,672	0,484	0,672	0,625	0,484	0,672
TAVBEKHO	1,938	0,594	0,859	0,875	0,578	0,906
PARSEKHO	0,703	0,438	0,391	0,422	0,313	0,500

Çizelge 5.22'de iki kümeden oluşan iris veri yığını için yapılan analiz çalışmasının işlem zamanları görülmektedir. KHO kümeleme yöntemi en hızlı yöntem olmakla

birlikte geliştirilen sezgisel yöntemlerin en hızlısı olarak PARSEKHO yöntemi dikkat çekmektedir. TABAKHO yöntemi işlem zamanı olarak PARSEKHO ya yakın zamana sahiptir. En yavaş çalışan yöntem ise TAVBEKHO yöntemi olmuştur.

Çizelge 5.23. Geçerlilik Analizi (iris, K =2)

METOD		*J(PC)*	*J(PE)*	*J(FS)*	*J(XB)*
KHO	p=1,5	0,975	0,108	-144,640	0,371
	p=2,0	0,966	0,090	-150,531	0,432
	p=2,3	0,962	0,084	-150,095	0,485
	p=2,5	0,960	0,081	-148,484	0,528
	p=3,0	0,954	0,078	-139,965	0,672
	p=3,5	0,942	0,085	-107,734	0,987
TABAKHO	p=1,5	0,975	0,108	-150,837	0,371
	p=2,0	0,966	0,090	-150,531	0,432
	p=2,3	0,962	0,084	-148,455	0,485
	p=2,5	0,960	0,081	-146,074	0,528
	p=3,0	0,954	0,078	-136,297	0,672
	p=3,5	0,945	0,081	-108,563	0,922
TAVBEKHO	p=1,5	0,975	0,108	-150,837	0,371
	p=2,0	0,966	0,090	-150,531	0,432
	p=2,3	0,962	0,084	-148,455	0,485
	p=2,5	0,960	0,081	-146,074	0,528
	p=3,0	0,954	0,078	-136,297	0,672
	p=3,5	0,948	0,078	-123,395	0,870
PARSEKHO	p=1,5	0,975	0,108	-150,837	0,371
	p=2,0	0,966	0,090	-150,531	0,432
	p=2,3	0,962	0,084	-148,455	0,485
	p=2,5	0,960	0,081	-146,074	0,528
	p=3,0	0,954	0,078	-136,297	0,672
	p=3,5	0,948	0,078	-123,099	0,872

Çizelge 5.23'de iki kümeden oluşan iris veri yığını için yapılan analiz çalışmasının farklı p parametre değerleri için değerlendirme indeks değerleri gösterilmektedir. KHO performans kriterine göre bu veri yığını ve *K=2* ve *p=1,5* değeri için elde edilen sonuçların en iyi sonuçlar olduğu Çizelge 5-15 de gösterilmişti. Geçerlilik indekslerinde *p=1,5* a karşılık gelen değerlere bakıldığında PC indeksi için 0,975, PE için 0,108, XB içinse 0,371 değerleri ile KHO ve geliştirilen sezgiseller aynı performansı göstermektedirler. FS indeksi için KHO -144,640 değerine sahipken, geliştirilen sezgiseller -150,837 ile daha iyi bir performans sergilemişlerdir. KHO

performans kriteri dikkate alınmaksızın sadece değerlendirme indeksleri ile yapılacak olan bir karşılaştırmada PC, FS ve XB indekslerine göre *p=1,5* değerinde tüm yöntemler en iyi sonucu vermekte, PE indeksine göre *p=3* değerinde tüm yöntemler, ilave olarak *p=3,5* değerinde TAVBEKHO ve PARSEKHO en iyi sonucu vermektedir.

Çizelge 5.24. KHO Performans kriterine göre sonuçlar (Bardak, *K* =2)

J(U,X)	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	645,541	1112,800	1616,300	2105,100	4247,900	8870,500
TABAKHO	642,900	1112,800	1616,300	2105,100	4247,900	8870,500
TAVBEKHO	642,871	1112,800	1616,300	2105,100	4247,900	8870,500
PARSEKHO	642,894	1112,800	1616,300	2105,100	4247,900	8870.500

Çizelge 5.24'de iki kümeden oluşan bardak veri yığını için yapılan analiz çalışmasının KHO performans sonuçları görülmektedir. Bu çalışma sonucunda KHO performans kriterine göre en iyi sonuçlar *p=1,5* değeri ile edilmiştir. *p=1,5* parametre değeri kullanıldığında TAVBEKHO 642,871 ile en iyi sonucu vermektedir. KHO yöntemi 645,541 ile en kötü sonucu verirken *p=1,5* değerinde diğer tüm sezgiseller KHO yönteminden daha iyi sonuçlar vermişlerdir. *p* nin 1,5 dan büyük değerleri için her dört yöntem de aynı sonucu vermekte, geliştirilen sezgiseller daha iyi bir sonuç üretememektedir.

Çizelge 5.25. İşlemci zamanları (Bardak, *K* =2)

cpu	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	0,422	0,109	0,172	0,156	0,063	0,172
TABAKHO	1,234	0,906	1,125	1,109	0,859	1,188
TAVBEKHO	2,297	1,516	1,781	1,844	1,469	1,859
PARSEKHO	0,828	0,719	0,906	0,922	0,766	0,922

Çizelge 5.25'de iki kümeden oluşan bardak veri yığını için yapılan analiz çalışmasının işlem zamanları görülmektedir. Buradaki sonuçlarda bundan önceki veri yığınları ile yapılan çalışmalarda varılan sonuçları doğrulamaktadır.

Çizelge 5.26. Geçerlilik Analizi (Bardak, K =2)

METOD		*J(PC)*	*J(PE)*	*J(FS)*	*J(XB)*
KHO	p=1,5	0,927	0,286	428,220	0,966
	p=2,0	0,909	0,218	259,728	1,529
	p=2,3	0,903	0,200	219,878	2,205
	p=2,5	0,900	0,191	205,497	2,862
	p=3,0	0,893	0,176	201,961	5,721
	p=3,5	0,868	0,186	229,398	11,551
TABAKHO	p=1,5	0,934	0,257	265,041	0,901
	p=2,0	0,910	0,218	259,775	1,529
	p=2,3	0,903	0,200	264,515	2,205
	p=2,5	0,900	0,191	271,201	2,862
	p=3,0	0,893	0,176	302,675	5,720
	p=3,5	0,867	0,186	362,922	11,547
TAVBEKHO	p=1,5	0,934	0,256	265,041	0,901
	p=2,0	0,910	0,218	259,775	1,529
	p=2,3	0,903	0,200	264,515	2,205
	p=2,5	0,900	0,191	271,201	2,862
	p=3,0	0,893	0,176	302,675	5,720
	p=3,5	0,867	0,186	362,922	11,547
PARSEKHO	p=1,5	0,934	0,258	266,458	0,904
	p=2,0	0,910	0,218	259,802	1,529
	p=2,3	0,903	0,200	264,515	2,205
	p=2,5	0,900	0,191	271,205	2,862
	p=3,0	0,893	0,176	302,678	5,720
	p=3,5	0,868	0,186	362,297	11,556

Çizelge 5.26'da iki kümeden oluşan bardak veri yığını için yapılan analiz çalışmasının farklı p parametre değerleri için değerlendirme indeks değerleri gösterilmektedir. KHO performans kriterine göre bu veri yığını ve $K=2$ ve $p=1,5$ değeri için elde edilen sonuçların en iyi sonuçlar olduğu Çizelge 5-18 de gösterilmişti. Geçerlilik indekslerinde $p=1,5$ a karşılık gelen değerlere bakıldığında PC indeksi için 0,934 değeri ile geliştirilen sezgiseller KHO yönteminin 0,927 değerinden daha iyi sonuç vermiştir. PE indeksi için en iyi değeri 0,256 ile TAVBEKHO yöntemi, FS indeksi için 265,041 ile TABAKHO ve TAVBEKHO, XB indeksi içinse 0,901 değeri ile TABAKHO ve TAVBEKHO en iyi sonuçları vermektedir. KHO performans kriteri dikkate alınmaksızın sadece değerlendirme indeksleri ile yapılacak olan bir karşılaştırmada PC indeksine göre p=1,5 değerinde geliştirilen sezgisellerin tümü, PE indeksine göre KHO dahil olmak üzere tüm

yöntemler 0,176 ile, FS indeksine göre *p=3* değerinde KHO yöntemi 201,961 ile, XB indeksine göre ise TABAKHO ve TAVBEKHO 0,901 ile en iyi sonuçları vermektedir.

Çizelge 5.27. KHO Performans kriterine göre sonuçlar (Şarap, *K* =3)

J(U,X)	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	0,40867e6	5,3926e6	26,224e6	75,98e6	1058,9e6	17092e6
TABAKHO	0,40756e6	8,7317e6	26,216e6	75,84e6	1058,8e6	14340e6
TAVBEKHO	0,39936e6	5,3882e6	26,216e6	75,84e6	1058,8e6	15001e6
PARSEKHO	0,40046e6	5,3884e6	26,217e6	75,84e6	1059,0e6	14592e6

Çizelge 5.27'de üç kümeden oluşan şarap veri yığını için yapılan analiz çalışmasının KHO performans sonuçları görülmektedir. Bu çalışma sonucunda KHO performans kriterine göre en iyi sonuçlar *p=1,5* değeri ile edilmiştir. *p=1,5* parametre değeri kullanıldığında TAVBEKHO 0,39936e6 ile en iyi sonucu vermektedir. KHO yöntemi 0,40867e6 ile en kötü sonucu verirken *p=1,5* değerinde diğer tüm sezgiseller KHO yönteminden daha iyi sonuçlar vermişlerdir. *p=2* değeri için TAVBEKHO 5,3882e6 ile, *p=3,5* değeri için 14340e6 ile TABAKHO, *p=2,3* ve *p=3* değerleri için TABAKHO ve TAVBEKHO 26,216e6 ve 1058,8e6 ile, p=2,5 değeri için ise her üç sezgisel 75,84e6 ile en iyi sonuçları vermektedir.

Çizelge 5.28. İşlemci zamanları (Şarap, *K* =3)

cpu	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	0,391	0,203	0,219	0,188	0,156	0,188
TABAKHO	1,109	0,625	1,078	1,016	0,703	1,016
TAVBEKHO	2,188	0,984	1,375	1,359	1,000	1,516
PARSEKHO	0,844	0,516	0,640	0,703	0,516	0,750

Çizelge 5.28'de üç kümeden oluşan şarap veri yığını için yapılan analiz çalışmasının işlem zamanları görülmektedir. Buradaki sonuçlarda bundan önceki veri yığınları ile yapılan çalışmalarda varılan sonuçları doğrulamaktadır.

Çizelge 5.29. Geçerlilik Analizi (Şarap, K =3)

METOD		*J(PC)*	*J(PE)*	*J(FS)*	*J(XB)*
KHO	p=1,5	0,942	0244	2517,5e3	9,746
	p=2,0	0912	0,223	1981,5e3	105,192
	p=2,3	0,903	0,206	1923,9e3	494,922
	p=2,5	0,879	0,225	1869,5e3	13399,000
	p=3,0	0,889	0,180	2009,2e3	14907,000
	p=3,5	0,875	0,183	2423,5e3	375330,000
TABAKHO	p=1,5	0,942	0,243	1975,0e3	9,081
	p=2,0	0,916	0,215	1985,4e3	106,829
	p=2,3	0,900	0,211	2005,9e3	490,666
	p=2,5	0,888	0,212	2023,6e3	1368,800
	p=3,0	0,888	0,182	2239,4e3	14932,000
	p=3,5	0,901	0,159	2336,7e3	196360,000
TAVBEKHO	p=1,5	0,942	0,243	1975,4e3	9,085
	p=2,0	0,916	0,215	1985,4e3	106,825
	p=2,3	0,900	0,211	2005,9e3	490,666
	p=2,5	0,888	0,212	2022,5e3	1370,300
	p=3,0	0,888	0,182	2239,4e3	14930,000
	p=3,5	0,885	0,164	2371,0e3	229800,000
PARSEKHO	p=1,5	0,943	0,241	1982,3e3	9,034
	p=2,0	0,917	0,214	1985,9e3	107,021
	p=2,3	0,901	0,210	2006,1e3	492,025
	p=2,5	0,889	0,211	2022,3e3	1374,000
	p=3,0	0,889	0,180	2242,1e3	14847,000
	p=3,5	0,895	0,153	2360,2e3	207120,000

Çizelge 5.29'da üç kümeden oluşan şarap veri yığını için yapılan analiz çalışmasının farklı p parametre değerleri için değerlendirme indeks değerleri gösterilmektedir. KHO performans kriterine göre bu veri yığını ve $K=3$ ve $p=1,5$ değeri için elde edilen sonuçların en iyi sonuçlar olduğu Çizelge 5.27'de gösterilmişti. Geçerlilik indekslerinde $p=1,5$ a karşılık gelen değerlere bakıldığında PC indeksi için 0.943 değeri ile PARSEKHO, geliştirilen sezgiseller ve KHO yönteminden daha iyi sonuç vermiştir. PE indeksi için en iyi değeri 0,241 ile yine PARSEKHO yöntemi, FS indeksi için 1975,4e3 ile TABAKHO, XB indeksi içinse 9,034 değeri ile PARSEKHO en iyi sonuçları vermektedir. KHO performans kriteri dikkate alınmaksızın sadece değerlendirme indeksleri ile yapılacak olan bir karşılaştırmada PC indeksine göre $p=1,5$ değerinde PARSEKHO dışında geliştirilen sezgisellerin tümü, PE indeksine göre $p=3$,5 değerinde PARSEKHO 0,153 ile, FS indeksine göre

p=1,5 değerinde TABAKHO yöntemi 1975,0e3 ile, XB indeksine göre ise yine PARSEKHO 9,034 ile en iyi sonuçları vermektedir.

Çizelge 5.30. KHO Performans kriterine göre sonuçlar (Kanser, K =2)

J(U,X)	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	11195	30615	57034	86830	251670	738610
TABAKHO	11195	30615	57034	86830	251670	738610
TAVBEKHO	11195	30615	57034	86830	251670	738610
PARSEKHO	11195	30615	57034	86830	251670	738600

Çizelge 5.30'da iki kümeden oluşan kanser veri yığını için yapılan analiz çalışmasının KHO performans sonuçları görülmektedir. Bu çalışma sonucunda KHO performans kriterine göre en iyi sonuçlar *p=1,5* değeri ile edilmiştir. *p=1,5* parametre değeri kullanıldığında KHO dahil tüm yöntemler 11195 ile en iyi sonucu vermektedir. Bu veri yığını için *p* parametresinin *1,5* ve *3* arasındaki değerleri için tüm yöntemler birbiriyle aynı sonucu vermektedir. *p=3,5* değerinde ise en iyi sonucu 738600 ile PARSEKHO yöntemi vermekte bunun dışındaki tüm yöntemler aynı sonucu vermektedir.

Çizelge 5.31. İşlemci zamanları (Kanser, K =2)

cpu	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	0,221	0,190	0,220	0,252	0,188	0,204
TABAKHO	4,696	3,622	4,672	4,688	3,591	4,688
TAVBEKHO	8,355	5,844	7,583	7,630	5,875	7,613
PARSEKHO	3,430	2,569	3,509	3,399	2,632	3,430

Çizelge 5.31'de iki kümeden oluşan kanser veri yığını için yapılan analiz çalışmasının işlem zamanları görülmektedir. Bu veri yığınında KHO yöntemi diğer veri yığınlarında olduğu gibi en hızlı sonuç veren yöntemdir. Ancak geliştirilen sezgisel yöntemlerin tümünün işlem zamanlarında dikkate değer bir artış gözlenmektedir.

Çizelge 5.32. Geçerlilik Analizi (Kanser, K =2)

METOD		*J(PC)*	*J(PE)*	*J(FS)*	*J(XB)*
KHO	p=1,5	0,952	0,195	13970	1,132
	p=2,0	0,933	0,168	12050	3,099
	p=2,3	0,927	0,156	11642	5,829
	p=2,5	0,924	0,150	11530	8,952
	p=3,0	0,917	0,138	11633	26,656
	p=3,5	0,912	0,123	12136	80,601
TABAKHO	p=1,5	0,952	0,195	11806	1,132
	p=2,0	0,933	0,168	12050	3,099
	p=2,3	0,927	0,156	12293	5,829
	p=2,5	0,924	0,150	12497	8,952
	p=3,0	0,917	0,138	13170	26,655
	p=3,5	0,912	0,130	14073	80,599
TAVBEKHO	p=1,5	0,952	0,195	11806	1,132
	p=2,0	0,933	0,168	12050	3,099
	p=2,3	0,927	0,156	12293	5,829
	p=2,5	0,924	0,150	12497	8,952
	p=3,0	0,917	0,138	13170	26,655
	p=3,5	0,912	0,130	14073	80,599
PARSEKHO	p=1,5	0,952	0,195	11806	1,132
	p=2,0	0,933	0,168	12050	3,099
	p=2,3	0,927	0,156	12293	5,829
	p=2,5	0,924	0,150	12497	8,952
	p=3,0	0,917	0,138	13170	26,655
	p=3,5	0,912	0,130	14075	80,611

Çizelge 5.32' de iki kümeden oluşan kanser veri yığını için yapılan analiz çalışmasının farklı p parametre değerleri için değerlendirme indeks değerleri gösterilmektedir. KHO performans kriterine göre bu veri yığını ve $K=2$ ve $p=1,5$ değeri için elde edilen sonuçların en iyi sonuçlar olduğu Çizelge 5.30'da gösterilmişti. Geçerlilik indekslerinde $p=1,5$ a karşılık gelen değerlere bakıldığında PC indeksi için 0,952 değeri ile, PE indeksi için 0,195 değeri ile XB indeksi içinse 1,132 değeri ile tüm yöntemler aynı performansı sergilemişlerdir. FS indeksi içinse geliştirilen sezgisellerin tümü 11806 ile 13970 değerine sahip olan KHO yönteminden daha iyi performans göstermişlerdir. Performans değerlendirmesi KHO kriteri dikkate alınmadan sadece değerlendirme indeksleri ile yapıldığında ise PC ve XB indeksine göre $p=1,5$ değerinde tüm yöntemler aynı sonucu vermektedir. PE indeksine $p=3,5$ değerinde KHO yöntemi 0,123 ile geliştirilen sezgisellerden daha

iyi sonuç vermiştir. FS indeksine göre yine KHO yöntemi 11633 ile en iyi sonucu vermektedir.

Çizelge 5.33. KHO Performans kriterine göre sonuçlar (Kanser, K =3)

J(U,X)	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	11757	29561	53240	79666	222500	633340
TABAKHO	11305	29394	53240	79666	222460	633300
TAVBEKHO	11305	29394	53235	79664	222460	633300
PARSEKHO	11306	29401	53235	79670	222460	633300

Çizelge 5.33'de üç kümeden oluşan kanser veri yığını için yapılan analiz çalışmasının KHO performans sonuçları görülmektedir. Bu çalışma sonucunda KHO performans kriterine göre en iyi sonuçlar *p=1,5* değeri ile edilmiştir. *p=1,5* parametre değeri kullanıldığında TAVBEKHO ve TABAKHO 11305 ile en iyi sonucu vermektedir. KHO yöntemi 11757 ile en kötü sonucu verirken *p=1,5* değerinde diğer tüm sezgiseller KHO yönteminden daha iyi sonuçlar vermişlerdir. *p=2* değeri için TAVBEKHO ve TABAKHO 29394 ile, *p=2,3* değeri için 53235 ile TAVBEKHO PARSEKHO, *p=2,5* için TAVBEKHO 79664 ile, *p=3* ve *p=3,5* değerlerii için ise her üç sezgisel 222460 ve 633300 ile en iyi sonuçları vermektedir.

Çizelge 5.34. İşlemci zamanları (Kanser, K =3)

cpu	*p=1,5*	*p=2,0*	*p=2,3*	*p=2,5*	*p=3,0*	*p=3,5*
KHO	0,262	0,185	0,278	0,233	0,202	0,264
TABAKHO	5,666	3,957	5,608	5,593	4,066	5,577
TAVBEKHO	10,240	6,250	8,821	8,884	6,421	8,884
PARSEKHO	4,131	2,946	4,115	4,130	2,946	4,161

Çizelge 5.34'de üç kümeden oluşan kanser veri yığını için yapılan analiz çalışmasının işlem zamanları görülmektedir. Bu veri yığınında KHO yöntemi diğer veri yığınlarında olduğu gibi en hızlı sonuç veren yöntemdir. Ancak aynı veri yığını ve k=2 de olduğu gibi geliştirilen sezgisel yöntemlerin tümünün işlem zamanlarında dikkate değer bir artış gözlenmektedir.

Çizelge 5.35'de üç kümeden oluşan kanser veri yığını için yapılan analiz çalışmasının farklı *p* parametre değerleri için değerlendirme indeks değerleri gösterilmektedir. KHO performans kriterine göre *K=3* ve *p=1,5* değeri için elde edilen sonuçların en iyi sonuçlar olduğu Çizelge 5.33'de gösterilmişti. Geçerlilik indekslerinde *p=1,5* a karşılık gelen değerlere bakıldığında PC indeksi için 0,899, FS indeksi için 5901,5 değeri ile PARSEKHO, PE indeksi için 0,509 değeri ile XB indeksi içinse 6,580 değeri ile geliştirlen sezgiseller en iyi performansı sergilemişlerdir. Performans değerlendirmesi KHO kriteri dikkate alınmadan sadece değerlendirme indeksleri ile yapıldığında ise PC indeksine göre *p=1,5* değerinde PARSEKHO yöntemi 0,899 ile, PE indeksine göre *p=3,5* değerinde geliştirilen sezgiseller 0,303 ile, FS indeksine göre *p=3* KHO yöntemi 4086,6 ile XB indeksine göre ise *p=1,5* değerinde geliştirilen sezgiseller 6,580 ile en iyi sonucu vermişlerdir.

Çizelge 5.35. Geçerlilik analizi (Kanser, *K*=3)

METOD		*J(PC)*	*J(PE)*	*J(FS)*	*J(XB)*
KHO	p=1,5	0,831	0,667	9684,5	6,842
	p=2,0	0,832	0,411	6219,3	9,066
	p=2,3	0,809	0,393	4997,2	17,567
	p=2,5	0,804	0,375	4579,5	25,504
	p=3,0	0,796	0,339	4086,6	60,856
	p=3,5	0,798	0,304	4193,7	157,153
TABAKHO	p=1,5	0,866	0,509	5901,6	6,580
	p=2,0	0,815	0,437	5976,0	13,308
	p=2,3	0,806	0,398	6173,5	18,354
	p=2,5	0,803	0,376	6331,9	24,762
	p=3,0	0,799	0,334	6807,5	59,503
	p=3,5	0,798	0,303	7402,9	156,803
TAVBEKHO	p=1,5	0,866	0,509	5901,6	6,580
	p=2,0	0,815	0,437	5976,6	13,322
	p=2,3	0,806	0,398	6173,7	18,356
	p=2,5	0,803	0,376	6331,9	24,762
	p=3,0	0,799	0,334	6807,5	59,503
	p=3,5	0,798	0,303	7402,9	156,803
PARSEKHO	p=1,5	0,899	0,509	5901,5	6,580
	p=2,0	0,821	0,427	6012,5	11,273
	p=2,3	0,806	0,398	6174,1	18,324
	p=2,5	0,804	0,375	6338,3	24,345
	p=3,0	0,799	0,334	6806,7	59,612
	p=3,5	0,798	0,303	7403,5	156,821

Çizelge 5.36'da başlangıçta tanımı ve içeriği verilen rasgele veri yığını ile yapılan analiz sonuçları K-Ortalama performans değerine göre verilmektedir. Bu veri yığınında K=2 değeri için en iyi performans değeri 83,889 ile KO yöntemi ile elde edilmesine rağmen 140,200 gibi bir sonuç ta yine KO yönteminden elde edilmiştir. KHO yöntemi K=3 ve K=4 değerleri için KO yönteminin en iyi değerlerinden daha iyi değerler üretmesine rağmen yine de yerel en iyi tuzağına takıldığı sonuçlardan görülmektedir. TABAKHO yöntemi K=4 değeri için iki değer aralığında sonuç vermekte diğer k değerleri için TAVBEKHO ve PARSEKHO gibi her seferinde aynı sonucu vermektedir. Geliştirilen sezgiseller burada ve diğer örneklerde de görüldüğü gibi ister KO performans kriterine göre, ister KHO performans kriterine, isterse küme değerlendirme indekslerine göre değerlendirilsin karşılaştırıldıkları diğer yöntemlerden daha tutarlı sonuçlar üretmektedir.

Çizelge 5.36. KO Performans kriterine sonuçlar (rasgele veri yığını)

K	KO	KHO	TABAKHO	TAVBEKHO	PARSEKHO
2	83,889-140,200	86,543-87,318	85,329	85,329	85,329
3	35,167 -75,792	33,079-33,126	33,074	33,074	33,074
4	24,133-30,667	22,155-24,589	22,161-24,567	22,161	22,161

Çizelge 5.37. Okul veri yığını merkez koordinatları ve performans değerleri

Yöntem	Küme merkezleri					Amaç fonksiyonu
K-Ortalama	8,682	8,392	6,727	7,316	10,329	168,214
	6,480	7,513	12,090	10,495	11,539	
	13,902	13,871	13,032	13,153	9,644	
Bulanık k-Ortalama	8,683	8,392	6,727	7,316	10,329	168,213
	6,480	7,513	12,080	10,494	11,538	
	13,901	13,870	13,033	13,153	9,644	
Sezgisel destekli K-Harmonik Ortalama	8,587	8,290	6,533	7,155	10,245	162,059
	6,838	7,811	12,219	10,741	12,280	
	13,920	13,879	13,174	13,244	9,547	

Çizelge 5.37'de başlangıçta tanıtılan okul veri kümesinin kümeleme analizi sonuçları küme merkez koordinatları ve K-Ortalama amaç fonksiyon değerleri ile gösterilmektedir. Çizelgede görüldüğü gibi geliştirlen sezgisel yöntemler 162,059 ile hem KO yönteminden hem de BKO yönteminden çok daha iyi sonuçlar vermektedir.

Küme merkez koordinatları incelendiğinde ise çok büyük farklar bulunmadığı ve her bir yöntemde elde edilen merkezlerin birbirlerine oldukça yakın değerler içerdiği görülmektedir.

Çizelge 5.38'de yerleşim yerleri veri kümesinin KO, BKO ve PARSEKHO yöntemleri ile kümelenmesiyle elde edilen kümeleme sonuçları gösterilmektedir. A ile başlayan sütunlarda veriler normalize edilmeden elde edilen sonuçlar B ile başlayan sütunlarda ise veriler normalize edildikten sonra elde edilen sonuçlar gösterilmiştir. Çizelgede görüldüğü gibi farklı kümeleme analizleri sonucunda farklı sonuçlar elde edilebilmektedir. Bu çalışmanın başlangıcında tüm veri yığınlarında ve her türlü probleme uygun genel geçerli kümeleme yöntemlerinin olmadığı anlatılmıştı. Yerleşim yerlerinin kümelenmesi örneğinde bu durum açıkça görülmektedir.

Çizelge 5.38. Yerleşim veri kümeleme sonuçları

İl	İlçe	A(PSE)	B(PSE)	A(KO)	B(KO)	A(BKO)	B(BKO)
Adıyaman	Merkez	1	1	1	2	3	2
Afyon	Merkez	3	2	3	3	1	3
Amasya	Merkez	3	1	3	3	1	2
Artvin	Merkez	3	1	3	2	1	2
Aydın	Merkez	3	3	1	3	1	3
Balıkesir	Merkez	1	3	1	3	3	3
Batman	Merkez	1	1	1	2	3	2
Bingöl	Merkez	3	1	3	2	1	2
Bolu	Merkez	3	3	3	3	1	1
Burdur	Merkez	3	1	3	3	1	2
Çanakkale	Merkez	3	3	3	3	1	1
Çorum	Merkez	1	3	1	3	3	1
Denizli	Merkez	1	2	2	1	3	3
Edirne	Merkez	3	3	3	3	1	1
Elazığ	Merkez	1	2	2	1	3	3
Erzincan	Merkez	3	1	3	2	1	2
Giresun	Merkez	3	3	3	3	1	1
Hakkari	Merkez	3	1	3	2	1	2
Isparta	Merkez	3	3	1	3	1	3
Kmaraş	Merkez	2	2	2	1	2	3
Kars	Merkez	3	1	3	2	1	2
Kırıkkale	Merkez	1	3	1	3	3	1
Kırşehir	Merkez	3	1	3	3	1	2
Malatya	Merkez	2	2	2	1	2	3
Manisa	Merkez	1	2	1	3	3	3
Muğla	Merkez	3	3	3	3	1	1
Nevşehir	Merkez	3	1	3	3	1	2
Niğde	Merkez	3	3	3	3	1	2
Ordu	Merkez	3	3	3	3	1	1
Rize	Merkez	3	1	3	3	1	2
Siirt	Merkez	3	1	3	2	1	2
Sinop	Merkez	3	1	3	3	1	2
Sivas	Merkez	1	2	1	1	3	3
Şanlıurfa	Merkez	2	2	2	1	2	3
Tekirdağ	Merkez	3	3	3	3	1	1
Tokat	Merkez	3	3	3	3	1	1
Trabzon	Merkez	1	2	1	1	3	3
Uşak	Merkez	3	3	3	3	1	1
Van	Merkez	1	2	2	1	3	3
Yozgat	Merkez	3	3	3	3	1	2
Zonguldak	Merkez	3	3	3	3	1	1

Çizelge 5.39'da her bir analiz sonucunda ortaya çıkan kümelerin eleman sayıları verilmektedir. Görüldüğü üzere en tutarlı ve anlamlı sonuçlar veri normalize edilerek yapılan Bulanık K-Ortalama ve parçacık sürü en iyilemesi K-Harmonik ortalamada elde edilmiştir.

Çizelge 5.39. Yerleşim veri kümeleme sonuçları (mevcutlar)

	Küme 1	Küme 2	Küme 3
A (PARSEKHO)	10	3	28
B (PARSEKHO)	14	10	17
A (KO)	10	6	25
B (KO)	8	8	25
A (BKO)	27	3	11
B(BKO)	12	16	13

Çizelge 5.40'da veriler normalize edildiğinde *PARSEKHO* sezgiselinе göre oluşan sonuçlar Çizelge 5.41'de ise veriler normalize edildiğinde BKO yöntemine göre oluşan sonuçlar verilmektedir.

Çizelge 5.40. *PARSEKHO*'ya göre yerleşim veri kümeleme sonuçları açıklaması

Küme 1	Küme 2	Küme 3
Adıyaman	Afyon	Aydın
Amasya	Denizli	Balıkesir
Artvin	Elazığ	Bolu
Batman	K.Maraş	Çanakkale
Bingöl	Malatya	Çorum
Burdur	Manisa	Edirne
Erzincan	Sivas	Giresun
Hakkari	Urfa	Isparta
Kars	Trabzon	Kırıkkale
Kırşehir	Van	Muğla
Nevşehir		Niğde
Rize		Ordu
Siirt		Tekirdağ
Sinop		Tokat
		Uşak
		Yozgat
		Zonguldak

Çizelge 5.41 *BKO'ya* göre yerleşim veri kümeleme sonuçları açıklaması

Küme 1	Küme 2	Küme 3
Adıyaman	Afyon	Bolu
Amasya	Aydın	Çanakkale
Artvin	Balıkesir	Çorum
Batman	Denizli	Edirne
Bingöl	Elazığ	Giresun
Burdur	Isparta	Isparta
Erzincan	K.Maraş	Kırıkkale
Hakkari	Malatya	Muğla
Kars	Manisa	Niğde
Kırşehir	Sivas	Tekirdağ
Nevşehir	Urfa	Tokat
Niğde	Trabzon	Uşak
Rize	Van	Zonguldak
Siirt		
Sinop		
Yozgat		

Bu bölümde yapılan çalışmalar sonucunda;

K-Ortalama kümeleme algoritmasının başlangıca karşı duyarlılığı örneklerle gösterilmiştir.
Benzer şekilde bulanık K-Ortalama probleminin de başlangıçta rasgele olarak belirlenen üyelik değerlerine karşı duyarlı olduğu gösterilmiştir.

K-Harmonik Ortalama kümeleme yönteminin bu duyarlılıkları büyük ölçüde ortadan kaldırdığı gösterilmiştir.

K-Harmonik Ortalama yöntemini öneren Zhang (1999) eniyi p değeri olarak 3,5 belirlemesine rağmen yapılan testlerde KO amaç fonksiyon değerine göre performans değerlendirmesi yapıldığında 2,3 değerinin daha iyi sonuçlar verdiği tespit edilmiştir.

Bulanık K-Ortalama için geliştirilen küme değerlendirme yöntemlerinin K-Harmonik Ortalama ve bu çalışmada geliştirlen sezgiseller için de uygulanabilir olduğu örneklerle test edilmiştir.

Tavlama benzetimi ile geliştirilen yöntem işlemci zamanı olarak en yavaş çalışan yöntemdir. Buna karşılık *PARSEKHO* sezgiseli ile geliştirilen yöntem ise üç yöntem arasında en hızlısıdır.

6. SONUÇ VE ÖNERİLER

Kümeleme analizi bir araştırmada incelenen birimleri aralarındaki benzerliklerine göre belirli gruplar içinde toplayarak sınıflandırma yapmayı, birimlerin ortak özelliklerini ortaya koymayı ve bu sınıflar ile ilgili genel tanımlar yapmayı sağlayan bir yöntemdir.

Veri kümeleme analizi problemi kavram olarak ilk defa 1939 yılında dile getirilmiş ve o tarihten bu yana üzerindeki çalışmalar devam etmektedir. Kümeleme analizinin bir alt problem veya model olarak ortaya çıktığı alanlar veri madenciliği ve vektör nicemleme problemleridir. Veri madenciliğinde bir model olarak, vektör nicemlemede ise bir alt problem olarak ele alınmaktadır.

Veri kümeleme analizi yöntemleri hiyerarşik ve hiyerarşik olmayan yöntemler olarak iki başlık altında ele alınmaktadır.

Merkez tabanlı kümeleme problemleri hiyerarşik olmayan kümeleme analizi literatüründe üzerinde en çok çalışılan problemlerden birisidir. Bunun en önemli nedenlerinden birisi problemin bir eniyileme problemi olarak görülmesi ve eniyileme problemlerine yönelik olarak geliştirilen çözüm yöntemlerinin çokluğu olarak belirtilebilir.

Merkez tabanlı kümeleme probleminin en temel çözüm algoritması K-Ortalama kümeleme algoritmasıdır. Bu algoritma çok hızlı ve anlaşılması çok kolay olmasına rağmen özellikle klasik küme mantığını kullanmasından dolayı esnek olmaması, ilk atanan merkezlere karşı duyarlı olması ve yerel en iyi tuzağına takılması nedeniyle üzerinde bir çok geliştirme çalışmaları yapılmıştır. Bu çalışmalardan birisi bulanık küme teorisi ve K-Ortalama kümeleme yaklaşımlarını birleştiren bulanık K-Ortalama kümeleme yöntemi, diğeri ise harmonik ortalama ve en küçük fonksiyonu arasındaki benzerlikten yararlanılarak geliştirilen K-Harmonik Ortalama kümeleme yöntemidir.

K-Harmonik Ortalama kümeleme algoritması K-Ortalama kümelemenin ilk merkezlere karşı duyarlılık problemine karşı geliştirilmiş bir yöntem olmakla birlikte amaç fonksiyonunun birden fazla yerel en küçük noktaya sahip olması nedeniyle yerel en iyi tuzağına takılma problemini çözememektedir.

Sezgisel arama yöntemleri amaç fonksiyonunun birden fazla yerel en küçük noktaya sahip olduğu eniyileme problemlerinde tümel en iyi çözüme yakın çözümler üretebilmektedir. Bu yöntemlerin en çok kullanılanları tabu arama yöntemi, tavlama benzetimi yöntemi ve genetik algoritmaların bir türevi olarak da nitelendirilen parçacık sürü en iyilemesi yöntemidir.

Kümeleme analizindeki önemli aşamalardan bir tanesi de elde edilen küme yapısının mevcut veri yığınına en uygun ve onu en iyi ayrıştıran yapı olduğunun değerlendirilmesi aşamasıdır. Bu amaçla hem klasik K-Ortalama kümeleme hem de bulanık kümeleme yöntemleri için küme değerlendirme kriterleri önerilmiş olmasına rağmen KHO kümeleme yöntemi henüz yeni bir yöntem olmasından dolayı bu yöntemle ilgili önerilen bir küme değerlendirme kriteri yoktur.

Bu çalışmada hem bulanık kümeleme yöntemleri için önerilen çeşitli değerlendirme kriterlerinin KHO kümeleme yöntemi ve onun türevleri içinde tutarlı sonuçlar üretebildiği gösterilmiş hem de tabu arama, tavlama benzetimi ve parçacık sürü en iyilemesi gibi sezgisel yöntemlerden yararlanarak K-Harmonik Ortalama kümeleme probleminde tümel en iyi çözüme yakın çözümler elde etmek için üç farklı algoritma geliştirilmiştir. Geliştirilen algoritmalara TABAKHO, TAVBEKHO ve PARSEKHO adları verilmiştir.

TABAKHO yöntemi ile klasik tabu arama sezgisel yöntemi ve KHO kümeleme yöntemini birleştirerek KHO kümeleme yönteminden daha iyi sonuçlar elde etmeye yarayan sezgisel bir yöntem olarak sunulmuştur. TAVBEKHO yöntemi ile tavlama benzetimi sezgisel yöntemi ile KHO kümeleme yöntemini birleştirerek KHO kümeleme yönteminden daha iyi sonuçlar elde etmeye yarayan sezgisel bir yöntem olarak sunulmuştur. PARSEKHO yöntemi ile ise sürü en iyilemesi yöntemleri ve

KHO kümeleme yöntemini birleştirerek KHO kümeleme yönteminden daha iyi sonuçlar elde etmeye yarayan sezgisel bir yöntem olarak sunulmuştur.

K-Ortalama kümeleme, bulanık K-Ortalama kümeleme ve K-Harmonik Ortalama kümeleme analizinde farklı performans değerlendirme kriterleri kullanıldığından bu yöntemleri ve bu çalışmada geliştirilen diğer yöntemleri karşılaştırmak için ortak bir kriter olarak K-Ortalama kümeleme performans değerlendirme kriteri kullanılmıştır. Bu kriter kullanıldığında en iyi sonuçlar $1{,}5 \leq p \leq 2{,}3$ değerleri ile edilmiştir. Merkez tabanlı kümeleme analizi ile ilgili yapılan çalışmalarda en çok kullanılan veri kümesi olan iris veri kümesi ile üç küme için yapılan analizde göre en iyi sonuç *p=2,3* ve *J(U,X)=79,060* ile PARSEKHO sezgiseli ile elde edilmiştir. Bardak veri kümesi ve iki küme için yapılan analizde geliştirilen her üç sezgisel yöntemde *p=2,3* ve *J(U,X)=824,523* ile en iyi sonuçları vermektedir. Şarap veri kümesinde *p=2,0*, kanser veri kümesinde ise *p=1,5* en iyi sonucu üretmektedir.

Bu çalışmada geliştirilen yöntemlerin amacının KHO kümeleme yönteminin zayıf taraflarını gidermeye yönelik olduğundan, bu çalışmada hem performans kriteri hem de küme değerlendirme kriteri olarak esnek kümeleme değerlendirme indeksleri ile de karşılaştırmalar yapılmıştır. İris veri kümesi ve üç küme için yapılan karşılaştırmada, PC değerlendirme indeksi için *p=1,5* ve *J(PC)=0,930* ile geliştirilen sezgisel yöntemler en iyi sonucu vermekte iken, PE değerlendirme indeksi için *p=3,5* ve *J(PE)=0,125* ile tüm yöntemler aynı sonucu, FS değerlendirme indeksi için *p=2,0* ve *J(FS)=-180,651* ile PARSEKHO en iyi sonucu vermekte, XB değerlendirme indeksi için ise *p=3,5* ve *J(XB)=0,642* ile TAVBEKHO en iyi sonucu vermektedir. Diğer veri kümeleri ile ilgili değerlendirmeler ilgili bölümlerde yapılmıştır.

Yapılan analiz çalışmalarında;

- KO kümeleme yönteminin ilk merkezlere karşı duyarlı olduğu,
- BKO yönteminin başlangıçta rasgele atanan üyelik değerlerine karşı duyarlı olduğu,

- KHO kümelemenin ilk merkezlere karşı duyarlılık problemini önemli ölçüde çözdüğü,
- Geliştirilen sezgisel yöntemlerin k ortalama, bulanık K-Ortalamave k harmoniK-Ortalamagibi alternatiflerinin karşısında katlanılabilir maliyette ve genellikle üstün performans sergiledikleri,
- BKO yöntemi için geliştirilen küme değerlendirme kriterlerinin KHO kümeleme ve geliştirilen sezgisel yöntemler için de kullanılabileceği,
- TAVBEKHO sezgisel yönteminin diğer sezgisellerle karşılaştırıldığında işlemci zamanı olarak en yavaş yöntem olduğu,
- Parçacık sürü en iyilemesi ile geliştirilen yöntemin (PARSEKHO) hem amaç fonksiyon değeri olarak hem de işlem zamanı olarak diğer yöntemlerden daha iyi bir performansa sahip olduğu örneklerle gösterilmiştir.

İlerleyen zamanlarda;

- Genetik algoritmalar, karınca koloni en iyilemesi gibi başka sezgiseller yardımıyla da K-Harmonik Ortalama kümeleme probleminin çözümüne yönelik yöntemler geliştirilmesi,
- Geliştirilen sezgisel yöntemlerin gerçek hayat problemlerine uygulanması,
- Geliştirilen sezgisel yöntemlerin daha süratli çalışabilmeleri için algoritmalarda iyileştirme çalışmalarının yapılması,
- KHO kümeleme ve geliştirilen sezgiseller ile oluşturulan kümeleme yapısının doğruluğunu test edecek özgün değerlendirme kriterlerinin geliştirilmesi,

planlanmaktadır.

KAYNAKLAR

Abraham, A., Guo, H., Liu, H., "Swarm Intelligence: Foundations, Perspectives and Applications, Swarm Intelligent Systems", Nadia Nedjah, Luiza Morelle, Studies in Computational Intelligence, ***Springer Verlag*****,** Germany, 3-25 (2006).

Agrawal, R., Gehrke, J., Gunopoulos, D., Raghavan, P., "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications", ***Proceedings of the ACM- SIGMOD International Conference on Management of Data***, Seattle, Washington, 94-105 (1998).

Akpınar, H., "Veri tabanlarında bilgi keşfi ve veri madenciliği", ***İ.Ü. İşletme Fakültesi Dergisi***, 29 (1):1-22 (2000).

Al-Sultan, K.H., "A Tabu Search Approach to the Clustering Problem", ***Pattern Recognition***, 28:1443-1451 (1995).

Al-Sultan, K., S., and Fedjki, A.,C., "A Tabu Search-Based Algorithm for the Fuzzy Clustering Problem", ***Pattern Recognition***, 30 (12):2023-2030 (1997).

Anderberg, M. R., "Cluster Analysis for Applications", ***Academic Press Inc.***, London, 1-359 (1973).

Ankerst, M., Breunig, M. M., Kriegel, H. P., Sander, J., "OPTICS: Ordering Points To Identify the Clustering Structure", ***Proc. International Conf. on Management of Data (SIGMOD)***, ACM Press, 28 (2):49-60 (1996).

Babu, G., Murty, M., "A near optimal initial seed value selection in K-Means algorithm using a genetic algorithm", ***Pattern Recognition Letters***, 14 (10): 763-769 (1993).

Bagirov, A. M., Yearwood, J., "A new nonsmooth optimization algorithm for en küçük sum-of-squares clustering problems", ***European Journal of Operations Research***, 170:578-596 (2006).

Ball, G., Hall, D., "A clustering technique for summarizing multivariate data", ***Behavioral Science***,12:153-155 (1967).

Belacel, N., Hansen, P., Mladenovic, N., "Fuzzy J-Means: a new heuristic for fuzzy clustering", ***Pattern Recognition***, 35:2193-2200 (2005).

Bezdek, J., "Pattern Recognition with fuzzy objective function algorithm", ***Plenum Press***, New York, 126-137, (1981).

Bischof, H., Leonardis, A., Selb, A., "MDL principle for robust vector quantization", ***Pattern Analysis and Applications***, 2:.59-72 (1999).

Blake, C.L., and Merz, C.J., UCI repository of machine learning databases, http://www.ics.uci.edu/_mlearn/MLRepository.html, (1998).

Brachman, R., Anand, T., "The Process of Knowledge Discovery in Databases: A Human Centered Approach", Advances in Knowledge Discovery and Data Mining, ***AAAI/MIT Press***, USA, 37-57 (1996).

Bradley, P. S., and Fayyad, U., M., "Refining initial points for K-Means clustering", ***Proc. 15th International Conf. on Machine Learning***, Morgan Kaufmann, San Francisco, CA, 91–99 (1998).

Bradley, P. S., Fayyad, U. M., and Mangasarian, O. L., "Mathematical programming for data mining: formulations and challenges", ***INFORMS Journal on Computing***, 11 (3):217–238 (1999).

Cano, J. R., Cordon, O, Herrera, F., Sanchez, L., "A greedy randomized search procedure applied to the clustering problem as an initialization process using K-Means as a local search procedure", ***Journal of Intelligent and Fuzzy Systems***, 12 (3-4):235:242 (2002).

Chen, M.S., Han, J., Yu, P.S., "Data mining: An overview from a database perspective", ***IEEE Transactions on Knowledge and Data Engineering***, 8 (6):866-883 (1996).

Cheung, Y-M., "k*-Means: A new generalized k-means clustering algorithm", ***Pattern Recognition Letters***, 24:2883-2893 (2003).

Chidand, A., Shalom, W., "Data mining with decision trees and decision rules", ***Future Generation Computer Systems***, 13 (2-3):197-210 (1997).

Chiu, S. L., "Fuzzy model identification based on cluster estimation", ***Journal of Intelligent and Fuzzy Systems***, 2 (3):267-278 (1994).

Chu, S., Roddick, J., "A clustering algorithm using tabu search approach with simulated annealing for vector quantization", ***Chinese Journal of Electronics***, 12 (3):349-353 (2003).

Cios, K., Kurgan, L., "Trends in data mining and knowledge discovery", Knowledge Discovery in Advanced Information Systems, ***Springer***, 1-27 (2002).

Clements, F., C., "Use of cluster analysis with anthropological data", ***American Anthropologist***, New Series, Part 1, 56 (2):180-199 (1954).

Cox, T. F., Cox, M., A., A., "Note on grouping", ***Journal of the American Statistical Association***, 52 (280):543-547 (1957).

Cui, X., Potok, T. E., "Document clustering analysis based on hybrid PSE+K-Means algorithm", ***Journal of Computer Sciences***, Special Issue, 27:33 (2005).

Cowgill, M. C., Harvey, R. J., Watson, L. T., "A genetic algorithm approach to cluster analysis", ***Computers and Mathematics with Applications***, 37:99-106 (1999).

Dave, R. N., "Characterization and detection of noise in clustering", ***Pattern Recognition Letters***, 12 (11):657-664 (1991).

Delgado, M., Skarmeta, A., Barbera, H., "A tabu search approach to the fuzzy clustering problem", ***Sixth IEEE International Conference on Fuzzy Systems***, Barcelona, 125-130 (1997).

Deogün, J. S., Raghavan, V.V., Sever, H., "Data mining: research trends, challenges and applications", Roughs Sets and Data Mining: Analysis of Imprecise Data (T. Y. Lin and N. Cercone, eds.), ***Kluwer Academic Publishers***, Boston, MA, 9-45 (1997).

Doring, C., Borgelt, C., Kruse, R., "Fuzzy clustering of quantitative and qualitative data", ***Proceedings of NAFIPS'04***, 1:84-89 (2004).

Duda, R. and Hart, P., "Pattern Classification and Scene Analysis", ***John Wiley and Sons***, Newyork, 201-202 (1973).

Dunham, M. H., "Data mining: Introductory and advanced topics", ***Prentice Hall***, USA, 11-13 (2003).

Dunn, J. C., "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters", ***Journal of Cybernetics***, 3:32-57 (1973).

Dunn, J. C., "Well seperated clusters and optimal fuzzy partitions", ***Journal of Cybernetics***, 4:95-104 (1974).

Eberhart R. C., Kennedy J., Shi Y., "Swarm Intelligence", 1st Ed., ***Morgan Kaufmann***, 287-324 (2001).

Elder, J.F., and Pregibon, D., "A statistical perspective on knowledge discovery in databases", Advances in Knowledge Discovery and Data Mining, U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds., ***AAAI/MIT Press***, Menlo Park, CA, 83-113 (1996).

Ester, M., Kriegel, H. P., Sander, J., Xu, X., "A density-based algorithm for discovering clusters in large spatial databases with noise", ***Proceedings of the 2nd International Conference on KDD***, Portland, Oregon USA, 226-231 (1996).

Famili, A., Shen, W.-M., Weber, R., Simoudis, E., "Data pre-processing and intelligent data analysis", ***Intelligent Data Analysis***, 1 (1):3-23 (1997).

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., "From data mining to knowledge discovery in databases", Advances in Knowledge Discovery and Data Mining, ***AAAI/MIT Press***, Cambridge, Massachussets, 37-54 (1996).

Feng, H-M., Chen, C-Y., Ye, F., "Evolutionary fuzzy particle swarm optimization vector quantization learning scheme in image compression", ***Expert Systems with Applications***, 31 (1):213-222 (2006).

Fisher,R. A., "The use of multiple measurements in taxonomic problems", ***Annual Eugenics***, 7 (2):179-188 (1936).

Fisher, W., D., "On grouping for maximum homogeneity", ***Journal of the American Statistical Association***, 53 (284): 789-798 (1958).

Forgy, E., W., "Cluster analysis of multivariate data: Efficiency versus interpretability of classifications", ***Biometrics***, 21 (3):768–769 (1965).

Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C. J., "Knowledge discovery in databases: An overview", Knowledge Discovery in Databases, Piatetsky-Shapiro, G., Frawley, W.J., ***AAAI Press/MIT Press***, Cambridge, 1-27 (1991).

Fraley, C., Raftery, A. E., "How many clusters? Which clustering method? Answers via model-based cluster analysis", ***The Computer Journal***, 41 (8):578-588 (1998).

Fränti, P., Kivijärvi, J., Kaukoranta, T., Nevalainen, O., "Genetic algorithms for codebook generation in vector quantization", ***3rd Nordic Workshop on Genetic Algorithms***, Helsinki, Finland, 207-222 (1997).

Fränti, P., Kivijrvi, J., Nevalainen, O., "Tabu search algorithm for codebook generation in vector quantization", ***Pattern Recognition***, 31 (8):1139-1148 (1998).

Fränti, P., Kivijarvi, J., "Randomised local search algorithm for the clustering problem", ***Pattern Analysis and Applications***, 3:358–369 (2000).

Fritzke, B., "The LBG-U method for vector quantization over LBG inspired from neural networks", ***Neural Processing Letters***, 5:35-45 (1997).

Garey, M. R., Johnson, D. S. and Witsenhausen, H. S., "The complexity of the generalized Lloyd-Max problem", ***IEEE Trans. Information Theory***, 8 (2):255-256 (1982).

Gath, I., Geva, A., "Unsupervised optimal fuzzy clustering", ***IEEE Transactions on Pattern Analysis and Machine Intelligence***, 11 (7):773-781 (1989).

Gilbert, E., W., "Pioneer maps of health and disease in England", ***Geographical Journal***, 124:172-183 (1958).

Glover, F., "Tabu Search- Part I"***, ORSA Journal of Computing***, 1:190-206 (1989).

Gray, R. M., Neuhoff, D. L., "Quantization", ***IEEE Transactions on Information Theory***, 44 (6):2325-2383 (1998).

Guan, Y., Ghorbani, A. A.,Belace, N., "K-Means+: An autonomous Clustering Algorithm", ***Technical Report (TR04-164), UNB Faculty of Computer Science***, Canada, 3-7 (2004).

Guha, S., Rastogi, R., Shim, K., "ROCK: A Robust Clustering Algortihm for Categorical Attributes", ***Information Systems***, 25 (5):345-366, (2000).

Gustafson, D., Kessel, W., "Fuzzy clustering with a fuzzy covariance matrix", ***Proceedings of IEEE CDC***, San Diego, 761-766 (1979).

Halkidi, M., Vazirgiannis, M., Batistakis, I., "Quality scheme assessment in the clustering process", ***Proceedings of PKDD***, Helsinki, Finland, 265-276 (2002).

Hammerly, G., and Elkan, C., "Alternatives to the k-means algorithm that find better clusterings", ***Proc.of the 11 th International Conference on Information and Knowledge Management***, Virginia, USA, 600-607 (2002).

Han, J., Kamber, M., "Data Mining: Concepts and Techniques", ***Morgan Kaufmann Publishers***, New York, 335-391 (2001).

Hand, D. J., Mannila, H., Smyth, P., "Principles of Data Mining", ***MIT Pres***, 211-270 (2001).

He, J., Lan, M., Tan, C-L., Sung, S-Y., Low, H-B., "Initialization of cluster refinement algorithms:a review and comparative study", ***Proceedings of the IEEE International Joint Conference Neural Networks***, Budapest, Hungary, 1:302-307 (2004).

Hinneburg, A., Keim, D. A., "An Efficient Approach to Clustering in Large Multimedia Databases with Noise", ***Proceedings of the 4th International Conference on KDD***, USA, 58-65 (1998).

Huang, Z., "A fast clustering algorithm to cluster very large categorical data sets in data mining", ***Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery***, Tucson, Arizona, 324-335 (1997).

Huang, H-C., Chu, S-C., Pan, J-S., Lu, Z-M., "A tabu search based maximum descent algorithm for VQ codebook design", ***Journal of Information Science and Engineering***, 17:753-762 (2001).

Huang, C. H., Pan, J. S., Lu, Z. H., Sun, S. H., Hang, H.M., "Vector quantization based on genetic simulated annealing", ***Signal Processing***, 81:1513-1523 (2001).

Huang, K., "A synergistic automatic clustering technique for multispectral image analysis", ***Photogrametric Engineering and Remote Sensing***, 1 (1):33-40 (2002).

Jain, A. K., Murty, M. N., and Flynn, P. J., "Data clustering: a review", ***ACM Computing Surveys***, 31 (3):264–323 (1999).

Jain, A.K. and Dubes, R.C., "Algorithms for Clustering Data", Prentice-Hall advanced reference series, ***Prentice Hall Inc.***, New Jerset, 55-141 (1988).

Jain, A. K., Topchy, A., Law, M. H. C., Buhmann, J. B., "Landscape of clustering algorithms", ***Proceedings of the 17th Int.Conf. on Pattern Recognition***, Cambridge, UK, 1260-1263 (2004).

Kanade, P., M., and Hall, L., O., "Fuzzy ants as a clustering concept", ***22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS)***, 227:232 (2003).

Kanade, P., M., and Hall, L., O., "Fuzzy ants by centroid positioning", ***Fuzzy Systems***, 1:371-376 (2004).

Kantardzic, M., "Data Mining: Concepts, Models, Methods, and Algorithms", ***John Wiley & Sons Inc.***, Hoboken, New Jersey, 157-169 (2003).

Karayiannisa, N. B., and Randolph-Gips, M. M., "Non-Euclidean c-means clustering algorithms", ***Intelligent Data Analysis***, 7:405–425 (2003).

Katsavounidis, I., Jay Kuo, C., Zhang, Z., "A new initialization technique for generalized Lloyd iteration", ***Signal Processing Letters***, 1:144-146 (1994).

Kaufman, L., Rousseeuw, P. J., "Finding Groups in Data, An Introduction to Cluster Analysis", ***Wiley***, Canada, 1-162 (1990).

Kearns, M., Mansour, Y., and Yang, A., "An information theoretic analysis of hard and soft assignment methods for clustering", ***Proceedings of Uncertainty in Artificial Intelligence***, San Fransisco, 282–293 (1997).

Kennedy, J., Eberhart, R. C., "Particle swarm optimization", ***Proceedings of the IEEE International Conference on Neural Networks***, Perth, Australia, 4:1942-1948 (1995).

Khan, S., S., Ahmad, A., "Cluster center initialization algorithm for K-means clustering", ***Pattern Recognition Letters***, 25:1293-1302 (2004).

King, B., "Step-wise clustering procedures", ***Journal of the American Statistical Association***, 69:86-101 (1967).

Kirkpatrick, S., Gerlatt, C. D., Vecchi, M. P., "Optimization by simulated annealing", ***Science***, 220:671-680 (1983).

Klawonn, F., Höppner, F., "What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier", Advances in Intelligent Data Analysis, M. R. Berthold, H. J. Lenz, E. Bradley, R. Kruse, C. Borgelt, ***Springer,*** Berlin, 254-264 (2003).

Klein, R., Dubes, R., "Experiments in projection and clustering by simulated annealing", ***Pattern Recognition***, 22:213-220 (1989).

Kohonen, T., "Self Organizing Maps", Springer Series in Information Sciences, 30, ***Springer-Verlag***, NewYork, USA, 1-12 (1995).

Kövesi, B., Boucher, J. M., Saoudi, S., "Stochastic K-means algorithm for vector quantization", ***Pattern Recognition Letters***, 22:603-610 (2001).

Krishna, K., Murty, M. N., "Genetic K-Means algorithm", ***IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics***, 29 (3):433-439 (1999).

Larose, D. T., "Data mining methods and models", ***John Wiley & Sons Inc.***, Hoboken, New Jersey, 147-161 (2006).

Likas, A., Vlassis, N., and Verbeek, J. J., "The global k-means clustering algorithm", ***Pattern Recognition***, 36 (2):451-461, (2002).

Lin, Y., Shiueng, C., "A genetic approach to the automatic clustering problem", ***Pattern Recognition***, 34:415 – 424 (2001).

Linde, Y., Buzai, A., Gray, R., "An algorithm for vector quantizer design", ***IEEE Transactions on Communications***, 28 (1):84-85 (1980).

Lloyd, S. P. , "Least squares quantization in PCM", ***IEEE Transactions Information Theory (Special Issue on Quantization)***, IT-28:129–137 (1982).

Lorette, A., Descombes, X., Zerubia, J., "Fully unsupervised fuzzy clustering with entropy criterion", ***International Conference on Pattern Recognition (ICPR'00)***, Barcelona, 3:3998-4001 (2000).

MacQueen, J., "Some methods for classification and analysis of multivariate observations", ***Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability***, University of California Press, 1:281-297 (1967).

Mahamed G. H. Omran, "Particle Swarm Optimization Methods for Pattern Recognition and Image Processing," Doctoral Dissertation submitted to Faculty of Engineering, ***Built Environment and Information Technology, University of Pretoria***, South Africa, 47-103 (2004).

Mannila, H., "Methods and problems in data mining", ***Proceedings of the International Conference on Database Theory,*** Greece, 41-55 (1997).

Masulli, F., Rovetta, S., "Soft transition from probabilistic to possibilistic fuzzy clustering", ***IEEE Transactions on Fuzzy Systems***, 14 (4):516-527 (2006).

Matheus, J. C., Chan, P. K., Piatetsky-Shapiro, G., "Systems for knowledge discovery in databases", ***IEEE Transactions on Knowledge and Data Engineering***, 5:903-913 (1993).

Maulik, U., Bandyopadhyay, S., "Genetic algorithm based clustering technique", ***Pattern Recognition***, 33:1455-1465 (2000).

Metropolis, N., Rosenbluth, M., Rosenbluth, A., Teller, E., Teller, J., "Equations of state calculations by fast computing machines", ***Journal of Chemical Physics***, 21 (6):1087-1092 (1953).

Mitra, S., Pal, S. K., and Mitra, P., "Data mining in soft computing framework: a survey", ***IEEE Transactions on Neural Networks***, 13 (1):3-14 (2002).

Newman, V., "Redefining knowledge management to deliver competitive advantage"**,** ***Journal of Knowledge Management***, 1 (2):123-32 (1997).

Ng, R. and Han, J., "Efficient and Effective Clustering Methods for Spatial Data Mining", ***Proceedings of the 20th VLDB Conference***, Santiago, Chile, 144-155 (1994).

Ng, M. K., Wong, J. C., "Clustering categorical data sets using tabu search techniques", ***Pattern Recognition***, 35:2783-2790 (2002).

Omran, G. H. M., Engelbrecht, A. P., Salman, A., "Dynamic Clustering Using Particle Swarm Optimization with Application in Unsupervised Classification", **Transactions on Engineering, Computing and Technology**, 9:199-204, (2005).

Pal, N. R., Bezdek, J. C., "On cluster validity for the fuzzy c-means model", **IEEE Transactions on Fuzzy Systems**, 3:370-379 (1995).

Pan, S-M., Cheng, K-S., "An evolution based tabu search approach to codebook design", ***Pattern Recognition***, Article in press (2006).

Patane, G., Russo, M., "The enhanced LBG algorithm", ***Neural Networks***, 14 (9):1219-1237 (2001).

Paterlini, S., Krink, T., "Differential evolution and particle swarm optimization in partitional clustering", ***Computational Statistics & Data Analysis***, 50:1220-1247 (2006).

Pedrycz, W., "Fuzzy Set Technology in Knowledge Discovery", ***Fuzzy Sets and Systems***, 98 (3): 279-290 (1998).

Pedrycz, W., Vukovich, G., "Fuzzy clustering with supervision", ***Pattern Recognition***, 37:1139-1349 (2004).

Pedrycz, W., "Knowledge based clustering: from data to information granules", ***John Wiley & Sons Inc.***, Hoboken, New Jersey , 1-26 (2005).

Pelleg, D., and Moore, A., "X-means: Extending K-means with efficient estimation of the number of clusters", ***Proceedings of the 17th International Conf. on Machine Learning***, Stanford, CA, 727–734 (2000).

Pena, J. M., Lozano, J.A., Larranaga, P., "An empirical comparison of four initialization methods for the K-Means algorithm", ***Pattern Recognition Letters***, 20:1027-1040 (1999).

Pham, D. L., Prince, J. L., "An adaptive fuzzy c means algorithm for image segmentation in the presence of intensity inhomogeneities", ***Pattern Recognition Letters***, 20 (1):57-68 (1999).

Portnoy, L., Eskin, E., Stolfo, S. J., "Intrusion Detection with Unlabeled data Using Clustering", ***Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)***, Philadelphia, 80-92 (2001).

Qin, A., K., Suganthan, P. N., "Initialization insensitive LVQ algorithm based on cost-function adaptation", ***Pattern Recognition***, 38:773-776 (2005).

Raghavan, V., Birchand, K., "A clustering strategy based on a formalism of the reproductive process in a natural system", ***Proceedings of the Second International Conference on Information Storage and Retrieval***, Dallas, Teksas, 10-22 (1979).

Ramkumar, G. D., Swami, A. N., "Clustering data without distance functions", ***IEEE Data Engineering Bulletin***, 21 (1):9-14 (1998).

Rosenberger, C., Chehdi, K., "Unsupervised Clustering Method with Optimal Estimation of the Number of Clusters: Application to Image Segmentation", ***International Conference on Pattern Recognition (ICPR'00)***, Barcelona, 1:1656-1659 (2000).

Rubinov, A. M., Soukhorokova, N. V., Ugon, J., "Classes and clusters in data analysis", ***European Journal of Operational Research***, 173:849-865 (2006).

Setnes, M., Kaymak, U., "Extended fuzzy c-means with volume prototypes and cluster merging", ***Proceedings of the 6th European Conference on Intelligent Techniques and Soft Computing (EUFIT'98),*** Aachen, Germany, 1360-1364 (1998).

Sheikholeslami, C., Chatterjee, S., and Zhang, A., "WaveCluster: A-multiresolution clustering approach for very large spatial database", ***Proceedings of 24th VLDB Conference***, New York, USA, 428-439 (1998).

Shelokar, P. S., Jayaraman, V. K., Kulkarni, B. D., "An ant colony approach for clustering", ***Analytica Chimica Acta***, 509:187-195 (2004).

Shen, W., Hasegawa, O., "An adaptive incremental LBG for vector quantization", ***Neural Networks***, 19:694-704 (2006).

Shi, Y. H., Eberhart, R. C., "A modified particle swarm optimizer", ***IEEE International Conference on Evolutionary Computation***, Anchorage, Alaska, 69-73 (1998).

Simoudis, E., "Reality check for data mining", ***IEEE Expert:Intelligent systems and their applications***, 11 (5):26-33 (1996).

Smyth, P., "Data mining at the interface of computer science and statistics", Invited chapter in Data Mining for Scientific and Engineering Applications, ***Kluwer Academic Publishers***, London, 35-62 (2001).

Sneath, P. H. A., Sokal, R. R., "Numerical Taxonomy", ***Freeman***, San Fransisco, California, 114-253 (1973).

Sokal, R., R., Sneath, P., H., A., "Principles of numerical taxonomy", ***W. H. Freeman and Co.***, San Fransisco, 169-215 (1963).

Sousa, T., Silva, A., Neves, A., "Particle swarm based data mining algorithms for classification tasks", ***Parallel Computing***, 30:767-783 (2004).

Sung, C., S., and Jin, H.,W., "A tabu search-based heuristic for clustering", ***Pattern Recognition***, 33:849-858 (2000).

Tantuğ, A. C., "Veri madenciliği ve Demetleme", Yüksek Lisans Tezi, ***İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü***, 26-42 (2002).

Tou, J., "DYNOC-A Dynamic Optimal Cluster Seeking Technique", ***International Journal of Computer and Information Sciences***, 8 (6):541-547 (1979).

Tryon, R., C., "Cluster Analysis", ***Edwards Brothers***, Ann Arbor, MI, 1-277 (1939).

Trejos, J., Piza, E., Murillo, A., "A tabu search algorithm for partitioning", ***Preprint 1-1999 CIMPA, University of Costa Rica***, 2060 San Jose, 1-13 (1999).

Veenman, C., Reinders, M., Backer, E., "A maximum variance cluster algorithm", ***IEEE Transactions on Pattern Analysis and Machine Intelligence***, 24 (9):1273-1280 (2002).

Vladimir, C., Filip, M., "Learning from data-Concepts, Theory and Methods", ***John Wiley***, Newyork, 15-57 (1997).

Wallace, C., "An improved program for classification", ***Proceedings of the Nineteenth Australian Computer Science Conference***, Australia, 8:357-366 (1986).

Wallace, C. S., Dowe, D.L., "Intrinsic classification by MML – the snob program", ***Proceedings to the 7th Australian Joint Conference on Artificial Intelligence***, UNE, Armidale, NSW, Australia, 37-44 (1994).

Wang, W., Yang, J., and Muntz, R., "STING: A Statistical Information Grid Approach to Spatial Data Mining", ***Proceedings of 23rd VLDB Conference***, Athens, 186-195 (1997).

Wang, X., Y., and Garibaldi, J., M., "Simulated Annealing Fuzzy Clustering in Canser Diagnosis", **Informatica**, 29:61-70 (2005).

Wang, L., Fu, X., "Data Mining with Computational Intelligence", ACM Computing Classification, ***Springer***, Berlin, Heidelberg Newyork, 1-20 (2005).

Ward, J. H., "Hierarchical grouping to optimize an objective function", **Journal of the American Statistical Association,** 58:236-244 (1963).

Witten, I. H., Frank, E., "Data mining practical machine learning tools and techniques", ***Morgan Kaufmann Publishers***, San Francisco, 254-270 (2005).

Wu, K-L., Yang, M-S., "Alternative c-mean clustering algorithms", ***Pattern Recognition***, 35:2267-2278 (2002).

Wu, K-L., Yang, M-S., "Unsupervised possibilistic clustering", ***Pattern Recognition***, 39:5-21 (2006).

Xie, X. L., Beni, G. A., "Validity measure for fuzzy clustering", ***IEEE Transaction on Pattern Analysis and Machine Intelligence***, 3:841-846 (1991).

Yager, R. R., Filev, D. P., "Approximate clustering via the mountain method", ***IEEE Transactions on Systems, Man and Cybernetics***, 24:1279-1284 (1994).

Ye, N., "The Handbook of Data Mining", ***Lawrence Erlbaum Associates Publishers***, Mahwah, New Jersey, London, 248-273 (2003).

Ye, F., Chen, C-Y., "Alternative KPSO-Clustering Algorithm", ***Tamkang Journal of Science and Engineering***, 8 (2):165-174 (2005).

Zhang, B., Hsu, M., and Dayal, U., "K-harmonic means - a data clustering algorithm", ***Technical Report HPL-1999-124, Hewlett-Packard Laboratories***, Palo-Alto, 1-25 (1999).

Zhang, B., Kleyner, G., Hsu, M., "A local search approach to K-Clustering", ***Technical Report HPL-1999-119, Hewlett-Packard Laboratories,*** Palo-Alto, 1-20 (1999).

Zhang, B., Hsu, M., Dayal, U., "K-Harmonic Means", ***International Workshop on Temporal, Spatial and Spatio-Temporal Data Mining, TSDM2000***, Lyon, 31-45 (2000).

Zhang, B., "Dependence of Clustering Algorithm Performance on Clusteredness of Data", ***HP Labs. White Papers***, Palo-Alto, 1-12 (2001).

Zhang, T., Ramakrishnan, R., Livny, M., "BIRCH: An efficient data clustering method for very large databases", ***Proceedings of the ACM SIGMOD International Conference on Management of Data***, Montreal, Canada, 103-114 (1996).

Zhou, Z.-H., "Three perspectives of data mining", ***Artificial Intelligence***, 143 (1):139-146 (2003).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	: ÜNLER, Alper
Uyruğu	: T.C.
Doğum tarihi ve yeri	: 11.06.1972 Ankara
Medeni hali	: Evli
Telefon	: 0 (312) 411 21 28
Faks	: 0 (312) 438 86 42
e-mail	: aunler@kkk.tsk.mil.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Yılı
Uzmanlık	ODTÜ/ Bilgisayar Mühendisliği	2000
Yüksek Lisans	Yeditepe Üniversitesi/Sistem Müh. Bölümü	1999
Lisans	Kara Harp Okulu/Sistem Müh. Bölümü	1994
Lise	Kuleli Askeri Lisesi	1990

İş Deneyimi

Yıl	Yer	Görev
1994-2000	K.K.K. Birlikleri	Tim ve Takım Komutanlığı
2000-2006	K.K.K. Karargahı	Sistem Analisti ve Prog.

Yabancı Dil

İngilizce

Yayınlar

1. Güngör, Z., Ünler, A., K-Harmonic Means Data Clustering with Tabu Search Method, Proceedings of 5th International Symposium on Intelligent Manufacturing Systems, 346-360, 2006.

2. Güngör, Z., Ünler, A., K-Harmonic Means Data Clustering with Simulated Annealing Heuristic, Applied Mathematics and Computation, In Press, Corrected Proof, 2006.

Hobiler

Yüzme, Bilgisayar ve yazılım teknolojileri, Yapay Zeka