T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİNDE KÜMELEME ANALİZİ VE HİBRİT VERİLERİN KÜMELENMESİ ÜZERİNE BİR ALGORİTMA

OSMAN ÇÖREKCİ

YÜKSEK LİSANS TEZİ
MATEMATİK MÜHENDİSLİĞİ ANABİLİM DALI
MATEMATİK MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
DOÇ. DR. AYLA ŞAYLI

İSTANBUL, 2017

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİNDE KÜMELEME ANALİZİ VE HİBRİT VERİLERİN KÜMELENMESİ ÜZERİNE BİR ALGORİTMA

Osman Çörekci tarafından hazırlanan tez çalışması 15.06.2017 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Ayla ŞAYLI

Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç. Dr. Ayla ŞAYLI

Yıldız Teknik Üniversitesi _____

Prof. Dr. İbrahim EMİROĞLU

Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Mustafa Zahid GÜRBÜZ

İstanbul Doğuş Üniversitesi _____

ÖNSÖZ

Akademik hayatımın bu ilk tezinde, günümüzün ve geleceğimizin kaçınılmaz uğraşlarından birine, veri madenciliğine değindim. Veri madenciliğinin önemli başlıklarından biri olan kümeleme analizi alanında yaptığım derlemeleri ve hibrit verilerin etkili ve hızlı bir biçimde kümelenmesi için geliştirdiğim bir algoritmayı paylaştım. Yaptığım araştırma ve literatür taraması esnasında bilimin birikimli olarak ilerleyen bir süreç olduğunu yaşayarak tecrübe ettim. Bu çalışmamın da yeni çalışmalar yapacak olanlara bir kaynak olabilmesi en büyük temennimdir.

Gerek yüksek lisans eğitimim de gerekse bu tezi hazırlarken benden yardımlarını esirgemeyen hocam, tez danışmanım Ayla ŞAYLI'ya, yüksek lisans eğitimimde ders aldığım tüm hocalarıma ve destekleriyle beni yalnız bırakmayan çok sevdiğim aileme teşekkürü bir borç bilirim.

Haziran, 2017

Osman ÇÖREKCİ

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ vi
KISALTMA LİSTESİ vii
ŞEKİL LİSTESİ viii
ÇİZELGE LİSTESİ ix
ÖZET x
ABSTRACT xi
BÖLÜM 1
GİRİŞ 1
1.1 Literatür Özeti 1
1.2 Tezin Amacı 6
1.3 Hipotez 7
BÖLÜM 2
VERİ HAZIRLAMA İŞLEMLERİ 8
2.1 Verinin Temizlenmesi 9
2.1.1 Kayıp Veri 9
2.1.2 Gürültülü Veri 10
2.2 Verinin Bütünleştirilmesi 12
2.3 Verinin Dönüştürülmesi 13
2.3.1 Min-Max Normalleştirmesi 13
2.3.2 Ondalıklı Normalleştirme 14
2.3.3 Z-Score Normalleştirmesi 14
2.4 Verinin İndirgenmesi 15
2.5 Veri Türleri 16
BÖLÜM 3
NOTASYON 19

BÖLÜM 4

KÜMELEME ANALİZİ 21

4.1 Uzaklık ve Benzerlik Ölçüleri 26
4.1.1 Uzaklık Ölçüleri 27
4.1.1.1 İkili Sınıflı Veriler İçin Uzaklık Ölçüleri 27
4.1.1.2 Sürekli Veriler İçin Uzaklık Ölçüleri 29
4.1.1.3 Kategorik Veriler İçin Uzaklık Ölçüleri 30
4.1.2 Benzerlik Ölçüleri 31
4.1.2.1 İkili Sınıflı Veriler İçin Benzerlik Ölçüleri 31
4.1.2.2 Sürekli Veriler İçin Benzerlik Ölçüleri 33
4.1.2.3 Kategorik Veriler İçin Benzerlik Ölçüleri 33
4.1.3 Kümeler Arası Uzaklık Ölçüleri 38
4.1.4 Kullanım Alanlarına Göre Uzaklık ve Benzerlik Ölçüleri 39
4.2 Kümeleme Teknikleri 41
4.2.1 Hiyerarşik Kümeleme Yöntemleri 41
4.2.1.1 Tek Bağlantı Yöntemi 42
4.2.1.2 Tam Bağlantı Yöntemi 42
4.2.1.3 Ortalama Bağlantı Yöntemi 42
4.2.1.4 Ortalama Vektör Yöntemi 43
4.2.1.5 Medoid Yöntemi 43
4.2.1.6 Ward Yöntemi 43
4.2.2 Hiyerarşik Olmayan (Bölücü) Kümeleme Yöntemi 44
4.2.2.1 K-Means Kümeleme Yöntemi 44

BÖLÜM 5

K-MEANS KÜMELEME YÖNTEMİ 45

5.1 Veri Türleri 46
5.2 Uzaklık Ölçüsü Seçimi 47
5.3 Küme Sayısının Belirlenmesi 48
5.4 Başlangıç Merkezlerinin Seçimi 49
5.5 Ağırlık Kavramı 51
5.6 Genişletilmiş K-Means Algoritmaları 52
5.6.1 K-Modes Algoritması 53
5.6.2 K-Prototypes Algoritması 54
5.6.3 K-Harmonik Ortalama Algoritması 56

BÖLÜM 6

W-K-PROTOTYPES ALGORİTMASI 58

6.1 Matematiksel Ön Hazırlık 58
6.2 Verinin Standartlaştırılması 60
6.3 W-K-Prototypes Algoritması 61

BÖLÜM 7

SONUÇ VE ÖNERİLER 63

KAYNAKLAR 79

ÖZGEÇMİŞ 83

SİMGE LİSTESİ

X	Veri kümesi
n	Eleman sayısı
x	X kümesinin herhangi bir elemanı
x'	x elemanının normalleştirilmiş hali
x_{max}	X kümesinin en büyük değeri
x_{min}	X kümesinin en küçük değeri
x_i	X kümesinin i'ninci elemanı
w	Ağırlık vektörü
$mean(X)$	Ortalama değer fonksiyonu
$sd(X)$	Standart sapma fonksiyonu
A_j	A kümesinin j'ninci özelliği
$DOM(A_j)$	j'ninci özelliğin alabileceği değerlerin kümesi
$f_k(x)$	k'nıncı özellikte x değerinin frekansını veren fonksiyon
$\acute{p}_k(x)$	k'nıncı özelliğin x değerini alma olasılığı
x_{ik}	i'ninci özelliğin k'nıncı özelliği
$d(x,y)$	Uzaklık fonksiyonu
$S(x,y)$	Benzerlik fonksiyonu
μ	Ortalama vektör
S^{-1}	Varyans-kovaryans matrisinin tersi
P_{ij}	Penrose uzaklık ölçüsü
T^2	Hotelling T^2 uzaklık ölçüsü
$SLINK(X,Y)$	Tek bağlantı uzaklık ölçüsü
$CLINK(X,Y)$	Tam bağlantı uzaklık ölçüsü
E	Maliyet fonksiyonu
Y	Bölüntüleme matrisi
Q	Küme merkezi
k	Küme sayısı
$\delta(x,y)$	Overlap uzaklık ölçüsü
$w(x)$	Ağırlık fonksiyonu
$m(x)$	Üyelik (Aidiyet) fonksiyonu
$KHM(x)$	K-Harmonik Means maliyet fonksiyonu

KISALTMA LİSTESİ

ENIAC	Electronic Numerical Integrator And Computer
IEEE	Institute of Electrical and Electronics Engineers
KHM	K Harmonik Means
W-K-Prototypes	Weighted K-Prototypes

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 Bilgi keşfi sürecinde veri madenciliğinin yeri 2
Şekil 2.1 Veri madenciliği sürecinde veri hazırlamanın yeri 8
Şekil 2.2 Ham verinin dönüşümü 18
Şekil 3.1 Verinin matris formatında gösterimi 20
Şekil 7.1 Bir normal dağılımlı örneklem 63
Şekil 7.2 Dört normal dağılımlı örneklem 64
Şekil 7.3 Bir normal dağılımlı üç boyutlu örneklem 64
Şekil 7.4 Dört normal dağılımlı üç boyutlu örneklem 65
Şekil 7.5 Ağırlık vektörü w=[0.33,0.33,0.34] alındığında oluşan dağılım 65
Şekil 7.6 Ağırlık vektörü w=[0.8,0.1,0.1] alındığında oluşan dağılım 66
Şekil 7.7 Ağırlık vektörü w=[0.1,0.8,0.1] alındığında oluşan dağılım 66
Şekil 7.8 Ağırlık vektörü w=[0.1,0.1,0.8] alındığında oluşan dağılım 67
Şekil 7.9 Ağırlık vektörü w=[0.2,0.2,0.6] alındığında oluşan dağılım 67
Şekil 7.10 Ağırlık vektörü w=[0.0,0.5,0.5] alındığında oluşan dağılım 68
Şekil 7.11 Ağırlık vektörü w=[0.33,0.33,0.34] alındığında oluşan dağılım 68
Şekil 7.12 Ağırlık vektörü w=[0.8,0.1,0.1] alındığında oluşan dağılım 69
Şekil 7.13 Ağırlık vektörü w=[0.1,0.8,0.1] alındığında oluşan dağılım 69
Şekil 7.14 Ağırlık vektörü w=[0.1,0.1,0.8] alındığında oluşan dağılım 70
Şekil 7.15 Ağırlık vektörü w=[0.2,0.2,0.6] alındığında oluşan dağılım 70
Şekil 7.16 Ağırlık vektörü w=[0.0,0.5,0.5] alındığında oluşan dağılım 71
Şekil 7.17 Bir normal dağılım ve 1500 örnek ile yapılan test sonucu 72
Şekil 7.18 Dört normal dağılım ve k=4 ile yapılan test sonucu 72
Şekil 7.19 Dört normal dağılım ve k=8 ile yapılan test sonucu 73
Şekil 7.20 Dört normal dağılım ve k=16 ile yapılan test sonucu 73
Şekil 7.21 Sekiz normal dağılım ve k=4 ile yapılan test sonucu 74
Şekil 7.22 Sekiz normal dağılım ve k=8 ile yapılan test sonucu 74

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 Kategorik verilerin iki sınıflı olarak gösterimi 18
Çizelge 4.1 Tanımlamalarda kullanılacak notasyon 27
Çizelge 4.2 Çaprazlama matrisi 28
Çizelge 4.3 Kategorik veriler için benzerlik ölçüleri 36
Çizelge 4.4 Veri Tipine Göre Uzaklık ve Benzerlik Ölçüleri 39
Çizelge 7.1 Farklı girişler ile K-Prototypes ve W-K-Prototypes için test sonuçları 76

VERİ MADENCİLİĞİNDE KÜMELEME ANALİZİ VE HİBRİT VERİLERİN KÜMELENMESİ ÜZERİNE BİR ALGORİTMA

Osman ÇÖREKCİ

Matematik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Ayla ŞAYLI

Büyük verilerin kendi içinde benzer kümelere ayrılması veri madenciliğinin en temel problemlerinden biridir. Günümüzde büyük verilerin organize edilmiş bir biçimde depolanabilmesi, kümeleme işlemi için geliştirilen yöntemlerin, büyük veriler için etkili çalışabilmesi kriterinin önemi de arttırmıştır. Hiyerarşik kümeleme yöntemleri etkili sonuçlar verse de hesaplama karmaşıklığı nedeniyle büyük veriler ile çalışma konusunda yetersiz kalmaktadır. Hiyerarşik olmayan kümeleme yöntemleri ise maliyet fonksiyonunun kategorik veriler ile çalışamaması nedeniyle tüm veri tipleri için kullanılamamaktadır. Yalnızca kategorik veriler için ve hibrit veriler için bazı hiyerarşik olmayan kümeleme yöntemleri de son zamanlarda geliştirilmiştir. Bunun yanında verideki özelliklerin kümeleme işlemindeki ağırlıkları, verinin doğası gereği veya kümeleme sonucunda elde edilmek istenen sonuçlara bağlı olarak, farklı olabilir. Bu dokümanda, büyük hibrit verilerin etkili bir şekilde kümelenmesi ve bu kümeleme işlemine özelliklerin ağırlıklarının da dâhil edilebilmesi için geliştirilen bir algoritma tanıtacağız. Temel olarak K-Prototypes algoritmasına dayanan bu algoritmayı W-K-Prototypes olarak isimlendireceğiz.

Anahtar Kelimeler: Veri madenciliği, kümeleme, kümeleme analizi, kümeleme yöntemleri, hibrit veri kümeleme, K-Prototypes algoritması, ağırlıklandırılmış özellikler

ABSTRACT

CLUSTERING ANALYSIS IN DATA MINING AND AN ALGORITHM ON CLUSTERING HYBRID DATA

Osman ÇÖREKCİ

Department of Mathematical Engineering

MSc. Thesis

Adviser: Assoc. Prof. Ayla ŞAYLI

Separating large data into similar clusters is one of the basic problems of data mining. Storing large data in an organized way has currently increased the importance of the methods developed for clustering. Even if the hierarchical clustering methods give effective results, they are still inadequate due to their computational complexity. Non-hierarchical clustering methods cannot be used for all data types because of the cost function which cannot run by categorical data. Recently, some non-hierarchical clustering methods have been improved for categorical and hybrid data. In addition, the weights of attributes in clustering might be different due to the nature of the data or the expected results. In this paper, we introduce an algorithm which has been improved for the clustering of large hybrid data in an effective way that also includes the weights of attributes. This algorithm, mainly based on the K-Prototypes algorithm, will be called "W-K-Prototypes". The computational results show that the algorithm can be used efficiently for clustering.

Keywords: data mining, clustering, clustering analysis, clustering methods, clustering hybrid data, K-Prototypes algorithm, weighted attributes

BÖLÜM 1

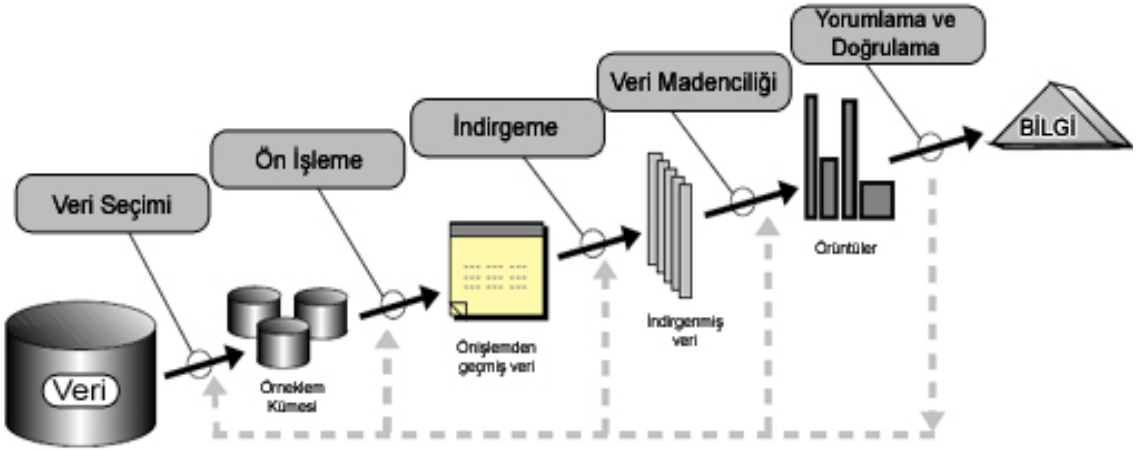
GİRİŞ

1.1 Literatür Özeti

Günümüzde bilişim sistemlerinin hayatımızın her alanında aktif rol oynaması ile birlikte veri, enformasyon, bilgi gibi kavramlara fazlasıyla aşina olmuş durumdayız. Bu kavramlar günlük konuşmada farklı anlamlarda da kullanılsa da, birer terim olarak bakıldığında bizim çalışmamız için anahtar kelimeler olarak karşımıza çıkarlar. Öğüt'ün [1] yaptığı tanımla veri, kayıt altına alınmış her türlü olay, durum, fikirdir. Veri dediğimizde temel olarak işlenmemiş ham kayıtlardan bahsediyoruz. Fakat düzenlenmiş, işlenmiş bir veri de farklı bir amaç için kullanılmak üzere veri halini koruyacaktır. İşlenmiş verinin, ilişkilendirilmesi, anlamlandırılması ile elde edilen sonuca ise enformasyon denir. Bilgi ise enformasyonun birey tarafından algılanması, özümsenmesi ve yorumlanması ile oluşturulan üçüncü aşamadır. Buradan açıkça söyleyebiliriz ki diğer süreçlerde de olmasına rağmen, bireyin kişisel yetenekleri, konu hakkındaki tecrübesi ve yaratıcılığı gibi nitelikleri, bilgi aşamasında oldukça etkili rol oynamaktadır. Çalışmamızın devamında da sıkça değineceğimiz üzere, veriden bilgiye giden yolda hemen her aşamada kaliteli ve etkili bir çıktı alınabilmesi için, her ne kadar bilişim sistemleri gelişmiş olsa da, kişisel yetenekler ve uzmanlık derecesi en etkili sistem girdisi olacaktır.

Kişisel bilgisayarlar, mobil cihazlar, akıllı saatler ve gözlükler derken tek bir insan bile bir gün içinde on binlerce satırlık veri üretir hale gelmiştir. Bu verileri anlamlandırmak ve bilgi üretmek hem ihtiyaç hem de büyüyen hizmet sektörü için ileri bir adım haline gelmiştir. Veriden bilgi çıkarmak sürecine literatürde veri madenciliği denilmektedir. Frawley veri madenciliğini "Daha önceden bilinmeyen ve potansiyel olarak yararlı olma

durumuna sahip verinin keşfedilmesi" olarak tanımlamıştır. Berry ve Linoff [2] bu kavrama "Anlamlı kuralların ve örüntülerin bulunması için geniş veri yığınları üzerine yapılan keşif ve analiz işlemleri" şeklinde bir açıklama getirirken, Sever ve Oğuz [3] çalışmalarında veri madenciliği hakkında "Önceden bilinmeyen, veri içinde gizli, anlamlı ve yararlı örüntülerin büyük ölçekli veritabanlarından otomatik biçimde elde edilmesini sağlayan veri tabanlarında bilgi keşfi süreci içerisinde bir adımdır." tanımını kullanmışlardır. Öğüt'ün [1] yaptığı daha genel bir tanımda ise veri madenciliği "Geniş veri yığınları içerisinde, yararlı olma potansiyeline sahip, aralarında beklenmedik / bilinmedik ilişkilerin olduğu verilerin keşfedilerek, veri sahibi için hem anlaşılır hem de kullanılabilir bir biçime getirilmesine yönelik geliştirilmiş yöntemler topluluğudur."



Şekil 1.1 Bilgi keşfi sürecinde veri madenciliğinin yeri [1]

II. Dünya Savaşı sırasında ABD ordusu için geliştirilen ve 30 ton ağırlık ve 170 m2lik alan hacmiyle ilk bilgisayar ENIAC'ı ve bugün cebimizdeki akıllı telefonları düşündüğümüzde bilişim teknolojisinin kat ettiği yolu görebiliriz. Üretilen ilk bilgisayarlar karmaşık hesaplamalar yapması için tasarlanmıştır. Bilgisayarların aktif ve verimli bir biçimde kullanılmaları ise, ortaya çıkan verinin depolanması ihtiyacını ortaya çıkarmıştır. Verilerin depolanmaya başlanması ile birlikte veritabanı kavramı ortaya çıkmıştır. Ancak bu, hızlı ve bol miktarda üretilen verinin tutulacağı ortamların da genişlemesini gerektirdi. Veri ambarı kavramı hemen hemen bu zamanlarda ortaya çıkmış bir kavramdır.

Veri ambarı, sadece organizasyonlarda bulunan işlemsel düzeydeki personele değil, karar verici (yönetici) kişiler için de tasarlanmış ortamdır, denilebilir. Esasen veri ambarı kuruluş çapında raporlama ve analiz için tutarlı bir kaynak sağlayan veri tabanıdır.

Günümüz koşullarında veri tabanı sistemlerinin karar destek uygulamalarında etkinliği tartışılmaktadır [4]. Böylelikle veri tabanın karşılaşılan ilgilenilen verinin boyut sorunu veri ambarı ile çözülebilmektedir [5].

Hiyerarşik, şebeke modeli, varlık-ilişki derken günümüzde yaygın olarak kullanılan İlişkisel ve nesne yönelimli veri gibi bir takım veri depolama modelleri ortaya çıkmıştır. Fakat zamanla verinin saklanmasından ziyade bu kadar çok veriden istenilen bilginin bulunup çıkarılması bir sorun haline gelmiştir. Veri madenciliği kavramsal olarak 1960'lı yıllarda, bu veri analizi ihtiyacı sonrasında ortaya çıkmıştır. Fakat o dönemki isimlendirme veri taraması (data dredging), veri yakalanması (data fishing) şeklinde idi. 1990'lı yıllarda ise bilgisayar mühendisleri tarafından bir isim olarak veri madenciliği kullanılmıştır. Bu camianın amacı, geleneksel istatistiksel yöntemler yerine, veri analizinin algoritmik bilgisayar modülleri tarafından değerlendirmesini vurgulamaktı. Bu noktadan sonra bilim adamları veri madenciliğine çeşitli yaklaşımlar getirmeye başladılar. Bu yaklaşımların kökeninde istatistik, makine öğrenimi (machine learning), veritabanları, otomasyon, pazarlama, araştırma gibi disiplinler ve kavramlar yatmaktaydı [1].

Veri madenciliği, istatistik yöntemlerini ağırlıklı kullanmaktadır. İstatistik çalışmalarına hız veren bir alan olarak da yine kendisi vardır ve bu anlamda veri madenciliği ve istatistik iç içedir diyebiliriz. Bilgisayarların hızlı gelişimi, bu iki alanında hızlı gelişmesine yol açmıştır. Üretilen verilerden bilgi çıkarma ihtiyacı ve bilgisayarların zorlu istatistiksel işlemleri basitçe yapabilmesi bu alanlar için itici güç olmuştur. Çalışmamızın ilerleyen bölümlerinde bu ilişkiye ve istatistiksel bazı yöntemlere yer vereceğiz. Veri madenciliği ile çift taraflı etkileşim halinde olan diğer alanlar ise makine öğrenmesi ve yapay zekâ çalışmalarıdır. Bunun yanında tıp, biyoloji, ziraat, ekonomi, kurumsal kaynak planlama işlemleri, müşteri ilişkileri gibi alanlar başta olmak üzere verinin olduğu hemen her alanda veri madenciliği kullanılmaktadır. Şirketlerin başarılı bir yönetim için nitelikli bilgiye duydukları ihtiyaç da [5] veri madenciliğinin yönetim süreçlerine de dâhil edilmesi ile sonuçlanmıştır. Veri madenciliğini uygulamadaki amaç, bir firmanın, işletmenin ya da kurumun sahip olduğu veriden faydalı olabilecek bilgilere ulaşması böylelikle kaynakları daha etkin kullanmak, tedarik noktalarını ya da potansiyel yatırımları, faydaları, durumları ortaya çıkartarak mikroekonomi oluşturmaktır. Dolayısıyla, pazar payı,

müşteri memnuniyeti, karlılık gibi bir şirketteki kritik noktalar hakkında bilgi sahibi olunabilir ve daha esnek, etkin kararlar alınabilir. Dünyada veri madenciliğini kullanan birçok şirket vardır ve bazıları şunlardır: Amazon.Com, APEX, Barclays, Carrefour, CopenhagenEnergy, FORTIS BANK, IMS Health, JCB CO, Laurentian Bank, Moscow City Telephone Network, Kuzey Carolina Eyalet Hazine Dairesi, Octopus, Office Depot, OneWorldHealth, Siemens, Sovereign Bank, Vodafone, Hepsiburada.com, Migros v.b. Türkiye'de ise veri madenciliğini kullanan şirketlerden bazıları şunlardır: Aviva Sigorta, AXA Sigorta, Garanti Bankası, Eureko Sigorta, HSBC Bank A.Ş., Kıraça Şirketler Topluluğu, Türk Ekonomi Bankası, Turkcell, Yapı Kredi, v.b. [5]

Veri madenciliğinin bir diğer kullanım amacı da aykırı verilerin tespitidir. Bu sayede kredi kartı dolandırıcıları, sigortacılık sektöründe yapılan hileler, hasta verilerindeki uç değerler tespit edilip bir takım tedbirler geliştirilebilir. Bunun yanı sıra marketlerin müşterilerini sınıflandırması ve müşteriye özel reklam ve promosyonlar geliştirmesi, raf düzenini en sık beraber tüketilen ürünlere göre düzenlemesi bilinen veri madenciliği uygulama örnekleridir. Veri madenciliği dünyasında sıkça bilinen bir örnek olarak Amerikan perakende zinciri Target, tükettiği ürünlere bakarak, hamile olduğunu ailesinin bile bilmediği bir müşterisine hamile ürünleri içeren bir reklam broşürü yollamış ve tahmininde yanılmamıştır.

Veri madenciliğinin temel tanımından yola çıkacak olunursa, veri madenciliğinin bir tahminleme ya da ispatlama yöntemi değil, daha önceden bilinmeyeni ortaya çıkarmasıyla diğer benzetilen analiz yöntemlerinden farklı olduğu görülür [6].

Veriden bilgiye giden yolda çeşitli yol ve yöntemler geliştirilmiştir. Bunlardan biri de verilerin arasındaki anlamlı ilişkilerden yararlanılarak, kullanıcıya potansiyel bilgi barındıran bir çıktı olarak, farklı özellikteki verileri farklı aynı özellikteki verileri ise aynı kümede olacak şekilde bir grup küme sunmaktır. Bu işleme kümeleme analizi denir ve veri madenciliğinin en geniş başlıklarından birini oluşturur. Kümeleme analizi ilk kez 1939 yılında Tryon tarafından kullanılmıştır. 1960'lı yıllardan sonra kullanımı yaygınlaşmıştır. 1963 yılında Robert Sokal ve Peter Sneath'in yazdığı "Sayısal Sınıflandırma İlminin Temelleri" adlı kitap bu alanda önemli bir adım olmuştur [7].

Kümeleme analizi için yapılan bazı tanımlar şunlardır;

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 [8].

Kümeleme analizi, elde bulunan veri yığınını belirlenen yöntemlerle analiz ederek daha önceden etiketleri belli olmayan gruplara ayrılması işlemidir. Bu işlem sonucunda elde edilen kümeler yüksek düzeyde küme içi homojenlik ve kümeler arası heterojenlik gösterirler [9].

Kümeleme analizi, birey veya nesneleri benzerliklerine göre kümelere veya gruplara ayırmak için kullanılan birçok değişkenli istatistik analiz tekniğidir [10]. Kümeleme analizi sonucu oluşturulan kümeler içinde aynı küme içinde yer alan birimler birbirlerine diğer kümenin içinde yer alan birimlerden daha çok benzeşirler.

X birimler üzerinden alınan gözlem değerlerinden oluşan veri matrisi olmak üzere kümeleme analizi, veri matrisini oluşturan ve doğal gruplamaları kesin olarak bilinmeyen birimleri birbirleri ile benzer olanları alt kümelere ayırmaya yardımcı olan yöntemler topluluğudur [11].

Bu tanımlardan da yola çıkarak kümeleme analizinin amacının, dağınık veri topluluğunu mantıksal gruplara ayırmak, bu sayede gruplar hakkında bir ön tahmin ve değerlendirme yapmak, aykırı değerleri tespit etmek olduğunu söyleyebiliriz.

Sosyal bilimler, tıp, ziraat başta olmak üzere tüm mühendislik bilimlerinde yaygın uygulama alanı bulunan kümeleme analizi, çok değişkenli varyans analizi, lojistik regresyon analizi, çok boyutlu ölçekleme gibi diğer çok değişkenli analizlerle de sıkı ilişkisi olan bir tekniktir.

Kümelemeyle günlük yaşantımızda sık sık karşılaşılabilir. Örneğin bir restoranda aynı masada oturan insanlar bir küme olarak sayılabilir. Bu tür örnekler çoğaltılabilir. Biyologların farklı hayvan türleri arasında anlamlı bir tanımlama yapmadan önce, hayvan türlerini doğru sınıflandırılması gerekir. Kısaca işin niteliği ne olursa olsun bir araştırmada ya da diğerinde bir kümeleme problemiyle karşılaşmak kaçınılmazdır [12].

Birçok araştırma alanında araştırmacıların karşılaştığı en genel sorun, gözlenen verileri anlamlı olarak nasıl organize (sınıflandırılacağı) edileceğidir. Bunun için değişik sınıflandırma teknikleri geliştirmek gerekmektedir. Başka bir ifadeyle, kümeleme

analizinin en genel amacı, aynı gruba ait olan verilerin arasında maksimum benzerliğin olması, diğer gruplardaki verilerle minimum benzerlikte olmasını sağlamaktır. Kümeleme teknikleri, araştırma problemlerinde geniş bir alanda uygulanır [13]. Tıpta hastalıkları, hastalık tedavilerini sınıflandırmada kümeleme çok kullanışlı bir yöntemdir. Böylelikle "herhangi bir hastalık hangi grupta, o grubun belirtileri nelerdir ve tedavi yöntemleri nelerdir?" bunları bilmek tedavi sürecinde fayda sağlar. Tıbbın psikolojiyle ortak alanı olan psikiyatri dalında, paranoya, şizofren, manik depresif gibi önemli hastalıkların belirtilerinin doğru kümelerde teşhis edilmesi, doğru tedavi için gereklidir. Sosyal bilimlerde de kümele analizi teknikleri uygulanmaktadır. Örneğin suç istatistiklerinde suçları sınıflandırmada kullanılabilir. Kümeleme analizinin en son uygulama alanı ise veri madenciliğidir, veri madenciliğinde büyük veri yığınlarından kurtulup özet veriler elde etmek hedeflenir. Bu aşamada kümeleme analizi kullanılır.

Veri madenciliği yapmak için kullanılan yöntem ne olursa olsun verinin bilgi keşfine hazırlanması süreci hemen hemen aynıdır. Verinin temizlenmesi, düzenlenmesi, indirgenmesi gibi aşamalardan oluşan veri hazırlama aşaması bir sonraki bölümde ele alınacaktır. Bölüm 3 de bu çalışmada kullanılacak notasyon tanıtılacak ve bu aşamada da geleneksel bazı isimlendirmelere sadık kalmaya çalışılacaktır. Bölüm 4 de kümeleme analizi ele alınacak, kümeleme analizi yapabilmek için kullanılacak materyallere de gerektiği kadar girilecektir. Kümeleme teknikleri de ortak bir örnek veri üzerinden uygulanarak basitçe anlatılacaktır. Bu çalışmada, kümeleme tekniklerinden bizi en çok ilgilendiren K-Means kümeleme yöntemi için ise ayrı bir başlık açılacak ve Bölüm 5 de detayları ile anlatılmaya çalışılacaktır. Bölüm 6 da ise çalışmamızın esas başlığı olan kategorik verilerin hızlı kümelenmesi adına alternatif bir algoritma; W-K-Prototypes tanıtılacak, avantajları ve dezavantajları ile ele alınacaktır. Bu algoritmanın uygulama sonuçları ve muadilleri ile kıyaslanma sonuçlarına da Bölüm 7 da yer verilecektir. Son olarak tanıtılan bu algoritma ile ilgili ileride yapılabilecek çalışmalara yine Bölüm 7 de yer verilecektir.

1.2 Tezin Amacı

Bu çalışmada veri madenciliği konusunda genel bilgiler verildikten sonra, kümeleme analizi hakkında detaylı bilgiler verilmiş ve özellikle hibrit verilerin kümelenmesi

konusunda çalışmalar yapılmıştır. Çalışmada amaçlanan, kümeleme yöntemleri arasındaki farklılıkları ortaya koyarak hibrit verilerin daha etkili kümelenebilmesinin yöntemlerini araştırmaktır.

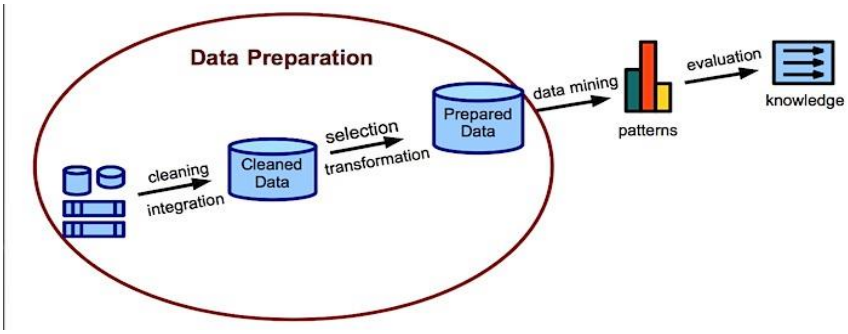
1.3 Hipotez

Veri madenciliğinin en önemli başlıklarından biri olan kümeleme analizinde yapılan çalışmalar genellikle sadece nümerik veriler ya da sadece kategorik veriler üzerinde yoğunlaşmıştır. Fakat gerçek hayatta elde edilen veriler çoğu zaman hem nümerik hem kategorik veriler içermektedir. Bu çalışmada özellikle bu tür hibrit veriler üzerine yapılan çalışmalar ve elde edilen sonuçlar gösterilmiş ve şimdiye kadar ki en iyi hibrit veri kümeleme algoritması olan K-Prototypes algoritması verinin özelliklerinin ağırlıklarına göre kümelenebilmesi için genişletilmiştir. Tezin ana konusu olan bu genişletilmiş algoritma W-K-Prototypes olarak isimlendirilmiştir.

BÖLÜM 2

VERİ HAZIRLAMA İŞLEMLERİ

İlk bilgisayar ENIAC'dan cebimizdeki akıllı telefonlara gelen süreçte, bilgi çıkarılma potansiyeline sahip verinin miktarı da bir hayli artmıştır. Artık sadece veri demenin yetmediği günümüzde büyük veri (big data) kavramı kullanılmaya başlanmıştır. İşletmelerde, hastanelerde, kamu kurumlarında, internet üzerinde sürekli veri üretimi ve bu veriden anlamlı bilgiler çıkarma ihtiyacı süregelmiştir. Ancak bu kadar çok veriden anlamlı bilgi çıkarma süreci, verinin miktarına ve özelliklerine bağlı olarak bize bir takım maliyetler getirmektedir. Üstelik süreçteki adımlardan biri dahi ihmal edilse sonuçların doğruluğu bile tartışılır duruma gelir. İşte bu süreçte belki de en önemli adım, veri madenciliği için kullanılacak verinin hazırlanması işlemidir. Örneğin, bir şirketin çalışanlarının verileri üzerinden yapılacak bir bilgi keşfi sürecinde, çalışan yaşının 1093 olarak girilmesi, bir marketin veritabanı üzerinde yapılacak veri madenciliği işleminde ürün barkodlarının yanlış girilmesi veya bir ürün ile ilgili özelliklerin bazılarının girilmemiş olması bilgi keşfi sürecinde sıkıntılara neden olacak, bir sonuç elde edilse bile doğruluğu tartışılır duruma gelecektir. İşte bu sıkıntıların giderilmesi için bazı veri işleme süreçleri geliştirilmiştir. Burada kavramsal olarak bu konulara değineceğiz.



Şekil 2.1 Veri madenciliği sürecinde veri hazırlamanın yeri [14]

2.1 Verinin Temizlenmesi

Verinin bilgi keşfine uygun hale getirilebilmesi için eksik verileri, aykırı değerleri ve yanlış girilmiş değerleri saptayıp temizlemek gerekir. Eğer bu işlemlerin yapılması zorsa ilgili kaydı veriden çıkarmak da bir yöntem olabilir [15].

2.1.1 Kayıp Veri

Veri akışının yoğun olduğu durumlarda sıkça karşılaşılan bu durum, bir kayıttaki değerlerden bazılarının veya birinin girilememesi ile oluşur. Bu durumla başa çıkabilmek için bazı yöntemler önerilmiştir.

- **Kaydın Çıkarılması**: Eksik veri girişine sahip kaydın veriden çıkarılmasıdır. Bu durumdaki kayıt sayısı verinin genel yapısını bozacak kadar çoksa tavsiye edilmez. Örneğin hasta kayıtlarından oluşan bir veride, tansiyon değeri girilmemiş kayıtlar çoğunluğu oluşturuyorsa, bu kayıtların çıkarılması diğer özellikler hakkında ortaya çıkacak bilgiyi de tartışılır hale getirebilir.
- **Kayıp Verilerin Elle Girilmesi**: Veri setindeki eksik değerlerin bir kullanıcı tarafından elle girilmesi durumudur. Eğer kullanıcı veri setini oluşturan kayıtlar ve değerler konusunda uzman ise faydalı olabilir. Ancak veri setindeki eksik verilerin çok olması durumunda ve kullanıcının uzman olmaması durumunda uygulanması zordur.
- **Genel Bir Sabit Değer Atanması**: Eksik değerlerin yerine, özellikle uyuşan sabit bir değer girilmesidir. Örneğin, Kimlik Numarası eksik kullanıcıların kimlik numarasına 99999999999 değerinin girilmesidir. Ancak bu durumda yeni bir değişken daha veriye eklenmiş olur. Sonucu olumsuz etkileme eğilimi çok fazladır.
- **Parametre Ortalamasının Atanması**: Eksik değerin yerine, eksik değerin ait olduğu özelliğin ortalamasının girilmesidir. Bunu yapmak sonucun ortalamasını etkilemeyeceği için güzel bir çözümdür. Fakat gerçek değer olmadığı için sonucun doğruluğunu yine garanti etmez. Örneğin eksik değerlerin gerçekte olması gereken ortalama 100 iken var olanların ortalaması 30 girildiğinde, eksik verilerin oranı nispetince sonucu olması gerekenden uzaklaştıracaktır. Buna rağmen eksik değerlerin temizlenmesi adına en sağlıklı işlemlerden biridir.
- **İlgili Sınıfın Ortalamasının Atanması**: Bir önceki maddeden farklı olarak özelliğin tümünün değil, benzer kayıtların ilgili özelliğinin ortalaması bulunarak bu değerin

girilmesidir. Örneğin, öğrenci bilgilerinin olduğu bir veride, bazı öğrencilerin yaş değerleri girilmemiş olsun. Tüm öğrencilerin yaş değerlerinin ortalaması yerine, aynı seviyedeki (ilkokul, ortaokul, lise vb.) öğrencilerin yaş ortalaması bulunarak bu değerin girilmesidir. Bu da sonuca olan etkisi açısından bakıldığında en sağlıklı yöntemlerden biridir.

- **Gerçekleşme Olasılığı En Yüksek Olan Değerin Atanması**: Veri setindeki verilerin frekans tablosu hazırlanarak olasılıksal olarak ilgili parametrede bulunan en yüksek olasılıklı değer boş hücrelere atanır. [16] Diyelim ki bir hastanın hangi polikliniğe geldiği bilinmiyor, bu durumda tüm hastalar içinden polikliniğe başvuru oranları bulunarak olasılıksal olarak en yüksek değere sahip olan bölüm boş hücrelere atanır.

Yukarıdaki yöntemler kayıp veri problemini gidermek için en sık kullanılan yöntemlerdir. Bunlardan sabit değer ataması ve gerçekleşme olasılığı en yüksek değerin atanması işlemleri sonucu olumsuz etkileme ihtimali en yüksek seçeneklerdir. Bunların dışında regresyon analizi, karar ağaçları, zaman serisi analizi, Bayesyen sınıflandırma yöntemleri de kullanılabilir.

2.1.2 Gürültülü Veri

Gürültülü veri, veri setindeki uç değerler, kişisel ölçüm yanlışlarından doğan veya hatalı veri girişi yapıldığında karşılaşılan veriler olarak tanımlanabilir. [17] Gürültülü verinin oluşmasının sebebi olarak; kullanıcının yanlış girmesini, verilerin eski olmasını, birim farklılıklarından kaynaklanan sıkıntıları, veri tabanının veri için uygun tasarlanmamasını sayabiliriz.

Sağlıklı bir veri madenciliği süreci için bu gürültülü verilerin giderilmesi şarttır. Gürültülü verilerin düzgünleştirilmesi (smoothing) için en bilinen yöntemler; kutulama (binning), doğrusal regresyon ve kümelemedir.

Örnek bir veri seti olarak; X: {1, 2, 4, 7, 9, 18, 19, 29, 130} 'ı dikkate alalım.

- **Kutulama**: Kutulama yönteminde gürültülü kayıtları da içeren veri, sıraya dizildikten sonra belirli sepetlere ayırılır ve daha sonra aynı sepetteki değerler sepete göre yeniden hesaplanır. Bu hesaplama yöntemi üç farklı şekilde yapılabilir. Burada dikkat çekilmesi gereken nokta sepet boyutlarının veri hakkında bilgisi olan bir kullanıcı

tarafından makul bir şekilde belirlenmesi gerektiğidir. Yukarıda bahsi geçen üç farklı hesaplama yöntemi şunlardır;

- **Ortalamaya Göre Kutulama**: Öncelikle veri seti eşit eleman sayısına sahip olacak şekilde alt kümelere ayrılır. Daha sonra her kümenin ortalaması bulunarak, kümedeki elemanların hepsine bu ortalama değer yazılır.

  X_1: {1, 2, 4} → (1+2+4)/3 = 2.33

  X_2: {7, 9, 18} →(7+9+18)/3 = 11.33

  X_3: {19, 29, 130} →(19+29+130)/3 = 59.33

  olarak hesaplandığından yeni veri kümemiz şu hali alır.

  X: {2.33, 2.33, 2.33, 11.33, 11.33, 11.33, 59.33, 59.33, 59.33}

- **Sınır Değerlerine Göre Kutulama**: ilk yöntemden farklı olarak, ortalama hesaplamak yerine, alt kümedeki alt ve üst sınırdaki elemanların farkının, alt kümenin eleman sayısına bölünmesi işlemi yapılır.

  X_1: {1, 2, 4} → (4-1)/3 = 1

  X_2: {7, 9, 18} →(18-7)/3 = 3.66

  X_3: {19, 29, 130} →(130-19)/3 = 37

  olarak hesaplandığından yeni veri kümemiz şu hali alır.

  X: {1, 1, 1, 3.66, 3.66, 3.66, 37, 37, 37}

- **Medyana Göre**: Alt kümenin eleman sayısı n olmak üzere, n tek ise (n+1)/2'inci eleman, n çift ise n/2 ve (n/2)+1'inci elemanların ortalaması alınarak kutudaki tüm elemanlara bu değer yazılır.

  X_1: {1, 2, 4} →2

  X_2: {7, 9, 18} →9

  X_3: {19, 29, 130} →29

  olarak hesaplandığından yeni veri kümemiz şu hali alır.

  X: {2, 2, 2, 9, 9, 9, 29, 29, 29}

- **Kümele Yöntemi İle Düzgünleştirme:** Veri kümelendikten sonra hiçbir kümeye ait olmayan aykırı değerler, en yakın oldukları kümenin ortalaması ile değiştirilebilirler.

Veya en yakın oldukları kümenin en küçük veya en büyük veya medoid elemanına eşitlenebilir. Kümeleme işlemi aykırı değer veriden çıkarıldıktan sonra yapılmalı ve daha sonra aykırı değerin bu kümelere olan uzaklığına bakılmalıdır. Çünkü aykırı değer dahi olsa veride bulunan tüm elemanlar kümeleme işleminde kendilerine zaten bir küme bulacaklardır.

Fourier Transformasyon Yöntemi ’de düzgünleştirme için kullanılabilir.

2.2 Verinin Bütünleştirilmesi

Üzerinde çalışılmak istenen veri her zaman tek bir tablodan, tek bir kaynaktan elde edilemeyebilir. Böyle durumlarda veriler birleştirilir. Veri birleştirme işlemlerinde birimlerin farklı olmasından, veri tabanındaki ilişkilerin doğru kurulmamış olmasından, aynı verinin tekrarlamasından veya veri türlerinin farklı olmasından kaynaklı bazı sıkıntılar olabilir. Örneğin, cinsiyet alanının birleştirilecek verilerden birinde erkek/kız diğerinde bay/bayan olarak girilmesi durumunda ortaya dört cinsiyet çıkacaktır. Başka bir örnek olarak alan bilgisinin bir veride santimetre kare diğer veride metre kare cinsinden girilmiş olması bu verilerin birleştirilmeden önce bir ön işleme tabi tutulması gerektiğini göstermektedir.

Veri türlerinin farklı olması da en sık karşılaşılan problemlerden biridir. Kilo bilgisinin bir veride integer diğer veride float olarak tutulması verilerin birleştirilmesi esnasında veri kaybına yol açabilir.

Verilerin birleştirilmesi esnasında tekrarlama problemini aşmak için de önerilen bazı yöntemler vardır. Eğer veri sürekli sayısal bir veri ise korelasyon analizi, kategorik bir veri ise Ki-Kare uyum analizi uygulamak en bilindik yöntemlerdir.

Bunun yanında türetilmiş veriler de tekrarlı kayıt problemine dâhil edilebilir. Mevcut özelliklerden oranlama, fark alma gibi bir takım işlemler sonrası elde edilebilecek bir özelliğin kendisinin veride bulunması veri madenciliği sürecini yavaşlatacaktır. Örneğin, doğum tarihinin olduğu bir veride yaş özelliğine gerek yoktur. Veya boy ve kilonun olduğu bir veride kilo/boy endeksi özelliğinin olmasına da gerek yoktur. Böylesi bir durumda veriden hangi özelliğin/özelliklerin çıkarılacağına, sonuca olan katkıları göz önüne alınarak karar verilebilir.

2.3 Verinin Dönüştürülmesi

Veri madenciliği uygulamalarının hemen hepsi uzaklık tabanlı bazı ölçüler kullanır. Bu ölçülerin gerçek dünyada anlamlı sonuçlar vermesi için verinin bazı işlemlerle dönüştürülmesi gerekebilir. Örneğin kişilerin bilgilerinin tutulduğu bir veride, ayak boyu ve bacak boyu gibi iki değişken olsun. Ayak boyları arasında ortalama 4-5 santimlik farklar olabilecekken bacak boylarında bu ortalama 15-20 santimleri bulacaktır. Böyle bir veride iki kişinin benzemesi veya benzememesine en büyük katkıyı bacak boyu yapacak ve ayak boyunun benzerlikteki etkisi bacak boyundan daha az olacaktır. Bu olumsuz etkiyi ortadan kaldırmak veri madenciliği sonuçlarının güvenirliğini arttırması açısından önemlidir. Özellikle benzerlik ölçmek için mesafe ölçümleri kullanılan süreçlerde gereklidir bile diyebiliriz. Bu dezavantajı ortadan kaldırmanın farklı yöntemleri vardır. Bunlardan en sık kullanılan birkaç yöntemi aşağıda açıklıyor olacağız.

2.3.1 Min-Max Normalleştirmesi

Verinin bir özelliğindeki değerleri 0-1 arasına sıkıştırmak için kullanılan yöntemdir. Örneğin kişi bilgilerinin tutulduğu bir veride, kilo özelliğinin değerleri 60 ile 95 arasında değişiyor olsun. Bu değerleri 10 ile bölmek, değerleri 6 ile 9.5 arasına sıkıştırmak için bir yöntem olabilir fakat verinin bütün özellikleri düşünüldüğünde, her birinde benzeri yöntemi uygulamak aslında özelliklerin veri üzerindeki ağırlıklarını değiştirmeyeceğinden iyi bir teknik değildir. Bahsettiğimiz bu kilo değerleri şu şekilde olsun; X: { 60, 62, 70, 75, 82, 91, 95 }. Bu değerleri aşağıdaki Min-Max normalleştirmesi formülünü kullanarak 0-1 arasına sıkıştıralım.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2.1}$$

Burada;

x : X kümesindeki herhangi bir eleman,

x' : x elemanının [0,1] aralığına normalleştirilmiş hali,

x_{max} : X kümesindeki en büyük değer,

x_{min} : X kümesindeki en küçük değerdir.

Buna göre X kümesini yeniden hesaplarsak X: { 0, 0.057, 0.286, 0.429, 0.629, 0.886, 1 } olur.

Min-Max normalleştirmesinin bir genişletmesi vasıtasıyla, sıkıştırma aralığı kullanıcı tarafından da belirlenebilir. Genişletilmiş Min-Max normalleştirmesi formülü şöyledir;

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} * \left(x_{yeni_max} - x_{yeni_min}\right) + x_{yeni_min} \quad (2.2)$$

Burada yukarıdaki formülden farklı olarak;

x_{yeni_max} : normalleştirme sonrası elde edilmek istenen yeni aralığın üst sınırı,

x_{yeni_min} : normalleştirme sonrası elde edilmek istenen yeni aralığın alt sınırıdır.

Örneğin, aynı X kümesini [2,7] aralığına sıkıştırmak istediğimiz de, x_{yeni_min} = 2 ve x_{yeni_max} = 7 olur. Ve X: { 2, 2.285, 3.43, 4.145, 5.145, 6.43, 7 } olarak bulunur.

2.3.2 Ondalıklı Normalleştirme

Verinin bir özelliğindeki değerleri [-1, 1] aralığına sıkıştırmak için kullanılan normalleştirme yöntemidir. Bu normalleştirme şu şekilde formülleştirilir;

$$x' = \frac{x}{10^k} \quad (2.3)$$

Burada k, $\max(|x'|) < 1$ olacak şekilde tam sayı değeridir. X: { -876, -23, 3, 17, 123, 454, 632 } olacak şekilde bir değer kümesi olsun. Burada k = 3 seçilmesi gerektiği görülür. Ondalıklı normalleştirme uyguladığımızda elde ettiğimiz yeni küme X: { -0.876, -0.023, 0.003, 0.017, 0.123, 0.454, 0.632 } olarak bulunur.

2.3.3 Z-Score Normalleştirmesi

Bu yöntem istatistiksel varsayımların tamamının sağlandığı normal dağılımda kullanılan Z değeri hesaplamasıyla aynıdır. [5] Z-Score normalleştirmesi sonucu veri tanınmayacak hale gelir. Fakat uzaklık ölçümleri kullanılarak yapılan benzerlik işlemlerinde en iyi sonucu verir. n elemanlı bir X veri kümesi ve her $x_i \in X$, $0 \leq i < n$ olmak üzere en geniş anlamda formül şu şekilde verilebilir;

$$x' = \frac{x - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)}{\sqrt{\frac{\sum_{i=1}^{n}\left(x_i - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)\right)}{n-1}}} \quad (2.4)$$

İstatistiksel işlemlerden ortalama ve standart sapma bilindiğinde ise formül şu şekilde sadeleştirilebilir;

$$x' = \frac{x - mean(X)}{sd(X)} \quad (2.5)$$

Örnek olarak X: { 1, 2, 3 } kümesini göz önüne alırsak, yeni X kümemiz X: { -1, 0, 1 } olur. Çalışmamızın esas konusu olan kümeleme analizi işlemlerinde de nümerik değerleri Z-Score ile normalleştirmek en uygun normalleştirme yöntemi olacaktır.

Ayrıca Z-Score ile normalleştirilen bir veri, Z-Score işlemi uygulanırken hesaplanan standart sapma ve ortalama değerleri kullanılarak tersine bir şekilde eski haline de getirilebilir.

$$x = x' \, sd(X) + mean(X) \quad (2.6)$$

Bu sayede Z-Score yöntemi kullanarak normalleştirilen ve bir takım işlemlerle veri madenciliği uygulanan bir verinin sonuç kümelerindeki kayıtlar, yukarıdaki formülle esas hallerine çevrilebilirler.

2.4 Verinin İndirgenmesi

Verinin indirgenmesi, özelliklerin azaltılması ve kayıtların azaltılması şeklinde iki türlü olabilir. Veri madenciliği sonucunda elde edeceğimiz model, bize bir takım tespitler ve geleceğe yönelik tahminlemeler yaptırabilmelidir. Bu modelin kalitesi kadar, büyük veriler için düşündüğümüzde, modelin hesaplanma süresi de bizim için önemli bir kriterdir. İşte bu modelin kalitesi ve sürecin hızını arttırabilmemiz için yapmamız gereken veri hazırlıklarının en önemlilerinden biri de verinin indirgenmesi işlemidir.

Veri içindeki, sonucu olumsuz etkileyecek aykırı ve gürültülü kayıtların, verinin tamamını etkilemeyecek şekilde temizlenmesi, yukarıda bahsedilen veri indirgeme yöntemlerinden biridir.

Bunun yanında bir diğer veri indirgeme yöntemi olarak özelliklerin azaltılması işlemi bizim sürecimizin hızlanmasına ve modelimizin kaliteli olmasına fayda sağlayacaktır. Örneğin, doğum günü ve yaş bilgisinin özellik olarak bulunduğu bir veriden birinin çıkarılması oluşacak modelimizi etkilemeyeceği gibi, veri madenciliği işlemimizin süresini

de kısaltır. Benzeri biçimde, bir hastane verisinde birim, yapılan işlem ve işlem sonucu gibi kategorik veriler, tek bir kategori altında birleştirilebilir.

İlişkili özellikleri birleştirmek de bir yöntem olabilir. Boy ve kilo bilgisinin ayrı iki özellik olarak bulunduğu bir veride, verinin kapsamı da göz önünde bulundurularak, boy/kilo endeksi bu iki özellik yerine tek bir özellik olarak veriye eklenebilir. "Verinin kapsamına da bakılarak" ifadesi burada çok önemlidir. Eğer veri madenciliği sonucu beklenen model, bu ikisinin ayrı iki özellik olarak değerlendirilmesini gerektiriyorsa, bu işlem bize fayda değil zarar getirebilir.

Bir verideki yüksek ilişkili (korelasyon) iki özellikten biri de, yine verinin kapsamına da bakılarak, birleştirilebilir veya sonucu etkilemesi ve süreye olacak katkısı göz önünde bulundurularak, bu iki özellikten biri feda edilebilir. Örneğin, futbol takımı verilerinin tutulduğu bir veri de atılan gol ve puan özellikleri yüksek ilişki içindeyse, bu özeliklerden birisi çıkarılabilir. Bu ilişkileri hesaplamak için birçok yöntem geliştirilmiştir. Bunlardan en sık bilineni Entropi Ölçümüdür.

Verinin indirgenmesi işlemi eğer doğru uygulanırsa bize şu faydaları sağlayacaktır;

- Veri madenciliği algoritması daha hızlı çalışır,
- Daha anlaşılır sonuçlar elde edilir,
- Daha doğru ve tutarlı sonuçlar elde edilir,

Veri toplama aşamasında bile ihtiyacımız olan esas özellikleri bize gösterme de fayda sağlar.

2.5 Veri Türleri

Verinin sahip olduğu özelliklerin türleri, veri madenciliği işlemlerinde büyük bir öneme sahiptir. Özellikle kümeleme analizi gibi mesafe ölçümlerinin kullanıldığı veri madenciliği uygulamalarında verinin türlerine göre uygun mesafe ölçüsünün kullanılması gereklidir. Bu seçim sonucu direkt olarak etkileyecektir. Veri türleri, veri toplanma türüne göre (sayma-ölçme) genel anlamda kategorik ve nümerik olarak iki başlık altında toplanabilir.

- **Kategorik Veriler:** Eşit veya eşit değil şeklinde karşılaştırılabilen veri türleridir.

- **Nominal (Atanmış)**: Göz rengi, posta kodu gibi sadece aynı veya değil denebilecek kategorik veri türüdür. Bu veri türü ile mod, entropi, çapraz korelasyon, X^2 testi gibi işlemler yapılabilir.
    - **Binary (Dikotom veya İki Sınıflı):** bu veri türü nominal veri türünün özel bir durumudur ve yalnızca 0 ve 1 değerlerinden birini alabilir. Örneğin var-yok, bay-bayan gibi iki sınıflı kategorik verilerde kullanılabilir.
- **Ordinal (Sıralı):** Sokak numaraları, eğitim seviyeleri gibi aralarında bir sıralama ilişkisi kurulabilecek kategorik veri türüdür. Bu veri türü ile medyan, yüzdelik işlemler, rank korelasyonu gibi işlemler yapılabilir.

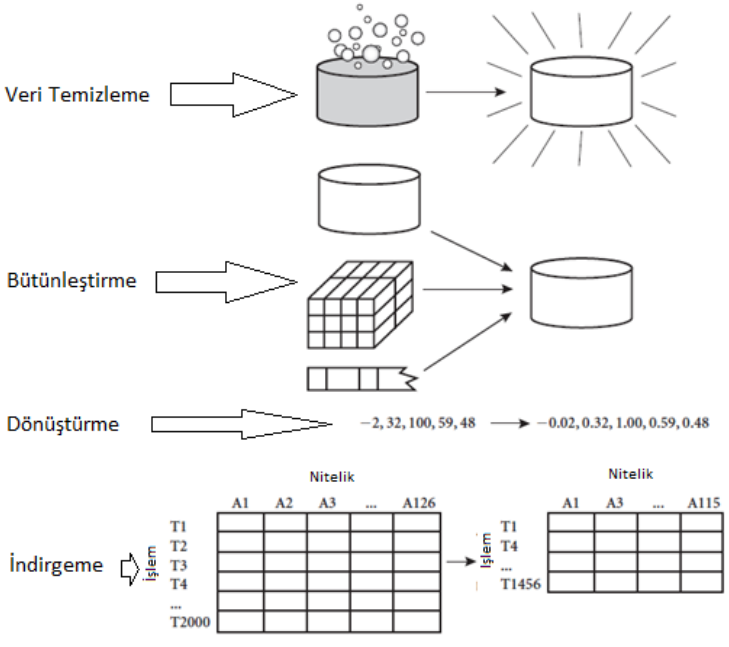
- **Nümerik Veriler:**
    - **Interval (Aralık):** Tarih, saat, sıcaklık derecesi gibi ordinal veri türüne benzeyen fakat ek olarak toplama ve çıkarma işlemlerine müsaade eden nümerik veri türüdür. Bu veri türü ile ortalama, standart sapma, Pearson korelasyonu gibi karmaşık istatistiksel işlemler yapılabilir.
    - **Ratio (Oransal):** Parasal miktar, ürün adedi, sıcaklık gibi birbirine oranlanabilir nümerik veri türleridir. Bu veri türü ile F testi, geometrik ortalama, harmonik ortalama, yüzde varyasyonu gibi işlemler yapılabilir.

Ayrıca bu veri türleri ayrık (nominal, ordinal) ve sürekli (interval, ratio) olarak da ayrılabilir. Ayrık veriler sonludur veya sayılabilir sonsuzdur. Sürekli veriler gerçek sayılardır ve ölçülebilirdiler. Özel bir durum olarak bazen kategorik veriler iki sınıflı veriler olarak temsil edilebilirler. Örnenğin göz rengi gibi tek bir özelliği şu şekilde dört ayrı özellikmiş gibi gösterebiliriz;

Çizelge 2.1 Kategorik verilerin iki sınıflı olarak gösterimi

Göz Rengi	İkilik Karşılığı
Siyah	0001
Mavi	0010
Yeşil	0100
Ela	1000

Bir verinin tüm özelliklerinin nümerik olması durumunda bu veriyi nümerik veri, tüm özelliklerinin kategorik olması durumunda kategorik veri, hem nümerik hem kategorik özelliklere sahip olması durumunda ise hibrit veri olarak isimlendireceğiz.



Şekil 2.2 Ham verinin dönüşümü [18]

BÖLÜM 3

NOTASYON

Bu çalışmamızda kullanacağımız temel notasyonu burada tanıtacağız. Gerekli diğer notasyonlar konuların içerisinde tanımlamalarla birlikte verilecektir.

$X = \{ x_i \mid 0 < i \leq n \}$ olacak şekilde n adet nesneden oluşan X kümesini göz önüne alalım. Burada ki her bir $x_i = \{ x_{ij} \mid 0 < j \leq m \}$ olacak şekilde m adet özellikten oluşur. j. özelliğin kategorik olması durumunda alabileceği değerlerin tümünü $DOM(A_j)$ kümesi ile göstereceğiz. Örneğin A_j bir göz rengi özelliği olsun. Bu durumda $DOM(A_j) = \{ Yeşil, Kahverengi, Mavi, Siyah, Ela \}$ olur.

Burada herhangi $a, b \in DOM(A_j)$ için ya $a = b$ ya da $a \neq b$ 'dir. Bir $x_i \in X$, elemanının bir gösterimi olarak $[A_{i1} = x_{i1}] \wedge [A_{i2} = x_{i2}] \wedge .. \wedge [A_{im} = x_{im}]$ 'de kullanılabilir. Özellik değer çifti şeklindeki bu gösterime *selector* denir [19]. Fakat biz daha çok $x_i = [x_{i1}, x_{i2}, \dots, x_{im}] \in X$ olacak şekilde vektörel gösterimi kullanacağız.

Eğer X hibrit bir veri ve r adet kategorik özelliğe sahip olsun. A_k'nıncı $(0 < k \leq r)$ özelliğin alabileceği farklı değerlerin sayısı n_k ile göstereceğiz. A_k'nıncı özellikte x değerinin kaç defa geçtiğini bize veren fonksiyonu $f_k(x)$ ile göstereceğiz. Eğer $x \notin A_k$ ise $f_k(x) = 0$ olacaktır. A_k özelliğinin x değerini alma olasılığını ise

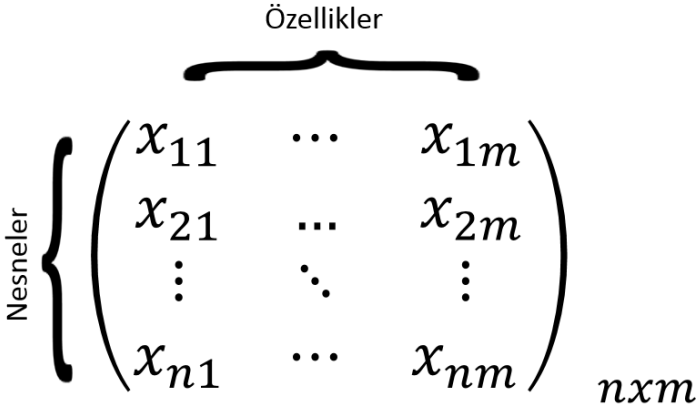
$$\acute{p}_k(x) = \frac{f_k(x)}{n} \tag{3.1}$$

olarak göstereceğiz. . A_k özelliğinin x değerini almasının olasılığını başka bir hesaplamayla

$$\acute{p}_k^2(x) = \frac{f_k(x)(f_k(x)-1)}{n(n-1)} \tag{3.2}$$

olarak da bulabiliriz.

Bazı durumlarda veri matrisinden sözü ediyor olacağız. Bu matris her bir satırı bir kayıdı ve her bir sütunu bir özelliği gösterecek şekilde verinin matris formatında yazılmış halidir. Örneğin 1000 kayıt ve 15 özellikten oluşan bir veri, 1000x15 lik bir matris ile temsil edilir.



Şekil 3.1 Verinin matris formatında gösterimi

$x_i, x_j \in X$ olmak üzere eğer her $0 < k \leq m$ için $x_{ik} = x_{jk}$ oluyorsa, $x_i = x_j$ denir. Fakat bu gerçek dünyada bu iki nesnenin aynı nesne olduğunu göstermez. Örneğin yaş, boy ve kilo özelliklerinde oluşan bir veride bu üç özelliği de aynı olan iki nesne olabilir fakat bu onların aynı nesneler olduğu anlamına gelmez. Bunun için veriler veritabanlarında tutulurken ID kolonları ile birlikte tutulurlar.

BÖLÜM 4

KÜMELEME ANALİZİ

Kümeleme analizinin geçmişi, Linnaeus'un 1753 yılında hayvanların ve bitkilerin sınıflandırılması üzerine yaptığı çalışmaya kadar dayanır [20]. 1963 yılında Robert Sokal ve Peter Sneath'in yazdığı "Sayısal Sınıflandırma İlminin Temelleri" adlı kitap bu alanda önemli bir adım olmuştur [7]. Son yıllarda da, veriden anlamlı bilgiler çıkarmak için, veri madenciliği disiplinler arası bir çalışma olarak hızla büyümüştür [21]. Kümeleme analizi, nesneleri benzerliklerine göre gruplamak ve bu gruplar üzerinden aynı gruba ait nesneler hakkında özet bilgiler elde edebilmek için yapılan, en önemli veri madenciliği süreçlerinden birisidir. Kümeleme analizi sürecinde sıklıkla istatistik işlemlerinden yararlanılır. Kümeleme analizi metotları istatistik ve bilişim disiplinlerinin arasına yerleşmiştir de denebilir [22].

Kümeleme işlemlerinde kullanılan değişkenler tabiri, özellik ve karakter terimleri ile eşanlamlıdır. Aynı şekilde gözlem terimi de veri deki kayıtları ifade etmek için eşanlamlı olarak kullanılırlar. Biz bu çalışmada verideki her bir kayıta nesne, nesnelerin sahip olduğu değişkenlere ise özellik diyeceğiz.

Kümeleme analizi ile ilgili yapılan bazı tanımlar şu şekilde sıralanabilir;

Kümelemenin en genel amacı benzeyeni benzemezden ayırmaktır [23].

Kümeleme, aynı kümedeki elemanların yüksek benzerlik oranlarına ve farklı kümedeki elemanların yüksek benzeşmezlik oranına sahip olacak şekilde nesneleri organize etmek olarak tanımlanabilir [7], [8], [24].

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 [8].

Kümeleme analizi, elde bulunan veri yığınını belirlenen yöntemlerle analiz ederek daha önceden etiketleri belli olmayan gruplara ayrılması işlemidir. Bu işlem sonucunda elde edilen kümeler yüksek düzeyde küme içi homojenlik ve kümeler arası heterojenlik gösterirler [9].

Kümeleme analizi, birey veya nesneleri benzerliklerine göre kümelere veya gruplara ayırmak için kullanılan birçok değişkenli istatistik analiz tekniğidir [10].

Sosyal bilimler, tıp, ziraat başta olmak üzere tüm mühendislik bilimlerinde yaygın uygulama alanı bulunan kümeleme analizi, çok değişkenli varyans analizi, lojistik regresyon analizi, çok boyutlu ölçekleme gibi diğer çok değişkenli analizlerle de sıkı ilişkisi olan bir tekniktir [25].

Veri madenciliği teknikleri, veriye dair geçmiş bir bilgiden istifade edilip edilmemesine göre denetimli ve denetimsiz olarak ikiye ayrılır. Kümeleme analizi denetimsiz bir veri madenciliği tekniğidir. Dolayısıyla başlangıç aşamasında sonuçta kaç kümenin oluşacağı ve hangi özelliklerin bu kümeleme işlemine etki edeceği bilinmemektedir [26]. Denetimli olarak yapılan benzer işleme ise sınıflama denir ve önceden belirlenen sınıflara göre nesneler sınıflandırılır. Örneğin yaş ve kilo özelliklerinden oluşan bir tabloda baştan kadın ve erkek olarak iki sınıfa ulaşmak amaçlanıyorsa bu sınıflandırmadır. Ancak sonuç kümelerinin etiketi bilinmeden kümeleme işi yapılıyor ve sonuç kümeleri buna göre değerlendiriliyorsa bu kümelemedir. Denetimli veri madenciliği tekniklerinde, verinin bir kısmı üzerinden bir takım ön bilgilere ulaşılır ve bu ön bilgiye dayanarak sınıflandırma işlemi yapılır. Kümeleme de böyle bir ön bilgi yoktur.

Kümeleme analizi verinin olduğu her yerde kullanılabilir. Bununla birlikte genel kullanım alanları şöyle sıralanabilir [27];

• Hayvan ve bitki ekolojilerinin araştırılması,

• Genler üzerinde; gen aileleri, değişimleri, evrimi gibi çalışmalarda,

• Görüntüleme tekniklerinden de faydalanılarak teşhis tedavi süreçlerinde,

• Sağ kalım çalışmalarında,

• Pazar araştırmalarında,

• Ürün konumlandırma,

• Pazar bölümlendirmesinde,

• Müşteri davranışı analizlerinde,

• Web ortamındaki pazarlama işlemleri, metin analizlerinde,

• Yazılım geliştirmede,

• Suçların ve sahtekârlıkların ortaya konmasında,

• Robotik biliminde, durumsal hareketlerin analizinde,

• Matematiksel kimyada,

• Atmosfer olaylarının tahmininde,

• Jeolojik araştırmalarda

Kümeleme analizi sonucunda bazı elemanlar hiçbir kümeye ait olmayabilir ya da bir küme etiketi veya eleman sayısı itibari ile diğerlerinden ayrılabilir. Böylesi durumlarda oluşan bu ayrı eleman veya küme yoluyla aykırı değerler de saptanabilir. Bu sayede kümeleme analizi özellikle bankacılık, sigorta ve finans sektöründe sahtekârlık tespiti için de kullanılabilir.

Kümeleme analizinin amacı, kullanıcının amacına bağlı olmakla birlikte genel olarak aşağıdaki şekilde verilmektedir [12];

- Uygun modeli bulmak
- Model uyumu sağlamak
- Gruplar içi ön tahmin hesapları yapmak
- Ön tahmin sınaması yapmak
- Veri yapısını bulmak
- Ön tahmini genelleştirmek
- Boyut indirgemek

Kümeleme analizinin amacını daha teknik bir şekilde ifade etmek gerekirse [28];

- n sayıda nesneyi m değişkene ilişkin özelliklerine göre, kendi içinde olabildiğince homojen ve kendi aralarında da heterojen alt kümelere ayırmak,
- m sayıda değişkeni, n sayıda nesnede saptanan değerlere göre ortak özellikleri açıkladığı varsayılan alt kümelere ayırmak ve ortak faktör yapıları ortaya koymak,
- Hem nesneleri, hem de değişkenleri birlikte ele alarak n nesneyi m değişkene göre ortak özellikli alt kümelere ayırmak,
- m değişkene göre saptanan değerler bakımından bireylerin biyolojik ve taksonomik sınıflamasını ortaya koymak

Kümeleme analizinin yöntemleri değerlendirilirken bazı avantaj ve dezavantajlarına bakılarak ihtiyaca en uygun yöntem seçilmelidir. Bir kümeleme algoritmasında başlıca şu özellikler aranır;

- **Ölçeklenebilirlik:** Seçilecek yöntem verideki kayıt sayısının artması durumunda hayal kırıklığı oluşturacak derecede olumsuz etkilenmemelidir. 100 kayıt bulunan bir veri için 1 birim sürede sonuç üreten bir kümeleme yöntemi, 1000 kayıt için anormal süre artışlarına sebep olmamalıdır.
- **Farklı Veri Türlerine Uyumluluk:** Çoğu kümeleme yöntemi nümerik veriler için tasarlanmıştır. Fakat gerçek dünya verileri her zaman nümerik olmadığı için zamanla kategorik veriler için de kümeleme yöntemleri geliştirilmiştir. Biz de çalışmamızda hibrit olarak adlandırdığımız hem nümerik hem kategorik özellik içeren veriler ile çalışabilen bir kümeleme yöntemi tanıtmayı amaçladık. Bir kümeleme yöntemi farklı veri türlerini de dikkate alarak çalışabilmelidir.
- **Gürültülü Verilerle Baş Edebilme:** Veride bulunan kayıtlar her zaman gerçek kayıtlar olmayabilir. Yukarıdaki bölümlerde de değinildiği gibi kayıp veya gürültülü kayıtlar olabilir. Her ne kadar bunlarla baş etmek zor olsa da, bir kümeleme yönteminin en azından bu değerlerden fazla etkilenmemesi beklenir.
- **Yüksek Boyutlu Verilerle Çalışma:** Bir kümeleme algoritması verideki özellik sayısının artmasından fazla etkilenmemelidir. Yukarıda verinin indirgenmesi başlığı altında incelediğimiz üzere, veriden gereksiz özelliklerin çıkarılmasının veya yüksek ilişkili

özelliklerin birleştirilmesinin veri madenciliği sürecini olumlu yönde etkileyeceğini söylemiştik. Fakat her zaman bu indirgeme işlemi az sayıda özellik kalacağını garanti etmez. Bu nedenle kümele yöntemlerinin de çok sayıda özellik ile çalışabilme yeteneğine sahip olması beklenir.

- **Girdi Değerlerinin Belirlenme Şekli:** Kümeleme işlemi başlarken, veri ile beraber kullanıcıdan beklenen girdilerin mümkün olduğunca az olması bir kümeleme tekniğinin sahip olması gereken olumlu özelliklerdendir. Çünkü kullanıcı müdahalesinin olduğu yerde her zaman hata ihtimali vardır. Hiyerarşik kümeleme teknikleri kullanıcıdan sadece veriyi alırken hiyerarşik olmayan K-Means kümeleme yöntemi veri ile beraber, istenen küme adedi ve bu adet kadar başlangıç küme merkezi girdilerini kullanıcıdan beklemektedir. Kullanıcının uzman olmaması durumunda bu girdilerin uygunluğu tartışmalı olacaktır. Bu nedenledir ki, aşağıda da değineceğimiz üzere, K-Means kümeleme tekniği üzerinde yapılan çalışmalar da bu girdilerin nasıl otomatik seçilebileceği üzerinde uzun uzadıya durulmuştur.
- **Yeni Veri Girişlerine Uygunluk:** Kümeleme yöntemlerinden beklenen bir diğer avantaj ise, yöntemin üzerinde çalıştığı verinin yeni kayıtlar ile büyümesine karşı uygun davranışlar sergilemesidir.

Bunun yanında kümeleme analizi yapan birinin, sonuçların doğruluğundan emin olmak adına, farklı yöntemlerle elde ettiği kümeleme sonuçlarını karşılaştırması da tavsiye edilmektedir [7].

Kümeleme analizinin uygulama aşamaları ise şöyle sıralanabilir [25];

1- Özelliklerin seçilmesi ve veri matrisinin oluşturulması,

2- Her bir nesnenin birbiriyle olan ilişkisini ölçmede kullanılacak benzerlik/uzaklık ölçüsünün seçilmesi,

3- Uygun bir kümeleme tekniği ile sonuç kümelerinin elde edilmesi,

4- Sonuç kümelerinin yorumlanarak, her bir küme için etiketlerin belirlenmesi.

Kümeleme analizinin değerlendirilmesi aşamasında en iyi yararı elde edebilmek için şu maddeler göz önünde bulundurulmalıdır [7];

- Kümeleme sonuçları ön tahmin sınaması için bir araçtır.

- Kümeleme sonuçları kesin sonuç olmayıp, tahmini görünümler olabilir.
- Kümeleme sonucunda verinin karmaşık yapısı ortaya çıkabilir.
- Verinin gerçek görünümü de ortaya çıkabilir.
- Tüm kayıtların tek bir kümede olması ya da her bir kayıtın ayrı birer küme oluşturması durumları ile karşılaşılabilir.

İlerleyen bölümlerde kümeleme tekniklerini ayrıntıları ile ele alacağız. Bundan önce ise her bir kümeleme tekniği için kümeleme sonucunu direkt etkileyici bir etken olan uzaklık ve benzerlik ölçülerini tanıtacağız.

4.1 Uzaklık ve Benzerlik Ölçüleri

Verideki herhangi iki nesnenin uzaklık veya benzerliğinin ölçülmesi işlemi çoğu veri madenciliği tekniği için sürecin çekirdeği durumundadır. Kümeleme, aykırı değerlerin tespiti, sınıflama v.b. Bu ölçüler kullanılarak nesneler arası mesafe matrisi oluşturulabilir. Hiyerarşik kümeleme analizi yönteminde bu matris kullanılır. Simetrik bir matristir.

Sürekli değişkenler için hemen hemen tüm istatistiksel işlemler uygulanabildiği için birçok uzaklık ölçüsü kullanılabilir. Pratikte en sık kullanılan uzaklık ölçüsü ise Minkowski uzaklık ölçüsüdür. Özel olarak Minkowski formülündeki üs derecesi 1 seçildiğinde Manhattan City Block uzaklık ölçüsü, 2 seçildiğinde ise Euclidean (Öklid) uzaklık ölçüsü elde edilir. Bu uzaklık ölçüleri ile ilgili anahtar gözlem ise, herhangi iki nesnenin uzaklığını veri kapsamını gözetmeksizin hesaplamalarıdır. Mahalanobis gibi bazı uzaklık ölçüleri ise veri kapsamını da hesaba katarak uzaklık ölçümü hesaplar.

Kategorik veriler için ise bu ölçüm sürekli verilerde olduğu kadar kolay değildir. Kategorik verilerin asıl karakteri, eşit olmadıklarında belli bir sıralama içermemeleridir. Bu yüzden farklı iki kategorik değeri direkt olarak karşılaştırmak kolay değildir. Akla ilk gelen yol, eğer aynıysa 1 farklıysa 0 sonucu dönmektir. Bu ölçü literatürde *overlap* (çakışma) olarak isimlendirilir. Bu ölçünün bilinen en büyük dezavantajı, farklı özellik değerlerine aynı davranabilmesidir. Bunun yanında sıralı kategorik verilere de genişletilememesidir [29].

Unutulmamalıdır ki uzaklık ölçüsünün seçimi kümeleme analizinin sonucunu doğrudan etkiler. Seçilen uzaklık ölçüsü çalışılacak verinin türüne de çalışma amacına da uygun seçilmelidir. Hatta kümeleme analizinin sonuçlarının tutarlılığının testi için, aynı veriyi

aynı yöntem ve farklı uzaklık ölçüleri ile yeniden kümelemek ve sonuçları karşılaştırmak da kullanılan bir yöntemdir. Öngörülebileceği gibi uzaklık ölçüsünün farklı seçilmesi sonuçların tamamen değişeceği anlamına da gelmemektedir.

Aşağıda tanıtılacak uzaklık ve benzerlik ölçülerinde aşağıda verilen tablodaki notasyon kullanılacaktır.

Çizelge 4.1 Tanımlamalarda kullanılacak notasyon

	Özellik				
Nesne	x_1	x_2	.	.	x_m
.	.	.	.	.	.
.	.	.	.	.	.
İ	x_{i1}	x_{i2}	.	.	x_{im}
J	x_{j1}	x_{j2}	.	.	x_{jm}
.	.	.	.	.	.
.	.	.	.	.	.
n	x_{n1}	x_{n2}	.	.	x_{nm}

4.1.1 Uzaklık Ölçüleri

Herhangi iki nesnenin birbirine olan uzaklığını ölçmek için kullanılırlar. Uzaklık ölçüsünün yüksek olması yüksek benzeşmezliği, düşük olması yüksek benzerliği gösterir. Farklı veri türleri için farklı uzaklık ölçüleri geliştirilmiştir. Bunların en bilinenlerini aşağıda tanıtacağız.

4.1.1.1 İkili Sınıflı Veriler İçin Uzaklık Ölçüleri

Veriler ikili veriler olabileceği gibi bazı durumlarda sürekli veya kesikli sayısal verilerde ikili veri formatın getirilebilir. Örneğin bireylerin boyunu gösteren bir veri yapısı, boyu

150 santimden uzun veya kısa olacak şekilde ikili veri haline getirilebilir. İkili verilerde 1 varlığı 0 yokluğu temsil eder.

İkili verilerin hesaplanmasında genellikle 2x2 lik bir çapraz matristen faydalanılır.

Çizelge 4.2 Çaprazlama matrisi

	Gözlem j		
Gözlem i	**1**	**0**	**Toplam**
1	a	b	a+b
0	c	d	c+d
Toplam	a+c	b+d	a+b+c+d=p

İkili veriler için birçok uzaklık ölçüsü söz konusudur. Bunun nedeni toplada gösterilen a, b, c ve d sayılarına verilen önemin ağırlığıdır. Örneğin bazı çalışmalarda İtalyanca konuşmak İtalyanca konuşamamaya göre daha kuvvetli bir benzerlik ölçüsü olarak görülebilir.

- **Öklit Uzaklık Ölçüsü**

$$d_{ij} = \sqrt{b + c} \tag{4.1}$$

- **Büyüklük Farkları Uzaklık Ölçüsü**

$$d_{ij} = \frac{(b-c)^2}{(a+b+c+d)^2} \tag{4.2}$$

En küçük değeri 0, en büyük değeri sınırsızdır.

- **Biçim Farkları Uzaklık Ölçüsü**

$$d_{ij} = \frac{bc}{(a+b+c+d)^2} \tag{4.3}$$

En küçük değeri 0, en büyük değeri 1'dir.

4.1.1.2 Sürekli Veriler İçin Uzaklık Ölçüleri

- **Öklit Uzaklık Ölçüsü**

En sık kullanılan uzaklık ölçüsüdür. Bir dik üçgenin hipotenüsünün hesaplanmasında kullanılan Pisagor bağıntısının tüm boyutlar için genelleştirilmesidir. n-boyutlu uzay için (n tane özelliğe sahip iki örnek için) formülü şu şekilde verilir;

$$d_{ij} = \sqrt{\sum_{k=1}^{m}(x_{ik} - x_{jk})^2} \quad (4.4)$$

Yukarıdaki tabloda da görüleceği gibi, burada x_{ik} ; i. örneğin k. Özelliğinin değeri anlamındadır.

Öklit uzaklık ölçüsü tanımından anlaşılacağı üzere, özelliklerden birinin ölçüm değerleri arasındaki farkların fazla olması iki değişken arasındaki benzerliği etkileyecektir. Örneğin bireylerin başparmak uzunluğu, başparmak çevresi ve boy uzunlukları özelliklerinin dikkate alındığı bir çalışmada, başparmak çevresi ve uzunluğu arasındaki farklar çok küçük olacak iken boyları arasındaki fark çok daha fazla değişkenlik gösterebilecektir. Dolayısıyla iki birey arasındaki uzaklığa, kişilerin boyları çok daha baskın etki edecektir. Bu etkinin yumuşatılması için verilerin standartlaştırılması çok sık kullanılan güzel bir çözümdür [30].

- **Kare Öklit Uzaklık Ölçüsü**

Öklit uzaklığının karesidir.

$$d_{ij} = \sum_{k=1}^{m}(x_{ik} - x_{jk})^2 \quad (4.5)$$

- **Chebychev Uzaklık Ölçüsü**

Farkların mutlak değerinin en büyüğü olarak tanımlanır.

$$d_{ij} = max_k^m|x_{ik} - x_{jk}| \quad (4.6)$$

- **Manhattan City Block Uzaklık Ölçüsü**

Farkların mutlak değerinin toplamı olarak tanımlanır. Kesikli sayısal veriler için daha çok önerilir.

$$d_{ij} = \sum_{k=1}^{m}|x_{ik} - x_{jk}| \quad (4.7)$$

- **Minkowski Uzaklık Ölçüsü**

  Aşağıdaki gibi verilir. p=1 için Manhattan, p=2 için ökliti verir. p artarken uzaklık Chebychev uzaklık ölçüsüne yaklaşır.

$$d_{ij} = \left[\sum_{k=1}^{m} \left|x_{ik} - x_{jk}\right|^{p}\right]^{\frac{1}{p}} \quad (4.8)$$

- **Karl Pearson Uzaklık Ölçüsü / Standartlaştırılmış Öklit Uzaklığı Ölçüsü**

  Öklit uzaklığındaki farkların $\frac{1}{s_k^2}$ ile düzeltilmesiyle/standartlaştırılmasıyla elde edilir. Burada s_k^2 standart sapmanın karesidir.

$$d_{ij} = \sqrt{\sum_{k=1}^{m} \frac{1}{s_k^2} (x_{ik} - x_{jk})^2} \quad (4.9)$$

- **Korelasyon Uzaklık Ölçüsü**

  Bu kapsamda 4 adet uzaklık ölçüsü vardır.

$$d_{ij} = \frac{(1 - r_{ij})}{2} \quad (4.10)$$

$$d_{ij} = 1 - \left|r_{ij}\right| \quad (4.11)$$

$$d_{ij} = 1 - r_{ij}^2 \quad (4.12)$$

$$d_{ij} = 1 - r_{ij} \quad (4.13)$$

  Bu ölçüler arasında en çok (4.13) ile karşılaşılır. Korelasyon katsayısı -1 ile 1 arasında değişir. Bu bağlamda uzaklık ölçüsü 0 ile 2 arasındadır.

4.1.1.3 Kategorik Veriler İçin Uzaklık Ölçüleri

Kategorik veriler bu çalışmada esas ilgilendiğimiz konulardan biri olduğundan, benzerlik ve uzaklık ölçülerini tek başlık altında karşılaştırmalı olarak ele almayı uygun gördük. Kategorik veriler için benzerlik ölçüleri başlığı altında gerekli tanım ve ölçüleri vereceğiz.

4.1.2 Benzerlik Ölçüleri

Herhangi iki nesnenin birbirine olan benzerliğini ölçmek için kullanılırlar. Benzerlik ölçüsünün yüksek olması yüksek benzerliği, düşük olması yüksek benzeşmezliği gösterir. Farklı veri türleri için farklı benzerlik ölçüleri geliştirilmiştir. Bunların en bilinenlerini aşağıda tanıtacağız.

4.1.2.1 İkili Sınıflı Veriler İçin Benzerlik Ölçüleri

- **Russell ve Rao Benzerlik Ölçüsü**

  Sadece "var" olarak uyuşan çiftlerin toplam içindeki payını verir.

$$Benzerlik_{ij} = \frac{a}{a+b+c+d} = \frac{a}{p} \tag{4.14}$$

- **Basit Benzerlik Ölçüsü**

  Toplam içerisinde var ve yok olarak eşleşen çiftlerin oranıdır.

$$Benzerlik_{ij} = \frac{a+d}{a+b+c+d} = \frac{a+d}{p} \tag{4.15}$$

- **Jaccard Benzerlik Ölçüsü**

  Benzerlik oranı da denir. Birlikte yok olarak eşleşen çiftleri pay ve paydada dikkate almaz. 0 ile 1 arasında değişkenlik gösterir.

$$Benzerlik_{ij} = \frac{a}{a+b+c} \tag{4.16}$$

- **Parçalı Benzerlik Ölçüsü**

  Bu ölçü Czekanowski ölçüsü veya Sorensen ölçüsü olarak da bilinir. Birlikte var olan çiftlere iki kat ağırlık verir. 0 ile 1 arasında değişim gösterir.

$$Benzerlik_{ij} = \frac{2a}{2a+b+c} \tag{4.17}$$

- **Rogers ve Tanimoto Benzerlik Ölçüsü**

  Birlikte uyuşum göstermeyen çiftlere iki katı ağırlık verilir. 0 ile 1 arasında değişiklik gösterir.

$$Benzerlik_{ij} = \frac{a+d}{a+2(b+c)+d} \tag{4.18}$$

- **Sokal ve Sneath Benzerlik Ölçüsü 1**

  Birlikte uyuşum gösteren çiftlere pay ve paydada iki kat ağırlık verilir. 0 ile 1 arasında değişim gösterir.

$$Benzerlik_{ij} = \frac{2(a+d)}{2(a+d)+b+c} \quad (4.19)$$

- **Sokal ve Sneath Benzerlik Ölçüsü 2**

  Birlikte yok olan çiftler pay ve paydada dikkate alınmazken, birikte uyuşum göstermeyen çiftlere paydada iki kat daha fazla ağırlık verilir.

$$Benzerlik_{ij} = \frac{a}{2(b+c)+a} \quad (4.20)$$

- **Sokal ve Sneath Benzerlik Ölçüsü 3**

  Birlikte uyum gösteren çiftler paydada dikkate alınmamaktadır. Bu nedenle uyuşan ve uyuşmayan çiftler eşit ağırlığa sahip olmaktadır. En küçük değeri 0 olan bu ölçünün üst sınırı yoktur. Hiç uyuşmazlık olmadığında tanımsızdır.

$$Benzerlik_{ij} = \frac{a+d}{b+c} \quad (4.21)$$

- **Kulczynski Benzerlik Ölçüsü**

  Bu ölçü, var olarak uyuşan çiftler sayısının yok olarak uyuşan çiftler toplamına bölünmesiyle elde edilir. 0 ile sonsuz arasında bir değer alır.

$$Benzerlik_{ij} = \frac{a}{b+c} \quad (4.22)$$

- **Ochiai Benzerlik Ölçüsü**

  0-1 arasında değişim gösterir.

$$Benzerlik_{ij} = \frac{a}{\sqrt{(a+b)+(a+c)}} \quad (4.23)$$

- **Yule Q Benzerlik Ölçüsü**

  Dağılım aralığı -1 ile 1 arasıdır.

$$Benzerlik_{ij} = \frac{ad-bc}{ad+bc} \quad (4.24)$$

4.1.2.2 Sürekli Veriler İçin Benzerlik Ölçüleri

- **Pearson Korelasyon Katsayısı**

Korelasyon ölçülerinde büyük değerler benzerliğin göstergesidir. Pearson korelasyon katsayısı değişken çiftleri arasındaki ilişkiyi ölçer. Satırlarla sutunların yer değiştirmesi yoluyla elde edilen korelasyon katsayısı ise değişkenlerin profilleri arasındaki benzerliği verir. Yani yapısal benzerlik söz konusudur. Bir birine en uzak iki örnek bu bağlamda yapısal olarak birbirine en yakın iki örnek olabilir.

- **Cosine Benzerlik Ölçüsü**

Aşağıdaki formül ile verilir.

$$Benzerlik_{ij} = \frac{\sum_{i,j}^{p} x_i x_j}{\sqrt{\sum_{i=1}^{p} x_i^2 \sum_{j=1}^{p} x_j^2}} \tag{4.25}$$

4.1.2.3 Kategorik Veriler İçin Benzerlik Ölçüleri

Kategorik verilerin benzerliğinin ölçümü üzerine daha önce birçok çalışma yapılmıştır. Bu çalışmaların başlıcalarını inceleyen Vipin Kumar vd.'nin "*Similarity Measures for Categorical Data A Comparative Evaluation*" çalışması [31] bu bölümde bizim için esas kaynak olacaktır.

Kategorik verilerin benzerliği kavramı sürekli verilerde olduğu kadar basit değildir. Kategorik verilerin temel karakteristiği sahip olduğu üyelerin belli bir sırasının olmamasıdır. Bu yüzden iki kategorik veriyi direkt olarak karşılaştırmak mümkün değildir. Akla gelen en basit yöntem aynı olan değerler için 1, farklı olan değerler için 0 vermektir. İki kategorik nesnenin eşleşen özelliklerin tüm özelliklere oranı olarak hesaplanan benzerlik ölçüsü de *overlap* olarak adlandırılır [29]. Bu yöntemin dezavantajlarına daha önce değinmiştik. En büyük dezavantajı ise eşleşen özellikler ile kayıp değerleri ayırt edememesidir. Bazı kategorik benzerlik ölçüleri de kategorik özellikteki değerlerin frekansını da göz önüne alır ki bu benzerlik hesaplamaya olumlu bir etki oluşturur.

Sneath ve Sokal (1973)'ın kategorik verilerin benzerliği üzerine yaptıkları çalışmalarda hesaplama karmaşıklığını da dikkate almaları bugün bile çoğu çalışmaya ilham kaynağı oluşturmaktadır. Diğer bazı çalışmalarda ([7], [13], [24]) kategorik veriyi iki sınıflı (binary) veriye dönüştürerek işlem yapma yöntemi üzerinde durulmuştur. Wilson ve Martinez

[32] hibrit veriler için uzaklık ölçüleri üzerine çalışmalar yapmışlardır. Onlar yaptıkları çalışmada denetimli veri madenciliği yoluyla elde edilen sınıf bilgilerini de ölçüm hesaplamak için kullanmışlardır. Ancak biz bu çalışmada daha çok veri dağılımına göre hesaplamalar yapan ölçülerle ilgileniyoruz. Geliştirilen bazı ölçüm teknikleri ise nesnelerin komşuluklarına göre hesaplamalar yapmaktadır. Böyle ölçülere komşuluk tabanlı benzerlik ölçüleri denir [31].

Kategorik veriler üzerine uzun süredir farklı kapsamlarda çalışılmıştır. Biz burada kategorik veriler üzerine benzerlik ölçülerini tanıtmadan önce, veri eğilimli benzerlik ölçülerine doğrudan etki eden kategorik verilerin benzerlik ölçülerinin bazı karakteristik özelliklerini sıralayacağız.

- **Verinin boyutu**: Çoğu ölçü verinin boyutundan etkilenmez. Smirnov gibi bazı ölçüler ise verinin boyutunu da dikkate alır.
- **Özellik Sayısı**: Çoğu ölçü verinin özellik sayısından etkilenmez. Benzerlik ölçüsünü özellik sayısı üzerine normalleştirir. Fakat Kumar vd. [31] kategorik verilerin benzerlik ölçüleri üzerine yaptığı çalışmada, özellik sayısının aykırı değer bulma çalışmasında etkisi olduğunu göstermiştir.
- **Her bir özelliğin alabileceği değerlerin sayısı**: kategorik verilerin benzerliğinin ölçüldüğü bazı ölçüler de bu değer dikkate alınır. Örneğin yüzlerce değer alabilen bir kategorik özelliğin eşleşmesine, birkaç değer alabilen bir kategorinin eşleşmesinden daha az önem verilir. Bizim bu çalışmada tanıtacağız alternatif K-Means algoritmasında da bu durumu dikkate alacağız. *Eskin* benzerlik ölçüsü böyle bir ölçüdür [31].
- $\boldsymbol{f_k(x)}$**'in dağılımı**: Bazı ölçülerde bir özellikteki değerlerin dağılımı da dikkate alınır. Daha az geçen bir değer daha yüksek öneme sahip olur. Örneğin, 100 kişilik bir toplulukta mavi gözlü yalnız iki kişi varsa bu kahverengi gözlü 56 kişinin birbirine benzemesine göre daha yüksek benzerlik değerine sahip tutulur.

Kategorik nesnelerin birbirine olan uzaklığını bulmak için kullanılan fonksiyonların değerini, benzerlik ölçümleri ile karşılaştırabilmek için şöyle bir dönüşüm de kullanacağız.

$$sim = \frac{1}{1+dist} \tag{4.26}$$

Kategorik veri ile ilgili verdiğimiz bu ön bilgilerden sonra, kategorik veriler için tasarlanmış bazı benzerlik ölçülerini verebiliriz. Burada vereceğimiz benzerlik ölçüleri tamamı kategorik olan kategorik veriler içindir. Hibrit veriler için bunların kullanımını ileride ele alacağız.

Hemen hemen bütün kategorik veri benzerlik ölçüleri aşağıdaki gibi genelleştirilmiş bir formülle gösterilebilir [31].

$$S(x,y) = \sum_{k=1}^{r} w_k S_k(x_k, y_k) \tag{4.27}$$

Burada $S_k(x_k, y_k)$, A_k'daki iki kategori değeri için benzerlik ölçüsüdür. w_k ise A_k özelliğine atanmış ağırlık katsayıdır. Ağırlık kavramına da ileride değineceğiz. Örneğin ölçü olarak overlap (çakışma)'yı seçersek $S(x,y)$ değeri, $x = y$ durumunda 1 diğer bütün durumlarda 0'dır. Yani bir benzerlik matrisi kurulduğunda sadece köşegen değerleri 1 ve diğerleri 0 olurdu. Kumar vd. [31], kategorik verilerin benzerlik ölçülerini bu benzerlik matrisini doldurma şekline göre 3 sınıfa ayırmıştır.

Bunun yanında bazı ölçüler ise aşağıdaki bilgileri de kullanırlar;

- $f(x)$, frekans değerlerini
- n, veri kümesinin eleman sayısını
- r, özelliğin alabileceği değerlerin sayısını

Kategorik verilerin benzerliğinin ölçümünde kullanılan benzerlik ölçülerini Kumar vd. [31] ve arkadaşlarının çalışmasındaki haliyle şu şekilde verelim.

Çizelge 4.3 Kategorik veriler için benzerlik ölçüleri

Ölçü	$S_k(x_k, y_k)$	$w_k, 0 < k \leq r$
Overlap (Çakışma)	$= \begin{cases} 1 & eğer\ x_k = y_k \\ 0 & diğer \end{cases}$	$\frac{1}{r}$
Eskin	$= \begin{cases} 1 & eğer\ x_k = y_k \\ \frac{n_k^2}{n_k^2 + 2} & diğer \end{cases}$	$\frac{1}{r}$
IOF	$= \begin{cases} 1 & eğer\ x_k = y_k \\ \frac{1}{1 + \log f_k(x_k)\, x \log f_k(y_k)} & diğer \end{cases}$	$\frac{1}{r}$
OF	$= \begin{cases} 1 & eğer\ x_k = y_k \\ \frac{1}{1 + \log \frac{n}{f_k(x_k)}\, x\ \log \frac{n}{f_k(y_k)}} & diğer \end{cases}$	$\frac{1}{r}$
Lin	$= \begin{cases} 2 \log \acute{p}_k(x_k) & eğer\ x_k = y_k \\ 2 \log(\acute{p}_k(x_k) + \acute{p}_k(y_k)) & diğer \end{cases}$	$\frac{1}{\sum_{i=1}^{r} \log(\acute{p}_i(x_i)) + \log(\acute{p}_i(y_i))}$
Lin1	$= \begin{cases} \sum_{q \in Q} \log \acute{p}_k(q) & eğer\ x_k = y_k \\ 2 \log(\sum_{q \in Q} \acute{p}_k(q)) & diğer \end{cases}$	$\frac{1}{\sum_{i=1}^{r} \sum_{q \in Q} \log \acute{p}_k(q)}$
Goodall 1	$= \begin{cases} 1 - \sum_{q \in Q} p_k^2(q) & eğer\ x_k = y_k \\ 0 & diğer \end{cases}$	$\frac{1}{r}$
Goodall 2	$= \begin{cases} 1 - \sum_{q \in Q} p_k^2(q) & eğer\ x_k = y_k \\ 0 & diğer \end{cases}$	$\frac{1}{r}$
Goodall 3	$= \begin{cases} 1 - p_k^2(x_k) & eğer\ x_k = y_k \\ 0 & diğer \end{cases}$	$\frac{1}{r}$
Goodall 4	$= \begin{cases} p_k^2(x_k) & eğer\ x_k = y_k \\ 0 & diğer \end{cases}$	$\frac{1}{r}$
Smirnov	$= \begin{cases} 2 + \frac{n - f_k(x_k)}{f_k(x_k)} + \sum_{q \in \{A_k \setminus x_k\}} \frac{f_k(q)}{n - f_k(q)} & x_k = y_k \\ \sum_{q \in \{A_k \setminus \{x_k, y_k\}\}} \frac{f_k(q)}{n - f_k(q)} & diğer \end{cases}$	$\frac{1}{\sum_{k=1}^{r} n_k}$
Anderberg $S(x, y) =$	$\frac{\sum_{k \in \{1 < k \leq r : x_k = y_k\}} (\frac{1}{p_k(x_k)})^2 \frac{2}{n_k(n_k + 1)}}{\sum_{k \in \{1 < k \leq r : x_k = y_k\}} (\frac{1}{p_k(x_k)})^2 \frac{2}{n_k(n_k + 1)} + \sum_{k \in \{1 < k \leq r : x_k \neq y_k\}} (\frac{1}{2 p_k(x_k) p_k(y_k)}) \frac{2}{n_k(n_k + 1)}}$	

Yukarıda verilen tabloda $S(x,y) = \sum_{k=1}^{r} w_k S_k(x_k, y_k)$ 'dır.

Lin 1 için $\{Q \subseteq A_k : \forall q \in Q, p_k(x_k) \leq p_k(q) \leq p_k(y_k)\}$, burada $p_k(x_k) \leq p_k(y_k)$ olduğu kabul edilir.

Goodall 1 için $\{Q \subseteq A_k : \forall q \in Q, p_k(q) \leq p_k(x_k)\}$'dür.

Goodal 2 için $\{Q \subseteq A_k : \forall q \in Q, p_k(q) \leq p_k(x_k)\}$'dür.

Bu kategorik veri ölçülerinin özelliklerini maddeler halinde inceleyelim.

- *Overlap:* iki kategorik verideki eşleşen özelliklerin sayısıdır.
- *Goodall:* bu ölçü iki özellik arasındaki benzerliği, özelliklerin veride rastlanma olasılığı ile normalleştirir. Bu ölçü daha seyek frekanstaki kategori değerlerine daha yüksek benzerlik değeri atar. Verideki özellikler arası bağımlılıkları da hesaplamaya dâhil eder. Hesaplama karmaşıklığının yüksek olması nedeniyle daha çok *Goodall1* ölçüsü kullanılır.
- *Goodall 1:* Goodall ölçüsünün aksine özellikler arası bağımlılık yerine özelliklerin benzerlik ortalamasını alır. Eşleşmeleri durumunda $S_k(x_k, y_k)$ nın alabileceği değerlerin aralığı $[0,1 - \frac{2}{n(n-1)}]$'dır. Eğer A_k özelliğinde tek bir değer varsa minimum değerini, iki değer varsa maksimum değerini alır.
- *Goodall 3:* Goodall ölçüsünün bir başka çeşididir. Diğer özellik değerlerinin frekansına bakmaksızın eşleşen değerlerin frekansı seyrek ise yüksek değer atar. Goodall1'de olduğu gibi, maksimum değerini A_k'da iki değer varsa, minimum değerini de tek değer varsa alır ve alabileceği değerlerin aralığı $[0,1 - \frac{2}{n(n-1)}]$'dır.
- *Goodall 4:* Bu ölçü eşleşen kayıtlara $1 - Goodall3$ değerini atar. $S_k(x_k, y_k)$'nın alabileceği değerlerin aralığı $[\frac{2}{n(n-1)}, 1]$ 'dır. Minimum değerini x_k, A_k'da bir kez geçiyorsa, maksimum değerini ise x_k, A_k'da tek değer ise alır.
- *Gambaryan:* Gambaryan verinin hemen hemen yarısında eşleşen kayıtlara daha fazla ağırlık veren bir ölçü tanıtmıştır. $S_k(x_k, y_k)$'nın alabileceği değerlerin aralığı $[0,1]$'dir. Eğer x_k, A_k'daki tek değerse minimum değerini, $\frac{n}{2}$ frekansına sahipse maksimum değerini alır.

Daha detaylı bir inceleme için Kumar vd.'nin [31] çalışmasına bakabilirsiniz.

4.1.3 Kümeler Arası Uzaklık Ölçüleri

Kümeler arası uzaklık hesaplamasında ilk olarak kümelerin ortalama, varyans ve koveryansların hesaplanması gerekir. Bunlar bilindikten sonra iki küme arasındaki mesafeyi hesaplamak için çeşitli ölçüler mevcuttur. Bunlardan en bilinenleri Mahalanobis uzaklık ölçüsü, Hotelling T^2 uzaklık ölçüsü ve Penrose uzaklık ölçüsüdür.

- **Mahalanobis Uzaklık Ölçüsü**

  Mahalanobis uzaklığı çok değişkenli X veri matrisindeki herhangi bir gözlemin verinin merkezine olan uzaklığını verir. Mahalanobis uzaklığı değişkenler arasındaki ilişkileri (varyans/kovaryans) dikkate alır. Bu yönüyle, değişkenlerin ölçüm birimlerinin farklılığından kaynaklı hataları minimuma indirger.

  $$D_{ij}^2 = \left(\mu_i - \mu_J\right)' S^{-1}(\mu_i - \mu_j) \tag{4.28}$$

  Burada;

  μ_i: i. kümenin ortalama vektörü

  S^{-1}: Ortak varyans-kovaryans matrisinin tersidir.

- **Penrose Uzaklık Ölçüsü**

  Penrose, Mahalanobis gibi değişkenler arası ilişkiyi dikkate almaz. Bu yüzden Mahalanobis daha fazla önemsenmektedir. Fakat değişken sayısının az (bir fikir olarak 100'den az örnekli kümelerde) olduğu örneklem uzayında varyans-kovaryans değerlerinin doğruluğu konusunda şüpheler olacağından Penrose tercih edilebilir.

  $$P_{ij} = \sum_{k=1}^{p} \frac{(\mu_{ik} - \mu_{jk})}{pS_k^2} \tag{4.29}$$

- **Hotelling T^2 Uzaklık Ölçüsü**

  İki kümenin ortalama vektörlerinin karşılaştırılmasında kullanılan Hotelling T^2 ölçüsü şu şekildedir;

  $$T^2 = \left(\frac{n_1 n_2}{n_1 + n_2}\right) \sum_{k=1}^{p} \left(\mu_{ik} - \mu_{jk}\right)' S^{-1}(\mu_{ik} - \mu_{jk}) \tag{4.30}$$

Burada;

μ_{ik}: i. kümenin k. Değişkeninin ortalaması

S^{-1}: toplanmış varyans-koveryans matrisinin tersini gösterir.

Görüldüğü gibi bu uzaklık, Mahalanobis uzaklık ölçüsünün $(\frac{n_1 n_2}{n_1+n_2})$ ile düzeltilmiş halidir.

4.1.4 Kullanım Alanlarına Göre Uzaklık ve Benzerlik Ölçüleri

Bir ölçünün kullanımına karar vermemizde en büyük rolü veri tipi oynar. Veri tipinin sürekli-kesikli sayısal veriler, nitelik veriler, sıklık sayıları ve iki kategorili veriler olup olmamasına göre birçok ölçü geliştirilmiştir. Bu uzaklık ve benzerlik ölçülerinin veri tipine göre bir sınıflaması aşağıda verilmiştir. Bunun dışında kümeler arası uzaklığı hesaplamak için Mahalanobis uzaklık ölçüsü, Penrose uzaklık ölçüsü, Hotelling T^2 uzaklık ölçüsü gibi ölçüler tercih edilir.

Çizelge 4.4 Veri Tipine Göre Uzaklık ve Benzerlik Ölçüleri

Uzaklık Ölçüleri	**Benzerlik Ölçüleri**
Sayısal Veriler	
Öklit	Pearson İlişki Katsayısı
Kare Öklit	Kosinüs Benzerlik Ölçüsü
Chebychev (Çebişev)	
Manhattan City Block	
Minkowski	
Karl Pearson/Standartlaştırılmış Öklit Uzayı Ölçü	
Korelasyon Uzaklığı Ölçüleri	
Sıklık Sayıları	

Çizelge 4.4 Veri Tipine Göre Uzaklık ve Benzerlik Ölçüleri (Devamı)

Ki-Kare Uzaklık Ölçüsü	
Phi-Kare Uzaklık Ölçüsü	
İkili Veriler	
Kare Öklit	Russell ve Rao Benzerlik Ölçüsü
Öklit	Basit Benzerlik Ölçüsü
Büyüklük Farkları	Jaccard
Biçim Farkları	Parçalı
Değişim Uzaklık Ölçüsü	Rogers ve Tanimoto
Durum Uzaklık Ölçüsü	Sokal ve Sneath Benzerlik Ölçüsü 1
Lance ve Williams Uzaklık Ölçüsü	Sokal ve Sneath Benzerlik Ölçüsü 2
	Sokal ve Sneath Benzerlik Ölçüsü 3
	Sokal ve Sneath Benzerlik Ölçüsü 4
	Sokal ve Sneath Benzerlik Ölçüsü 5
	Kulczynski Benzerlik Ölçüsü 1
	Kulczynski Benzerlik Ölçüsü 2
	Hamann Benzerlik Ölçüsü
	Goodman ve Lamda Benzerlik Ölçüsü
	Anderberg D Benzerlik Ölçüsü
	Yule Q

Çizelge 4.4 Veri Tipine Göre Uzaklık ve Benzerlik Ölçüleri (Devamı)

	Yule Y
	Ochiai
	Fi 4 Nokta
	Yayılım

4.2 Kümeleme Teknikleri

Çoğu kümeleme algoritması iki popüler yaklaşım üzerine bina edilmiştir [9]. Bunlar hiyerarşik kümeleme algoritmaları ve bölücü kümeleme algoritmalarıdır. Hiyerarşik kümeleme algoritmaları da uygulanma stratejisine göre, yukarıdan aşağı ve aşağıdan yukarı olmak üzere ikiye ayrılır. Başlangıçta veri tek bir küme iken son adımda bütün elemanlar ayrı birer küme halini alır. Bu sonuç bir dendogram veya buz saçağı formatında gösterilebilir. Bölücü kümeleme algoritmalarından en bilineni ise K-Means kümeleme algoritmasıdır ve biz bu çalışma da bu algoritmanın bir genişletilmesini de sunacağız. K-Means algoritması, hiyerarşik kümeleme algoritmalarından farklı olarak, sonuç küme sayısını da girdi olarak alır. K-Means algoritması küme içi dağılımı minimize, kümeler arası dağılımı maksimize etmeyi amaçlar. Aşağıda kümeleme algoritmalarını tanıtıp, başlıca özelliklerine değineceğiz.

4.2.1 Hiyerarşik Kümeleme Yöntemleri

Hiyerarşik kümeleme yöntemleri girdi olarak X veri kümesini ve d uzaklık veya s benzerlik ölçüsünü alırlar. Çıktı olarak dendogram ve ya buz saçağı diyagramları vasıtasıyla gösterilebilecek, her bir kümeleme adımındaki kümeleri dönerler. Eğer yukarıdan aşağı bir strateji kullanılmışsa son adımda her bir nesne bir küme olacak şekilde sonuç üretilir. Kullanılan kümeler arası uzaklık ölçme stratejisine göre aşağıdaki şekilde sınıflandırılabilirler.

4.2.1.1 Tek Bağlantı Yöntemi

Bu yöntemde herhangi iki küme arasındaki uzaklığı ölçmek için, kümelerden seçilen temsilci nesneler birbirine en yakın iki nesnedir.

X_1, X_2 iki veri kümesi olmak üzere $D(X_1, X_2) = SLINK(X_1, X_2) = \min_{x_1 \in X_1, x_2 \in X_2} d(x_1, x_2)$ olarak tanımlanır.

Bu yöntem aşağıdan yukarı strateji için uygundur. Bu yöntemin en büyük dezavantajı ise ince uzun kümeler üretmesidir. Yani aynı kümedeki yakın elemanlar arası uzaklık oldukça düşük, bir kümenin zıt iki ucundaki nesneler arası uzaklık diğer kümenin nesnelerine olan uzaklıktan daha fazladır.

4.2.1.2 Tam Bağlantı Yöntemi

Bu yöntemde herhangi iki küme arası uzaklığı ölçmek için, kümelerden seçilen temsilci nesneler birbirine en uzak iki nesnedir.

X_1, X_2 iki veri kümesi olmak üzere $D(X_1, X_2) = CLINK(X_1, X_2) = \max_{x_1 \in X_1, x_2 \in X_2} d(x_1, x_2)$ olarak tanımlanır.

Bu yöntem yukarıdan aşağı strateji için uygundur. İlk adımda her nesne kendi kümesine aitken son adımda bütün nesneleri barındıran tek kümeye ulaşılır. Bu yöntem tek bağlantı yönteminin yukarıda bahsedilen dezavantajından uzaktır ve daha kompakt kümeler üretmeye meyillidir. Yani küme içindeki elemanlar birbirine çok yakın, fakat farklı kümelerdeki elemanlar birbirine çok uzak olacak şekilde sonuç üretir.

4.2.1.3 Ortalama Bağlantı Yöntemi

Bu yöntem iki küme arası uzaklık olarak, tek bağlantı ve tam bağlantı yöntemiyle hesaplanan uzaklıkların ortalamasını alır.

X_1, X_2 iki veri kümesi olmak üzere $D(X_1, X_2) = \frac{SLINK(X_1, X_2) + CLINK(X_1, X_2)}{2}$ olarak tanımlanır.

Bu yöntemin dezavantajı ise her uzaklık ölçümünde hem tek bağlantı hem tam bağlantı yöntemleri ile uzaklık hesaplaması yapmasıdır.

4.2.1.4 Ortalama Vektör Yöntemi

Bu yöntemde iki küme arası uzaklığı ölçmek için, kümelerden seçilen temsilci nesneler, kümelerin merkezleridir. Bir kümenin merkezi (*centroid*) şu şekilde tanımlanır;

X nümerik veri kümesinin merkezi c olsun. $c = [c_1, c_2, .., c_m]$ ile gösterelim. Her $0 < i \leq m$ için, $c_i = \frac{1}{n}\sum_{k=1}^{n} x_{ik}$ 'dir.

X_1, X_2 iki veri kümesi ve c_1, c_2 sırasıyla bu iki kümenin merkezi olmak üzere $D(X_1, X_2) = d(c_1, c_2)$' dir. Bu yöntemde küme merkezleri kümenin birer elemanı olmayabilir. Bu bazı uygulamalarda dezavantajdır.

4.2.1.5 Medoid Yöntemi

Bu yöntemde iki küme arası uzaklığı ölçmek için, kümelerden seçilen temsilci nesneler, yukarıda tanıtılan küme merkezine en yakın küme içindeki nesnelerdir. Bir kümenin merkezine en yakın elemanına o kümenin *medoid*'i denir. Matematiksel olarak ifade etmek gerekirse;

$med \in X$ 'e X kümesinin medoid'i denir. Öyle ki her $x \in X$ için $c \in X$ küme merkezi olmak üzere $\min(d(x, c))$ değerine sahip x elemanı için $med = x$'dir.

4.2.1.6 Ward Yöntemi

Ward Bağlantı kümeleme yöntemi, iki küme arasındaki uzaklığın hesaplanmasında merkezden sapmaları yani varyansı esas alır. 1963 yılında Joe Henry Ward tarafından önerilmiştir. Bu yönteme en küçük varyans yöntemi de denilmektedir. Ward bağlantı kümeleme yönteminde amaç kümeler içi kareler toplamını minimize etmektir. Benzer eleman sayısına sahip küme elde etme eğilimindedir, uç değerlere karşı duyarlıdır. Bu yöntemde j'nin k ve l kümeleri arasındaki uzaklığı aşağıda verilen eşitlik yardımı ile bulunabilir [28].

$$d_{(k,l),j} = \frac{(n_j+n_k)d_{(k,j)}+(n_j+n_l)d_{(l,j)}+n_j d_{(k,l)}}{n_j+n_k+n_l} \quad (4.31)$$

Burada;

$d_{(k,l),j}$, k ve l'ninci kümelerin birleşmesinin j'ninci kümeye olan uzaklığı,

$d_{(k,j)}$, k'nıncı kümenin j'ninci kümeye olan uzaklığı,

$d_{(l,j)}$, l'ninci kümenin j'ninci kümeye olan uzaklığı,

$d_{(k,l)}$, k'nıncı kümenin l'ninci kümeye olan uzaklığı,

n_j , j'ninci kümenin eleman sayısı,

n_k , k'nıncı kümenin eleman sayısı,

n_l , l'ninci kümenin eleman sayısıdır.

Aşağıda kümeleme algoritmalarını tanıtıp, başlıca özelliklerine değineceğiz.

4.2.2 Hiyerarşik Olmayan (Bölücü) Kümeleme Yöntemi

Hiyerarşik kümeleme yöntemlerinde verideki tüm elemanların birbirine olan mesafeleri veya benzerlikleri hesaplanmak yani mesafe matrisi oluşturmak gereklidir. Bu matris 100 kayıtlı bir veri için 100x100 boyutlarındadır ve 10.000 mesafe ölçülmesi demektir. Kayıt sayısı 10.000 olduğunda ise mesafe matrisinin hesaplanması için 100.000.000 mesafe ölçülmesi gerektiği anlamına gelir. Bu yüzden büyük veri kümeleri için mesafe matrisi oluşturmadan kümeleme işleminin yapılmasının yolları araştırılmıştır.

Hiyerarşik olamayan kümeleme yöntemleri büyük mesafe matrislerine ihtiyaç duymaz. Zaman ve bellek kullanımı açısından hiyerarşik kümeleme yöntemlerine göre çok daha elverişlidir. En bilinen bölücü kümeleme algoritması K-Means algoritmasıdır.

4.2.2.1 K-Means Kümeleme Yöntemi

K-Means algoritması [7] bölücü kümeleme algoritmalarının en çok kullanılanlarından biridir [24]. Mac Queen tarafından 1967'de geliştirilmiştir [33]. Verilen bir X veri kümesi ve istenen k sonuç küme sayısı ile başlar ve X kümesini küme içi kare hatalar toplamı minimum olacak şekilde k adet kümeye böler [34]. Küme benzerliği, kümenin ağırlık merkezi kabul edilen bir birim ile kümedeki diğer birimler arasındaki uzaklıkların ortalama değeri ile ölçülmektedir [21]. K-Means kümeleme yöntemi bir sonraki bölümde detaylı olarak ele alınacaktır.

BÖLÜM 5

K-MEANS KÜMELEME YÖNTEMİ

K-Means algoritması dört temel işlem üzerine bina edilmiştir [35];

- k adet başlangıç küme merkezinin seçimi,
- nesneler arası uzaklıkların ve küme merkezlerinin hesaplanması,
- nesnelerin, merkezine en yakın oldukları kümeye atanması,
- küme içi benzeşmezlik minimum indirgenene kadar hesaplama ve atamaların devam etmesi.

İlk adım hariç diğer üç adım, küme içi kareler toplamı minimuma indirgenene kadar tekrarlanır. Minimize edilecek maliyet fonksiyonu;

$$E = \sum_{l=1}^{k} \sum_{i=1}^{n} y_{i,l} d(X_i, Q_l) \quad (5.1)$$

dur. Burada n, X kümesinin eleman sayısı, $X_i \in X$ ve Q_l , l kümesinin merkezidir. $y_{i,l}$ ise bölüntüleme matrisinin elemanıdır. Y_{nxl} bölüntüleme matrisinin elemanları l'ninci elemanın n'inci kümeye ait olup olmaması durumuna göre 1 veya 0'dır. d uzaklık ölçüsü ise aksi belirtilmedikçe Öklid uzaklık ölçüsüdür. Buradaki tanımlardan anlaşılacağı üzere K-Means algoritması, seçilen ölçümler nedeniyle sadece nümerik veriler ile ilgilenir.

K-Means kümeleme algoritmasının, merkezlerin seçimi, uzaklıkların ölçülmesi ve başlangıç küme merkezlerinin seçiminin farklılaşması ile ortaya çıkan farklı versiyonları vardır [7]. En gelişmiş bir K-Means çeşidi ISODATA algoritmasını içeren versiyonu [36] ve Fuzzy K-Means algoritmasıdır [37],[38].

K-Means algoritması şu temel özelliklere sahiptir [35];

1- Büyük veri kümeleri için etkilidir. Algoritmanın hesaplama karmaşıklığı $O(tkmn)$'dir. Burada; n nesne sayısı, m özellik sayısı, t yineleme sayısı ve k küme sayısıdır. Genel olarak $m, t, k \ll n$ 'dir. Büyük veri kümelerini K-Means ile kümelemek, hiyerarşik kümeleme tekniklerine göre çok daha hızlıdır. Çünkü hiyerarşik kümeleme tekniklerinin hesaplama karmaşıklığı $O(n^2)$'dir.
2- K-Means algoritması yerel eniyileme ile sonlanır [39]. Global eniyilemeyi de bulmak için genetik algoritmalarla birleştirilebilir.
3- K-Means sadece nümerik değerler ile çalışır. Çünkü maliyet fonksiyonun da kümenin ortalaması hesaplanır.
4- Kümeler dışbükey bir şekle sahiptir [7]. Bu yüzden içbükey şekiller elde etmek zordur.

K-Means algoritmasının bir diğer zorluğu da k küme sayısının belirlenmesidir. ISODATA gibi bazı K-Means türevleri ise bu k değerini de kendi içinde hesaplamaya uygundur [35].

K-Means algoritması büyük verileri işlemedeki başarısı ile veri madenciliğine en uygun tekniktir. Buna rağmen sadece nümerik veriler ile çalışması onun kullanımını kısıtlamaktadır. Çünkü gerçek veriler çoğu zaman kategorik veriler de içerir. Bu kısıtı ortadan kaldırmak için K-Modes gibi bazı genişletilmiş K-Means algoritmaları da geliştirilmiştir [35]. Zhexue Huang'ın ortaya koyduğu bu algoritmayı ilerleyen bölümlerde detaylı olarak ele alacağız. Bizim de bu çalışmadaki esas amacımız hibrit veriler ile çalışabilen bir K-Means genişletmesi sunmaktır. Bu yüzden Zhexue Huang'ın bu çalışması bize yol gösterici olacaktır.

5.1 Veri Türleri

Yukarıda da bahsedildiği üzere K-Means kümeleme algoritması maliyet fonksiyonundaki küme ortalaması hesabından dolayı sadece nümerik veriler ile çalışabilmektedir. Her $x \in X$ için $x = [x_1, x_2, .., x_m]$ olmak üzere, eğer her $0 < i \leq m$ için x_i değeri nümerik ise X veri kümesine nümeriktir denir. Aynı şekilde eğer tüm $x_i$değerleri kategorik ise bu durumda X veri kümesine kategoriktir denir. Eğer bazı x_i değerleri kategorik bazı x_i değerleri nümerik ise böyle verilere de hibrit veriler denir ve özel bir notasyon olmak üzere $x = [x_1^r, x_2^r, x_3^c, x_4^c .., x_m^c]$ ile gösterilir. Burada x_k^c kategorik, x_k^r değerleri ise nümeriktir [40].

K-Means kümelemesi başlığı altında veri türlerini yeniden ele almamızın sebebi, K-Means algoritmasının veri türlerine göre genişletilmiş algoritmalarını sunarken kullanacağımız notasyonu tanıtmaktı. Verinin tek tip olması durumunda (nümerik ya da kategorik) karmaşıklığa yol açmayacağından hibrit veri gösterimindeki üstel simgeyi kullanmayacağız.

5.2 Uzaklık Ölçüsü Seçimi

Verinin nümerik olması durumunda tercih edilebilecek birçok uzaklık ölçüsü vardır. Bunların bir kısmını yukarıda tanıtmıştık. Ancak unutulmamalıdır ki bu ölçünün seçimi algoritmanın sonucuna direkt etki edecektir.

K-Means kümeleme algoritmasında aksi belirtilmedikçe Öklit uzaklık ölçüsü kullanılır. Bu ölçü bir küme merkezi ile bir nesne arasındaki uzaklığı ölçmek için kullanılır. Ancak uzaklık ölçülerinin her biri bazı veri tiplerine göre özellikle iyi sonuçlar verebilir. Örneğin veri iki değerli özelliklerden oluşuyorsa seçilecek uzaklık ölçüsü veriyi kümeleme stratejimize uygun olarak yukarıda tanıtılan ikili veriler için uzaklık ölçülerinden biri seçilebilir. Veya kesikli sayısal veriler için Manhattan ölçüsü daha başarılı sonuçlar verebilir. Verinin standart hale getirilmediği durumlarda Öklit uzaklık ölçüsü büyük farkların olduğu özelliğin benzerliğe olumsuz etkide bulunmasına neden olabilir.

Kategorik veriler için seçilecek uzaklık ölçüsünde ilk akla gelen, aynı olan değerler için uzaklık değeri 0 farklı olan değerler için uzaklık değeri 1 olacak şekilde çalışan çakışma fonksiyonudur. Fakat kategorik veriler her zaman düzenli dağılım göstermezler. Bu yüzden çakışma fonksiyonu çoğu zaman istenen sonuçları elde etmemize yardımcı olmaz. Kategorik veriler için seçilecek uzaklık ölçüsünün kategorik değerlerin frekanslarını da dikkate alması beklenir. Aynı şekilde bir özelliğin alabileceği değerler kümesinin eleman sayısı da dikkate alınabilir. Örneğin, cinsiyet özelliği ile göz rengi özelliklerinin alabileceği değerler farklı sayılardadır. Bir başka yöntem olarak cinsiyet gibi iki değerli kategorik özellikler ikili sınıf verisine dönüştürülerek uygun ikili sınıf uzaklık ölçüleri de kullanılabilir.

5.3 Küme Sayısının Belirlenmesi

K-Means kümeleme algoritmasının girdilerinden biri olan k küme sayısının ne olması gerektiği, nasıl belirleneceği ve k'yı ve k adet başlangıç merkezini otomatik belirleme algoritmaları üzerine birçok çalışma yapılmıştır [25], [33], [41], [42], [43]. Başlangıç merkezlerinin seçimi konusu bir sonraki başlıkta detaylı incelenecektir. k'yı belirlemek için basit ve yaygın kullanılan bir formül şöyle verilebilir;

$$k = \sqrt{\frac{n}{2}} \tag{5.2}$$

Burada n veri kümesinin eleman sayısıdır. Bu seçimin küçük örneklem kümeleri için kullanılması tavsiye edilir. Büyük örneklemler için sağlıklı sonuçlar vermeyebilir [12].

Bir diğer yöntem olarak Mariott tarafından önerilen formül kullanılabilir [44];

$$M = k^2|W| \tag{5.3}$$

Burada W küme içi kare hataları toplamı matrisidir. M değerini minimum yapan k değeri uygun küme sayısı olarak seçilmektedir. Everitt (1974) farklı ölçümlerle ilgili yaptığı çalışmada, Mariott yönteminin diğer yöntemlere göre daha iyi sonuç verdiğini ortaya koymuştur [12].

Küme sayısının belirlenmesi adına bir diğer yöntem ise Lewis ve Thomas tarafından geliştirilmiştir. Lewis ve Thomas' a göre küme sayısına karar vermek için iki kriter göz önüne alınır. Kümeler, toplam varyansın %80'ini açıklamalı ve varyansta %5'e kadarlık bir artış durumunda yeni bir küme ilave edilebilmelidir [45].

Benzer biçimde Calinski ve Harabasz (1974) da bir yöntem ortaya koymuşlardır [25].

Everitt (1979) ve Anderberg (1973) kümeler arası varyansın küme içine göre maksimum olduğu durumları belirleme açısından Wilk's Lamda ölçütünün çok duyarlı olduğunu belirtmişlerdir [25].

Her ne kadar k küme sayısını belirleme üzerine formüller verilse de, eğer belli bir küme sayısına ulaşmak için kümeleme işlemi yapılıyorsa, en uygun k kullanıcının bu ön bilgi ile gireceği k değeridir. Kullanıcın bu kararında ulaşmak istediği detay seviyesi, global dağılımın mı yerel dağılımın mı önemsendiği gibi ölçütler göz önünde bulundurulur.

D.T. Pham (2004)'ın yaptığı çalışmada, k'yı belirlemenin bazı yöntemleri verilmiştir. [46];

- K-Means performansı k'nın seçiminden direkt etkilenebilir. Bu yüzden tek bir k değeri yerine bir grup k değeri ile çalışılabilir. Bu k sayıları verinin karakteristiğine uygun olmalıdır. Her biri için çalıştırılan algoritmanın sonuçlarından optimumu dikkate alınır.
- k kullanıcı tarafından belirlenir. Bu yaklaşımda, çok boyutlu veri kümelerinde kümeleme sonuçlarının değerlendirilmesi kullanıcı için zordur.
- Bazı ön kümeleme araçları tarafından belirlenebilir.
- Yapay veri kümeleri için k değeri, yapay veri üreteci sayısına eşitlenebilir.
- k'nın seçimi için bazı istatistiksel ölçümler kullanılabilir. Bu ölçüler verinin dağılımını göz önünde bulundurarak çalışırlar. Bayes kriteri veya Akeike kriteri, Gauss dağılımına sahip veriler üzerinde uygulanabilir. Verinin Possion dağılımı göstermesi durumunda Hardy ölçümü kullanılabilir. Monte Carlo tekniği kümeleme sonuçlarını değerlendirmek ve k'yı belirlemek için kullanılabilir.
- Her ne kadar K-Means denetimsiz yani ön bilgisiz bir kümeleme tekniği olsa da verideki bir kategorinin sınıf sayısı olarak seçilebilir.
- Veriyi görselleştirerek görsel üzerinden küme sayısı öngörümü yapmak da basitliği nedeniyle sıkça kullanılan bir yöntemdir. Bununla beraber bu yöntem kümeleme sonuçlarını değerlendirmek için de kullanılır.

Bir komşuluk ölçüsü K-Means algoritmasının maliyet fonksiyonuna eklenebilir. Bu yöntem birkaç veri seti için umut verici şekilde güzel sonuçlar vermiştir. Bu yöntemin pratik kullanım örnekleri ile yeteneğinin ispat edilmesine ihtiyacı vardır. Maliyet fonksiyonunun değişmesi nedeniyle bu çözümün uygulandığı algoritmalar K-Means olarak değerlendirilmez [46].

5.4 Başlangıç Merkezlerinin Seçimi

K-Means kümeleme yöntemi ve K-Means'den türetilen hiyerarşik olmayan diğer kümeleme yöntemlerinde k adet başlangıç küme merkezi ile kümeleme işlemine başlanır. K-Means kümeleme algoritması başlangıç merkezlerini rassal olarak seçer ve sonuç kümeleri doğrudan bu seçimden etkilenir. Dolayısıyla bu seçim hesaplama zamanını ve sonuç kümelerinin doğruluğunu doğrudan etkiler [43]. K-Means kümeleme

algoritması en iyi sonucu sadece bu başlangıç merkezleri sonuç başlangıç merkezlerine en yakın seçildiklerinde verir [24],[41].

Bu yüzden bu başlangıç merkezlerinin nasıl seçileceği üzerine de önemli çalışmalar yapılmıştır. Hesaplama zamanını iyileştirmek için Fahim vd. [47] bir yöntem önermişlerdir. Buna göre bir nesnenin bulunduğu kümenin merkezine olan uzaklığı ölçülür ve bir sonraki adımda eğer bu uzaklık önceki ölçüden küçük veya eşitse yine bu kümeye aittir denir ve diğer küme merkezleri ile uzaklık ölçüsü hesaplanmaz. Bu yöntemin hesaplama zamanını olumlu etkilediğini yaptıkları testler ile göstermişlerdir. Fakat hala başlangıç merkezi seçimi rassaldır.

Mahmud vd. [48] ise başlangıç merkezlerinin seçimi için tanıttıkları bir algoritmada, önce verinin tümünü ağırlıklı skorlarına göre sıralar ve k adet alt kümeye ayırır. Bu alt kümelerin her birinin merkezi alınarak k adet başlangıç küme merkezi elde edilir. Bu yönteminde daha iyi sonuç kümeleri ürettiği ve hesaplama süresini de iyileştirdiği gözlenmiştir.

Kumar ve Gupta [43] benzeşmezlik ağacını kullanarak başlangıç merkezi seçmenin bir yolunu önermişler ve yaptıkları testler ile bu seçimlerin sonuç küme merkezlerine yakın olduklarını gözlemlemişlerdir.

Khan ve Ahmad [41] tarafından önerilen bir başka yöntem ise her bir özelliğin kendi içinde merkezlerinin bulunması ve daha sonra bunların birleştirilmesidir. Cluster Center Initialization Algorithm (CCIA) olarak bilinen bu yöntem küme merkezi belirlemek için etkili bir yöntemdir ve bu algoritma [41]'dan incelenebilir. Algoritma birbirine benzeyen iki temel gözleme dayanır. Her bir özellik için standart sapma ve ortalama değerleri hesaplanır ve belli bir bölüntüleme ile normal dağılım eğrisinde işaretlenir. İşlem bu esas üzerinden sürdürülür.

Arai ve Barakbah [33] tarafından Hiyerarşik K-Means olarak adlandırılan bir algoritmada ise girilen bir p değeri kadar K-Means uygulanır ve sonuç küme merkezleri bir havuzda toplanır. Bu havuz hiyerarşik kümelemeye sokularak k adet kümeye sahip adımın küme merkezleri başlangıç merkezleri olarak alınır ve esas veri bu merkezler ile K-Means kümeleme işlemine sokulur. Bu algoritmada p kez K-Means ve bir kez de hiyerarşik kümeleme yapılması hesaplama zamanı açısından bir kayıptır.

K-Modes kümeleme algoritmasının başlangıç merkezlerinin seçimi için Khan ve Kant [49] tarafından bir algoritma önerilmiştir. Buna göre en iyi başlangıç küme merkezlerini seçmek için önce birkaç kez rastgele merkezler seçilerek kümeleme işlemi yapılır ve bu kümeleme işlemlerinin her adımındaki mod değerleri bir havuzda toplanır. Bu havuzda en sık geçen modlar başlangıç merkezleri olarak seçilirler. Bu algoritmada hesaplama zamanından kayıp olsa da sonuç kümelerinin doğruluğu açısından bir kazanç elde edilir.

Bunun dışında çalışmamızın esas konusu olan kategorik ve hibrit verilerin kümelenmesi işlemi için alternatif bir başlangıç merkezi seçme yöntemi Huang tarafından K-Modes algoritmasının önerildiği çalışmada sunulmuştur [35]. Biz de testlerimizde bu yöntemi kullanarak başlangıç merkezlerimizi belirleyeceğiz.

Görüldüğü gibi başlangıç merkezlerinin seçimi aşaması kümeleme işleminin kritik adımlarından biridir. Herkes tarafından kabul edilen ortak bir seçim yöntemi olmamakla birlikte standart K-Means kümeleme algoritmasında bu seçim rassal olarak yapılır. Ancak bu sağlıklı bir yöntem olmadığından, verinin boyutu, türü ve özellikleri açısından uygun seçim yöntemi tercih edilmelidir.

5.5 Ağırlık Kavramı

Bazen çalışılacak verinin özellikleri eşit önem derecesine sahip olmayabilir. Bu önem derecesi veriden çıkarılmak istenen bilginin kapsamına göre de değişiklik gösterebilir. Örneğin öğrenci bilgilerinin tutulduğu bir veride yapılacak kümeleme işlemini düşünelim. Eğer öğrencilerin başarı durumlarına göre bir gruplama yapılmak isteniyorsa, kilo, boy, okul numarası gibi özelliklerin kümeleme işlemine olan etkisinin daha az olması gerekir. Hatta veriden çıkarılması (0 ağırlık) bile istenebilir. Öğrencilerin fiziksel olarak benzerliklerine göre yapılacak bir gruplama işleminde ise ders notlarının kümeleme işlemine etkisinin daha az olması gerekir. Bunun yanında bazı verilerin doğasında, kümeleme işleminin kapsamı dikkate alınmaksızın, bir ağırlık dağılımı olabilir. Böyle durumların da kümeleme işleminde göz önüne alınması gerekir.

K-Means ve diğer hiyerarşik olmayan kümeleme algoritmalarının dezavantajlarından biri de bütün özelliklere eşit önem derecesindeymiş gibi davranarak kümeleme işlemini yapmalarıdır. K-Means algoritmasının, özellikleri önemlerine göre ağırlıklandırarak kümeleme işlemi yapması için yapılan çalışmalar bundan 40 yıl öncelerine dayanır.

Renato Cordeiro de Amorim tarafından yakın zamanlarda yapılan "*A survey on feature weighting based K-Means algorithms*" adlı çalışmada bu zamana kadar yapılan bu çalışmalardan bazıları incelenmiştir [50]. K-Means için ağırlıklandırma çalışmaları yapıldığı gibi K-Modes ve K-Prototypes için de benzeri çalışmalar yapılmıştır.

Bilinen ilk ağırlıklı K-Means türevi çalışma De Sarbo ve arkadaşları tarafından 1984'de yapılmıştır. Bu çalışmada özellikleri ilişkilerine göre gruplamak ve önem derecelerine göre gruplara ağırlıklar atamak önerilmiştir.

Chan ve arkadaşları tarafından 2004 yılında önerilen bir başka yöntem ise kümelerin küme içi durumlarını göz önüne alarak her adımda özelliklerin ağırlıklarını değiştirmektir. Bu yöntem hiç şüphesiz hesaplama karmaşıklığına yol açacak ve işlem süresini uzatacaktır.

İlk akla gelen ve en basit yöntem ise her bir özellik için başta bir ağırlık belirlemek ve işlem süresince bu ağırlıkları kullanmaktır. Bu yöntem Weighted-K-Means olarak adlandırılır.

Kategorik veriler ile çalışıldığında, kategorik özelliğin değerlerinin frekansları dikkate alınarak da ağırlık belirlemek bir yöntemdir ve Ji ve arkadaşları tarafından 2013 yılında Improved K-Prototypes adıyla önerilmiştir.

Ağırlık belirlemenin etkili bir yöntemi de veri hakkında bilgi sahibi biri tarafından ağırlıkların belirlenmesidir. Aslında kümeleme işleminde hangi özelliklerin daha yüksek öneme sahip olduğu ve sonuç olarak aslında nasıl bir yapı beklendiğini en iyi bu kişiler bilir. Bu yöntemin dezavantajı ise verinin yüksek boyutlu olması durumlarında zaman kısıtı ve hata ihtimalinin yüksek olmasıdır. Bizim bu çalışmada önereceğimiz algoritma ağırlıklandırılabilen bir algoritmadır ve ağırlık değerlerini bir girdi olarak kullanıcıdan alır.

5.6 Genişletilmiş K-Means Algoritmaları

K-Means algoritması, amaç fonksiyonunun sadece nümerik veriler ile çalışabilmesi nedeniyle kategorik ve hibrit verileri kümeleme noktasında yetersiz kalmaktadır. Hâlbuki karşılaşılan verilerin çoğu kategorik veriler barındırır veya tamamen kategorik verilerden oluşur. İşte bu tip verileri hiyerarşik olmayan bir yöntemle kümeleme ihtiyacını gidermek için K-Means algoritmasının sadece kategorik veya hibrit veriler için genişletilmiş bazı

versiyonları vardır. Bunun dışında başlangıç merkezlerinin seçimi, sonuç küme sayısının belirlenmesi gibi aşamalarında özelleştirilmesiyle farklı K-Means genişletmeleri ortaya çıkmıştır. Burada bazı K-Means genişletmelerine göz atacak ve esas çalışmamız olan W-K-Prototypes algoritması için de bir ön hazırlık yapmış olacağız.

5.6.1 K-Modes Algoritması

Zhexue Huang tarafından "*A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining*" çalışmasında önerilen bir algoritma olan K-Modes algoritması, K-Means algoritmasının, farklı uzaklık ölçüsü kullanılarak ve küme merkezi olarak ortalama (mean) yerine mod (mode) değerinin alınmasıyla genişletilmesi ile elde edilmiştir. Bu mod değerinin hesaplanması için ise bilinen istatistiksel mod bulma yöntemi yerine frekans tabanlı bir yöntem kullanılır. K-Modes algoritmasının kategorik veriler için kullandığı uzaklık ölçüsü yukarıda tanıttığımız overlap uzaklık ölçüsüdür.

$$d(x,y) = \sum_{i=1}^{m} \delta(x_i, y_i) \tag{5.4}$$

$$\delta(x_i, y_i) = \begin{cases} 0 & (x_i = y_i) \\ 1 & (x_i \neq y_i) \end{cases} \tag{5.5}$$

Yukarıda ki basit overlap uzaklık ölçüsünde verideki tüm özellikler eşit ağırlığa sahiptir. Aynı şekilde bir özelliğin sahip olabileceği kategorik değerlerin, verideki sıklığı da dikkate alınmamaktadır. Yine Huang tarafından önerilen ve özelliklerin alabileceği değerlerin frekanslarını da uzaklık hesabına dâhil eden, daha etkin bir uzaklık ölçüsü şu şekildedir;

$$d_{\chi^2}(x,y) = \sum_{i=1}^{m} \frac{(n_{x_i} + n_{y_i})}{n_{x_i} n_{y_i}} \delta(x_i, y_i) \tag{5.6}$$

Burada n_{x_i}, i. özellikte, x_i değerine sahip nesne sayısıdır. Bu uzaklık ölçüsünde daha sık geçen değerlere daha az önem verilmektedir ve verinin türüne ve araştırmaya göre bu uzaklık ölçüsünü kullanmak daha iyi bir çözüm olabilir. Örneğin bir sigorta acentesinin verisi üzerinde yapılan dolandırıcılık tespiti işleminde kullanılması daha etkindir.

K-Modes algoritmasında küme merkezi olarak seçilecek mod değeri ise $Q = [q_1, q_1, \dots, q_m]$ şeklinde bir vektördür. Öyle ki aşağıdaki toplamı minimize eder.

$$D(Q,X) = \sum_{i=1}^{n} d(x_i, Q) \tag{5.7}$$

Burada $X = \{x_1, x_2, \dots, x_n\}$ olacak şekilde veri kümesidir. Görüleceği üzere mod değeri Q, X veri kümesinin elemanı olmak zorunda değildir.

Burada verilen uzaklık ölçüsü ve mod hesaplama yöntemleri kullanılmak üzere K-Modes algoritmasının adımları K-Means ile aynıdır. Huang, K-Modes algoritmasını uygularken başlangıç merkezlerinin seçimi için iki farklı yöntem kullanmıştır. İlk yöntemde ilk k adet farklı kayıt küme merkezi olarak seçilirken ikinci yöntem daha etkin bir şekilde seçim yapmak için önerilmiştir. Kategorilerin sıklığına göre sıralanıp, bir yöntemle içlerinden seçim yapılarak mümkün olan en farklı küme merkezlerini seçmek amaçlanmıştır. Bu yöntemin detayı için Huang'ın ilgili çalışmasına [35] bakılabilir.

5.6.2 K-Prototypes Algoritması

Zhexue Huang'ın, *"Clustering Large Data Sets with Mixed Numeric and Categorical Values"* adlı çalışmasında[40] tanıttığı bir algoritma olan K-Prototypes algoritması, K-Means algoritmasının sadece nümerik veriler ile çalışabilme kısıtının kaldırılmış ama etkili çalışma özelliğinin korunduğu bir versiyonudur. Bu algoritmada kullanılan benzerlik ölçüsü nümerik ve kategorik özellikler ile çalışabilmektedir. Sadece nümerik veriler ile çalışılırsa K-Prototypes algoritması tıpkı K-Means gibi davranacaktır. Gerçek veriler ile SUN Sparc10 iş istasyonunda yapılan testlerde etkili ve hızlı sonuçlar alındığı görülmüştür.

K-Prototypes algoritmasında kullanılan uzaklık ölçüsü aşağıdaki gibidir.

$$d(x_i, Q_l) = \sum_{j=1}^{m_r}(x_{ij}^r - q_{lj}^r)^2 + \gamma_l \sum_{j=1}^{m_c} \delta(x_{ij}^c, q_{lj}^c) \quad (5.8)$$

Burada $\delta(x, y)$, x ve y kategorik değişkenleri aynı ise 0, farklı ise 1 değerini alan fonksiyondur. x_{ij}^r ve q_{lj}^r değerleri nümerik x_{ij}^c ve q_{lj}^c değerleri kategoriktir. m_r nümerik özelliklerin sayısı, m_c kategorik özelliklerin sayısıdır. γ_l ise l kümesi için kategorik verilerin ağırlığıdır.

Bu uzaklık ölçüsü kullanıldığında maliyet fonksiyonu şu şekilde yeniden yazılabilir;

$$E_l = \sum_{i=1}^{n} y_{il} \sum_{j=1}^{m_r}(x_{ij}^r - q_{lj}^r)^2 + \gamma_l \sum_{i=1}^{n} y_{il} \sum_{j=1}^{m_c} \delta(x_{ij}^c, q_{lj}^c) = E_l^r + E_l^c \quad (5.9)$$

Burada E_l^r nümerik özelliklerin maliyet fonksiyonu ve E_l^c kategorik özelliklerin maliyet fonksiyonudur. Bu maliyet fonksiyonu göz önüne alındığında E_l^c maliyet fonksiyonunu

minimize etmek için seçilmesi gereken merkezin kategorik özelliklerinin küme içinde en sık rastlanan (çünkü sigma fonksiyonu aynı olması durumunda 0 değerini veriyor ve toplama katkıyı azaltıyor.) değerlerden seçilmesi gerektiği anlaşılıyor. Buna dayanarak Huang'ın merkez seçme önerisi; nümerik bileşenlerin ortalamasının ve kategorik bileşenlerin frekansı en yüksek olanlarının seçilmesi.

Yukarıdaki bilgiler ışığında K-Prototypes algoritması şu şekildedir;

(i) X veri kümesi içinden kümeleri temsil etmek üzere k adet başlangıç merkezi (prototype) seçilir.

(ii) X veri kümesindeki her bir nesne, en yakın oldukları prototype'ın bulunduğu kümeye atanır.

(iii) Bütün nesneler atandıktan sonra kümelerin prototype'ları yeniden hesaplanır. Ve her bir nesne yeniden en yakın oldukları prototype'ın bulunduğu kümeye atanır.

(iv) Hiçbir nesne yer değiştirmeyinceye kadar adım (iii) tekrarlanır.

(3) denklemindeki γ_l katsayısı ise kategorik verilerin uzaklık ölçüsündeki ağırlığını belirlemek amacıyla önerilmiştir. Örneğin nümerik değişkenlerin değer aralıkları çok geniş olduğunda kategorik değişkenlerin farklı olması durumunda uzaklığa ekleyecekleri 1 değeri çok anlamsız ve küçük kalabilir. Tersine nümerik değerlerin [0.5,0.6] gibi küçük bir aralıkta değişmesi durumunda 1 değeri uzaklık belirlemede daha baskın rol oynayacağı için istenmeyen bir durum oluşturur. İşte bu durumun önüne geçmek için Huang'ın önerisi γ_l değerinin belirlenmesinde σ_l standart sapma değerinin (l kümesindeki nümerik özelliklerin standart sapmalarının ortalaması) kullanılmasıdır. Bu sayede kategorik değişkenlerin farklı olması durumunda uzaklık ölçüsüne etkisi hemen hemen nümerik değerlerin farklı olduğu durumdaki kadar olacaktır. Bu önerinin her zaman doğru sonuç vermesi beklenemeyeceği gibi tüm veri dikkate alındığında iyi bir çözüm olduğu açıktır. Huang, çalışmasında γ_l'nin farklı değerleri için bazı sonuçlar paylaşmıştır. γ_l ağırlığının her adımda ve her küme için yeniden hesaplanması bu algoritmanın bir dezavantajıdır.

Aslında algoritmada kullanılan uzaklık ölçüsü iki adet uzaklık ölçüsünün sonuçlarının toplanmasından oluşur. Bu ölçüler nümerik değerler için *kare öklit* uzaklık ölçüsü,

kategorik değerler için *overlap* uzaklık ölçüsüdür. K-Prototypes'da kullanılan uzaklık ölçüsünü daha modüler hale getirmek için, bu ölçülerin yerine alternatif başka ölçüler de düşünülebilir.

5.6.3 K-Harmonik Ortalama Algoritması

K-Harmonik Ortalama Algoritması (KHM Algorithm) basit uygulaması ve az sayıda yineleme adımı sayesinde basit ve hızlı bir algoritmadır. KHM algoritması, her x_i veri noktasının en yakın c_j küme merkezine karşılıklı uzaklıklarının toplamının minimize edileceği şekilde $(c_1, c_2, \dots, c_k)$ küme merkezlerini bulmaya çalışır. KHM algoritmasının en büyük problemi ise başlangıç küme merkezlerinin seçimine olan yüksek bağımlılığıdır. Bunun nedeni, veri noktaları ile en yakın merkez arasında güçlü bir ilişki ortaya çıkaran ve merkezlerin yerel bir veri yoğunluğundan uzak durmasını sağlayan, "kazanan her şeyi alır" (winner-takes-all) bölüntüleme stratejisidir.

KHM algoritması, her veri noktasından tüm merkezlere olan uzaklığın harmonik ortalaması ile KHM'de kullanılan bir veri noktasından merkez noktalara olan minimum mesafeyi yer değiştirerek içsel bir problemi giderir. KHM algoritması her veri noktası herhangi bir merkeze yakın olduğunda en iyi sonucu verir. Bu harmonik ortalamanın bir özelliğidir; KHM'nin kullandığı minimum işleve benzemekle birlikte yumuşak diferensiyellenebilir bir işlevdir. KHM algoritmasını formüle etmek için aşağıdaki gösterimler kullanılır.

$X = \{x_1, x_2, \dots, x_n\}$ kümelenecek veriyi, $C = \{c_1, c_2, \dots, c_k\}$ küme merkezlerini, $m(c_j|x_i)$ ise c_j kümesine ait olan x_i veri noktasının oranıyla tanımlanan üyelik fonksiyonudur. $w(x_i)$ ağırlık fonksiyonu da, bir sonraki yinelemede merkez parametrelerin yeniden hesaplanmasında x_i veri noktasının ne kadar etkili olduğunu tanımlar. Bu tanımlamalar eşliğinde KHM algoritması aşağıdaki adımlardan oluşur.

I. Tahmini merkezlerden oluşan C kümesi oluşturulur. (Rassal seçimler de yapılabilir.)

II. Aşağıdaki formüle uygun olarak amaç fonksiyonu değeri hesaplanır.

$$KHM(X, C) = \sum_{i=1}^{n} \frac{k}{\sum_{j=1}^{k} \frac{1}{\|x_i - c_j\|^p}} \tag{5.10}$$

Buradaki p sayısı algoritma girdisidir ve tipik olarak $p \geq 2$ dir.

III. Her x_i veri noktası için aşağıdaki formüle uygun olarak $m(c_j|x_i)$ üyelik fonksiyonu değeri hesaplanır.

$$m(c_j|x_i) = \frac{\|x_i - c_j\|^{-p-2}}{\sum_{j=1}^{k}\|x_i - c_j\|^{-p-2}} \quad (5.11)$$

IV. Her x_i veri noktası için aşağıdaki formüle uygun olarak $w(x_i)$ değeri hesaplanır.

$$w(x_i) = \frac{\sum_{j=1}^{k}\|x_i - c_j\|^{-p-2}}{\left(\sum_{j=1}^{k}\|x_i - c_j\|^{-p}\right)^2} \quad (5.12)$$

V. Her c_j küme merkezi için, bütün x_i veri noktalarının yeri, ağırlık ve üyelik değerleri dikkate alınarak aşağıdaki formül ile yeniden hesaplanır.

$$c_j = \frac{\sum_{j=1}^{n} m(c_j|x_i) w(x_i) x_i}{\sum_{j=1}^{n} m(c_j|x_i) w(x_i)} \quad (5.13)$$

VI. Adım 2'den Adım 5'e kadar olan adımlar önceden belirlenen bir yineleme sayısı kadar veya $KHM(X, C)$ fonksiyonunun değişim değeri önceden belirlenmiş bir eşik değerin altında kalana kadar tekrar edilir.

VII. $m(c_j|x_i)$ fonksiyonu en büyük değerine sahip olacak şekilde her bir x_i veri noktasını j kümesine atanır.

Burada ele aldığımız K-Means genişletilmiş algoritmalarından da görüleceği gibi, farklı genişletilmiş K-Means algoritmaları elde edebilmek için uzaklık ölçüsü seçimi ve maliyet fonksiyonunun güncellenmesi gibi değişikliklere gidilmiştir. Biz de K-Prototypes algoritmasını aşağıdaki bölümde ele aldığımız W-K-Prototypes algoritmasına benzeri bir yöntemle genişletmiş bulunmaktayız.

BÖLÜM 6

W-K-PROTOTYPES ALGORİTMASI

Buraya kadar olan kısımda gerek hiyerarşik gerekse hiyerarşik olmayan kümeleme yöntemlerinden bahsettik. Hiyerarşik kümeleme yöntemleri etkili sonuçlar verse de hesaplama karmaşıklığı nedeniyle büyük veriler ile çalışma konusunda yetersiz kalmaktadır. Hiyerarşik olmayan kümeleme yöntemleri ise maliyet fonksiyonunun kategorik veriler ile çalışamaması nedeniyle tüm veri tipleri için kullanılamamaktadır. Yalnızca kategorik veriler için ve hibrit veriler için bazı hiyerarşik olmayan kümeleme yöntemleri de son zamanlarda geliştirilmiştir. Bunun yanında verideki özelliklerin kümeleme işlemindeki ağırlıkları, verinin doğası gereği veya kümeleme sonucunda elde edilmek istenen sonuçlara bağlı olarak, farklı olabilir. Bu bölümde, büyük hibrit verilerin etkili bir şekilde kümelenmesi ve bu kümeleme işlemine özelliklerin ağırlıklarının da dâhil edilebilmesi için geliştirilen bir algoritma tanıtacağız. Temel olarak K-Prototypes algoritmasına dayanan bu algoritmayı W-K-Prototypes olarak isimlendireceğiz.

6.1 Matematiksel Ön Hazırlık

n adet nesneden oluşan bir veriyi, $X = \{ x_i \mid 0 < i \leq n \}$ olacak şekilde X ile temsil edelim. Burada ki her bir $x_i = \{ x_{ij} \mid 0 < j \leq m \}$ olacak şekilde m adet özellikten oluşan bir vektördür ve açık olarak $x_i = [x_{i1}, x_{i2}, \dots, x_{im}] \in X$ notasyonu ile de gösterilebilir. Hiyerarşik olmayan kümeleme analizinde bizim problemimiz bu X kümesini önceden belirlenen bir k sayısı kadar kümeye ayırmaktır. Bu kümeleme işleminin kalitesini belirlemede ve kümeleme işlemine rehberlik etmede nesnel bir kritere ihtiyacımız vardır. Bunun için maliyet fonksiyonu denilen bir fonksiyon kullanılır [35]. Bu fonksiyon;

$$E = \sum_{l=1}^{k} \sum_{i=1}^{n} y_{i,l} d(x_i, Q_l) \quad (6.1)$$

olarak formüle edilir. Burada n , X kümesinin eleman sayısı $x_i \in X$ ve Q_l , l kümesinin merkezidir. $y_{i,l}$ ise bölüntüleme matrisinin elemanıdır. Y_{nxl} bölüntüleme matrisinin elemanları l'ninci elemanın n'inci kümeye ait olup olmaması durumuna göre sırasıyla 1 veya 0'dır. d uzaklık ölçüsü ise aksi belirtilmedikçe kare Öklid uzaklık ölçüsüdür. $y_{i,l}$ 0 ve 1 kapalı aralığında değer alır ve $\sum_{l=1}^{k} y_{i,l} = 1$'dir. Eğer $y_{i,l} \in \{0,1\}$ ise Y_{nxl} katı bölüntüleme (hard partition), eğer $y_{i,l} \in [0,1]$ ise bulanık bölüntülemedir (Fuzzy partition) denir. Biz bu dokümanda katı bölüntüleme ile ilgileneceğiz. Yani bir nesne ancak ve ancak bir kümeye ait olabilir.

Küme merkezi $Q_l = \{q_{l1}, q_{l2}, .., q_{lm}\}$ olacak şekilde;

$$q_{lj} = \frac{1}{n_l} \sum_{i=1}^{n} y_{il} x_{ij} \quad (6.2)$$

olarak hesaplanır. Burada n_l, l kümesinin eleman sayısıdır. Görüldüğü gibi hiyerarşik olmayan kümeleme yöntemlerinde nesneler arası uzaklığı belirlemek için bir uzaklık ölçüsüne ve her bir kümeyi temsil etmek üzere küme merkezlerini belirleme yöntemine ihtiyacımız vardır. Sadece nümerik özellikler içeren kümelerin vektörleri arası uzaklıkta kare öklit uzaklık ölçüsü kullanılır. Küme temsilcisi olarak da kümedeki vektörlerin ortalaması merkez olarak alınır. Kategorik veriye de sahip olan veriler için Huang tarafından önerilen uzaklık ölçüsü ise şu şekildedir;

$$d(x_i, Q_l) = \sum_{j=1}^{m_r} (x_{ij}^r - q_{lj}^r)^2 + \gamma_l \sum_{j=1}^{m_c} \delta(x_{ij}^c, q_{lj}^c) \quad (6.3)$$

Burada $\delta(x, y)$, x ve y kategorik değişkenleri aynı ise 0, farklı ise 1 değerini alan fonksiyondur. x_{ij}^r ve q_{lj}^r değerleri nümerik x_{ij}^c ve q_{lj}^c değerleri kategoriktir. m_r nümerik özelliklerin sayısı, m_c kategorik özelliklerin sayısıdır. γ_l ise l kümesi için kategorik verilerin ağırlığıdır.

Bu uzaklık ölçüsü kullanılırsa maliyet fonksiyonu şu şekilde yeniden yazılabilir [40];

$$E_l = \sum_{i=1}^{n} y_{il} \sum_{j=1}^{m_r} (x_{ij}^r - q_{lj}^r)^2 + \gamma_l \sum_{i=1}^{n} y_{il} \sum_{j=1}^{m_c} \delta(x_{ij}^c, q_{lj}^c) = E_l^r + E_l^c \quad (6.4)$$

Burada E_l^r nümerik özelliklerin maliyet fonksiyonu ve E_l^c kategorik özelliklerin maliyet fonksiyonudur. Bu maliyet fonksiyonu göz önüne alındığında E_l^c maliyet fonksiyonunu

minimize etmek için seçilmesi gereken merkezin kategorik özelliklerinin küme içinde en sık rastlanan (çünkü sigma fonksiyonu aynı olması durumunda 0 değerini veriyor ve toplama katkıyı azaltıyor.) değerlerden seçilmesi gerektiği anlaşılıyor. Buna dayanarak Huang'ın merkez seçme önerisi; nümerik bileşenlerin ortalamasının ve kategorik bileşenlerin frekansı en yüksek olanlarının seçilmesi.

(3) denklemindeki γ_l katsayısı ise kategorik verilerin uzaklık ölçüsündeki ağırlığını belirlemek amacıyla önerilmiştir. Örneğin nümerik değişkenlerin değer aralıkları çok geniş olduğunda kategorik değişkenlerin farklı olması durumunda uzaklığa ekleyecekleri 1 değeri çok anlamsız ve küçük kalabilir. Tersine nümerik değerlerin [0.5,0.6] gibi küçük bir aralıkta değişmesi durumunda 1 değeri uzaklık belirlemede daha baskın rol oynayacağı için istenmeyen bir durum oluşturur. İşte bu durumun önüne geçmek için Huang'ın önerisi γ_l değerinin belirlenmesinde σ_l standart sapma değerinin (l kümesindeki nümerik özelliklerin standart sapmalarının ortalaması) kullanılmasıdır. Bu sayede kategorik değişkenlerin farklı olması durumunda uzaklık ölçüsüne etkisi hemen hemen nümerik değerlerin farklı olduğu durumdaki kadar olacaktır. Bu önerinin her zaman doğru sonuç vermesi beklenemeyeceği gibi tüm veri dikkate alındığında iyi bir çözüm olduğu açıktır. Huang, çalışmasında γ_l'nin farklı değerleri için bazı sonuçlar paylaşmıştır.

6.2 Verinin Standartlaştırılması

Değişkenlerin standartlaştırılması ile ilgili olarak çeşitli yöntemler olmakla birlikte bunlardan en bilineni Z-Score standartlaştırmasıdır. Bu yöntemle veriler ortalaması 0 standart sapması 1 olacak şekilde dönüştürülür. Bu sayede verideki özelliklerin ortalamanın altında veya üstünde olduğu basit bir şekilde işaretine bakılarak anlaşılabilir. Z-Score standartlaştırmasında özelliklerin yeni değerleri aşağıdaki formülle hesaplanır.

$$x' = \frac{x - mean(X)}{sd(X)} \tag{6.5}$$

Burada $mean(X)$; X veri kümesinin ortalaması, $sd(X)$; X veri kümesinin standart sapmasıdır. Z-Score yöntemi veriyi tanınmayacak hale getirse de uzaklık hesaplamalarında en iyi sonucu verir. Kümeleme işlemi sonucunda hangi verinin hangi

kümede olduğunu anlayabilmek için verilerin bir ID değeri ile işleme sokulması ve bu ID değerinin uzaklık hesaplamalarına katılmaması bir yöntemdir.

Hibrit verilerde verinin sadece nümerik kısmının standartlaştırılması yeterlidir. Z-Score standartlaştırılmasındaki formülde kategorik verilerin standartlaştırılamayacağı açıktır. Kategorik veriler için kullanılan uzaklık ölçüsünün farklı olması nedeniyle standartlaştırılmamaları bir probleme neden olmaz.

6.3 W-K-Prototypes Algoritması

Bazen çalışılacak verinin özellikleri eşit önem derecesine sahip olmayabilir. Bu önem derecesi veriden çıkarılmak istenen bilginin kapsamına göre de değişiklik gösterebilir. Örneğin öğrenci bilgilerinin tutulduğu bir veride yapılacak kümeleme işlemini düşünelim. Eğer öğrencilerin başarı durumlarına göre bir gruplama yapılmak isteniyorsa, kilo, boy, okul numarası gibi özelliklerin kümeleme işlemine olan etkisinin daha az olması gerekir. Hatta veriden çıkarılması (0 ağırlık) bile istenebilir. Öğrencilerin fiziksel olarak benzerliklerine göre yapılacak bir gruplama işleminde ise ders notlarının kümeleme işlemine etkisinin daha az olması gerekir. Bunun yanında bazı verilerin doğasında, kümeleme işleminin kapsamı dikkate alınmaksızın, bir ağırlık dağılımı olabilir. Böyle durumların da kümeleme işleminde göz önüne alınması gerekir [50].

Huang'ın K-Prototypes algoritmasında kullandığı uzaklık ölçüsündeki γ_l ağırlığı, kategorik verilerin tümünün, uzaklığa olan etkisi anlamında bir ağırlık olup, tam olarak yukarıda bahsettiğimiz ağırlık değildir. γ_l 'nin her adımda her küme için yeniden hesaplanması gerekliliği algoritmaya ciddi bir yük bindirir. Zaten Huang, çalışmasında algoritmasının testi için her adımda γ_l hesaplamak yerine işlem boyunca kullanılacak tek bir γ değeri kullanmıştır. Bizim W-K-Prototypes olarak isimlendireceğimiz algoritmamızın kullanacağı nesneler arası uzaklık ölçüsünde böyle bir γ ağırlığına gerek yoktur. Biz veriyi W-K-Prototypes algoritmasıyla kümelemeden önce Z-Score yöntemiyle standartlaştırarak, bu katsayıya olan ihtiyacı ve kullanılması durumunda ortaya çıkan γ'nın değerinin ne olacağı problemini ortadan kaldırıyoruz. Z-Score normalleştirmesi sayesinde ortalama standart sapma değeri zaten 1 olacağından aslında γ katsayısını her adımda 1 alıyormuşuz gibi düşünebiliriz.

$$d(x,y) = \sum_{i=1}^{m_{num}} w_i (x_i^r - y_i^r)^2 + \sum_{i=1}^{m_{cat}} w_i \delta(x_i^c, y_i^c) \quad (6.6)$$

Burada w_i , i'ninci özelliğin ağırlığıdır ve kategorik ve nümerik tüm özellikler için $\sum_{i=1}^{m} w_i = 1$ 'dir. Kategorik verilerin uzaklığı için kullanılan uzaklık fonksiyonu;

$$\delta(x, y) = \begin{cases} 0 & if\ x = y \\ 1 & else \end{cases} \tag{6.7}$$

olarak tanımlanır. İki nesne arası bu uzaklık ölçüsü kullanılmak üzere algoritma adımları K-Prototypes algoritması ile aynıdır;

(i) X veri kümesi içinden kümeleri temsil etmek üzere k adet başlangıç merkezi (prototype) seçilir.

(ii) X veri kümesindeki her bir nesne, en yakın oldukları prototype'ın bulunduğu kümeye atanır.

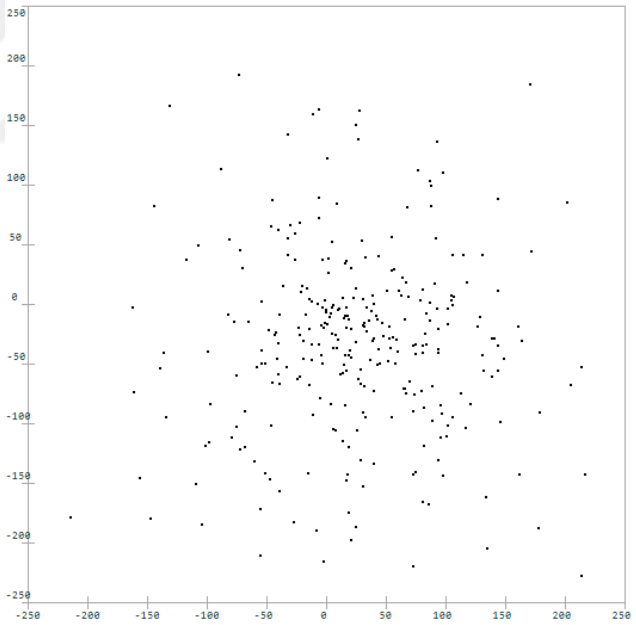
(iii) Bütün nesneler atandıktan sonra kümelerin prototype'ları yeniden hesaplanır. Ve her bir nesne yeniden en yakın oldukları prototype'ın bulunduğu kümeye atanır.

Hiçbir nesne yer değiştirmeyinceye kadar adım (iii) tekrarlanır.

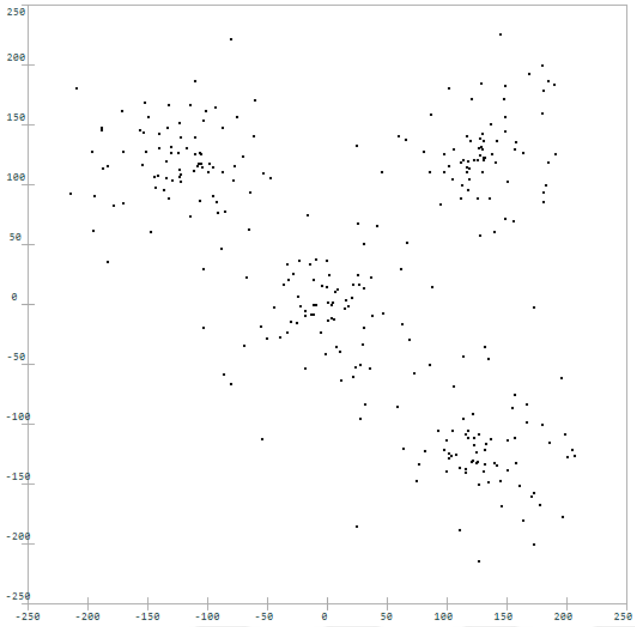
BÖLÜM 7

SONUÇ VE ÖNERİLER

W-K-Prototypes algoritmasının testi ve K-Prototypes algoritması ile karşılaştırılması için kullanılacak veriyi yapay olarak oluşturacağız. Bunun için Huang'ın çalışmasında kullandığı yapay verilerin bir benzerini üreteceğiz [40]. Üretilen verilerin dağılımı Şekil 7.1 ve Şekil 7.2'de gösterilmiştir.

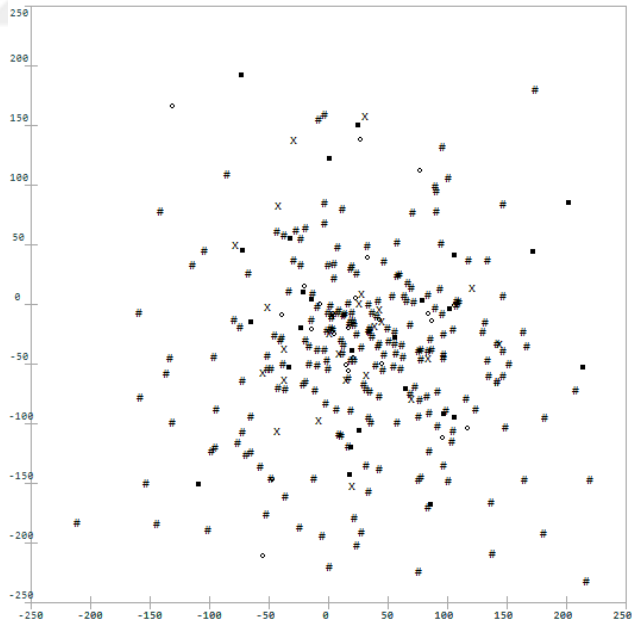


Şekil 7.1 Bir normal dağılımlı örneklem

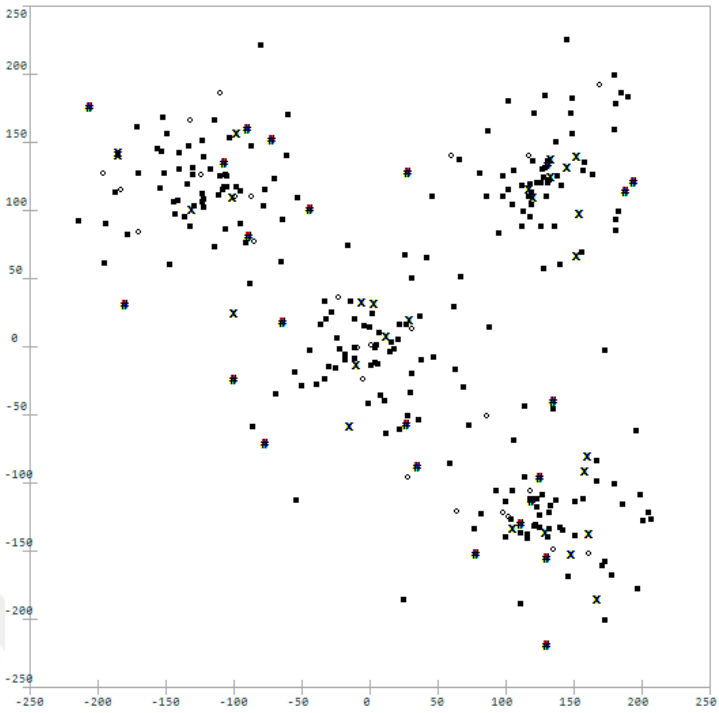


Şekil 7.2 Dört normal dağılımlı örneklem

Bu verilere kategorik üçüncü boyutu eklediğimizde oluşan dağılımı da Şekil 7.3 ve Şekil 7.4'de görülebilir. Üçüncü boyut olan kategoriyi grafik üzerinde gösterebilmek için noktaların şekillerini kullanacağız.

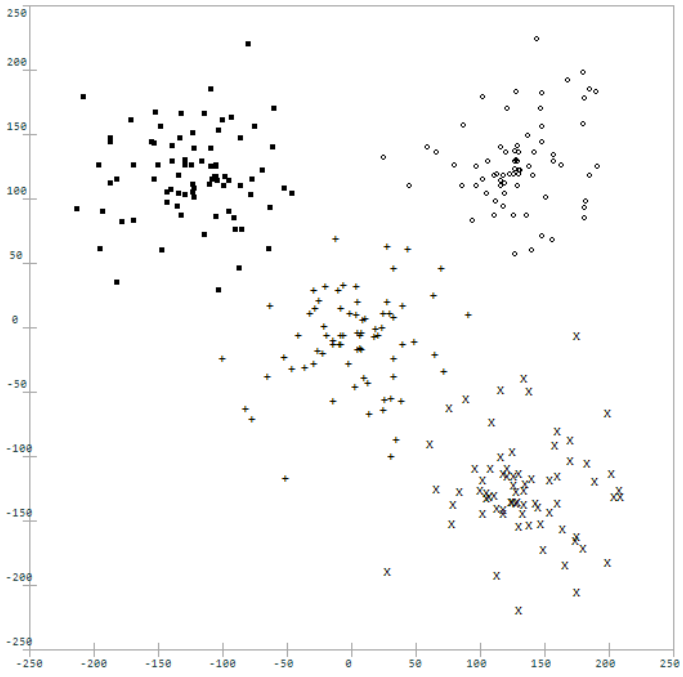


Şekil 7.3 Bir normal dağılımlı üç boyutlu örneklem

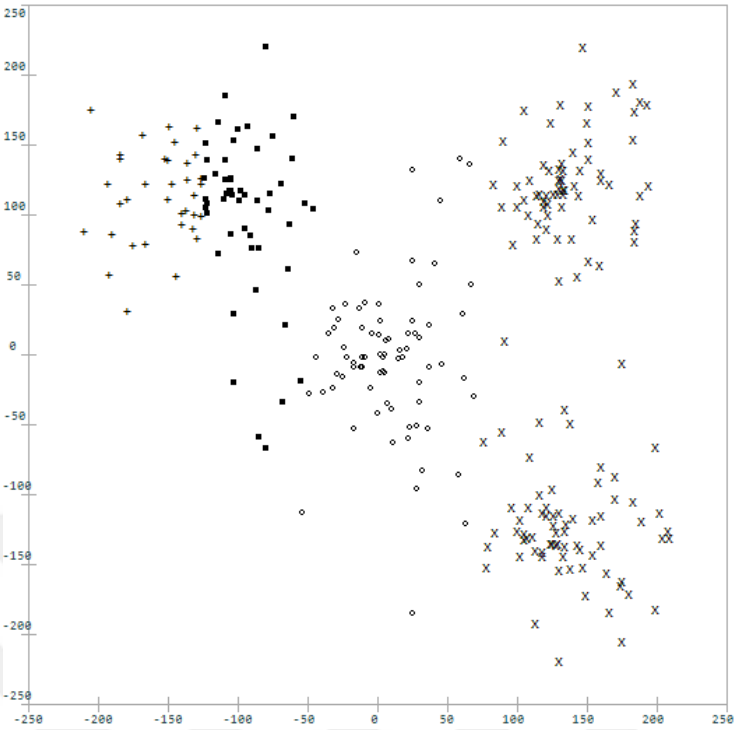


Şekil 7.4 Dört normal dağılımlı üç boyutlu örneklem

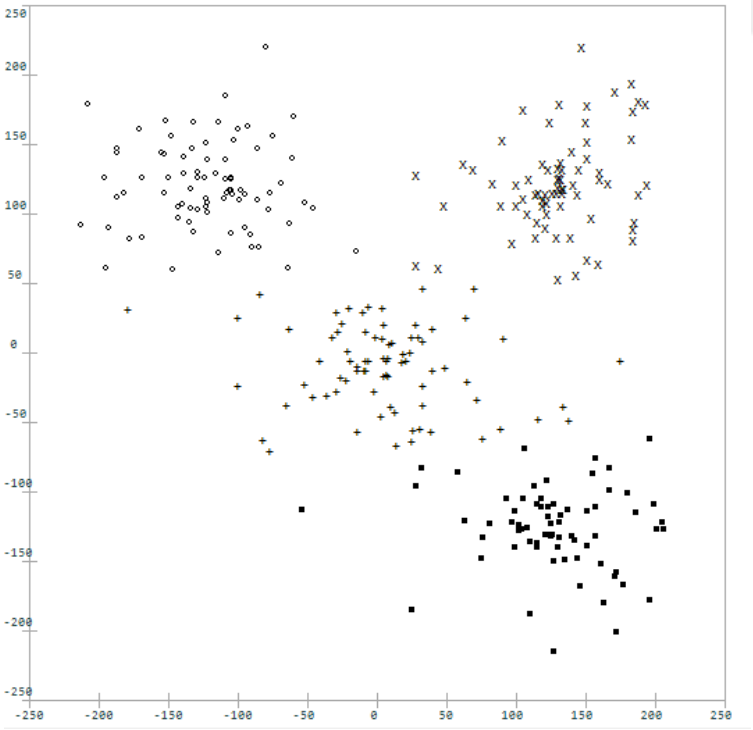
Bu verileri özelliklerin farklı ağırlık değerleri için kümeleyerek ortaya çıkan kümeleme sonuçları, dört normal dağılımlı örneklem için Şekil 7.5, 7.6, 7.7, 7.8, 7.9 ve Şekil 7.10'da ve bir normal dağılımlı örneklem için de Şekil 7.11, 7.12, 7.13, 7.14, 7.15 ve Şekil 7.16'da gösterilmiştir.



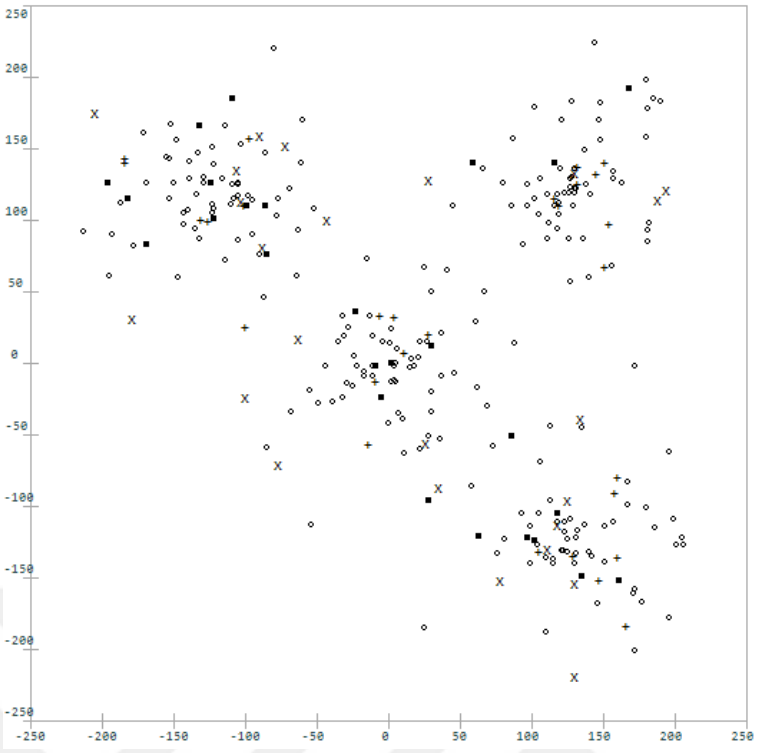
Şekil 7.5 Ağırlık vektörü w=[0.33,0.33,0.34] alındığında oluşan dağılım



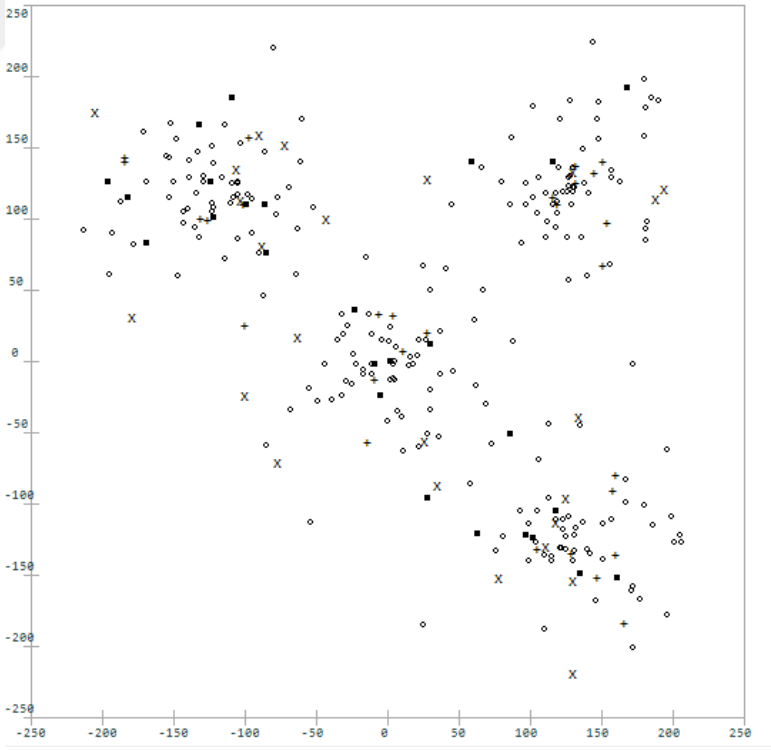
Şekil 7.6 Ağırlık vektörü w=[0.8,0.1,0.1] alındığında oluşan dağılım



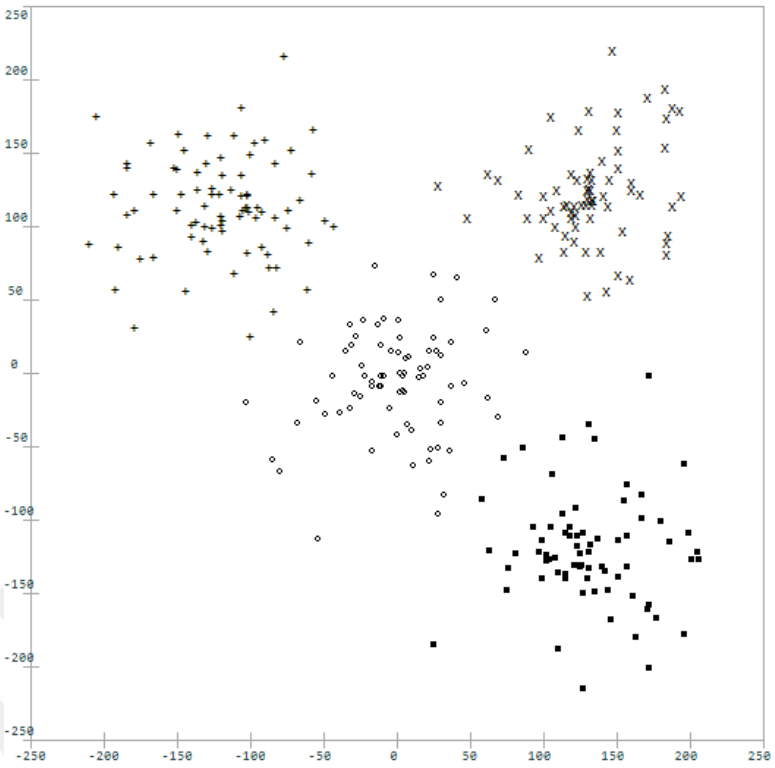
Şekil 7.7 Ağırlık vektörü w=[0.1,0.8,0.1] alındığında oluşan dağılım



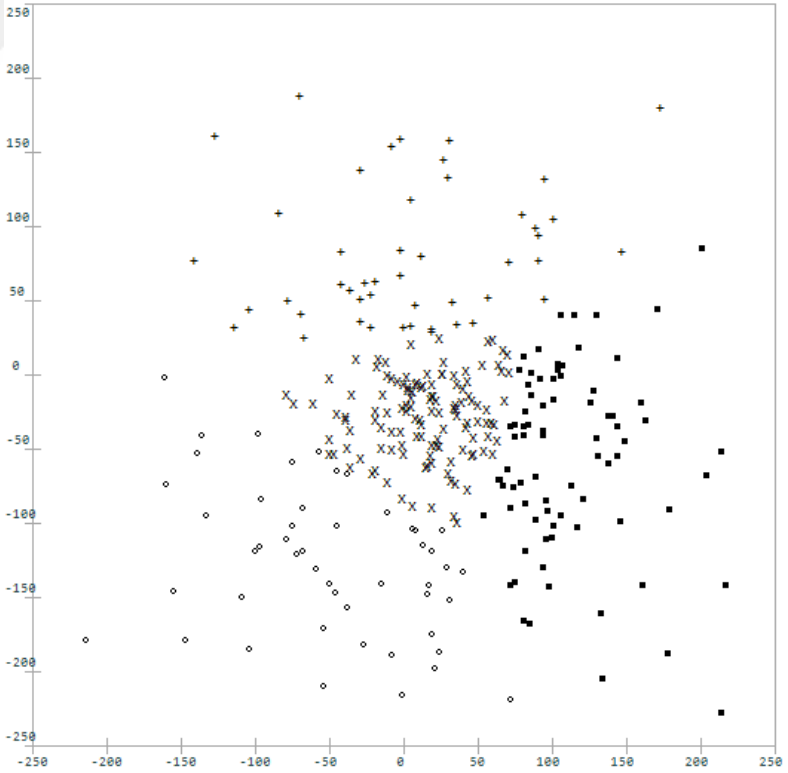
Şekil 7.8 Ağırlık vektörü w=[0.1,0.1,0.8] alındığında oluşan dağılım



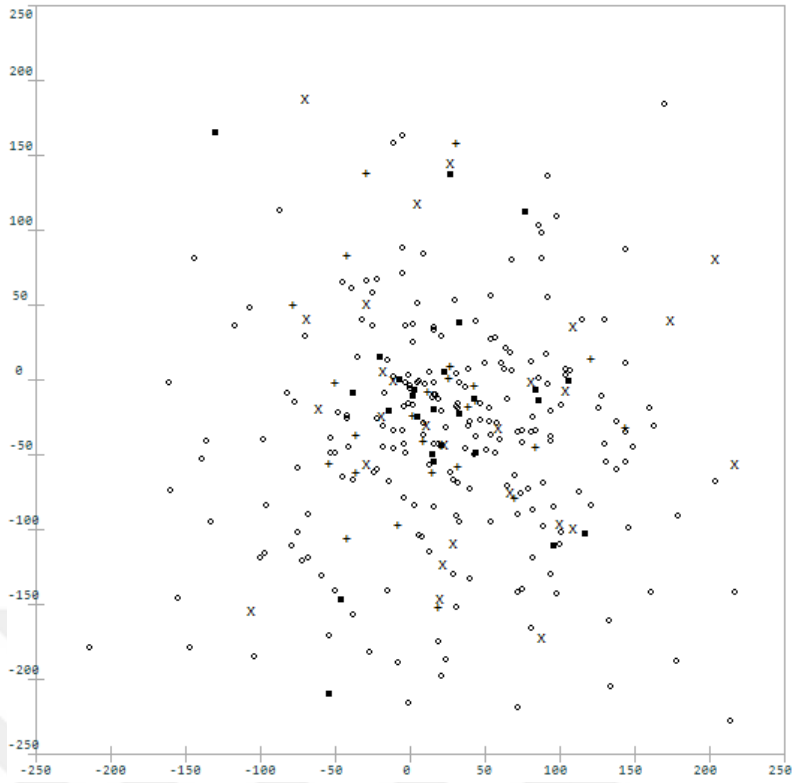
Şekil 7.9 Ağırlık vektörü w=[0.2,0.2,0.6] alındığında oluşan dağılım



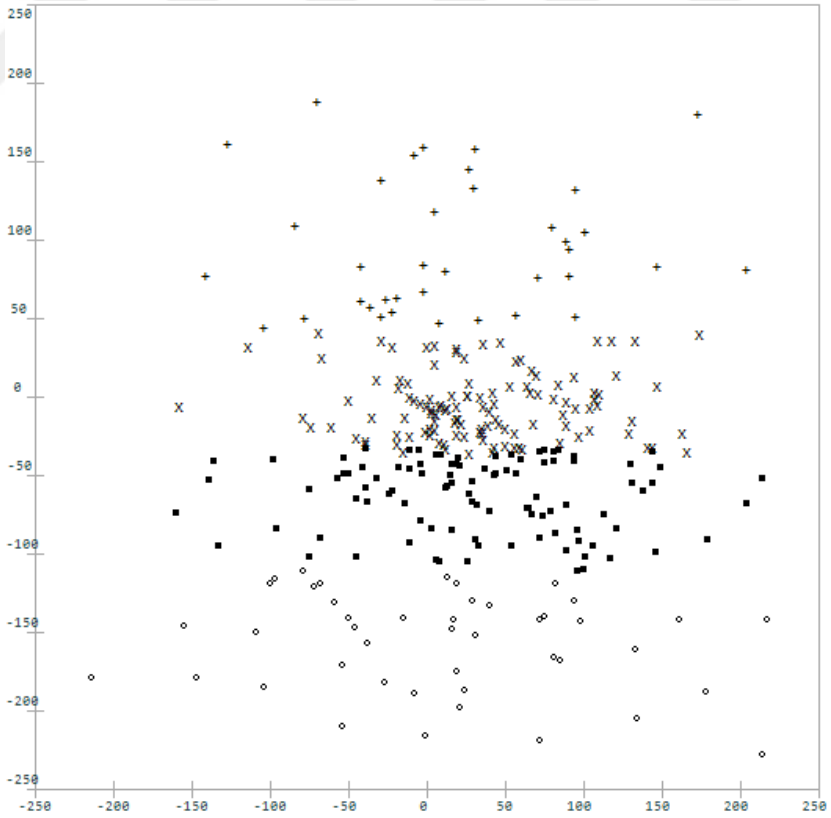
Şekil 7.10 Ağırlık vektörü w=[0.0,0.5,0.5] alındığında oluşan dağılım



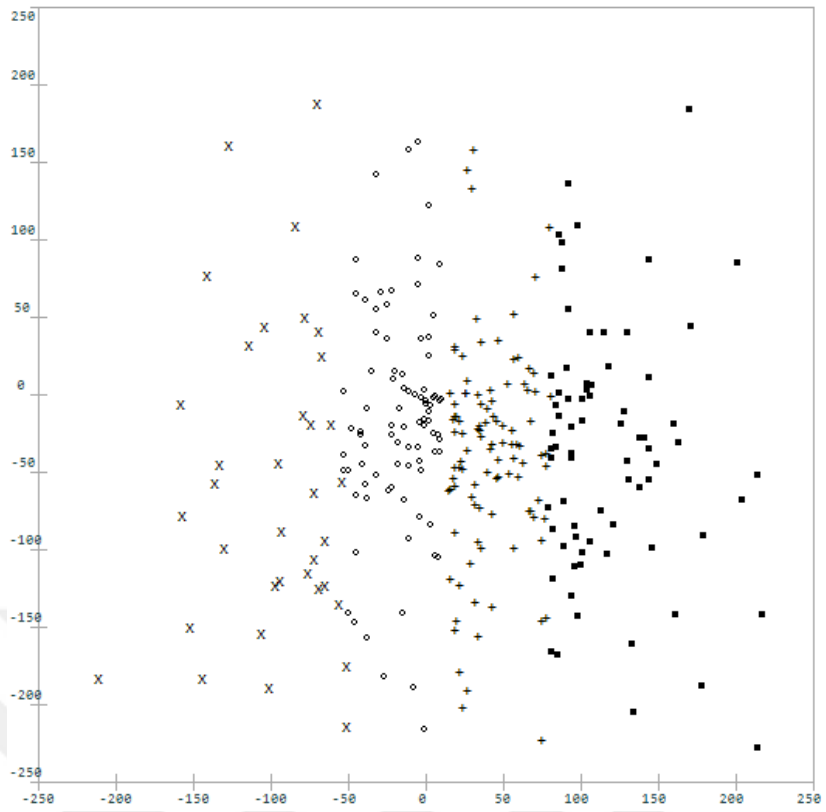
Şekil 7.11 Ağırlık vektörü w=[0.33,0.33,0.34] alındığında oluşan dağılım



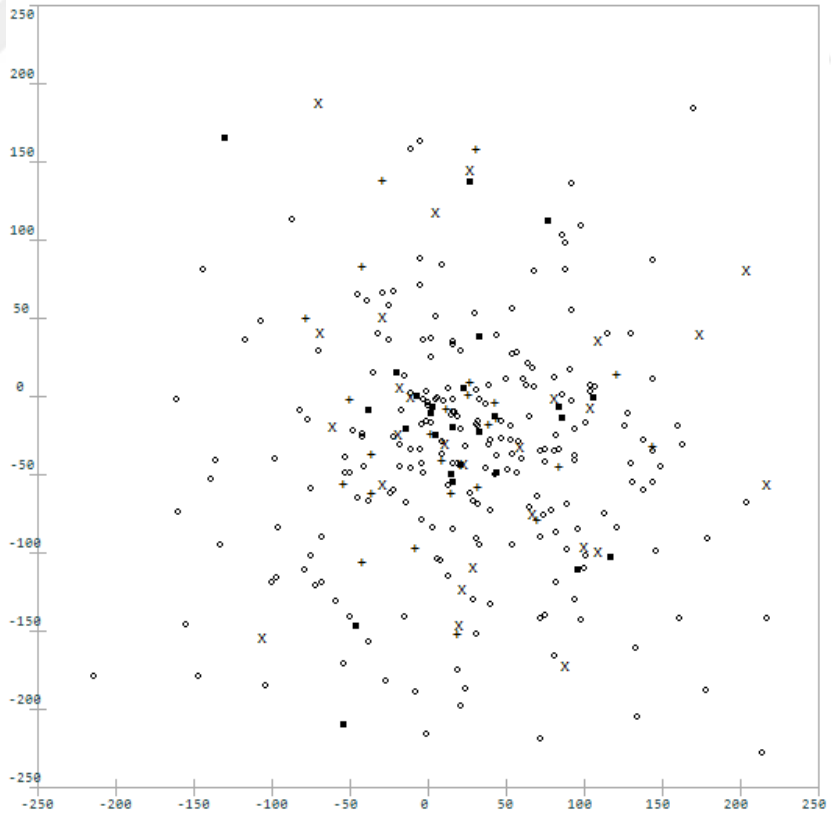
Şekil 7.12 Ağırlık vektörü w=[0.8,0.1,0.1] alındığında oluşan dağılım



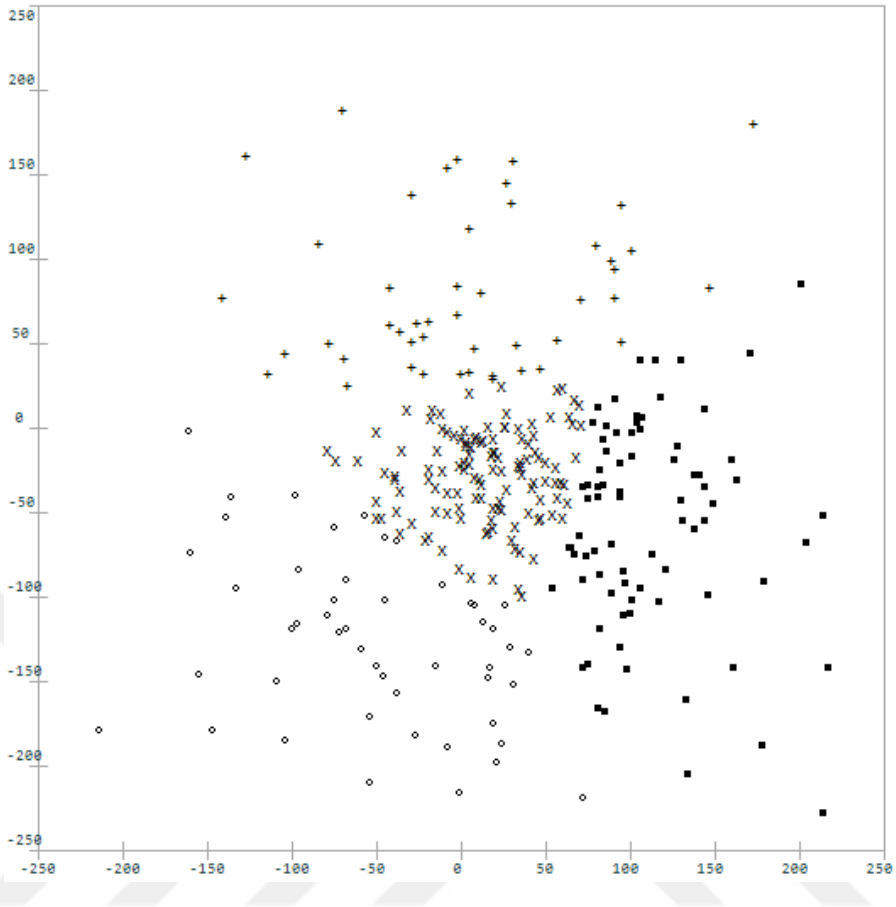
Şekil 7.13 Ağırlık vektörü w=[0.1,0.8,0.1] alındığında oluşan dağılım



Şekil 7.14 Ağırlık vektörü w=[0.1,0.1,0.8] alındığında oluşan dağılım



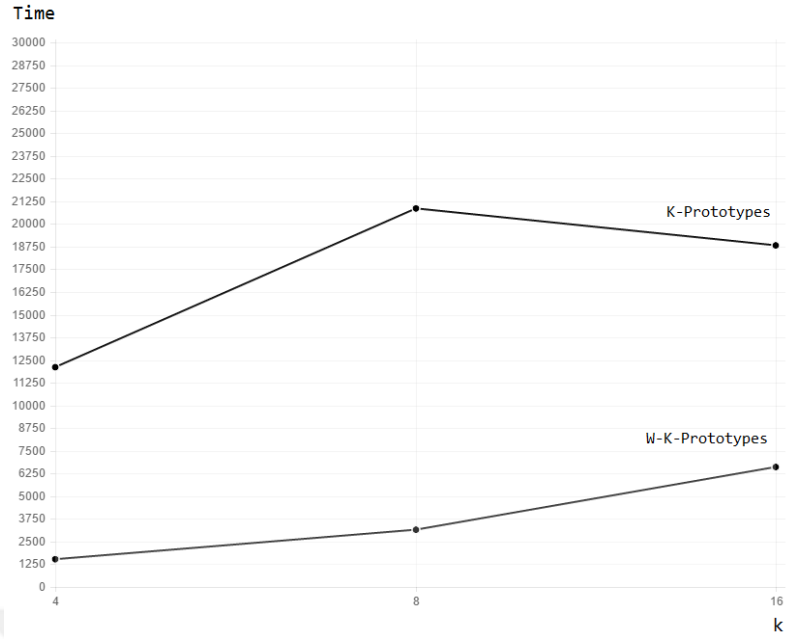
Şekil 7.15 Ağırlık vektörü w=[0.2,0.2,0.6] alındığında oluşan dağılım



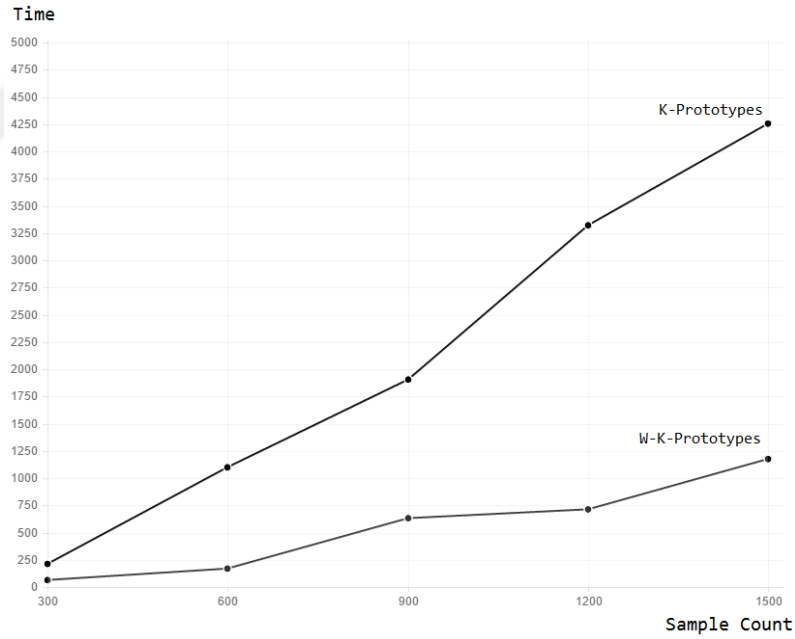
Şekil 7.16 Ağırlık vektörü w=[0.0,0.5,0.5] alındığında oluşan dağılım

Algoritmamızın girdileri, k küme sayısı, X veri kümesi ve her bir özelliğe bir ağırlık karşılık gelecek şekilde bir ağırlık vektörü $w = [w_1, w_2, \ldots, w_m]$'dür. Oluşturulan iki yapay veri kümesinin, farklı k değerleri ve farklı w ağırlık vektörleri ile kümeleme işlemi sonuçları yukarıdaki şekillerde gösterilmiştir.

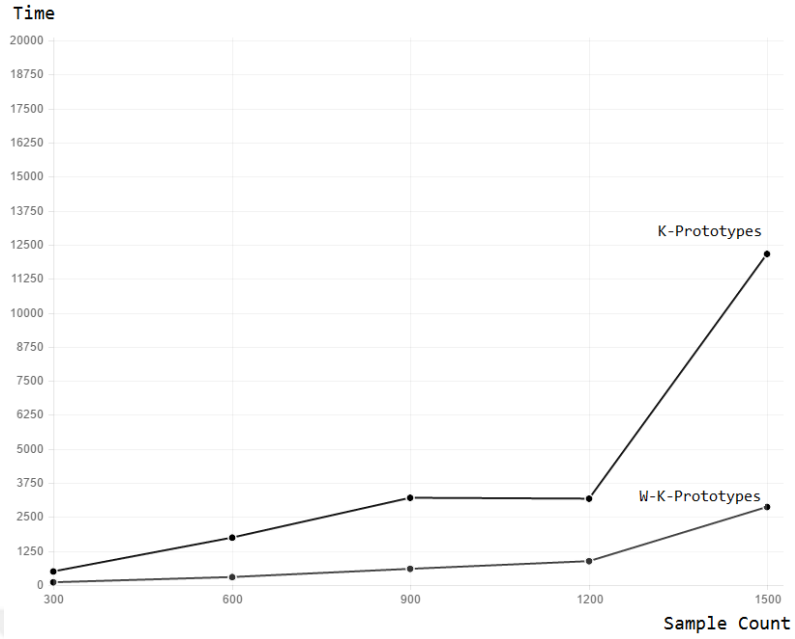
Bununla birlikte küme sayısı, k değeri, nesne sayısı ve bu girdilerin değişiminin çalışma süresine olan etkisi de Çizelge 7.1'de ve Şekil 7.17, 7.18, 7.19, 7.20, 7.21 ve Şekil 7.22'de gösterilmiştir.



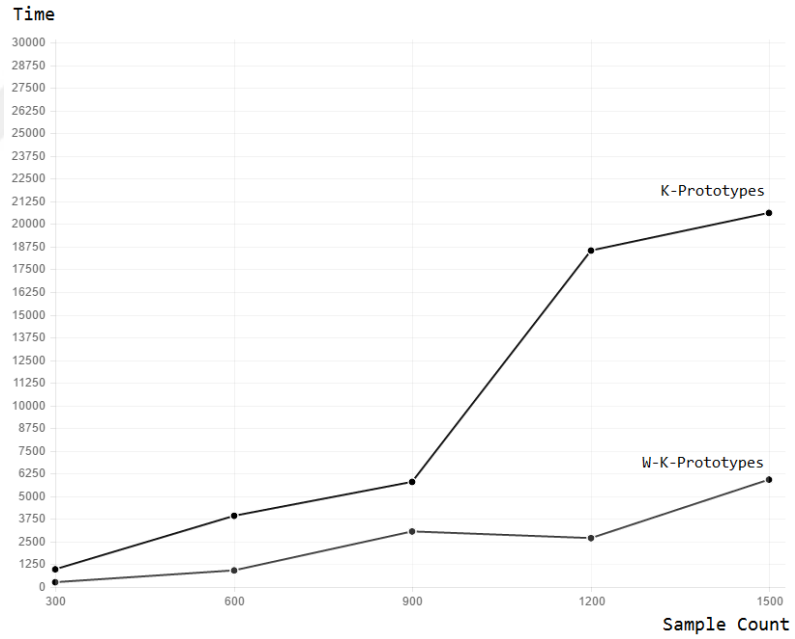
Şekil 7.17 Bir normal dağılım ve 1500 örnek ile yapılan test sonucu



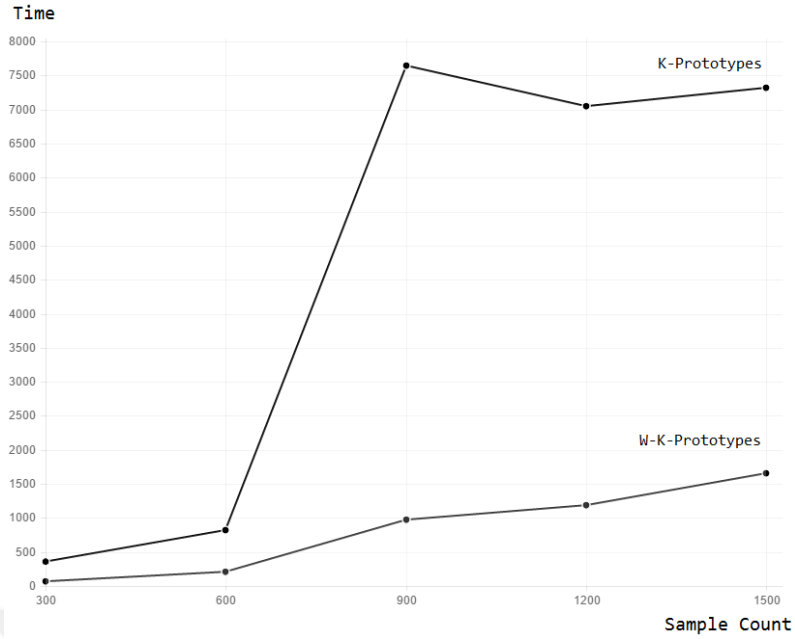
Şekil 7.18 Dört normal dağılım ve k=4 ile yapılan test sonucu



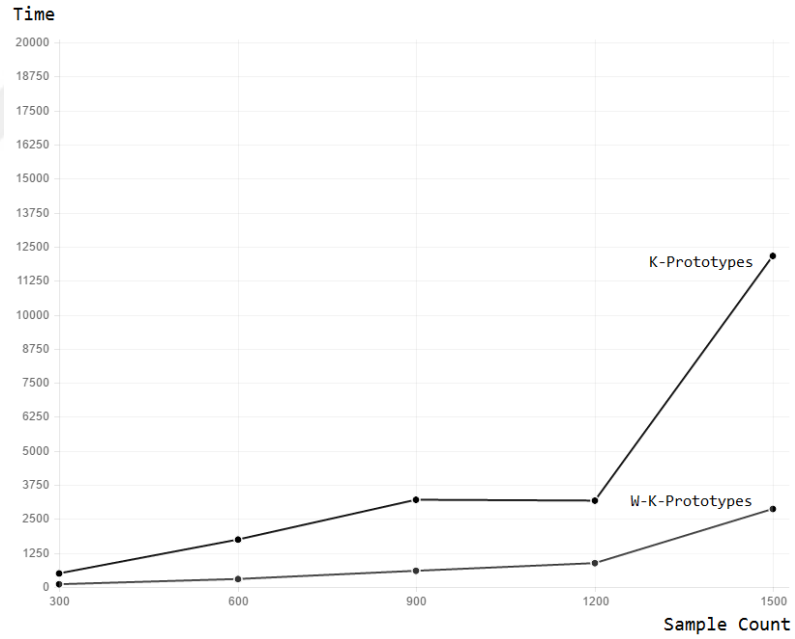
Şekil 7.19 Dört normal dağılım ve k=8 ile yapılan test sonucu



Şekil 7.20 Dört normal dağılım ve k=16 ile yapılan test sonucu



Şekil 7.21 Sekiz normal dağılım ve k=4 ile yapılan test sonucu



Şekil 7.22 Sekiz normal dağılım ve k=8 ile yapılan test sonucu

Bu algoritmaların testi için kullanılan bilgisayar Intel Core i7 işlemci, 32 GB bellek, Windows 8.1 (64 bit) işletim sistemi ve 8 çekirdeğe sahip bir kişisel bilgisayardır.

İlk örnek olarak dört normal dağılımlı X veri kümesi için kümeleme işlemi sonuçlarımızı inceleyelim. Bu veri kümesinin nümerik özellikleri, iki boyutta 4 normal dağılım

gösterecek kümenin birleşimi şeklinde oluşturulmuştur. Bu yüzden $k = 4$ değeri ile farklı ağırlık değerleri için sonuçlar elde edelim.

Aşağıdaki sonuçların gösteriminde kullanılan şekiller incelendiğinde, $w = [0.33, 0.33, 0.34]$ ağırlık vektörü için (Şekil 7.5) tüm özelliklere eşit ağırlık verilmiş ve kendi içinde normal dağılıma sahip 4 küme ayrılmıştır. $w = [0.2, 0.2, 0.6]$ ağırlık vektörü için (Şekil 7.9) kategorik özelliğe ağırlık verilmiş ve beklediğimiz bir sonuç olarak nümerik verilerinde kümelemede etkili olmasına rağmen nesneler iç içe bir kümelenme göstermişlerdir. $w = [0.8, 0.1, 0.1]$ ağırlık vektöründe ise (Şekil 7.6) birinci boyuta baskın bir ağırlık verilmiş ve kümelenme sonucu da beklentiye uygun olarak yatay eksene göre bir dağılım göstermiştir. $w = [0.1, 0.8, 0.1]$ ağırlık vektörü için (Şekil 7.7) ikinci boyuta baskın ağırlık verilmiş ve bir önceki sonuca benzer olarak bu kez dikey eksene göre bir dağılım görülmüştür. Ağırlık vektörünü belirlerken bir özelliğin kümeleme işleminde hiçbir ağırlığının olmadığı anlamında 0 olarak belirlenmesini de gözlemlemek için $w = [0.0, 0.5, 0.5]$ olarak seçilmiş ve sonucu Şekil 7.10 ile gösterilmiştir. $w = [0.1, 0.1, 0.8]$ ağırlık vektörü ile de kategorik özelliğin kümeleme işleminde baskın bir özellik olarak belirlenmesi durumunda karşılaşılan dağılım Şekil 7.8 ile gösterilmiştir.

Şekil 7.2 ve Şekil 7.4'de verilen ve yukarıda farklı ağırlıklar için aldığımız sonuçlara benzer olarak, beklenildiği gibi baskın olan özelliğe uygun kümelenme sonuçları gözlemlenmiştir. Şekil 7.11'de $w = [0.33, 0.33, 0.34]$ için yapılan kümeleme işleminin sonuçları verilmiş ve düzgün bir şekilde dağılmış kümeler gözlemlenmiştir. Şekil 7.15'de $w = [0.2, 0.2, 0.6]$ için yapılan kümeleme işleminin sonuçları verilmiş ve beklenildiği gibi ilk iki özellik bakımından birbirine yakın olan nesnelerin kümesini belirleme de kategorik veri baskın davranmıştır. Şekil 7.12 ve Şekil 7.13'de ise ilk deney verimizdeki gibi sırasıyla birinci ve ikinci özelliklerin baskın davrandığı ve kümelerin önce yatay sonra dikey olarak gruplanan nesnelerden oluştuğu gözlemlenmiştir.

Çizelge 7.1 Farklı girişler ile K-Prototypes ve W-K-Prototypes için test sonuçları

Cluster Count	Sample Count	k	İteration Count		Estimation Time (milliseconds)	
			K-Prototypes	W-K-Prototypes	K-Prototypes	W-K-Prototypes
4	300	4	8	12	212	64
4	600	4	12	7	1099	169
4	900	4	10	14	1905	632
4	1200	4	10	9	3321	713
4	1500	4	8	10	4254	1175
8	300	8	18	14	499	100
8	600	8	18	12	1742	293
8	900	8	17	14	3203	594
8	1200	8	9	10	3171	876
8	1500	8	23	20	12158	2866
4	300	8	13	13	342	84
4	600	8	19	22	1908	550
4	900	8	19	8	3854	443
4	1200	8	15	24	5418	2459
4	1500	8	34	20	19322	2646
8	300	4	15	11	359	68

Çizelge 7.1 Farklı girişler ile K-Prototypes ve W-K-Prototypes için test sonuçları (devamı)

8	600	4	10	9	821	209
8	900	4	40	23	7644	974
8	1200	4	22	15	7048	1187
8	1500	4	14	12	7319	1656
4	300	16	32	34	975	263
4	600	16	37	34	3921	915
4	900	16	27	56	5790	3059
4	1200	16	51	33	18534	2692
4	1500	16	36	42	20608	5915
1	1500	4	22	13	12108	1527
1	1500	8	38	21	20851	3154
1	1500	16	32	44	18817	6609
1	3000	16	63	26	138977	15029

Çizelge 7.1'deki ve Şekil 7.17, Şekil 7.18, Şekil 7.19, Şekil 7.20, Şekil 7.21 ve Şekil 7.22 'deki sonuçlar incelendiğinde W-K-Prototypes algoritmasının K-Prototypes algoritmasına işlem süresi açısından bariz bir üstünlüğü görülür. W-K-Prototypes algoritmasının, sürelerin oranlarının ortalaması alındığında yaklaşık 5 kat daha hızlı çalıştığı gözlemlenmiştir. Şekil 7.21'de görüldüğü gibi, özellikle kendi içinde normal dağılmış çok sayıda kümeyi, nispeten daha az sayıda kümeye ayırma işleminde W-K-Prototypes algoritmasının çok daha etkili sonuçlar verdiği görülmüştür.

Kümeleme sonuçları incelendiğinde ise K-Prototypes ve W-K-Prototypes algoritmalarının hemen hemen aynı sonuç kümelerini ürettiği gözlemlenmiştir. Çizelge 7.1'deki W-K-Prototypes algoritmasının işlem süresine, Z-Score ve tersine Z-Score işlemlerinin süresi de dâhildir.

Bu dokümanda, büyük boyutlardaki hibrit verilerin kümelenmesinde, verinin özelliklerinin farklı ağırlık dereceleri ile kümeleme işlemine dâhil edilmesini sağlayacak bir algoritma tanıttık. Temelde K-Prototypes algoritmasına dayanması nedeniyle W-K-Prototypes olarak isimlendirdiğimiz bu algoritma, hiyerarşik olmayan bir yöntemle ve K-Prototypes'daki γ_l katsayısına ihtiyaç duymaksızın etkili bir şekilde kümeleme işlemi yapmaktadır. Bu algoritmayı oluştururken, K-Prototypes algoritmasında kullanılan uzaklık ölçüleri, küme merkezi belirleme yöntemi ve kategorik verilerin uzaklığa olan etkisi anlamındaki γ_l katsayısı ayrı ayrı bileşenler olarak düşünülmüş ve buradan yola çıkılmıştır. Benzeri şekilde, ileride yapılacak çalışmalarda verinin ve beklenen sonucun kapsamına göre kullanılan bu bileşenler değiştirilebilir. Örneğin kategorik verilerin uzaklığını hesaplamak için, özellikteki değerlerin frekansını da dikkate alan başka bir ölçü kullanılabilir. Ya da nümerik özelliklerin dağılımına göre kare öklit uzaklık ölçüsü yerine başka bir uzaklık ölçüsü kümeleme kalitesini arttırmak için kullanılabilir [31]. W-K-Prototypes algoritmasını uygulayabilmek için yapılan Z-Score standartlaştırmasının büyük veriler için bir dezavantaj olduğu açıktır. Bunun yerine K-Prototypes'daki γ katsayısının etkili bir şekilde belirlenmesi üzerine çalışmalar yapılabilir.

KAYNAKLAR

[1] Öğüt, S., (2009), Veri Madenciliği Kavramı ve Gelişim Süreci, http://www.sertacogut.com/blog/wp-content/uploads/2009/03/sertac_ogut_-_veri_madenciligi_kavrami_ve_gelisim_sureci.pdf, 26 Mart 2017.

[2] Linoff, G. S. and Berry, M. J. A., (2011). Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management, Third Edition, John Wiley and Sons, Londra.

[3] Sever, H. ve Oğuz, B., (2002), "Veritabanlarında Bilgi Keşfine Formal Bir Yaklaşım, Kısım 1: Eşleştirme Sorguları ve Algoritmalar", Bilgi Dünyası, 3(2):173-204.

[4] Özkan, Y., (2008). Veri Madenciliği Yöntemleri, Üçüncü Baskı, Papatya Yayıncılık, İstanbul.

[5] Çam, S., (2016). "Veri Madenciliğinde Kümeleme Analizi ve Sağlık Sektöründe Bir Uygulaması", Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi, 17(2):85-96.

[6] Silahtaroğlu, G., (2004). Veri Madenciliğinde Kümeleme Analizi ve Öğretim Başarısını Değerlendirmeye İlişkin Bir Uygulama, Doktora Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul.

[7] Anderberg, M., (1973). Cluster Analysis for applications, Birinci Baskı, Academic Press, New York.

[8] Kaufman, L. ve Rousseuw, P.J., (2008). Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley and Sons, New Jersey.

[9] Kantardzic, M., (2003). Data Mining: Concepts, Models and Algorithms, IEEE Press and John Wiley, New York.

[10] Tatlıdil, H., (1996). Uygulamalı Çok Değişkenli Analiz, Birinci Baskı, Nadir Kitap, Ankara.

[11] Romesburg, H., (2004). Cluster analysis for researchers, Lulu Press, Morrisville.

[12] Everitt, B., (2011). Cluster Analysis, Beşinci Baskı, John Wiley and Sons, Londra.

[13] Hartigan, J.A., (1975). Clustering Algorithms, John Wiley and Sons, New York.

[14] Data Preparator, http://www.datapreparator.com/what_is_data_preparation.html, 26 Mart 2017.

[15] Young, R. ve Johnson, D. R., (2010). Imputing the Missing Y's: Implications for Survey Producers and Survey Users, 64th Annual Conference of the American Association for Public Opinion Research, Pennsylvania.

[16] Ader, H. J. ve Mellenbergh, G. J., (2008). Advising on research methods: A consultant's companion, Johannes van Kessel Publishing, Huizen.

[17] Chakrabarti, S., Cox. E., Frank, E., Güting, R. H., Han J., Kamber, M., Lighstone, S. S., Nadeau, T. P., Neapolitan R. E., Pyle, D., Refaat, M., Schneider, M., Teorey, T. J. ve Witten I. H., (2009). Data Mining Know It All, Birinci Baskı, Morgan Kaufmann, Massachusetts.

[18] Slide Player, http://slideplayer.com/slide/6656610/, 26 Mart 2017.

[19] Michalski, R. S. ve Stepp, R. E., (1983). "Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy", IEEE Transactions on Pattern Analysis and Machine Intelligence, 5(4):396-410.

[20] Hofman I. ve Jarvis, R. (1998)., Robust and Efficient Cluster Analysis Using a Shared Near Neighbours Approach. 14. Pattern Recognition Conference-Volume 1, 16-20 Ağustos 1998, Washington.

[21] Han, J. ve Kamber, M., (2006). Data Mining: Concepts and Techniques, İkinci Baskı, Morgan Kaufmann Publishers, San Francisco.

[22] Rezankova H., (2014). "Evaluation of Selected Approaches to Clustering Categorical Variables", Statistics in Transition, 15:591-610.

[23] Everitt, B. Landau, S., ve Leese, M., (2001). Cluster Analysis, Oxford University Press, London.

[24] Jain A. K. ve Dubes R. C., (1988). Algorithms for Clustering Data. Prentice-Hall, New Jersey.

[25] Atbaş, A. C. G., (2008). Kümeleme Analizinde Küme Sayısının Belirlenmesi Üzerine Bir Çalışma, Yüksek Lisans Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

[26] Ghahramani, Z. (2004)., "Unsupervised Learning", Advances Lectures on Machine Learning, 3-32.

[27] Wikipedia, Cluster Analysis, http://en.wikipedia.org/wiki/Cluster_analysis, 29 Mart 2017.

[28] Özdamar, K., (1999). Paket Programlar ile İstatistiksel Veri Analizi 2 (Çok Değişkenli Analizler), İkinci Baskı, Kaan Kitabevi, Eskişehir.

[29] Stanfill C. ve Waltz D., (1986). "Toward Memory-Based Reasoning", Magazine Communications of the ACM, 29(12):1213-1228.

[30] Alpar, R., (2011). "Uygulamalı, Çok Değişkenli İstatistiksel Yöntemler", Detay Yayıncılık, Ankara.

[31] Boriah, S., Chandola, V. ve Kumar, V., (2008). Similarity Measures for Categorical Data A Comparative Evaluation, In Proceedings of the 8th SIAM International Conference on Data Mining, SIAM, 24-26 Nisan 2008, Atlanta.

[32] Wilson D. R. ve Martinez T. R., (1997). "Improved Heterogeneous Distance Functions". Journal of Artificial Intelligence, 6(1):1-34.

[33] Arai, K. ve Barakbah, A. R., (2007). "Hierarchical K-means: an algorithm for centroids initialization for K-means", Reports of the Faculty of Science and Engineering, 36(1):25-31.

[34] Huang, Z., (1998). "Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values", Data Mining and Knowledge Discovery, 2(3):283-304.

[35] Huang, Z., (1997). "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining", Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1-8.

[36] Ball, G. H. ve Hall, D. J., (1967). "A Clustering Technique for Summarizing Multivariate Data", System Research and Behavioral Science, 12(2):153-155.

[37] Ruspini, E. R., (1969). "A New Approach to Clustering", Information and Control, 15(1):22-32.

[38] Ruspini, E. R., (1973). "New Experimental Results in Fuzzy Clustering", Information Sciences, 6:273-284.

[39] Selim, S. Z. ve Ismail, M. A., (1984). "K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality", IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(1):81-87.

[40] Huang, Z., (1997). "Clustering large data sets with mixed numeric and categorical values", Proceedings of the First Pacific Asia Knowledge Discovery and Data Mining Conference, 1997, Singapur.

[41] Khan, S. S. ve Ahmad, A., (2004). "Cluster Center Initialization Algorithm for K-Means Clustering", Pattern Recognition Letters, 25(11):1293-1302

[42] Mohd, W. M. B. W., Beg, A.H., Herawan, T., Rabbi, K. F., (2012). "Max-D clustering K-means algorithm for Autogeneration of Centroids and Distance of Data Points Cluster", Communications in Computer and Information Science, 316:192-199.

[43] Kumar, A. ve Gupta, S. C., (2015). "A new Initial Centroid finding Method based on Dissimilarity Tree for K-means Algorithm", Electrical Engineering Department Indian Institute of Technology.

[44] Marriot, F. H. C., (1971), "Practical Problems in a method of Cluster Analysis", Biometrics, 27(3):501-514

[45] Ruiz, F. J. M., (1998). "Strategic Group Analysis in Strategic Marketing: An application to Spanish Saving Banks", Marketing intelligence & Planning, 16/4.

[46] Pham D. T., (2005). "Selection of K in K-means clustering", Journal of Mechanical Engineering Science, 219(1):103-119.

[47] Fahim A. M., Salem A. M., Torkey F. A. ve Ranadan M. A., (2006). “An Efficient Enhanced K-means Clustering Algorithm”, Journal of Zhejiang University, 10(7): 1626-1633.

[48] Mahmud S., Rahman M. ve Akhtar N., (2012). “Improvement of K-means Clustering Algorithm with Better Initial Centroids based on Weighted Average”, 7th International Conference on Electrical and Computer Engineering, Ekim 2012, Dhaka, 647-650.

[49] Khan S. S. ve Kant S., (2007). “Computation of Initial Modes for K-modes Clustering Algorithm using Evidence Accumulation”, Proceeding IJCAI'07 Proceedings of the 20th international joint conference on Artifical intelligence, 6-12 Ocak 2007, Hyderabad.

[50] Amorim R. C., (2015). “A Survey on Feature Weighting Based K-Means Algorithms”, Journal of Classification, 33(2):210-242.

[51] Sneath P. H. A., Sokal R. R. ve Freeman W. H., (1975). “Numerical Taxonomy: The Principles and Practice of Numerical Classication”, Systematic Zoology, 24(2):263-268.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Osman ÇÖREKCİ

Doğum Tarihi ve Yeri : 26.06.1989 / Bakırköy

Yabancı Dili : İngilizce

E-posta : osmancorekci.math@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Matematik	Bülent Ecevit Üniversitesi	2012
Lise	Fen Bilimleri	Hüseyin Bürge Lisesi	2006

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2014-…	Metro İstanbul A.Ş.	Software Developer
2013-2014	FutureCom	Software Developer
2013-2013	Signera	Stajyer .NET Developer

YAYINLARI

Makale

1. Corekci, O ve Sayli, A., (2016). “A New Clustering Algorithm of Hybrid Data to Weights of Attributes”, European Journal of Science and Technology, 9(5):28-37.

ÖDÜLLERİ

1. Matematik Bölümü Onur Belgesi (Bülent Ecevit Üniversitesi), 2012.