VERİ MADENCİLİĞİ YÖNTEMLERİ İLE SPAM FİLTRELEME

Serdar Kürşat SARIKOZ

YÜKSEK LİSANS TEZİ
BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

ARALIK 2010
ANKARA

Serdar Kürşat SARIKOZ tarafından hazırlanan VERİ MADENCİLİĞİ YÖNTEMLERİ İLE SPAM FİLTRELEME adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç. Dr. M.Ali AKCAYOL
Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Bilgisayar Bilimleri Anabilim Dalında Yüksek lisans tezi olarak kabul edilmiştir.

Başkan: : _____

Üye : _____

Üye : _____

Üye : _____

Üye : _____

Tarih : ...../……/……

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

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

Serdar Kürşat SARIKOZ

VERİ MADENCİLİĞİ YÖNTEMLERİ İLE SPAM FİLTRELEME

(Yüksek Lisans Tezi)

Serdar Kürşat SARIKOZ

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Aralık 2010

ÖZET

Ticaretin internet kanalları üzerinden gelişmesi, hızlı ve ekonomik haberleşme olması nedeni ile elektronik posta haberleşmesinin hayatımızda giderek önemini artırmıştır. İşlem maliyetinin çok düşük olması, çok büyük miktardaki verilerin çok uzak mesafelere saniyeler içinde aktarılmasına olanak sağlaması yaygınlaşmasını sağlamıştır. İnternet üzerinde aynı mesajın yüksek sayıdaki kopyasının, bu tip bir mesajı alma talebinde bulunmamış kişilere, zorlayıcı nitelikte gönderilmesi spam olarak adlandırılır. E-posta yolu ile gönderilen spam türlerinden ticari içerikli olan UCE (Unsolicited Commercial E-mail) ve UBE (Unsolicited Bulk E-mail) adından da anlaşılacağı gibi istenmediği halde size gönderilen bir ürünü ya da hizmeti tanıtıcı elektronik posta iletileridir. İstenmeyen elektronik posta problemini tamamen çözebilmiş tek bir teknik ya da tekniklerin birleşmesinden oluşan bir çözüm mevcut değildir. İstenmeyen iletilerin belirlenmesine yönelik birçok veri madenciliği çalışması da yapılmıştır. Veri madenciliği açıkça verinin bir parçası olmayan veride ilginç örüntüleri bulma sürecine denir. Spam filtrelemede iki tür yaklaşım söz konusudur. Bunlardan birincisi bilgi mühendisliği (knowledge engineering) yöntemi ile kurallar oluşturarak filtreleme yapmaktır. Diğeri ise makine öğrenimi ya da

makine öğrenimi tekniklerini büyük veri setleri üzerinde uygulayarak makine öğreniminden ayrılan veri madenciliği olarak bilinen yöntemler ile önceden hazırlanmış veri setleri ile sınıflandırmanın yapılmasıdır.

Bu tez kapsamında e-posta veri setleri üzerinden oluşturulmuş olan nitelik uzayı üzerinde veri madenciliği yöntemleri uygulanarak spam filtreleme yapılmıştır.

Bilim Kodu : **902**
Anahtar Kelime : **spam filtreleme, veri madenciliği, kümeleme analizi, yapay sinir ağları**
Sayfa Adedi : **132**
Tez Yöneticisi : **Doç. Dr. M. Ali AKCAYOL**

SPAM FILTERING USING DATA MINING METHODS

(M.Sc. Thesis)

Serdar Kürşat SARIKOZ

GAZİ UNIVERSITY
INFORMATICS INSTITUTE
December 2010

ABSTRACT

The importance of e-mail communication in our lives has continually increased since the commerce is developed over internet channels, and there is fast and economic communication. Very low operation cost provides transferring a large number of data within a few seconds over long distances.

Sending a large number of copies of the same message stringently to the people who are not willing to receive over the internet is called spam.

UCE (Unsolicited Commercial e-mail) and UBE (Unsolicited Bulk e-mail) which are kinds of spam messages sent via e-mail, as it can be inferred from the names, are introductory e-mails which is actually undesirable.

There is not an available unique technique or an available solution combined by the techniques in which the problem of undesirable e-mail is solved. There have been lots of data mining approaches aimed at determining unsolicited e-mails.

Data mining is the process of finding the interesting patterns which are obviously not part of the data. In spam filtering, there are two kinds of approaches. One is filtering by constructing the rules by knowledge engineering.

Second is classification within datasets prearranged via the techniques known as data mining separated from machine learning by applying machine learning techniques over very large datasets.

Within the scope of this thesis, spam filtering has been implemented by applying data mining techniques over attribute space model formed on the basis of e-mail datasets.

Science Code : **902**
Key Words : **spam filtering, data mining, clustering analysis, artificial neural networks**
Page Number : **132**
Adviser : **Assoc. Prof. Dr. M. Ali AKCAYOL**

TEŞEKKÜR

Tez çalışmam süresince bana rehberlik edip yol gösteren, çalışmama şekil veren tez danışmanım Doç. Dr. M. Ali AKCAYOL'a, çalışmalarımda emek ve vakit harcayarak yardımlarını esirgemeyen bilgisayar mühendisi Abdullah ÖNER'e ve Muhammed ŞEN'e, her zaman yanımda olarak desteklerini esirgemeyen eşime, kardeşime ve aileme teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

SİMGELER VE KISALTMALAR

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

Kısaltmalar	Açıklama
ODBC	Open Database Connectivity – Açık Veritabanı Bağlantısı
OLAM	Çevrim İçi Analitik Madencilik
OLAP	Çevrim İçi Analitik İşleme
OLEDB	Object Linking & Embedding Databases
SKICAT	Sky Image Classification & Archiving
VM	Veri Madenciliği
DM	Data Mining – Veri Madenciliği
VTBK	Veri Tabanlarında Bilgi Keşfi
VTYS	Veri Tabanı Yönetim Sistemleri
MÖ	Makine Öğrenimi
VAYS	Veri Ambarı Yönetim Sistemleri
SQL	Structured Query Language
EM	Expectation Maximization
AGNES	Agglomerative Nesting
BIRCH	Balanced Iterative Reducing and Clustering Using Hierarchies
CHAMELEON	Hierarchical Clustering Using Dynamic Modeling
CLARA	Clustering Large Applications
CLARANS	Clustering Large Applications based upon Randomized Search
CLIQUE	Clustering High-Dimensional Space
COICOP	Classification of Individual Consumption by Purpose
CURE	Clustering Using Representatives
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DENCLUE	Density-Based Clustering
DIANA	Divisive Analysis

Kısaltmalar	Açıklama
KDD	Knowledge Discovery in Databases
OPTICS	Ordering Points To Identify the Clustering Structure
MKS	Machine Learning Kernel System
YSA	Yapay Sinir Ağları
BP	Back Propagation
DNS	Domain Name Service
VSM	Vector Space Model
IR	Information Retrieval
AI	Artificial Intelligence
KDD	Knowledge Discovery in Databases
KP	Koşut Programlama
DDM	Dağıtık Veri Madenciliği
CDM	Collective Data Mining
ISP	İnternet Service Provider
Email	Electronic Mail
E-posta	Elektronik Posta
DoS	Denial of Service
OCR	Optic Character Recognation
GA	Genetik Algoritma
UBE	Unsolicited Bulk Email
UCE	Unsolicited Commercial Email
SPAM	İstenmeyen E-posta

1. GİRİŞ

İnternetin küreselleşmeyi körüklemesi, ticaretin internet kanalları üzerinden gelişmesi, hızlı ve ekonomik haberleşme olması nedeni ile elektronik posta haberleşmesinin hayatımızda giderek önemini artırmıştır. İşlem maliyetinin çok düşük olması, çok büyük miktardaki verilerin çok uzak mesafelere saniyeler içinde aktarılmasına olanak sağlaması yaygınlaşmasını sağlamıştır.

Bir taraftan internet üzerinden elektronik posta ile ürünlerinin (çoğunlukla güvenilmeyen, hileli ürünler) ticari reklâmını yapmak veya yasal olmayan duyurularda bulunmak isteyenler, diğer taraftan da elektronik posta adres listelerinin sayısının artması "istenmeyen elektronik posta" (spam) kavramının öneminin artmasına neden olmuştur. Bugün çok ileri boyutlara ulaşan ve kullanıcılar açısından değerlendirildiğinde zaman kaybı, işletmeler açısından değerlendirildiğinde ise çalışanların spam e-postaları ayıklamak için ayırdıkları zamandan dolayı verimlilik kaybına neden olduğu görülmektedir. İstenmeyen elektronik posta probleminin önüne geçebilmek amacıyla bu tür elektronik postaları otomatik olarak filtreleyebilen metotlar bir gereklilik haline gelmiştir.

Bununla birlikte e-posta haberleşmesinin avantajları büyük sayıda potansiyel müşterilere ulaşmak isteyen firmalar içinde son derece önem arz etmektedir. Bu nedenle bu tür firmalar ürünleri daha geniş kitlelere tanıtmak ve satmak amacı ile e-posta ile müşteriye ulaşmayı tercih etmektedirler.

Spam e-posta göndericilerinin, spam e-postaları cinsel içerikli görüntüler ile hedef kitlere ulaştırmak istemesi, bu görüntüler ile bilgi hırsızlığı gibi yöntemleride kullanmaları olayı daha da tehlikeli ve rahatsız edici hale getirmektedir.

Spam e-posta gönderiminde bulunan kişiler; ulaştıkları hedef miktarı, topladıkları e-posta adres sayısı, göndermiş oldukları maillerden dolayı yapılan işlem hacmi arttıkça kendi gelir kaynaklarında da bir artış söz konusu olmaktadır.

Spam e-posta gönderiminde bulunan kişilerin kullandığı teknikler ile spam e-postaları önlemede kullanılan yöntemler sürekli gelişim halindedir. Spam e-posta

gönderimini sağlayan kişilerin spam gönderimi nedeni ile yüksek kazançlarından dolayı spam e-posta'lardan kurtulmak yakın zamanda mümkün gözükmemektedir.

İstenmeyen elektronik posta problemini tamamen çözebilmiş tek bir teknik ya da tekniklerin birleşmesinden oluşan bir çözüm mevcut değildir. İstenmeyen iletilerin belirlenmesine yönelik birçok veri madenciliği çalışması da yapılmıştır.

Veri madenciliği (VM) büyük veri tabanlarından önceden öngörülmeyen bilgiyi çıkarmanın ve sonuçları karar vermeye uygulamanın çok aşamalı sürecidir. Veri madenciliği araçları veriden örüntüleri algılar ve onlardan ilişkiler ve kurallar çıkarır. Çıkarılmış bilgi tahmin ve sınıflandırma modellerinde kullanılmak üzere veri üzerindeki ilişkiler tanımlanarak uygulanabilir. Bu örüntüler ve kurallar karar vermeye rehberlik edebilir.

Bugün veri madenciliği; veri tabanı teknolojisi, yapay zeka, yapay sinir ağları, istatistik, örüntü algılama, bilgi kazanımı, yüksek performans hesaplama ve veri görselleştirme gibi adımlar ile iç içe geçmiş çok disiplinli bir alandır [4].

Spam e-postaların veri madenciliği yöntemleri tespiti ve filtrelenmesine yönelik çalışmalar veri madenciliği ve spam ile mücadeleye yönelik düzenlenen Data Mining Cup, CEAS gibi uluslararası çalışmaların temel konularından birisi haline gelmiştir. Söz konusu çalışmalarda; karar ağaçları, kümeleme metodları, bayesian analizi, yapay sinir ağları gibi birçok yöntem spam tespiti ve spam filtrelemede uygulanmıştır.

Bu çalışma kapsamında danışmansız öğrenme tekniklerinden olan k-means kümeleme metodu ile danışmanlı öğrenme metodu olan yapay sinir ağlarına yer verilmiştir.

Bu tez altı bölümden oluşmaktadır. İkinci bölümde veri madenciliği ve yöntemlerine, üçüncü bölümde spam ve detaylarına, dördüncü bölümde spam filtrelemede kullanılan yapay sinir ağları ve kümeleme analizi yöntemlerine, beşinci bölümde spam e-posta filtreleme için geliştirilmiş olan çalışmanın detaylarına, altıncı bölümde sonuca yer verilmiştir.

2. VERİ MADENCİLİĞİ

Bu bölümde veri madenciliğinden, veri tabanlarında bilgi keşfinden, kısaca veri madenciliği algoritmalarından, veri madenciliği çalışmalarında çoğunlukla yer alan ön veri işlemeden ve veri madenciliğinde karşılaşılan sorunlardan bahsedilecektir. En son olarak veri madenciliğinde kullanılan araçlara değinilecektir.

2.1. Veri Madenciliğine Giriş

Bilgisayar sistemleri her geçen gün ucuzlamakta ve güçleri artmaktadır. İşlemciler gittikçe hızlanmakta, disklerin kapasiteleri ise baş döndürücü bir hızda artmaktadır. Artık bilgisayarlar daha büyük miktardaki veriyi saklayabilir ve daha kısa sürede işleyebilir hale gelmiştir [1].

Bilgisayar ağlarındaki ve erişim teknolojilerindeki ilerleme ile veriye hızla ulaşabilmek mümkün hale gelmiştir. Bu gelişmeler ile birlikte her türlü verinin saklanmasına, değerlendirilmesine olan gereksinimi artırmıştır.

Bilgi teknolojilerinin ticaret, tıp, askeri, iletişim, vb. birçok alanda kullanılması ile birlikte veri hacminin yaklaşık olarak her yirmi ayda iki katına çıktığı tahmin edilmektedir [2].

Verilerin ne kadar hızlı toplandığını ve işlemesinin imkânsız bir noktaya geldiğini en belirgin bir şekilde NASA'da görülmektedir. NASA'nın kullandığı uyduların sadece birinden, bir günde terabayt'larca veri gelmektedir [3].

Veri kendi başına değersizdir. İstediğimiz, amacımız doğrultusunda bilgidir. Bilgi bir amaca yönelik işlenmiş veridir. Veriyi bilgiye çevirmeye veri analizi veya bilgi keşfi (BK) denir. Bu tanımda keşif sözcüğünün kullanılmasının amacı, gizli olan ve daha önceden bilinmeyen örüntülerin bulunmasından kaynaklanmaktadır. Bilgi, bir soruya yanıt vermek için veriden çıkardığımız anlam olarak da tanımlanabilir. Veri sadece sayılar veya harfler değildir; veri, sayı ve harfler ve onların anlamıdır.

Veri hacminin hangi boyutlara ulaşabileceği ve bunların işlenmesinin ne kadar güç olduğu kolayca anlaşılabilmektedir. Süper market örneği incelendiğinde, veri analizi yaparak her mal için bir sonraki ayın satış tahminleri çıkarılabilir; müşteriler satın aldıkları mallara bağlı olarak gruplanabilir; yeni bir ürün için potansiyel müşteriler belirlenebilir; müşterilerin zaman içindeki hareketleri incelenerek onların davranışları ile ilgili tahminler yapılabilir. Binlerce malın ve müşterinin olabileceği düşünülürse bu analizin gözle ve elle yapılamayacağı, otomatik olarak yapılmasının gerektiği ortaya çıkar. Veri madenciliği de burada devreye girmektedir.

2.2. Veri Madenciliğinin Tarihçesi

VM yaklaşımı ortaya çıkmadan önce, büyük veri tabanlarından faydalı örüntüler elde etmek için, çevrim-dışı veri üzerinde çalışan istatistiksel paketler kullanılırdı. İstatistiksel yaklaşımların kullanımında bu paketlerin dezavantajları ortaya çıkmaktaydı. Bu dezavantajlardan en önemlisi; istenen verilerin toplanmasından ve amacın belirlenerek istatistiksel yaklaşımların uygulanmasından sonra bir uzman tarafından değerlendirilmesi gerekliliği idi. Başka bir dezavantajı ise her farklı ihtiyaç için bu işlemlerin tekrarlanması gerekmekte idi.

Veri madenciliği tekniklerinin yapı taşları 1950'lere, matematikçilerin, mantıkçıların ve bilgisayar bilimcilerin işlerini yapay zekâyı ve makine öğrenimini yaratmak için birleştirdikleri zamana dayanmaktadır.

1960'larda, yapay zekâ ve istatistik uzmanları, olasılık analizi, yapay sinir ağları ve doğrusal sınıflandırma modelleri gibi yeni algoritmalar geliştirmişlerdir. Veri madenciliği terimi bu on yıllık süreçte bulunmuş olmasına rağmen istatiksel önemi bulunmayan örüntüleri bulmada kullanılmıştır.

Aynı zamanda 1960'larda, bilgi kazanımı (IR) alanı kümeleme tekniklerinde ve benzerlik ölçümlerinin gelişimine katkıda bulunmuştur. Metin belgelerine bu teknikler uygulanmış, fakat sonrasında veritabanlarında ve büyük ölçekli dağıtık veri kümelerinde veri madenciliği yapılırken kullanılmıştır. 1960'ların sonunda, bilgi kazanımı ve veri tabanı sistemleri paralel bir şekilde gelişmiştir.

1971'de, Gerard Salton akıllı bilgi kazanımı hakkındaki çığır açan yazısını yayınlamıştır. Bu modelde cebirsel bazlı vektör uzay modelini (vector space model - VSM) kullanan bilgi kazanımı yaklaşımı sunulmuştur. VSM veri madenciliği araçlarında anahtar bir yöntem olacağını göstermiştir.

1970, 1980 ve 1990'lı yıllar boyunca, yapay zekâ, bilgi kazanımı, istatistik ve veri tabanı sistemlerinin birbirine yakınsaması ya da benzer sorunları birlikte ele alması, bilgisayar teknolojisindeki gelişme ile birlikte veriyi elde etme ve analizi için imkânlar artmıştır. Bunun sonucu olarak genetik algoritmalar, kümeleme algoritmaları, karar ağaçları gibi yeni yöntemler ve programlama dilleri gelişme göstermiştir.

1990'ların başlangıcında, Veri Tabanlarında Bilgi Keşfi (VTBK) terimi kullanılmaya başlanılmış ve KDD (Knowledge Discovery in Databases) yarışmaları düzenlenmiştir.

1990'lar operasyonel ve hareketsel veri tabanı verisinden yaratılan veri tabanı ambarlarının gelişimini görmüştür, büyük veritabanlarını tanımlamada kullanılan bir terim (tek bir şemadan oluşan). Veri ambarlarının gelişimi boyunca çevrimiçi analitik işleme (OLAP), karar destek sistemleri, veri değiştirme ve birleşim kural algoritmaları gelişmiştir.

1990'lar boyunca veri madenciliği üzerinde araştırmalar yapılan ilginç bir teknolojiden pratikte yer almaya başlamıştır. Yeni müşterileri elde etmeyi, var olan müşterilerden geliri artırmayı ve iyi müşterileri elinden kaçırmamayı içeren müşteri hayat döngüsünün her bir evresini yönetmede yardım etmek için veri madenciliği kullanılmaya başlamıştır.

Mayıs 2004'te Amerika Birleşik Devletleri hazinesi tarafından hazırlanmış olan rapora göre, NSA'in istihbari çalışmaları dışında yapım aşamasında ya da planlanan 199 veri madenciliği operasyonu olduğu belirtilmiştir [4].

2.3. Veri Madenciliğine Genel Bakış

Veri madenciliği açıkça verinin bir parçası olmayan veride ilginç örüntüleri bulma sürecine denir. Bu İlginç örüntüler bize yeni bir şey söylemek ve tahminler yapmak için kullanılabilir. Veri madenciliği süreci analiz için veriyi seçme, veriyi hazırlama, veri madencilik algoritmalarını uygulama, sonrasında çevirim ve sonuçları değerlendirmeyi içeren birçok adımdan oluşmaktadır. Bazen veri madenciliği terimi veri madenciliği algoritmalarının uygulandığı adımlar olarak ta belirtilmektedir. Bu literatürde karışıklık yaratmıştır. Fakat daha sık olarak veri madenciliği terimi verideki ilginç örüntüleri bulmada ve kullanmak olarak belirtilmiştir.

Veri madenciliği tekniklerin uygulaması ilk olarak veri tabanlarına uygulanmıştır. Bu süreci Knowledge Discovery in Databases (KDD) - Veri Tabanlarında Bilgi Keşfi (VTBK) daha iyi ifade etmektedir.

Veri madenciliği (VM) büyük veri tabanlarından önceden öngörülmeyen bilgiyi çıkarmanın ve sonuçları karar vermeye uygulamanın çok aşamalı sürecidir. Veri madenciliği araçları veriden örüntüleri algılar ve onlardan ilişkiler ve kurallar çıkarır. Çıkarılmış bilgi tahmin ve sınıflandırma modellerinde kullanılmak üzere veri üzerindeki ilişkiler tanımlanarak uygulanabilir. Bu örüntüler ve kurallar karar vermeye rehberlik edebilir.

Bugün veri madenciliği; veri tabanı teknolojisi, yapay zeka, yapay sinir ağları, istatistik, örüntü algılama, bilgi kazanımı, yüksek performans hesaplama ve veri görselleştirme gibi adımlar ile iç içe geçmiş çok disiplinli bir alandır [4].

Geleceğin, en azından yakın geleceğin, geçmişten çok fazla farklı olmayacağını varsayarsak geçmiş veriden çıkarılmış olan kurallar gelecekte de geçerli olacak ve ilerisi için doğru tahmin yapmamızı sağlayacaktır.

Büyük miktarlarda verinin, veri tabanlarındatutulduğu bilindiğine göre bu verilerin VM teknikleriyle işlenmesine de veri tabanlarında bilgi keşfi denir (VTBK). Büyük hacimli olan ve genelde veri ambarlarında tutulan verilerin işlenmesi yeni kuşak araç ve tekniklerle mümkün olabilmektedir. Bundan dolayı bu konularda yapılan

çalışmalar güncelliğini korumaktadır. Bazı kaynaklara göre; VTBK daha geniş bir disiplin olarak görülmektedir. Veri seçimi, veri temizleme, veri ön işleme, veri indirgeme, veri madenciliği algoritmasının uygulanması ve sonuçların değerlendirmesi VTBK'yi oluşturan basamaklardır. Kısaca büyük ölçekli veri tabanlarından anlamlı ve gizli örüntülerin çıkarılması olarak anılan Veri Madenciliği (VM), VTBK içinde bir adım olarak kabul edilir [5].

2.4. Veri Tabanlarında Bilgi Keşfi ve Aşamaları

Varolan verilerden bilgiyi yani kullanılabilecek örüntüleri elde etmeye geniş çapta ihtiyaç duyulmuştur. Bu ihtiyacı gidermek için araştırma kurumları ve üniversiteler çalışmalarıyla yeni disiplinler ortaya çıkarmıştır. Veri madenciliği bu yeni disiplinlerden biridir. Veri madenciliğinin veri tabanları üzerine uygulanmasıyla Veri Tabanlarında Bilgi Keşfi (VTBK) ortaya çıkmıştır.

Veri Tabanlarında Bilgi Keşfi (VTBK) genelde çok büyük hacimli verileri ele almakta kullanılmaktadır. Verileri ya tam ya da yarı otomatik olarak analiz eden yeni sistemlerle, bu disiplin son zamanlardaki en güncel konulardan biri haline gelmiştir. Veri seçimi, veri temizleme, veri ön işleme, veri indirgeme, veri madenciliği algoritmasının uygulanması ve sonuçların değerlendirmesi VTBK'yi oluşturan basamaklardır.

VTBK sürecinde yer alan adımlar şu şekilde sıralanmaktadır [4]:

2.4.1. Görevin belirlenmesi

Veri madenciliği operasyonunun amaçları, süreç başlamadan önce iyi belirlenmelidir. Analist çözülecek sorunun ne olduğunu bilmelidir. Görev keşfinin ilk adımında, konunun uzmanı, analistle ortak çalışma içine girmelidir.

2.4.2. Verinin belirlenmesi

Bu adımda, analist ve son kullanıcı sorularını cevaplamak istedikleri soru için; hangi veriyi analiz edeceklerini belirlemelidir. Sonrasında ihtiyaç duydukları uygun veriyi belirlerler.

2.4.3. Veri seçimi ve temizlenmesi

Veri seçildikten sonra temizlenmelidir. Eksik değerlerin düzeltilmesi; tamamlanmamış kayıtların elenmesi, elle doldurulmaları veya her bir eksik değer için sabit bir değer girilmesi ya da bir değer tahmini yapılması ile çözümlenmelidir. Hatalı veriler çözümün kalitesini etkileyeceği için hatalı veriler tutarlı bir şekilde ele alınmalıdır.

2.4.4. Veri dönüşümü

Veriler dönüştürülerek veri madenciliği için uygun forma getirilmelidir. Veri madenciliği metotlarını kullanmak için, yüksek yapısal veri formatı ve geniş veri hazırlıkları gereklidir.

Veri dönüşüm süreci; düzleme (veri hatalarını gidermek için kova metodu kullanımı), toplama (örneğin, günlük yerine aylık verinin ele alınması), genelleme (örneğin insanları tam kendi yaşları yerine; genç, orta yaş ya da yaşlı olarak tanımlama), normalizasyon (veriyi sabit bir değerde ölçeklendirme) ve nitelik yapılandırılmasını (yeni niteliklerin eklenmesi) kapsamaktadır.

2.4.5. Verilerin azaltılması

Süreci daha ucuza mal etmek ve daha kolay yönetebilmek için verilerin azaltılması gerekebilir. Veri azaltımı teknikleri, veri küpleri yığma, boyut azaltılması (konu dışı ve gereksiz niteliklerin azaltılması), veri sıkıştırması (boyut küçültmek için veri şifrelenmesi, asıl veriler yerine modeller ve örneklerin kullanılması) ve kesikli modele çevirme (discretization) yöntemini içerir.

2.4.6. Örüntülerin bulunması

Aynı zamanda veri madenciliği olarak bilinen bu adımda, veri; ilginç veya kullanışlı modeller veya ilişkileri bulmak için, yinelemeli olarak veri madenciliği algoritmalarının içinden geçer. Sıklıkla sınıflandırma ve kümeleme algoritmaları birleşik kurallar uygulanabilsin diye ilk olarak kullanılmaktadır.

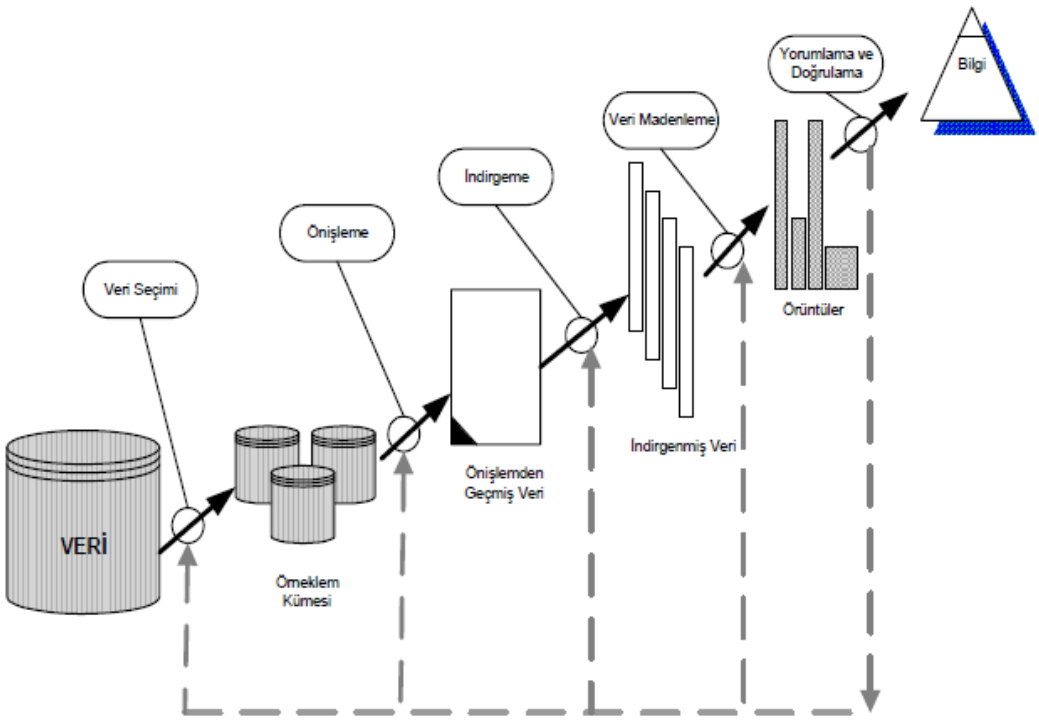
Bazı kurallar diğerlerinden daha ilginç olan sonuçlar verebilir. İlginçlik özel algoritmanın efektifliğini belirlemek için kullanılan ölçümlerden biridir. İlginçliğin genellikle model değerlerinin tüm ölçümleri olarak alındığı, geçerliliklerin birleştirildiği, yenilik, kullanışlılık ve basitlik olarak ta kabul görmektedir. Bir model; eğer ilginçlik üst limitini aşarsa, bilgi olarak düşünülebilir. Üst limitin kullanıcı tarafından belirtilmesi konuya göre değişkenlik göstermektedir. Bu da kullanılacak fonksiyonlarında değişkenlik göstermesine neden olacaktır [5].

2.4.7. Sonuçların yorumlanması ve görselleştirme

Veri madenciliği adımından elde edilen sonuçlar, bu sonuçları kullanacak kullanıcılar tarafından yorumlaması için grafiksel araçlar ile görsel olarak sunulur.

2.4.8. Bilginin kullanıma sunulması

Son olarak, en son kullanıcı sonucun kullanımını yapmak zorundadır. Orijinal problemi çözmenin yanında, yeni bilgi yeni modellerin içine birleştirilebilir ve tüm bilgi ve veri madenciliği döngüsü tekrardan başlayabilir. Bu basamaklar Şekil 2.1.'de ifade edilmiştir.



Şekil 2.1. VTBK sürecinde yer alan basamaklar.

2.5. Veri Tabanlarında Bilgi Keşfi ve Diğer Disiplinler Arası İlişki

VTBK; veri madenciliği, makine öğrenimi (MÖ), veri tabanı yönetim sistemi (VTYS), veri ambarları (VA), koşut programlama (KP) ve istatistik gibi birçok farklı disiplinin kullandığı teknikleri kullanmaktadır. Bundan dolayı çok disiplinli bir yaklaşımdır.

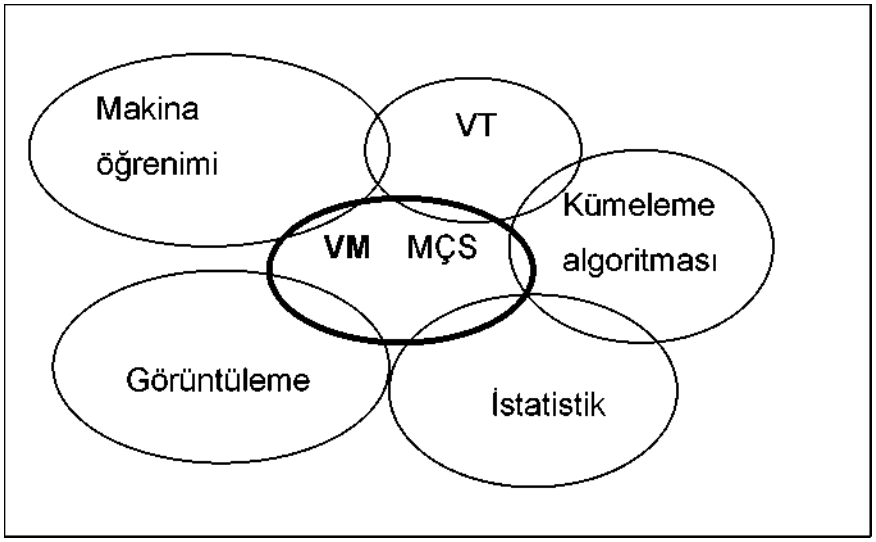
VM, MÖ ve istatistik birbirlerine yakın disiplinlerdir. Bu üç disiplinin ortak noktaları veri içindeki ilginç örüntüleri bulmayı amaçlamalarıdır. VM algoritmalarının çekirdeğini MÖ'de kullanılan algoritmalar oluşturur. Makine öğreniminde kullanılan sınıflama, kümeleme ve ilişkilendirme algoritmaları gibi birçok algoritma veri madenciliğinde kullanılmaktadır. MÖ ile VM arasında bu söylenen benzerliklerin bulunmasının yanı sıra aralarında çok büyük farklar da vardır. Örneğin, MÖ küçük hacimli ve genelde deneysel verilerle uğraşırken, VM büyük hacimli gerçek dünya verileriyle uğraşır. MÖ'nün örnekleme kümesi genelde 100-1000 arasındayken, VM uygulandığında milyonlarca veriden söz edilmektedir. VM ve MÖ arasındaki diğer

bir fark da, MÖ'nün aksine VM'nin gürültülü, eksik, artık ve boş (NULL) değerleri işleyebilmesidir.

VM, istatistik alanındaki birçok metodu kullanmasına rağmen, nesnelerin nitelik değerlerine bağlı çıkarsama yapmada bilinen istatistiksel metotlardan ayrılmaktadır. Örneğin ki-kare veya t testi gibi istatistiksel test yöntemleri, birden fazla nitelik arasında ilişki derecesini belirli bir güvenilirlik düzeyinde verebilirken, belirli nitelik değerleri arasındaki ilişkinin derecesini açığa çıkaramazlar.

İstatistiksel yöntemler karar verme mekanizmasında VM ortaya çıkmadan önce çok sık kullanılırdı. Ancak bu yöntemlerin kullanım zorluğu (uzman kişileri tutma/ uzman kişilere başvurma), VM algoritmalarının uygulama zorluğu yanında çok fazladır.

VTBK farklı disiplinleri bir araya getiren bir sistemdir. VM çekirdek sistemi (MKS-*The Mining Kernel System*) Şekil 2.2'de gösterilmiştir.



Şekil 2.2. VM MKS gösterimi.

2.5.1. VTBK ile makine öğrenimi arasındaki ilişki

Makine öğrenimi gözlem ve deneye dayalı kuralların otomatik biçimde bulunması olan VTBK sistemleri ile yakından ilgilidir. Genel olarak makine öğrenimi ve örüntü

tanıma alanlarında yapılan çalışmaların sonuçları VTBK'de veri modelleme ve örüntü çıkarmak için kullanılmaktadır. Bu çalışmalardan bazıları:

Örneklerden öğrenme, düzenli örüntülerin keşfi, gürültülü ve eksik veri ile eksik belirsizlik yönetimi olarak sayılabilir.

VTBK'nın makine öğreniminden en büyük farkı aşağıda sıralanmıştır:

• VTBK büyük veri kümeleriyle çalışabilir,

• VTBK gerçek dünya verileriyle uğraşır.

Veri görselleştirmede kullanılan yöntemler, VTBK sistemi ile elde edilen örüntülerin, kullanıcıya grafikler aracılığı ile sunumunu sağlamaktadır.

2.5.2. VTBK ile istatistik arasındaki ilişki

İstatistik ile VTBK arasındaki ilişkinin ana sebebi veri modelleme ve verideki gürültüyü azaltmadan kaynaklanmaktadır. İstatistiğin VTBK'de kullanılan tekniklerinden bazıları aşağıda sıralanmıştır:

- Özellik seçimi [7],
- Veri bağımlılığı [8],
- Sürekli değerlerin ayrımı [8],
- Tanıma dayalı nesnelerin sınıflandırılması [9],
- Veri özeti [10],
- Eksik değerlerin tahmini [11].

2.5.3. VTBK ile veri tabanı arasındaki ilişki

VM sorgularına girdi sağlamak amacıyla VT kullanılmaktadır. VT'deki sorgu cümlecikleri VM'nin istediği örneklem kümesini elde etmek amacıyla kullanılmaktadır. Özellikle ilişkilendirme sorgusunda fazla miktarda VT sorgusu yapmak gerekmektedir.

VM, VT'den farklıdır, çünkü VT'de var olan örüntüler için sorgular çalıştırılırken, VM'deki sorgular genelde keşfe dayalı ve ortada olmayan örüntüleri keşfetmeye dayalıdır.

2.6. Veri Madenciliği Yöntemleri

Veri madenciliği yöntemleri uygulandıkları probleme ve verinin yapısına göre farklılık arz edebilir [37]. Genel veri madenciliği metotları; sınıflandırma, regresyon analizi, kümeleme, özetleme, ilişkilendirme, değişme ve sapmaların belirlenmesi olarak sıralanmaktadır.

2.6.1. Sınıflandırma

Sınıflandırma 2 adımdan oluşur: örnek veri seti kullanarak veri modelinin öğrenilmesi (danışmanlı öğrenme) ve sonrasında modellere göre verileri sınıflandırmadır. Bazı çok bilinen sınıflandırma algoritmaları; Bayesian sınıflandırma, karar ağaçları, yapay sinir ağları, k-en yakın komşu sınıflandırıcıları ve genetik algoritmaları içerir.

Karar ağaçları, bütün küme analiz edilene kadar verileri yaprak ve düğüm bölümlerine ayıran, sınıflandırmayı yukarıdan aşağıya doğru gerçekleştiren meşhur yaklaşımdır.

YSA, hazırlanmış bir veri kümesinden (öğrenme kümesi) öğrenen, doğrusal olmayan önceden tahmin edici araçlardır ve öğrenme kümesi sonucu verinin yapısını öğrendiği düşünülerek daha büyük kümelere uygulanır.

Genetik algoritmalar yapay sinir ağları gibidir. Yapay sinir ağlarından farklı olarak doğal seçim ve mutasyon özellikleri bulunmaktadır.

En yakın komşu yöntemi bir öğrenme eğitim seti aracılığı ile bir grubun benzerliğini ölçmektedir. Daha sonra elde ettiği sonuçları test verilerini analiz etmede kullanılmaktadır.

2.6.2. Regresyon analizi

Regresyon analizi; var olan bilgiler üzerinde formüller uygulayarak tahminler yapabilmek için kullanılır. Var olan verilerden doğrusal ve lojistik regresyon tekniklerini (istatistikler yardımı ile) kullanarak bir fonksiyon öğrenilebilir. Yeni veriler; tahminler yapabilmek için fonksiyonlara haritalanır. Yapraklardaki ortalama değerlere sahip karar ağaçları olan regresyon ağaçları, genel bir regresyon tekniğidir.

2.6.3. Kümeleme

Kümeleme; verileri tanımlamak için sonlu küme setini tanımlamayı içerir. Kümeler karşılıklı özel, hiyerarşik ve de üst üste gelme olabilir. Kümenin her bir üyesi kümenin içindeki diğer üyelere benzer olmalı ve diğer kümelerdeki üyelere göre farklı olmalıdır. Kümeleme yöntemleri arasında: verinin sıklıkla "k-means algoritması" ile parçalara ayrılması (partitioning), hiyerarşik tabanlı yöntemler (grup üyelerinin ağaçlara yerleştirilmesi), yoğunluk tabanlı yöntemleri ve grid yöntemlerini sayabiliriz.

Aykırı değer analizi; diğer kümelere düzgün bir şekilde uymayan maddelere odaklanan küme analizinin bir formudur. Bazen bu nesneler verilerdeki hataları temsil eder, bazen de ilginç modelleri ortaya çıkarabilir.

2.6.4. Özetleme

Özetleme altkümelerin içine verileri haritalar ve sonrasında alt kümeler için efektif bir tanım uygular. Aynı zamanda karakterleme ya da genelleme olarak tanımlanan şey verinin içinden özet çıkartır veya içeriği özlü bir şekilde karakterize eden gerçek veri parçalarını çıkarır.

2.6.5. İlişkilendirme kural madenciliği

İlişkilendirme kural madenciliği, veri kümesindeki ilginç ilişkilerin araştırılmasını içerir. Market sepet analizi bu modelin güzel bir örneğidir. Birleşik kuralın bir örneği bilgisayar alan müşterilerin aynı zamanda yazılımı da alması beklenir. İlişkilendirme kuralları her zaman ilginç veya kullanışlı olmadığı için; kısıtlar, özelleştirilen

bilgileri, istatistiksel ölçümlerin üst limitleri (kural ilginçliliği, destek ve güven) son kullanıcılar tarafından bilginin tipine uygun olarak uygulanır.

Yukarıda belirtilen metotlar birçok veri madenciliği uygulamasının temelini oluşturmaktadır. Yukarıda belirtilen temel yaklaşımları; algoritmalarda değişiklikler yapılarak uzaysal verilere, çok boyutlu veritabanlarına ve web'e uygulandığına ilişkin birçok çalışmaya literatürde rastlanmaktadır.

2.7. Veri Madenciliği Algoritmaları

Veri madenciliği algoritmaları verilerde var olan bilgiyi anlaşılabilecek kurallar olarak çıkartmaya yarayan metotlardır. Veri madenciliği algoritmaları genel olarak iki ana gruba ayrılır [12]:

Doğrulamaya dayalı algoritmalar: Kullanıcı tarafından ispatlanmak istenen bir hipotez ortaya sürülür ve VM algoritmalarıyla bu hipotez ispatlanmaya çalışılır. Çok boyutlu analizlerde ve istatistiksel analizlerde tercih edilen metottur. Hipotez testi buna örnektir.

Keşfe dayalı algoritmalar: Doğrulamaya dayalı algoritmaların tersine bu algoritmalarda ortada ispatlanması istenen hipotezler yoktur. Tam tersine bu algoritmalar otomatik keşfe dayanmaktadır. Keşfe dayalı algoritmaların birçok kullanım alanı vardır: istisnai durumların keşfi, karar ağacı, kümeleme gibi algoritmalar bu yaklaşıma göre kurulmuştur.

Veri madenciliği algoritmaları büyük hacimli veriler üzerinde çalışabilecek ve istenen bilgileri sağlayacak algoritmalardır. VM ile çıkarılan başlıca bilgi türleri aşağıdaki başlıklar altında sıralanabilir:

2.7.1. Hipotez testi

Hipotez testi algoritmaları doğrulamaya dayalı algoritmalardır. Doğrulanacak hipotez VT üzerindeki verilerle belli doğruluk ve destek değerlerine göre sınanır. Sınama işlemi uzman tarafından aşağıdaki ihtiyaçlardan dolayı yapılır:

- Bir kural ortaya çıkarılmak istendiğinde,
- Ortaya çıkarılmış bir kuralın budanması veya genişletilmesinde.

Hipotez testi sorgusu algoritması, doğrulamaya dayalı bir algoritmadır. Bir hipotez öne sürülür ve seçilen veri kümesinde hipotez doğruluğu test edilir. Öne sürülen hipotez genellikle belirli bir örüntünün veri tabanındaki varlığıyla ilgili bir tahmindir. Bu tip bir analiz özellikle keşfedilmiş bilginin genişletilmesi veya arıtılması (refine) işlemleri sırasında yararlıdır.

Hipotez ya mantıksal bir kural ya da mantıksal bir ifade ile gösterilir. Her iki biçimde de seçilen veri tabanındaki nitelik alanları kullanılır. X ve Y birer mantıksal ifade olmak üzere "IF X THEN Y" biçiminde bir hipotez öne sürülebilir. Verilen hipotez seçilen veri tabanında doğruluk ve destek kıstasları baz alınarak sistem tarafından sınanır.

2.7.2. Sınıflandırma algoritması:

Sınıflandırma algoritması yeni bir veri elemanını daha önceden belirlenmiş sınıflara atamayı amaçlar [13]. Veri tabanında yer alan çokluklar bir sınıflandırma fonksiyonu yardımıyla kullanıcı tarafından belirlenir veya karar niteliğinin bazı değerlerine göre anlamlı ayrık alt sınıflara ayırır. Bu yüzden sınıflandırma, danışmanlı öğrenmeye (supervised learning) girer. Sınıflandırma algoritması bir sınıfı diğerinden ayıran örüntüleri keşfeder. Sınıflandırma algoritmaları iki şekilde kullanılmaktadır [14]:

Karar değişkeni ile sınıflandırma

Seçilen bir niteliğin aldığı değerlere göre sınıflandırma işlemi yapılır. Seçilen nitelik karar değişkeni adini alır ve veri tabanındaki çoklular karar değişkeninin değerlerine göre sınıflara ayrılır. Bir sınıfta yer alan çoklular karar değişkeninin değeri açısından özdeştir.

Örnek ile sınıflandırma

Bu biçimdeki sınıflandırmada veri tabanındaki çoklular iki kümeye ayrılır. Kümelerden biri pozitif, diğeri negatif çoklukları içerir [14]. Yaygın kullanım alanları, banka kredisi onaylama işlemi, kredi kartı sahteciliği tespiti ve sigorta risk analizidir.

2.7.3. Kümeleme algoritması

Kümeleme (clustering) algoritması veri tabanını alt kümelere ayırır. Her bir kümede yer alan elemanlar dahil oldukları grubu diğer gruplardan ayıran ortak özelliklere sahiptir. Bu yüzden kümeleme, danışmansız öğrenmeye girer. Danışmansız tekniklerden olan kümeleme, danışmanlı sınıflama için ön işlem olarak da çok sıkça kullanılır. Bilgi kazanımı (information retrieval) disiplini kümeleme konusundaki çalışmalar açısından oldukça zengin bir geçmişe sahiptir ve bu çalışmalar gömü adı altında toplanabilir. Tipik bir bilgi geri erişim sistemi için gömü, terimlerin belli bir ilişkiye göre düzenlenmesidir. Gömü, dizinleme ve erişim hizmetlerinde terimlerin kullanımına rehberlik eder. Bu özelliği ile gömünün bir yetki kütüğü (authority file) olduğu söylenebilir. Gömü ile amaçlanan; kullanıcı sorgusunu, sorguda kullanmadığı ama bilgi ihtiyacı ile ilişkili terimler ile genişletmektir. Sorgu genişletmede kullanılacak terimler gömü ile belirlenir. Böylece sorgular kullanıcının ifade şeklinden kısmen bağımsızlaştırılır ve sorguya eklenen terimler ile daha fazla ilgili belgeye erişme imkânı ortaya çıkar. Bir gömünün performansı da dizinleme ve/veya erişim aşamasında kullanıldığı ve kullanılmadığı durumlarda anma (recall) ve duyarlılık (precision) parametrelerinin karşılaştırılması ile ölçülür. Bu alanda yapılan çalışmalar gömünün üretildiği derleme benzer derlemelerde kullanılması şartıyla anma değerinde %20'lere yaklaşan artışlar elde edilebildiğini göstermiştir [14].

2.7.4. İlişkilendirme algoritması

İlişkilendirme algoritması sınıflandırma algoritması gibi nesnelerin anahtarları dışında verilen özellikleri arasındaki örüntüleri keşfeder. İlişkilendirme algoritmasının sınıflandırma algoritmasından farkı, danışmansız bir algoritma

olmasıdır. Yani algoritmaya verilen öğrenme ve test verilerinde girdi ve çıktı değerleri bulunmaz. Keşfe dayalı bir VM algoritmasıdır [15].

2.7.5. Sıra örüntüleri

Belirli bir olay veya eylemin bir başkasını izlemesindeki örüntüleri yakalamak için kullanılır. Zaman serileri ile arasındaki en büyük fark bu yaklaşımda sebep sonuç ilişkilerinin ön planda olmasıdır. Mesela markete giden bir müşteri tatlı aldıktan sonra belli bir zaman aralığında içecek de satın alıyorsa bu sıralı örüntüdür. Bu tip örüntüler perakende satış, telekomünikasyon ve tıp alanlarında yararlıdır [15].

2.7.6. Zaman serileri arasındaki bağımlılıklar

Bu yaklaşım veri nesnelerindeki belirli nitelik değerlerinin belirli zaman aralıklarında tekrarlanmasındaki örüntüleri bulmak için kullanılan yaklaşımdır. Belli frekanslarla tekrarlanan örüntüler bu yaklaşımla bulunur [15].

2.7.7. Regresyon analizi

Regresyon analizi; var olan bilgiler üzerinde formüller uygulayarak tahminler yapabilmek için kullanılır. Var olan verilerden doğrusal ve lojistik regresyon tekniklerini (istatistikler yardımı ile) kullanarak bir fonksiyon öğrenilebilir. Yeni veriler; tahminler yapabilmek için fonksiyonlara haritalanır. Yapraklardaki ortalama değerlere sahip karar ağaçları olan regresyon ağaçları, genel bir regresyon tekniğidir

2.8. Veri Madenciliğinde Dikkat Edilmesi Gereken Hususlar

Etkin bir VM algoritması geliştirebilmek için aşağıdaki hususlara dikkat edilmesi gerekmektedir [15]:

2.8.1. Veri gizliliği ve güvenliğinin sağlanması

Bir VTBK sisteminde keşfedilen bilgi pek çok farklı açıdan ve soyutlama düzeyinden izlenebildiği için, gizlilik ve veri güvenliği, VM sistemini kullanan kullanıcının haklarına ve erişim yetkilerine göre sağlanmalıdır.

2.8.2. Sonuçların yararlılık, kesinlik ve anlamlılık kıstaslarını sağlaması

Elde edilen sonuçlar analiz için kullanılan VT'yi doğru biçimde yansıtmalıdır. Bunun yanı sıra gürültülü ve aykırı veriler işlenmelidir. Bu işlem elde edilen kuralların kalitesini belirlemede önemli bir rol oynar.

2.8.3. Farklı tipteki verileri ele alma

Gerçek hayattaki uygulamalar makine öğreniminde olduğu gibi yalnızca sembolik veya kategorik veri türleri üzerinde değil, aynı zamanda tamsayı, kesirli sayı, çoklu ortam verisi ve coğrafi veri gibi farklı tipteki veriler üzerinde de işlem yapılmasını gerektirir. Kullanılan verinin saklandığı ortam, düz bir kütük veya ilişkisel VT'de yer alan tablolar olabileceği gibi, nesneye yönelik VT'ler, çoklu ortam VT'leri ve coğrafi VT'ler vb. de olabilir. Saklandığı ortama göre veri, basit tipte olabileceği gibi karmaşık veri tipleri (çoklu ortam verisi, zaman boyutlu veri, yardımlı metin, coğrafi veri vb.) de olabilir. Bununla birlikte veri tipi çeşitliliğinin fazla olması bir VM algoritmasının tüm veri tiplerini ele alabilmesini olanaksızlaştırmaktadır. Bu yüzden veri tipine özgü adanmış VM algoritmaları geliştirilmektedir.

2.8.4. Farklı ortamlarda yer alan veri üzerinde işlem yapabilme

Kurumlar yerel ağlar üzerinden pek çok dağıtık ve heterojen VT üzerinde işlem yapmaktadır. Bu VM'nin farklı kaynaklarda birikmiş biçimli ya da biçimsiz veriler üzerinde analiz yapabilmesini gerektirir. Veri büyüklüğünün yanı sıra verinin dağıtık olması, yeni araştırma alanlarının ortaya çıkmasına sebep olmuştur. Bunlar, koşut ve dağıtık VM algoritmalarıdır.

2.8.5. Veri madenciliği algoritmasının etkinliği ve ölçeklenebilirliği

Çok büyük hacimli veri içinden bilgi elde etmek için kullanılan VM algoritmasının etkin ve ölçeklenebilir olması gerekir. Bu, VM algoritmasının çalışma zamanının tahmin edilebilir ve kabul edilebilir bir süre olmasını gerektirir. Üssel veya çok terimli bir karmaşıklığa sahip bir VM algoritmasının uygulanması kullanışlı değildir.

2.8.6. Keşfedilen kuralların çeşitli biçimlerde gösterimi

Bu özellik keşfedilen bilginin gösterim biçiminin seçilebilmesini sağlayan yüksek düzeyli bir dil tanımının yapılmasını ve grafik arayüzünü gerektirir.

2.8.7. Farklı soyutlama düzeyi ve etkileşimli veri madenciliği

Büyük VT'lerden VM sorgularıyla elde edilecek bilginin edinilmesi güçtür. Bu yüzden VM sorgusu, elde edilen bilgilere göre kullanıcıya etkileşimli olarak sorgusunu değiştirebilmeyi, farklı açılardan ve farklı soyutlama düzeylerinden keşfedilen bilgiyi inceleyebilme esnekliğini sağlamalıdır.

2.9. Veri Madenciliğinde Yeni Trendler

2.9.1. Dağıtık veri madenciliği

Büyük miktarda dikkat çeken veri madenciliğinin bir alanı ise dağıtılmış ve kolektif olanıdır. Şu an yapılan veri madenciliğinin çoğu bir veritabanına ya da bir yere fiziksel olarak yerleştirilmiş bilginin veri ambarına odaklanır. Buna rağmen, durum bilginin farklı yerlere, farklı fiziksel lokasyonlara nasıl yerleştirilebileceğinde ortaya çıkmaktadır. Bu genel olarak dağıtık veri madenciliği (DDM) olarak bilinmektedir. Bu yüzden, amaç heterojen bölgelerde olan dağıtılmış bilgilerin etkili bir şekilde madenciliğini yapmaktır. Bunun örnekleri farklı veri tabanlarında, bulunan biyolojiksel bilgileri, farklı 2 firmanın veri tabanlarından gelen verileri ya da bir kurumun farklı dallarından gelen verilerin analizini hangisinin pahalı zaman tüketen bir süreç olduğunu bütünleştirmeyi içermektedir.

Dağıtık veri madenciliği (DDM) küresel veri modeli ile birlikte lokalize veri analizinin bir kombinasyonunu kullanarak geleneksel yaklaşımlar analizine farklı yaklaşımları sunmak için kullanılmaktadır. Daha özel terimlerle bu durum:

- Bölümsel veri modellerini oluşturmak için yerel veri analizini gerçekleştirme
- Küresel modeli geliştirmek için farklı veri sitelerinden gelen yerel veri modellerini birleştirme ile özelleştirilir.

Bu küresel model ayrı analizlerin sonuçlarını birleştirir. Sık sık üretilen küresel model, özellikle değişik lokasyonlardan gelen veriler değişik özellik ve karakteristik gösterirse, yanlış ve belirsiz olabilir. Bu problem özellikle dağıtılmış sitelerdeki verilerin homojen olmaktan ziyade hetrejon olduğunda kritiktir. Bu heterojen veri kümeleri dikey bölünmüş veri kümeleri olarak bilinir.

Kargupta tarafından sunulan bir yaklaşım dikey bölünmüş veri kümelerine daha iyi bir yaklaşımı, ortonormal temel fonksiyonların kavramını kullanmayı sağlayan ve verilerin küresel modellerini oluşturmak için temel katsayıları hesaplayan, kolektif veri madenciliğini (CDM) konuşmaktadır [16].

2.9.2. Metin madenciliği

Web sayfalarında yer alan hypertext, hypermedia bilgilerinin diğer çeşitli formlarını içeren veri madenciliği olarak ifade edilebilir. Bu bölüm ayrı ayrı işlenen, hem web madenciliği hem de çoklu ortam madenciliği ile yakından ilgilidir. Ama gerçekte içerikleri ve de uygulamaları birbirine yakındır. Dünya çapında web ciddi manada hypertext, hypermedia elementlerinden meydana gelmiştir, web'de bulunmayan diğer çeşit hypertext, hypermedia veri kaynakları da bulunmaktadır. Bunların örnekleri çevrimiçi kataloglarda, dijital kütüphanelerde, çevrimiçi bilgi veritabanlarında ve benzerlerinde bulunan bilgileri içermektedir. Başlık ve de alt başlıkların bu taksonomileri büyük bir ağ ya da başlıkların, birleşmiş link ve sayfaların hiyerarşik ağacını oluşturmak için bağlantılıdır. hypertext, hypermedia veri madenciliğinde kullanılan önemli veri madenciliği tekniklerinin bazıları sınıflandırma (danışmanlı öğrenme), kümelendirme (danışmansız öğrenme), yarı-yapılı öğrenme ve sosyal ağ analizini içermektedir.

Sınıflandırma durumunda ya da danışmanlı öğrenimde, süreç parçaların belirli bir sınıfın ya da grubun parçası olarak işaretlendiği eğitim verileri gözden geçirilerek başlar. Eğitim verisi algoritmanın eğitilmesi nedeni ile kullanılmaktadır. Sınıflandırmanın diğer bir uygulaması ise web başlık analizidir. Kullanıcıların aranılan kelime ile aynı anlamdaki ya da türde hece yapısına sahip benzer sayfalara erişmesini amaçlamaktadır. Buradaki sınıflandırma sadece kelime bazlı değil aynı

zamanda kategori bazlı sınıflandırmayı da amaçlamaktadır. Bu tür sınıflandırmada Naive Bayes, ilişkilerin modellemesi ve maksimum entropi gibi yöntemler kullanılmaktadır [16].

Danışmansız öğrenme tekniği olan kümeleme yönteminin, sınıflandırmadan temel farkı eğitim verisinin kullanılmıyor olmasıdır. Kümeleme birbirine benzer belgeleri bir gruplar. Bir karar ağacı gibi düşünürsek; daha az benzerlikli belgeler ağaç içerisinde daha yüksekte yer alırlar iken benzer belgeler ise ağaç hiyerarşinde yaprak seviyelerinde yerleştirilmesi ile sonuçlanır. Bu tür araştırmada kullanılan danışmansız öğrenme teknikleri k-means kümeleme yöntemi, birleştirici kümeleme yöntemi (Agglomerative Hierarchical Clustering), ve anlamsal indekslemeyi içermektedir.

Yarı danışmanlı öğrenme ve sosyal ağ analizi hiper ortam ağırlıklı veri madenciliği için önemli olan diğer tekniklerdir. Yarı danışmanlı öğrenme hem etiketlenmiş hem de etikentlenmiş belgelerin olduğu durumdur. Her iki tür belgeden de öğrenme ihtiyacı olmaktadır. Sosyal ağ analizi içinde uygundur çünkü web üzerinde yer alan her türlü kaynak bir sosyal ağ olarak düşünülmektedir. Graf uzaklıkları ve bağlantıları sosyal ağ çalışmalarında önem arz etmektedir [16].

2.9.3. Çoklu ortam veri madenciliği

Çoklu ortam veri madenciliği görüntü, video, ses ve animasyon içeren çeşitli tipteki verilerin madenciliği ve analizidir. Değişik tipteki bilgilerin madenciliğinin fikri çoklu ortam veri madenciliğin asıl amacıdır. Çoklu ortam veri madenciliği hiper metin ve hiper ortam veri madenciliği kadar metin madenciliğinin alanları ile birleştiği için bu alanlar birbirleri ile çok yakındır. Diğer alanları tanımlayan bilginin çoğu aynı zamanda çoklu ortam veri madenciliğine de uygulanmaktadır. Bu olan tercihen yeni olmakta ama bir o kadar da gelecek için umut vermektedir.

Çoklu ortam bilgisi, doğasında çok büyük bir miktarda çoklu ortam nesnesi olduğu için verilerin alışılagelmiş formlarından farklı bir şekilde temsil edilmelidir. Bir yaklaşım ise çoklu ortam tipi veriyi temel veri madenciliği tekniklerinden birini

kullanan analize uygun olan forma çeviren ama verinin kendine özgü karakteristiğini de göz önünde bulunduran şekilde kullanılabilecek çoklu ortam veri küpünü oluşturmaktır. Bu durum doku, biçim, renk ve ilgili niteliklerin ölçümü ve boyutlarının kullanımını içermektedir. Esas itibariyle, çok boyutlu uzamsal veri tabanını oluşturmak mümkündür. Çoklu ortam veri kümeleri üzerinde yürütülebilecek analizlerin tipleri arasında ilişkilendirme, kümeleme, sınıflandırma ve benzerlik analizi yer almaktadır.

Çoklu ortam veri madenciliğinde gelişen diğer alan ise ses veri madenciliğidir. Temel fikir verilerin örüntülerini belirlemek ya da veri madenciliği sonuçlarının özelliğini temsil etmek için ses sinyallerini basit bir şekilde kullanmaktır. Ses veri madenciliği ile veri analizi görsel veri madenciliğinden daha kolaylıkla yapılmaktadır [16].

2.10. Veri Madenciliğinde Karşılaşılan Problemler

Makina öğrenimi ile (MÖ), VM arasındaki farklar sıralanırken şu önemli detay hemen söylenir: MÖ küçük deneysel verilerle uğraşırken VM büyük hacimli gerçek dünya verileriyle uğraşır. Bu fark VM'de büyük sorunlar oluşturur. Bundan dolayı mesela küçük veri setleriyle ve yapay hazırlanmış verilerle doğru çalışan sistemler, büyük hacimli, eksik, gürültülü, NULL değerli, artık, dinamik verilerle yanlış çalışabilir. Bundan dolayı bu sorunların aşılması gerekmektedir [15].

2.10.1. Veri tabanı boyutu

Veri tabanlarında tutulan veriler iki boyutlu olarak genişlemektedir:

- Yatay Boyut: nesnelerin özellik sayılarıyla genişlemektedir.
- Dikey Boyut: nesnelerdeki kayıt sayısıyla genişlemektedir.

Geliştirilen pek çok algoritma yüzler mertebesindeki verilerle uğraşacak şekilde geliştirildiğinden aynı algoritmanın yüz binlerce kat daha fazla kayıtlarla çalışabilmesi için azami dikkat gerekmektedir. Veri hacminin büyüklüğünden kaynaklanan sorunun çözümü için uygulanacak alternatif çözümlerden bazıları:

- Örnekleme kümesinin yatay ve dikey boyutta indirgenmesi,
    - Yatay indirgeme: Nitelik değerlerinin önceden belirlenmiş genelleme sıra düzenine göre, bir üst nitelik değeri ile değiştirilme işlemi yapıldıktan sonra aynı olan çokluların çıkarılma işlemidir.
    - Dikey indirgeme: Artık niteliklerin indirgenmesi işlemidir.
- VM yöntemleri sezgisel/buluşsal bir yaklaşımla arama uzayını taramalıdır, vb.

Örnekleme kümesinin geniş olması bulunacak örüntüleri ne kadar iyi tanımlıyorsa, bu büyük kümeyle uğraşma zorluğu da o kadar artmaktadır [15].

2.10.2. Gürültülü veri

Veri girişi veya veri toplanması esnasında oluşan sistem dışı hatalara gürültü denir. Veri toplanması esnasında oluşan hatalara ölçümden kaynaklanan hatalar da dahil olmaktadır. Bu hataların sonucu olarak VT'de birçok niteliğin değeri yanlış olabilir.

Günümüz ticari ilişkisel veri tabanları bu tür hataların ele alınması için az bir destek sunmaktadır. VM'de kullanılan gerçek dünya verileri için bu sorun ciddi bir problemdir. Bu sebepten dolayı VM tekniklerinin gürültülü verilere karşı daha az duyarlı olması gerekir.

Sistemin gürültülü veriye daha az duyarlı olmasından kasıt, gürültülü verilerin sistem tarafından tanınması ve ihmal edilmesidir.

Chan ve Wong (1991), gürültünün etkisini azaltmak için istatistiksel yöntemler kullanmıştır. Sınıflama üzerine yaptığı çalışmalardan tanınan Quinlan'nın gürültünün sınıflama üzerine etkileri konusunda yaptığı çalışmada; etiketli öğrenmede etiket üzerindeki gürültünün öğrenme algoritmasının performansını doğrudan etkileyerek düşürdüğünü tespit etmiştir [15].

Tüme varımsal karar ağaçlarında uygulanan metotlar bağlamında gürültülü verinin yol açtığı problemler araştırılmıştır [15].

2.10.3. Null değerler

Eğer VT'de bir nitelik değeri NULL ise o nitelik bilinmeyen ve uygulanamaz bir değere sahiptir. VT'de birincil anahtar haricindeki herhangi bir niteliğin özelliği NOT NULL (NULL olamaz) şeklinde tanımlanmadığı sürece bu niteliğin değeri NULL olabilir.

Kurulacak bir ilişkide kullanılacak verilerin aynı sayıda niteliğe ve NULL olsa bile aynı sayıda nitelik değerine sahip olması gerekir.

Lee, NULL değerini ilişkisel veri tabanlarını genişletmek için aşağıdaki üç gruba girecek şekilde ayırmıştır [17]:

- bilinmeyen,
- uygulanamaz,
- bilinmeyen veya uygulanamaz.

Bu ayrımda şu ana kadar sadece bilinmeyen değer üzerinde çalışmalar yapılmıştır [18].

Veri kümelerinde var olan NULL değerler için çeşitli çözümler söz konusudur [17]:

- NULL değerli kayıtlar tamamıyla ihmal edilebilir,
- NULL değerli kayıtlardaki NULL değerler olası bir değerle güncellenebilir. Bu güncelleme için çeşitli yöntemler söz konusudur:
    - NULL değer yerine o nitelikteki en fazla frekansa sahip bir değer veya ortalama bir değer konulabilir,
    - NULL değer yerine varsayılan bir değer konulabilir,
    - NULL değerinin bulunduğu kaydın diğer özelliklerine göre, NULL değerinin kendine en yakın değerle güncellenmesi sağlanabilir vb.

2.10.4. Eksik veri

VM'de ilişkilerin kurulabilmesi ve istenen problemin çözümüne ulaşabilmek için gereken örnekleme kümesindeki 2 boyutun eksik olmaması gerekir. Bu boyuttaki eksiklikler şu şekilde olabilir:

• Yatay boyutta: Yatay boyuttaki eksiklik, örneklem kümesinde olması gereken nitelik veya niteliklerin olmamasıdır. Örnek olarak eğer insanların göz rengiyle alakalı bir hastalığın neye bağlı olduğu bulunmaya çalışılıyor ise, niteliklerden göz renginin örnekleme kümesinde bulunması gerekmektedir.

• Dikey boyutta: Dikey boyuttaki eksiklik örnekleme kümesindeki kayıtların eksik olmasıdır.

Örneğin bir süper markette yaşı 10 ve 25 yaşındaki kişiler her yaptıkları alışverişte bir ürünü sürekli alıyorlarsa, bu örüntünün keşfedilmesi için örnekleme kümesinde yeterli sayıda 10-25 yaş aralığına giren kayıtların bulunması gerekir. Eğer örnekleme kümesinde bu kayıtlar bulunmazsa gerçek hayatta var olan bir örüntü kaçırılmış olur [15].

2.10.5. Artık veri

Artık veri, problemde istenilen sonucu elde etmek için kullanılan örneklem kümesindeki gereksiz niteliklerdir.

Artık nitelikleri elemek için geliştirilmiş algoritmalar, özellik seçimi olarak adlandırılır. Özellik seçimi arama uzayını küçültür ve sınıflama işleminin kalitesini de artırır [19].

2.10.6. Dinamik veri

İçeriği sürekli değişen veri tabanlarıdır. Bunlara örnek kurumsal çevrim-içi veri tabanları gösterilebilir. Bir veri tabanındaki içeriğin sürekli değişmesi VM uygulamalarının uygulanabilmesini önemli ölçüde zorlaştırıcı sorunlar doğurmaktadır. Bu sorunlardan bazıları şunlardır [15,19]:

- Ortaya çıkan VM örüntülerinin sürekli değişim halinde olan verilerden hangisini ifade ettiğinin tespitinin zorluğu ve bu üretilen sonuçların zaman içinde eski üretilen sonuçlardan farkının tespiti ve gereken yerlerin güncellenme zorluğu olmaktadır.
- VM algoritmalarının çalışabilmesi için verilerin üzerine okuma kilidi konulması gerektiğinde, bu verilerin başka uygulamalar tarafından değişime açık olmaması veritabanı güncelleme işlemleri açısından sorun oluşturmaktadır.
- VM algoritmalarının ve çevrim-içi VT uygulamalarının aynı anda uygulanmasından kaynaklanan ciddi performans düşüşleri yaşanmaktadır.

2.11. Veri Madenciliği Araçlarının Karşılaştırılması

North Carolina Üniversitesi tarafından yayınlanan bir rapora göre veri madenciliği yazılımlarının karşılaştırma sonuçları takip eden bölümlerde yer almaktadır [20].

Çizelge 2.1. Veri madenciliği araçlarının listesi

Araç İsmi	Şirket İsmi	URL Adresi
CART	Salford System	http://www.salford-systems.com/
MARS	Salford System	http://www.salford-systems.com/
SEE5	RuleQuest Research	http://www.rulequest.com/
DIGITAL LOGIC	Reduct &Lobbe Technologies Inc.	http://www.reduct.com/
GRIT BOT	RuleQuest Research	http://www.rulequest.com/
SAS	SAS Institute	http://www.sas.com/
MAGNUM OPUS	RuleQuest Research	http://www.rulequest.com/
SPSS	SPSS Inc.	http://www.spss.com/
LERS	RS System	
WEKA	University of Waikato	http://www.cs.waikato.ac.nz/

Çizelge 2.2. Veri madenciliği araçlarının üzerinde çalıştıkları platformlar

Araç İsmi	Microsoft (98/2000/XP/NT)	Unix	Müşteri/Sunucu	Veri Tabanı Bağlanırlığı
CART	Evet	Evet	Evet	Hayır
MARS	Evet	Evet	Hayır	Hayır
SEE5	Evet	Evet	Hayır	Hayır
DIGITAL LOGIC	Evet	Bilinmiyor	Hayır	Hayır
GRIT BOT	Evet	Evet	Hayır	Hayır
SAS	Evet	Hayır	Evet	Evet
MAGNUM OPUS	Evet	Hayır	Hayır	Hayır
SPSS	Evet	Hayır	Hayır	Hayır
LERS	Evet	Hayır	Hayır	Hayır
WEKA	Evet	Evet	Hayır	Evet

Çizelge 2.3. Veri madenciliği araçlarında veri giriş yöntemleri

Araç İsmi	İstatiksel Analiz Paketleri (SAS SPSS)	Hesap Çizelgeleri (Excel Lotus)	Rasyonel Veri Tabanları (Informix Oracle)	Veri Çeviri Aracı
CART	Evet	Evet	Evet	DBMS/COPY
MARS	Evet	Evet	Hayır	DBMS/COPY
SEE5	Hayır	Hayır	Hayır	Sadece Kendi Formatında Kullanılan
DIGITAL LOGIC	Evet	Evet	Hayır	(*.typ) format dosyasını tanımlamalı
GRIT BOT	Hayır	Hayır	Hayır	Sadece Kendi Formatında Kullanılan
SAS	Evet	Hayır	Hayır	
MAGNUM OPUS	Hayır	Hayır	Hayır	Sadece Kendi Formatında Kullanılan
SPSS	Evet	Hayır	Hayır	
LERS	Hayır	Hayır	Hayır	Sadece Kendi Formatında Kullanılan
WEKA	Hayır	Evet	Evet	Menüde kurulu AREF formatına

Çizelge 2.4. Veri madenciliği araçlarının algoritma destekleri yönünden karşılaştırılması

Araç İsmi	Decision Tree	Doğrusal/İstatiksel	Çok Katmanlı Algılayıcı	KNN	Radyal Bazlı Fonksiyonlar	Bayes	Kural Tümevarımı	Çok Terimli Ağlar	Genelleşmiş Doğrusal Model	Zaman Serileri	Ardışık Keşif	Kmeans	Birleşim Kuralları
CART	Evet												
MARS		Evet											
SEE5	Evet						Evet						
DIGITAL LOGIC	Evet						Evet						
GRIT BOT													
SAS	Evet	Evet	Evet		Evet				Evet	Evet		Evet	Evet
MAGNUM OPUS													Evet
SPSS		Evet							Evet				
LERS													Evet
WEKA	Evet	Evet	Evet			Evet	Evet						

Çizelge 2.5. Veri madenciliği araçlarının görselleştirme destekleri açısından karşılaştırılması

Araç İsmi	Histogramlar	Pasta Grafiği	Dağılım/Şerit Çizimleri	Sınıflandırma Ağacı	Korelasyon Çizimleri
CART			Evet	Evet	
MARS	Evet		Evet		
SEE5	Evet				
DIGITAL LOGIC					
GRIT BOT					
SAS	Evet	Evet	Evet		Evet
MAGNUM OPUS					
SPSS	Evet	Evet	Evet		
LERS					
WEKA				Evet	

Çizelge 2.6. Veri madenciliği araçlarının kullanıldığı alanların karşılaştırılması

Araç İsmi	**Pazarlama**	**Doğrudan Posta Gönderimi**	**Finansal Servis**	**Üretim**	**Sağlık Hizmetleri**	**Askeri**
CART	Market Segmentasyonu, Müşteri Profili, Saklama/Zayiat Analizi	Market Segment Karlılığı, Kampanya Hedef Tespit, Tepki Tahmini	Kredi Kartı Puantajı, Dolandırıcılık Tespiti	Birleşke Şerit Hataları, Kalite Kontrol	Klinik Denemeleri, Biomedikal Araştırma	
MARS	Yeni Müşteri Elde edimi		İflas Tahmini			
SEE5	Web de reklam tahmini		Kredi Risk Erişimi, Gerçek Arazi Profili	Üretim Süreç Kontrolü		
DIGITAL LOGIC	Market Örüntü Araştırması		Stok Trend Analizi,	Tarım Örüntü Araştırması, Kalite Kontrol, Süreç Model Kontrolü	Medikal Teşhis Etme	
GRIT BOT		Müşteri Segmentasyonu	İflas Tahmini	Tarım Analizi	Hastalık Teşhis Etme, Gen Analizi	
SAS	Pazarlama Zamanlaması, Müşteri Mansiyon Analizi, Müşteri İlişki Yönetimi	Müşteri Sınıflandırma ve Segmentasyonu	Portfolyo Performans Analizi			
MAGNUM OPUS	Müşteri Bölmesi		Kredi Puantajı			
SPSS	Müşteri Yaşam Değer Analizi, Çapraz Satış İçermesi, Saklama, Elde Etme		Finansal Tahmin, Bütçe Analizi	Kalite Geliştirme Analizi		
LERS			Kredi Puantajı		Teşhis Etme İçin Karar Desteği	İş Performansı Anketi
WEKA						

Çizelge 2.7. Veri madenciliği araçlarının genel olarak değerlendirilmesi

Araç İsmi	Avantajlar	Dezavantajlar
CART	Ağaç Seçeneklerinin Derinliği	Zor Dosya Girdi/Çıktısı, Sınırlı Görselleştirme
MARS	Çoklu Bağlanım	Sınırlı Algoritmalar
SEE5	Ağaç Seçeneklerinin Derinliği	Çok Az Veri Seçeneği, Sınırlı Görselleştirme
DIGITAL LOGIC	kesin olmayan, eksik ve tutarsız veri ile çalışabilir	Sınırlı Algoritmalar
GRIT BOT	Kuvvetli Anormallikleri Bulma	Sınırlı Algoritmalar
SAS	Algoritmaların Derinliği, Görsel Arabirim, Makul Grafik Çıktısı	Kullanımı Zor, Master SAS Programlama Dili Gerekli
MAGNUM OPUS	Hızlı Devir (Doğrusal Hesaplama Zamanı), İyi Esneklik, Büyük Veri Kümeleriyle İlgilenebilir	Sınırlı Kullanım
SPSS	Algoritmaların Derinliği, Yaygın kullanım, Güçlü İşlevsellik	Kullanımı Zor
LERS	Kullanımı Kolay	Sınırlı Algoritmalar, Görselleştirme Araçları Yok
WEKA	Herhangi Bir Platformda Çalışabilir	Düşük Hız

3. SPAM

3.1. SPAM E-posta

Spam, bilgisayar kullanıcıları tarafından karşılaşılan en zorlu problemlerden birisi haline gelmiştir. Son 10 yılda spam önlemede kullanılan yöntemlerin gelişmesine rağmen spam kaynaklarının da metotlarında değişiklikler olmuştur.

Spam gönderiminde bulunanlar gelişen anti-spam teknikleri ile mücedelede kimliklerini gizlemek amacı ile e-posta mesajlarında sahte e posta başlıklarını kullanmaya aşlamışlardır.

Spam gönderiminde bulunanlar aynı zamanda e posta adreslerini elde etmek için internet üzerinde forumlarda yer alan açık bilgileri ya da internet sitelerine izinsiz erişim ile erişerek kullanıcı e-posta adreslerini ele geçirmek için yaratıcı stratejiler kullanmışlardır.

Yakın zamanda spam gönderiminde bulunanların; spam e-posta göndermek için internet üzerindeki kontrolsüz (güvensiz) bilgisayarları ele geçirerek gaspedilmiş bilgisayarlar (Botnet) aracılığı ile büyük ölçekli spam e-posta gönderimlerinde bulundukları gözlenmiştir. Bu tür bir olaya karışmış bilgisayarın kullanıcısı bir botnet'e dahil olduğunun farkında değildirler. Botnetler gönderdikleri mail sayısı ve aştıkları ISP filtrelerine göre derecelendirilmektedirler.

Spam e-postanın bu yeni jenerasyonu e-posta alıcılarına ve ISP'lerde yüke neden olmadan kriminal suçlarda kullanılmaktadır.

Spam e-posta içeriklerdeki diğer tehlike ise alıcının mesaj içerisindeki bir linke tıklaması ya da işlemi yapması sonucu bilgisayarındaki kişisel bilgilerinin farkında olmadan spam gönderiminde bulunanlara göndermekte ya da bilgisayar virüslerinin yayılımına veya bir botnete dahil olarak internet üzerinden işlenen suçlara dahil olmaktadır.

Kapsamı sürekli genişlemekte ve değişmekte olan spamlarin etkisini ve miktarını belirlemek oldukça zordur.

3.2. SPAM E-posta Özellikleri

İnternet üzerinden aynı mesajın yüksek sayıdaki kopyasının, bu tip bir mesaji alma talebinde bulunmamış kişilere, zorlayıcı nitelikte gönderilmesi SPAM olarak adlandırılmaktadır. Spam çoğunlukla ticari reklam niteliğinde olup, bu reklamlar sıklıkla güvenilmeyen ürünlerin, çabuk zengin olma kampanyalarının duyurulması amacına yöneliktir.

Spam, gönderimde bulunan açısından çok küçük bir maliyet ile gerçekleştirilebilirken mali yük büyük ölçüde mesajin alıcıları veya taşıyıcı, servis sağlayıcı kurumlar tarafından karşılanmak zorunda kalınmaktadır.

İnternet kullanıcıları üzerindeki etkileri incelendiğinde iki tip spam vardır. E-posta aracılığıyla gönderilen spam, doğrudan gönderilen mesajlarla, bireysel kullanıcıları hedef almaktadır. E-posta spam listeleri genellikle Usenet gönderilerinin taranması, tartışma gruplarının üye listelerinin çalınması veya web üzerinden adres aramalarıyla oluşturulmaktadır. E-posta tipindeki spam gönderileri tipik olarak e-postal alıcısının en basit anlamı ile internet erişimi için ödediği bir maliyete neden olmaktadır.

E-posta yolu ile gönderilen spam türlerinden ticari içerikli olan UCE (Unsolicited Commercial E-mail - talep edilmemiş ticari e-posta) adından da anlaşılacağı gibi istenmediği halde gönderilen bir ürünü ya da hizmeti tanıtıcı elektronik posta iletileridir.

İçeriğinin mutlaka ticari olması gerekmeyen UBE (Unsolicited Bulk E-mail - talep edilmemiş kitlesel e-posta), aynı anda yüzbinlerce e-posta hesabına gönderilen e-posta iletileridir. Bu iletiler ticari içerikli olabileceği gibi politik bir görüşün propagandasını yapmak ya da bir konu hakkında kamuoyu oluşturmak amacı ile gönderilen e-posta iletileri de olabilir.

Spam ile ilgili diğer önemli bir nokta, bir iletinin spam olarak nitelendirilmesi için kullanılacak ölçüt iletinin içeriği ile alakalı olmak zorunda değildir. Herkesin üzerinde hemfikir olduğu, önemli bir toplumsal duyarlılığa sahip bir konu hakkında

görüş bildirmek için kitlesel olarak gönderilen bir iletide spam olarak nitelendirilebilir.

Bir diğer sık rastlanılan spam e-posta tipi ise MMF (Make Money Fast – Kolay Para Kazanın) iletileri; zincir iletiler ya da piramit benzeri pazarlama yapıları ile ilgili gelen iletilerdir. Piramitin en üstündeki isme para gönderip listenin altına kendinizi eklediğinizde para kazanmaya başlayacağınıza ilişkin e-posta alıcısını yönlendiren bu tip iletiler spam iletilerine örnek olarak verilebilir.

Spam gönderiminde bulunanlar çok sayıda doğru e-posta adresini ele geçirmek amacı ile mümkün olduğu kadar çok listeye üye olmaya çalışırlar. 20 veya daha fazla haber öbeğine aynı anda gönderilen bir ileti spam kapsamında incelenmektedir.

Spam e-postalar genel olarak aşağıdaki karakteristik özellikleri sergilerler:

- Birden fazla alıcıya aynı içerik ile gönderilirler.
- Çoğunlukla alıcıya hiçbir şey ifade etmezler.
- Çirkin ya da yasadışı içerikle gelirler ya da onlara yönlendirirler.
- İçerikleri yanıltıcıdır.
- Mesaj başlık bilgileri genellikle doğru değildir.
- Dolayısıyla geriye dönük izleme hayli zor olur.
- Alıcıların dağıtımdan ileti almak istemediklerini belirtebilecekleri fonksiyonel bir adres sunmazlar.
- Elde edilmesi ve kullanılması kişilik haklarına tecavüz niteliği taşıyan içeriklere sahip olurlar.

3.3. SPAM E-Postaların İçerikleri

Spam e-postalar çok çeşitli içeriklerle kullanıcının karşısına çıkabilmektedir. Spam hareketinin doğası itibarı ile kanun dışı ve normal koşullarda pazarlanması yasak olan ürünler ve pornografi, diğer alternatiflerinin önüne geçmektedir.

Aşağıdaki tablo, spam e-postaların 2003 – 2004 yılları arasındaki içerik değişimini göstermektedir.

Çizelge 3.1. Spam e-posta içeriklerinin 2003-2004 yılları arasındaki değişimi

Ürün	2003	2004	Değişim	Açıklama
Porno / Sex (Grafiksiz)	%17	%34	17%	Porno içerikli sitelere linkler, sex yazıları
Sigorta Hizmetleri	%1	%4	3%	Ev, otomobil, sağlık ile ilgili sigorta hizmetleri
Bitki / İlaç	%8	10	2%	Ucuz ilaçlar, uyuşturucular
Finansal	%12	%13	1%	Çabuk para kazanma yöntemleri
Seyahat / Kumar	%2	%3	1%	Uçak biletleri, rezervasyonlar, Internet kumarhane reklamları
Saadet Zincirleri	%8	%7	-1%	İnsan getirdikçe kazanacaksınız konsepti
Haberler	%9	%6	-3%	Kullanıcıya hiçbir şey ifade etmeyen haberler
Diğerleri	%13	%8	-5%	Geri kalan ve SPAM gibi görünen her şey
Porno / Sex (Grafikli)	%13	%7	-5%	Porno resimler içeren her şey
Şüpheli Ürünler	%20	%10	-10%	Kırılmış yazılımlar, düzmece diplomalar vs.

3.4. E-posta Adreslerinin Elde Edilmesi

Spam e-posta gönderiminde bulunanlar, kullanıcıların e-posta adreslerini ele geçirmek için değişik yöntemler uygulamaktadırlar. Bunlardan en önemlisi e-posta adreslerinin zincir e-postalar aracılığı ile ve web sayfalarından temin edilmesidir. Zincir e-posta gönderiminden uzak durarak ya da mesaj alıcı kısmında gizli kısmına alıcıları yazarak, spam gönderiminde bulunanların e-posta adreslerini zincir e-postalar aracılığı ile elde etmesinin önüne geçilebilinir.

3.4.1. Web sayfaları

Spam yapmak üzere e-posta adresi elde etme yöntemlerinin başında e-posta adreslerini, web sayfalarını linkleri takip ederek teker teker dolaşan web bot'lar ile elde etmek gelmektedir. Web bot'lar, kendi kendine çalışan ve internet'i bir arama motoru gibi tarayarak ele geçirdiği e-posta adreslerini bir veritabanına saklayan basit uygulamalardır ve son derece yaygındırlar.

3.4.2. Zincir e-postalar

Zincir e-postalar birçok kişinin birbirine ilettiği e-postalara verilen isimdir. Elden ele binlerce e-posta adresine ulaşan e-postaların başlık bilgileri içerisinde daha önce hangi adreslere CC (carbon copy)'lendiği bilgisi kolaylıkla çıkarılabilmektedir. Bu sebeple spam yapmak için e-posta adresi toplayan şahıs ya da şirketler, insanların çok fazla ilgisini çekebilecek çoğunlukla da yalan olan haberleri, dini sömürü içeren iletileri ya da duygusal sömürü içerikli e-postaları "bu e-postayı listendeki herkese ilet" konsepti ile insanlara dağıtmaktadırlar. Bu e-postalar kendilerine yeniden döndüğünde e-posta adreslerini spam veritabanlarına eklemektedirler. 3 kişi tarafından forward edilmiş ortalama bir zincir mail içerisinde yaklaşık 200 e-posta adresi bulunabilmektedir.

3.4.3. Alan adı kayıtları

Alan adı kaydının doğal süreci gereği, bu alan adından sorumlu kişinin bir iletişim e-posta adresi alan adı ile ilişkilendirilir. Herhangi bir (örneğin http://ripe.net gibi) 'whois' veritabanından sorgulanan alan adı için bir e-posta adresi elde edilebilir. Alan adı kayıtlarından elde edilen e-posta adreslerin son derece düşük bir yüzdeye sahiptir.

3.4.4. E-posta adresi satışları

E-posta adreslerinin spam yapanlar arasında ya da spam ile ürünlerini duyurmak isteyen şirketlere satılması büyük bir illegal endüstri halini almıştır.

3.4.5. Güvenlik ihlalleri, virüsler ve diğerleri

E-posta adresleri daha önce bahsedilen yöntemlerin dışında e-posta sunucularına gerçekleştirilen saldırılar sonucunda elde edilen e-posta sunucusu kayıtlarından da elde edilebilmektedir. E-posta sunucularının tamamı dışarıya gönderdikleri ve kabul ettikleri e-postaların tarih bilgisi ve alıcı/gönderici adres bilgilerini bir günlük dosyasına kaydetmektedir ve bu dosyalar içerisinde çok fazla sayıda geçerli e-posta hesabı yer almaktadır.

Ayrıca virüs vb. diğer zararlı yazılımlar bilgisayarlara spam e-postalar yolu ile ya da kişinin bilgisayarına taktığı bir medya üzerinden bulaşarak e-posta adresi toplama ve spam yapma karakteristiklerini ortaya koymaktadırlar. 2001 yılında yapılan bir araştırmaya göre; istenmeyen ticari e-postaların yaklaşık %98'lik kısmı virüs ya da worm'lar aracılığı ile gönderilmektedir.

Son kullanıcıların dikkatsizliği ve çoğunlukla Microsoft işletim sistemi ailesindeki uygulamaların güvenlik açıklarından faydalanan warmlar bulaştıkları bilgisayarlardaki adres defterlerini kullanarak, aynı adres defterinde yer alan iki kişinin birbirini tanıma ihtimalinin yüksekliğini göz önünde bulundurarak sahte (spoofed) başlıklarla e-postalar göndermekte ve yayılmaya çalışmaktadır.

3.5. Spam Üzerine Bazı İstatistikler

Spamlaws.com sitesinden yayımlanan istatistiklere göre [21];

Spam e-postaların yaklasık %14'ü okunmakta, yine bu spam e-postalarda geçen reklam ürünlerinden yalnızca %4'lük bir kısmı ise alınmaktadır.

Spam gönderiminde bulunanlar spam e-postayı coğunlukla ürün satmak ve kişisel bilgilerin ele geçirilmesi amacı ile kullandıkları görülmektedir. Bununla birlikte spam e-posta ile hedeflenilen ürün satışlarının başarılı olamamasının arkasında; hedef kitlenin ürün ile ilgili olmaması ya da spam e-postaları okumaması gösterilmektedir.

Tek bir ürün satışı gerçekleştirebilmek için yaklaşık olarak 12.5 milyon spam e-postanın gonderilmesi gerektiği tespit edilmiştir.

Toplam e-posta trafiğinin %80'ninin spam olması problemin büyüklüğüne işaret etmektedir.

Bir gün içerisinde yaklaşık olarak 14.5 milyar spam e-posta dolaşmaktadır. Bu da günlük e-posta trafiğinin %45'ine karşılık gelmektedir.

- Spam e-postayı en çok üreten ülkeler arasında ABD ve Kore yer almaktadır.

- Spam e-postaların %36'sı reklam amaçlı e-postalardır.
- Spam e-postaların %31.7'si cinsel içerikli e-postalardır.
- Spam e-postaların %26.5'i finansal konuları içeren e-postalardır.
- Spam e-postaların %2.5'i dolandırıcılık amaçlıdır.
- Tüm spam e-postaların ~%73'ü ise kimlik bilgilerini ele geçirmeyi amaçlayan (phishing) e-postalardır.
- Spam e-postaların ~20.5 milyar $'lık üretim verimliliği kaybına neden olduğu, bunun çalışan başına 1934 $'a karşılık geldiği tespit edilmiştir.

Önümüzdeki 2 sene içerisinde günlük spam e-posta sayısının yaklaşık 58 milyar adedi bulacağı, bunun da yaklaşık olarak 198 milyar dolarlık iş gücü kaybına neden olacağı tahmin edilmektedir.

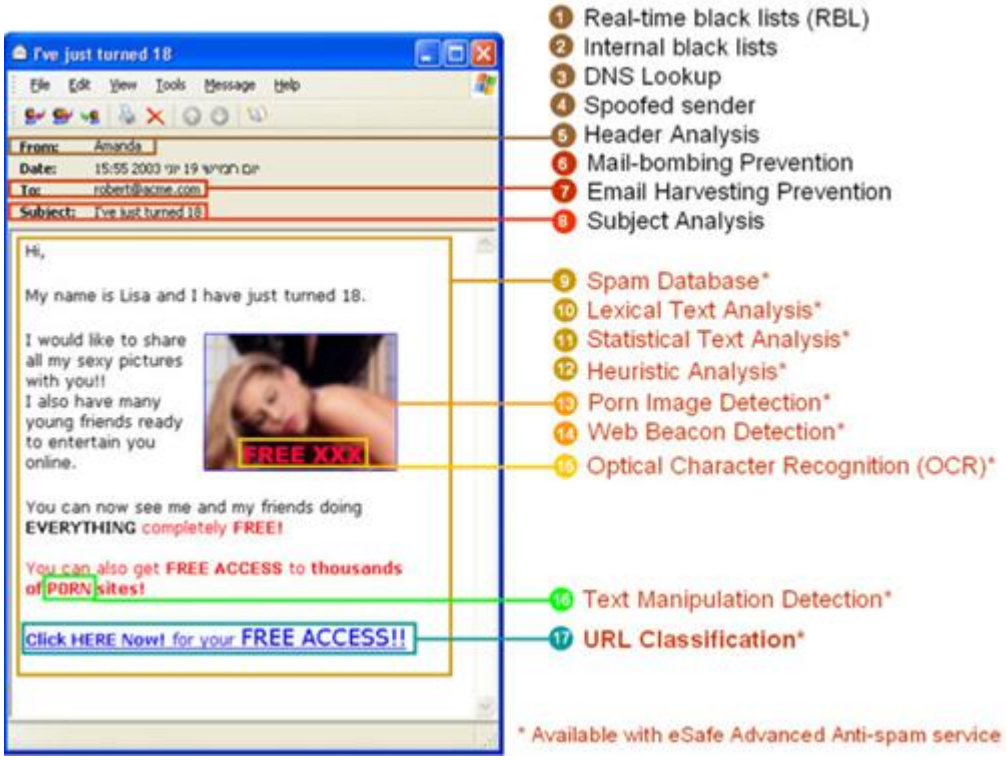
Yine bu tahminlere göre anti-spam maliyetinin; her bir e-posta hesabı icin ~49$ lık bir maliyete ve toplamda 257 milyar $'lık bir iş gücü kaybına neden olacağı öngörülmektedir [21].

3.6. Anti-Spam Yazılımları

Spam e-postalar istenmeyen e-postalar olmasının yanında virüs yayma, casus yazılımlar aracılığı ile bilgilerinizi ele geçirme özellikleri nedeni ile tehlike arz etmektedirler. Anti-spam yazılımları her bir posta kutusu için ortalama 25 - 40 $'lık bir maliyet gerektirirler.

3.7. Spam Analizinde Dikkat Edilen Genel Yaklaşımlar

Spam ile savaşmanın birçok tekniği vardır; hiçbiri teknik ya da tekniklerin tamamı spamı tamamen engelleyemez. Bu tekniklerin kombinasyonu kullanılarak, spam en düşük seviyeye düşürülebilir. Resim 3.1.'de yer alan tipik bir spam mesaja baktığımızda numaralandırılmış alanlarda spam içerik olduğu görülmektedir [22].



Resim 3.1. Spam e-posta örneği.

- Gerçek zaman kara listeleri (RBL)
- Dahili kara listeleri
- DNS arama
- Sahte gönderici adresi
- Başlık Analizi
- Posta-bombalama önlemi
- E-posta hasat önlemi
- Konu analizi
- Spam veri tabanı
- Anlamsal metin analizi
- İstatistiksel metin analizi
- Bulgusal analiz
- Porno görüntü tespiti
- Web uyarı tespiti
- Optik karakter tanımlaması

- Metin manipulasyon tespiti
- URL sınıflandırması

Spam e-posta filtrelemedekullanılan metotlara ilerleyen kısımlarda yer almaktadır.

3.7.1. Gerçek zaman kara listeleri

Bu teknik, RBL (Real time black list) olarak bilinen, gelen IP adreslerini yasaklı olup olmadığını doğrulamak için kontrol eder. RBL listeleri değişik organizasyonlar tarafından güncellenmekte ve paylaşılmaktadır.

MAPS (www.mail-abuse.org), ORDB (www.ordb.org) gibi adresler spam gönderiminde bulunanlar tarafından kulanılan e-posta sunucularının IP adreslerinin düzenli olarak güncellemekte ve paylaşmaktadırlar Değişik kara listelerin kombinasyonu www.declude.com/junkmail/support/ip4r.htm adresinden bulunabilir.

Değişik kurum ya da organizasyonlar tarafından farklı RBL'ler oluşturulduğu için bunların kesişiminden faydalanmak yararlı olacaktır.

3.7.2. Dahili kara listeler ve beyaz listeler

RBL bazen yeterli değildir. Bazı durumlarda daha spesifik ve doğru listelere ihtiyaç olacaktır. Göndericiyi e-posta adresi ile engelleme eski bir teknik olmasına rağmen, e-postaları engelleme iyi sonuçlar verebilir.

Beyaz listeler, kara listelerin tersidir. Abone olunmuş e-posta listeleri ve haber bültenleri gibi güvenilen alanlardan gelen e-postaların erişimine izin vermektedir.

3.7.3. DNS kontrolü

Bu teknikte e-posta domainlerine ait sunucuların DNS listelerine kayıtlı olması, DNS'te kayıtlı olmayan e-posta sunucularından gelen e-postaların güvensiz olarak değerlendirilmesi mantığına dayanmaktadır. Bu nedenle sistem yöneticilerinin e-posta sunucularının host ismini DNS'te doğru olarak kayıt ettirmesi önem arz etmektedir.

3.7.4. Aldatmaya karşı koruma (Anti-Spoofing)

E-posta adresi dolandırıcılığı, spam gönderiminde kullanılan e-posta adresinin, hedef domaindeki e-posta adresleri ile aynı uzantılı gösterilmesi şeklinde yapılmaktadır.

- Bir örnek alice@company.com olarak göstererek john@company.com a e-posta göndermektir.
- Diğer bir örnek john@company.com olarak göstererek john@company.com a e-posta göndermektir. Alıcı kendinden e-posta almış gibi görür.

3.7.5. Başlık analizi (Header Analysis)

Başlık doğrulaması; e-posta SMTP başlığının spamcılar tarafından değiştirilmediğine emin olmak amacı ile standartlara uygun olup olmadığının kontrol edilmesi işlemidir. Bazı spam gönderici uygulamalar (X-MAILER) kullanılan e-posta müşterisinin ismi ve diğer veriler gibi SMTP başlığına kesin tanımlanabilir bilgi sokabilir.

3.7.6. E-posta bombalama

E-posta bombalama yönteminde; sözlük yöntemi ile alıcı e-posta adresleri üretilerek bir posta sunucusuna çok sayıda e-posta göndererek e-posta sunucunun çalıştırılamaz hale gelmesi (DoS) hedeflenmektedir.

Çok sayıda e-posta alıcısı ile e-posta kullanımı, SMTP sunucusunun işlem yapmasına engel olan başka bir servis engelleme tekniğidir. Her bir e-posta için maksimum sayıda alıcıyı limitleme, doğru bir postada beklenenden daha fazla alıcıyı içeren piramit e-posta engellemesini etkin hale getirir.

Anti-spam yazılımının DoS önleme desteği ile aşırı yüklenmelere neden olabilecek trafik akışı ve bir mail içerisinde çok sayıda alıcının tespit edilerek e-postanın göz ardı edilmesi özellikleri servis engellemeye yönelik spamlar için önem arz etmektedir.

3.7.7. Dizin hasat saldırılarının önlemesi (Directory Harvesting Attacks)

Dizin hasat saldırısı (DHA) spam gönderiminde bulunanların hedef almış olduğu organizasyonların geçerli e-posta adreslerini toplamaya yönelik olan bir saldırı yöntemidir. Bir DHA boyunca, spammerlar johna@compnay.com, john@company.com gibi adreslere posta iletme çabasında bulunurlar.

Alıcı posta sunucusu tarafından reddedilemeyen adresler geçerli olarak kabul edilir ve sonra derlemek ve spam postalama listesi olarak satmak için kullanılır.

Anti-spam yazılımının bu tür saldırılarıda önlemesi önem arz etmektedir.

3.7.8. Konu analizi

Birçok spam mesajı e-posta konu kısmında genel metini içermektedir. Spamı açık bir şekilde tanımlamak için kullanılabilecek bu gibi konuların örnek listesi:

- Hızlı zengin ol
- Üniversite diploması
- Para biriktir
- Viagra çevrimiçi
- Kredi geri ödeme
- Büyüt ..
- Para kazan..
- Vs.

Anti-spam yazılımın başlık kısmında spam gönderiminde bulunanlar tarafından kullanılan anahtar kelimeleri statik ve dinamik olarak güncelleme ve bu tür spamleri engelleme yeteneğine sahip olması önem arz etmektedir.

3.7.9. Spam veritabanı

Spam veri tabanı teknolojisi; alınan e-postaların aktarma imzalarını çıkartarak; daha önceden spam e-posta olarak tasniflenmiş e-postaların veritabanındaki doğrulama

imzaları ile kontrol eder. Gerçek zamanda spam engelleme potansiyeline sahip olduğu için eğer doğru uygulanırsa bu çok güçlü bir teknolojidir.

Spam çok biçimli olabilir, yani her bir toplu gönderimle, hatta her bir gönderilen posta ile ara ara belirsiz bir şekilde modifiye edilebilir. İyi bir doğrulama imza sistemi spam örneklerindeki varyasyonlar için izin vermelidir. Spamın değişik bölümlerini kontrol ederek ve akıllı doğrulama imzası oluşturarak herhangi bir özel spamın birçok değişkeni için tanımlanabilir.

Hatta daha fazla etkili olmak için, birçok "junk" karakterlerinden, anti-spamı atlatmak girişimindeki veri ve etiketlerden her bir şüpheli spamı "temizlemek" için ihtiyaç vardır.

Doğrulama imzalarının bulunduğu veritabanının güncellenebilir olması önem arz etmektedir.

3.7.10. Anlamsal metin analizi

Anlamsal metin analizi e-postanın içeriğinde spam olarak çevirilebilen metin dizilerini arar. Boolean mantığını içeren anlamsal kurallar üzerinedir.

Örnek bir kural:

"Bütün doğal OR bütün-doğal OR doğal içerikler AND kilo-kaybı OR kilo kaybı" şeklinde olabilir.

3.7.11. Bayesian filtrelemesi

İstatiksel analiz veya Bayesian filtrelemesi çok sayıdaki spam mesajın analizinden elde edilen istatistikten baz alınarak yapılmaktadır. Yüksek öğrenme oranı ve düşük yanlış pozitif oranlarından dolayı bu yöntem önem arz etmektedir.

3.7.12. Bulgusal analiz

Bulgusal analiz karışık yabancı karakter seti, sunucu sorgusu olan imaj linkleri, belirsiz, çıkarılamayan karakterlerin karışımı, farklı şifreleme metodları vs. gibi

kesin genel karakteristiklerin varlıkları baz alınmış spam e-posta gibi görünen e-postaları tanımlamada kullanılır.

Spam gönderiminde bulunanlar spam e-postanın yakalanmasını zorlaştırmak için kullandığı bir genel teknik otomatik olarak rasgele birçok junk metini ya da e-posta alıcısına görünmeyen fakat aktarma imzalarını çıkaran ürünleri karıştırabilen rasgele junk HTML etiketleri eklemektedirler.

Örnek olarak bir spam e-postasında gömülü junk HTML etiketlerinin örnek bir bölümü aşağıda yer almaktadır:

<!B>D<!D>ig<!X>ital<!D> Cab<!E>l<!X>e<!D>FI<!X>LTE<!D>RS a<!X>re<!Y>Fi<!D>na<!E>lly Ava<!E>ilab<!D>le

HTML yorum etiketleri kaldırıldığında, mesaj şu şekilde olmaktadır:

Digital Cable FILTERS are Finally Available

3.7.13. Porno görüntü tespiti

Spam e-postaların çok büyük bir bölümü görsel pornografik içerik bulundurmaktadır. Bu içerik sadece zaman ve kaynak tüketimi değildir aynı zamanda saldırı amaçlı da olabilir. Bu tür e-postalar genelde resim üzerinden linkler sunmaktadır.

3.7.14. Web uyarıları

Web uyarıları canlı e-posta adreslerini tanımlayan ve bu adreslere spam göndermeyi çoğaltan gelişmiş spamcıların elindeki çok güçlü bir araçtır. Web uyarılı bir e-posta kullanıcının gelen kutusuna geldiğinde, genellik ekranın önceki panelinde gösterilir. Eğer bu gibi bir e-posta gizli HTML komutu içeriyorsa, e-posta müşterisi www.website.com sunucusuna parametre olarak:

<img src=”http://www.website.com/script.php?youraddress@somewhere.com”

border=”0” width=”0” height=”0”></img>

kullanıcı e-posta adresi ile birlikte bir istek gönderecektir. Bu web sitesine sahip olan spammer youraddress@somewhere.com gelen kutusuna iletildiğini ve kullanıcının içeriği görüntülediğini bilecektir. Bu belirti kullanıcıyı daha fazla spam ile bombalamak için yeterlidir.

3.7.15. OCR metin tanıması

Birçok spam mesajı grafik imaj olarak gelir. Birçok anti spam sistemleri grafik görüntü üzerindeki metinleri analiz edemezler. Bu yüzden spam e-postayı tanımlayamazlar. OCR (Optic character recognation) tekniği grafik görüntü üzerinde olsa bile metini okur.

3.7.16. Metin manipülasyonu tespiti

Birçok spam mesajı anti-spam araçlarının metinsel olarak içeriği analiz etmesini zorlaştırmak için değişik hileler kullanır. Metin manipülasyonu bir metottur ya da görsel olarak benzer karakterlere sahip spam metninde kesin karakterlerin yerini değiştirmek, bütün bir kelime olarak analiz etmesini zorlaştırmak için karakterleri ayırmak, işitsel olarak benzer sembolleri ya da kelimelerin bölümlerini ve kelimeleri temsil edecek harfleri ve daha fazlasını kullanmaktır. Bazı örnekler:

- PORNO yerine P0RN0 (büyük O harfi yerine 0 rakamı)
- Warez yerine \/\/arez (W harfini oluşturmak için kullanılan \/\/)
- VIAGRA yerine V.I.A.G.R.A
- “for you” yerine 4u
- Vs.

3.7.17. URL sınıflandırması

Görsel olarak herhangi bir spam mesajı spamcılar için büyük bir eylem çağırısı olacağı için bir URL içermektedir. Ayrıca “daha fazla bilgi için buraya tıkla” gibi görünür linkler, ya da bir görüntü linki, birçok e-postalar sadece e-posta görüntülendiğinde görünür olan görüntüler metni ve reklamları gibi dinamik olarak yüklenen içerikleri bulundurmaktadır.

Bilinen sınıflandırılmış URL'lerin veritabanına karşı bütün bu URL'leri kontrol etme süpriz bir şekilde doğru sonuçlar vermektedir. Daha da fazlası, nerdeyse 0 oranda yanlış pozitif oranı ile ulaşır. Hatta bir pornografik siteye ait olan URL ile birlikte gelen spam olmayan bir e-posta gelse bile, organizasyonlar büyük bir ihtimalle engellemek isteyeceklerdir.

3.7.18. Anti-relay

Anti-relay sistemleri e-posta sunucuların spamcılar tarafından istenmeyen e-postaların yayınlanması için kullanılmasından korumayı amaçlamaktadır.

4. SPAM FİLTRELEMEDE KULLANILAN YÖNTEMLER

Bu bölümde spam filtrelemede sıklıkla uygulanan yapay sinir ağları ve kümeleme analizine yer verilecektir.

4.1. Yapay Sinir Ağları

4.1.1. YSA'nın tanımı ve tarihçesi

Yapay Sinir Ağları (YSA), beyindeki sinirlerin çalışmasını taklit ederek sistemlere öğrenme, genelleme yapma, hatırlama gibi yetenekler kazandırmayı amaçlayan bilgi işleme sistemidir.

Beyinin üstün özellikleri, bilim adamlarını üzerinde çalışmaya zorlamış ve beynin nörofiziksel yapısından esinlenerek matematiksel modeli çıkarılmaya çalışılmıştır. Beynin bütün davranışlarını modelleyebilmek için fiziksel bileşenlerinin doğru olarak modellenmesi gerektiği düşüncesi ile çeşitli yapay hücre ve ağ modelleri geliştirilmiştir. Böylece, YSA denen günümüz bilgisayarlarının algoritmik hesaplama yöntemlerinden farklı bir bilim alanı ortaya çıkmıştır.

Genel anlamda YSA, beynin bir işlevini yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanabilir. Bir YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasında oluşur. YSA'lar öğrenme algoritmaları ile öğrenme sürecinden geçtikten sonra, bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip olurlar. YSA'ları yapılarına göre farklı öğrenme yaklaşımları kullanırlar.

Dr. Robert HECHT-NIELSEN'e göre

"Yapay sinir ağları dışarıdan gelen girdilere dinamik olarak yanıt oluşturma yoluyla bilgi işleyen, birbiriyle bağlantılı basit elemanlardan oluşan bilgi işlem sistemidir."

Diğer bir tanım ise YSA alanında çok tanınan Teuvo KOHONEN'e ait bir tanımdır;

" Yapay sinir ağları paralel olarak bağlantılı ve çok sayıdaki basit elemanın, gerçek dünyanın nesneleriyle biyolojik sinir sisteminin benzeri yolla etkileşim kuran olan, hiyararşik bir organizasyonudur" [23].

Yapay sinir ağlarının dayandığı ilk hesaplama modelinin temelleri 1940'ların başında araştırmalarına başlayan W.S. McCulloch ve W.A. Pitts'in, 1943 yılında yayınladıkları bir makaleyle atılmış olmuştur. Daha sonra 1954 yılında B.G. Farley ve W.A. Clark tarafından bir ağ içerisinde uyarılara tepki veren, uyarılara adapte olabilen model oluşturulmuştur. 1960 yılı ise ilk YSA uygulamalarının ortaya çıkış yılıdır. 1963 yılında basit modellerin ilk eksiklikleri fark edilmiş, ancak başarılı sonuçların alınması 1970 ve 1980'lerde termodinamikteki teorik yapıların doğrusal olmayan ağların geliştirilmesinde kullanılmasına kadar gecikmiştir. 1985 yapay sinir ağlarının oldukça tanındığı, yoğun araştırmaların başladığı yıl olmuştur [24].

4.1.2. Biyolojik sinir sistemi

Biyolojik sinir sistemi, merkezde sürekli olarak bilgiyi alan, yorumlayan ve uygun bir karar üreten beynin bulunduğu üç katmanlı bir sistem olarak açıklanır. Bunlar; çevreden gelen girdileri elektriksel sinyallere dönüştürerek beyine ileten alıcı sinirler (receptor), beynin ürettiği elektriksel sinyalleri çıktı olarak uygun tepkilere dönüştüren tepki sinirleri ile alıcı ve tepki sinirleri arasında ileri ve geri besleme yaparak uygun tepkiler üreten merkezi sinir ağı.

Şekil 4.1. Biyolojik sinir sisteminin yapısı.

Sinir Hücresi (Nöron)

Sinir hücreleri, sinir sisteminin temel işlem elemanıdır. Birbiriyle bağlantılı iki nöronun axon, dentrite, synapse ve soma olma üzere dört önemli bölümü bulunmaktadır.

- Dendritler
- Hücre Gövdesi (Soma)
- Axonlar
- Synapse

Dentritler

Nöronun ağaç köküne benzeyen, görevi hücreye girdilerin sağlanması olan uzantılardır.

Hücre Gövdesi (Soma)

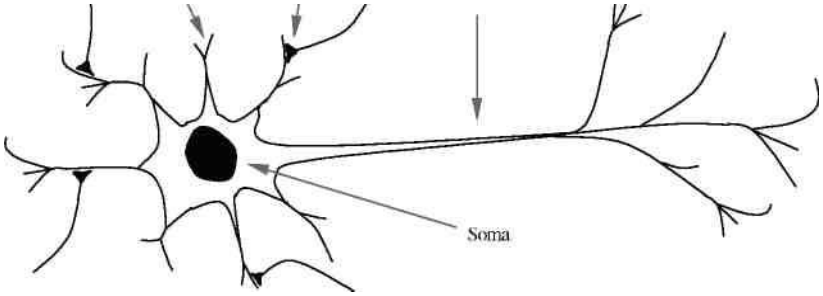
Bir nöronun gövdesine soma adı verilir. Soma nucleus adı verilen hücre çekirdeğini içermektedir. Hücrenin yaşamasını sağlayan işlevleri görür. Synapslar aracılığıyla dentritelere geçirilen iletiler birleşerek axon üzerinde elektriksel bir çıktı oluştururlar. Bu çıktının olup olmayacağı veya çıktının elektriksel olarak yoğunluğu, synapsların etkileri sonucu hücreye gelen tüm girdilerin, toplam değeri tarafından belirlenmektedir. Somaya gelen girdilerin ağırlıklı toplamı axon üzerinde çıktı oluşturacak değere ulaştığında, bu değere "eşik değer" adı verilmektedir ve nöron eşik değerini geçti ya da ateşlendi olarak ifade edilmektedir. Bu şekilde girdiler nöron tarafından değerlendirilerek çıktıya dönüştürülmüş olur.

Axon

Hücre çıktısını göndermeye yarayan uzantısıdır. Bir hücrenin tek bir axon uzantısı bulunur. Ancak bu axon uzantıdan çıkan çok sayıda uzantı ve bunların ucunda synapstik bağlantılar bulunur.

Synapse

Synapslar, sinir hücrelerindeki axonlarının, diğer sinir hücreleri ve/veya onların dentriteleri üzerinde sonlanan özelleşmiş bağlantı noktalarıdır. Bu bağlantı noktalarının görevi axondaki elektriksel iletinin diğer hücrelere aktarılmasıdır. Bu bağlantı noktalarında iletiler elektro-kimyasal süreçlerle diğer hücrelere geçirilir. Synapslar bağlandıkları dentrite veya nöronda bölgesel olarak elektrik kuvvetini pozitif veya negatif yönde etkileyebilme yeteneğine sahiptirler. Böylelikle bir nöronun diğerini etkileyebilmesi söz konusu olmaktadır.



Şekil 4.2 Biyolojik sinir hücresi.

Bir sinir hücresinin çalışması şu şekildedir;

Sinir hücresi, diğer sinir hücrelerinden gelen uyarıları (elektriksel sinyaller) snapsları üzerinden dentritlerine alır. Bu sırada gelen sinyaller snapslar tarafından güçlendirilir ya da zayıflatılır. Dentritler sinyalleri hücre gövdesine iletirler. Hücre gövdesi gelen sinyalleri birbirlerini kuvvetlendirme ve zayıflatma etkilerine göre işler. Eğer sonuçta sinyaller birbirlerini yeteri kadar kuvvetlendirerek bir eşik değerini aşabilirlerse, aksona sinyal gönderilir ve sinir aktif hale getirilir. Aksi halde, aksona sinyal gönderilmez ve sinir pasif durumda kalır.

4.1.3. YSA'nın uygulama alanları ve üstünlükleri

Son yıllarda YSA, özellikle günümüze kadar çözümü güç ve karmaşık olan ya da ekonomik olmayan çok farklı alanlardaki problemlerin çözümüne uygulanmış ve genellikle başarılı sonuçlar alınabilmiştir. YSA aşağıdaki özellikleri gösteren alanlarda kullanıma uygun bir araçtır:

- Çok değişkenli problem uzayı,
- Probleme ilişkin değişkenler arasında karmaşık etkileşim,
- Çözüm uzayının bulunmaması, tek bir çözümün olması veya çok sayıda çözüm bulunması.

YSA insan beyninin fonksiyonel özelliklerine benzer şekilde aşağıdaki konularda başarılı bir şekilde uygulanmaktadır.

- Öğrenme
- İlişkilendirme

- Sınıflandırma
- Genelleme
- Tahmin
- Özellik Belirleme
- Optimizasyon

YSA çok farklı alanlara uygulanabildiğinden bütün uygulama alanlarını burada sıralamak zor olmakla birlikte genel olarak; finans, arıza analizi ve tespiti, tıp, savunma sanayi, haberleşme, otomasyon, log analizi gibi konularda kullanılmaktadır.

Doğrusal Olmama

YSA' nın temel işlem elemanı olan hücre doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen YSA da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır Yapay sinir ağları özellikle doğrusal olmayan sistemlerde öngörüler açısından istatistiksel tekniklere göre daha kolaylık sağlayan bir özelliğe sahiptir. Bundan dolayı başta işletmecilik ve finans olmak üzere birçok değişik alanlarda kullanım imkânı bulur.

Paralellik

Alışılmış bilgi işlem yöntemlerinin çoğu seri işlemlerden oluşmaktadır. Bu da hız ve güvenilirlik sorunlarını beraberinde getirmektedir. Seri bir işlem gerçeklenirken herhangi bir birimin yavaş oluşu tüm sistemi doğruca yavaşlatırken, paralel bir sistemde yavaş bir birimin etkisi çok azdır.

Gerçeklenme Kolaylığı

Yapay sinir ağlarında basit işlemler gerçekleyen türden hücrelerden oluşması ve bağlantıların düzgün olması ağların gerçeklenmesi açısından büyük kolaylık olmasını sağlamaktadır.

Yerel Bilgi İşleme

YSA'da her bir işlem birimi, çözülecek problemin tümü ile ilgilenmek yerine, sadece problemin gerekli parçası ile ilgilenmektedir ve problemin bir parçası işlemektedir. Hücrelerin çok basit işlem yapmalarına rağmen, sağlanan görev paylaşımı sayesinde, çok karmaşık problemler çözülebilmektedir.

Hata Toleransı

Seri bilgi işlem yapan bir sistemde herhangi bir birimin hatalı çalışması, hatta bozulmuş olması tüm sistemin hatalı çalışmasına veya bozulmasına sebep olacaktır. Paralel bilgi işleme yapan bir sistemde ise, sistemin ayrı ayrı işlem elemanlarında meydana gelecek olan hatalı çalışma veya hasar, sistemin performansında keskin bir düşüşe yol açmadan, performansın sadece hata birimlerinin bir oranınca düşmesine sebep olur. YSA, çok sayıda hücrenin çeşitli şekillerde bağlanmasından oluştuğundan paralel dağılmış bir yapıya sahiptir ve ağın sahip olduğu bilgi, ağdaki bütün bağlantılar üzerine dağılmış durumdadır. Bu nedenle, eğitilmiş bir YSA'nın bazı bağlantılarının hatta bazı hücrelerinin etkisiz hale gelmesi, ağın doğru bilgi üretmesini önemli ölçüde etkilemez. Bu nedenle, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir.

Öğrenebilirlik

Alışılagelmiş veri işleme yöntemlerinin çoğu programlama yolu ile hesaplamaya dayanmaktadır. Bu yöntemler ile tam tanımlı olmayan bu problemin çözümü yapılamaz. Bunun yanında, herhangi bir problemin çözümü için probleme yönelik bir algoritmanın geliştirilmesi gerekmektedir. Yapay sinir ağları problemleri verilen örneklerle çözer. Çözülecek problemler için yapı aynıdır. YSA'nın arzu edilen davranışı gösterebilmesi için amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiğini ifade eder. YSA' nın karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Bu nedenle YSA, istenen

davranışı gösterecek şekilde ilgilendiği problemden aldığı eğitim örneklerini kullanarak problemi öğrenmelidir.

Genelleme

YSA, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de arzu edilen tepkiyi üretebilir. Örneğin, karakter tanıma amacıyla eğitilmiş bir YSA, bozuk karakter girişlerinde de doğru karakterleri verebilir ya da bir sistemin eğitilmiş YSA modeli, eğitim sürecinde verilmeyen giriş sinyalleri için de sistemle aynı davranışı gösterebilir.

Uyarlanabilirlik

YSA, ilgilendiği problemdeki değişikliklere göre ağırlıklarını ayarlar. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişimlere göre tekrar eğitilebilir, değişimler devamlı ise gerçek zamanda da eğitime devam edilebilir. Bu özelliği ile YSA, uyarlamalı örnek tanıma, sinyal işleme, sistem tanılama ve denetim gibi alanlarda etkin olarak kullanılır.

Donanım ve Hız

YSA, paralel yapısı nedeniyle büyük ölçekli entegre devre (VLSI) teknolojisi ile gerçeklenebilir. Bu özellik, YSA'nın hızlı bilgi işleme yeteneğini artırır ve gerçek zamanlı uygulamalarda arzu edilir.

Analiz ve Tasarım Kolaylığı

YSA'nın temel işlem elemanı olan hücrenin yapısı ve modeli bütün YSA yapılarında benzerlik göstermektedir. Bu nedenle, farklı uygulama alanlarında kullanılan YSA'ları benzer öğrenme algoritmalarını ve teorilerini paylaşabilirler. Bu özellik, problemlerin YSA ile çözümünde önemli bir kolaylık getirecektir.

4.1.4. YSA'nın çalışması

YSA hesaplamalarında istenilen dönüşüm için, adım adım yürütülen bir yöntem gerekmez. YSA ilişkilendirmeyi yapan iç kuralları kendi üretir ve bu kuralları, bunların sonuçlarını örneklerle karşılaştırarak düzenler. Deneme ve yanılma ile ağ

kendi kendine işi nasıl yapması gerektiğini öğretir. YSA'larda bilgi saklama, verilen eğitim özelliğini kullanarak eğitim örnekleri ile yapılır. Sinirsel hesaplama, algoritmik programlamaya bir seçenek oluşturan, temel olarak yeni ve farklı bir bilgi işleme olayıdır. Uygulama imkânının olduğu her yerde, tamamen yeni bilgi işleme yetenekleri geliştirebilir. Bu sayede de geliştirme harcamaları ile geliştirme süresi büyük ölçüde azalır.

Bir yapay sinir ağı girdi setindeki değişiklikleri değerlendirerek öğrenir ve buna bir çıktı üretir. Öğrenme işlemi benzer girdi setleri için aynı çıktıyı üretecek bir öğrenme algoritması ile gerçekleşir. Öğrenme setindeki girdilerin istatistiksel özelliklerinin çıkarılarak benzer girdilerin gruplandırılmasını sağlayan bir işlemdir.

Sinir yapılarına benzetilerek bulunan ağların eğitimi de, normal bir canlının eğitimine benzemektedir. Sınıfların birbirinden ayrılması işlemi (dolayısıyla kendini geliştirmesi), öğrenme algoritması tarafından örnek kümeden alınan bilginin adım adım işlenmesi ile gerçeklenir. YSA kullanılarak makinelere öğrenme genelleme yapma, sınıflandırma, tahmin yapma ve algılama gibi yetenekler kazandırılmıştır.

4.1.5. YSA’nın eğitimi ve testi

Geleneksel bilgisayar uygulamalarının geliştirilmesinde karşılaşılan durum, bilgisayarın belli bilgisayar dilleri aracılığıyla ve kesin yazım algoritmalarına uygun ifadelerle programlanmasıdır. Bu oldukça zaman alan, uyumluluk konusunda zayıf, teknik personel gerektiren, çoğu zaman pahalı olan bir süreçtir. Oysa biyolojik temele dayalı yapay zekâ teknolojilerinden biri olan yapay sinir ağlarının geliştirilmesinde programlama, yerini büyük ölçüde "eğitime" bırakmaktadır. Proses elemanlarının bağlantı ağırlık değerlerinin belirlenmesi işlemine “ağın eğitilmesi” denir. Yapay sinir ağının eğitilmesinde kullanılan girdi ve çıktı dizileri çiftinden oluşan verilerin tümüne "eğitim seti" adı verilir [25].

Yapay sinir ağı öğrenme sürecinde, gerçek hayattaki problem alanına ilişkin veri ve sonuçlardan, bir başka deyişle örneklerden yararlanır. Gerçek hayattaki problem alanına ilişkin değişkenler yapay sinir ağının girdi dizisini, bu değişkenlerle elde

edilmiş gerçek hayata ilişkin sonuçlar ise yapay sinir ağının ulaşması gereken hedef çıktıların dizisini oluşturur.

Öğrenme süresinde, seçilen öğrenme yaklaşıma göre ağırlıklar değiştirilir. Ağırlık değişimi, öğrenmeyi ifade eder. YSA'da ağırlık değişimi yoksa öğrenme işlemi de durmuştur. Başlangıçta bu ağırlık değerleri rastgele atanır. YSA'lar kendilerine örnekler gösterildikçe, bu ağırlık değerlerini değiştirirler. Amaç, ağa gösterilen örnekler için doğru çıktıları üretecek ağırlık değerlerini bulmaktır. Ağın doğru ağırlık değerlerine ulaşması örneklerin temsil ettiği olay hakkında, genellemeler yapabilme yeteneğine kavuşması demektir. Bu genelleştirme özelliğine kavuşması işlemine, "ağın öğrenmesi" denir.

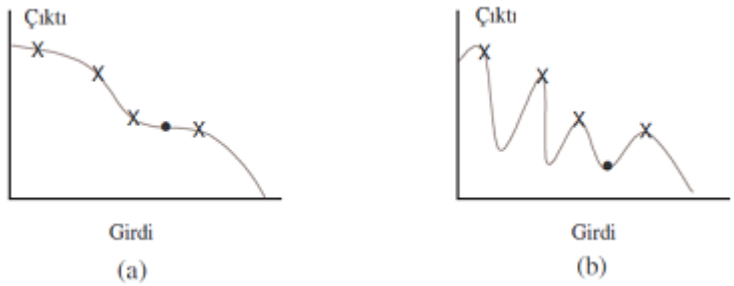
Yapay sinir ağının öğrenme sürecinde temel olarak üç adım bulunmaktadır.

- Çıktıları hesaplamak,
- Çıktıları hedef çıktılarla karşılaştırmak ve hatayı hesaplanmak,
- Ağırlıkları değiştirerek süreci tekrarlamak.

Eğitim süreci sonucunda YSA'da hesaplanan hatanın kabul edilebilir bir hata oranına inmesi beklenir. Ancak hata kareleri ortalamasının düşmesi her zaman için YSA'nın genellemeye ulaştığını göstermez. YSA'nın gerçek amacı girdi-çıktı örnekleri için genellemeye ulaşmaktadır.

Genelleme, yapay sinir ağının eğitimde kullanılmamış ancak aynı evrenden gelen girdi-çıktı örneklerini ağın doğru bir şekilde sınıflandırabilme yeteneğidir. İstatistiksel açıdan genelleme bir uygun eğrinin bulunması (curve-fitting) veya doğrusal olmayan ara değer atama işi (interpolation) olarak görülebilir. Şekil 4.3(a) 'da genellemenin nasıl gerçekleştiği görülmektedir. Şekilde (x) ile görülen noktalar eğitim verileridir. Bunların arasında kalan eğri ise ağ tarafından oluşturulmaktadır. Bu eğri üzerindeki farklı bir girdi değeri için üretilen doğru çıktı değeri, ağın iyi bir genelleme yaptığını gösterir. Ancak ağ gereğinden fazla girdi-çıktı ilişkisini öğrendiğinde, ağverileri "ezberlemektedir" (memorization). Bu durum genellikle gereğinden fazla gizli katman kullanıldığında verilerin synaptic bağlantılar üzerinde

saklanmasından veya gereğinden fazla veri kullanılarak eğitilmesinden (overtraining) kaynaklanmaktadır. Ezberleme, genellemenin iyi gerçekleşmediğini ve girdi-çıktı eğrisinin düzgün olmadığını gösterir (Şekil 4.3 (b)). Verilerin ezberlenmiş olması yapay sinir ağı için istenmeyen bir durum olup, verileri ezberleyen ağa ait eğitim hatası oldukça düşme, test verilerinde ise hata artma eğilimi gösterir. Bundan dolayı birçok yapay sinir ağı yazılımı ağın eğitim ve test verilerine ait hataları grafik olarak göstermektedir. Verileri ezberleyen ağ gerçek hayattaki örüntüyü iyi temsil edemeyeceği için kullanılamaz. Şekil 4.4 (a) 'da ağ verileri ezberlediği için eğitim hatası azalma, test hatası ise artma eğilimi göstermektedir. Şekil 4.4 (b) 'de ise ağ kabul edilebilir bir genellemeye ulaşmıştır.



Şekil 4.3. Genelleme ve ezberleme

Şekil 4.4. Verileri ezberleyen (a) İyi genellemeye ulaşan (b) ağlardaki hata eğrileri

En uygun öğrenme seviyesi, öğrenme fonksiyonunun önceden amaçlanan bir değere ulaşması ile sağlanamayabilir. Uygulamalarda eğitim süreci boyunca performans fonksiyonunun izlenmesi ile birlikte sık sık genelleme testlerinin gerçekleştirilmesi yolu ile en uygun öğrenme seviyesi elde edilebilir. Eğer en uygun öğrenme seviyesine, performans fonksiyonunun öngörülerinden önce ulaşılmış ise eğitim süresi daha erken dönemlerde de sona erdirilebilir.

YSA sistemlerinin problemi öğrenme başarısı, gerçekleştirilen testlerle sınanmalıdır. YSA geliştirme sürecinde veriler ikiye ayrılır; bir bölümü ağın eğitilmesi için kullanılır ve eğitim seti adını alır, diğer bölümü ise ağın eğitim verileri dışındaki performansını ölçmede kullanılır ve "test seti" olarak adlandırılır.

Eğitim ve test setleriyle ilgili temel sorun, yeterli eğitim ve test verisinin miktarının ne olduğudur. Sınırsız sayıda verinin bulunabildiği durumlarda, yapay sinir ağı mümkün olan en çok veriyle eğitilmelidir. Eğitim verisinin yeterli olup olmadığı konusunda emin olmanın yolu, eğitim verisinin miktarının arttırılmasının, ağın performansında bir değişiklik yaratmadığını takip etmektir. Ancak bunun mümkün olmadığı durumlarda YSA'nın eğitim ve test verileri üzerindeki performansının yakın olması da verilerin sayıca yeterli olduğuna ilişkin bir gösterge olarak kabul edilebilir. Bununla birlikte eğitim setinin içermesi gereken veri miktarı değişik yapay sinir ağı modellerine göre ve özellikle problemin gösterdiği karmaşıklığa göre farklılık gösterebilmektedir.

Test işlemi için, eğitim setinde kullanılmayan verilerden oluşan test seti kullanılır. Test setindeki girdiler YSA modeline verilir ve YSA'nın çıktı değeri ile istenilen çıktı değeri karşılaştırılır. Amaç, YSA modelinin yeterli bir genelleme yapıp yapamadığını görmektir. Eğitim ve test aşamalarında istenilen başarı elde edilirse YSA modeli kullanılabilir. Eğitim ve test ile çapraz geçerlilik (cross validation) setinin %25 ile %90 arasında değişen miktarı eğitim seti olarak seçilir. Geri kalan kısım ise test seti olarak ayrılır. Çapraz geçerlilik tekniğinde ise, YSA'nın eğitilmesinde ve test edilmesinde tüm veri seti kullanılır. Bu yaklaşımda, tüm veri seti k adet örtüşmeyen kümeye ayrılır ve k farklı YSA elde edilir. Her YSA'nın testinde farklı bir küme kullanılmak üzere, eğitim işlemi geri kalan k-1 adet küme ile gerçekleştirilir. Uygulama kullanılacak YSA ise, tüm veri seti kullanılarak eğitilir. Bu YSA' nın performansı, k farklı YSA'nın test sonuçlarının ortalaması ile ölçülür.

4.1.6. YSA'nın yapısı

Sinir hücreleri bir grup halinde işlev gördüklerinde ağ (network) olarak adlandırılırlar ve böyle bir grupta binlerce nöron bulunur. Yapay nöronların birbirleriyle bağlantılar aracılığıyla bir araya gelmeleri yapay sinir ağını oluşturmaktadır.

Yapay sinir ağıyla aslında biyolojik sinir ağının bir modeli oluşturulmak istenmektedir. Nöronların aynı doğrultu üzerinde bir araya gelmeleriyle katmanlar oluşmaktadır. Katmanların değişik şekilde bir birleriyle bağlanmaları değişik ağ mimarilerini doğurur. YSA lar üç katmadan oluşur. Bu katmanlar sırasıyla;

- Girdi katmanı,
- Ara katman,
- Çıktı katmanıdır.

Girdi katmanı

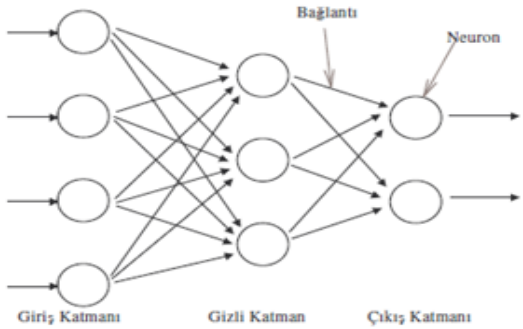
Bu katmandaki proses elemanları dış dünyadan bilgileri alarak ara katmanlara transfer ederler. Bazı ağlarda girdi katmanında herhangi bir bilgi işleme olmaz.

Ara katman (Gizli katman)

Girdi katmanından gelen bilgiler işlenerek çıktı katmanına gönderilirler. Bu bilgilerin işlenmesi ara katmanlarda gerçekleştirilir. Bir ağ içinde birden fazla ara katman olabilir.

Çıktı katmanı

Bu katmandaki proses elemanları ara katmandan gelen bilgileri işleyerek ağın girdi katmanından sunulan girdi seti için üretmesi gereken çıktıyı üretirler. Üretilen çıktı dış dünyaya gönderilir.



Şekil 4.5. YSA modeli.

4.1.7. Yapay sinir hücresi

Biyolojik sinir ağlarında olduğu gibi yapay sinir ağlarında da temel unsur, yapay sinir hücresidir. Yapay sinir hücresi, YSA' nın çalışmasına esas teşkil eden en küçük ve temel bilgi işleme birimidir Ağ içinde yer alan tüm nöronlar bir veya birden fazla girdi alırlar ve tek bir çıktı verirler. Bu çıktı yapay sinir ağının dışına verilen çıktılar olabileceği gibi başka nöronlara girdi olarak da kullanılabilirler. Geliştirilen hücre modellerinde bazı farklılıklar olmakla birlikte genel özellikleri ile bir yapay hücre modeli 5 bileşenden oluşmaktadır. Bunlar;

- Girdiler
- Ağırlıklar
- Birleştirme Fonksiyonu
- Aktivasyon Fonksiyonu
- Çıktı

Girdiler

Girdiler, diğer hücrelerden ya da dış ortamlardan hücreye giren bilgilerdir.

Ağırlıklar

Bilgiler, bağlantılar üzerindeki ağırlıklar üzerinden hücreye girer ve ağırlıklar, ilgili girişin hücre üzerindeki etkisini belirler. Ağırlıklar bir nöronda girdi olarak kullanılacak değerlerin göreceli kuvvetini (matematiksel katsayısını) gösterir. YSA içinde girdilerin nöronlar arasında iletimini sağlayan tüm bağlantıların farklı ağırlık

değerleri bulunmaktadır. Böylelikle ağırlıklar her işlem elemanının her girdisi üzerinde etki yapmaktadır.

Birleştirme fonksiyonu

Birleştirme fonksiyonu, bir hücreye gelen net girdiyi hesaplayan bir fonksiyondur ve genellikle net girdi, girişlerin ilgili ağırlıkla çarpımlarının toplamıdır. Birleştirme fonksiyonu, ağ yapısına göre maksimum alan, minimum alan ya da çarpım fonksiyonu olabilir.

$$y = F(v) = \quad x_i w_i + \emptyset$$

w: Hücrenin ağırlıklar matrisini

x: Hücrenin giriş vektörünü

v: Hücrenin net girişini

y: Hücre çıkışını

Aktivasyon fonksiyonu

Transfer fonksiyonu olarak da geçen aktivasyon fonksiyonu, birleştirme fonksiyonundan elde edilen net girdiyi bir işlemden geçirerek hücre çıktısını belirleyen ve genellikle doğrusal olmayan bir fonksiyondur. Hücre modellerinde, hücrenin gerçekleştireceği işleve göre çeşitli tipte aktivasyon fonksiyonları kullanılabilir. Aktivasyon fonksiyonları sabit parametreli ya da uyarlanabilir parametreli seçilebilir. En uygun aktivasyon fonksiyonu tasarımcının denemeleri sonucunda belli olur. Aktivasyon fonksiyonunun seçimi büyük ölçüde yapay sinir ağının verilerine ve ağın neyi öğrenmesinin istendiğine bağlıdır. Geçiş fonksiyonları içinde en çok kullanılanı sigmoid ve hiperbolik tanjant fonksiyonlarıdır. Örneğin eğer ağın bir modelin ortalama davranışını öğrenmesi isteniyorsa sigmoid fonksiyon, ortalamadan sapmanın öğrenilmesi isteniyorsa hiperbolik tanjant fonksiyon kullanılması önerilmektedir.

Aktivasyon fonksiyonları bir YSA'da nöronun çıkış genliğini, istenilen değerler arasında sınırlar. Bu değerler genellikle [0,1] veya [-1,1] arasındadır. YSA'da

kullanılacak aktivasyon fonksiyonlarının türevi alınabilir olması ve süreklilik arz etmesi gereklidir. Lineer veya doğrusal olmayan transfer fonksiyonlarının kullanılması YSA'nın karmaşık ve çok farklı problemlere uygulanmasını sağlamıştır. Aşağıda, hücre modellerinde yaygın olarak kullanılan çeşitli aktivasyon fonksiyonları tanıtılmıştır.

Lineer Fonksiyon

Doğrusal bir problemi çözmek amacıyla kullanılan doğrusal hücre ve YSA'da ya da genellikle katmanlı YSA'nın çıkış katmanında kullanılan doğrusal fonksiyon, hücrenin net girdisini doğrudan hücre çıkışı olarak verir. Doğrusal aktivasyon fonksiyonu matematiksel olarak y=Av şeklinde tanımlanabilir. "A" sabit bir katsayıdır. YSA'nın çıkış katmanında kullanılan doğrusal fonksiyon şekilde verilmiştir.

$$F\ x\ = a * x$$

Eşik Fonksiyonu

Lineer fonksiyon (-t, +t) sınırları arasında kısıtlandığında Şekil 4.6.b'deki rampa eşik fonksiyonu olur ve denklemi;

$$F\ x\ = \begin{matrix} +r : Eğer\ x \geq r\ ise \\ x : Eğer\ \ x\ < r\ ise\ \ yani - r < x < r \\ -r : Eğer\ x \geq -r\ ise \end{matrix}$$

Eğer eşik fonksiyonu bir giriş işaretine bağlı ise yaydığı +t giriş toplamı pozitif, bağlı değilse eşik basamak fonksiyonu |-d| olarak adlandırılır. Şekil 4.6.c, basamak eşik fonksiyonunu gösterir ve denklemi

$$F\ x\ = \begin{matrix} +r\ :\ Eğer\ x > 0\ ise \\ -\delta\ :\ Diğer\ Durumlar \end{matrix}$$

Sigmoid Aktivasyon Fonksiyonu

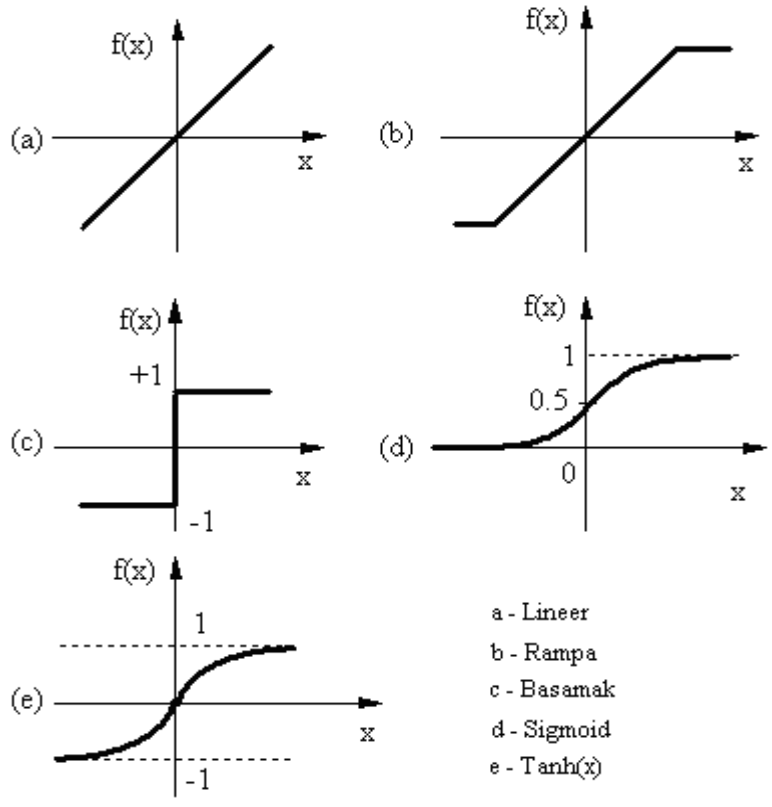
Türevi alınabilir, sürekli ve doğrusal olmayan bir fonksiyon olması nedeniyle uygulamada en çok kullanılan aktivasyon fonksiyonudur. Bu fonksiyon, girdinin her değeri için sıfır ile bir arasında bir değer üretir.

$$F\ x\ = \frac{1}{1+e^{-x}}$$

Tanjant hiperbolik fonksiyonu

Sigmoid fonksiyonunun biraz farklı şeklidir. Giriş uzayının genişletilmesinde etkili bir aktivasyon fonksiyonudur. Sigmoid fonksiyonun çıktı aralığı 0 ve 1 olurken, hiperbolik tanjant fonksiyonunun çıktısı -1 ve 1 aralığında oluşmaktadır.

$$\tanh\ x\ = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = \frac{e^{2x}-1}{e^{2x}+1}$$



Şekil 4.6. YSA'da sıkça kullanılan eşik fonksiyonları.

Çıktı

Aktivasyon fonksiyonundan geçirildikten sonra elde edilen değer, çıktı değeridir.

4.1.8. YSA'nın sınıflandırılması

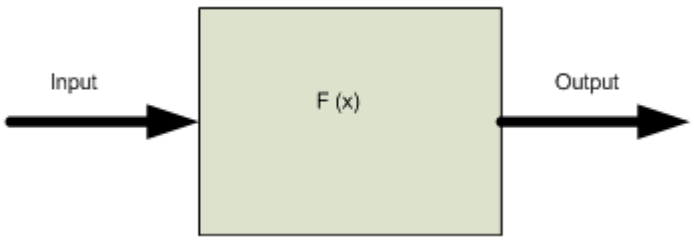
YSA'lar, genel olarak birbirleri ile bağlantılı işlemci birimlerden (sinir hücresi) oluşurlar. Her bir sinir hücresi arasındaki bağlantıların yapısı ağın yapısını belirler. İstenilen hedefe ulaşmak için bağlantıların nasıl değiştirileceği öğrenme algoritması tarafından belirlenir. Kullanılan öğrenme algoritmasına göre, hatayı sıfıra indirecek şekilde, ağın ağırlıkları değiştirilir. YSA'lar yapılarına ve öğrenme algoritmalarına göre sınıflandırılırlar.

YSA'nın yapılarına göre sınıflandırılması

YSA, yapılarına göre, ileri beslemeli (feedforward) ve geri beslemeli (feedback) ağlar olmak üzere iki şekilde sınıflandırılırlar.

İleri Beslemeli Ağlar

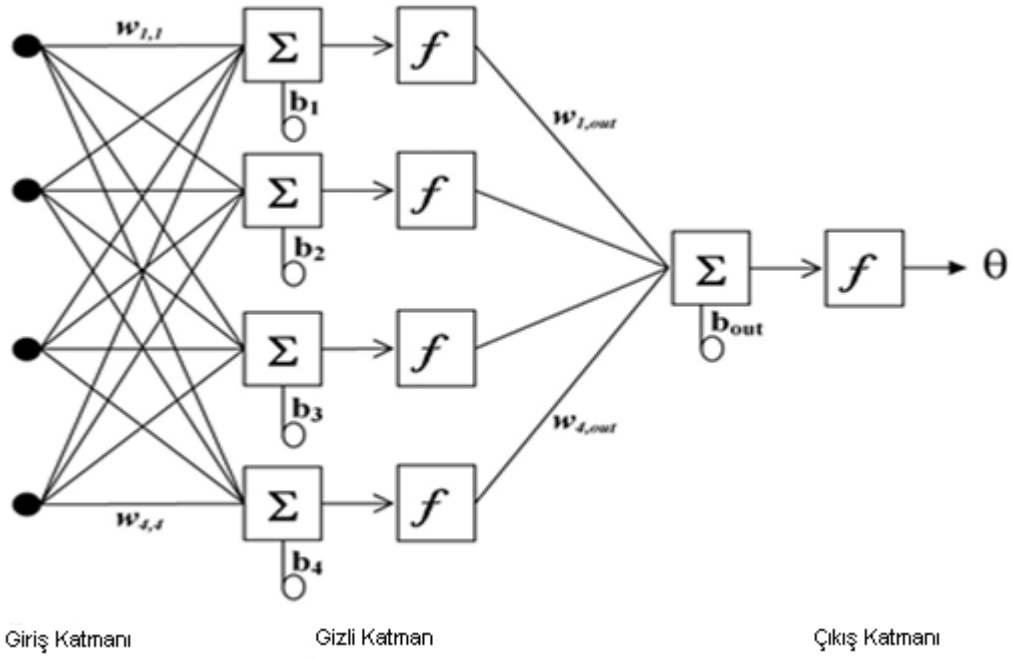
İleri beslemeli bir ağda işlemci elemanlar (İE) genellikle katmanlara ayrılmışlardır. İşaretler, giriş katmanından çıkış katmanına doğru tek yönlü bağlantılarla iletilir. İE'ler bir katmandan diğer bir katmana bağlantı kurarlarken, aynı katman içerisinde bağlantıları bulunmaz. Şekil 'de ileri beslemeli ağ için blok diyagram gösterilmiştir. İleri beslemeli ağlara örnek olarak çok katmanlı perseptron (Multi Layer Perseptron-MLP) ve LVQ (Learning Vector Quantization) ağları verilebilir.



Şekil 4.7 İleri beslemeli ağ için blok diyagram.

İleri beslemeli YSA'da, hücreler katmanlar şeklinde düzenlenir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan orta (gizli)

katmandaki hücrelere iletir. Bilgi, orta ve çıkış katmanında işlenerek ağ çıkışı belirlenir. Bu yapısı ile ileri beslemeli ağlar, doğrusal olmayan statik bir işlevi gerçekleştirir. İleri beslemeli 3 katmanlı YSA'nın, orta katmanında yeterli sayıda hücre olmak kaydıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yaklaştırabileceği gösterilmiştir. En çok bilinen geriye yayılım öğrenme algoritması, bu tip YSA'nın eğitiminde etkin olarak kullanılmakta ve bazen bu ağlara geri yayılım ağları da denmektedir. Şekil 4.8'de giriş, orta ve çıkış katmanı olmak üzere 3 katmanlı ileri beslemeli YSA yapısı verilmiştir.



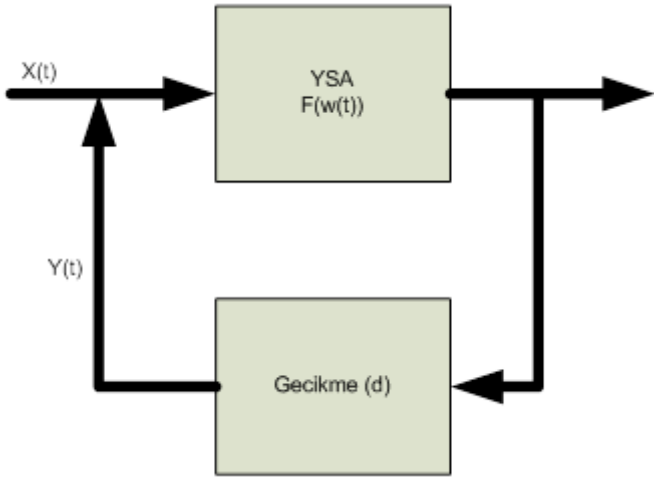
Şekil 4.8. İleri beslemeli üç katmanlı YSA.

Şekil 4.8.'de 4 girişi 1 çıkışı olan 3 katmanlı bir YSA yapısı görülmektedir. Giriş değerleri ağırlıklar ile ($w_{i,j}$) ile çarpılır ve ağırlık değerleri ile (b:bias) toplanır. Oluşan yeni değer f fonksiyonuna verilerek gizli katmanın çıkış değerleri elde edilir. Her bir gizli katmandan elde edilen değer çıkışa aynı şekilde uygulanarak çıkış fonksiyonunun değeri hesaplanmış olur.

Herhangi bir problemi çözmek amacıyla kullanılan YSA'da, katman sayısı ve orta katmandaki hücre sayısı gibi kesin değerler olmamasına karşın, nesne tanıma, sinyal işleme gibi alanlarda yaygın olarak kullanılmaktadır.

Geri Beslemeli Ağlar

Bir geri beslemeli sinir ağı, çıkış ve ara katlardaki çıkışların, giriş birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır. Böylece, girişler hem ileri yönde hem de geri yönde aktarılmış olur. Şekil 4.9'da bir geri beslemeli ağ görülmektedir. Bu çeşit sinir ağlarının dinamik hafızaları vardır ve bir andaki çıkış hem o andaki hem de önceki girişleri yansıtır. Bundan dolayı, özellikle önceden tahmin uygulamaları için uygundurlar. Geri beslemeli ağlar çeşitli tipteki zaman-serilerinin tahmininde oldukça başarı sağlamışlardır. Bu ağlara örnek olarak Hopfield, SOM (Self Organizing Map), Elman ve Jordan ağları verilebilir.



Şekil 4.9. Geri beslemeli ağ için blok diyagram.

Geri beslemeli YSA'da, en az bir hücrenin çıkışı kendisine ya da diğer hücrelere giriş olarak verilir ve genellikle geri besleme bir geciktirme elemanı üzerinden yapılır. Geri besleme, bir katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler arasında da olabilir. Bu yapısı ile geri beslemeli YSA, doğrusal olmayan dinamik bir davranış gösterir. Dolayısıyla, geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli YSA yapıları elde edilebilir.

Geriye doğru hesaplamada, ağın ürettiği çıktı değeri, ağın beklenen çıktıları ile kıyaslanır. Bunların arasındaki fark, hata olarak kabul edilir. Amaç bu hatanın düşürülmesidir. Çıktı katmanında m. proses için oluşan hata, $E_m = B_m - Ç_m$ olacaktır. Çıktı katmanında oluşan toplam hatayı bulmak için, bütün hataların toplanması gereklidir. Bazı hata değerleri negatif olacağından, toplamın sıfır olmasını önlemek amacıyla ağırlıkların kareleri hesaplanarak sonucun karekökü alınır. Toplam hata aşağıdaki formül ile bulunur.

$$Toplam\ Hata = \sum_{m=1}^{n} {E_m}^2$$

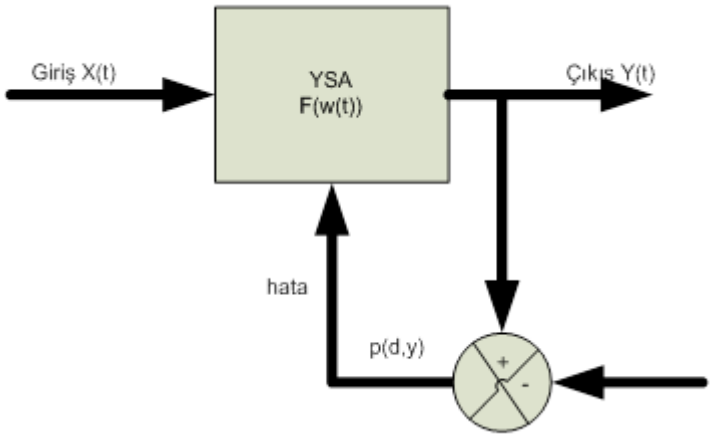
Toplam hatayı en azaltmak için, bu hatanın kendisine neden olan proses elemanlarına dağıtılması gerekmektedir. Bu da, proses elemanlarının ağırlıklarını değiştirmek demektir.

YSA'nın öğrenme algoritmalarına göre sınıflandırılması

Öğrenme; gözlem, eğitim ve hareketin doğal yapıda meydana getirdiği davranış değişikliği olarak tanımlanmaktadır. Birtakım metot ve kurallar, gözlem ve eğitime ile ağdaki ağırlıkların değiştirilmesi sağlanmalıdır. Bunun için genel olarak üç öğrenme metodundan ve bunların uygulandığı değişik öğrenme kurallarından söz edilebilir. Bu öğrenme kuralları aşağıda açıklanmaktadır.

Danışmanlı öğrenme

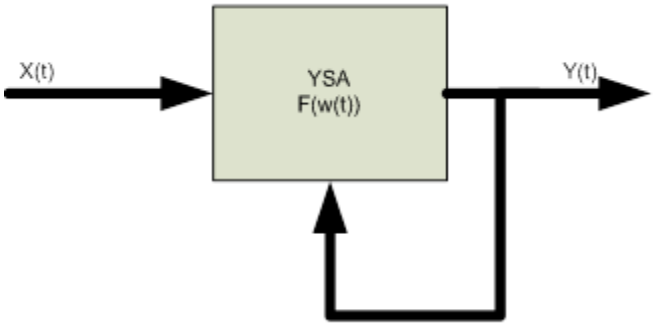
Bu tip öğrenmede, YSA'ya örnek olarak bir doğru çıkış verilir. Bu öğrenmede ağın ürettiği çıktılar ile hedef çıktılar arasındaki fark hata olarak ele alınır ve bu hata minimize edilmeye çalışılır. Bunun için de bağlantıların ağırlıkları en uygun çıkışı verecek şekilde değiştirilir. Bu sebeple danışmanlı öğrenme algoritmasının bir "öğretmene" veya "danışmana" ihtiyacı vardır. Şekil 4.10'da danışmanlı öğrenme yapısı gösterilmiştir. Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı veya geri besleme (back propagation) algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir.



Şekil 4.10. Danışmanlı öğrenme yapısı.

Danışmansız Öğrenme

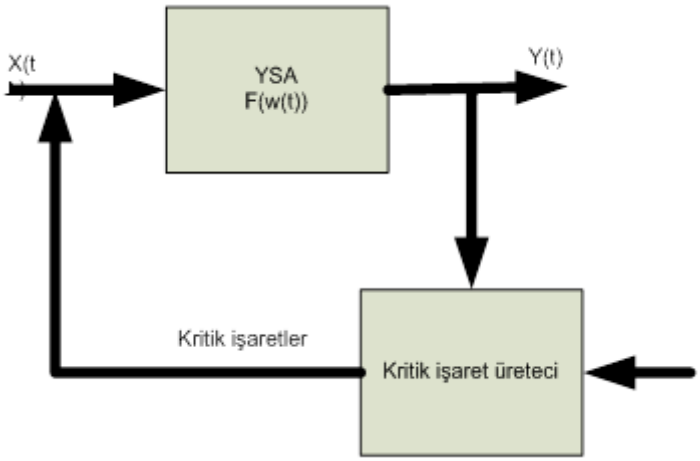
Bu tür öğrenmede ağa sadece girdiler verilir. Ağın ulaşması gereken hedef çıktılar verilmez. Girişe verilen örnekten elde edilen çıkış bilgisine göre ağ sınıflandırma kurallarını kendi kendine geliştirir. Ağ daha sonra bağlantı ağırlıklarını aynı özellikleri gösteren desenler (patterns) oluşturmak üzere ayarlar. Şekil 4.10'da danışmansız öğrenme yapısı gösterilmiştir. Grossberg tarafından geliştirilen ART (Adaptive Resonance Theory) veya Kohonen tarafından geliştirilen SOM (Self Organizing Map) öğrenme kuralı danışmansız öğrenmeye örnek olarak verilebilir.



Şekil 4.11. Danışmansız öğrenme yapısı.

Takviyeli Öğrenme

Takviyeli öğrenme algoritması, istenilen çıkışın bilinmesine gerek duymaz. Takviyeli öğrenme (reinforcement training) yöntemi danışmanlı öğrenme yöntemine benzemekle birlikte, ağa hedef çıktılar yerine, ağın çıktılarının ne ölçüde doğru olduğunu belirten bir skor veya derece bildirilir. Şekil 4.11'de takviyeli öğrenme yapısı gösterilmiştir. Optimizasyon problemlerini çözmek için Hinton ve Sejnowski'nin geliştirdiği Boltzmann kuralı veya GA (Genetik Algoritma) takviyeli öğrenmeye örnek olarak verilebilirler.



Şekil 4.12. Takviyeli öğrenme yapısı.

Uygulamaya göre öğrenme algoritmaları

Çevrim içi (on-line) öğrenme

Bu kurala göre öğrenen sistemler, gerçek zamanda çalışırken bir taraftan fonksiyonlarını yerine getirmekte, bir taraftan da öğrenmeye devam etmektedirler. ART ve Kohonen öğrenme kuralı bu sınıfta bulunan öğrenme bu öğrenme kuralına örnek olarak verilebilir.

Çevrim Dışı (Offline) Öğrenme

Bu kurala dayalı sistemler, kullanıma alınmadan önce örnekler üzerinde eğitilirler. Bu kuralı kullanan sistemler eğitildikten sonra gerçek hayatta kullanıma alındığında

artık öğrenme olmamaktadır. Delta öğrenme kuralı bu tür öğrenmeye örnek olarak verilebilir.

4.1.9. YSA'nın tasarımı

YSA'daki sinir sayısı, sinirlerin birbirine göre konumu ve sinirler arası sinyallerin akış yönleri YSA yapısını belirlemektedir. Yapısı belirlenmiş bir YSA eğitilerek YSA ile ilgili uygulamalarda kullanılır.

YSA yapıları arasında performans ve karakteristik özellikleri bakımın farklar vardır. YSA yapıları, özellikle ağın modelleme yeteneğini belirledikleri için oldukça önemlidirler. Yapay sinir ağının tasarımı aşamasında bu ağ yapıları arasından uygulamaya en elverişli olanı seçilir.

YSA uygulamasının başarısı, uygulanacak olan yaklaşımlar ve deneyimlerle yakından ilgilidir. Uygulamanın başarısında uygun metodolojiyi belirlemek büyük önem taşır. Yapay sinir ağının geliştirilmesi sürecinde ağın yapısına ve işleyişine ilişkin şu kararların verilmesi gerekir.

- Ağ mimarisinin seçilmesi ve yapı özelliklerinin belirlenmesi (katman sayısı, katmandaki nöron sayısı gibi)
- Nörondaki fonksiyonların karakteristik özelliklerinin belirlenmesi,
- Öğrenme algoritmasının seçilmesi ve parametrelerinin belirlenmesi,
- Eğitim ve test verisinin oluşturulması

Bu kararların doğru verilememesi durumunda, YSA'ları sistem karmaşıklığı artacaktır. Sitem karmaşıklığı yapısal ve toplam hesaplama karmaşıklığının bir fonksiyonudur. Toplam hesaplama karmaşıklığı ise, genellikle yapısal karmaşıklığın bir fonksiyonu olarak ortaya çıkar ve bu hesaplamanın en aza indirilmesi amaçlanır. Bu hesaplama karmaşıklığının ölçülmesinde de genellikle YSA sisteminin toplam tepki süresi veya sisteme ait bir işlemci elemanın tepki süresi değeri temel alınır. Bunun yanında kapladığı hafıza ve zaman karmaşıklığı bazı uygulamalarda hesaplanmaktadır.

Bir YSA'nın uygun parametrelerle tasarlanması durumunda YSA sürekli olarak kararlı ve istikrarlı sonuçlar üretecektir. Ayrıca sistemin tepki süresinin yeterince kısa olabilmesi için de ağ büyüklüğünün yeterince küçük olması gerekir. İhtiyaç duyulan toplam hesaplama da bu sayede sağlanmış olacaktır.

YSA ağ yapısının seçimi

YSA'nın tasarımı sürecinde ağ yapısının seçilmesi, uygulama problemine bağlı olarak seçilmelidir. Hangi problem için hangi ağın daha uygun olduğunun bilinmesi önemlidir. Kullanım amacı ve o alanda başarılı olan ağ türleri Çizelge 4.1'de verilmiştir.

Çizelge 4.1.Ağ türleri ve başarılı oldukları alanlar

Kullanım Amacı	Ağ Türü	Ağın Kullanımı
Tahmin	■ ÇKA	Ağın girdilerinden bir çıktı değerinin tahmin edilmesi
Sınıflandırma	■ LVQ ■ ART ■ Counterpropagation ■ Olasılıklı Sinir Ağları	Girdilerin hangi sınıfa ait olduklarının belirlenmesi
Veri İlişkilendirme	■ Hopfield ■ Boltzman Machine ■ Bidirectional Associative Memory	Girdilerin içindeki hatalı bilgilerin bulunması ve eksik bilgilerin tamamlanması

Uygun YSA yapısının seçimi, büyük ölçüde ağda kullanılması düşünülen öğrenme algoritmasına da bağlıdır. Ağda kullanılacak öğrenme algoritması seçildiğinde, bu algoritmanın gerektirdiği mimaride zorunlu olarak seçilmiş olacaktır. Örneğin geri yayılım algoritması ileri beslemeli ağ mimarisi gerektirir.

Bir YSA'nın karmaşıklığının azaltılmasında en etkin araç, YSA ağ yapısını değiştirmektir. Gereğinden fazla sayıda işlemci eleman içeren ağ yapılarında, daha düşük genelleme kabiliyeti ile karşılaşılır.

Öğrenme algoritmasının seçimi

YSA yapısının seçiminden sonra uygulama başarısını belirleyen en önemli faktör öğrenme algoritmasıdır. Genellikle ağ yapısı öğrenme algoritmasının seçiminde belirleyicidir. Bu nedenle seçilen ağ yapısı üzerinde kullanılabilecek öğrenme algoritmasının seçimi ağ yapısına bağlıdır. YSA'nın geliştirilmesinde kullanılacak çok sayıda öğrenme algoritması bulunmaktadır. Bunlar içinde bazı algoritmaların bazı tip uygulamalar için daha uygun olduğu bilinmektedir. Bu algoritmalar eğer uygun oldukları uygulama alanlarına göre sınıflandırılacak olursa, gruplar ve içinde yer alacak öğrenme algoritmaları aşağıdaki gibi özetlenebilir.

Çizelge 4.2.Öğrenme algoritmaları ve uygulandıkları alanlar

Uygulama Tipi	Yapay Sinir Ağı
Öngörü Tanıma	Geri yayılım
	Delta Bar Delta
	Geliştirilmiş Delta Bar Delta
	Yönlendirilmiş Rastsal Tarama
	Geri yayılım içinde Self Organizing Map
	Higher Order Neural Networks
Sınıflandırma	Learning Vektor Quantization
	Counter-Propagation
	Olasılıklı Yapay Sinir Ağları
Veri İlişkilendirme	Hopfield
	Boltmann Makinesi
	Bidirectional Associative Memory
	Spantion -temproal Pattern Recogniation
Veri Kavramlaştırma	Adaptive Resonance Network
	Self Organizing

Ara katman sayısının belirlenmesi

YSA'nın tasarımı sürecinde tasarımcının yapması gereken diğer işlemde, ağdaki katman sayısına karar vermektir. Çoğu problem için 2 veya 3 katmanlı bir ağ tatmin edici sonuçlar üretebilmektedir. Nöronların aynı doğrultu üzerinde bir araya gelmeleriyle katmanlar oluşmaktadır. Katmanların değişik şekilde bir birleriyle bağlanmaları değişik ağ yapılarını oluşturur. Girdi ve çıktı katmanlarının sayısı, problemin yapısına göre değişir [26]. Katman sayısını belirlemenin en iyi yolu, birkaç deneme yaparak en uygun yapının ve yapının ne olduğuna karar vermektir.

Nöron sayısının belirlenmesi

Ağın yapısal özelliklerinden birisi her bir katmandaki nöron sayısıdır. Katmandaki nöron sayısının tespitinde de genellikle deneme-yanılma yöntemi kullanılır. Bunun için izlenecek yol, başlangıçtaki nöron sayısını istenilen performansa ulaşıncaya kadar arttırmak veya tersi şekilde istenen performansın altına inmeden azaltmaktır. Bir katmanda kullanılacak nöron sayısı olabildiğince az olmalıdır. Nöron sayısının az olması yapay sinir ağının "genelleme" yeteneğini arttırırken, gereğinden fazla olması ağın verileri ezberlemesine neden olur. Ancak gereğinden az nöron kullanılmasının verilerdeki örüntünün ağ tarafından öğrenilememesi gibi bir sorun yaratabilir.

Nörondaki fonksiyonların da karakteristik özellikleri de YSA'nın tasarımında önemli kararlardan biridir. Nöronun geçiş fonksiyonunun seçimi büyük ölçüde YSA'nın verilerine ve ağın neyi öğrenmesinin istenildiğine bağlıdır. Geçiş fonksiyonları içinde en çok kullanılanı sigmoid ve hiperbolik tanjant fonksiyonlarıdır. Daha önce belirtildiği gibi sigmoid fonksiyonun çıktı aralığı 0 ve 1 arasında olurken, hiperbolik tanjant fonksiyonunun çıktısı -1 ve 1 aralığında oluşmaktadır. Eğer ağın bir modelin ortalama davranışını öğrenmesi isteniyorsa sigmoid fonksiyon, eğer ortalama sapmanın öğrenilmesi isteniyorsa hiperbolik tanjant fonksiyonunun kullanılması önerilmektedir.

Normalizasyon

YSA'ların en belirgin özelliklerinden olan doğrusal olmama özelliğini anlamlı kılan yaklaşım, verilerin bir normalizasyona tabii tutulmasıdır. Verilen normalizasyonu

için seçilen yöntem YSA performansını doğrudan etkileyecektir. Çünkü normalizasyon, giriş verilerinin transfer edilirken fonksiyonun aktif olan bölgesinden aktarılmasını sağlar. Veri normalizasyonu, işlemci elemanlarını verileri kümülatif toplamların oluşturacağı olumsuzlukların engellenmesini sağlar. Veri normalizasyonu, işlemci elemanlarını verileri kümülatif toplamlarla koruma eğilimleri nedeniyle zorunludur ve aşırı değerlenmiş kümülatif toplamların oluşturacağı olumsuzlukların engellenmesini sağlar. Genellikle verilerin [0,1] veya [-1,+1] aralıklarından birine ölçeklendirilmesi önerilmektedir. Ölçekleme verilerin geçerli eksen sisteminde sıkıştırılması anlamı taşıdığından veri kalitesi aşırı salınımlar içeren problemlerin YSA modellerini olumsuz yönde etkileyebilir. Bu olumsuzluk, kullanılacak öğrenme fonksiyonunu da başarısız kılabilir.

Örneğin, Bir transfer fonksiyonuna uygulanan girişler ile ağırlıkların çarpım toplamının 10 ve 100 olduğunu farz edelim. Bu toplamı bir tanjant hiperbolik fonksiyonundan geçirdiğimizde;

$$y = \tanh\ 10\ = \frac{e^{10} - e^{-10}}{e^{10} + e^{-10}} = 1.00000$$

$$y = \tanh\ 10\ = \frac{e^{100} - e^{-100}}{e^{100} + e^{-100}} = 1.00000$$

sonuçları elde edilir. Görüldüğü gibi, 10 ve 100 skalar değerlerine karşılık gelen fonksiyon sonuçları arasında fark yoktur. Bu durumda da, birbirinden oldukça farklı skalar değerler sistemde sanki aynı değerlermiş gibi ele alınacak ve hem uygulama hem de öğrenme algoritması açısından olumsuz sonuçlar ortaya çıkacaktır.

Bunun için, X veri kümesi [-1,+1] ya da [0,1] aralığına ölçeklendirilmelidir. Veri kümesinin [0 1] arasında bir ölçeklendirmeye tabi tutulabilmesi için o kümenin Xmin ve Xmax aralığı bulunur ve aşağıdaki formüle göre ölçeklendirme yapılabilir.

$$x_{yeni} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Performans fonksiyonun seçimi

Öğrenme performansını etkileyen önemli hususlardan bir de performans fonksiyonudur. İleri beslemeli ağlarda kullanılan tipik performans fonksiyonu karesel ortalama hatadır (Mean Square Error).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(t_i - td_i)^2$$

ile hesaplanır. İleri beslemeli ağlarda kullanılan tipik performans fonksiyonlarından bir diğeri de toplam karesel hatadır. (Sum Square Error)

$$SSE = \sum_{i=1}^{N}(t_i - td_i)^2$$

ile hesaplanır.

Bu ağlarda kullanılan diğer bir performans fonksiyonu da karesel ortalama hata karekökü (Root Mean Square) fonksiyonudur.

$$RMS = \overline{\frac{1}{N}\sum_{i=1}^{N}(t_i - td_i)^2}$$

4.1.10. YSA'da kullanılan temel öğrenme kuralları

Literatürde kullanılan çok sayıda öğrenme algoritması bulunmaktadır. Bu algoritmaların çoğunluğu matematik tabanlı olup ağırlıkların güncelleştirilmesi için kullanılırlar. YSA'nın mimarisine, karşılaşılan sorunun niteliğine göre farklılık gösteren bu öğrenme algoritmalarının yüzden fazla çeşidi bulunmaktadır. Bu algoritmaların birçoğu aşağıda sırlanan kurallardan esinlenerek geliştirilmiştir.

- Hebb Kuralı
- Delta Kuralı
- Kohonen Kuralı

- Hopfield Kuralı

Hebb kuralı

1949 yılında Kanadalı psikolog Donald Hebb tarafından biyolojik temele dayalı olarak geliştirilmiş olan Hebb algoritması en eski ve en ünlü öğrenme algoritmasıdır. Bu öğrenme algoritması basit bir mantığa dayanmaktadır: Eğer nöron (A) başka bir nöron' dan (B) girdi alıyorsa ve her ikisi de aktifse, (A) ve (B) arasındaki ağırlık artar. Bu düşüncenin en çok kullanılan şekli:

$$\Delta w_{jk} = \propto y_j x_k$$

$$\Delta w_{jk} = \Delta w_{jk}^{(t)} = w_{jk}^{(t+1)} - w_{jk}^{t} \; ve \; \Delta > 0$$

Bu formülde wjk nöron uk ' dan nöron uj'ya olan ağırlık , yj, uj nöronun çıktısı ve xk ise uk nöronun çıktısıdır. ∝ "öğrenme oranı" veya " öğrenme katsayısı" olarak adlandırılmaktadır ve birçok öğrenme algoritması tarafından kullanılır. Öğrenme katsayısı "0" ile "1" arasında bir değer alır ve bu değer ağın öğrenme hızını belirler. α'nın büyük değerleri için daha hızlı öğrenme, küçük değerleri için daha yavaş öğrenme gerçekleşmektedir. Ancak hızlı öğrenme ağın "genelleme" yeteneğini azaltır. Genelleme yeteneği ağın eksik ve gürültülü verilerle doğru sonuçlar üretebilmesi için oldukça önemlidir. Hebb algoritmasıyla ilgili bir diğer konuda ağın eğitimden önce ağırlıklarının 0 olması gerektiğidir.

Delta kuralı

Delta kuralı ilk olarak Widrow ve Hoff tarafından geliştirilmiş daha çok mühendislik kökenli bir algoritmadır. Bu kural, nöronun gerçek çıkışı ile istenilen çıkış değerleri arasındaki farkı azaltan, giriş bağlantılarını güçlendiren ve sürekli olarak değiştiren bir düşünceye dayanmaktadır. Delta kuralı, ortalama karesel hatayı, bağlantı ağırlık değerlerinin değiştirilmesi ile düşürme prensibine dayanır. Bu nedenle de bu algoritma en küçük kareler kuralı olarak da bilinmektedir (Least-Mean-Square Rule LMS). Hata aynı anda bir katmandan bir önceki katmanlara geri yayılarak azaltılır.

Ağın hatalarının düşürülmesi işlemi, çıkış katmanından giriş katmanına ulaşıncaya kadar devam eder.

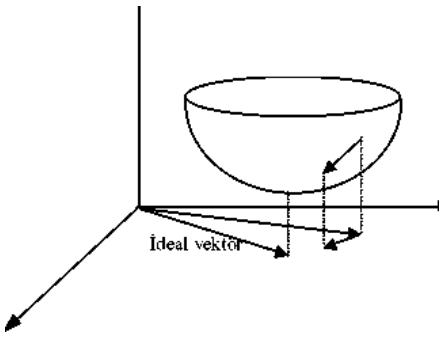
Bir nöronun çıktısının Y_j, hedeflenen çıktı D_j'den farkı (hata) δ_j, nöron j için şöyle hesaplanır:

$\delta_j = d_j - y_j$

Bu algoritma, hata karelerinin ortalamasını alarak, bu değerin en küçük olduğu çözümü bulmaya amaçlar. Ağ için hata karelerinin ortalaması ξ aşağıdaki gibi hesaplanır:

$$\xi = E\ \frac{1}{2}\ \sum_{j=1}^{N} \delta^2$$

Burada E istatistiksel olarak beklenen değeri (ortalamayı) ifade etmektedir. Delta algoritması hataların karesinin en küçük olduğu noktayı bulurken dereceli azaltma yöntemini (gradient descent) kullanmaktadır. Bu yöntemde, hata kareleri, koordinatlarını ağırlıkların oluşturduğu uzayda bir çanak oluşturmaktadır (Şekil 4.13). Delta algoritması mevcut ağırlık vektörünü bulunduğu konumdan hatanın en küçük olduğu çanağın dibine doğru ilerletir.



Şekil 4.13. Delta algoritmasında ağırlık değişimi

İşlem elemanının doğrusal geçiş fonksiyonuna sahip olduğu kabul edildiğinde;

$$y_j = \sum_k w_{jk}\, x_k$$

Hata karelerinin ortalamasının en küçük olduğu noktayı bulmak için ağırlıklar t zaman olmak üzere t=1,2,3, için aşağıdaki gibi değiştirilir:

$$w_{jk}^{(t+1)} = w_{jk}^{(t)} + \Delta w_{jk}$$

$$\Delta w_{jk} = \propto\ w_j^t x_k^t$$

Kohonen kuralı

Bu kural, biyolojik sistemlerdeki öğrenmeden esinlenerek Kohonen tarafından geliştirilmiştir. Bu kuralda nöronlar öğrenmek için yarışırlar. Kazanan nöronun ağırlıkları güncellenir. Bu kuralı "kazanan tamamını alır" olarak da bilinir. En büyük çıkışa sahip işlemci nöron kazanır. Bu nöron, komşularını uyarma ve yasaklama kapasitesine sahiptir. Kohonen kuralı, hedef çıkışa gereksinim duymaz. Bu nedenle danışmansız bir öğrenme metodudur.

Hopfield kuralı

Bu kural, zayıflatma ve kuvvetlendirme büyüklüğü dışında Hebb Kuralına benzerdir. Eğer istenilen çıkış ve girişin her ikisi aktif veya her ikisi de aktif değilse öğrenme oranı tarafından bağlantı ağırlığı arttırılır. Diğer durumlarda ise azaltılır. Birçok öğrenme algoritmasında öğrenme katsayısı oranı veya sabiti vardır. Genellikle bu terim 0 ile 1 arasında değerler almaktadır.

4.2. Kümeleme Analizi

Bu bölümde sınıflandırmada sıklıkla kullanılan, danışmansız öğrenme tekniği olan kümeleme analizine yer verilmiştir. Kümeleme analizinin temel amacı nesneleri sahip oldukları karakteristik özellikleri baz alarak gruplamaktır.

4.2.1. Kümeleme analizinin tanımı

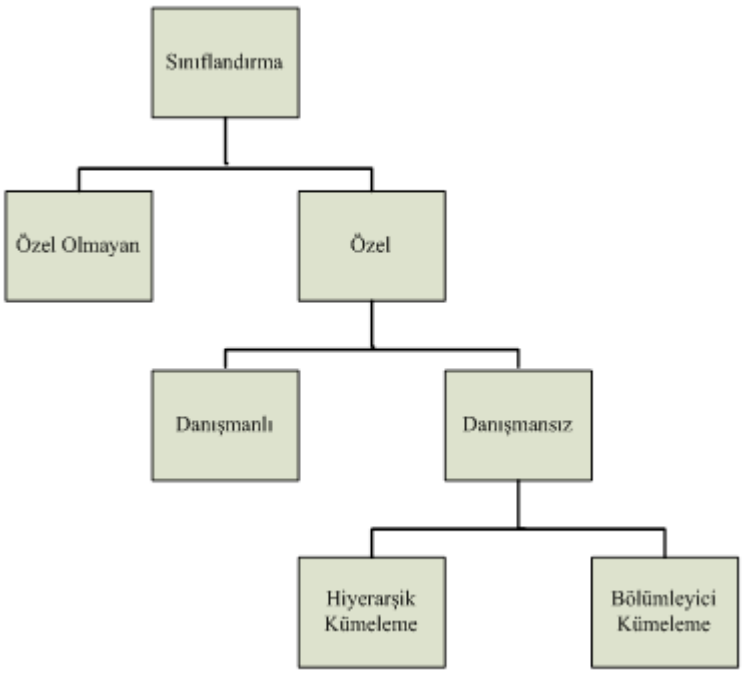
Kümeleme analizi en yaygın olarak kullanılan veri madenciliği tekniklerinden biridir. Literatürde kümeleme analizini açıklayan birçok tanım bulunmaktadır [27-32]. Kümeleme, en basit tanımıyla benzer özellik gösteren veri elemanlarının kendi aralarında gruplara ayrılmasıdır.

Kümeleme, bir dizi örüntüyü (gözlem sonuçları, veri nesneleri ve özellik vektörleri) ayrık ve homojen gruplar oluşturacak şekilde gruplandırma işlemidir. Bu işlem örüntülerin benzerlik derecelerine göre sınıflara veya kümelere ayrılmasıyla gerçekleştirilir. Kümeleme işlemi sonucunda elde edilen her başarılı (geçerli) küme içinde yer alan nesneler arasında maksimum benzerlik ve kümeleme sonucu elde edilen her kümedeki nesneler arasında ise maksimum farklılık oluşması sağlanır [27,28,30].

Karypis'in kümeleme tanımı da benzerdir: Kümeleme işlemi nesneleri aralarındaki benzerliğe bağlı olarak gruplara ayırır [28]. Böylece benzer nesneler aynı grup içinde toplanırken farklı nesneler farklı gruplarda yer alır. Benzerlik nesneleri tanımlayan özelliklerdir. Kümeleme analizi yüksek veri yoğunluğu sağlama tekniği olarak da tanımlanmaktadır. Bu tanımlamaya göre daha az bellek kullanılarak daha çok veriye ulaşılabilmektedir. Witten ve Frank kümeleme analizine daha karmaşık bir tanım getirmişlerdir [33]. Bu tanıma göre: Kümeleme işlemi ile elde edilen kümeler, örneklerin ve veri noktalarının çizildiği alanda bir mekanizma oluşturur. Bu mekanizma, bazı nesnelerin kendi aralarında diğer nesnelerden daha güçlü bir benzerlik oluşturmasını sağlar. Mercer, kümeleme algoritmalarını "ölçüm" uzayını "anlam" uzayına yönlendiren işlemler olarak tanımlamıştır [34]. Bu tanıma göre, ölçüm uzayı çok boyutlu olup sürekli, ayrık veya kategorik veriler içerebilir.

Anlam uzayı ise sonlu ve ayrık bir etiketler dizisidir. Bir nesnenin ölçüm uzayındaki değeri bulunduktan sonra, nesneyi bir kümeye dahil etmek için bu nesneye anlam uzayından bir etiket verilir. Kümeleme için yapılan ve literatürde önemli bir yere sahip bir diğer tanımlama ise şöyledir: Kümeleme analizi nesnelerin kendine özgü bir anlamı olan alt gruplara sınıflandırılması işlemidir. Kümeler temsil ettikleri nesneleri en iyi şekilde ifade edecek şekilde düzenlenir [32]. Bu tanıma göre, kümeleme sonlu bir nesne dizisi üzerine uyarlanmış, bir sınıflandırma yöntemidir.

Şekil 4.14 ’de kümelemenin özel (exclusive) ve danışmansız (unsupervised) bir sınıflandırma yöntemi olduğu görülmektedir. Özel sınıflandırma ile elde edilen nesne grubu veritabanının bir parçası gibidir. Bu tip sınıflandırma işlemlerinde her nesne sadece bir kümeye ait olabilir. Özel Olmayan (Nonexclusive) sınıflandırma işlemlerinde ise her nesne birden fazla kümeye dahil edilebilmektedir. Danışmanlı sınıflandırmada nesneler önceden tanımlanır ve her nesnenin bir sınıf etiketi vardır. Yeni eklenen veya sınıfı belirlenmemiş nesnelerin sınıflandırılması için kullanılır. Kümeleme analizi gibi, gözetimsiz sınıflandırmalarda ise nesnelerin önceden belirlenmiş bir sınıf etiketi yoktur. Bu tip sınıflandırmalarda grupların belirlenmesi için yakınlık matrisi (proximity matrix) ile beraber nesnelerin özellikleri de kullanılır [30,32].



Şekil 4.14. Sınıflandırma ağacı [32]

Kümeleme analizi, veri indirgeme veya nesnelerin gerçek sınıflarını bulma gibi çeşitli amaçlarla kullanılmaktadır [28]. Kümeleme analizinin kullanıldığı sayısız uygulama alanı bulunmaktadır. Bu alanlardan en çok gündemde olanlar örüntü tanıma, veri analizi, görüntü tanıma, pazarlama, metin madenciliği, doküman toplama, istatistik araştırmaları, makine öğrenimi, şehir planlama, coğrafik analizler (deprem, meteoroloji, yerleşim alanları), uzaysal veritabanı uygulamaları, web uygulamaları, CRM, sağlık araştırmaları ve biyolojik araştırmalardır [30,31,35].

4.2.2. Kümeleme analizi nitelikleri

Kümeleme analizi algoritmalarını niteleyen ve ideal bir kümeleme algoritmasının taşıması gereken başlıca özellikler şunlardır [27,30,40];

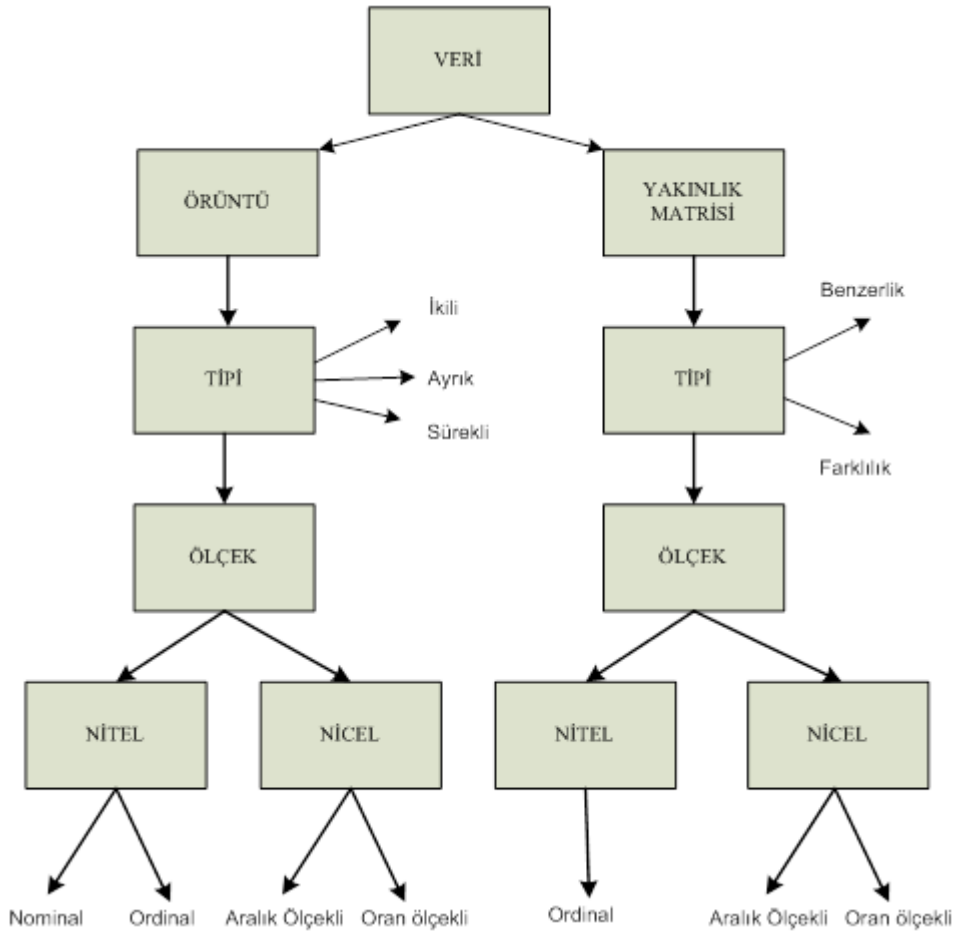
- Ölçeklenebilirlik: Kümeleme metodu çok büyük veritabanları üzerinde uygulanabilmelidir. Sadece küçük veritabanları üzerinde başarılı olan ve sadece bu veritabanları üzerinde uygulanabilen kümeleme metotları kullanışsızdır.
- Farklı yapılardaki veri türleri ile kullanılabilme: Kümelenecek veri seti

sayısal, ikili (boolean) veya kategorik veri gibi çeşitli veri tipleri içerebilir. İdeal bir kümeleme yöntemi tüm veri tipleri üzerinde uygulanabilir olmalıdır.

- Değişik şekil ve boyutlardaki kümeleri bulabilme: Bu nitelik uzaysal veri kümeleme için önemli bir gereksinimdir. Başarılı bir kümeleme algoritması küresel, uzamış, seyrek ve yoğun gibi çeşitli dağılım yapılarına sahip kümeleri bulabilmelidir.
- Danışmansız çalışabilmelidir.
- Veritabanını bir kez tarayarak kümeleri bulabilme yeteneğine sahip olmalıdır.
- Minimum sayıda giriş parametresi ile çalışma: Tarafsız bir kümeleme işleminin gerçekleşmesi için kümeleme algoritması mümkün olduğunca kullanıcı kararlarından bağımsız olmalıdır.
- Sıradışı veriler için özel önlemlere sahip olma: Kümeleme algoritmasının sonuçları, kullanılan veritabanında gürültülü ve sıradışı verilerin olması durumundan etkilenmemelidir.
- Veri kayıt sıralamasından bağımsız olma: Kümeleme algoritması, veritabanının hangi elemanından başlanırsa başlansın aynı kümeleme sonucunu vermelidir.
- Çok boyutlu veritabanlarında uygulanabilme: Kümeleme algoritmasının çalışması belli bir veritabanı boyutu (dimension) ile sınırlı olmamalıdır.
- Veri kümesinin sınırlılıklarını dikkate almalıdır.
- Kolay yorumlanabilir sonuçlar üretme ve işlevsel olma: Kümeleme sonucu elde edilen sonuçlar anlaşılır ve kullanışlı olmalıdır.

4.2.3. Küme analizi veri tipleri

Kümeleme algoritmaları, nesneleri, nesne çiftleri arasındaki benzerlik sıralamasına göre gruplamaktadır. Kümeleme analizlerinde kullanılan nesne dizileri iki kategoride incelenecek olan işlenmemiş veriden oluşmaktadır: Örüntü (Pattern) matrisi ve yakınlık (Proximity) matrisi. Bu matrislerin veri tipi ve ölçeklerine bağlı ana hatları Şekil 4.15'te görülmektedir [32].



Şekil 4.15. Kümeleme analizi veri türleri [32]

Örüntü matrisi (nesne-değişken yapısı) (pattern matrix)

p adet özelliği tanımlanan n adet nesneden her biri örüntü olarak tanımlanır. Nesnelerin tümünün birleşmesinden oluşan [n x p] boyutundaki matris, örüntü matrisidir. Bu matrisin her satırı bir veri değerini (teknik anlamda bir örüntüyü) ve her sütunu bir özelliği ifade etmektedir. Örüntü matrisinde sütunları oluşturan bu özellikler için bir hastanedeki hastaların yaş, boy, ağırlık, cinsiyet gibi kişisel özellikleri ve tahlil sonuçları örnek olarak verilebilir [27,31,32]. Örüntü matrisleri aşağıdaki gibi bir yapıya sahiptir. Kümeler birbirine yakın veya aralarında uzaysal bir ilişki olan örüntülerle görüntülenir. Kümeleme algoritmasının görevi nesnelerin uzaydaki bu gruplaşmalarını bulmaktır.

x_{11}	…	x_{1f}	…	x_{1p}
…	…	…	…	…
X_{i1}	…	x_{if}	…	x_{ip}
…	…	…	…	…
X_{n1}	…	x_{nf}	…	x_{np}

Benzerlik Matrisi (nesne-nesne yapısı) (Proximity Matrix)

Kümeleme algoritmaları örüntü çiftleri arasındaki benzerlik, farklılık, birliktelik ve yakınlık ilişkilerine bağlı bir sıralamanın oluşturulmasını gerektirmektedir. Bu sıralama örüntü matrisleri veya işlenmemiş veri kullanılarak bulunabilir. Benzerlik matrisi, her satır ve her sütunu bir örüntüyü temsil eden, nesnelerin karşılıklı benzerlik bilgilerinin tutulduğu [n x n] boyutlu bir matristir. Nesneler arasındaki benzerlik değişme özelliğine sahip olduğu için benzerlik matrisleri simetriktir. Diğer bir deyişle benzerlik değeri bir nesne çiftinin hangisi için bakılırsa bakılsın aynı değer elde edilir [31,32]. Bu matrisin genel yapısı aşağıda görülmektedir.

0	…	…	…	…
$d_{(2,1)}$	0	…	…	…
$d_{(3,1)}$	$d_{(3,2)}$	0	…	…
…	…	…	…	…
$d_{(n,1)}$	$d_{(n,2)}$	…	…	0

Burada d(i,j) i ve j nesneleri arasında ölçülen benzerlik veya farklılıktır. $D_{(i,j)}$ değeri, i ve j nesneleri arasındaki benzerlik yüksek ise 0'a yakın ve negatif olmayan bir değer alır. Nesneler birbirinden farklılaştıkça bu değer artmaktadır.

Veri matrisinde satırlar ve sütunlar farklı bilgileri taşıdığından bu matris iki yönlü (two mode) matris olarak adlandırılır. Benzerlik matrisinde ise satır ve sütunlar aynı nesneleri taşıdığından bu matris tek yönlü (one mode) matristir. Kümeleme algoritmalarının büyük çoğunluğu benzerlik matrisini kullanır. Bu nedenle, işlenmemiş verilerin önce örüntü matrisi oluşturulur ve daha sonra kümeleme algoritmalarını uygulamadan önce de benzerlik matrisine dönüştürülür [27].

Veri tipi, özelliklerin nicelik derecesini ifade etmektedir. Basit bir özelik ikili (binary), ayrık (discrete) veya sürekli (continuous) olabilir. İkili özelliklerin iki farklı değeri olabilir. Bu tür özelliklere örnek olarak cevabı "evet-hayır" olan sorular verilebilir. Ayrık özelliklerin ise sonlu, genellikle küçük sayıda farklı olası değerleri vardır. Belli bir aralık içinde herhangi bir değeri alabilen özellikler ise sürekli özelliklerdir.

Özelliklerin ikinci önemli özelliği ise sayıların bağıl etkisini gösteren veri ölçeğidir. Veri ölçekleri nitel (nominal ve ordinal) ölçekler ve nicel (aralık ölçekli ve oran ölçekli) ölçekler olmak üzere iki grupta incelenebilir. Nominal ölçekler gerçekte tam bir ölçek değildir, çünkü sayılar isimlerin yerine kullanılmaktadır. Örneğin bir (evet/hayır) sorusunun cevabı (0,1) olarak veya bir başarı seviyesi (kötü, orta, iyi) sorusunun cevabı (1, 2, 3) olarak kodlanmaktadır. Kullanılan sayıların nicel olarak bir anlamı yoktur. Bu tür değişkenler arasında benzerlik hesabı için

$$d_{(i,j)} = \frac{p - m}{p}$$

formülü kullanılır [27]. Formülde,

- $d_{(i,j)}$: i ve j nesneleri arasındaki benzerlik değerini,
- m : eşleşme (i ve j değişkenlerinin aynı değeri aldığı özellik) sayısını,
- p : i ve j nesnelerinin sahip olduğu toplam özellik sayısını ifade etmektedir.

Ordinal ölçek, sayıların etkisinin en zayıf olduğu ölçektir. Ordinal değişkenler arasında bir sıralama söz konusudur. Sıralamada bir üstte olan nesne daha alttaki nesnelerden daha değerlidir. Ordinal ölçekte, sayılar sadece birbiriyle ilişkili olarak bir anlam taşır. Örneğin (1, 2, 3) ve (10, 20, 30) değerleri sadece ordinal ölçekte eşdeğerdir. Ordinal değişkenlerde benzerlik değerinin hesaplanması için çeşitli yöntemler mevcuttur. Bunlardan en yaygın olarak kullanılan yöntem ordinal değişken değerlerinin [0-1] aralığındaki sayı değerlerine denkleştirilerek standartlaştırılması ve benzerliğin bu standartlaştırılmış değerler kullanılarak

ölçülmesidir. İkili ve ayrık özellikler ordinal ölçek kullanılarak kodlanabilir [8,14]. Aralık (interval) ölçekli veri türlerinde sayılar arasındaki aralık dikkate alınır. Ölçümlerin bir birimi vardır ve sayıların değeri bu birimlere bağlıdır. Örneğin sıcaklık ölçümlerinde 75° Kelvin ve 75° Celcius farklı değerler ifade etmektedir. Yapılan bir oylamada iki farklı politikacı için verilen (40, 50) ve (10, 90) puanları ölçeğin sınırlarına göre ([0-100], [1-10] veya [10-100]) farklı sonuçlar elde edilmesine neden olur. Bu nedenle ölçüm verileri üzerinde işlem yapılıyorsa birimlerin standartlaştırılması gerekmektedir. Aralık ölçekli değişkenlerde benzerlik değerinin hesaplanması için Öklit Uzaklığı, Öklit Uzaklığının Karesi, Manhattan Uzaklığı veya ChebyChev Uzaklığı formülleri kullanılmaktadır.

Sayıların gerçek değerleriyle kullanıldığı ve iki sayının oranının bir anlam ifade ettiği ölçek, oran (ratio) ölçekleridir. Oran ölçekli değişkenler, doğrusal olmayan ölçekler üzerinde yapılan ölçümler sonucu elde edilir. Ölçüm birimi ne olursa olsun iki farklı oran ölçekli değişkenin birbirine oranı değişmez. Örneğin iki şehir arasındaki mesafeyi ölçmek için metre, mil veya inç birimlerinden hangisi kullanılırsa kullanılsın, uzaklık değeri iki katına çıkarıldığına gidiş ve dönüş yolu arasında bir fark oluşmaz. Benzer şekilde bir kişinin maddi geliri iki kat arttığında, para birimi ne olursa olsun alım gücü de iki katına çıkacaktır. Oran ölçekli değişkenler Ae^{Bt} veya Ae^{-Bt} ile gösterilir. A ve B pozitif sabit sayılardır. Oran ölçekli değişkenlerde benzerlik hesaplamaları için kullanılan üç farklı yöntem vardır. Bunlardan ilki bu tür değişkenlerin aralık ölçekli değişkenler gibi kabul edilerek işlem yapılmasıdır. Ancak ölçümler doğrusal olmayan ölçekte yapıldığından bu yöntem ölçeğin bozulmasına neden olarak sonuçların hatalı olmasına yol açmaktadır. İkinci yöntem oran ölçekli değişkenlerin logaritmik dönüşümlerinin yapılmasıdır.

$$y_{if} = \log(x_{if})$$

logaritma işlemi y_{if} değerini doğrusallaştıracağından bu değişken üzerinde aralık ölçekli değişken gibi işlem yapılır. Oran ölçekli değişkenlerde benzerlik hesaplamaları için kullanılabilecek üçüncü yöntem bu değişkenlerin oran ölçekli değişken olarak kabul edilerek işlem yapılmasıdır.

Kümeleme işlemlerinde veri tipi ve ölçek seçiminin yapılması her zaman mümkün olmamaktadır. Kümeleme algoritmasının örüntü ve farklılık matrislerinin veri tipi ve ölçeğini tanıyarak işlem yapması önemlidir.

4.2.4. Kümeleme metadolojisi

Kümeleme analizinin gerçekleşmesi için uygulanan işlem basamakları aşağıdaki gibi sıralanmaktadır [32]:

1. Örüntü sunumu-özellik seçme/özellik oluşturma (pattern representation),
2. Veri bölgesine uygun örüntü yakınlık ölçütlerini tanımlama,
3. Kümeleme,
4. Veri soyutlama (Data abstraction) (gerekli ise),
5. Sonucu değerlendirme (Assesment of output) (gerekli ise)

Şekil 4.16'da kümeleme işleminin ilk üç adımı görülmektedir. Kümeleme işleminin sonucunda elde edilen yeni kümeler, benzerlik hesaplamaları ve özellik değerlerinin okunması işlemlerini etkileyebileceğinden buradan birinci adıma bir geri besleme gönderilmektedir [31].

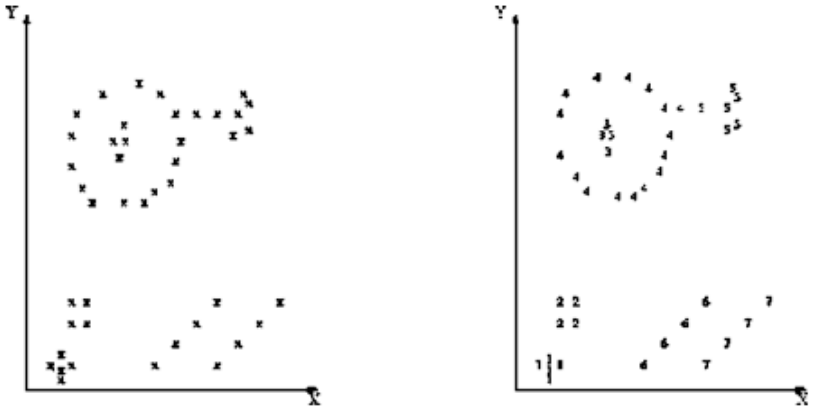
Şekil 4.16. Kümeleme işlemine genel bakış.

Örüntü sunumu, küme sayısı, örüntü sayısı ve kümeleme algoritmasında mevcut olan özelliklerin sayı, tür ve ölçeklerinin tanımlanması aşamasıdır. Özellik seçme, kümeleme işleminde örüntülerin yakınlık ve benzerliklerini en iyi ifade edecek özelliğin (boyut) seçilmesi işlemidir.

Özellik oluşturma ise, örüntü özelliklerinde bir veya daha çok dönüşüm yapılarak yeni ve kümeleme işleminde daha etkili olacak özelliklerin oluşturulması işlemidir. Örüntü yakınlığı ölçütlerinin tanımlandığı aşama örüntü çiftleri arasındaki benzerliğin hesaplandığı aşamadır.

Kümeleme adımı, çeşitli şekillerde gerçekleştirilebilir. Kümeleme işlemi sonunda elde edilen kümeler kesin (hard) veya bulanık (fuzzy) olabilir. Diğer bir deyişle her örüntünün her küme için değişen bir yakınlık derecesi bulunmaktadır.

Şekil 4.17a'da verilen veritabanı üzerinde kümeleme işlemi uygulandığında elde edilen kümelerin görüntüsü Şekil 4.17b'de verilmektedir. Burada aynı kümeye ait noktalar aynı etiketle gösterilmektedir. Küme gösteriminde, örüntüler arasındaki benzerliğin ölçümünde ve örüntülerin gruplanmasında kullanılan tekniklerin çeşitliliği zengin kümeleme algoritmalarının bulunmasını sağlamaktadır [31].



(a) Kümelenecek veritabanı (b) Bulunan kümeler.

Şekil 4.17. Kümeleme analizi örneği.

4.2.5. Değişken türlerine göre benzerlik ve uzaklık ölçüleri

Bir veri setinde yer alan birimlerin kümelenmesi, temel bileşenler analizi, çok boyutlu ölçekleme ve kendinden düzenlenen haritalar gibi boyut azaltma işlemlerinin yapılabilmesi, bu birimlerin birbirleriyle olan benzerlikleri (similarity) ya da birbirine olan uzaklıkları (dissimilarity) kullanılarak gerçekleştirilmektedir.

Benzerlik ve uzaklık ölçülerinden kısaca bahsetmek gerekirse, benzerlik ölçüleri, birimlerin birbirilerine olan benzerliklerini göstermekte kullanılan ölçümlerdir. Benzerlik ölçüleri maksimum 1 değerini benzerlik değeri olarak alabilirler. Benzerlik ölçülerinin değerleri arttıkça birimler arasındaki benzerlikler artar, azaldıkça da birimler arasındaki benzerlikler azalır. Uzaklık ölçüleri ise benzerlik ölçülerinin tam tersi bir yaklaşım sergilerler. Uzaklık ölçülerinin küçük olması birimlerin birbirilerine benzer olduğunu gösterir. Uzaklık ölçülerinde 0 maksimum benzerliği ifade eder. Uzaklık ölçülerinin değerleri arttıkça birimler arasındaki benzerlik azalır, azaldıkça da birimler arasındaki benzerlik artar.

Değişkenlerin kesikli ya da sürekli olmalarına ya da değişkenlerinin nominal, ordinal, aralık ya da oransal ölçekte olmalarına göre hangi uzaklık ölçüsünün ya da benzerlik ölçüsünün kullanılacağına karar verilir. Aşağıdaki alt bölümlerde değişken türlerine göre kullanılan uzaklık veya benzerlik ölçülerinden bahsedilecektir. Uzaklık ve benzerlik ölçülerini tanımlarken kullanılan i ve j indisleri 1,2,…,n değerlerini, k indisi 1,2,…,p değerlerini alabilir. Burada n birim sayısı, p değişken sayısıdır.

Aralık ve oransal ölçekli değişkenler

Aralık ölçeği sayısal olarak ifade edilebilen, toplama ve çıkarma gibi matematiksel işlemleri mümkün kılan ölçeklerdir. Aralık ölçekte gerçek anlamda bir sıfır yoktur. Ölçüm farklılıklarının ve düzenlerinin önemli olduğu bir ölçektir. Aralık ölçeğine ısı ölçümlerinde kullanılan Celcius ölçeğini örnek verebiliriz. Bilindiği gibi suyun donma derecesi Celcius ölçeğinde 0^{o}' dir. 0 ^{o}C hiçlik anlamına gelmemektedir. 0 ısının yok olduğunu göstermeyen keyfi bir değerdir. Aralık ölçeklerinin oranları da herhangi bir anlam taşımaz. 4 ^{o}C, 8 ^{o}C iki katı değildir, ancak ondan daha düşük bir ısıyı ifade etmektedir. Oran ölçeği ise, aralıkölçeğinin özelliğini taşıdığı gibi belli bir sıfır değerine sahip olan ölçeklerdir. Ölçümler arasında düzen ve uzaklık olduğu gibi ölçümler arasındaki oranda önemlidir. Örneğin, 20 metrekarelik bir alan 10 metrekarelik bir alanın 2 katı olduğu gibi. Aralık ve oran ölçekli veriler arasındaki uzaklık ve benzerlikler aşağıdaki gibi hesaplanır.

Öklidyen Uzaklık Ölçüsü

Öklidyen uzaklık ölçüsü kullanılarak iki birim arasındaki uzaklık

$$d\ i,j\ = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2}$$

formülü ile hesaplanır. En çok kullanılan ölçü birimidir. Değişkenler belirli bir önem derecesinde ağırlıklandırılmış ise, Öklidyen uzaklık ölçüsü formülü aşağıdaki gibi olur.

$$d\ i,j\ = \sqrt{\sum_{k=1}^{p} w_k\ x_{ik} - x_{jk}\ ^2}$$

Pearson Uzaklık Ölçüsü

Pearson uzaklık ölçüsü kullanılarak iki birim arasındaki uzaklık;

$$d\ i,j\ = \sqrt{\sum_{k=1}^{p} \frac{x_{ik} - x_{jk}\ ^2}{s_k\ ^2}}$$

formülü ile hesaplanır. Bu formülde kullanılan s_k, uzaklığın hesaplandığı değişkene ait standart sapmadır. Bununla birlikte farklı gruplar hakkında önceden bilgi sahibi olunmadığı için, uzaklık hesaplanmasında s değerinin kullanılması doğru olmaz. Bu nedenle Pearson uzaklık ölçüsü yerine genellikle Öklidyen uzaklık ölçüsü tercih edilir. Kümeleme analizinde kullanılacak olan değişkenler belirli önem derecelerine göre ağırlıklandırılmış ise, Pearson uzaklık ölçüsü formülü aşağıdaki gibi olur.

$$d\ i,j\ = \sqrt{\sum_{k=1}^{p} \frac{w_k\ x_{ik} - x_{jk}\ ^2}{s_k\ ^2}}$$

Pearson uzaklık ölçüsüne, "karesel Pearson uzaklık" ya da "standart Öklid uzaklığı" adı da verilir.

Manhattan Uzaklık (City Block) Ölçüsü

Manhattan uzaklık ölçüsü kullanılarak iki birim arasındaki uzaklık;

$$d\ i,j\ = \sum_{k=1}^{p} (|x_{ik} - x_{jk}|)$$

formülü ile hesaplanır. Bu ölçü de birimler arasındaki mutlak uzaklık kullanılır. Değişkenler eğer belirli bir önem derecesinde ağırlıklandırılmış ise, Manhattan uzaklık ölçüsü formülü aşağıdaki gibi olur.

$$d\ i,j\ = \sum_{k=1}^{p} (w_k|x_{ik} - x_{jk}|)$$

Manhattan uzaklık ölçüsüne, "city block uzaklık ölçüsü" adı da verilir.

Minkowski Uzaklık Ölçüsü

Minkowski uzaklık ölçüsü genel bir formüldür. Formülde yer alan λ değerinin alacağı farklı değerlere göre yeni formüller türetilir. Minkowski uzaklık ölçüsü kullanılarak iki birim arasındaki uzaklık

$$d\ i,j\ = \sum_{k=1}^{p} |x_{ik} - x_{jk}|^{\lambda}{}^{\frac{1}{\lambda}}$$

formülü ile hesaplanır. Değişkenler belirli önem derecelerine göre ağırlıklandırılmış ise, Minkowski uzaklık ölçüsü formülü aşağıdaki gibidir.

$$d\ i,j\ = \sum_{k=1}^{p} w_k|x_{ik} - x_{jk}|^{\lambda}{}^{\frac{1}{\lambda}}$$

Minkowski uzaklık ölçüsündeki λ değeri büyük ve küçük farklara verilen ağırlığı değiştirir. Minkowski uzaklık ölçüsü $\lambda = 1$ için Manhattan uzaklık ölçüsüne, $\lambda = 2$ için Öklidyen uzaklık ölçüsüne dönüşür.

Mahalanobis uzaklık ölçüsü

Mahalanobis uzaklık ölçüsü kullanılarak birimler arasındaki uzaklık:

$$d^2_{(i,j)} = (x_i - x_j)^T \quad ^{-1} \quad (x_i - x_j)$$

formülü ile hesaplanır. Burada **Σ** kovaryans matrisidir. Çoğunlukla mevcut veriler kullanılarak **Σ** kovaryans matrisi tahmin edilir. Burada bulunan **Σ** kovaryans matrisi aşırı değerlere karşı duyarlıdır.

Açısal benzerlik ölçüsü (Cosine Similarity Measure)

Açısal benzerlik ölçüsü iki veri noktasının özellik vektörleri arasındaki açının kosinüsüdür. [+1,-1] arasında değerler alır. İki vektör arasındaki açısal benzerlik ölçüsü aşağıdaki gibi belirlenir.

$$\boldsymbol{s_{i,j}} = \frac{\boldsymbol{x_i^T x_j}}{\boldsymbol{x_i^T x_i} \quad \overline{\boldsymbol{x_j^T x_j}}}$$

Korelasyon benzerlik ölçüsü (Correlation Similarity Measure)

İki özellik vektörü arasındaki korelasyon değeri bu iki vektörün birbirine olan benzerlik derecesini gösterir, [+1,-1] aralığında değerler alır. Herhangi iki vektör arasındaki korelasyon benzerlik ölçüsü aşağıdaki gibi belirlenir.

$$\mathbf{S}_{ij} = \frac{(\boldsymbol{X}_i - \vec{\boldsymbol{X}})^T (\boldsymbol{X}_j - \vec{\boldsymbol{X}})}{\sqrt{(\boldsymbol{X}_i - \vec{\boldsymbol{X}})^T (\boldsymbol{X}_i - \vec{\boldsymbol{X}})} \sqrt{(\boldsymbol{X}_j - \vec{\boldsymbol{X}})^T (\boldsymbol{X}_j - \vec{\boldsymbol{X}})}}$$

Uzaklık fonksiyonun özellikleri

- $d_{(i,j)} \geq 0$ Uzaklık negatif değil
- $d_{(i,i)} = 0$ Her birim kendisine olan uzaklığı sıfırlar.
- $d_{(i,j)} = d_{(j,i)}$ Uzaklık fonksiyonu simetriktir.
- $d_{(i,j)} \leq d_{(i,h)} + d_{(h,j)}$ İki birimin arasındaki uzaklık bu iki birimin üçüncü bir birime olan uzaklıkları toplamından büyük olamaz (üçgen eşitsizliği).

Nominal ölçekli değişkenler

Bu ölçekte kullanılan rakamlar veya isimler birimleri sınıflara veya kategorilere ayırmaktadır. Örneğin nüfusu cinsiyet özelliğine göre sınıflandırırken kadın ve erkek gibi şıklar kullanılabileceği gibi kadınlar için 1, erkekler için 0 rakamları birimlerin sınıflandırılması için de kullanılabilir. Bu ölçekte bulunan verilerin toplamları hiçbir anlam taşımaz. Nominal ölçekli değişkenler ikili (binary), ikili olmayan ölçekli değişkenler olarak ikiye ayrılırlar.

İkili nominal değişkenler

Erkek - Kadın, Evet – Hayır, Olumlu – Olumsuz gibi iki seçenek alabilen değişkenler nominal ölçekli değişkenlerdir. İkili nominal değişkenlerde, aralık ve oransal ölçekli verilerde kullanılan Pearson, Öklidyen, Manhatten (City block) Minkowski gibi birimler arası uzaklıklarının kullanılması uygun değildir.

İkili nominal ölçekli değişkenlerde dört gözlü kontenjans tablolarından yararlanarak benzerlik ölçüleri elde edilebilir. Dört gözlü kontenjans tablosundan elde edilen 1-1, 0-0, 0-1, 1-0 eşleşmelerin frekansları kullanılarak sözü edilen benzerlik ölçüleri hesaplanır. Çizelge 4.3'te dört gözlü kontenjans tablosu yer almaktadır.

Çizelge 4.3. İkili değişkenler için kontenjans tablosu

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

Çizelge 4.3.'te yer alan kontenjans tablosundaki frekans değerleri kullanılarak aşağıdaki Çizelge 4.4 ’ te yer alan 6 adet benzerlik ölçüsü elde edilir.

Çizelge 4.4. Benzerlik Ölçüleri

Katsayı	Benzerlik Ölçüleri
Jaccard Benzerlik Katsayısı	$\frac{a}{a+b+c}$
Ochiai Benzerlik Katsayısı	$\frac{a}{\sqrt{(a+b)(a+c)}}$
Rao Benzerlik Katsayısı	$\frac{a}{a+b+c+d}$
Basit Eşleşme Benzerlik Katsayısı	$\frac{a+d}{a+b+c+d}$
Öklid Uzaklığı	$\sqrt{b+c}$
Karesel Öklid Uzaklığı	b+c

İkili Olmayan Nominal Değişkenler

İki seçenekten daha fazla seçeneğe sahip olan değişkenlerin uzaklıklarının hesaplanması ise şu formülle hesaplanır;

$$d_{i,j} = \frac{p-m}{p}$$

p toplam değişken sayısı, m eşleşen değişken sayısı olmak üzere eşleşmeyen değişken sayısı (p-m) toplam değişken sayısına oranlanarak iki birim arasındaki uzaklık bulunur. Bu uzaklık 1’ den çıkarılırsa, iki değişken arasındaki benzerlik katsayısı bulunur.

Ordinal ölçekli değişkenler

Bu ölçekte rakamlar, büyüklük, tercih gibi çeşitli özelliklerin sıralanmasında kullanılır. Markaların en çok beğenilenden en az beğenilene doğru 1’ den başlayarak sıralanmasında ordinal ölçek kullanılmaktadır. Burada bireylerin markalardan ne

kadar memnun olduğu anlaşılamamakta, birinci sıradaki markanın dördüncü sıradakinden 4 katı kadar beğenildiği anlamı çıkartılamamaktadır. Ordinal ölçekli değişkenlere sahip birimlerin uzaklık ölçüsü aşağıdaki adımları takip ederek elde edilir.

- x_f değişkenine ait i. birimin sıralaması M birim içerisinde r_{if} olur.
- Her bir ordinal değişken farklı sayıda durumlara sahip olacağından, her bir değişken aşağıdaki formül ile [0,1] aralığına indirgenir.

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- [0,1] aralığına indirgenen değişkenlere ait birimler arasındaki uzaklıklar oransal ve aralık ölçekte kullanılan uzaklık ölçüleri ile bulunur.

Uzaklık ve benzerlik ölçülerinin birbirilerine dönüşümü

Benzerlik ölçüleri, uzaklık ölçülerine çeşitli dönüşümlerle dönüştürülebilirler. Burada uzaklık ölçüsü $d_{i,j}$ benzerlik ölçüsü $s_{i,j}$ olmak üzere aşağıdaki gibi dönüşümlerle birbirilerine dönüştürülürler. Benzerlik ölçüsü aşağıdaki gibi uzaklık ölçüsüne dönüştürülebilir.

$d_{i,j}=1-s_{i,j}$

$d_{i,j}=c-s_{i,j}$

$$di,j = \overline{2\ 1 - s_{i,j}}$$

$$d_{i,j} = \overline{s_{i,i} - 2s_{i,j} + s_{j,j}}$$

Bazı durumlarda uzaklık ölçüsünden benzerlik ölçüsünü elde etmek isteriz. Bu durumda aşağıdaki dönüşümü kullanabiliriz.

$$s_{i,j} = (1 + d_{i,j})^{-1}$$

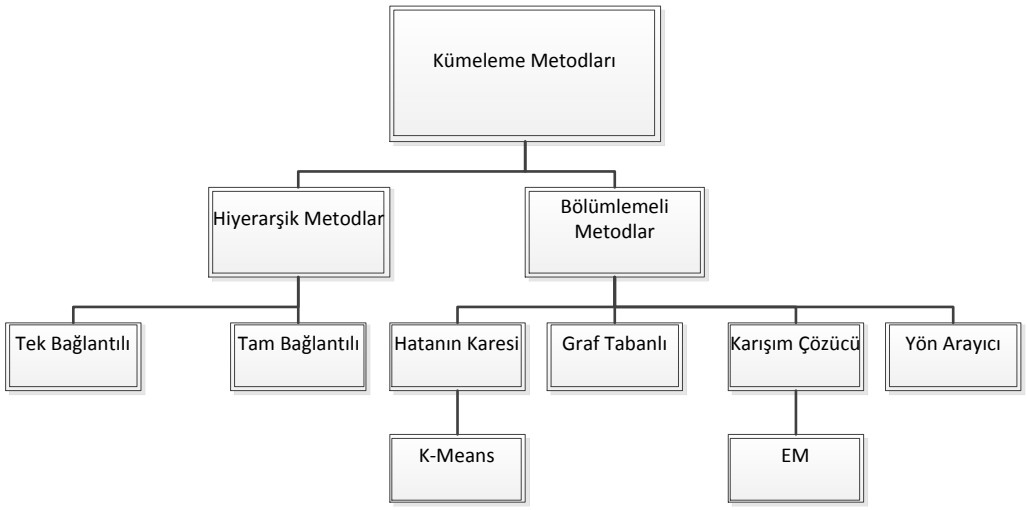
4.2.6. Kümeleme metotları

En yaygın olarak kullanılan veri madenciliği tekniklerinden biri olan kümeleme analizini gerçekleştirmek için birçok kümeleme metodu geliştirilmiştir. Kümeleme metotları kullandıkları kümeleme yöntemlerine bağlı olarak bir hiyerarşi oluşturmaktadır. Jain ve Dubes tarafından oluşturulan kümeleme metotları hiyerarşisi Şekil 4.18'de görülmektedir [31]. Bu hiyerarşik yapının en üst seviyesinde kümeleme metotları hiyerarşik ve bölümleyici metotlar olarak ikiye ayrılmaktadır. Hiyerarşik metotlar iç içe yerleşmiş küme serileri oluştururken bölümleyici metotlar tek seviyeli kümeler oluşturur. Şekil 4.18'deki hiyerarşik yapının oluşmasında etkili olan diğer yaklaşımlar şöyledir:

- Birleştirici - Ayrıştırıcı (Agglomerative-Divisive) : Birleştirici (Agglomerative) yaklaşım her örüntüyü ayrı bir (singleton) kümeye yerleştirerek başlar ve bir sonlanma koşulu sağlanana kadar kümeleri birleştirir.
- Ayrıştırıcı (Divisive) metotlar ise tüm örüntüleri tek bir kümeye yerleştirerek işleme başlar. Bir sonlandırma koşulu sağlanana kadar bu küme üzerinde parçalama işlemi devam eder.
- Tekli - Çoklu (Monothetic - Polythetic) : Bu yaklaşım, kümeleme işlemindeki özelliklerin ardışık ve eş zamanlı kullanımı ile ilgilidir. Algoritmaların çoğu çokludur (polythetic). Diğer bir deyişle; örüntüler arasındaki benzerlik hesaplamalarında örüntülerin tüm özellikleri kullanılır ve kümeleme işlemi bu benzerlik değerine bağlı olarak yapılır.
- Kesin - Bulanık (Hard - Fuzzy) : Kesin (Hard) kümeleme algoritmalarında her örüntü tek bir kümeye yerleştirilebilir. Bulanık (Fuzzy) kümeleme algoritmalarında ise, her örüntünün oluşturulan her küme için bir üyelik derecesi vardır. Bulanık bir kümeleme metodu, her örüntünün en büyük üyelik derecesine sahip olduğu kümeye atanmasıyla bir hard kümeleme algoritmasına dönüştürülebilir.
- Belirleyici - Belirsiz (Deterministic - Stochastic) : Hatanın karesi fonksiyonunu en iyi şekilde kullanmak için tasarlanmış bölümleyici metotlarda etkili olan bir yaklaşımdır. Optimizasyon işleminin gerçekleştirilmesi geleneksel teknikler

kullanılarak veya tüm etiketleri içeren durum uzayında rasgele bir arama ile sağlanır.

- Artan - Artmayan (Incremental - Nonincremental) : Bu yaklaşım, kümelenecek veri setinin büyük olduğu ve çalışma süresi veya bellek boyutu sınırlamalarının algoritma mimarisini etkilediği durumlarda etkilidir. Veri madenciliği ve kümeleme işlemlerinin gelişmesiyle ortaya çıkan algoritmalar, veri seti üzerindeki okuma (scan) sayısını minimize eden, algoritmanın icrası sırasında incelenen örüntü sayısını azaltan veya algoritma işlemlerinde kullanılan veri yapılarının boyutunu düşüren kümeleme algoritmalarının gelişimini tetiklemiştir.



Şekil 4.18. Kümeleme metotları hiyerarşisi

Bölümleyici metotlar (Partitioning Methods)

Bölümleyici metotlar, n adet nesneden oluşan veri setini, her bölümün bir kümeyi temsil ettiği k adet bölüme ($k \leq n$) ayıran metotlardır. Veri setindeki nesnelerin birleşerek küme oluşturması farklılık fonksiyonları (dissimilarity function) hesaplanarak nesneler arasındaki uzaklıkların minimum değerlerinin bulunmasıyla sağlanır. Kümeleme işlemi sonucunda bulunan kümelerde, küme içi nesneler arası benzerlik maksimum, farklı kümelerdeki nesneler arası benzerlik ise minimum olur.

Bölümleyici metotlar çok büyük olmayan veritabanlarında küresel ve benzer boyutlardaki kümelerin bulunmasında en iyi sonuç vermektedir. En yaygın olarak kullanılan bölümleme algoritmaları k-means, k-medoids, EM ve CLARA-

CLARANS'tır. Bölümleme algoritmalarının tümü eşdeğer kümeleme kalitesine sahiptir. Ancak bu algoritmaların ortak problemleri de vardır [27,30,39] :

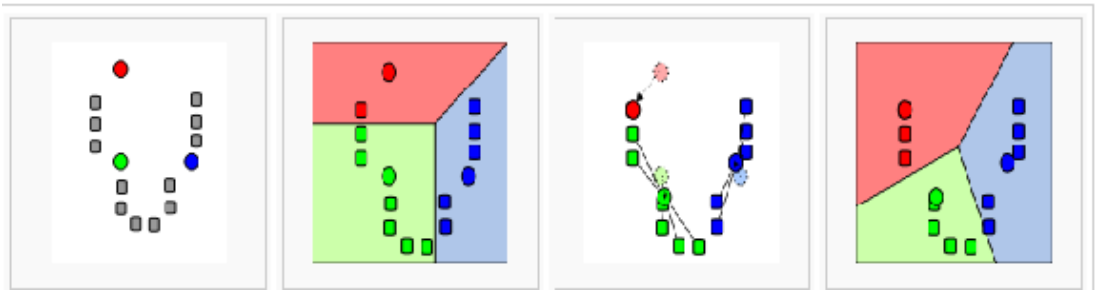
- Küme sayısı (k) kümeleme işleminden önce belirlenmelidir. k değerinin doğru tahmin edilmediği durumlarda başarılı sonuçlar elde edilmez ve kümeleme işleminin tekrarlanması gerekir;
- Boyutları farklı kümelerin bulunması zordur. Büyük boyutlu kümelerin bölünmesi olasıdır;
- Sadece içbükey küresel kümeler bulunabilmektedir.

K-means algoritması

1967 yılında Mac Queen tarafından bulunan k-means algoritması, kümeleme problemini çözen en basit danışmansız öğrenme algoritmalarından biridir. Bilimsel ve endüstriyel uygulamalarda en yaygın olarak kullanılan kümeleme algoritmasıdır. K-means algoritmasının ismindeki 'k' oluşturulacak küme sayısını, 'means', kümeyi oluşturan elemanların ağırlıklı ortalamasını ifade etmektedir [30].

K-means algoritmasında, her küme kümenin ağırlıklı ortalama değerine sahip veya bu noktaya en yakın olan nokta kümenin ağırlık merkezini oluşturur ve küme merkezi (centroid) olarak adlandırılır [42]. Algoritmada oluşturulacak küme sayısı (k) önceden belirlenir. Algoritmanın ilk adımında küme merkezleri belirlenir. İlk küme merkezleri rasgele, doğrusal ya da istatistiksel olarak seçilebilir [43,44]. En yaygın olarak kullanılan yöntem k adet küme merkezinin rasgele olarak seçilmesidir [30]. Algoritmanın ikinci adımında veri setindeki her nokta kendine en yakın merkez noktayla birleşir. Bu işlem sonunda k adet küme oluşmuş ve her bir veri noktası bu kümelerden birine yerleşmiş olur. Oluşan kümelerin ağırlıklı ortalama değeri hesaplanarak bu değere sahip veya bu değere en yakın nokta kümenin yeni merkez noktası olarak atanır. Merkez noktalar değişeceğinden veri setindeki noktaların merkez noktalara uzaklıkları da değişir. Bu nedenle noktaların kümelere yerleştirilmesi ve bu yerleşime göre oluşan kümelerin merkez noktalarının bulunması işlemi tekrarlanır. Bu işlem, bir kümedeki veri noktalarının kümenin merkez noktasına uzaklıkları toplamının değerindeki düşüş minimum oluncaya kadar devam

eder. Burada amaç, küme merkezlerinin birbirinden maksimum uzaklıkta olması ve küme içindeki nesnelerin mümkün olduğunca birbirine yakın olmasıdır [27,31,41]. K-means algoritmasının işlem basamakları Şekil 4.19'da görülmektedir.



Şekil 4.19. K-means kümeleme algoritması.

K-means algoritması, küçük ve orta boyutlu veritabanlarında küresel kümelerin bulunmasında başarılı bir algoritmadır. K-means algoritmasının hesaplanabilir karmaşıklığı $O^{(tkn)}$ olduğundan etkili ve uygulaması kolay bir algoritmadır. Algoritmanın tekrar sayısı (t), veri noktaları sayısı (n) ve küme sayısından (k) küçüktür (t<<n,k) olmalıdır. K-means algoritmasının pek çok alanda uygulanmasını sağlayan avantajlarının yanında birtakım dezavantajları da bulunmaktadır. K-means algoritması ile elde edilen kümeler, küme sayısı, ilk küme merkezlerinin seçim yöntemi, veri noktalarının sıralaması ve verinin geometrik özelliklerine bağlıdır. K-means algoritmasında k değeri önceden tespit edilemediğinden başarılı bir kümeleme elde etmek için deneme – yanılma yönteminin kullanılması gerekmektedir [41]. Sıradışı ve gürültülü verilere karşı duyarlı bir algoritmadır. Ortalama değeri hesaplanamayan kategorik verilerde kullanılamaz. Konveks olmayan kümelerin bulunmasında uygun değildir [45]. K-means algoritması çok büyük veritabanlarında uygulandığında büyük kümelerin bölünmesine neden olmaktadır [39].

EM Algoritması

EM (Expectation Maximization) algoritması, her kümeyi tek bir nokta ile temsil etmek yerine, her kümeyi bir olasılık dağılımı (probability distribution) ile temsil eder. Genel olarak Gaussian olasılık dağılımı kullanılmaktadır. Çünkü bu yöntemde

yoğunluk tahmini kuramı (density estimation theory) ve yoğunluk dağılımı, yaklaşık olarak tahmin edilebilmektedir [35].

K-medoids Algoritması

1987 yılında Kaufman ve Rousseeuw tarafından sunulan k-medoids algoritmasında küme merkezleri k-means algoritmasından farklı olarak kümedeki noktaların ağırlıklı ortalama değeri hesaplanarak değil kümenin en merkez noktasında yerleşmiş nokta bulunarak belirlenir. Bu işlem sıradışı noktaların küme merkezini etkilemesini engellemektedir. Merkez noktaların bulunmasında ortalama değerin bulunması gibi matematiksel işlemlere gerek olmadığından k-medoids algoritması kategorik verilerde kullanılabilmektedir.

K-medoids algoritmasının en eski iki türevi PAM (Partitioning Around Medoids) ve CLARA (CLustering LArge Applications)'tır. PAM algoritmasında, küme sayısı (k) kadar merkez nokta (medoid) rasgele olarak seçilir. Merkez noktalar dışında kalan diğer noktalar kendilerine en çok benzeyen merkez noktayla kümelendirilir. Elde edilen kümelenme en kaliteli durumu alana kadar, merkez nokta, merkez olmayan noktalarla yer değiştirir [27,30].

PAM algoritması küçük veritabanlarında çok iyi sonuçlar vermektedir. Ancak büyük veritabanlarında düşük performans göstermektedir ve küme sayısı kullanıcı tarafından belirtilmelidir [27]. Büyük veritabanlarında kullanılan kmedoids algoritması CLARA algoritmasıdır.

CLARA-CLARANS Algoritmaları

1990 yılında Kaufman ve Rousseeuw tarafından bulunan CLARA algoritması büyük veritabanlarında etkili olan ve örnekleme-tabanlı bir bölümleyici metottur [46]. CLARA algoritmasında kümeleme işlemi için tüm veritabanını kullanmak yerine veritabanının temsilcisi olarak küçük bir örnekleme kümesi kullanılır. Bu örnekleme kümesi içindeki merkez noktalar (medoids) PAM algoritması kullanılarak bulunur. Örnekleme kümesi doğru bir şekilde oluşturulursa bulunan merkez noktalar, tüm veritabanı kullanılarak bulunan merkez noktalarla aynı olur. CLARA algoritmasının

daha iyi sonuçlar vermesi için, başlangıçta birden fazla örnekleme seçilerek ve PAM algoritması bu örneklemeler için ayrı ayrı uygulanabilir. Bulunan kümelemelerden en iyisi sonuç olarak verilir.

CLARA algoritmasının en önemli avantajı büyük veri setlerinde kullanılabilmesidir. Ancak bu algoritmanın performansı örnekleme kümesinin boyutuna bağlıdır. Ayrıca, CLARA algoritmasında merkez noktalar seçilen örnekleme noktalar arasında aranmaktadır. Gerçek merkez noktaların örnekleme içine alınmadığı durumlarda CLARA algoritması başarılı sonuçlar vermemektedir.

CLARA'nın kalitesini ve ölçeklenebilirliğini arttırmak ve sonuçların örnekleme kümesinin seçimine bağlı olmasını önlemek için 1994 yılında VLDB'94 konferansında Ng ve J.Han tarafından, CLARANS (CLustering ALgorithm based on RANdomized Search) algoritması sunulmuştur [27,46].

CLARANS algoritmasında CLARA'dan farklı olarak örnekleme noktalar algoritmanın her aşamasında değişebilmektedir. Böylece tüm veritabanının örnekleme seçimi tarafsızca gerçekleşmiş olur [46].

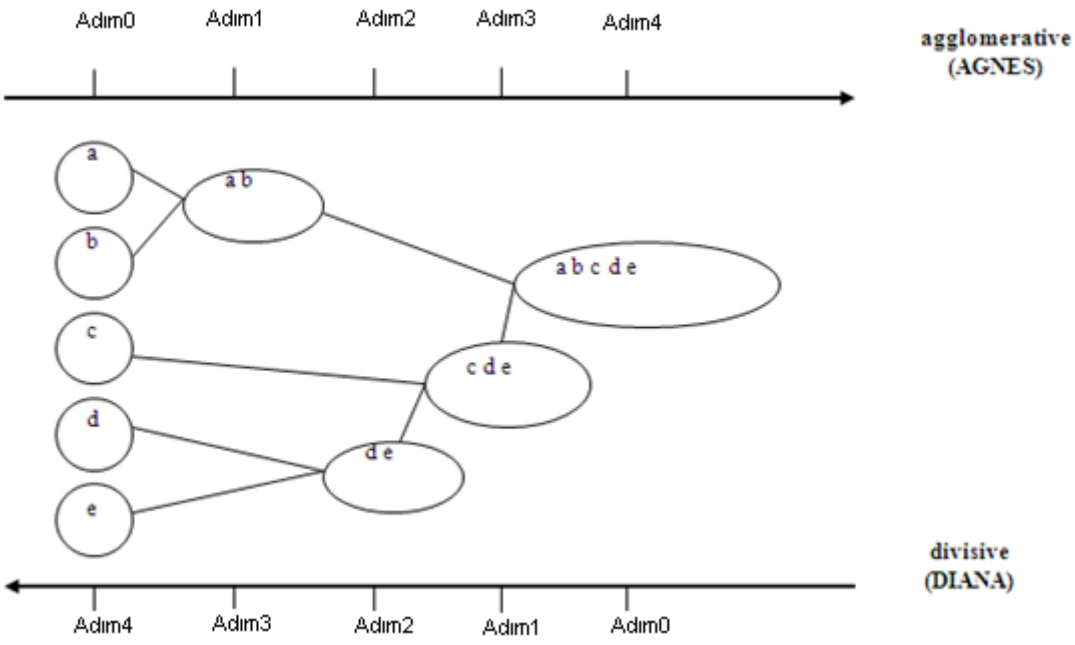
Hiyerarşik metotlar

Hiyerarşik kümeleme metotları veri nesnelerini dendogram adı verilen ağaç yapısı içerisine gruplandırmaya çalışır. Bu ağaç, yapraklardan gövdeye doğru veya gövdeden yapraklara doğru kurulabilir. Hiyerarşik kümelemeye toplayıcı ve bölücü kümeleme (Agglomerative and Divisive Hierarchical Clustering), BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies), CURE (Clustering Using Representatives) ve CHAMELEON (A Hierarchical Clustering Algorithm Using Dynmaic Modeling) algoritmaları örnek olarak verilebilir. Toplayıcı hiyerarşik kümelemede hiyerarşik ayrışma aşağıdan yukarıya doğru olur. Bu nedenle aşağıdan-yukarıya yaklaşım olarak geçmektedir. Bölücü hiyerarşik kümelemede hiyerarşik ayrışma yukarıdan aşağıya doğru olmaktadır. Bölücü hiyerarşik kümeleme yukarıdan-aşağıya yaklaşım olarak geçmektedir.

Toplayıcı ve bölücü algoritmalar

Her bir nesnenin ilk başta farklı bir küme oluşturmasıyla başlayıp sonlandırma kıstası sağlanıncaya kadar kümelerin birleştirme işlemine devam eden toplayıcı hiyerarşik kümeleme yaklaşımı vardır. Ayrıca bütün nesnelerin aynı kümede olması ile başlayıp sonlandırma koşulu sağlanıncaya kadar kümeleri bölen bölücü hiyerarşik kümeleme yaklaşımı vardır.

Toplayıcı hiyerarşik kümeleme aşağıdan-yukarıya yaklaşımlı hiyerarşik kümelemedir. Kendi kümesi içindeki her bir nesnenin yerini değiştirmesiyle başlar ve daha sonra atomik kümeleri, tüm nesneler tek bir küme içinde toplanıncaya kadar veya belirli bir son koşulu sağlanana kadar, benzerliklere dayalı olarak birbirini takip edecek şekilde daha geniş olan kümelere birleştirir. N nesne için, n–1 birleştirme yapılır. Hiyerarşik algoritmalarla bir birleştirme yapıldığında geriye dönüş yoktur. Toplayıcı hiyerarşik kümeleme algoritmaları az hesaplama maliyetine sahip olmasına rağmen yanlış bir birleştirme yapılması sorunlara yol açar. Bu yüzden birleştirme noktalarının dikkatlice seçilmesi gerekir. Çoğu hiyerarşik kümeleme metodları bu kategoride yer almaktadır. AGNES (AGglomerative NESting) olarak adlandırılan Kaufman tarafından ortaya atılan metod bu yaklaşımı kullanır. Şekil 4.20’ da gösterildiği gibi {a, b, c, d, e} nesneleri AGNES metodu ile başlangıçta her nesne kendisine ait olan bir küme içinde olacak yerleştirilir. Daha sonra kümeler diğer kümeler içindeki en yakın nesneyle arasındaki minimum öklit uzaklığına göre adım adım birleştirilmiştir.



Şekil 4.20. Veri nesneleri üzerinde toplayıcı ve bölücü hiyerarşik kümeleme [27].

Bölücü hiyerarşik kümeleme yukarıdan-aşağıya yaklaşımlı hiyerarşik kümelemedir. Başlangıçta tüm nesneleri tek bir küme içinde saklar. Her bir nesne kendi içinde bir küme oluşana kadar veya istenen sayıda küme elde edilmesi ya da iki en yakın küme arasındaki uzaklığın eşik değerin üstünde olması gibi belirli bir son koşul sağlanana kadar daha küçük parçalara kümeyi böler. Yüksek seviyede bölümleme için doğru bir seçim gerektiği durumlarda uygulanması zor olduğundan bölücü metot kullanılmaz. DIANA (DIvisive ANAlysis), Kaufman tarafından ortaya çıkarılan bölücü hiyerarşik kümeleme metodudur. Bu metodda; Şekil 4.20' de görüldüğü gibi ilk olarak bütün nesneler tek bir kümeye yerleştirilir. Küme, küme içindeki en yakın komşu nesneler arasındaki maksimum öklit uzaklığı gibi bazı kriterler sağlanıncaya kadar bölünür. Küme bölme işlemi, her bir yeni küme sadece tek bir nesne içerinceye kadar devam eder. Toplayıcı ve bölücü hiyerarşik kümeleme yaklaşımlarının her ikisinden de kullanıcı sonlandırma kıstası olarak kümelerin sayısı belirtilebilir. İşlem istenilen küme sayısına ulaştığı zaman hiyerarşik kümeleme işlemi sonlanır. Kümeler arasındaki uzaklık için kullanılan 4 ölçüm aşağıda gösterilmiştir. Bu ölçümlerdeki p-p', p ve p' noktaları arasındaki uzaklığı, m_i: C_i kümesinin ortalamasını, n_i: C_i kümesi içindeki nesnelerin sayısını ifade eder [27].

Hiyerarşik kümeleme yaklaşımı basit olsa bile bölme ve birleştirme noktalarının dikkatlice seçilmesi gerekir. Birleştirme ve bölme noktaları doğru bir şekilde seçilmezse düşük kalitede kümeler oluşabilir.

BIRCH algoritması

Birleştirilmiş hiyerarşik kümeleme metodu olan BIRCH Zhang tarafından geliştirilmiştir. Bu metod; kümeleme özelliği ve kümeleme özelliği ağacı (CF-Clustering Feature Tree) olmak üzere 2 kavrama dayanmaktadır. BIRCH yeni nesneleri artımlı ve dinamik olarak kümelemede etkilidir.

CF ağacı hiyerarşik kümeleme için kümeleme özelliklerini depolayan yükseklik dengeli ağaçtır. CF ağacı dallanma çarpanı ve eşik değeri olmak üzere 2 parametreye sahiptir. Dallanma çarpanı parametresi her bir yapraksız düğüm için çocuklarının maksimum sayısını ifade eder. Eşik değeri parametresi yaprak düğümlerde depolanan alt kümelerin maksimum çapını ifade eder. Bu iki parametre sonuçlanan ağacın boyutunu etkileyebilir [27]. BIRCH algoritması 2 aşamaya sahiptir. Birinci aşamada, bir başlangıç belleğindeki CF ağacını inşa etmek için veritabanını taranır. Çok seviyeli bir sıkıştırma olarak görülebilir ve verinin doğasında olan kümeleme yapısını korumaya çalışır. İkinci aşama da CF ağacının yaprak düğümlerini kümeleme için isteğe göre bir kümeleme algoritması uygulanır.

Birinci aşamada CF ağacı nesneler eklenerek dinamik olarak inşa edilir. Bu metod artımlı bir metoddur. Nesne en yakın yaprak girişine eklenir. Eğer alt kümenin çap değeri yaprak düğüme yapılan eklemeden sonra eşik değerinden daha büyük olursa, yaprak düğümü ve belki de diğer düğümler bölünür. Yeni nesneyi ekleme işleminden sonra ağacın köküne doğru bilgi geçirilir. Eşik değerinin değişmesiyle ağacında boyutu değişebilir. Eğer CF ağacının saklanması için gerekli olan bellek boyutu ana belleğin boyutundan daha büyükse daha küçük bir eşik değeri belirlenir ve CF ağacı yeniden inşa edilir. Eski ağacın yaprak düğümlerinden yeni bir ağaç inşa edilir. Böylece ağacın yeniden yapılanma işlemi tüm noktaların okunmasına gerek kalmaksızın yapılır. Bu B+ ağaçlarının oluşturulmasındaki ekleme ve bölme işlemine benzer. Bu yüzden ağaç inşa etmek için veri yalnızca bir kere okunur. Bazı metodlar

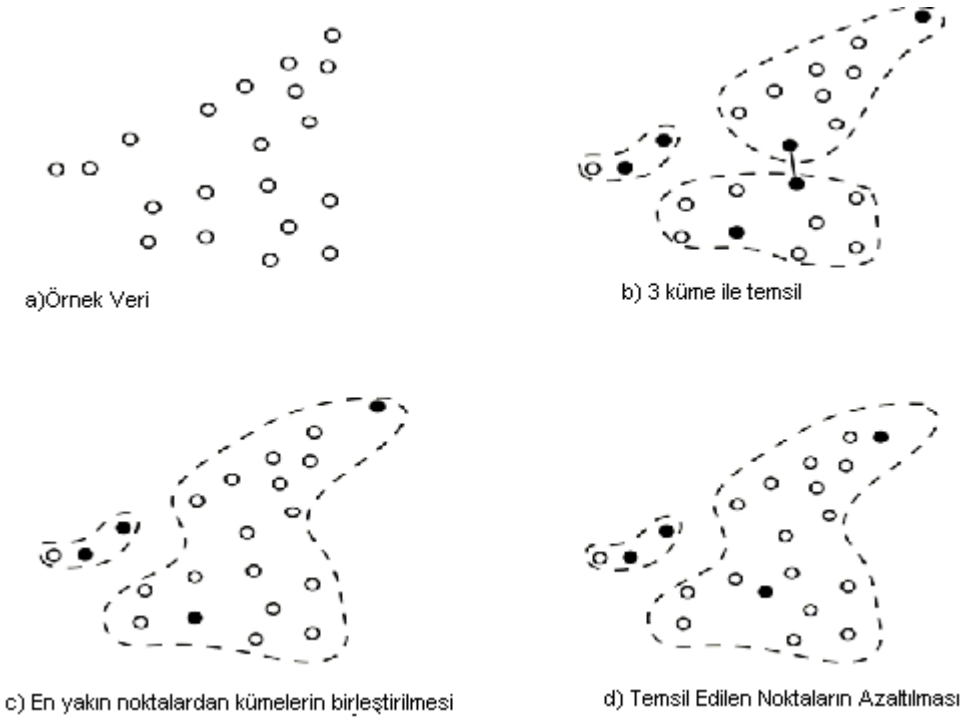
CF ağacının kalitesini geliştirmek için veri de ek taramalar gerçekleştirebilir. CF ağacı inşa edildikten sonra tipik bir bölümleme algoritması gibi herhangi bir kümeleme algoritması CF ağacı üzerinde 2. aşama için kullanılabilir [27].

BIRCH mevcut kaynaklarla en iyi kümeleri üretmeye çalışır. Ana belleğin limitiyle, I/O işlemleri için gerekli olan zamanı minimize etmek önemli bir husustur. BIRCH çok aşamalı kümeleme tekniğini kullanır. Veri kümesini bir kere tarama iyi bir kümeleme için kazanç sağlarken bir ya da daha fazla ek tarama kaliteyi geliştirmek için kullanılabilir. Algoritmanın hesapsal karmaşıklığı $O^{(n)}$'dir. N kümelenmiş nesnelerin sayısını ifade etmektedir. Deneyimler algoritmanın doğrusal ölçülebilirliğinin nokta sayısına uyulmasına ve verinin kümelenmesindeki kalitesine bağlı olduğunu göstermiştir. CF ağacındaki her bir düğüm, boyutuna göre girişlerin sınırlı sayıda bir kısmını tutabilir. Bir CF ağacı düğümü doğal kümeleme için asla uygun olmaz. Eğer kümeler küresel şekilde değilse BIRCH iyi bir şekilde kümelemeyi gerçekleştiremez. Çünkü BIRCH bir kümenin sınırını kontrol etmek için yarıçap veya çap fikrini kullanır [27].

CURE algoritması

Guha, Rastogi ve Shim tarafından ilk olarak SIGMOD 1998 konferansında sunulan CURE algoritması birleştirici bir kümeleme metodudur. Birçok kümeleme algoritması ya küresel şekilli ve eşit büyüklükteki kümelerde etkili çalışır ya da uzakta bulunan noktaların varlığında etkili çalışamazlar. CURE, centroid tabanlı ve temsilci-nesne tabanlı kümeleme yaklaşımlarını birleştirir. Kümeyi belirtmek için bir tek centroid ya da temsilci kullanmak yerine, küme sabit sayıda temsilci ile gösterilir. Kümenin temsilci elemanlarını belirtmek için önce iyi saçılmış nesneler belirlenir ve küme merkezine belirli bir oranda yaklaştırılır. Birçok temsilci nokta olması küresel şekilli olmayan noktalar için daha uygun kümeler oluşturulmasını sağlar. Daraltma işlemi ise dışarda kalan noktaların etkisini azaltmakta etkilidir. Büyük veritabanları ile başedebilmek için CURE rasgele örnekleme ve bölümleme kullanır. Önce rasgele örnek bölümlenir, daha sonra da bu bölümler kümelendirilir [38].

CURE algoritmasında ilk önce rasgele bir örnek kümesi S seçilir. S kümesi bölümlere ayrılır. Her bölüm kısmen kümelenir. Rasgele örnekleme yapılarak dışta kalan noktalar elenir. Kısmi kümeler, yeniden kümelenir. Her yeni kümenin temsilci noktası belirli oranda küme merkezine doğru kaydırılır. Bu noktalar kümenin şeklini temsil eder.



Şekil 4.21. Cure algoritmasının işleyişi.

CURE yüksek kaliteli kümeler oluşturur. Oluşan kümeler karmaşık şekilli ve farklı büyüklükte olabilir. CURE algoritması veritabanını sadece bir kez tarar, yani karmaşıklığı $O^{(n)}$'dir. CURE algoritmasının başarısı başlangıç parametrelerine büyük ölçüde bağlıdır. Bu nedenle en iyi kümeleme sonucunun bulunabilmesi için algoritmanın aynı veri seti üzerinde birkaç kez tekrarlanması gerekebilmektedir.

CURE algoritması sınıflandırlımış niteliklerle çalışamaz, bunun yerine ROCK (Robust Clustering Algorithm) algoritması kullanılır. ROCK algoritması iki kümenin birbirine benzerliğini, iki küme arasındaki toplam ara bağlantı (interconnectivity) miktarını hesaplayarak ölçer. İki küme arasındaki ara bağlantı miktarı kümeler arasındaki çapraz bağlantı miktarıdır. Her bir bağlantı ise iki noktanın ortak

komşularının sayısıdır. Yani kümelerin benzerliği, farklı kümelerdeki noktaların ortak komşularının sayısı ile belirlenir.

CHAMELEON algoritması

Chameleon algoritması dinamik modellemeyi hiyerarşik kümelemeye uygular. Kümeleme işleminde eğer iki küme arasındaki ara bağlantı ve yakınlık değerleri kümelerin kendi içlerindeki ara bağlantı ve yakınlık değerleri ile yüksek oranda ilişkili ise bu kümeler birleştirilir. Bu birleştirme işlemi doğal ve homojen kümelerin ortaya çıkarılmasını sağlar ve benzerlik fonksiyonunun tanımlanabildiği her veri tipi için uygulanabilir. Chameleon, hem CURE hem de ROCK algoritmalarının eksik taraflarını kapatır. CURE algoritması iki küme arasındaki toplam ara bağlantı miktarını göz ardı ederken, ROCK algoritması da iki kümenin birbirine ne kadar yakın olduğunu göz ardı eder.

Chameleon algoritması önce graf bölümleme algoritması kullanarak veriyi çok sayıda küçük alt kümeye ayırır. Daha sonra bu küçük alt kümeleri birleştirerek özel kümeleri oluşturur. Birbirine en çok benzeyen alt kümeleri belirlerken hem alt kümelerin kendi içlerindeki ara bağlantı ve yakınlık değerlerini hem de alt kümelerin kendi aralarındaki alt bağlantı ve yakınlık değerlerini kullanır. Böylece algoritma kendini verinin yapısına göre ayarlar.

Yoğunluk tabanlı metotlar

Yoğunluk tabanlı metotlar, şekilsiz kümelerin bulunması için geliştirilmiştir. Bu metotlar, veri uzayındaki düşük yoğunluklu bölgelerle birbirinden ayrılan yoğun veri bölgelerini küme olarak kabul eder. Yoğunluk tabanlı metotlar, sıradışı ve gürültülü verilerin bulunmasında etkili olan metotlardır. Bu tip algoritmalarda küme sayısının önceden belirtilmesine gerek yoktur [27,29]. DBSCAN (Density Based Spatial Clustering of Applications with Noise) ,OPTICS (Ordering Points To Identify the Clustering Structure), DENCLUE (DENsity based CLUstering) ve TURN algoritmaları yoğunluk tabanlı metotların başlıcalarıdır.

DBSCAN Algoritması

İlk yoğunluk tabanlı metot olan DBSCAN (Density Based Spatial Clustering of Applications with Noise) Ester, Sander Kriegel ve Xu tarafından KDD'96 konferansında sunulmuştur [24]. DBSCAN algoritması, veri noktalarını uzayda oluşturdukları çeşitli yoğunluklardaki bölgelere göre kümelere ayırmaktadır. Yoğun bölgeler kümeleri oluştururken, sıradışı ve gürültülü verilerin oluşturduğu seyrek bölgeler tespit edilerek kümelere alınmaz. Şekilsiz yada farklı şekillerdeki kümelerin bulunmasında etkili bir algoritmadır. En büyük dezavantajı kümelerin yoğunluğunun tanımlanmasında kullanılan giriş parametrelerine karşı duyarlı olmasıdır [47,48].

OPTICS Algoritması

OPTICS (Ordering Points to Identify the Clustering Structure) algoritması, DBSCAN algoritmasının parametrelere karşı duyarlılığını ortadan kaldırmak üzere, SIGMOD'99 konferansında Ankerst, Breunig, Kriegel ve Sander tarafından sunulan bir algoritmadır [49]. OPTICS algoritmasında, önceden belirlenmiş bir yarıçapa sahip belirgin kümeler oluşturmak yerine, yoğunluk tabanlı kümeleme yapısını gösteren artan (augmented) bir küme sıralaması oluşturulur. Bu sıralama her veri noktasına ait çekirdek uzaklığı ve uygun erişilebilirlik uzaklığı değerlerini taşıyan bir sıralamadır. Bu küme sıralaması veri setindeki her küme seviyesi için bilgi taşımaktadır. Yoğun bir kümede olması gereken en az nokta sayısı (MinPts) ve tanımlanan küme içi uzaklık mesafesi (Eps) değerlerine bağlı olarak oluşturulan küme sıralaması sadece bir kez taranır ve MinPts ve küme uzaklığına (Eps'≤Eps) bağlı olarak yoğunluk tabanlı kümeler bulunur. Küme elemanları erişilebilirlik uzaklıkları ve çekirdek uzaklıklarına göre bu kümelere yerleştirilir. OPTICS algoritmasında, küme sıralamaları grafiksel olarak gösterilebilir. OPTICS etkileşimli ve otomatik bir kümeleme algoritmasıdır [27,35,39,49].

DENCLUE Algoritması

Hinneburg ve Keim tarafından KDD'98 konferansında sunulan DENCLUE (DENsity based CLUstEring) algoritması bir dizi yoğunluk dağılım fonksiyonuna

dayalıdır. DENCLUE algoritmasının çalışmasında etkili olan üç farklı kavram bulunmaktadır:

- Her veri noktasının kümelemedeki etkisi, bu noktanın kendi komşuluğundaki etkisini tanımlayan etkileme fonksiyonu (influence function) ile modellenir.
- Veri uzayının genel yoğunluğu tüm veri noktalarının etkileme fonksiyonlarının toplamı ile bulunur.
- Yoğunluk merkezleri genel yoğunluk fonksiyonunun yerel maksimum noktalarıdır. Bu merkezler bulunarak kümeler matematiksel olarak tanımlanabilir [50].

TURN Algoritması

TURN* algoritması genel bir algoritma ve bir yardımcı algoritma ile gerçekleşmektedir. Çözünürlüğe bağlı (resolution dependent) TURN-RES yardımcı algoritması, bir kümeleme oluşturur ve sonuç olarak kümeyle birlikte bu kümelemenin özellikleri değerlerini bulur. TURN_RES ile elde edilen çözünürlük bilgilerinden önemli ve uygun çözünürlükleri bulan otomatik bir metottur. Herhangi bir çözünürlük için, TURN-RES algoritması noktaların her nokta etrafında ne kadar sıkı yerleştiklerini hesaplayarak yüksek değerlere sahip olan noktalar dahili (interval) olarak işaretlenir. Aynı zamanda, her noktanın etrafındaki yakın noktalar da işaretlenir. Kümeleme işlemi tüm yakın komşu noktaları kendi dahili noktalarına yerleştirir. Bu işlem dahili noktalar bitinceye kadar devam eder [51].

Izgara tabanlı metotlar (Grid Based Methods)

Izgara tabanlı kümeleme metotları, veri uzayını sonlu sayıda karelere bölerek kümeleme işlemlerinin tümünün gerçekleştirileceği bir ızgara yapısı elde etmektedir. Bu yaklaşımın en büyük avantajı, performansının sadece ızgara çözünürlüğüne diğer bir deyişle ızgaradaki kare sayısına bağlı olup veritabanının büyüklüğünden bağımsız olmasıdır. Bu nedenle genellikle çok sayıda veri noktası içeren yüksek yoğunluklu veritabanlarında kullanılır. STING (Statistical Information Grid), WaveCluster, CLIQUE (Clustering High-Dimensional Space) algoritmaları ızgara tabanlı metotların başlıcalarıdır.

STING Algoritması

Wang, Yang ve Mutz tarafından 1997 yılında sunulan STING (STatistical INformation Grid), algoritması veri uzayının dikdörtgen şeklindeki hücrelere bölündüğü ızgara tabanlı bir kümeleme algoritmasıdır [27,52]. Bu algoritmada, piramite benzer hiyerarşik bir yapı oluşturulur. Her hücrenin içinde bu hücreye ait ortalama, standart sapma, minimum, maksimum ve dağılım türü istatistiksel bilgileri saklanır. Her hücrenin içindeki istatistiksel bilgiler aşağıdan yukarıya doğru hesaplanır ve sorguların yanıtlanmasında kullanılır. Sorgular işlenirken ise hiyerarşik yapı yukarıdan aşağıya doğru incelenir. Alt tabakalara inildikçe kümelenme alanları daha hassas olarak ortaya çıkar.

STING algoritmasının ızgara yapısı, paralel işlemciliği ve güncelleme işlemlerinin kolay bir şekilde gerçekleşmesini sağlamaktadır. STING, veritabanını bir kere, hücrelerin istatistiksel parametrelerini hesaplamak için okur. Dolayısıyla küme oluşturma işleminin hesaplanabilir karmaşıklığı n toplam nesne sayısı olmak üzere, O(n)'dir. Hiyerarşik yapı oluşturulduktan sonra alt seviyedeki toplam hücre sayısı (g) n'den küçük bir değerde olmak üzere sorgu işlem süresi $O^{(g)}$'dir. Bu yüzden algoritmanın kümeleri bulma performansı oldukça yüksektir.

WaveCluster Algoritması

Sheikholeslami, Chatterjee ve Zhang tarafından VLDB'98 konferansında sunulan WaveCluster algoritması, wavelet dönüşümüne (Wavelet Transform) dayalı bir kümeleme yaklaşımıdır [53]. WaveCluster, öncelikle veri uzayını çok boyutlu ızgara yapısına yerleştirir ve daha sonra gerçek veri uzayı çok boyutlu sinyaller olarak kabul edilerek, ızgaradaki hücrelere sinyal işleme teknikleri (Wavelet dönüşümü) uygulanır [39]. WaveCluster algoritması ızgara-tabanlı ve yoğunluk-tabanlı bir algoritmadır [27]. İyi bir kümeleme metodunun pek çok gereklerini sağlamaktadır: Büyük veri tabanları üzerinde etkilidir, şekilsiz kümeleri bulur, istisna ve gürültülü verileri atar, küme sayısı veya komşuluk yarıçapı gibi giriş parametrelerine gerek yoktur. Ancak bu algoritma verilerin sıralamasından etkilenmektedir ve çok boyutlu veritabanlarında yeteri kadar başarı göstermemektedir [27].

CLIQUE Algoritması

Aggrawal, Gehrke, GunoPulos ve Raghavan tarafından SIGMOD'98 konferansında sunulan CLIQUE (Clustering High-Dimensional Space) algoritması yoğunluk tabanlı ve ızgara tabanlı algoritmaları birleştirmektir. Çok boyutlu veritabanlarında yoğun olmayan kümelerin bulunmasını sağlar. Çok boyutlu veri uzayındaki noktalar genellikle dağınık olarak yerleşmektedir. CLIQUE metodunun kümeleme yöntemi, bu uzaydaki yoğun ve seyrek bölgeleri tanımlar ve böylece veri dizisindeki genel örüntü dağılımını tespit etmiş olur. Her bölüm eğer bu bölümdeki toplam veri noktaları sayısı eşik değerden büyükse yoğun olarak kabul edilir. CLIQUE metodunda, bir küme, birleşmiş yoğun bölümler dizisi olarak tanımlanmaktadır.

CLIQUE algoritması çok boyutlu ve noktaların dağınık olarak yerleştiği seyrek veri uzayında verilerin kümelenmesinde yüksek bir performans göstermektedir. Ancak bu algoritma veri sıralamasına karşı duyarlıdır. Veritabanının büyüklüğü ile doğru orantılıdır ve veri boyutu (dimension) arttıkça daha iyi bir ölçeklenebilirlik kazanmaktadır.

Model tabanlı metotlar

Model-tabanlı kümeleme metotları veri ile matematiksel modeller arasındaki ilişkiyi kullanmaktadır. Bu metotlar verinin veri uzayında yerleşiminin olasılık teorilerinin karışımından oluşan bir mantık ile gerçekleştiğini kabul etmektedir. Model-tabanlı kümeleme metotları, istatistiksel ve yapay zeka olmak üzere iki önemli yaklaşıma dayanır [27].

İstatistiksel Yaklaşım

Kavramsal kümelemede kümeleme ve sınıflandırma işlemlerinin her ikisi de kullanılmaktadır. İlk olarak veritabanı üzerinde kümeleme işlemi gerçekleştirilir ve ardından bulunan kümelerin genel özelliklerini taşıyan bilgiler bulunur. Kavramsal kümelemede kavram ve kümeleri belirlemek için genellikle olasılık ölçütlerini kullanan istatistiksel yaklaşımları kullanılmaktadır [27].

COBWEB popüler ve basit bir kavramsal kümeleme metodudur [27]. Bu modelin giriş nesneleri kategorik özellik-değer çiftleri ile tanımlanır. COBWEB, sınıflandırma ağacına benzer bir hiyerarşik kümeleme oluşturur.

Yapay Zeka Yaklaşımı

Yapay zekâ yaklaşımı her kümeyi bir örnek olarak tanımlar. Bir örnek, kümenin prototipi gibi davranır ve bu prototipin herhangi bir nesneye benzemesi gerekmemektedir. Yeni eklenen bir nesne, bazı uzaklık ölçütlerine göre en çok benzedikleri örneğe ait kümeye yerleştirilir. Bir kümedeki nesneye ait özellikler o kümenin örneğinin özellikleri kullanılarak tahmin edilir.

Sınırlılık tabanlı metotlar

Sınırlılık tabanlı metotlar fiziksel sınırlılıkların bulunduğu uzaysal verilerin kümelenmesinde kullanılmaktadır [29,35]. COD (Clustering with Obstructed Distance), COD-CLARANS, AUTOCLUST+ ve DBCluC algoritmaları sınırlılık tabanlı kümeleme metotlarının başlıcalarıdır.

COD ve COD-CLARANS Algoritmaları

COD (Clustering with Obstructed Distance) algoritması, kümeleme algoritmalarının sonuçlarını etkileyen fiziksel koşulların (dağlar, nehirler, siteler,vb..) kontrol edilmesi için geliştirilmiştir [36]. COD yaklaşımında, fiziksel engeller çokgenlerle gösterilir. Bu yaklaşımda iki nokta arasındaki uzaklık olarak, bu iki nokta arasında, herhangi bir engel olmadan ölçülen en kısa uzaklık değeri alınır. Kümeleme işlemi veri noktalarının aşağıdaki hata parametresinin değeri en az olacak şekilde bölümlenmesiyle gerçekleştirilir.

$$\sum_{i=1}^{k} \sum_{p \in c_i} d^2(p, m_i)$$

- E : hata parametresi,
- p : veri noktaları $\{p_1,p_2,\ldots,p_n\}$dizisi,
- k : küme sayısı,

- C : kümeler $\{C_1,C_2,\ldots,C_k\}$,
- m : fiziksel engel sayısı.

Fiziksel engellerin çokgenlerle gösterilmesi farklı sonuçlar oluşturmaktadır. Bazı engeller dar ve uzun bir dörtgen yapısı oluştururken bazıları geniş bir yedigen oluşturabilir. Bu sorunun çözülebilmesi için COD-CLARANS algoritması sunulmuştur [36]. Bu algoritma, kümeleme işleminin performansını arttırmak için iki ayrı teknik kullanmaktadır. Birinci teknikte, BIRCH [27] ve CHAMELEON [54] algoritmalarına benzer bir ön-kümeleme işlemi gerçekleştirilerek mikro-kümeler oluşturulur. Mikro-kümeler birbirine çok yakın ve aynı kümede olabilecek kümeler hakkında özet bilgiler taşır. Kümeleme işleminin mikro-kümeler üzerinde gerçekleştirilmesi işlemlerin ana bellekte gerçekleşerek nesnelerle küme merkezleri arasındaki uzaklık hesaplamalarının maliyetinin azalmasını sağlar. İkinci teknikte, arama uzayını küçültmek için hatanın karesi fonksiyonunun (e) alt sınırı kullanılır. Bu iki teknik algoritmanın çalışma süresini azaltarak kümeleme kalitesini arttırmaktadır [35].

AUTOCLUST+ Algoritması

AUTOCLUST+ algoritması, bir grafik bölümleme algoritması olan AUTOCLUST'ın geliştirilmiş bir sürümüdür [27]. AUTOCLUST algoritmasında tüm veri noktaları Delaunay Diagramı üzerinde gösterilir [27]. Delaunay Diagramında tüm veri noktaları veri noktaları arasındaki uzaklıkların standart sapmalar ve ortalama değerlere bağlı kenarlarla gösterildiği ve bu kenarlarla birbirine bağlandığı bir grafiktir. Kısa kenarlarla birbirine bağlanan noktalar kümelendirilir. Diğer kenar çizgileri kümeler arası ve küme-istisna arası ilişkileri göstermektedir. AUTOCLUST algoritmasında kümeleme işlemi sadece küme oluşturan noktalar kalacak şekilde kenarlar atılarak gerçekleştirilir. AUTOCLUST+ algoritmasında ise fiziksel bir engel, kenarları Delaunay Diagramından atan bir dizi çizgi parçası ile modellenir. Daha sonra, fiziksel bir engelin çizgi parçaları ile engellenen kenarlar Delaunay diagramından atılır. Atılan bir kenar iki nesne arasındaki en yakın yola eşit bir varyant yol ile değiştirilir.

DBCluc Algoritması

DBCluc algoritması DBSCAN algoritmasında türetilmiş yoğunluk tabanlı bir algoritmadır [27]. Şekilsiz kümeleri bulup istisna verileri ayrırırken sadece fiziksel nesnelerin neden olduğu ayrıklığı değil köprülerin neden olduğu bağlantıları da dikkate alır. Fiziksel engeller ve köprüler çokgenlerle gösterilir. Ancak köprü-çokgenlerinin, giriş noktaları gibi özel kenarları vardır.

4.2.7. Kümeleme analizinin kullanıldığı alanlar

Kümeleme analizi birçok farklı alanda kullanılmaktadır. Kümeleme analizinin kullanıldığı belli başlı alanlar aşağıda belirtilmiştir:

Biyoloji: Yaşayan varlıkların taksonomosini oluşturmak için biyologlar uzun yıllar harcamışlardır. Bunlar; alem, film, sınıf, tür, aile gibi kümelerdir. Son zamanlarda biyologlar var olan genetik bilginin analizi için kümeleyi uygulamışlardır. Örneğin; kümeleme benzer fonksiyonlara sahip gen gruplarını bulmak için kullanılmaktadır.

Bilgi Çıkarma : Web milyonlarca web sayfasından oluşur ve arama motoruna yapılacak bir sorgu binlerce sayfa ile geri dönebilir. Kümeleme bu arama sonuçlarını küçük sayıdaki kümelere gruplamak için kullanılabilir. Örneğin; film için sorgu yaptığımızda web sayfalarını eleştiriler, fragman, yıldızlar ve tiyatrolar kategorilerine gruplayabilir. Her kategori(küme) hiyerarşik bir yapı oluşturarak alt kategorilere bölünebilir. Bu yapı kullanıcıya daha ileri keşifler için yardımcı olur.

İş : Kuruluşlar, müşterilerilerinin ve potansiyel müşterilerinin hakkında büyük miktarda bilgi toplarlar. Kümeleme ilave analizler yapmak ve piyasa faaliyetleri için müşterileri küçük gruplara bölmede kullanılabilir.

Özetleme : Regreasyon ve PCA gibi birçok veri analizi teknikleri zaman ve alan karmaşıklığına sahiptir ve bu gibi teknikler geniş veri kümeleri için uygun değildir. Giriş veri kümesine bu algoritmaları uygulamak yerine sadece küme prototiplerini içeren azaltılmış veri kümesine başvurulabilir. Analiz çeşitine dayalı olarak prototiplerin sayısı ve veriyi temsil eden prototipin doğruluğu, analizi tüm veri

üzerine uygulayarak elde edilecek sonuçlarla prototip üzerine uygulanarak elde edilecek sonuçlarla karşılaştırılabilir.

Sıkıştırma : Küme prototipleri veri sıkıştırmak için kullanılabilir. Özellikle, her bir küme için prototipleri içeren bir tablo yaratılır. Örneğin her bir prototipe tablo için pozisyonunu belirten bir tamsayı değeri atanır. Her bir nesne onun kümesi ile ilişkili olan prototipin indeksi ile gösterilir. Sıkıştırmanın bu çeşidi vektör niceleme (vector quantization) olarak bilinir ve resim, ses ve video verileri gibi veri nesnelerinin birçoğunun birbirine çok benzediği, az veri kayıplarının kabuledilebilir olduğu ve veri büyüklüğünde önemli azalmaların tercih edildiği durumlarda genellikle uygulanır.

5. UYGULAMA

Yaygınlaşmış hali ile email olarak tanımlanan e-posta ya da elektronik posta, verinin dağıtım kanalları üzerinden iletilmesi ile yapılır. Dağıtım kanalları için genelde internet üzerinden yapılan haberleşme kastedilmektedir. E-posta ile haberleşme yönteminin ilk zamanlarında metin haberleşmesi yapılırken artan bant genişlikleri ve de çoklu ortam veri türlerinin de gelişimi ile birlikte gönderilen veri içeriğinde artış meydana gelmiştir.

E-posta ile haberleşme spam ve spam olmayan olarak ikiye ayrılmaktadır. Spam olmayan e-postalar alıcıların gönderenini bildiği ya da e-posta içeriğinden rahatsız olmadığı düşünülen e-postalardır. Spam e-postalar ise ticari amaçlı ya da toplu duyurularda bulunmak isteyen kişi ya da organizasyonların çok sayıdaki kişiyi bilgilendirmek, kişisel bilgilerini ele geçirmek için tercih ettiği yöntemdir.

Veri madenciliği (DM) büyük veri tabanlarından önceden öngörülmeyen bilgiyi çıkarmanın ve sonuçları karar vermeye uygulamanın çok aşamalı sürecidir. Veri madenciliği araçları veriden örüntüleri algılar ve onlardan ilişkiler ve kurallar çıkarır. Çıkarılmış bilgi tahmin ve sınıflandırma modellerinde kullanılmak üzere veri üzerindeki ilişkiler tanımlanarak uygulanabilir. Bu örüntüler ve kurallar karar vermeye rehberlik edebilir.

Bugün veri madenciliği; veri tabanı teknolojisi, yapay zeka, istatistik, örüntü algılama, bilgi kazanımı, yüksek performans hesaplama ve veri görselleştirme gibi adımlar ile iç içe geçmiş çok disiplinli bir alandır.

Spam maillerin filtrelenmesinde değişik yöntemler kullanılmaktadır. Spam e-postanın veri madenciliği yöntemleri tespiti ve filtrelenmesine yönelik çalışmalar, veri madenciliği ve spam ile mücadeleye yönelik düzenlenen Data Mining Cup, CEAS gibi uluslararası çalışmaların temel konularından birisi haline gelmiştir. Söz konusu çalışmalarda; karar ağaçları, kümeleme metodları, bayesian analizi, yapay sinir ağları gibi birçok yöntem spam tespiti ve spam filtrelemede uygulanmıştır [57,58,59,61,62].

Bu çalışma kapsamında e-postaların çevrimdışı olarak spam ve spam olmayan şeklinde sınıflandırılarak tespiti amaçlanmıştır. Bu çalışmada, literatürde çalışmalarında araştırmacılar tarafından kullanılan, KDD data mining cup (Knowledge Discovery in Databases) ve CEAS (Collaboration, Electronic messaging, Anti-Abuse and Spam Conference) çalışmalarında veri seti olarak kullanılan Enron e-posta veri seti tercih edilmiştir.

Spam filtreleme için hali hazırda birçok metot kullanılmaktadır.

- Kural tabanlı filtreler ile izin verilen e-postaların erişimine olanak sağlanır,
- Kara liste yaklaşımı ile spam gönderen IP (Internet Protocol) adreslerinden ve e-posta adreslerinden gelecek olan erişiminin engellenmesi sağlanır,
- Beyaz liste yaklaşımı ile izin verilen IP adresi ve e-posta adreslerinden gelen e-postaların erişimine izin verilmesi sağlanır,
- Doğrulama bilgisi veritabanı (checksum database) yaklaşımı ile spam olduğu tespit edilen e-postaların doğrulama bilgisi değeri hesaplanarak bir veritabanına kaydedilir. Sonrasında gelen e-postaların doğrulama (checksum) bilgisi hesaplanarak kayıtlı spam e-postanın doğrulama bilgisi ile eşleşmesi durumunda erişimin engellenmesinin sağlanması şeklindedir.

Literatürde yer alan çalışmalar makine öğrenimi, veri madenciliği ve anti spam çalışmaları arasında yer almaktadır. Her yıl düzenli olarak düzenlenen CEAS (Collaboration, Electronic messaging, Anti-Abuse and Spam Conference) ve KDD (Conference on Knowledge Discovery and Data Mining) çalışmalarında spam e-posta tespiti ve engellenmesine yönelik araştırmalara yer verilmektedir [57,58,59,61,62].

Söz konusu araştırmalarda veri madenciliği ya da makine öğrenimi alanlarında Bayesian filter, yapay sinir ağları, kümeleme analizi, genetik algoritmalara yer verilmiştir.

Bu çalışma kapsamında danışmansız öğrenme tekniklerinden olan k-means kümeleme metodu ile danışmanlı öğrenme metodu olan yapay sinir ağlarına yer verilmiştir.

5.1. Kullanılan Veri Seti

Bu tez kapsamında Amerika Devlet Enerji Düzenleme Kurumu (FERC - Federal Energy Regulatory Commission) tarafından dünya genelinde enerji sektöründe büyük bir şirket olan Enron firması çalışanlarına ait yayımlanmış olan, e-posta veri seti kullanılmıştır. Veri seti içerisinde 151 kullanıcıya ait 517431 e-posta yer almaktadır [63]. Söz konusu veri seti Carnegie Mellon Üniversitesi tarafından düzenlenerek araştırmacılar için yayınlanmıştır [64].

Söz konusu veri setleri araştırmacılar tarafından Ham_Messages, Pre_Processed, Spam_Messages konu başlıkları şeklinde düzenlenerek araştırmacıların dikkatine sunulmuştur.

Bu konu başlıklarından ham_messages spam olmayan e-postaları (6 ana dizin ve 19088 adet dosya-email), spam_messages spam e-postaları (3 ana dizin ve 32970 adet dosya-email), pre_processed spam ve spam olmayan e-postaları (6 ana dizin ve toplamda 33722 adet dosya-email) olarak sunulmuştur. Bu şekilde düzenlenmiş olan veri setleri CSMining Group tarafından yayınlanmıştır [65]. Her bir e-posta içerisinde gönderici ve alıcı adresleri, tarih bilgisi, e-posta içeriği yer almaktadır.

5.2. Kullanılan Yöntem

Sınıflandırma sınıfı belli olmayan nesnelerin sınıfını tahmin etmek için veri sınıf ve konseptlerini tanımlayan ve ayıran model bulma sürecidir. Şekil 5.1 sınıflandırma sürecindeki genel yapıyı göstermektedir. Örnekte, nesne e-postaya karşılık gelir ve nesne sınıfı etiketi e-posta kategorisine karşılık gelir. Her e-posta içerisinde yer alan anahtar kelimeler, gönderici-alıcı detayları, mesajın başlığı vb. mesajın kategorisini belirlemede kullanılan nitelikleri ifade etmektedir.

Spam filtreleme projesinde kullanacağımız yöntem spam olduğu/olmadığı belirlenmiş e-postaların analiz edilerek öğrenme yoluyla kendini geliştiren bir sistemin kurulmasıdır.

K-means, Naive Bayes, YSA vb. diğer yöntemlerin de uygulanabilmesi amacı ile ortak girdi yapısına sahip, veri setini yansıtan uygun niteliklerin seçilerek nitelik uzayı oluşturulmuş ve söz konusu yöntemlerin uygulanabilirliği amaçlanmıştır.

Satırları öğrenme setindeki e-postaları, sütunları ise belirlenen nitelikleri ve sonuç bilgisini (0: SPAM e-posta veya 1:normal e-posta) içerecek olan nitelik uzayını temsil edecek olan matris elde edilerek istenilen örüntü tanıma yöntemleri uygulanabilir hale gelmiştir.

Sözlük çalışması

Sözlük çalışması kapsamında İnternet'te yer alan ücretsiz olarak kullanıma sunulan spam listelerinden faydalanılmıştır.

Nitelik Matrisi Çalışması

Nitelik matrisi çalışmasında spam ve spam olmayan e-posta olarak sınıflandırılmış olan, literatürdeki çalışmalarda kullanılan Enron şirketine ait (Enron e-mail data set) e-postalar kullanılmıştır. Nitelik matrisini oluşturmada aşağıdaki adımlar uygulanmıştır:

- Her bir e-posta, nitelik matrisinde bir satıra karşılık gelmektedir.
- E-postadaki başlık bilgileri anlamlı şekilde çözümlenmiştir:
    - "To:", "Cc:" alanlarında geçen adres sayısı nitelik olarak değerlendirilmiştir.
    - "Subject:" alanı e-posta içeriği gibi kabul edilerek analiz edilmiştir.
- E-posta içeriği ve konusu çözümlenerek kelimelerine ayrılmış, eğer e-posta html formatında ise html etiketlerinden temizlenmiştir.
- Ayrıştırma sonucu e-postadan elde edilen kelimelerin, her bir kelimenin değişik ekli hallerini içeren WorldNet verisi sözlükte olup olmadığı kontrol

edilmiştir. Spam olduğu düşünülen, belirlenen sözlükte geçen kelimelerin e-postaların tamamında geçme sayıları tutulmuştur.

- Sözlükte geçmeyen kelimeler ise akıllı bir benzetme işlemine sokularak açık kaynaklardan elde edilmiş olan spam kelimeleri ile karşılaştırılarak nitelik uzayına eklenmiştir. Eğer herhangi bir eşleşme olmazsa, eşleştirilemeyen kelime sayısı da ayrı bir nitelik olarak kullanılmıştır.
- Metinde geçen url sayısı da bir nitelik olarak değerlendirilmiştir. Sözlükte geçen kelimeler ise nitelik uzayımız için aday konumunda olup, tüm öğrenme setinde kullanım sıklığı belli bir yüzdenin üzerinde olarak daha çok spam e-postalarda ya da spam olmayan e-postalarda karşılaşılan kelimeler nitelik uzayımıza eklenmiştir. Bu işlemde bağlaçlar gibi genel olarak çok sık kullanılan kelimeler elenmiştir, çünkü hem spam hem de normal e-postalarda da ortak olarak rastlanmıştır.
- Worldnet sözlüğünde kelimelerin isim, sıfat vb türleri de yer almaktadır. E-postada geçen sıfat-zarf sayısı da bir nitelik olarak eklenmiştir. Bunun nedeni spam maillerde övücü ifadelere sıklıkla rastlanmış olunmasıdır.
- Sonuçta nitelik matrisimizin kolonlarında aşağıdaki niceliklerin geçiş sıklık değerlerine yer verilmiştir:
    - Ünlem işareti sayısı,
    - URL sayısı,
    - Dolar işareti sayısı,
    - Zarf sayısı
    - Sıfat sayısı
    - İsim sayısı
    - Yüklem sayısı
    - Özel isim sayısı
    - Sözlükte geçmeyen kelime sayısı
    - Spam URL sayısı
    - E-posta içerisinde geçen toplam kelime sayısı
    - E-postada içerisindeki toplam alıcı sayısı
    - ‘*’,’>’,’<’ gibi özel karakter sayısı

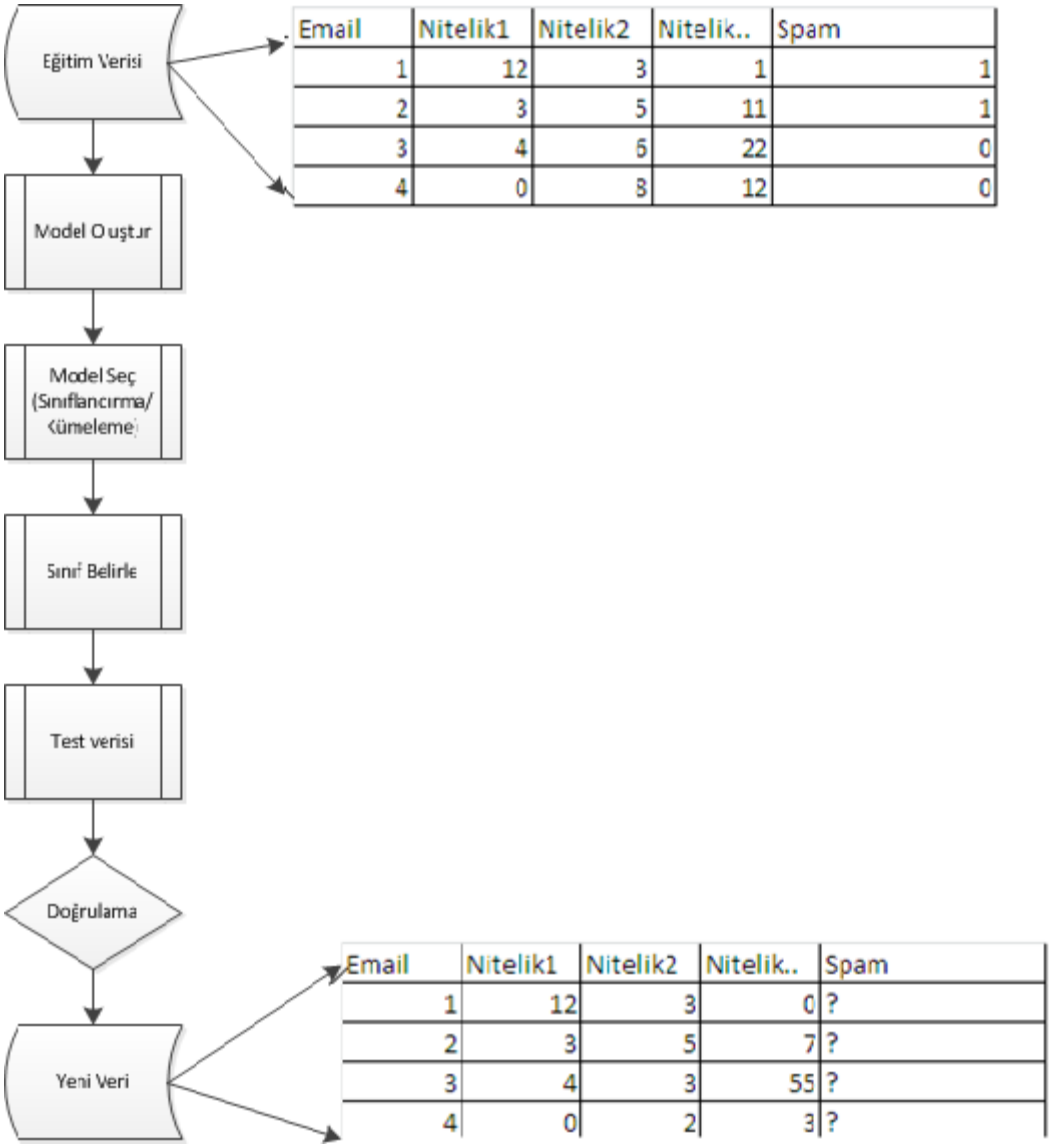
- Gizlenmek istenen kelime sayısı
- E-posta içerisinde geçen sayı sayısı
- E-posta içerisinde geçen eklenti sayısı
- E-posta içerisinde geçen spam kelime sayısı

Nitelik matrisini spam ve spam olmayan e-postalar üzerinde ayrı ayrı uygulayarak iki ayrı veri seti elde edilmiştir.

Nitelik matrisi oluşturulduktan sonra danışmansız öğretim tekniği olan K-means algoritması uygulanarak iki temel sınıf oluşturulmuş ve sonrasında verilen veri setlerinin kümeleme analizi yapılmıştır. Son kolona göre doğruluk sınaması da yapılarak belirlenen sınıf merkezleri yeni gelecek e-posta adaylarına uygulanarak hangi sınıfa yakınsa o sınıfa ayrılarak spam olup olmadığı tespit edilmiştir.

İkinci olarak, sınıf değer kolonu da (son kolon) kullanılarak, danışmanlı öğrenme tekniği olan YSA uygulanmış, verilen e-posta seti ile sonuç arasındaki doğru bulma oranları elde edilmiştir.

Çalışma kapsamında kümeleme analizi açık kaynak kodlu veri madenciği aracı olan WEKA programı üzerinde uygulanarak sonuçlar elde edilmiştir.



Şekil 5.1. Spam e-posta sınıflandırmanın genel yapısı

Çalışma aşamaları aşağıdaki şekilde olmuştur;

- Çalışmalarda kullanılmak üzere email veri setleri araştırılmış, literatürdeki çalışmalarda sıklıkla geçen Enron e-posta veri seti kullanılmıştır. Düzenlenmiş olan veri setleri CSMining Group tarafından yayınlanmıştır [65].
- Spam listeleri internet üzerindeki değişik kaynaklar üzerinden elde edilmiştir. Sonrasında sozluk.exe programı aracılığı ile kelime listeleri tek bir dosyada birleştirilmiştir. Söz konusu dosya içerisinde 434949 kayıt mevcuttur. Bu

dosyadaki son alan kelime grubunun türünü temsil etmektedir. Kelime türü 0-255 arasında değer almıştır.

- E-posta verileri üzerinden nitelik matrisini oluşturmak için mailparser.exe programı kullanılmıştır. Bu program kendisine gösterilen dizin içerisindeki tüm alt klasörlerdeki emailleri inceleyerek, belirlenen 19 niteliğe ait geçme sayılarını ve normalize edilmiş değerlerini hesaplamaktadır. Söz konusu nitelik matrisi elde edikten sonra veri madenciliği yöntemleri uygulanabilir duruma gelinmiş olunmaktadır.
- YSA yöntemi nitelik matrisine uygulanarak sonuçlar elde edilmiştir.
- Kümeleme metodlarından k-means yöntemi nitelik matrisine uygulanarak sonuçlar elde edilmiştir.

Nitelik matrisi çalışması sonrasında elde edilen örnek kayıtlar Çizelge 5.1.'de yer almaktadır.

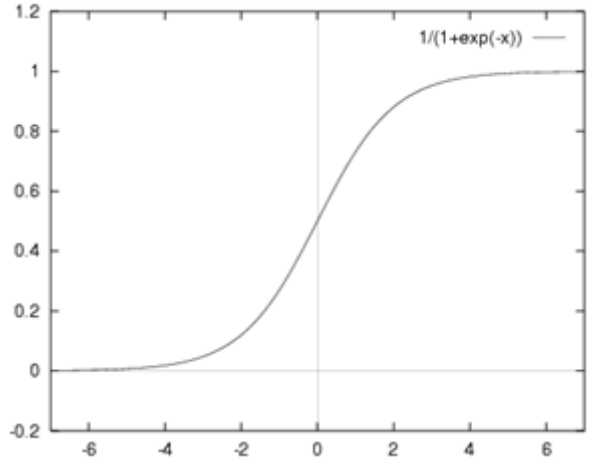
Çizelge 5.1. Nitelik matrisi

UnlemSayisi	UrlLinkSayisi	DolarSayisi	AdvSayisi	AdjSayisi	NounSayisi	VerbSayisi	OzelIsimSayisi	SozlukDisiSayisi	SpamPhraseSayisi	SpamWordSayisi	SpamUrlSayisi	ToplamKelimeSayisi	E-postaAdresSayisi	ExtraKarakterSayisi	GizlemeCabaSayisi	SayiSayisi	EklentiSayisi	NadirSpam	SPAM
4	1	1	1	6	18	0	3	2	1	30	16	61	3	8	24	1	0	24	1
4	0	0	4	7	85	3	47	24	1	195	45	267	2	5	112	0	0	155	1
3	1	0	0	0	19	1	2	2	0	21	10	42	5	12	14	3	0	17	1
2	1	0	0	2	9	0	0	1	0	16	7	30	6	5	14	0	0	14	1
4	1	1	0	10	44	1	5	5	0	72	25	104	2	11	50	5	0	57	1
5	1	2	0	5	15	1	1	2	2	27	8	51	2	8	20	3	0	22	1
0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	1
12	1	7	2	26	97	7	1	4	0	178	31	187	3	38	104	13	0	146	1
2	0	0	0	0	14	0	1	0	0	22	2	32	5	12	18	3	0	19	1
4	0	0	0	4	25	0	4	3	0	69	20	93	6	23	52	5	0	60	1
10	0	5	1	16	67	1	7	5	1	109	27	145	4	31	65	9	0	79	1
10	21	0	0	0	0	0	0	6	0	1	0	7	2	73	1	0	0	1	0
7	1	7	11	32	109	8	6	4	0	363	68	310	3	70	181	3	0	320	0
0	7	0	4	4	26	1	1	4	1	71	13	82	3	32	39	1	0	64	0
3	1	0	3	13	47	6	4	10	0	95	20	134	3	47	62	0	0	80	0
0	3	2	2	21	89	8	9	7	0	116	40	168	2	33	71	6	0	91	0
1	35	0	0	72	178	14	30	47	0	321	73	108	3	59	31	29	0	220	0
10	2	0	0	2	10	0	3	2	0	20	4	32	2	15	16	0	0	18	0
8	7	0	2	3	11	2	0	1	0	31	0	24	2	23	15	0	0	25	0
0	0	0	2	5	19	2	2	0	0	42	0	40	5	125	14	1	0	28	0
0	0	0	0	1	3	1	0	0	0	9	2	14	3	58	7	0	0	9	0
10	2	0	0	4	16	0	1	2	1	45	11	61	2	138	34	0	0	40	0

Nitelik matrisinde her bir satırda 20 adet veri bulunmakta olup, 20. alan 0 ise kaydın spam e-posta olduğunu, 1 ise normal e-posta olduğunu işaret etmektedir.

5.3. Nitelik Matrisinde YSA'nın Kullanımı

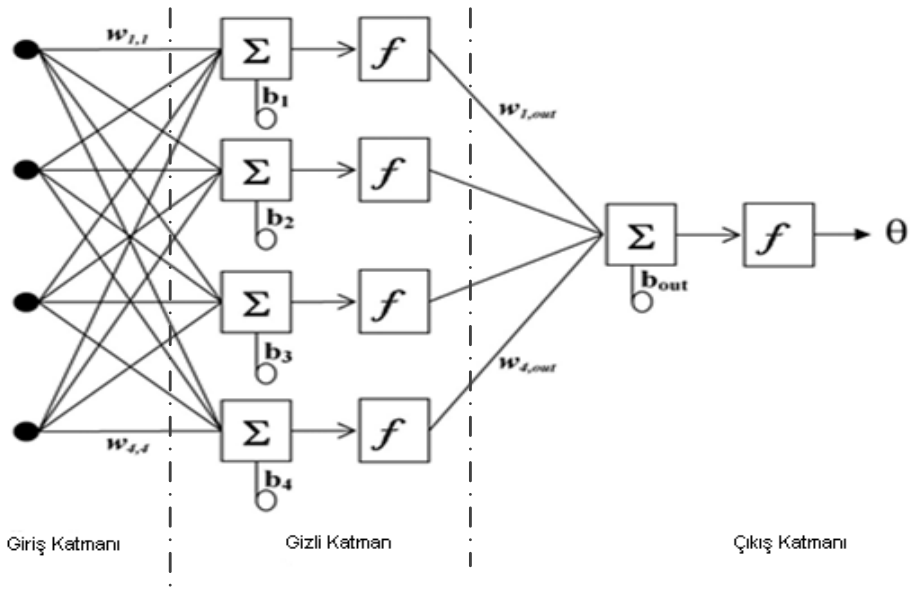
Yapay sinir ağı olarak geri yayılımlı model seçilmiştir. Aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmıştır.



Şekil 5.2. Aktivasyon fonksiyonu

$$F(x) = \frac{\mathbf{1}}{1 + e^{-x}}$$

Hata hesaplama yöntemi olarak hataların karesi toplamı (sum square error) seçilmiştir. Önerilen modelde yapay sinir ağının 19 girdisi 1 çıktısı bulunmaktadır. 3 katmanlı yapı düşünülmüş olup, α değeri 0.9, momentum katsayı 0.5 seçilmiştir. Gizli katmandaki nöron sayısı 4'dür. Kullanılan YSA yapısı şekil 5.3.'te gösterilmiştir.



Şekil 5.3. Kullanılan YSA yapısı.

.kur uzantılı parametre dosyasının içeriği aşağıdaki gibidir.

3 : YSA Seviye sayısını

0.9 : Alfa Katsayısı

0.5 : Momentum katsayısı

0.01 : Hata payı

19 : YSA Girişi Sayısı

4 : Ara katmandaki nöron sayısı

1 : Çıkış sayısını ifade eder.

Neunet.exe programı öğrenme ve sınama işlemlerinin yapıldığı programdır. Programa giriş dosyalarının bulunduğu dizin işaret edilerek çalıştırılır.

Program dizin içerisindeki. kur dosyasından katsayıları alır, sonrasında. egt uzantılı dosyayı kullanarak ağı eğitir.

Sonrasında .tst uzantili dosyadan almış olduğu veri seti için bulduğu sonuçları .son uzantili dosyaya yazar.

Programda 100-1000-10000 kayıtlı eğitim setleri ile 100 kayıtlı test seti denenmiştir. Eğitim setlerinde ve test setinde spam/spam olmayan mesaj oranı 1 dir.

Nitelik matrisinde her sütunu kendi içerisindeki maksimum ve minimum değerler için 0-1 aralığında normalize edilmiş örnek YSA kayıtları Çizelge 5.2.'de yer almaktadır.

Çizelge 5.2. YSA'da kullanılan normalize edilmiş nitelik matrisi

UnlemSayisi	UrlLinkSayisi	DolarSayisi	AdvSayisi	AdjSayisi	NounSayisi	VerbSayisi	OzelIsimSayisi	SozlukDisiSayisi	SpamPhraseSayisi	SpamWordSayisi	SpamUrlSayisi	ToplamKelimeSayisi	E-postaAdresSayisi	ExtraKarakterSayisi	GizlemeCabaSayisi	SayiSayisi	EklentiSayisi	NadirSpam	SPAM
0,33	0,03	0,14	0,09	0,08	0,1	0	0,06	0,04	0,5	0,08	0,22	0,2	0,25	0,06	0,13	0,03	0	0,08	1
0,33	0	0	0,36	0,1	0,48	0,21	1	0,51	0,5	0,54	0,62	0,86	0	0,04	0,62	0	0	0,48	1
0,25	0,03	0	0	0	0,11	0,07	0,04	0,04	0	0,06	0,14	0,14	0,75	0,09	0,08	0,1	0	0,05	1
0,17	0,03	0	0	0,03	0,05	0	0	0,02	0	0,04	0,1	0,1	1	0,04	0,08	0	0	0,04	1
0,33	0,03	0,14	0	0,14	0,25	0,07	0,11	0,11	0	0,2	0,34	0,34	0	0,08	0,28	0,17	0	0,18	1
0,42	0,03	0,29	0	0,07	0,08	0,07	0,02	0,04	1	0,07	0,11	0,16	0	0,06	0,11	0,1	0	0,07	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0,25	0	0	0	0	0	1
1	0,03	1	0,18	0,36	0,54	0,5	0,02	0,09	0	0,49	0,42	0,6	0,25	0,28	0,57	0,45	0	0,46	1
0,17	0	0	0	0	0,08	0	0,02	0	0	0,06	0,03	0,1	0,75	0,09	0,1	0,1	0	0,06	1
0,33	0	0	0	0,06	0,14	0	0,09	0,06	0	0,19	0,27	0,3	1	0,17	0,29	0,17	0	0,19	1
0,83	0	0,71	0,09	0,22	0,38	0,07	0,15	0,11	0,5	0,3	0,37	0,47	0,5	0,22	0,36	0,31	0	0,25	1
0,83	0,6	0	0	0	0	0	0	0,13	0	0	0	0,02	0	0,53	0,01	0	0	0	0
0,58	0,03	1	1	0,44	0,61	0,57	0,13	0,09	0	1	0,93	1	0,25	0,51	1	0,1	0	1	0
0	0,2	0	0,36	0,06	0,15	0,07	0,02	0,09	0,5	0,2	0,18	0,26	0,25	0,23	0,22	0,03	0	0,2	0
0,25	0,03	0	0,27	0,18	0,26	0,43	0,09	0,21	0	0,26	0,27	0,43	0,25	0,34	0,34	0	0	0,25	0
0	0,09	0,29	0,18	0,29	0,5	0,57	0,19	0,15	0	0,32	0,55	0,54	0	0,24	0,39	0,21	0	0,28	0
0,08	1	0	0	1	1	1	0,64	1	0	0,88	1	0,35	0,25	0,43	0,17	1	0	0,69	0
0,83	0,06	0	0	0,03	0,06	0	0,06	0,04	0	0,06	0,05	0,1	0	0,11	0,09	0	0	0,06	0
0,67	0,2	0	0,18	0,04	0,06	0,14	0	0,02	0	0,09	0	0,08	0	0,17	0,08	0	0	0,08	0
0	0	0	0,18	0,07	0,11	0,14	0,04	0	0	0,12	0	0,13	0,75	0,91	0,08	0,03	0	0,09	0

YSA'da kullanılacak olan nitelik matrisi Çizelge 5.2'de görüldüğü üzere normalize edilmiş değerlerden oluşmaktadır. Her bir satırda 20 adet veri bulunmakta olup, 20.

alan 0 ise kaydın spam e-posta olduğunu, 1 ise normal e-posta olduğunu işaret etmektedir.

Programın durma şartı uygulama hata payının % 0.01 den küçük eşit olması ya da iterasyon sonucunun 3 000 000'a ulaşmasıdır.

5.4. Nitelik Matrisinde Kümeleme Yöntemlerinin Kullanımı

Tez çalışması kapsamında nitelik matrisi üzerinde k-means kümeleme algoritması Weka veri madenciliği yazılımında kullanılmıştır. Program, nitelik matrisini okur, tasarlama aşamasından geçilir ve sonrasında k-means ile küme merkezlerini hesaplar. Daha sonra bu merkezler kullanılarak test vektörlerinin başarı oranları çıkarır.

5.5. Deneysel Bulgular

YSA programı ile yapılan testlerde aşağıdaki sonuçlar ile karşılaşılmıştır. YSA sonuçlarına Çizelge 5.3.'te yer verilmiştir.

- Test kümesinin sayısını öğrenme kümesinin sayısına oran ile çok az sayıda olduğu durumda öğrenme yapamadığı görülmüştür. Söz konusu durumda 100 adet spam maile ait nitelik değerleri ve 100 adet te spam olmayan maile ait kayıdın bulunduğu 200 adet verinin bulunduğu test seti ile 5000 adet spam mail ve 5000 adet spam olmayan mailden oluşan 10000 adet verinin bulunduğu öğrenme setinin olduğu durumdur.
- Test seti ve öğrenme setinin karışık olmaması yani önce spam mail sonrasında spam olmayan maillerin gelmesi de öğrenmeyi olumsuz etkilemektedir.
- Alfa katsayısının artırılması öğrenmeyi olumlu yönde etkilemektedir.

Çizelge 5.3. YSA sonuçları

alfa=0.99, katman sayısı=3					
Eğitim seti/test seti	TP	TN	FP	FN	Öğrenme Oranı %
100/100	50	50	0	0	100
1000/100	50	50	0	0	100
10000/100	50	50	0	0	100

Weka veri madenciliği yazılım aracında nitelik matrisin K-means yöntemi ile elde edilen sonuçları Çizelge 5.4.'te verilmiştir.

Çizelge 5.4. Spam filtreleme için Weka'da K-means sonucu

Nitelikler	Cluster 0	Cluster1	Full Data
UnlemSayisi	2,913	0,783	1,857
UrlLinkSayisi	3,276	0,338	1,807
DolarSayisi	2,250	0,544	1,399
AdvSayisi	3,660	2,729	3,195
AdjSayisi	14,864	11,086	12,975
NounSayisi	51,778	48,300	50,039
VerbSayisi	3,020	2,400	2,712
OzelIsimSayisi	4,743	6,110	5,430
SozlukDisiSayisi	7,324	4,798	6,061
SpamPhraseSayisi	0,760	0,455	0,608
SpamWordSayisi	135,701	99,439	117,570
SpamUrlSayisi	25,596	20,974	23,285
ToplamKelimeSayisi	124,764	104,644	114,704
E-postaAdresSayisi	2,000	3,200	2,600
ExtraKarakterSayisi	81,030	33,774	57,402
GizlemeCabaSayisi	68,247	52,159	60,203
SayiSayisi	4,780	4,613	4,696
EklentiSayisi	0,042	0,000	0,021
NadirSpam	114,537	82,245	98,391
SPAM	0,000	1,000	0,500
Küme Dağılım Oranları	5000 (%50)	5000 (%50)	

Weka veri madenciliği yazılım aracında nitelik matrisin x-means yöntemi ile elde edilen sonuçları Çizelge 5.5'te yer almaktadır.

Çizelge 5.5. Spam filtreleme için Weka'da x-means sonucu

Nitelikler	Cluster 0	Cluster1	Cluster2	Cluster3
UnlemSayisi	3,683	0,594	3,941	2,712
UrlLinkSayisi	19,705	0,232	3,597	3,206
DolarSayisi	4,604	0.279	4,340	1,803
AdvSayisi	17,888	1,740	12,434	1,758
AdjSayisi	68,330	7,354	52,039	6,803
NounSayisi	300,254	31,875	180,434	23,880
VerbSayisi	15,826	1,525	9,672	1,581
OzelIsimSayisi	30,856	4,504	12,932	2,967
SozlukDisiSayisi	19,316	3,851	18,835	4,828
SpamPhraseSayisi	3,840	0,234	1,934	0,506
SpamWordSayisi	590,722	67,413	471,507	62,884
SpamUrlSayisi	110,339	15,148	82,052	13,354
ToplamKelimeSayisi	478,401	80,279	374,459	70,620
E-postaAdresSayisi	9,400	2,800	2,000	2,000
ExtraKarakterSayisi	215,238	21,945	148,618	66,374
GizlemeCabaSayisi	234.316,000	40,284	221,427	35,031
SayiSayisi	21,480	3,513	9,314	3,797
EklentiSayisi	0,000	0,000	0,013	0,048
NadirSpam	485,029	55,980	397,772	53,120
SPAM	1,000	1,000	0,000	0,000
Küme Dağılım Oranları	306 (%3)	4694 (%47)	891(%9)	4109(%41)

6. SONUÇ VE ÖNERİLER

E-posta gerek hızı ve büyük miktarlarda veri gönderebilmesi gerekse haberleşme maliyetinin çok düşük olması nedeni ile günümüzün vazgeçilmez haberleşme araçlarından birisi olduğu kesindir. Bununla birlikte bu avantajları bunu ticari pazarlama, reklam, haber ve bilgi çalma amaçlı kullanmak isteyenler için de vazgeçilmez bir araç olduğu kesindir.

Spam e-postanın önüne geçilmesinde kullanıcılara ve ISP'lere ve e-posta servis sağlayıcılarına büyük görevler düşmektedir. Kullanıcılara düşen en temel görev; kaynağı bilinmeyen e-posta adreslerinden gelen e-postaları dikkate almamaları, zincir e-posta iletimine katkıda bulunmamaları şeklinde özetlenebilir. ISP'ler ve e-posta servis sağlayıcıları ise gerekli anti-spam yazılımlarını kullanarak spam e-posta trafiğini büyük ölçüde azaltmış olacaklardır.

Gerek spam e-postalarda kullanılan dil olması, gerekse de literatürde yapılan çalışmaların tamamına yakınında yer alması nedeni ile, tez kapsamında İngilizce dili üzerinde uygulamalar yapılmıştır.

Bu çalışma kapsamında veri madenciliği ve spam filtreleme üzerine araştırmalar yapılmış, kullanılan yöntemler ve sonuçları incelenmiş, araştırmalarda yer alan çalışmalarda göz önünde bulundurarak farklı bir yöntem denenmiştir.

E-posta verilerin değerlendirilebilmesi amacı ile üzerinde danışmanlı ve danışmansız öğrenim metotlarının uygulanabileceği bir sistem tasarlanmıştır.

Söz konusu sistem esnek bir sistem olup, üzerinde ilave değişiklikler yapılmasına imkân vermektedir.

Çalışmada; e-posta dosyaları içerisinde spam unsuru olduğu düşünülen kelime türlerine göre analiz edilmiş ve analiz sonucuna göre nitelik matrisi oluşturulmuştur. Bu nitelik matrisi üzerinde YSA ve K-means kümeleme yöntemi denenmiştir. Yine nitelik matrisi üzerinde Weka veri madenciliği yazılımı ile K-means, X-means, EM ve Optics algoritmaları denenmiştir.

Yapay sinir ağları ve kümeleme metotları ile veri setleri üzerinde yapılan testlerde spam ve spam olmayan e-postaların tespiti yapılmıştır.

Bu çalışmanın diğer mevcut çalışmalardan farkı; kelime türlerine, özel spam kelimeleri ve işaretlerine ya da url'lerine, spam ve normal kelimelerin birlikte yer aldığı bir sözlük ile karşılaştırılarak önceden belirlenmiş alanlara göre nitelik uzayının oluşturulması ve problemin çözümünde başarı elde edilmesidir. Elde edilen nitelik uzayı YSA, kümeleme analizi, Naive Bayes vb. yöntemler için uygulanabilir yapıda olması da çalışmanın diğer önemli bir özelliğidir.

Yapılan çalışmalar sonrasında herhangi bir problemin çözümünde oluşturulabilecek uygun bir nitelik bir matrisinin oluşturulması ile tüm metotların uygulanabileceği görülmüştür.

Sonraki çalışmaların farklı niteliklerin seçimi, örüntü tanıma, yöntemlerin bileşkesi üzerine yapılması uygun olacaktır.

KAYNAKLAR

1. Internet : Boğaziçi Üniversitesi "Zeki Veri Madenciliği: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri" www.cmpe.boun.edu.tr/~ethem/files/papers/veri-maden_2k-notlar.doc (2010).

2. Frawley, W. J., Piatetsky-Shapiro, G., Matheus, C. J., "Knowledge discovery in databases: An overview" , ***American Association for Artificial Intelligence***,13 (3): 1-27 (1991).

3. Fayyad, U. M., Weir, N., Djorgovski, S., "Automated analysis of a large-scale sky survey: The SKI CAT System" , ***American Association for Artificial Intelligence***, 2: 1-13 (1993).

4. Internet : Tech Essence "Data Mining For It Professionals" http://techessence.info/files/Ayre_DataMiningForInformationProfessionals_June2006.pdf (2006).

5. Uthurusamy, R., "From data mining to knowledge discovery: current challenges and future directions", Advances in Knowledge Discovery and Data Mining, Fayyad, U. M., Piatetsky-Shapiro, G.,Uthurusamy, R., ***AAAI Press/MIT Press***, California, 561-572 (1996).

6. Fayyad, U. M., Piatetsky-Shapiro, G., "The KDD process for extracting useful knowledge from volumes of data", ***Communications of the ACM***, 39 (11): 27-34 (1996).

7. Corinna, C., Drucker, H., Hoover, D., Vapnik, V., "Capacity and complexity control in predicting the spread between harrowing and lending interest rates", The First International Conference on Knowledge Discovery and Data Mining, U. Fayyad, R. Uthurusamy, ***AAAI Press***, Montreal, 51-76 (1995).

8. Zhong, N., Ohsuga, S., "Discovering concept clusters by decomposing databases", ***Data & Knowledge Engineering***, 12: 223-244 (1994).

9. Chan, K. C. C., Wong, A. K. C., "A statistical technique for extracting classificatory knowledge from databases", In Knowledge Discovery In Databases, ***AAAI Press/MIT Press***, California, 107-123 (1991).

10. Shapiro, G. P., Matheus, C. J., "Knowledge discovery workbench for exploring business databases", ***International Journal of Inteldigent Systems***, 7: 675-686 (1992).

11. Elder, J. F., Pregibon, D., "A statistical perspective on KDD", The First International Conference on Knowledge Discovery and Data Mining, U. Fayyad, R. Uthurusamy, ***AAAI Press***, Montreal, 87-93 (1995).

12. Simoudis, E., "Reality check for data mining", ***IEEE Expert: Intelligent Systems and Their Applications***, 2(5): 26-33 (1996).

13. Weiss, S. M., Kulikowski, C. A., "Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems" , ***Morgan Kaufman Publishers***, San Francisco, 218-223 (1990).

14. Sever, H., Oğuz., B., "Veri tabanlarında bilgi keşfine formel bir yaklaşım", ***Bilgi Dünyası***, 2: 173-204 (2002).

15. Aydogan, L., "E-Ticarette veri madenciliği yaklaşımlarıyla müşteriye hizmet sunan akıllı modüllerin tasarımı ve gerçekleştirimi", Yüksek Lisans Tezi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 10-20 (2003).

16. Hsu, J., "Data mining trends and developments: the key data mining technologies and applications for the 21st century", ***ISECON***, San Antonio, 1-7 (2002).

17. Quinlan, J. R., "Induction of decision trees", ***Machine Learning***, 1: 81- 106 (1986).

18. Luba, T., Lasocki, R., "On unknown attribute values in functional dependencies", ***Proceedings Of The International Workshop On Rough Sets And Soft Computing***, San Jose, 490-497 (1994).

19. Deogun, J. S., Raghavan, V. V., Sarkar A., Sever, H., "Data mining: trends in research and development", Rough Sets and Data Mining: Analysis for imprecise Data, Lin, T. Y., Cercone, N., ***Kluwer Academic Publishers***, California, 9-45 (1997).

20. Internet : North Carolina University "Data Mining Tool Comparission" http://coitweb.uncc.edu/~mirsad/ (2010).

21. Internet : Center for Information Technology and Privacy Law "Spam and etc.." www.spamlaws.com (2010).

22. Internet: SafeNet "Anti Spam White Paper" www.aladdin.com (2010).

23. Peterson, C., Anderson, J. R., "A mean field theory learning algorithm for neural networks", Artificial Neural Networks Concepts and Theory,Pankaj, M., Wah, W. B., ***IEEE Computer Society Press***, California, 430-455 (1992).

24. Caudill, M., "Neural networks primer part I", ***AI Expert***, 2(12): 46-52 (1987).

25. Öztemel, E., " Yapay Sinir Ağları", ***Papatya Yayıncılık***, İstanbul, 55 (2003).

26. Stern, H., "Neural networks in applied statistics", ***Technometrics***, 38 (3): 205-211 (1996).

27. Han, J., Kamber, M., "Data Mining Concepts and Techniques 2nd ed.", ***Morgan Kauffmann Publishers Inc.***, San Francisco, 408-418 (2006).

28. Karypis, G., Han, E. H., Kumar, V., "CHAMELEON: A hierarchical clustering algorithm using dynamic modeling", ***IEEE Computer***, 32(8): 68-75 (1999).

29. Boutsinas, B., Gnardellis, T., "On distributing the clustering process", ***Pattern Recognition Letters***, 23: 999-1008 (2002).

30. Demiralay, M., Çamurcu, A. Y., "Cure, agnes ve k-means algoritmalarındaki kümeleme yeteneklerinin karşılaştırılması", ***İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi***, 8: 1-18 (2005).

31. Jain, A. K., Murty, M. N., Flynn, P. J., "Data clustering: A review", ***ACM Computing Surveys***, 31(3): 265-323 (1999).

32. Jain, A. K., Dubes, R. C., "Algorithms For Clustering Data", ***Prentice Hall***, New Jersey, 288-301 (1988).

33. Witten, I. H., Frank, E., "Data Mining: Practical Machine Learning Tools with Java Implementations", ***Morgan Kaufmann***, San Francisco, 210-226 (1999).

34. Mercer, D. P., "Clustering Large Datasets", ***Linacre College,*** Oxford, 1-48 (2003).

35. Han, J., Kamber, M., Tung, A. K. H., "Spatial clustering methods in data mining: A survey", Geographic Data Mining and Knowledge Discovery, ***Taylor and Francis***, London, 188-217 (2001).

36. Kohavi, R., Provost, F., "Applications of data mining to electronic commerce", ***Data Mining and Knowledge Discovery***, 5 (1-2): 1-7 (2001).

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

38. Valgeirsson, A. G., Erlingsson, B., Einarson, İ. S., "Using clustering to index image descriptors: A performance evaluation ", ***Reykjavik University,*** Iceland, 1-42 (2003).

39. Zaiane, O. R., Foss, A., Lee, C. H., Wang, W., "On data clustering analysis: Scalability, constraints and validation", Lecture Notes in Computer Science, ***Springer Science***, Berlin, 28-39 (2002).

40. Kolatch, E., "Clustering algorithms for spatial databases: A survey", ***University of Maryland Department of Computer Science,*** Maryland, 1-22 (2001).

41. Syed, A. A., ”Performance analysis of k-means algorithm and kohonen networks”, Yüksek Lisans Tezi, ***Florida Atlantic University Department of Computer Science***, Florida, 15-32 (2004).

42. Tarsitano, A., ”A computational studyof several relocation methods for k-means algorithms”, ***Pattern Recognition***, 36: 2955 – 2966 (2003).

43. Zaiane, O., “Data clustering”, Principles of Knowledge Discovery in Databases, ***University of Alberta***, Edmonton, 1-45 (1999).

44. Pena, J. M., Lozano, J. A., Larranaga, P., “An empirical comparison of four initialization methods for the k-means algorithm”, ***Pattern Recognition Letters***, 20(10): 1027-1040 (1999).

45. Kaufman, L., Rousseeuw, P. J., “Finding Groups in Data: An Introduction to Cluster Analysis”., ***Wiley-Interscience***, New York, 340-350 (2005).

46. Ester, M., Kriegel, H. P., Sander, J., Xu, X., ”A density based algorithm for discovering clusters in large spatial databases with noise”,***Proceedings of 2nd International Conference onKnowledge Discovery and Data Mining***, Portland, 226-331 (1996).

47. Xu, X., Ester, M., Kriegel, H. P., Sander, J., “Clustering and knowledge discovery in spatial databases”, ***Vistas in Astronomy***, 41(3): 397-403 (1997).

48. Ankerst, M., Breuning, M., Kriegel, H. P., Sander, J., “OPTICS: Ordering Points To Identify The Clustering Structure”, ***Proc. ACM SIGMOD Int. Conf. Management of Data***, Philadelphia,49-60 (1999).

49. Hinneburg, A., Keim, D. A., ”An efficient clustering approach to clustering in large multimedia databases with noise”, Knowledge Discovery and Data Mining,***The Institution of Electrical Engineers***, London, 58-65 (1998).

50. Foss, A., Zaiane, O. R., “TURN* unsupervised clustering of spatial data”, ***ACM-SIKDD Intl. Conf. On Knowledge Discovery and Data Mining***, Edmonton, 55-72 (2002).

51. Wang, W., Yang, J., Muntz, R., ”STING: A statistical information grid approach to spatial data mining”, ***Int. Conference On Very Large Databases***, Atina, 1-18 (1997).

52. Sheikholeslami, G., Chatterjee, S., Zhang, A., ”WaveCluster: A multi-resolution clustering approach for very large spatial databases”, ***Int. Conf. On Very Large Databases***, New York, 428-439 (1998).

53. Mastrogiannis, N., Boutsinas, B., Giannikos, I., “A method for improving the accuracy of data mining classification algorithms”, ***Computer & Operation Research***, 36: 2829-2839 (2009).

54. Guzella, T. S., Caminhas, W. M., "A review of machine learning approaches to spam filtering", ***Expert Systems with Applications***, 36: 10206-10222 (2009).

55. Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., Stamatapoulos, P., "A memory-based approach to anti-spam filtering for mailing lists", ***Information Retrieval***, 6: 49-73 (2003).

56. Gordillo, J., Conde, E., "An HMM for detecting spam mail", ***Expert Systems with Applications***, 33: 667-682 (2007).

57. Thorsten, T., "Application of machine learning techniques to spam filtering", Yüksek Lisans Tezi, ***Universitat Paderborn Fakultat fur Elektrotechnik, Informatik and Mathematik***, Gutersloh, 77-78 (2004).

58. Balamurugan, S. A., Rajaram, R., Athiappan, G., Muthupandian, M., "Data mining techniques for suspicious e-mail detection: a comparative study", ***IADIS European Conference Data Mining***, Lisbon, 213-217 (2007).

59. Zorkadis, V., Karras, D. A., Panayotou, M., "Efficient information theoretic strategies for classifier combination, feature extraction and performance evaluation in improving false positives and false negatives for spam e-mail filtering", ***Neural Networks***, 18: 799-807 (2005).

60. Chen, Y. L., Cheng, L. C., "Mining maximum consensus sequences from group ranking data", ***European Journal of Operational Research***, 198: 241-251 (2009).

61. Tretyakov, K., "Machine learning tecniques in spam filtering", ***Data Mining Problem-Oriented Seminar***, Tartu, 60-79 (2004).

62. Fawcett, T., "In vivo spam filtering: A challenge problem for data mining", ***KDD Explorations***, 5(2): 140-148 (2003).

63. Shetty, J., Adibi, J., "The enron email dataset databaset database schema and brief statistical report", ***Computational & Mathmatical Organization Theory***, 11(3): 1-7 (2005).

64. Internet: Carnigee Mellon University "Enron Email Dataset" http://www-2.cs.cmu.edu/~enron/ (2010).

65. Internet: CSMining Group "Enron Email Dataset" http://csmining.org/index.php/enron-spam-datasets.html (2010).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : SARIKOZ, Serdar Kürşat

Uyruğu : T.C.

Doğum tarihi ve yeri : 31.10.1978 Erzincan

Medeni hali : Evli

Telefon : 0 (312) 586 55 00

Faks : 0 (312) 586 36 36

e-mail : sksarikoz@hotmail.com

Eğitim

Derece	Eğitim Birimi	MezuniyetTarihi
Lisans	Sakarya Üniversitesi/ Bilg. Müh. Bölümü	2002
Lise	Yenimahalle Teknik Lisesi	1996

İş Deneyimi

Yıl	Yer	Görev
2003-2003	T.C. Adalet Bakanlığı B.İ.D.B	Çözümleyici
2003-2006	Türk Telekom A.Ş.	Uzman Yrd.
2006	Telekomünikasyon İletişim Bşk.	İletişim Uzmanı

Yabancı Dil

İngilizce