T.C
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MADENCİLİĞİNDE BULANIK MANTIK UYGULAMASI

Eyüp SIRAMKAYA

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Konya, 2005

Bu tez 24/01/2006 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

Prof.Dr.Ahmet ARSLAN	Prof.Dr. Ferruh YILDIZ	Doç.Dr.Şirzat KAHRAMANLI
(A.B.D Bşk.- Danışman)	(Üye)	(Üye)

ÖZET

Bu tezde öncelikle bir veri madenciliği uygulaması yapılmıştır. Bu uygulamada, internet üzerinden ulaşılabilen basın-yayın kaynaklarında yer alan görsel ve metinsel verilerin hızlı ve etkin bir şekilde erişimi ve bu kaynaklardan anlamlı ve önemli bilgilerin çıkarılması hedeflenmektedir. Çalışmalar öncelikli olarak istihbarat açısından önem taşıyan kişi ve örgütlerle ilgili haberler üzerinde yoğunlaşmıştır. Sunucu bilgisayarda İnternet üzerinde yer alan haber kaynaklarından toplanmış ve işlenmiş metinsel belgelerden oluşan veri-tabanı ile bu bilgileri işleyen uygulama yazılımları bulunmaktadır. Bir arayüz ile kullanıcının bu bilgileri sorgulaması sağlanmıştır. Çalışma, Birliktelik Kural Madenciliği tekniği ile uygulanmıştır. Bu teknik uygulanırken Apriori Algoritması kullanılmıştır.

Yapılan veri madenciliği çalışmasında Bulanık Mantık çalışması, kişi-kişi ilişkilerini bulmakta uygulanmıştır. Bu uygulamadaki amaç kullanıcıların arama yapmak istedikleri kişilerin isimlerini yazarken yapabilecekleri yazım hatalarını elemektir. İsimlerdeki harflerin konumlarının birbirlerine göre uzaklıklarını temel alarak bulanım mantık kurallarının uygulandığı bir algoritma kullanılmıştır.

Anahtar Kelimeler: Bulanık Mantık, Veri Madenciliği, Bulanık Arama, Bulanık İsim Arama, Birliktelik Kural Madenciliği, Apriori Algoritması.

ABSTRACT

In this thesis firstly a data mining application is done. In this application, it is aimed to reach the visual and text data placed in the pressed-publish sources on internet in a speed and effective way and expose meaningful and important information from these sources. The studies are focused especially on the news about the people and organizations which are important from the intelligence point of view. In the server computer, there are a data base formed by the processed text documents, which are collected from the news sources on internet, and the application software processing this information. It is provided for the user to interrogate the information by the means of an interface. The study is applied with the association rule mining technique. During this technique applied apriori algorithm is used.

In data mining study, the fuzzy logic study is applied to find person-person relationship. The aim in this application is to eliminate the writing errors the users may do while they write the names of people they want to search. An algorithm applying the fuzzy logic rules which takes the distances of the letters in the names according to each other as basis is used.

Key Words: Fuzzy Search, Fuzzy Name Search, Apriori Algorithm, Data Mining, Association Rule Mining.

.

TEŞEKKÜR

Yüksek lisansım boyunca katkılarını, yönlendirici desteğini ve anlayışını hiçbir zaman esirgemeyen danışmanım Sayın Prof. Dr. Ahmet ARSLAN'a; bu süreç boyunca göstermiş oldukları destek, yardım ve gayretlerinden dolayı sevgili meslektaşlarım Sayın Ömer Kaan BAYKAN ve Sayın Erkan ÜLKER'e, çalışmalarım boyunca manevi ve maddi destekleri ile her zaman yanımda olan aileme, sevgili eşim Süheyla BÜYÜKŞAHİN SIRAMKAYA'ya,
Teşekkürler…

İÇİNDEKİLER

ÖZET ……...……i
ABSTRACT ...……ii
TEŞEKKÜR ……..iii
İÇİNDEKİLER ……..iv
ŞEKİLLER, RESİMLER VE TABLOLAR..…….vii
1. GİRİŞ ……..……..1
1.1. Tezin Konusu, Amacı ve Önemi ..…….……..2
2. VERİ MADENCİLİĞİ …….…….3
2.1. Veri Madenciliği Kavramının Tanımı, Gelişimi, Özellikleri ……3
2.1.1. Veri Madenciliği Nedir? …….…….…….3
2.1.2. Veri madenciliği ne değildir? …….3
2.1.3. Veritabanı Teknolojilerinin Gelişimi …….3
2.1.4. Neden veri madenciliği? ……4
2.1.5. Neden Veri Madenciliğine Gereksinim Duyuldu? ……4
2.2. Veri madenciliği ne tür veriler üzerinde yapılır? …….5
2.3. Veri Madenciliğinde Çözümleme Yaklaşımları …….……5
2.3.1.Aşağıdan yukarıya …...……5
2.3.2.Yukarıdan aşağıya …..…….……5
2.3.3.Yöntemler …….....……6
2.3.4.Özellikler …….……...……..6
2.3.5.Çeşitli Tanımlamalar ……...……7
2.4. Veri Madenciliğinin Diğer Disiplinlerle İlişkisi ……...8
2.4.1. İstatistikle karşılaştırılması ……..8
2.4.2. Veri madenciliği yöntemleri ikiye ayrılabilir ……..…9
2.5. VM Türleri ……..……..9
2.5.1. Doğrudan veri madenciliği ……..……9
2.5.2. Varsayım deneme ve varsayımı daha iyi hale getirme …..……10
2.5.3. Dolaylı veri madenciliği ……10
2.6. Bilgi Bulma Süreci …….…10
2.6.1. Bilgi bulma sürecinin önemli özellikleri ……..….10
2.6.2. Bilgi bulma sürecinin önemli Sorunlar ……..…11
2.6.3. Bilgi Bulma Sürecinde Kullanılan Yöntemler ……12

2.6.3.1. Seçim ……12
2.6.3.2. İşlem Öncesi ……13
2.6.3.2.1. İşlem Öncesi Süreçler (Ön Hazırlık) ……13
2.6.3.3. Veri Madenciliği Aşaması ……14
2.6.3.4. Yorum ……14
2.7. Veri Madenciliği Teknikleri ……14
2.7.1. Kümeleme ya da Parçalama……14
2.7.2. Sınıflandırma ……15
2.7.3. Kavram Tanımlaması ……15
2.7.4. Bağımlılık Çözümlemesi ……15
2.7.5. Bağlantı Çözümlemesi ya da İlişkiler ……15
2.7.6. Dizi Çözümlemesi ……15
2.7.7. Tahmin ……16
2.7.8. Keşfedici Veri Çözümlemesi ……16
2.8. Veri Madenciliği Tekniklerinin Modellenmesi ……16
2.8.1. Parametrik Modeller ……16
2.8.2. Parametrik Olmayan Modeller ……16
2.8.2.1. Parametrik ve Parametrik Olmayan Modellerin Karşılaştırılması ……17
2.8.2.2. İstatistiksel Yaklaşımlar ……17
2.8.2.2.1. Sayı Tahmini (Point Estimation) ……18
2.8.2.2.2. Özetlemeye Dayalı Modeller ……18
2.8.2.2.3. Bayes Teoremi ……18
2.8.2.2.4. Varsayım Sınama (Hypothesis Testing) …19
2.8.2.2.5. Gerileme ve Korelasyon ……19
2.8.2.2.6. Benzerlik Ölçümü ……20
2.8.2.3. Karar Ağaçları ……20
2.8.2.4. Yapay Sinir Ağları ……21
2.8.2.5. Genetik Algoritmalar ……22
3 BULANIK MANTIK ……25
3.1. Bulanık Mantık (Fuzzy Logic) Nedir? ……25
3.2. Bulanık Sistem Ne Demektir ……28

3.3. Üyelik Fonksiyonları……33
3.3.1. Üyelik Fonksiyonunun Kısımları……37
3.3.2. Bulanıklaştırma……41
3.3.3. Durulaştırma……42
3.3.3.1. Durulaştırma İşlemleri……43
4. BİRLİKTELİK KURAL MEDENCİLİĞİ ……51
4.1. Geniş Veritabanlarından Birliktelik Kuralları Çıkarılması İşlemi…51
4.2. Birliktelik Kural Madenciliği Nedir? ……51
4.3 Birliktelik Kuralı……52
4.4. Apriori Algoritması……..53
5. BULANIK VERİ MADENCİLİĞİ UYGULAMASI……55
5.1. Basın-Yayın Alanında Birliktelik Kural Madenciliği Sistemi……55
5.2. Basın-Yayın Alanında Birliktelik Kural Madenciliği Sistemine Bulanık Mantık Uygulaması……..……56
5.2.1. Çalışma Prensibi……57
5.2.2. Algoritmanın Açıklaması……...57
5.2.2.1. Bulanıklaştırma Fonksiyonu……59
5.2.2.1.1. Üyelik Fonksiyonları……59
5.2.2.2. Bulanık Çıkarım Motoru……60
5.2.2.2.1.Bulanık Çıkarım Motoru Fonksiyonları……61
5.2.2.3. Bulanık Kural Tablosu……61
5.2.2.4. Durulaştırıcı……62
5.2.3. Algoritmanın Bir Örnekle Açıklaması……...……62
5.2.4. Algoritmanın Özel Durumları……...66
5.2.4.1. Kullanıcı tarafından girilen kelimenin veri tabanından karşılaştırılan kelimeden uzun olması durumu…...…66
5.2.4.2. Kullanıcı tarafından girilen kelimenin veri tabanından karşılaştırılan kelimeden kısa olması durumu…...…67
5.2.5. Algoritmanın Performans Analizi ……69
5.2.6. Algoritmanın Lineer Algoritma ile Karşılaştırılması ……71
5.2.6.1. Lineer Algoritmanın Açıklaması ……71

5.2.6.2. Bulanık Sistemin Lineer Sisteme Göre Avantajları ...72

6. SONUÇLAR ……74
6.1 Çalışmanın Bulanık Mantık Kısmının Sonuçları……74
6.2. Çalışmanın Veri Madenciliği Kısmının Sonuçları……74
7. KAYNAKLAR……....78

ŞEKİLLER, RESİMLER VE TABLOLAR

Resim 2.1. Veri Madenciliği Süreci……..……...……..5
Resim 2.2. Veri Madenciliğinin ilişkili olduğu diğer disiplinler…....……..…7
Şekil 3.1. Bulanık Mantık ile Klasik Mantık Modelleri….……...…….26
Şekil 3.2. Klasik Sistem……...……...27
Şekil 3.3. Genel Bulanık Sistem……...……..28
Şekil 3.4. TSK Bulanık Sistem…...……30
Şekil 3.5. Bulanıklaştırma-Durulaştırma birimli bulanık sistem..……..31
Şekil 3.6. Bitişik Dikdörtgen Gösterimi…….32
Şekil 3.7. Bitişik Üçgen Gösterimi…….33
Şekil 3.8. Örtüşmeli Üçgen Gösterimi……34
Şekil 3.9. Bulanık Küme…….35
Şekil 3.10. Yamuk ve Çan Eğrisi üyelik fonksiyonları……..36
Şekil 3.11. Bulanık kümeler, (a) normal, (b) normal olmayan……...37
Şekil 3.12. Bulanık kümeler, (a) dış bükey, (b) dış bükey olmayan……...38
Şekil 3.13. Dış bükey bulanık kümelerin kesişimi…….39
Şekil 3.14. Gauss (a) bulanık kümesi (b) dağılım fonksiyonu……40
Şekil 3.15 Hassaslık (a) bulanık (b) klasik…….41
Şekil 3.16. İki bulanık kümenin (a) birleşimi, (b) kesişimi……42
Şekil 3.17. Tipik bulanık küme çıktısı, (a) bulanık girdi ilk kısım, (b) bulanık girdi ikinci kısım, (c) ikisinin birleşimi……...43
Şekil 3.18. En büyük üyelik derecesi durulaştırması……..44
Şekil 3.19. Sentroid yöntemi ile durulaştırma……45
Şekil 3.20. Ağırlıklı ortalama yöntemi durulaştırması……46
Şekil 3.21. Ortalama en büyük üyelik durulaştırılması……...47
Şekil 3.22. Toplamların merkezi durulaştırması…….48
Şekil 3.23. En büyük alan merkezi ile durulaştırma……...49
Şekil 3.24. İlk ve son en büyük üyelik dereceleri ile durulaştırma…….50
Şekil 4.1. The pseudo code of Apriori algorithm……54
Şekil 5.1. The System of the Mining Association Rules……56
Şekil 5.2. Bulanık Mantık Uygulamasının Şematik Gösterimi……...57

Şekil 5.3. Üyelik Fonksiyonu……..…59
Şekil 5.4. Durulaştırma Fonksiyonu ……..60
Şekil 5.6. n=8 için oluşan üyelik fonksiyonu……..63
Şekil 5.7. Aynı uzunlukta olmayan kelimeler için daraltma işlemi ……...67
Şekil 5.8. Aynı uzunlukta olmayan kelimeler için sündürme işlemi …….....68
Tablo 5.1. Performans analiz tablosu …….70
Grafik 5.1. Performans Analiz Grafiği ……..71
Tablo 6.1. Bulanık mantık ile Lineer algoritmanın Karşılaştırma Sonuçları……..74
Tablo 6.2. Destek ve Güven Değeri İle kişi-yer, kişi-kişi, kişi-olay Birliktelik Kurallarından Bazıları…….76
Şekil 6.1. The PHP interface of the application……..77
Şekil 6.2. Reached the news…….....77

1. GİRİŞ

İnternet kullanımının hızla artması ve son kullanıcının da internet ortamına ulaşımının kolaylaşması ile çoğu sektörde olduğu gibi basın alanında da kuruluşlar ürünlerini internet ortamına taşımışlardır. Basın verilerinde kişilerin kişisel ilgilerine yönelik güncel ve geçmişteki haberlere yönelik sorgular yapması da oldukça kolaylaşmıştır. Ama sorgu sonuçlarının çok fazla yanıt döndürmesi kişilerin bazen istemedikleri ve alakasız sorgu sonuçları ile yüzyüze gelmelerine de neden olmaktadır. Bu nedenle, İnternet üzerinden ulaşılabilen basın-yayın kaynaklarında yer alan metinsel verilerin hızlı ve etkin bir şekilde erişimi ve bu kaynaklardan anlamlı ve önemli bilgilerin çıkarılması yönünde çalışmaların ortaya konulması gerekmektedir.

Veri madenciliği veri tabanında bulunan gizli bilgilerin keşfidir ve bilgi keşfi işleminin bir adımı olarak görülebilir. Veri madenciliği fonksiyonları; kümelemeyi, sınıflandırmayı, tahmin etmeyi ve birliktelikleri içerir. En önemli veri madenciliği uygulamalarından birisi birliktelik kurallarının madenciliğidir. Birliktelik kuralları veri madenciliğinin en çok araştırma yapılan alanlarından birisidir ve son zamanlarda veri tabanı iletişiminden daha fazla ilgi çekmiştir. İşlem iki ayrı aşamaya sahiptir; birincisi sıklık öğe kümelerini bulmaktır ve ikincisi bu öğe kümelerden kurallar tanımlamaktır. Bu çalışmada, İnternet üzerinden ulaşılabilen gazetelerde yer alan metinsel verilerden anlamlı ve önemli bilgilerin çıkarılması amaçlanmaktadır. Kişi-kişi, kişi-yer, olay-yer, olay-tarih ilişkileri veri madenciliği teknikleri kullanılarak birliktelik kuralları şeklinde öğrenilecektir. 1993'de R.Agrawal ve ark. Tarafından önerilen iyi bilinen Apriori algoritması; Boolean birliktelik kuralları için sıklık öğe kümeleri madenciliği için etkili bir algoritmadır. Bu nedenle birliktelik kuralları madenciliği çalışması genel kabul görmüş olan Apriori algoritması ile gerçekleştirilmiştir.

1.1. Tezin Konusu, Amacı ve Önemi

Bu tezin asıl konusu "Veri Madenciliğinde Bulanık Mantık Uygulaması"dır. Bu nedenle Apriori algoritması kullanılarak gerçekleştirilmiş ve uygulamaya geçirilmiş olan veri madenciliği uygulamasının "kişi-kişi ilişkileri" kısmına bulanık mantık uygulaması gerçekleştirilmiştir.

Bu uygulamadaki amaç, kişi-kişi ilişkilerini aramak isteyen son kullanıcıların bu isimleri yazarken yapabilecekleri yazım hataları ya da ismin yazılışını bilmedikleri için yapabilecekleri yazım hatalarından dolayı istenilen sonuçların alınamamasını engellemektir. Yapılan uygulamanın önem kazanan kısmı da burasıdır. Yazılan isimlerin veri tabanımızdaki isimlerden en uygun olanı bulunarak bu isim üzerinden arama yapmaktadır. Bu sayede kullanıcılar hatalı isim girdiklerinde bile uygulama sonuç verebilecektir.

2. VERİ MADENCİLİĞİ

2.1. Veri Madenciliği Kavramının Tanımı, Gelişimi, Özellikleri

2.1.1. Veri Madenciliği Nedir?

"*Veritabanlarında bilgi bulma*"

Büyük veritabanlarından, ilginç (önemli) bilgilerin ya da örüntülerin bulunup çıkartılmasıdır.

"İlginç" (Önemli) bilgi ise sıradışı, önemli, kesin, önceden bilinmeyen ve yararlı olabilecek bilgidir.

Ya da temel olarak bilgisayar destekli bir bilgi çözümleme işlemi olarak da tanımlanabilir. (Baykal 2003)

2.1.2. Veri madenciliği ne <u>değildir</u>?

Sorgu işlemi değildir.

Uzman sistemler (*expert system*), öğrenen sistemler ya da istatistik programları değildir.

2.1.3. Veritabanı Teknolojilerinin Gelişimi

1960'lar: Veri toplama, veritabanları oluşturulması

1970'ler: İlişkisel veri modeli, ilişkisel veritabanı yönetim sistemleri, OLTP

1980'ler: RDBMS, ileri veri modelleri ve uygulama kaynaklı veritabanı yönetim sistemleri (uzamsal, bilimsel, mühendislik vb)

1999'lar-2000'ler: Veri madenciliği ve veri ambarlama, çoklu ortam veritabanları ve Web veritabanları. (Baykal 2003)

2.1.4. Neden veri madenciliği?

Veritabanı çözümlemesi ve karar desteği için. Örneğin, pazar araştırması ve yönetim, risk çözümlemesi ve yönetim, hile araştırması vb., metin ve web çözümlemesi, akıllı sorgular vb.

2.1.5. Neden Veri Madenciliğine Gereksinim Duyuldu?

Veri patlaması: Otomatik veri toplama araçları ve veri tabanı teknolojilerindeki gelişme, veritabanlarında, veri ambarlarında ve diğer bilgi depolarında çok miktarda bilgi depolanması sonucunu doğurmuştur. (Baykal 2003)

Çok fazla veri var, ancak bilgi yok…

Çözüm: Veri ambarları ve veri madenciliği

Büyük miktarlardaki veriler içindeki gizli örüntüler, geleneksel çözümleme araçlarıyla bulunamaz. Toplanan veri miktarı büyüdükçe ve toplanan verilerdeki karmaşıklık arttıkça, daha iyi çözümleme tekniklerine olan gereksinim de artmaktadır. Bu tür bilgiler, ***bilgi bulma/keşfetme (knowledge discovery)*** ya da ***veri madenciliği (data mining)*** olarak bilinen teknikler yardımıyla çözümlenebilir. (Baykal 2003)

2.2. Veri madenciliği ne tür veriler üzerinde yapılır?

- İlişkisel veritabanları
- Veri ambarları
- Gelişmiş veritabanları ve bilgi depoları (nesne kaynaklı, nesne ilişkili, uzamsal, metin, çoklu ortam, heterojen veritabanları, zamansal veriler ve WWW.

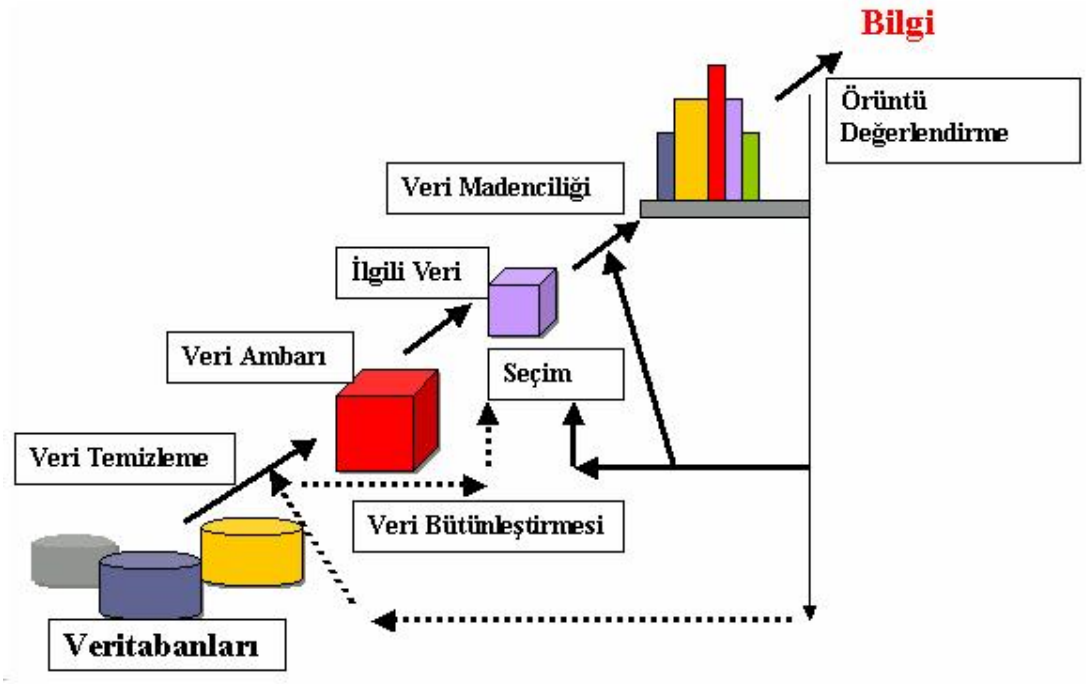
2.3. VERİ MADENCİLİĞİNDE ÇÖZÜMLEME YAKLAŞIMLARI

Veri madenciliğinde çözümleme iki şekilde gerçekleştirilir: (Baykal 2003)

1. **Yukarıdan aşağıya**
2. **Aşağıdan yukarıya**

2.3.1. Aşağıdan yukarıya: Bu yaklaşımda, gizli eğilimleri ve grupları bulma amacıyla ham verinin çözümlemesi yapılır.

2.3.2. Yukarıdan aşağıya: Bu yaklaşımda ise belirli varsayımlar denenir.



Resim 2.1. Veri Madenciliği Süreci

2.3.3. Yöntemler

Veri madenciliği, çeşitli tekniklerle gerçekleştirilebilir. Bunlar, akıllı araçlar, güçlü veritabanı sorguları ve çok boyutlu çözümleme araçlarıdır. Çok boyutlu çözümleme yöntemlerinde, ***sinir ağları*** kullanılabilir. (Baykal 2003)

2.3.4. Özellikler

- Elde edilen (ya da edilemeyen) sonuçlar, verinin anlamına ilişkin öngörüşlerden etkilenir. Örüntüler ve eğilimler de böylece ortaya çıkarılır.

- Veri madenciliği asıl olarak veritabanlarıyla ilgilidir. Veritabanlarını kullanıcı için daha kolay hale getirerek, büyük miktarlardaki veriyi yönetmeyi kolaylaştırır.
- Veri madenciliğinde girdi ve çıktı kolay anlaşılır ve kullanıma uygun olmalıdır. Bu yüzden görselleştirme tekniklerine büyük bir gereksinim vardır, çünkü karmaşık sonuçlar daha iyi anlaşılabilmesi için görselleştirilebilir.
- Bir başka sorun da veri madenciliği algoritmalarının iç çalışmasıyla ilgilenmeyen ve buna da gerek duymayan kullanıcıları göz önüne almaktır.
- En önemli sorun ise, **ölçek** sorunudur. Veri madenciliği alanındaki birçok algoritma, orta büyüklükte veritabanlarıyla iyi çalışır, ancak çok büyük veritabanlarıyla zorluk çeker. Çünkü büyük veritabanlarında öznitelik ve nesne sayısı çok fazladır.(Örneğin 200 milyon telefon görüşmesinden oluşan bir veritabanını yönetebilecek bir algoritma yoktur.)
- Yani veri madenciliği, veri boyutlarının küçültülmesi ve modeller oluşturulmasıyla ilgilidir.
- Veri madenciliği algoritmalarının çoğu da büyük veritabanlarından veri gruplarının seçilmesinden sonra, bu azaltılmış veri gruplarını yönetebilir. (Megaputer 2000)

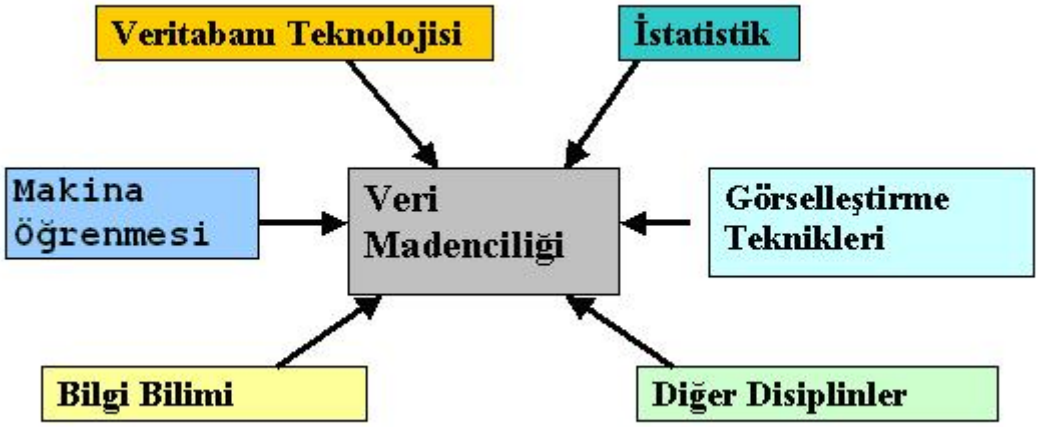
2.3.5. Çeşitli Tanımlamalar

- VM yöntemleri, verinin hazırlanmasından sonra, model oluşturma ya da örüntü/ eğilim bulma için veritabanlarında kullanılan algoritmalardır.
- "Veritabanlarında bilgi bulma, verideki geçerli, kullanılabilecek ve anlaşılır örüntüleri tanımlamadır." (*Frawley et al*., 1991).

- Belirli bir veri üzerinde VM alanındaki birçok yöntem kullanılabileceği için, bilgi bulma süreci zordur ve yeni, kullanışlı bilgi bulmak için ne yapılabileceği konusunda hiçbir kuram yoktur (*Cios* et al., 2000).

2.4 Veri Madenciliğinin Diğer Disiplinlerle İlişkisi

Veri madenciliğinin, birçok disiplinle çakışma noktaları vardır. (Baykal 2003)



Resim 2.2. Veri Madenciliğinin ilişkili olduğu diğer disiplinler

2.4.1. İstatistikle karşılaştırılması:

- Metotları farklıdır.
- Verilerin boyutları farklıdır.
- Ancak veri madenciliği, yalnızca heterojen sayılarla değil, heterojen veri alanlarıyla da ilgilenir. Örneğin tıbbi veriler, görüntüler, EKG

gibi sinyaller, ateş, kolesterol düzeyi vb klinik bilgiler ve standart olmayan bir dille yazılmış hekim yorumundan oluşabilir.

- Veri madenciliği, başka birçok disiplinin, özellikle de bilgisayarlı öğrenme/makine öğrenmesi (*machine learning*) ve veritabanı teknolojilerinin de ilgisini çekmektedir. (Hand J. D. 1998)

2.4.2. Veri madenciliği yöntemleri ikiye ayrılabilir:

- **Model oluşturma:** Machine learning ile kural üretme, sinir ağları ve gerileme (regression)
- **Örüntü algılama:** Normal EKG dalgalarındaki sapmaların algılanmasından, alışveriş ya da yolculuk örüntülerine kadar değişebilir. (Megaputer 2000)

2.5. VM Türleri

En üst düzeyde, veri madenciliğinin 3türü vardır. (Megaputer 2000)

a. Doğrudan veri madenciliği

b. Varsayım deneme ve varsayımı daha iyi hale getirme

c. Dolaylı veri madenciliği

2.5.1. Doğrudan veri madenciliği

Bu yöntemde, örneğin hekim, belirli bazı bilgilerin edinilmesiyle (örneğin tıkalı damarların bulunması) ilgilidir. Birçok tıbbi uygulama bu kategoriye girer.

2.5.2. Varsayım deneme ve varsayımı daha iyi hale getirme

Kullanıcı bazı varsayımlar üretir ve bunun sistem tarafından doğrulanmasını, değiştirilmesini ve daha uygun olabilecek yenilerinin önerilmesini bekler.

2.5.3. Dolaylı veri madenciliği

Sistemde hiçbir kısıtlamanın ve kullanıcının bulacağı bilgi ya da ne tür bir bilgiyle ilgilenildiğine ilişkin bir beklentinin olmadığı en genel senaryo budur. Bu aynı zamanda en güç yöntemdir.

2.6. BİLGİ BULMA SÜRECİ

2.6.1. Bilgi bulma sürecinin önemli özellikleri

a. İnsan-sistem etkileşimine dayanır. Sistemin alanla ilgili tüm bilgilere sahip olması ve kullanıcının amacını tanıyabilmesi gerektiğinden, bilgi bulma sürecinin tümünü kafamızda canlandırmak çok zordur.

b. Bilgi bulma işlemi, büyük veritabanlarıyla ilgilidir; bunlara erişmek ve işlemek için gereken yardımı sağlar, ancak büyük dosyalarla çalışmanın etkin yollarını gerektirir.

c. Bilgi bulma işlemi, disiplinler arası araştırmalara dayanır ve tamamlayıcı nitelikte bilgi teknolojileri gerektirir. Bu disiplinler arasılık, bilgi bulma yöntemlerini istatistik, makine öğrenmesi ve gerileme çözümlemesi gibi araçlardan ayırır. (Nariman A. 2005)

2.6.2. Bilgi bulma sürecinin önemli Sorunlar

a. Veritabanlarının büyük boyutları

Sırf veritabanlarının boyutu yüzünden, veri madenciliği yöntemlerinden herhangi birinin ham veriyle başarılı olma olasılığı yoktur. Veri madenciliği yöntemleri, bu şekilde elde edilen sonuçların tüm veritabanını temsil edebileceğini umarak, veritabanından bir örneğin çıkarılmasını gerektirebilir. Bir veritabanının boyutlarının küçültülmesi iki yolla olabilir: (Nariman A. 2005)

a.1.Veri alanında örnekleme; genellikle rasgele bazı kayıtlar seçilir ve veri madenciliğinin sonraki aşamalarında kullanılır.

***a.2. Özellik alanında örnekleme*;** her veri kaydının bazı özellikleri seçilir. Yine, birçok özellik varsa, seçim rasgele yapılır.

b. Dinamik veri yapısı

Veritabanları, belli aralıklarla güncellenir; yeni kayıtlar eklenir; örneğin, yeni bir SPECT görüntüsü eklenir (aynı hastaya ait yeni bir görüntü ya da yeni bir hastaya ait görüntü) ya da var olanlar yenileriyle değiştirilir (örneğin, bir görüntünün teknik nedenlerle değiştirilmesi gerekebilir). (Nariman A. 2005)

c. Eksik ya da kesin olmayan veri

Veritabanlarında toplanan bilgiler eksik olabilir ya da kesin olmayabilir. Sorunu çözmek için bu tür veriler geliştirilir. (Nariman A. 2005)

d. Gürültü

Herhangi bir veri toplama tekniğinin, gürültüden tümüyle arınmış olması çok zordur. Bu nedenle, veri madenciliğinde, gelecekte toplanacak verideki gürültü miktarının yaklaşık olarak o anki veridekiyle aynı olmasına dikkat gösterilmelidir. (Nariman A. 2005)

e. Eksik değerler

Bir değer bilinmiyor ya da yanlışlıkla girilmemiş olabilir. Veri madenciliğindeki birçok yöntem, her veri nesnesi için sabit bir boyut (özellik sayısı) gerektirdiğinden, eksik değerler sorun yaratır. (Nariman A. 2005)

Eksik değerler sorununu çözmek için:

- Eksik değerlerin yerine en olası değerleri koymaktır.
- Bilinmeyen değerlerin yerine o öznitelik için olası tüm değerleri koymaktır.

f. Gerekenden fazla, anlamsız ya da tutarsız veri

Veri grubunda gerekenden fazla, anlamsız ya da tutarsız veri nesneleri ve/veya öznitelikleri olabilir. Aynı veri öğesi, birden çok kategoriye aitse, tutarsız veri söz konusudur. (Nariman A. 2005)

2.6.3. Bilgi Bulma Sürecinde Kullanılan Yöntemler

Bilgi bulma sürecinde çeşitli yöntemler önerilmektedir. Bunlardan biri 4 temel aşamaya ayrılmaktadır: (Nariman A. 2005)

- **Seçim**
- **İşlem öncesi**
- **Veri madenciliği**
- **Yorum**

2.6.3.1. Seçim

Hedef bir veri grubu yaratılır. Modern veri grupları çok büyük ve karmaşıktır. Bu nedenle, araştırmanın amacını yansıtan verileri içeren hedef bir veri grubu yaratılır. (Nariman A. 2005)

2.6.3.2. İşlem Öncesi

Bu aşamada, kullanılacak veri grubu, veri madenciliği yazılımı tarafından çözümleme için hazırlanır. Bunun için; Veri madenciliği yönteminde girdi olarak, veritabanındaki hangi bilgilerin kullanılacağına karar verilmesi gerekir. Sonra, seçilen veriler temizlenir. (Nariman A. 2005)

2.6.3.2.1. İşlem Öncesi Süreçler (Ön Hazırlık)

Veri, özel veri madenciliği yöntemlerinin gereklerini karşılamak üzere yeniden biçimlendirilir. Bu aşamada, kayıtların ve özniteliklerin yeniden düzenlenmesi gerekebilir. Veri şifrelenir. Eksik veri (tamamlanmamış alanlar), ilgisiz alanlar, değişken olmayan alanlar, eğrilmiş alanlar ve uzak veri noktaları gibi, verideki istenmeyen özelliklerle ilgili sorunların çözülmesi gerekir. ***Tamamlanmamış veri alanları*****,** veri grubunda bilgi içermeyen kayıtlardır. Bazı görüşlere göre, bu oran % 70 ya da daha fazla ise bu alan tamamlanmış olarak kabul edilir. Bazı görüşlere göre ise bu oran düşüktür. Bu sorunun çözümü için, görece daha fazla alan içeren kayıtlardan oluşan veri grupları önerilmektedir. ***İlgisiz alanlar,*** çözümlemenin amacı bağlamında önemsiz kabul edilen değerlerin olduğu alanlardır. (Örneğin veri grubunda, belirli bir nedenle hastaneye kabul edilen hastaların adları, telefon numaraları gibi bilgiler bulunabilir.) Bir alanın ilgisiz olduğu kararı dikkatle verilmelidir. ***Eğrilmiş alanlar*****,** değerleri, veri grubunda normal dağılım göstermeyen alanlardır. Değerlerin eğri dağılımını gösterecek bir değişken bulunursa, bu değişken temizlenmiş veride göz ardı edilmelidir ya da eğri, uygun bir matematiksel işlevin uygulanmasıyla düzeltilmelidir. ***Uzak veri noktaları*****,** belirli bir veri alanının normal değer aralığı dışındaki yerlerdir. Bu tür veri noktaları, elde edilecek sonuçları bozabilir. Bu tür kayıtların veri grubundan çıkarılması önerilmektedir. Çıkarılan

veriler ilginç veriler içerebilir, ancak bu veriler geleneksel anlamda incelenebilir ve otomatik (veri madenciliği) işlemin bir parçası olmayabilir. (Nariman A. 2005)

2.6.3.3. Veri Madenciliği Aşaması

Veri modelleme teknikleri seçilir, eğitim ve deneme yordamlarına karar verilir, model oluşturulur ve kalitesi değerlendirilir. Temizlenmiş veri, gizli örüntü ya da eğilimlerin tanımlanması ya da belirli varsayımların denenmesi amacıyla, çözümleme için veri madenciliği yazılımına gönderilir. (Nariman A. 2005)

2.6.3.4. Yorum

Tüm bilgi bulma süreci çözümlenir. Sonuçların anlaşılmasına çalışılır, yeni bilgilerin ilginç olup olmadığı denetlenir, sonuçlar söz konusu alan açısından (tıbbi açıdan) yorumlanır ve bunların amaç üzerindeki etkileri araştırılır. Bu aşama, araştırılan alanda uzman olan bir kişi tarafından yürütülür. (Nariman A. 2005)

2.7. Veri Madenciliği Teknikleri

Veri madenciliğinde birçok farklı yöntem ve teknik kullanılır. Bu teknikler, veriyi kümeleme, sınıflandırma, kavramların tanımlanması, eksik değerleri tahmin vb amaçlarla kullanılır.

2.7.1. Kümeleme ya da Parçalama: Temel amaç, çok boyutlu veriler içindeki doğal grupları (kümeleri) bulmaktır. Nesneler, birbirlerine benziyorlarsa

(aynı ölçüye göre) ve başka kümelerdeki nesnelere benzemiyorlarsa, aynı kümeye alınabilir. Kümelemede, alan bilgisinin kümeleme mekanizmalarıyla nasıl birleştirilebileceğiyle ilgilenilir. (Tapan S., Zhao D. 2004)

2.7.2. Sınıflandırma: Bu terimin örüntü tanımada kaynakları vardır. Amaç, yeni bir nesnenin, belirli sınıflar içinde hangi sınıfa ait olup olmadığını belirleyecek bir sınıflayıcı (*classifier*) oluşturmaktır. Sınıflayıcı, gerileme modellerinin özel bir biçimi olarak kabul edilebilir. (Tapan S., Zhao D. 2004)

2.7.3. Kavram Tanımlaması: Amaç, kavramların ya da kategorilerin anlaşılabilir tanımlarını üretmektir. Bu amaç için, bilgisayarlı öğrenme, kavramsal kümeleme, genetik algoritmalar ve bulanık mantık yöntemleri kullanılır (*Cios* et al 2000).

2.7.4. Bağımlılık Çözümlemesi: Bu çözümleme, veritabanındaki alanlar arasındaki ilişkilerin (bağımlılıkların) belirlenmesiyle ilgilidir. (Tapan S., Zhao D. 2004)

2.7.5. Bağlantı Çözümlemesi ya da İlişkiler: Amaç, öznitelikler ya da nesneler arasındaki ilişkileri bulmaktır. Bu ilişkiler, aynı ya da başka bir nesneye ait öznitelikleri içerebilir. Bu çözümleme, zaman içinde gerçekleştirilirse dizi çözümlemesi adını alır. (Tapan S., Zhao D. 2004)

2.7.6. Dizi Çözümlemesi: Bu çözümleme, bir dizi oluşturan verilerin modellenmesindeki sorunlara karşı kullanılır. Yöntemler, zaman dizileri çözümlemesi, zaman dizileri modelleri ve zamansal sinir ağlarıdır. (Tapan S., Zhao D. 2004)

2.7.7. Tahmin: Amaç, bir tahmin modeliyle bir nesnenin öznitelik değerini tahmin etmektir. Bu yöntem, varsayım deneme modelinde kullanılabilir. (Tapan S., Zhao D. 2004)

2.7.8. Keşfedici Veri Çözümlemesi: Bunun için genellikle grafik modeller kullanılır; örneğin, yeni örüntülerin bulunması için, insanın görsel tanıma özelliklerinden yararlanılabilir. (Tapan S., Zhao D. 2004)

2.8. Veri Madenciliği Tekniklerinin Modellenmesi

Veri madenciliğinde kullanılan teknikler, hem özel türde veri yapıları, hem de belirli algoritmik yaklaşımlar gerektirir. Bunlar, önce iki genel gruba ayrılabilir: (Sorin D. 2000)

- ***Parametrik modeller***
- ***Parametrik olmayan modeller***

2.8.1. Parametrik Modeller: Girdi ile çıktı arasındaki ilişkiyi, bazı değişkenlerin belirlenmediği cebirsel eşitlikleri kullanarak açıklar. Bu belirlenmemiş değişkenler, girdi örnekleri sağlanarak belirlenir. Parametrik modelleme bazen kullanılsa da gereken veri hakkında çok fazla bilgi gerektirdiği için, gerçek yaşamla ilgili sorunlarda kullanışlı olmayabilir. (Sorin D. 2000)

2.8.2. Parametrik Olmayan Modeller: Bu modeller, veri madenciliği için daha uygundur, çünkü bu modeller veriyi temel alır. Burada, modeli belirlemek için

hiçbir eşitlik kullanılmaz. Bu, modelleme işleminin, eldeki veriye uyumlu hale getirilebileceği anlamına gelir. Veriye göre bir model oluşturulur. (Sorin D. 2000)

Parametrik olmayan teknikler şunlardır:

- **Sinir ağları** ***(neural networks)***
- **Karar ağaçları** ***(decision trees)***
- **Genetik algoritmalar** ***(genetic algorithms)***

2.8.2.1. Parametrik ve Parametrik Olmayan Modellerin Karşılaştırılması:

- Parametrik modellemede, başlangıçta belirli bir model varsayılır. Parametrik olmayan modellemede ise girdiye göre bir model oluşturur.
- Parametrik modellemede, modelleme işleminden önce veri hakkında çok fazla bilgi gerekir, parametrik olmayan modellemede ise, modelleme işleminin kendisi için girdi olarak çok miktarda veri gerekir. (Sorin D. 2000)

2.8.2.2. İstatistiksel Yaklaşımlar

Veri madenciliği tekniklerinin temeli olan istatistiksel kavramlar vardır. Bunlar;

- **Sayı tahmini** (Estimation)
- **Özetlemeye dayalı modeller**
- **Bayes teoremi**
- **Varsayım sınama -hypothesis testing**
- **Gerileme ve korelasyon**
- **Benzerlik ölçümü**

2.8.2.2.1. Sayı Tahmini (Point Estimation):

Bir bağımsız değişkenin (*parameter*) tahmin etme işlemidir. Bu işlem, ortalama, değişken (*variance*), standart sapma gibi istatistiksel değerleri tahmin etmek için yapılır. Veri madenciliğinde, eksik değerlerin tahmini için kullanılır. (Sorin D. 2000)

2.8.2.2.2. Özetlemeye Dayalı Modeller

Özetlemede amaç, veriyi az sayıda özellik/öznitelikle karakterize etmektir. Verinin tümümün bir özetini sağlayan birçok temel kavram vardır. İyi bilinen temel istatistiksel kavramlar; ortalama, değişken, standart sapma, orta (*median*) ve kiptir (*mode*). Bir öğe grubunu belirli bir sıklık dağılımına (*frequency distribution*) sığdırmak, veri için daha iyi bir model sunar. Ancak bu, çok sayıda karmaşık özniteliğin (*attribute*) olduğu ve sürekli değişen büyük veritabanlarında pek kolay ve uygulanabilir bir şey değildir. (Sorin D. 2000)

2.8.2.2.3. Bayes Teoremi

Bir sınıflandırma sorununun olasılık terimleriyle açıklanabileceği varsayımına dayanır. Bayes kuralı (Bayes rule), bir veri grubunda bir özelliğin olasılığını tahmin etme yöntemidir; belirli bir veri değerinde çeşitli varsayımların olasılığını araştırır. (Sorin D. 2000)

2.8.2.2.4. Varsayım Sınama (Hypothesis Testing)

Bu teknik, önce bir varsayım oluşturup sonra da bu varsayımı veri üzerinde sınayarak gözlenen veriyi açıklayan bir model bulmaya çalışır. Bu, önce herhangi bir tahminde bulunmadan gerçek veriden bir model oluşturan birçok veri madenciliği yönteminin tersidir. Burada, model oluşturma sürecini gerçek veri yönetir. Varsayım genellikle bir veri örneği incelenerek doğrulanır. Varsayım bu örneğe uyarsa genelde tüm veriye uyduğu kabul edilir. Belirli bir veride sınanacak olan başlangıç varsayımına ***boş varsayım*** (*null hypothesis*) denir. Boş varsayım kabul edilmezse başka varsayımlar doğar. Bunlara da ***alternatif varsayımlar*** denir. (Sorin D. 2000)

2.8.2.2.5. Gerileme ve Korelasyon

Her iki teknik de iki değişken arasındaki ilişkinin gücünü değerlendirmek için kullanılabilir. ***Gerileme*** *(regression)*, genellikle geçmişteki değerleri temel alarak gelecekteki değerleri tahmin etmek için kullanılır. ***Doğrusal gerileme***, girdi verisi ile çıktı verisi arasında doğrusal bir ilişki olduğunu varsayar. ***Korelasyon***, iki değişkenin değerlerinin benzer davranma derecesini incelemek için kullanılır. Burada sorun, iki değişkenin birbirlerine ne kadar benzediğidir. Korelasyon doğrusal, pozitif ve negatif olabilir. Adlarından da anlaşılacağı gibi, pozitif korelasyon pozitif, negatif korelasyon ise negatif sayıların gösterdiği korelasyondur. Doğrusal korelasyonu ölçmek için kullanılan standart formül korelasyon katsayısıdır. İki değişken arasında güçlü bir korelasyon varsa, bunlar benzerdir. Yani, korelasyon katsayısı, kümeleme ve sınıflandırmada benzerliği tanımlamak için kullanılabilir. (Sorin D. 2000)

2.8.2.2.6. Benzerlik Ölçümü

Bu yöntem, soyutlanıp genel sınıflandırma sorunlarına uygulanabilir. Burada zorluk, benzerliğin nasıl ölçüleceği, tanımlanacağı ve veritabanındaki öğelere nasıl uygulanacağıdır. Birçok benzerlik ölçüsü, sayısal değerleri kabul ettiğinden, daha genel veri türlerinin kullanımında zorluk olabilir. Bunun için, özniteliklerle tamsayılar eşleştirilebilir (*mapping*). Böyle bir benzerlik eşleştirmesi nasıl tanımlanacaktır? "Benzemezlik" kavramı genellikle iyi tanımlanmamıştır. Benzerlik düşüncesi sınıfların önceden tanımlandığı sınıflandırma sistemlerinde kullanıldığında bu, sınıfların bilinmediği kümeleme yöntemindekinden daha kolay gibidir. (Sorin D. 2000)

2.8.2.3.Karar Ağaçları

Karar ağaçları *(decision trees),* sınıflandırma, kümeleme ve tahmin modellerinde kullanılan bir tahmin tekniğidir. Sorunla ilgili araştırma alanını alt gruplara ayırmak için kullanılır. Karar ağaçlarında kök ve her düğüm bir soruyla etiketlenir. Düğümlerden ayrılan dallar ise ilgili sorunun olası yanıtlarını belirtir. Her dal düğümü de söz konusu sorunun çözümüne yönelik bir tahmini temsil eder. Karar ağaçları, üç bölümden oluşan bir modeldir. (Baykal 2003)

- Tanımdaki gibi bir karar ağacı
- Ağacı oluşturacak bir algoritma
- Ağacı veriye uygulayacak ve söz konusu sorunu çözecek bir algoritma.

Karar ağaçları;

Eğitici örnekteki veriyi sınayan bir algoritma aracılığıyla gerçekleştirilir ya da Alanın bir uzmanı tarafından oluşturulur. Karar ağacı tekniklerinin çoğu, birbirlerinden ağacın nasıl oluşturulduğuyla ayrılır. (Baykal 2003)

2.8.2.4.Yapay Sinir Ağları

İlk kez 1943'te ortaya çıktı ama bilgisayarlarda kullanımı 1980'lerde başladı. ***Yapay sinir ağları*** *(artificial neural networks)*, beynin yapısından esinlenilmiş bir bilgi işleme sistemidir. Nöronlara benzeştirilmiş işlem öğeleri arasındaki ilişkilerle yapılandırılmıştır. İnsan beyni gibi yapay sinir ağı da birbirine bağlı birçok işlem biriminden oluşmuştur. Sonra, birçok düğüm (işlem birimi) ve arkla (iç bağlantılar) yönetilen bir grafik olarak yapılandırılır. Bu işlem birimleri birbirlerinden bağımsız işlev görürler ve yalnızca yerel veriyi (düğüme gelen girdi ve düğümden çıkan çıktı) kullanırlar. Bu özellik, sinir ağlarının dağıtık ya da paralel ortamlarda kullanımını kolaylaştırır. Sinir ağları, kaynak (*girdi*), *çıktı* ve iç (*gizli*) düğümlerle yönetilen bir grafik olarak görülebilir. Girdi düğümü *girdi katmanı*nda, çıktı düğümü ise *çıktı katmanı*nda bulunur. Gizli düğümler, bir ya da daha çok gizli katmanda bulunur. Veri madenciliğinde, çıktı düğümü tahmini belirler. Tek bir girdi düğümünün olduğu (ağacın kökü) karar ağaçlarından farklı olarak sinir ağlarında, her öznitelik değeri için bir girdi düğümü vardır. Sinir ağları karmaşık sorunları çözebilir, ayrıca temel uygulamalardan "öğrenebilir". Yani, soruna kötü bir çözüm bulunduysa, ağ bu soruna bir dahaki sefer daha iyi bir çözüm bulacak biçimde değiştirilir. Sinir ağları üç bölümden oluşur: (Baykal 2003)

- Sinir ağının veri yapısını tanımlayan sinir ağı grafiği.
- Öğrenmenin nasıl gerçekleşeceğini belirten öğrenme algoritması.
- Bilginin ağdan nasıl elde edileceğini belirleyen teknikler.

Yapay Sinir Ağları

- Örüntü tanımada
- Ses tanıma ve çözümlemede
- Tıbbi uygulamalarda (tanı, ilaç)
- Hata algılamada
- Sorun tanılamada
- Robot denetiminde
- Herhangi bir işlevi hesaplamada

Kullanılabilir.

Yapay Sinir Ağları, bağlantı ve öğrenme türlerine göre sınıflandırılabilir.

- ***İleri beslemeli bağlantı***da bağlantılar yalnızca yapıda daha sonraki katmanlaradır.
- ***Geri beslemeli bağlantı***da ise bazı bağlantılar daha önceki katmanlardır.

Yapay Sinir Ağları öğrenme türleri ise

- ***Denetimli*** (*supervised*) öğrenme
- ***Denetimsiz*** (*unsupervised*) öğrenme

Denetimli öğrenme, temel olarak iki aşamalı bir işlemdir:

- Sinir ağını, örnek dizileri göstererek verideki farklı sınıfları tanıyacak biçimde eğitmek.
- Önceden görmediği bir veri grubu sağlayarak sinir ağının bu örneklerden ne kadar öğrendiğini denemektir.

Denetimsiz öğrenmede ise sinir ağına, sunulan verinin doğru olarak sınıflandırılmasına ilişkin hiçbir ön bilgi verilmez. Sinir ağı, denetimsiz öğrenmeyi, o veride doğal olarak var olan kümeleri ve altkümeleri bulmak amacıyla çok boyutlu bir veri grubunu çözümlemek için kullanır. Sinir ağları denetimsiz öğrenme tekniği, sağlanan verinin yapısını temel alarak kendi sınıflandırma şemalarını tanımlamak için kullanır. Denetimsiz örüntü tanımaya bazen ***küme çözümlemesi*** *(cluster analysis)* de denir. (Baykal 2003)

2.8.2.5. Genetik Algoritmalar

Genetik algoritmalar *(genetic algorithms)*, ***evrimsel hesaplama*** (*evolutionary computing*) yöntemlerinin ve ***iyileştirme türü*** (*optimatization-type*) algoritmaların örnekleridir.

Evrimsel hesaplama algoritmasının temeli, en iyi uyum sağlayabilenin yaşayabilmesine dayalı biyolojik evrimdir.

Veri madenciliğinde genetik algoritmalar;

- Kümeleme
- Tahmin ve İlişki kuralları oluşturmak için kullanılabilir.

Bu teknikler, çeşitli modeller arasından, verinin gösterilmesine en "uygun" olanını bulmak olarak düşünülebilir.

Bu yaklaşımda birçok iterasyon arasından bir başlangıç modeli varsayılır ve modeller yeni modeller oluşturmak üzere birleştirilir.

Bunların en iyisi, uygunluk işlevi (*fitness function*) tarafından tanımlanarak, bir sonraki iterasyon için girdi olur.

Algoritmalar, birbirlerinden modelin nasıl gösterildiği, modeldeki farklı öğelerin nasıl birleştirildiği ve uygunluk işlevinin nasıl kullanıldığıyla ayrılır.

Bir sorunun çözümünde genetik algoritmalar kullanıldığında en zor iş, sorunun öğeler grubu olarak nasıl belirlenmesi gerektiğidir. Gerçek dünyada bireyler, DNA yapısının tam şifresi olarak tanımlanabilir ve bireysel farklılık, değer dizileri olarak görülebilir.

Genetik algoritmalarda yeniden üretim (*reproduction*), belirli bir öğe grubunun, yeni bir öğe üretmek için nasıl birleştirileceğini belirten kesin bir algoritmayla tanımlanır. Bunlara ***çaprazlama*** (*crossover*) algoritmaları denir. İki birey varsa (popülasyondaki *ebeveyn*), çaprazlama teknikleri, dizideki diğer sıralara geçerek yeni bireyler üretebilir (*çocuk*).

Çaprazlama yaklaşımının, çaprazlama noktalarının rasgele belirlenmesi de içinde olmak üzere birçok biçimi vardır. Çaprazlama olasılığı, çaprazlama ile kaç yeni ürün (*offspring*) yaratılacağını belirlemek için kullanılır. (Baykal 2003)

Çaprazlama noktası, bir algoritma içinde değişebilir.

Ancak doğada, bazen ***mutasyonlar*** görünür ve bunlar da genetik algoritmalarda gösterilebilir.

Mutasyon işlemi, çocuktaki (üründeki) karakterleri rasgele değiştirir. Mutasyonda karakterin değişip değişmeyeceğini belirleyecek çok küçük bir olasılık oluşturulur.

Genetik algoritmalar doğayı model almaya çalıştığı için, yalnızca güçlü olan yaşar. Yeni öğeler (bireyler) yaratıldığında, hangisinin yaşayacağının seçilmesi

gerekir. Bu yeni bireyler olabilir, eskileri olabilir ya da daha büyük olasılıkla ikisinin bir bileşimidir.

Genetik algoritmalarda üçüncü büyük bileşen, en iyi (en iyi uyum sağlayan) bireyleri belirleyen kısımdır.

En önemli bileşen, bireylerin nasıl seçileceğidir. Popülasyondaki en iyi bireyleri belirlemek için uygunluk işlevi kullanılır. Sonra bu, ebeveynin seçiminde kullanılır. Belirli bir amaç söz konusu olduğunda uygunluk işlevi, bu amacın birey tarafından iyi yerine getirilip getirilmediğini belirtir.

Genetik algoritma beş bölümden oluşur:

- Öğe grubunun (popülasyon) başlatılması
- Çaprazlama tekniği
- Mutasyon algoritması
- Uygunluk işlevi
- Popülasyondaki en iyi bireyleri belirlemek için uygunluk işlevini kullanarak çaprazlama ve mutasyon tekniklerini popülasyona uygulayan algoritma.

Olası tüm bireylerin aranması, en iyi bireyin ya da çözümün bulunmasıyla sonuçlanır. Aramanın kapsamı çok geniş olduğundan, sorunu çözemeyecek bireyler arama kapsamından çıkarılır. Ayrıca, daha önce sınananlardan çok farklı olabilecek bireyler de yaratılır.

Genetik algoritmalar tüm alanı aramadığından, en iyi sonuç ortaya çıkmayabilir. Ancak, zor sorunlara uygun çözümler üretebilir.

Bazı dezavantajları:

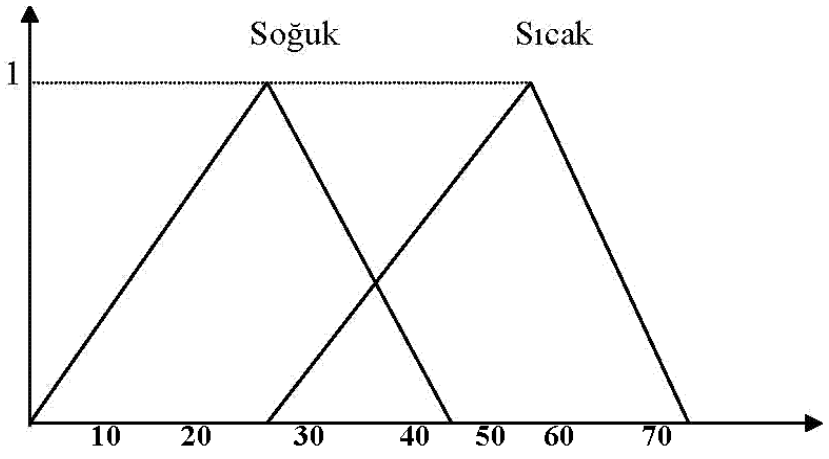
- Kullanıcıya açıklanması ve anlatılması zordur.
- Sorunu soyutlamak ve bireyleri (öğeleri) temsil etmek için kullanılan modeller zordur.
- Uygunluk işlevini belirlemek zordur.
- Çaprazlama ve mutasyonun nasıl yapılması gerektiğine ilişkin sorunun çözümü zordur. (Baykal 2003)

3. BULANIK MANTIK

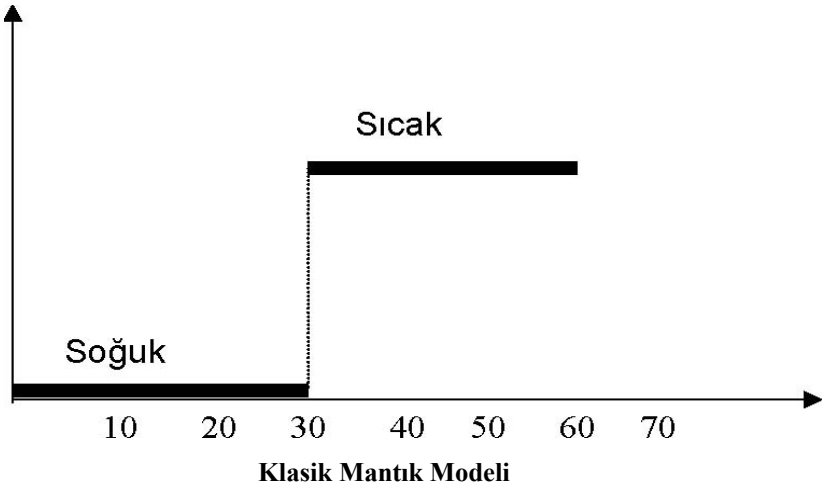
3.1. Bulanık Mantık (Fuzzy Logic) Nedir?

Bulanık mantık, bir şey hakkında yargı ortaya atarken, aynı anda, bu yargıyı oluştururken dayandığı matematiksel sınıflandırmaların ne kadar içinde, ne kadar dışında olduğundan bahseder... Verinin ne kadar o yargı kümesine ait, ne kadar ait olmadığı bilgisine dayanarak o veriye yeni bir tanım getirir. Örnek olarak, farz edin ki işyerinizde klimanızın otomatik devreye girme sistemi bozuldu ve bu nedenle içerideki sıcaklığı dengede tutamıyorsunuz. Çalışmak için içerisi "çok sıcak" ... Sekreterinizi çağırıp, ondan sürekli olarak termometreyi gözlemlemesini ve içerideki sıcaklığın 30 C^0 'yi geçmemesini sağlamasını istediğinizi söylüyorsunuz... Bu durumda muhtemelen sekreteriniz, bu sorunu klimanın kumandasıyla, klimayı açıp kapatarak çözebileceğini düşünecektir. Sekreteriniz klasik bir yaklaşım gösterir ise, termometre 30 C^0 ye yakın bir değeri gösterdiğinde klimayı çalıştırması gerektiğini düşünecek, içerisinin yeterince serin olduğu kanısına varınca da klimayı kapatacaktır. Eğer ki sekreteriniz akılcı bir yaklaşım gösterir ise, oda sıcaklığı için uygun sabit bir değer belirleyecektir. Bu değer aşıldıkça klimayı düşük bir güç değeri ile çalıştıracak ve bu gücü sıcaklık yükselişinin duraksadığını ve ardından gerilemenin başladığını gözlemleyinceye kadar yavaşça artırmaya devam edecek, bu sabit sıcaklık, değerinin altına inildiğinde bu kez aynı işlemleri ve özveriyi! tersten bir mantıkla sıcaklık artışı için yapacaktır... İşte Bulanık Mantık, Aristotales'in "Sadece doğrular ve yanlışlar vardır" mantığına alternatif olarak kendini ifade eder. Modern Teknolojinin kullandığı kodlama biçimi olan 0,1 mantığına karşın Bulanık Mantık, 0 ile 1 arasındaki değerlerin varlığından bahseder... Klasik Mantık 30 C^0 'yi "sıcak" kümesinin sınırı olarak kabul ediyorsa, 29,9 C^0 'yi sıcak olarak kabul etme hakkını kaybeder... Oysaki aradaki bu küçük fiziksel fark, Klasik Mantık için hayati anlam

ifade etmektedir... Çünkü bu değerin üyelik kümesi değişmiştir artık. 30 C^0 , 29,9 C^0 olmakla, "sıcak" olmayı reddetmiş ve bunun sonunda "sıcak" olma kümesinden dışlanmıştır... Bu sosyal manada oldukça klasik bir tutum olsa gerek... Hangi birimiz fiziksel dünyada 30 C^0 yi sıcak kabul ederken, 29,9 C^0 'nin sıcak olmadığını iddia ederiz ki... Oysaki Bulanık Mantık bu tür keskin sınırları kaldırarak, 29,9 C^0 'yi "hemen hemen" tamamen (1'e yakın bir değerle) sıcak olarak kabul eder. Fiziksel dünya ile bilimsel dünyanın paralel bakış açısıyla çalışan bir sistemi ortaya atar Bulanık Mantık... Klasik mantık için "soğuk" ya da "sıcak" olma vardır... Oysaki Bulanık Mantık "soğuk-sıcak" gibi kavramların yanında, "az soğuk", "çok sıcak", biraz sıcak" gibi söylemleri de kabullenir ve bunları matematiksel olarak tanımlamaya çalışır... Böylece fiziksel hayat birdenbire matematiksel bir söyleme dönüşüverir ya da matematik hayatın ta kendisi oluverir... (Koyuncu E. 2004)



Bulanık Mantık Modeli



Şekil 3.1. Bulanık Mantık ile Klasik Mantık Modelleri (Koyuncu E. 2004)

Bulanık Mantığın tarihi çok eskilere dayanmaktadır. Aristoteles'in "Var ya da Yok" yasalarına karşın Heraclitus, bir şeyin hem doğru hem yanlış olabileceği fikrini ortaya sürmüştür. Plato ise bu durumu ileriye götürerek "doğru" ve "yanlış" olmanın dışında, doğru ve yanlışın iç içe olduğu üçüncü bir durumdan bahseder. Ancak ilk kez Lukasiewicz 1900'lerin başında "olası" kavramını ortaya atmıştır. Bu kavram Bulanık Mantığın temelini oluşturur. Lukasiewicz, Doğru ile Yanlış arasında sonsuz farklı değer olduğundan bahsetmiş ve ancak bu mantık uygulamalarda çok başarı elde edememiştir. Nihayet 1965 yılında Lotfi A. Zadeh, bu değerleri [0.0, 1.0] aralığındaki sayılarla ifade ettiği teorisinin adına "Bulanık Mantık (Fuzzy Logic)" ismini vermiştir. (Koyuncu E. 2004)

Bulanık Mantık çözümleme yaparken, bilimin kendine has, insanların öznel tecrübelerini küçümseyen tarzından farklı olarak, gözlemlere yer verir. Odanın sıcak oluşu öznel bir gözlemdir. Bu konuda ortaya bir yargı atabilmek için bu gözleme tabi ki ihtiyaç vardır. Ancak sınırları da aynı öznel tecrübe belirlemelidir. Odanın sıcaklığının ne olması gerektiğine sekreteriniz karar verir. Bu durumda Klasik Mantık işlemdeki "sıcak" oluş, öznel bir gözlem olsa da sınırlar ve ara geçişleri matematik tarafından (Aristotales tarafından...) belirlenir. Bulanık Mantık işlemlerinde ise yargılar öznel (bilirkişi) gözlemlere dayanır ve bu yargılar

matematiksel olarak analize hazır bir şekilde modellenir. Matematik kural olmaktan çıkar, hayat için birkaç rakam oluverir... (Koyuncu E. 2004)

Bulanık Mantığın tıpkı matematik gibi uygulamasının olmadığı bir alandan bahsetmek çok zor... Endüstriyel Sistem modellemelerinden, yazılım geliştirmeye; otomatik kontrol sistemlerinden, veri analizine; yöneylem araştırma tekniklerinden, sosyolojik değişim kurallarını izleme gibi birçok alanda Bulanık Mantık uygulamalarını başarılı bir şekilde görmek mümkün. Özellikle Modern Kontrol Sistemleri, Bulanık Mantık bilimini üstlenmiş durumda... Bu beraberlikten en çok yarar gören başarılı uygulamalarıyla Otomatik Kontrol Sistemleri bilimi gibi görünüyor. Bunun yanında Bulanık mantık önüne çıkan daha karmaşık problemlerle kendini ispatlama fırsatını yakalıyor... Örnek olarak, günümüzde Robotik Hareket Sistemlerinin karmaşık kontrol problemleriyle çoğunlukla Bulanık Mantık ilgileniyor... (Koyuncu E. 2004)

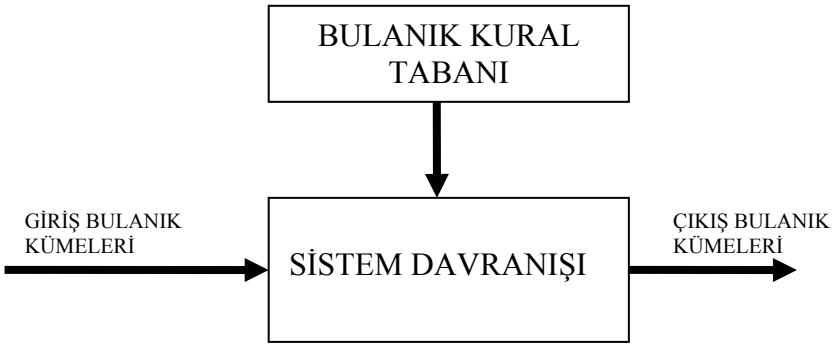
3.2. Bulanık Sistem Ne Demektir

Klasik sistemlerin hemen hepsi Şekil 3.2 'de verilen üç ayrı birimden ibarettir.



Şekil 3.2 Klasik Sistem

Bu kısımlar giriş, sistem davranışı ve çıkış kısımlarıdır. Bulanık sistemlerin bu klasik tasarımdan farkı, Şekil 3.3'te gösterildiği gibi sistem davranışı kısmının ikiye ayrılarak aralarında bağlantılı dört birimin olmasıdır. (Zekai Ş. 2001)



Şekil 3.3 Genel Bulanık Sistem

Burada bulunan birimlerin her birinin farklı, fakat birbiri ile ilişkili olabilen aşağıdaki görevleri vardır.

- Genel Bilgi Tabanı Birimi: incelenecek olayın maruz kaldığı girdi değişkenlerini ve bunlar hakkındaki tüm bilgileri içerir. Buna veri tabanı veya kısaca giriş adı da verilir. Genel veri tabanı denmesinin sebebi buradaki bilgilerin sayısal ve/veya sözel olabilmesidir. (Zekai Ş. 2001)

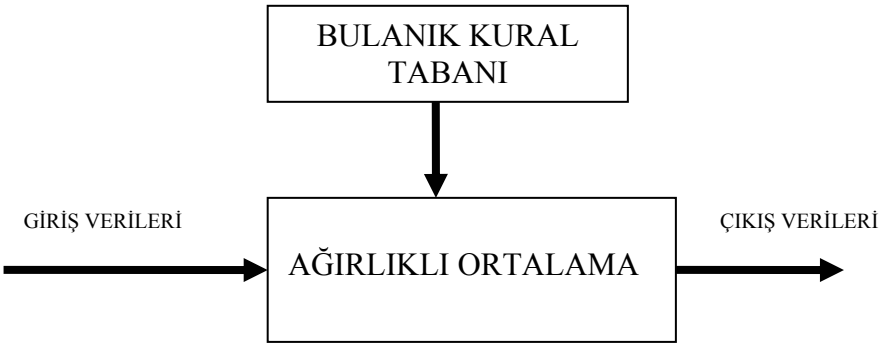
- Bulanık Kural Tabanı Birimi: Veri tabanındaki girişleri çıkış değişkenlerine bağlayan mantıksal EGER-İSE türünde yazılabilen bütün kuralların tümünü içerir. Bu kuralların yazılmasında sadece girdi verileri ile çıktılar arasında olabilecek tüm aralık (bulanık küme) bağlantıları düşünülür. Böylece, her bir kural girdi uzayının bir parçasını Çıktı uzayına mantıksal olarak bağlar. İşte bu bağlamların tümü kural tabanını oluşturur. (Zekai Ş. 2001)

- Bulanık Çıkarım Motoru Birimi: Bulanık kural tabanında giriş ve çıkış bulanık kümeleri arasında kurulmuş olan ilişkilerin hepsini bir araya toplayarak sistemin bir çıkışlı davranmasını temin eden işlemler topluluğunu içeren bir mekanizmadır. Bu motor, her bir kuralın çıkarımlarını bir araya toplayarak tüm sistemin girdiler altında nasıl bir çıktı vereceğinin belirlenmesine yarar. (Zekai Ş. 2001)

- Çıktı Birimi: Bilgi ve bulanık kural tabanlarının, bulanık çıkarım motoru vasıtası ile etkileşimi sonunda elde edilen çıktı değerlerinin topluluğunu belirtir. (Zekai Ş. 2001)

Şekil 3.3 genel bir bulanık sistemi temsil eder. Burada dikkat edilmesi gereken bir nokta genel olarak girdi yani veri tabanındaki bilgilerin ve çıktıların bulanık değerler olmasıdır. Yani Şekil 3.3'teki sistemde, her birim tamamen bulanık kümelerden oluşmaktadır. Temel bulanık sistemin en önemli mahzuru, sayısal olan veri tabanının böyle bir genel bulanık sisteme girememesi ve çıktıların sayısal olmaması, dolayısı ile mühendislik tasarımlarında doğrudan kullanılamamasıdır. (Zekai Ş. 2001)

Genel bulanık sistemin mahzurlarını bir dereceye kadar ortadan kaldırabilmek için *Takagi ve Sugello(1985) ve Sugello ve Kallk* (1988) tarafından teklif edilen ve Takagi-Sugeno-Kank (TSK) bulanık sistemi denilen sistem kullanılır. Burada veri tabanındaki girdiler birer sayı, bulanık kural ve çıkarım motorunun çalışması sonunda elde edilen çıktılar ise girdilerin bir fonksiyonu şeklindedir. Yani kural tabanındaki öncül kısımların değişkenleri olduğu gibi, İSE kelimesinden sonraki kural soncul kısmına bu değişkenlerin birer doğrusal fonksiyonu olarak yansıtıldığı düşünülmüştür. Mesela, 3 tane öncül değişkenin (xı, X2 ve X3) bulunması halinde soncul değişken olan y genel olarak bulanık sistemin kurallarından birinde EGER Xı çok ve Xı yüksek ve X3 dar İSE y = 3X1 + X2 + 2X3 şeklinde ifade edilebilir. Bütün kuralların soncul kısımları sanki çok terimli bir doğrusal denklemden ibarettir. Böyle bir yapıya sahip olan bulanık sistemde soncullar bulanık küme şeklinde olmadıklarından Şekil 3.3'teki Bulanık Çıkarım Motoru birimi yerine, her bir kuralın öncül kısmından hesaplanan üyelik dereceleri ağırlık olmak üzere ağırlıklı çıkarım hesaplaması birimi gelir (Şekil 3.4).

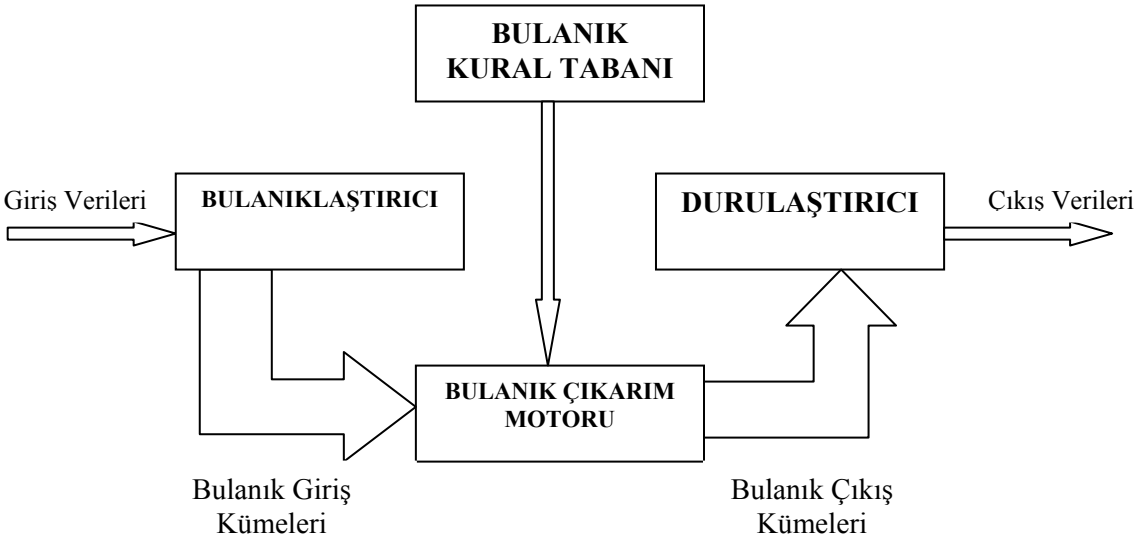


Şekil 3.4 TSK Bulanık Sistem

Aslında böyle bir bulanık sistemde çıktı uzayı girdilerin fonksiyonu olarak, her bir alt uzayda geçerli bir kural olmak üzere temsil edilmiştir. TSK yaklaşımı ile çıktı yüzeyinin doğrusal olmaması halinde bile, bu yüzeyin alt uzaylar üzerinde girdi değişkenleri cinsinden düzlem parçaları şeklinde modellendiği anlaşılır. Ancak, TSK bulanık sisteminin mahzurları arasında İSE kısmından sonra matematik bir ilişki bulunduğundan, kuralların soncul kısımlarının insan tarafından verilecek sözel bilgileri modelleyememesi ve giriş-çıkış değişkenleri arasında yazılması mümkün olan tüm kuralların soncul kısımlarının bulanık olmaması dolayısı ile yazılamamasıdır. İşte bu mahzurları ortadan kaldırabilmek için Şekil 3.5'te verilen ve girdi ve çıktı birimlerinde sırası ile bulanıklaştırma ve durulaştırma işlemleri yapıldığından bu birimlerin de kutu şeklinde gösterildiği bir bulanık sistem karşımıza çıkar.

Burada genel bir bulanık sistemdeki bulanık kural tabanı ve çıkarım motoru aynen kalmaktadır. Girişlerin sayısal olanları durumunda bir işleme tabi tutularak bulanıklaştırılmasına yarayan bulanıklaştırıcı birim ile yine bulanık olan çıktıların sayısallaştırılmasına yarayan durulaştırıcı birim ilave edilmiştir. Bulanıklaştırma ve durulaştırma sırası ile giriş sayılarını bulanıklaşması ve bulanık sayıların sayısallaştırılması anlamına gelir. Bu sisteme, bulanık sözel bilgiler ile bulanıklaştırılmış sayısal bilgiler bir arada toplanarak, sanki Şekil 3.3'te gösterilen genel bulanık sistemin girdisine indirgenmiş bir durum ortaya çıkar. Bulanık sistem çıkışlarının mühendislik tasarımlarında kullanılması amacı ile sayısallaştırılması için

durulaştırma birimi ilave edilmiştir. Bu bulanıklaştırıcı durulaştırıcı sistem, genel ve TSK bulanık sistemlerinde bulunan tüm mahzurları ortadan kaldırır. (Zekai Ş. 2001)

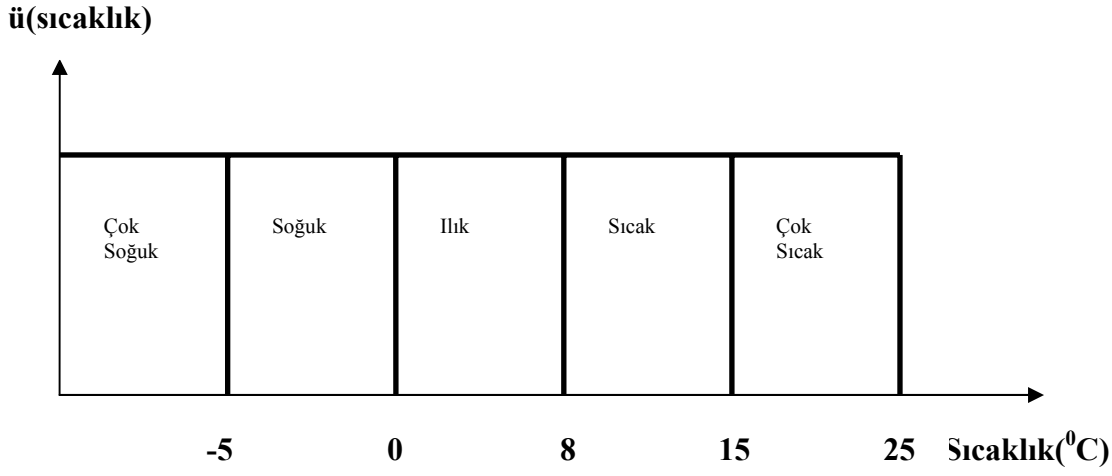


Şekil 3.5 Bulanıklaştırma-Durulaştırma birimli bulanık sistem

Bulanık sistemlerin başlıca özellikleri arasında en önemli, konu olarak, çoklu girdileri, kural tabanı ve çıkarım motoru ile işleyerek tek çıktı haline dönüştürmesi gelir. Bazı özel durumlarda, çıktılar birden fazla olabilir. Ancak, hemen her mühendislik çalışmasında en az bir tane çıktı bulunur. Bulanık sistem doğrusal olmayan bir şekilde girdileri oluşturan değişkenleri, Çıktı değişkenine dönüştürerek, sistemin davranışını tespit eder. Böylece bilgi tabanının doğrusal olmayan dönüşümlere maruz bırakılması ile istenen sonuçlara ulaşmak için incelenen sistemin kontrol altına alınması mümkün olmaktadır. Bulanık sistemler sayesinde mühendislikte görüntü işleme, zaman serileri esaslı tahmin yapmak, kontrol sorunlarını çözmek ve haberleşme yani iletişim konularında uygulamalar yapmak mümkün olmaktadır. Bunun dışında bulanık sistemler mühendislik, tıp, sosyoloji, psikoloji, işletme, uzman sistemler, yapay zekâ, sinyal işlenmesi, ulaştırma, kavşak sinyalizasyon onu gibi birçok alanda rahatlıkla kullanılabilir. (Zekai Ş. 2001)

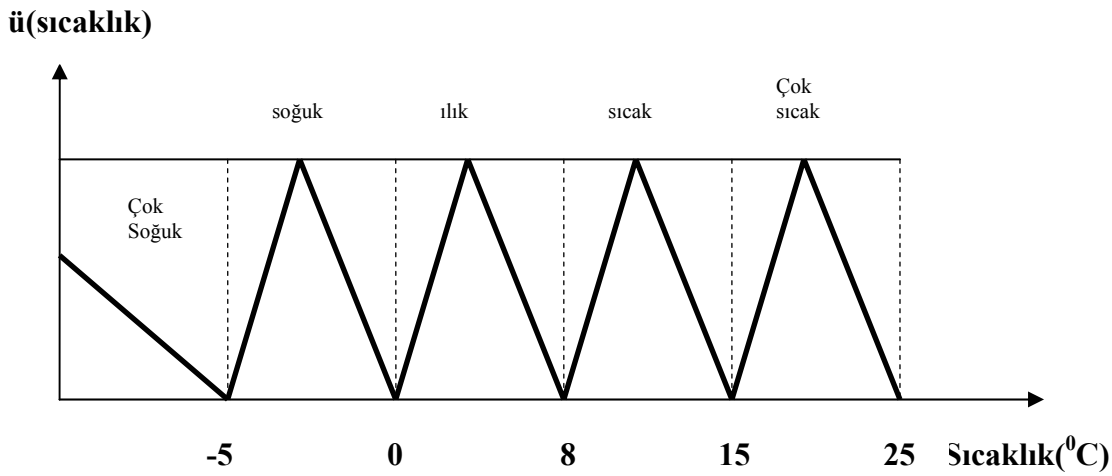
3.3. Üyelik Fonksiyonları

Araştırılmak istenilen ya da üzerinde çalışılan verilerin temsil ettiği sayısal aralık, konu hakkında bilgi sahibi olan uzman kişiler tarafından belirlenebilir. Örneğin, Ankara'da sıcaklık derecesinin değişim aralığının -5 ^{0}c 'den + 35'^{0}C olduğu bilinsin. İşte bu aralık sıcaklık kümesinin Ankara için öğelerinin bulunabileceği aralığı belirtir. Böylece tüm sıcaklık uzayı belirlenmiştir. Ancak, günlük konuşmalarda bu sıcaklık uzayının da bir takım alt aralıklardan oluştuğu düşünülür. Mesela, 'çok soğuk', 'soğuk', 'ılık', 'sıcak', 'aşırı sıcak' gibi... Burada, önce her bir alt terimin aralığının ne olduğuna karar veriniz gibi bir emirle karşılaşılırsa, belki mühendis olanlar bu alt kümelerin her birinin üst üste örtüşmeyen, ancak birbirinin sınırında devamlarıymış gibi olduklarını söyleyebilir. Mesela çok soğuğun - 5 ^{0}c ile O ^{0}c, soğuğun O ^{0}c ile +8 ^{0}c, ılığın + 8 ^{0}c ile + 15 ^{0}c, sıcağın + 15 ^{0}c ile + 25 ^{0}c, çok sıcağın ise +25 ^{0}c' den başladığı söylenebilir. Burada dikkat edilirse aralık tahminlerinde bulunulmuş ve her bir alt aralıktan biri bitince diğeri başlamıştır (Şekil 3.6).



Şekil 3.6 Bitişik Dikdörtgen Gösterimi

Bu aralıkların sınırlarında yine Aristo mantığına göre katı kararlar alınmalıdır. Örneğin, 7,9 °C'nin soğuk, 8.1 °C'nin ise ılık olduğuna karar verilir. Bu şekilde gösterim bakımından önemli bir nokta, her alt aralığa düşen sıcaklık değerinin üyelik derecesinin, sadece o aralıkta 1 'e, diğer aralıklarda ise 0'a eşit olduğudur. Bu nedenle, her sıcaklık alt kelimesinin üyelik fonksiyonu yüksekliği 1' e eşit olan bir dikdörtgen şeklindedir.

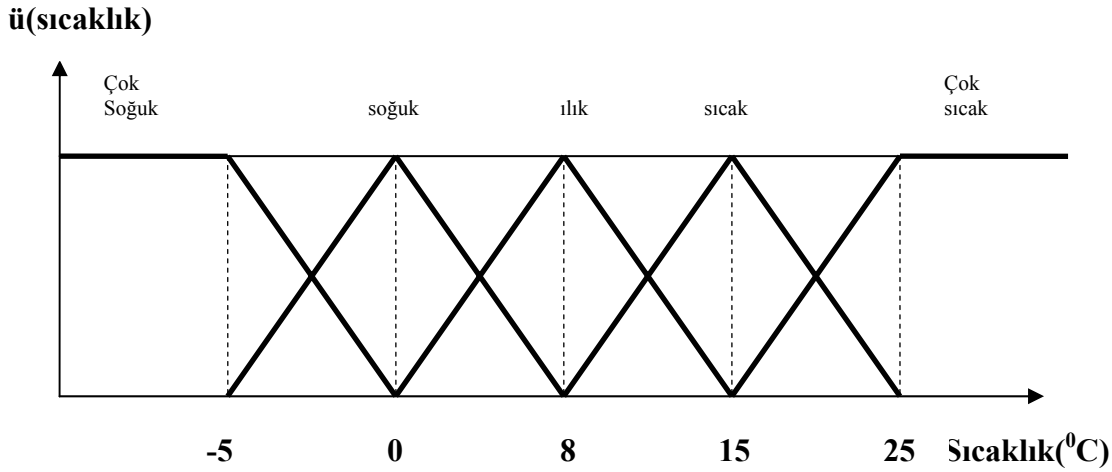


Şekil 3.7 Bitişik Üçgen Gösterimi

Şekil 3.7'de yukarıdaki tartışmanın bir doğal sonucu olarak en basit üçgen üyelik fonksiyonları bitişik olarak alınmıştır. Bu üçgenlerin de sıcaklık alt kümelerini tam yansıtmadığı açıktır. Çünkü burada da sınırlardaki sıcaklık değerlerinin üyelik dereceleri sıfır olarak düşünülmüştür. Ayrıca, bu sınır değerleri, ne alttaki ne de üstteki sıcaklık alt kümelerine dâhildir. Böylece, sınır değerler için tam anlamı ile bir belirsizlik vardır. Diğer taraftan, bu şekildeki alt aralıklar halen Aristo mantığına göre işlem görür. Çünkü bir alt aralığa düşen sıcaklık değeri, sadece o alt aralığa aittir. Fakat Şekil 3.6'dan farklı olarak üyelik derecesi 1'e eşit değildir.

Biraz daha makul düşünen birisi, bu aralıkların arasındaki geçiş kısımlarının böyle birbirinin devamı olmayacağını ve bir örtüşmenin söz konusu olabileceğini söylerse, daha mantıklı, günlük hayatta geçerli ve uzlaştırıcı çözümlere gitmiş olur. Çünkü

herkesin ılık sınırlarının +5 ile +15 °C'de sıfır üyelik derecelerine sahip olacağını kabul etmesini savunmak mümkün değildir. Halbuki, günlük hayatta sınıra yakın olan değerlerin hangi aralığa düşeceği oldukça müphem ve şüpheli, yani bulanıktır. Böylece, sıcaklık alt aralıklarının birbiri ile örtüşmeli geçişlere sahip olmasının gerekliliği ile sonuçta Şekil 3.8'de verilen üyelik fonksiyonları karşımıza çıkar.

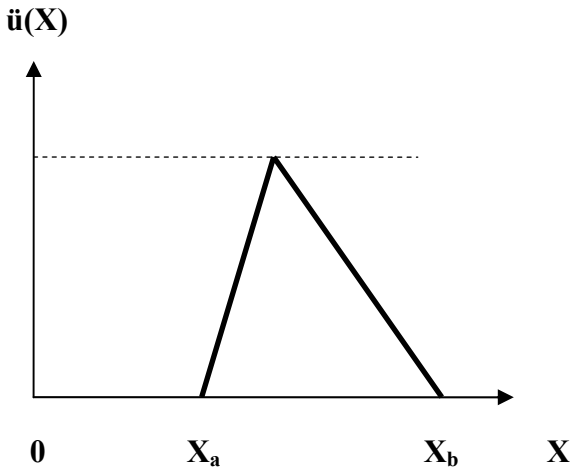


Şekil 3.8 Örtüşmeli Üçgen Gösterimi

Yukarıda söylenenlerden sonra ilk ve son alt aralıktaki sıcaklık durumlarının 'çok çok soğuğa' veya 'çok çok sıcağa' doğru giderken başka alt aralıklar olmadığından, üyelik derecelerinin 1'e eşit kalmasının makul olacağı anlaşılır. Bunun doğal bir sonucu olarak da, ilk ve son üyelik fonksiyonlarının üçgen değil de yamuk şeklinde olacağı sonucuna varılır. Böylece, her alt aralığa girişimli olarak bir üyelik fonksiyonu şekli tayin edilmiştir.

Diğer taraftan, sorun her alt aralığa, örneğin 'ılık' aralığına düşen sıcaklık derecelerinin hepsinin aynı önemde olup olmayacağıdır. Tabii olarak, ılık aralığının alt ve üst uçlarına yaklaştıkça onun komşusu olan altta soğuk, üstte ise sıcak alt kümelerine doğru geçişler beklendiği için, o geçiş bölgelerine rastlayan kısımların tam anlamı ile ılık vasfına sahip olacağı söylenemez. Böylece, her bir alt aralığa düşen sıcaklık derecelerinin, o alt aralığın uçlarına yakın kısımlarında önemlerini

ortaya kıyasla göreceli olarak kaybedeceği sonucuna, buradan da eğer bir alt aralıkta önem derecesi diye bir değer düşünülecek olursa bunun en büyük değerlerinin o alt aralığın ortalarında, en düşük değerlerinin ise uçlarda olacağını söyleyebiliriz. Bu düşünceler bizi Şekil 3.9'da gösterilen bir geometrik gösterime sürükler. Genel olarak, her alt aralığın ayrık üyelik fonksiyonu bu şekilde gösterildiği gibi olur. Bu fonksiyonların simetrik olması gerekmez. Böylece X_a ve X_b gibi alt ve üst sınırlara sahip X değişkeninin bu aralıktaki her değerine ayrı bir üyelik derecesi, ü(x), tayin edilmiş olur. Bu aralıkta k i tüm X değerleri, o X değişkeninin bir alt kümesini teşkil eder. (Zekai Ş. 2001)

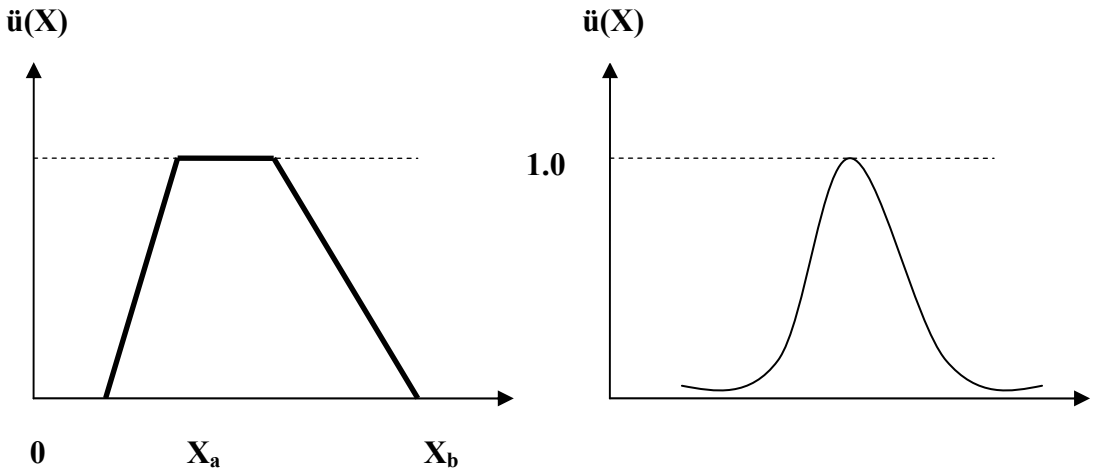


Şekil 3.9 Bulanık Küme

Genel olarak, küme üyelerinin değerleri ile değişiklik gösteren böyle bir eğriye üyelik fonksiyonu (önem eğrisi) adı verilir. Bunun en önemli özellikleri, alt küme sınırlarındaki değerlerinin orta öğelerinkine göre daha düşük olmasıdır. Ancak klasik kümelere bir benzerlik teşkil etmesi açısından en büyük önem derecesine sahip olan ortaya yakın öğelere 1 değeri verilirse, diğerlerinin O ile 1 arasında ondalıklı ve sürekli değiştiği sonucuna varılır. İşte bu şekilde, O ile 1 arasındaki değişimin, her bir öğe için değerine, üyelik derecesi, bunun bir alt küme içindeki değişimine ise,

üyelik fonksiyonu adı verilir. Böylece, üyelik fonksiyonu şemsiyesi altında toplanan öğeler önem derecelerine göre birer üyelik derecesine sahiptir. (Zekai Ş. 2001)

Bu kısma son vermeden, matematik kurallarına uygun olarak düzgün şekilli üyelik fonksiyonlarının Şekil 3.9'da gösterilen üçgenden başka, yamuk veya çan eğrisi şeklinde olacağı aklımıza gelebilir (Şekil 3.10). Pratik uygulamalarda bunlardan en fazla üçgen, ondan sonra da yamuk olanı kullanılır. (Zekai Ş. 2001)



Şekil 3.10 Yamuk ve Çan Eğrisi üyelik fonksiyonları

3.3.1. Üyelik Fonksiyonunun Kısımları

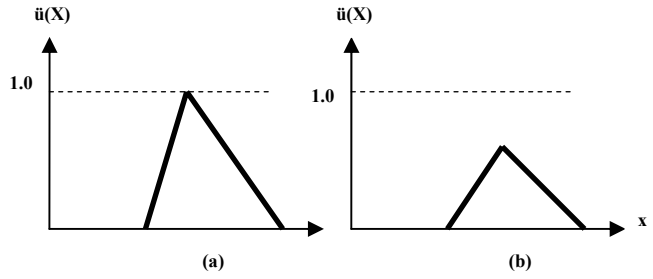
Yukarıda yapılan açıklamalardan, genel olarak bir üyelik fonksiyonunda bulunması gereken kısımlar hakkında fikirler oluşmuştur. Bunların, daha bilimsel terminoloji olarak açıklaması aşağıda yapılacaktır. En genel hali ile, yamuk şeklindeki bir üyelik fonksiyonu, Şekil 3.6'da gösterildiği gibi, değişik kısımlara ayrılabilir. (Zekai Ş. 2001)

Görüldüğü gibi verilen bir bulanık alt kümede bir değil, birden fazla öğenin üyelik derecesi 1'e eşit alınabilir. Bu durumda, i üyelik dereceli öğelerin tam anlamı

ile, hiçbir şüpheye düşmeksizin, sadece o alt kümeye ait olduğu sonucuna varılır. Böyle üyelik derecesine sahip olan öğeler alt kümenin orta kısmında toplanmıştır. İşte üyelik dereceleri 1'e eşit olan öğelerin toplandığı alt küme kısmına, o alt kümenin özü *(core)* denir. Burada ü(x) = 1'dir. Üçgen şeklindeki üyelik fonksiyonunda bir tane öğenin üyelik derecesi 1'e eşit olduğundan, üçgen üyelik fonksiyonlarının özü bir nokta olarak karşımıza çıkar. (Zekai Ş. 2001)

Bunun aksine bir alt kümenin tüm öğelerini içeren aralığa o alt kümenin dayanağı *(support)* adı verilir. Dayanakta bulunan her öğenin az veya çok değerde (O ile 1 arasında) üyelik dereceleri vardır. Bunun matematik gösterilişi ü(x) > O şeklindedir. Aslında bu öğeler topluluğu önceki kısımda belirtilen aralığa karşı gelir. Üyelik dereceleri 1'e veya 0'a eşit olmayan öğelerin oluşturduğu kısımlara üyelik fonksiyonunun sınırları *(boundary)* veya geçiş bölgeleri denir. Bunun matematik tanımı O < ü(x) < 1 şeklindedir. Bunlar alt kümenin kısmi öğeleridir. Aslında bir alt kümeye bulanıklık özelliğinin takılması bu geçiş yerlerinin bulunması sonucundadır. Genel olarak, tüm üyelik fonksiyonlarında biri sağda diğeri de solda olmak üzere iki tane geçiş bölgesi vardır. Şekil 3.8'de en sol ve en sağdaki bulanık kümelerde birer tane geçiş bölgesi vardır. (Zekai Ş. 2001)

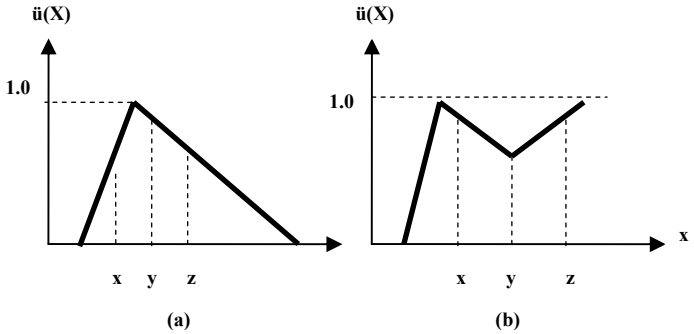
Yukarıda şekil olarak açıklanan bu üç özelliğe ilave olarak üyelik fonksiyonunun sahip olması gerekli olan iki tane daha özellik bulunmaktadır. Bunlardan birincisi, bulanık kümenin normal olduğunu tespit etmemize yarayan bir kavramdır. Buna göre normal bulanık kümede, en azından bir tane üyelik derecesi 1'e eşit olan öğe bulunmalıdır. Şekil 3.11 normal ve normal olmayan bulanık kümeleri göstermektedir. (Zekai Ş. 2001)



Şekil 3.11 Bulanık kümeler, (a) normal, (b) normal olmayan

İkinci özellik ise bulanık kümenin dış bükey (konveks) olmasıdır. Dış bükey olan bulanık kümelerde üyelik fonksiyonu kümenin dayanağı üzerinde, ya sürekli

artar veya sürekli azalır veya üçgen üyelik fonksiyonunda olduğu gibi önce sürekli olarak üyelik derecesi bir öğede I' e eşit oluncaya kadar artar ondan sonraki dayanağa düşen öğeler için sürekli azalır. Bunun aksi durumlar da söz konusudur. Ancak, onlar bulanık kümelere üyelik fonksiyonu olamaz. Şekil 3.12'de dış bükey olan ve olmayan bulanık alt kümelere bazı misaller gösterilmektedir. (Zekai Ş. 2001)

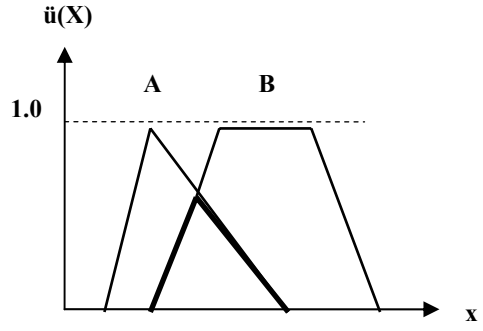


Şekil 3.12 Bulanık kümeler, (a) dış bükey, (b) dış bükey olmayan

Dış bükeyliğin matematik olarak tanımlanmasında, aynı bulanık alt kümeye düşen x, y ve z gibi üç tane öğe düşünülürse ve bunlar arasında değerce büyüklük olarak x < y < z gibi bir sıra bulunuyor ise, bunlardan ortadakinin üyelik fonksiyonu önceki ve sonrakine göre

ü(y) >= EK [ü(x), ü(z)]

şeklindedir ve bu bağıntı daima geçerli olmalıdır. Burada EK en küçükleme (minimizasyonu) işlemi demektir. Yani y'nin üyelik derecesi, x ve z'nin üyelik derecelerinin en küçüğünden daha büyüktür. İşte bu durumda o kümeye dış bükey bulanık küme adı verilir. (Zekai Ş. 2001)



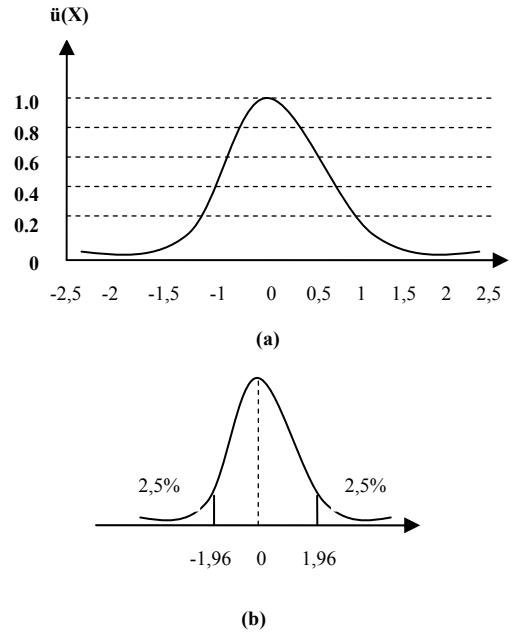
Şekil 3.13 Dış bükey bulanık kümelerin kesişimi

Bulanık kümelerin üyelik fonksiyonlarında üyelik derecesinin 0,5'e eşit olması durumundaki noktaya geçiş noktası *(cross-over)* adı verilir. Yani geçiş noktasında, ü(x) = 0,5'dir. Ayrıca, bulanık kümenin yüksekliği denilen bir büyüklük, üyelik derecesinin en büyük olduğu öğelere karşı gelir. Yukarıda söylenenlerden sonra, normal bulanık kümelerde yüksekliğin I' e eşit olması gerekliliği anlaşılır. Diğer bir ifade ile yüksekliği 1'e eşit olmayan bulanık kümeler, normal olmadıklarından herhangi bir bulanık küme, mantık ve sistem çalışmasında kullanılamaz. Normal olmayan bulanık kümeleri normal hale dönüştürmek için, kümenin her bir üyelik derecesinin, en büyük üyelik derecesine bölünmesi gereklidir. Böylece normal olmayan bulanık kümelerin dış bükey olmaları şartı ile nasıl normal bulanık kümeler haline dönüştürüleceği anlaşılmış olur. (Zekai Ş. 2001)

Temel bulanık kümeler normal ve dış bükey olmasına karşılık, küme işleminin yapılması sonucunda elde edilen kümeler, bulanık normal küme çıkmayabilir. Örneğin, iki normal ve dışbükey bulanık alt kümenin birleşimi normal ve dış bükey olmayan bulanık küme çıkabilir (Şekil 3.13).

Üyelik fonksiyonları simetrik olabilir de olmayabilir de. Genel olarak, bir boyutlu uzayda tanımlanan bulanık kümelerin iki veya daha fazla boyutta, az da olsa tanımlanması mümkündür. Bir boyutlu uzayda çizgi şeklinde olan üyelik fonksiyonları iki boyut bu uzayda yüzey şeklinde görülür.

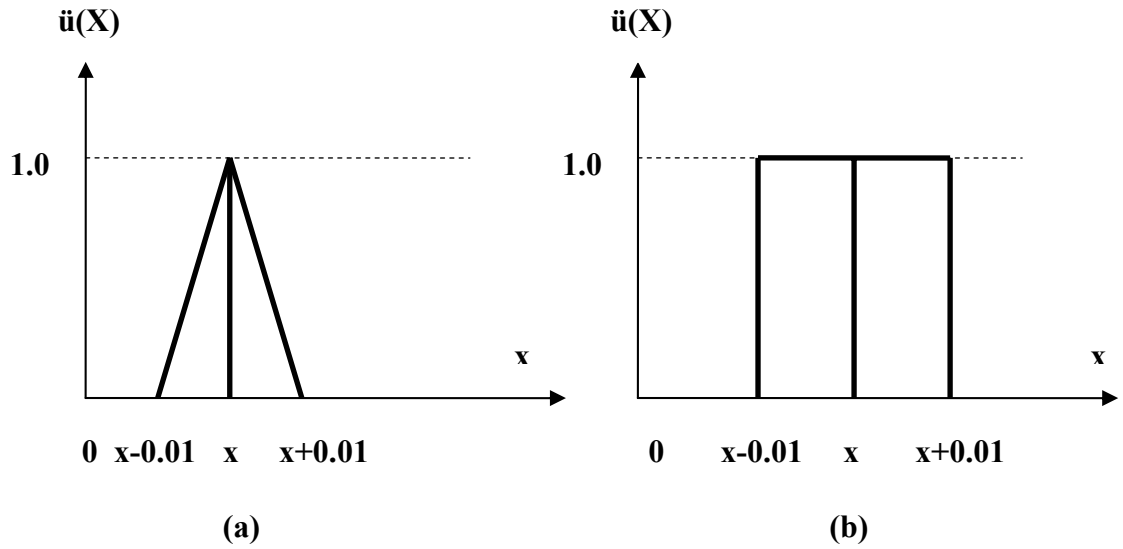
Yukarıda söylenenlerden ihtimaller teorisi veya istatistikteki dağılım fonksiyonları hakkında bilgisi olanlar, üyelik fonksiyonunun sanki dağılım fonksiyonlarına benzediği sonucunu çıkarabilir. Dağılım fonksiyonlarında tepe noktasının 1'e eşit olması söz konusu değildir. Ancak histogram olarak dağılım fonksiyonunun altındaki alanın 1'e eşit olması gereklidir. Şekil 3.14'te Gauss eğrisi şeklinde dağılım ve üyelik fonksiyonları ayrı ayrı gösterilmiştir. Bunlardan bulanık kümeyi temsil eden Gauss eğrisinde üyelik fonksiyonunun tepe noktasının 1'e eşit olduğuna dikkat ediniz. Bu şekilde f(x), x rasgele değişkeninin ihtimal yoğunluk fonksiyonunu gösterir. (Zekai Ş. 2001)



Şekil 3.14 Gauss (a) bulanık kümesi (b) dağılım fonksiyonu

3.3.2. Bulanıklaştırma

Genel olarak, klasik küme şeklinde beliren değişim aralıklarının bulanıklaştırılması, bulanık küme, mantık ve sistem işlemleri için gereklidir. Bunun için, bir aralıkta bulunabilecek öğelerin hepsinin, 1' e eşit üyelik derecesine sahip olacak yerde, 0 ile 1 arasında değişik değerlere sahip olması düşünülür. Bazı öğelerin belirsizlik içerdikleri kabul edilir. Bu belirsizliğin sayısal olmayan durumlardan kaynaklanması halinde bulanıklıktan söz edilir. Özellikle, bazı cihazların prezisyonu diye tabir edilen durumlarda mesela +- %1'lik hassaslık (prezisyon), ölçülen x büyüklüğünün x + 0.01 ve x - 0.01 arasında değişeceği beklentisini ifade eder. Bunun klasik ve bulanık kümelerde gösterilişi Şekil 3.15'teki gibidir.



Şekil 3.15 Hassaslık (a) bulanık (b) klasik

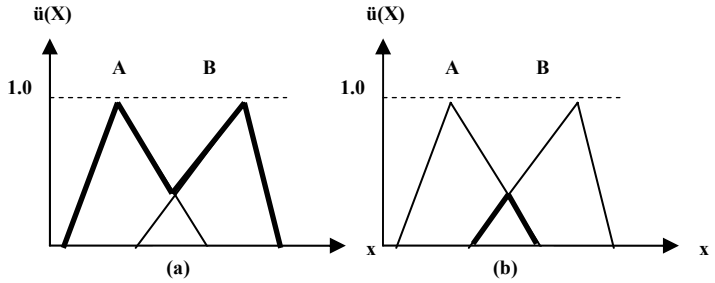
Buradan, bulanık presizyonun pratikte mantıki olarak daha sağlıklı bir tanım olduğu ortaya çıkar. Buna göre, prezisyon kelimesinden ve değerinden bulanık üyelik fonksiyonunun üçgen şeklinde olması akla ilk gelen durumdur.

3.3.3. Durulaştırma

Bulanık kümelerin üyelik fonksiyonlarının nasıl bulunacağı hakkında gerekli bilgi ve yöntemlerden bahsedildi. Pratik uygulamalarda, özellikle cihaz ve mühendislik plan, proje ve tasarımlarında boyutlandırmalar için kesin sayısal değerlere gerek duyulmaktadır. İşte bu durumlara bulanık olarak elde edilmiş veya verilmiş bilgilerden yararlanarak gerekli cevapların verilmesi için bulanık olan bilgilerin durulaştırılması gerekmektedir. İnsanlar için yapay zekâ çalışmalarında bulanık değişken, küme, mantık ve sistemler öneme sahip olmasına mukabil, bunların bulanık olabilecek çıkarımlarının kesin sayılar haline dönüştürülmesi gerekir. İşte bulanık olan bilgilerin kesin sonuçlar haline dönüştürülmesi için yapılan işlemlerin tümüne birden durulaştırma (defuzzification) işlemleri adı verilir.

3.3.3.1. Durulaştırma İşlemleri

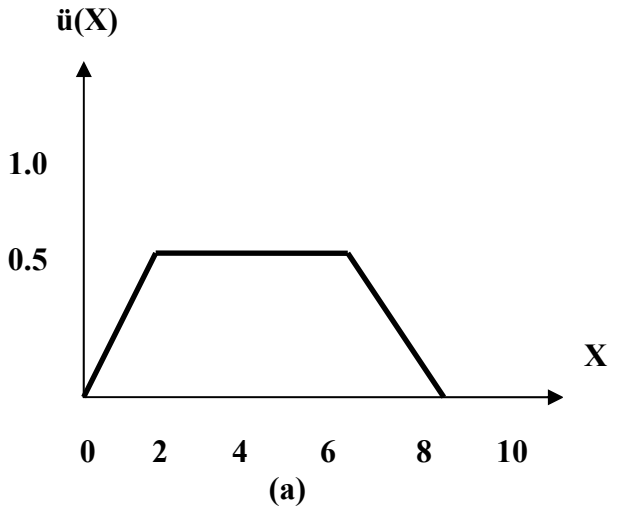
Daha önce de belirtildiği gibi, bir bulanık küme işlemi sonucundaki bulanık kümenin tek sayı haline dönüştürülmesi gerekebilir. Bu, bulanıklaştırma işleminin aksi olan durulaştırma işlemi ile yapılır. Yapılan işlemler sonrasında bulanık sonuçlardan bir tanesi Şekil 3.16a'daki gibi yamuk, diğerinin ise Şekil 3.16b'deki gibi üçgen şeklinde olduğunu düşünelim. Bunların ikisinin birleşimi ile yapılan son işlem sonrası bulanık çıkarım kümeleri elde edilir. Şimdi bu son dış bükey olmayan bulanık kümeden tek sayılı bir tasarım büyüklüğünün çıkartılması düşünülsün. İşte bunun için durulaştırma işleminin yapılması gerekecektir.

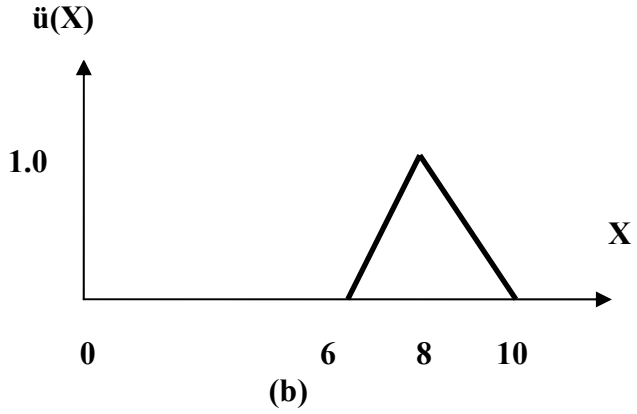


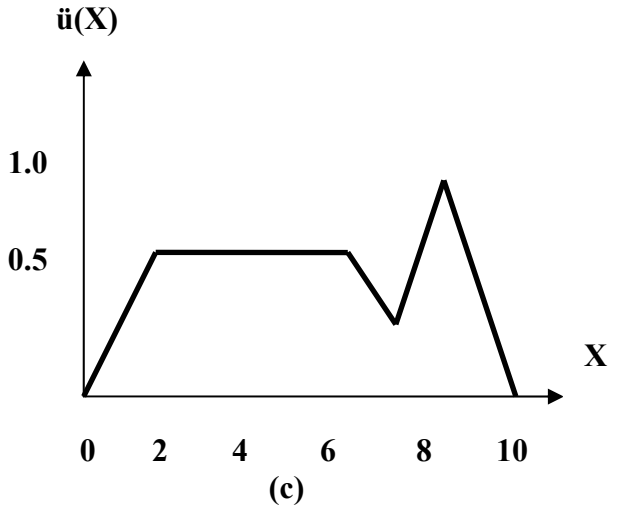
Şekil 3.16 İki bulanık kümenin (a) birleşimi, (b) kesişimi

Tabii olarak Şekil 3.16'da, iki tane bulanık kümenin birleşimi sonucunda elde edilen bulanık çıkarım gösterilmiştir. Halbuki, değişik şekilleri olan çıkarımların iki veya daha fazla sayıdaki temel bulanık kümelerden çıkması mümkündür.

Aşağıda yedi tane durulaştırma işleminin esasları verilecektir. Bunların hangisinin kullanılacağına, elindeki sorunun türüne göre araştırma veya tasarımı yapan mühendisin karar vermesi gereklidir. Aşağıdaki çıkarım bulanık kümesinin Z, öğelerinin z ve durulaştırılmış değerinin ise z* ile gösterildiklerine dikkat edilmelidir.

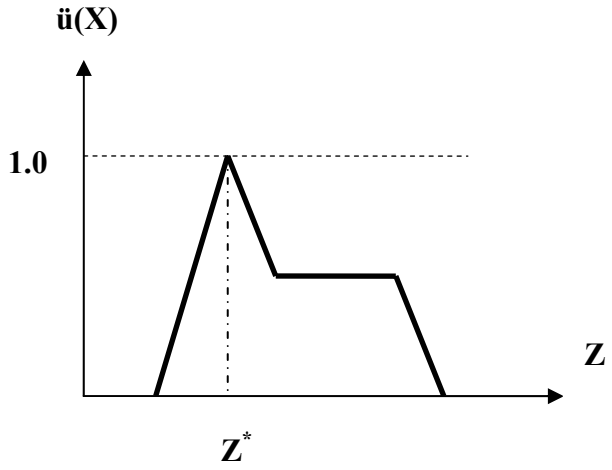






Şekil 3.17 Tipik bulanık küme çıktısı, (a) bulanık girdi ilk kısım, (b) bulanık girdi ikinci kısım, (c) ikisinin birleşimi

1. En büyük üyelik ilkesi: Bunun diğer bir adı da yükseklik yöntemidir. Kullanılabilmesi için tepeleri olan çıkanın bulanık kümelerine gerek vardır. Şekil 3.18'de gösterilen bu durulaştırma işleminin aritmetik notasyon şeklinde gösterimi; (Zekai Ş. 2001)

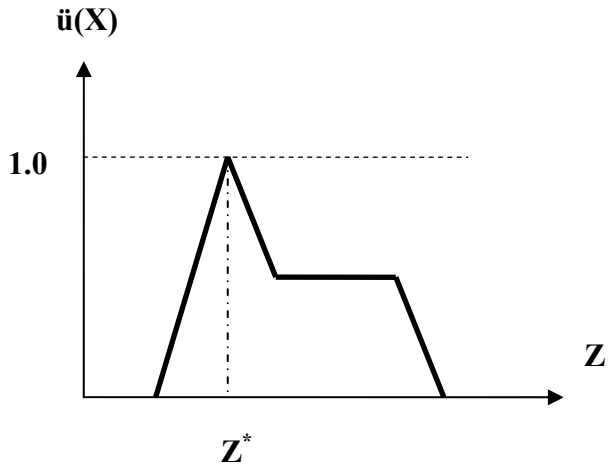


Şekil 3.18 En büyük üyelik derecesi durulaştırması

2. Sentroid yöntemi: Bunun diğer bir adı da ağırlık merkezi yöntemidir. Durulaştırma işlemlerinde, belki de en yaygın olarak kullanılan işlem budur. Şekil 3.19'da gösterilmiş olan bu durulaştırmanın matematik işlemi aşağıdaki denklem vasıtası ile yapılır. (Zekai Ş. 2001)

$$z^* = \frac{\int \ddot{u}_{Ç}(z).zdz}{\int \ddot{u}_{Ç}(z)dz}$$

Burada J integral işaretini gösterir.



Şekil 3.19 Sentroid yöntemi ile durulaştırma

3. Ağırlıklı ortalama yöntemi: Bunun kullanılabilmesi için simetrik üyelik fonksiyonunun bulunması gereklidir. İşlemler matematik olarak

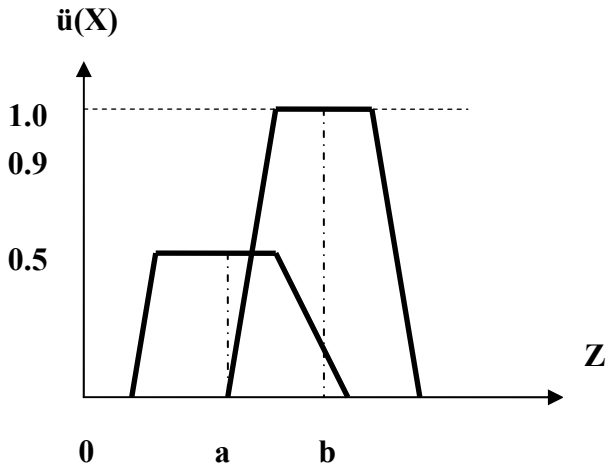
$$z^* = \frac{\sum \ddot{u}_{\text{Ç}}(\bar{z}).\bar{z}}{\sum \ddot{u}_{\text{Ç}}(\bar{z})}$$

şeklinde yapılır. Burada *L* işareti cebir anlamında toplamayı gösterir. Bu durulaştırma işlemi Şekil 3.20'de gösterilmiştir. Böylece çıkışı oluşturan bulanık kümelerin üyelik fonksiyonlarının her biri sahip oldukları en büyük üyelik derecesi değeri ile çarpılarak ağırlıklı ortalamaları alınır.

Misal olarak Şekil 3.20'deki iki bulanık kümenin ağırlıklı ortalaması (durulaştırılmış değer).

$$Z^* = \frac{a(0.6) + b(0.9)}{0.6 + 0.9}$$

olarak bulunur. Bu durulaştırma işlemi sadece simetrik olan üyelik fonksiyonları için geçerli olduğundan, a ve b değerleri temsil ettikleri şekillerin ortalamalarıdır. (Zekai Ş. 2001)

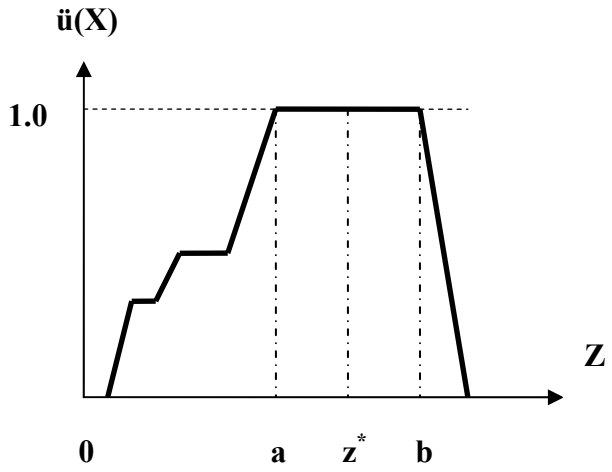


Şekil 3.20 Ağırlıklı ortalama yöntemi durulaştırması

4. Ortalama en büyük üyelik: Bu yöntem aynı zamanda en büyüklerin ortası diye de bilinir. Bu bakımdan birinci durulaştırma ilkesine çok yakındır. Ancak, en büyük üyeliğin konumu tekil olmayabilir. Bunun anlamı üyelik fonksiyonunda en büyük üyelik derecesine sahip olan, ÜA(Z) = 1, bir nokta yerine plato gibi düzlük kısmı da bulunabilir. Şekil 3.21'de durulaştırma işlemi gösterilmiş olan bu yönteme göre durulaştırılmış değer

$$Z^* = \frac{a+b}{2}$$

olarak bulunur. Buradaki a ve b değerleri şekilde gösterilmiştir. (Zekai Ş. 2001)

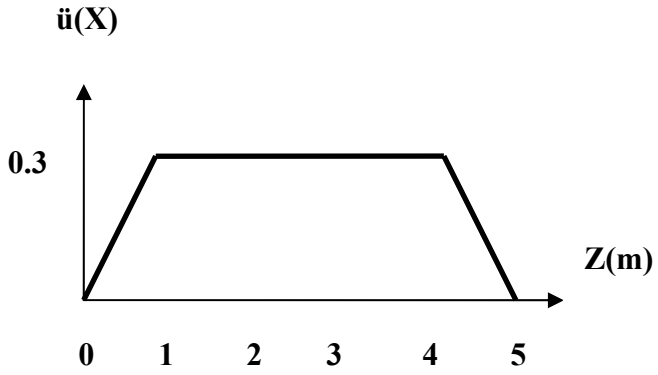


Şekil 3.21 Ortalama en büyük üyelik durulaştırılması

5. Toplamların merkezi: Kullanılan durulaştırma işlemleri arasında en hızlı olanı bu yöntemdir. Bu yöntemde iki bulanık kümenin birleşimi yerine onların cebirsel toplamları kullanılır. Bunun bir mahzuru örtüşen kısımların iki defa toplama girmesidir. Durulaştırılmış değer

$$Z^* = \frac{\int_Z Z \sum_{k=1}^{n} \ddot{u}_ç(z)dz}{\int_z \sum_{k=1}^{n} \ddot{u}_ç(z)dz}$$

olarak hesap edilebilir. Bir bakıma, bu hesaplama tarzı, ağırlıklı ortalama durulaştırmasına benzer Ancak toplamların merkezi yönteminde ağırlıklar ilgili üyelik fonksiyonlarının alanlarıdır. Ortalama ağırlıklar yönteminde ise bu üyelik derecesidir. Toplamların merkezi ile durulaştırma işlemleri Şekil 3.22'de gösterilmiştir. (Zekai Ş. 2001)



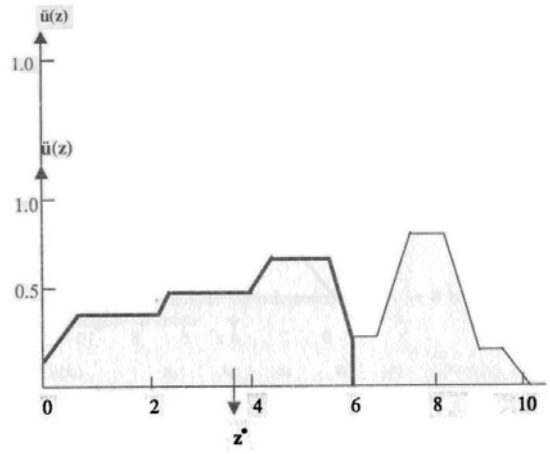
Şekil 3.22 Toplamların merkezi durulaştırması.

6. En büyük alanın merkezi: Eğer çıkış bulanık kümesi en azından iki tane dış bükey alt bulanık kümeyi içeriyor ise, dış bükey bulanık kümelerin en büyük alanın ağırlık merkezi durulaştırma işleminde kullanılır. Şekil 3.23'de gösterilen durulaştırma işleminin matematik hesaplaması

$$z^* = \frac{\int \ddot{u}_{ebÇ}(z) z dz}{\int \ddot{u}_{ebÇ}(z) dz}$$

eşitliğine göre yapılır. Burada $\ddot{u}_{ebç}(Z)$ en büyük alanlı dış bükey bulanık kümenin hakim olduğu alt bölgeyi gösterir.

Bu şart tüm çıkarım bulanık kümesinin dış bükey olmadığı zaman kul1anıhr, ama tüm çıkarımın dış bükey olması durumunda z. sentroid yöntemi ile elde edilenin aynısıdır. (Zekai Ş. 2001)

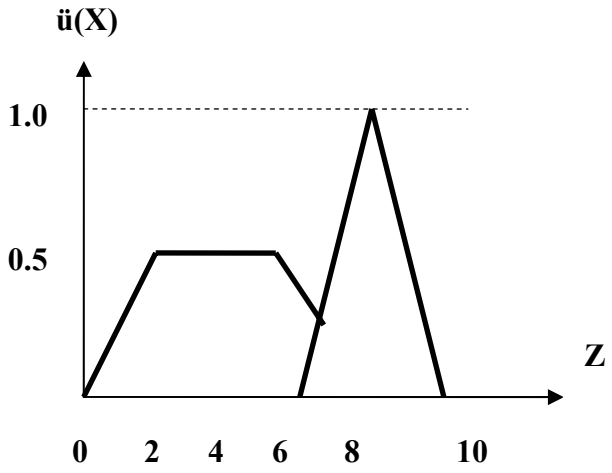


Şekil 3.23 En büyük alan merkezi ile durulaştırma

7. En büyük ilk veya son üyelik derecesi: Bu yöntem de, tüm çıktıların birleşimi olarak ortaya çıkan bulanık küme de en büyük üyelik derecesine sahip olan en küçük (veya en büyük) bulanık küme değerini seçmek esasına dayanır. Hesaplamaların vereceği z. için aşağıdaki denklemler geçerlidir. Önce bulanık küme çıkarımı, B, birleşiminde en büyük yükseklik, Y_{eb} tespit edilir.

$Y_{eb}(B) = EB\ [Ü_B(Z)]$

Bundan sonra birinci en büyük değer, z., bulunur. Bu yöntemin bir diğer seçeneği ise ilk yerine son en büyük bulanık küme değerinin, z., bulunmasıdır. Bu durumlar Şekil 3.24'te gösterilmiştir. (Zekai Ş. 2001)



Şekil 3.24 İlk ve son en büyük üyelik dereceleri ile durulaştırma

4.BİRLİKTELİK KURAL MEDENCİLİĞİ

4.1. Geniş Veritabanlarından Birliktelik Kuralları Çıkarılması İşlemi

Birleşik veri madenciliği yöntemi geniş veri kümeleri arasında ilginç birleştirme ve çevreleme ilişkileri bulur. Sürekli biriktirileni depo edilen veriler sayesinde devleşen veri tabanları, birçok endüstrinin dikkatini veri madenciliği birleştirme yöntemine çekmiştir. İşyerinin dev işlem kayıtları arasında birleşik ilişki ile ilgilenmek, pek çok iş adamına uygulamalar konusunda (katalog dizaynı, kar-zarar analizi.. vb) yardımcı olmaktadır.(Erkan Ü. ve ark. 2005)

Birleştirme yöntemine tipik bir örnek olarak market sepeti analizini verebiliriz. Bu işlem müşterinin sepetindeki farklı ürünler arasında ilişki kurarak müşteri alışkanlıklarını analiz eder. Bu tür ilişkiler satıcının hangi ürünlerin daha çok tercih edildiğini, ürünün hangi ürünle birlikte satıldığını göz önünde bulundurarak yeni satış stratejisi oluşturmasına yardımcı oluyor. Örneğin süt alan müşterilerin ekmek alma olasılığı nedir ve hangi tip ekmekleri almaktadır sorularına bu yöntemle cevap verebiliriz. Bu tür bilgiler satıcılara seçici bir market oluşturma ve raf düzenleme gibi konularda yardımcı olarak satışların artmasına sebep olabilir. Örnek olarak süt ve ekmek reyonlarını birbirine yakın bir şekilde yerleştirerek kısa zamanlı alışverişlerde müşteri sütle beraber ekmek almaya da teşvik edilebilir.(Erkan Ü. ve ark. 2005)

4.2. Birliktelik Kural Madenciliği Nedir?

Büyük bir alışveriş merkezinde elektronik – bilgisayar bölümünün yöneticisi olduğunuzu düşünün. Müşterilerinizin alışkanlıkları hakkında daha çok bilgiye sahip olmak istiyorsunuz. Ve özellikle “Müşteriler hangi grup ürünleri bir arada satın

olmayı tercih ediyor?" sorusuna cevap arıyorsunuz. Sorunuza cevap bulmak için kayıtlarınızdaki veriler üzerinde bir market sepeti analizi yapabilirsiniz. Elde edeceğiniz sonuçları reklam stratejinizde, katalog dizaynı gibi işlerinizde kullanabilirsiniz. Ayrıca bu analiz satıcılara değişik mağaza dizaynlarında yardımcı olabilir. (Erkan Ü. ve ark. 2005)

Markette bulunabilecek tüm ürünlerin kümesini evren olarak düşünecek olursak, her ürünün varlığını veya yokluğunu gösteren boolean bir değişkeni olacaktır. Böylece her bir sepeti bu boolean değerlerden oluşan bir vektör olarak tasvir edebiliriz. Bu vektörlerden alınan numuneler hangi ürünlerin beraber satıldığını ortaya koyabilir. Bu numuneler ilişkisel kurallar formunda tasvir edilebilir. Örneğin bilgisayar alırken finansal yönetim yazılımı da almaya eğilimli olan müşteriler bilgisi aşağıdaki ilişkisel kural(1) gibi tanımlanabilir. (Erkan Ü. ve ark. 2005)

Bilgisayar → finansal_yönetim_yazılımı [Destek = 2%, Güven = 60%] (1)

Destek(support) ve Güven(confidence) değerleri ilgililiği tanımlayan ölçülerdir. Bulunan kuralın kesinliğini ve kullanabilirliğini ifade ederler. Destek = 2% değeri analiz altındaki tüm işlemlerin (satış hareketleri) %2 sinde bilgisayar ve finansal yönetim yazılımının beraber satıldığını ifade eder. Güven=60% değeri ise bilgisayar alan müşterilerin %60 nın bu söz konusu yazılımı da aldığını ifade der. Genelde ilişkisel kurallar, eğer destek ve güven değerleri belirli bir seviyeyi (eşik) geçiyorsa göz önünde tutulur. Bu eşik değeri uzmanlar tarafından belirlenebilir.

4.3. Birliktelik Kuralı

Birliktelik kural madenciliği aşağıdaki gibi belirtilebilir[1,2]: $I=\{I_1,I_2,...,I_n\}$ bir öğeler kümesi olsun. D görev-uyumlu veri olan bir hareketler kümesi olsun, burada her bir T hareketi $T \subseteq I$ olan bir öğeler kümesidir. Satın alınan öğelerin miktarı dikkate alınmaz. Her bir hareket TID olarak bilinen bir tanımlayıcıya atanır. A kümesi bir öğeler kümesi olsun, bir T hareketi eğer ve sadece eğer $A \subseteq T$ ise A'yı

içerir demektir. Bir birliktelik kuralları $A \Rightarrow B$ gerektirme formundadır, burada $A \subset I$, $B \subset I$ ve $A \cap B = \varnothing$ dir. $A \Rightarrow B$ kuralı s desteği ile D hareket kümesi içinde tutulur, burada s desteği D içinde $A \cup B$'yi (örneğin A ve B nin her ikisinide) içeren hareketlerin yüzdelik oranıdır. Bu, olasılık olarak alınır, $P(A \cup B)$ ile gösterilir. D içinde A'yı içeren hareketler aynı zamanda B yide içeriyorlarsa eğer bu hareketlerin yüzdelik oranına c dersek $A \Rightarrow B$ kuralı; D hareket kümesi içinde c güvenine sahiptir. Bu, koşulsal olasılık olarak alınmaktadır, $P(B|A)$ ile gösterilir. Yani, Destek($A \Rightarrow B$)= $P(A \cup B)$=s, Güven($A \Rightarrow B$)= $P(B|A)$=Güven($A \Rightarrow B$)/Güven(A)=c dir. (Erkan Ü. ve ark. 2005)

Birliktelik kuralları madenciliği; kullanıcının belirttiği minimum destek eşiği ve minimum güven eşiğinin her ikisinide sağlayan güçlü birliktelik kurallarını çıkarmaktır. (Erkan Ü. ve ark. 2005)

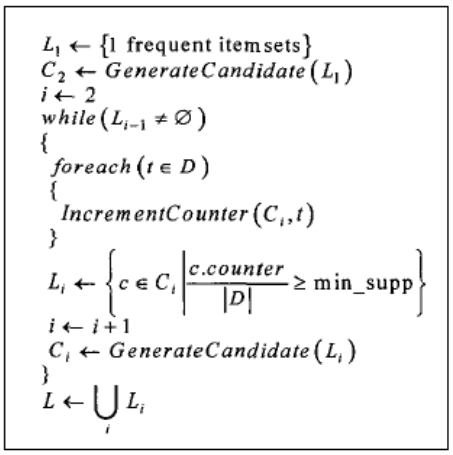
Birliktelik kural madenciliği iki-adımlı bir işlemdir[3]:

- Hareket veritabanındaki sıklık öğe kümelerinin tümünü bul. Eğer A öğekümesinin desteği minsup'dan büyük yada eşitse, support(A)≥minsup, A bir sıklık öğekümesidir. Aksi takdirde A bir sıklık öğe kümesi değildir.
- Sıklık öğe kümelerinden güçlü birliktelik kurallarını üret. Her bir A sıklık öğekümesi için eğer $B \subset A$, $B \neq \varnothing$ ve support(A)/support(B)≥miconf ise $B \Rightarrow (A-B)$ birliktelik kuralına sahibizdir.

4.4. Apriori Algoritmasi

Apriori algoritması [4,5] arama uzayını indirgemede aşağıdaki teoremi kullanır: Eğer bir öğe küme sıkça tekrarlanıyorsa kendisinin altkümelerinin hepsi de sık sık tekrarlanmaktadır. Bunun anlamı şudur; i öğekümesini kullanma potansiyel i+1 öğekümeyi üretmeyi mümkün kılmaktadır. i+1 aday öğekümenin her bir altkümesi tekrarlanmalıdır. Burada veritabanının tekrarlı taranması yapılarak tüm tekrarlı öğekümeleri bulmak mümkündür. i'inci veritabanı taraması süresince Apriori algoritması i öğekümenin meydana çıkma sayısını sayar ve i geçişin sonunda i+1 öğe

içeren adayları üretir. Şekil 4.1 Apriori algoritmasının yalancı kodunu göstermektedir. (Erkan Ü. ve ark. 2005)



Şekil 4.1 The pseudo code of Apriori algorithm

Apriori algoritması seviye-akıllı bir algoritmadır. Bu nedenle veritabanına erişim bir çok defa ve en uzun sıklık öğekümesinin uzunluğu kadar olmaktadır.

Veriseti çok büyükse çoklu veri tabanı taraması Apriori algoritmasının yürütülmesini çok uzun yapar. Bu nedenle apriori algoritmasının hızlandırılması için çeşitli algoritmalar geliştirilmiştir. Bu iyileştirilen algoritmalar farklı şekillerde Giriş/Çıkış maliyetini azaltmaktadır. For example[4]; Apriori-TID algorithm, DIC-Dynamic Itemset Counting algorithm, DHP-Direct Hashing and Pruning algorithm, Frequent Pattern Growth algorithm etc. (Erkan Ü. ve ark. 2005)

5. BULANIK VERİ MADENCİLİĞİ UYGULAMASI

5.1. Basın-Yayın Alanında Birliktelik Kural Madenciliği Sistemi

Çalışmada, İnternet üzerinden ulaşılabilen basın-yayın kaynaklarında yer alan görsel ve metinsel verilerin hızlı ve etkin bir şekilde erişimi ve bu kaynaklardan anlamlı ve önemli bilgilerin çıkarılması hedeflenmektedir. Çalışmalar öncelikli olarak istihbarat açısından önem taşıyan kişi ve örgütlerle ilgili haberler üzerinde yoğunlaşmıştır. Sunucu bilgisayarda İnternet üzerinde yer alan haber kaynaklarından toplanmış ve işlenmiş metinsel belgelerden oluşan veri-tabanı ile bu bilgileri işleyen uygulama yazılımları bulunmaktadır. Bir arayüz ile kullanıcının bu bilgileri sorgulaması sağlanmıştır. (Erkan Ü. ve ark. 2005)

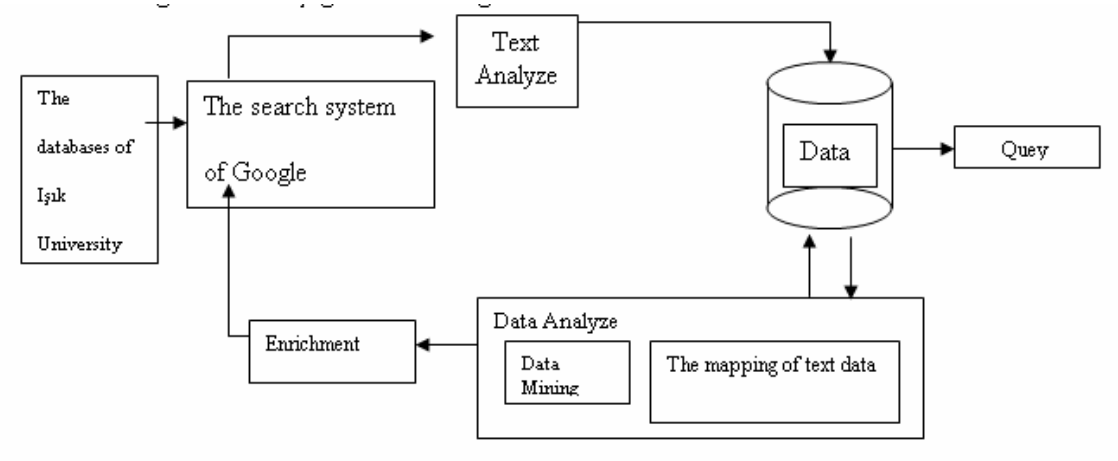
Verilerin toplanması sırasında İstihbarat açısından önemli kişi, olay ve tarihlere erişim için İnternet üzerindeki haber kaynakları taranmıştır. Başlangıç noktası olarak, Işık Üniversitesi tarafından hazırlanan Terör Örgütleri ile ilgili sayfadaki bilgiler ([6]) incelenmiş ve önemli isimler kaydedilmiştir. Daha sonra bu kaynaklarda geçen isimler üzerinde Google kullanılarak otomatik arama yapılmış, fotoğraf ve yazı içeren sonuçlar bulunarak işlenmek üzere veri tabanında saklanmıştır. Kaydedilen belgeler gereksiz bilgileri de içermektedir. Metin analizi yöntemleriyle veriler kısmen işlenerek ve sadece gerekli bilgiler daha sonra kullanılmak üzere veri tabanına kaydedilmiştir. (Erkan Ü. ve ark. 2005)

Toplanan verilerde Kişi-kişi, kişi-organizasyon, olay-yer, olay-tarih ilişkileri Apriori algoritması kullanılarak öğrenilmektedir. Öğrenilen ilişkiler sonucu ortaya çıkan yeni isimler üzerinde arama yapılması sağlanarak, yukarıda bahsedilen işlemler tekrarlanarak, veri tabanı sürekli olarak zenginleştirilebilir. (Erkan Ü. ve ark. 2005)

Kullanıcıya aranan kişiler hakkında bütün bilgileri içeren bir belge sunulmaktadır. Bu bilgiler içerisinde kişinin ismi, haberde geçen olaylar, yer ve tarihleri, birlikte bulunduğu kişilerin ismi (eğer varsa) yer almaktadır. İlişkiler de

kullanıcıya sunulduğu için, kullanıcı yeni bilgiler ışığında yeni sorgulamalar yapabilecektir. (Erkan Ü. ve ark. 2005)

Sistemin genel hatları şekil 5.1 de gösterilmektedir.

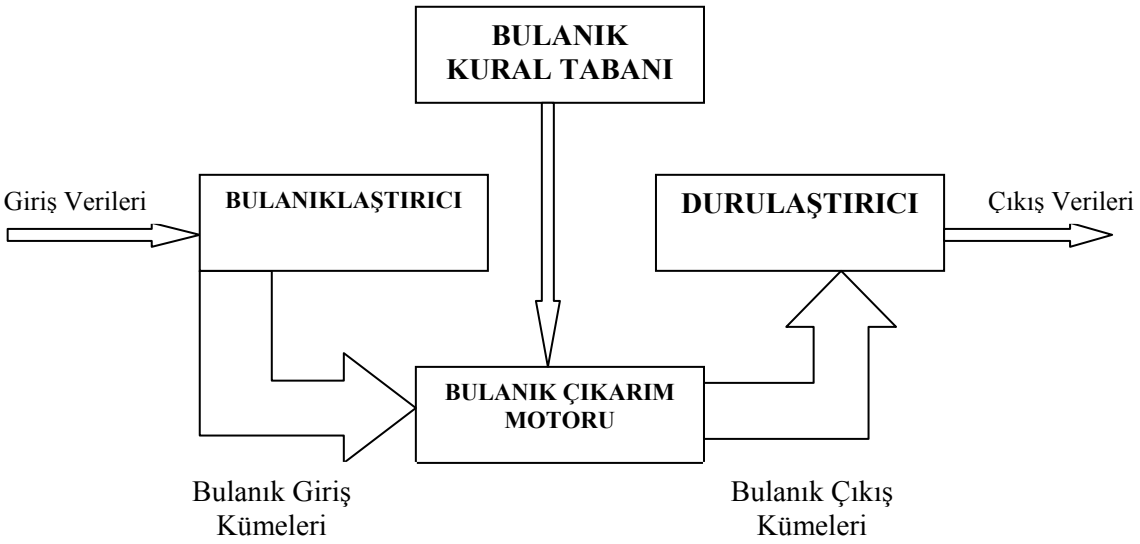


Şekil 5.1. The System of the Mining Association Rules. (Erkan Ü. ve ark. 2005)

5.2. Basın-Yayın Alanında Birliktelik Kural Madenciliği Sistemine Bulanık Mantık Uygulaması

Basın-yayın alanında birliktelik kural madenciliği sisteminde arama yapılacak kelimeler "büyük-küçük harf duyarlı" olacak şekilde ele alınır. Çalışmada uyguladığım bulanık mantık uygulaması ile bu alana girilen isimler "büyük-küçük harf duyarlı" olacak şekilde değil, veri tabanımızdaki isimler içerisinde en uygun olan isim bulunarak bu isim üzerinden arama yapılması düşünülmüştür. Çalışmanın kullanıcıya olacak yararı, arama yapılacak isimler girilirken olası yazım hataları, ismin yazılış şeklinin bilinmemesinden kaynaklanan yanlış yazımları ayıklamaktadır.

5.2.1. Çalışma Prensibi



Şekil 5.2. Bulanık Mantık Uygulamasının Şematik Gösterimi

Girilen isim ile veri tabanından karşılaştırılan ismin harf harf karşılaştırılması esas alınmıştır. Girilen isimdeki harfler tek tek veri tabanında karşılaştırılan isim içinde aranır ve en yakın konumdaki harf uzaklığı üzerinden *bulanık mantık algoritması* uygulanır. Tüm harflerden gelen sonuçlar değerlendirilip girilen isim ile veri tabanındaki isim arasındaki benzerlik oranı 0 – 1 aralığında hesaplanır.

5.2.2. Algoritmanın Açıklaması

Veri tabanımızda bulunan isim alanları algoritmamız için VT anlamına gelmektedir. VT'deki isimler ilk isimden itibaren son isim gelene kadar kullanıcı tarafından girilen kelime ile karşılaştırılır. Tüm karşılaştırma sonucunda elde edilen en yüksek benzerlik oranı olan kelime yazılmak istenilen ancak yanlış yazılan kelime olarak seçilir.

Algoritma:

-Kullanıcı tarafından girilen kelime(KW) için;

-VT'deki her isim(Name) için;

-KW_i Name içinde aranır;

-Eğer $KW_i = Name_k$ ise;

-Mesafe(x) = i-k olur.

-x değerine göre üyelik fonksiyonlarında yerine konur.

-Üyelik fonksiyonlarından elde edilen değer bulanık çıkarım motoruna gönderilir.

-Bulanık çıkarım motorunda bu değer Bulanık Kural Tablosundaki kurallara göre durulaştırma fonksiyonuna gönderilir.

-Durulaştırma fonksiyonundan elde edilen sonuç

-KW_i için benzerlik oranı olur.

-KW_i nin benzerlik oranı bir dizi(D) değişkeninde saklanır.

-D'deki tüm değerlerin ortalaması alınarak, KW ile Name arasındaki benzerlik oranı(B_{name}) bulunur.

-Bulunan her B_{name}, en yüksek benzerlik oranı ile karşılaştırılır.

-Eğer B_{name} o anki en yüksek benzerlikten yüksek ise en yüksek benzerlik B_{name} olarak atanır.

Sisteme Giriş Değerleri:

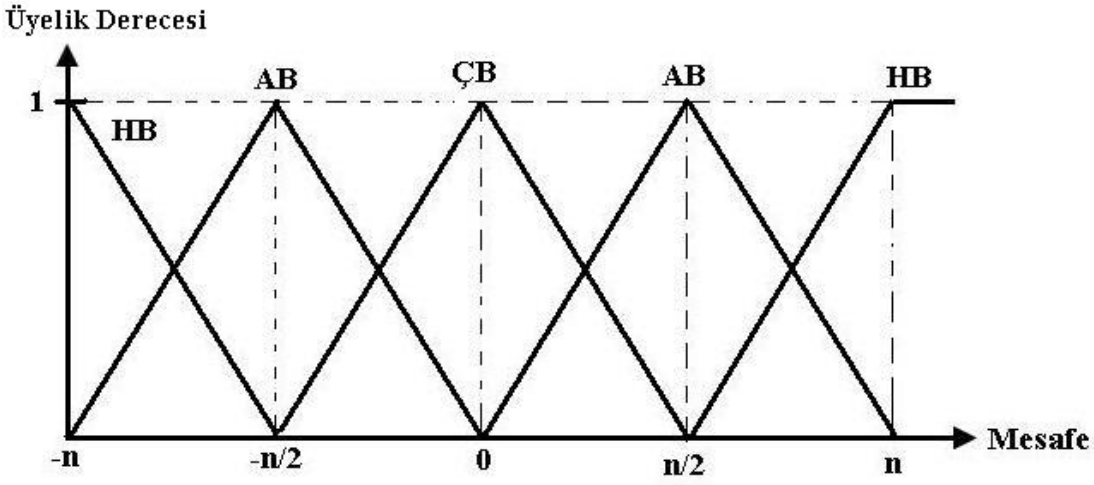
Sistemin ana girişleri kullanıcı tarafından girilen isim ile veri tabanından gelen, karşılaştırma yapılacak olan isimdir. Ancak bulanık sistemimizin giriş değerleri ele aldığımız karşılaştırma yapılacak olan isimlerin harfleridir. Sisteme kullanıcı tarafından girilen ismin harfleri asıl harflerdir. Veri tabanından gelen ismin harfleri ise içinde arama yapılacak olan kısımdır. Yani sistem kullanıcının girdiği ismi temel almaktadır.

Sistemin Çıkış Değerleri:

Sistemin çıkış değeri, kullanıcının girdiği kelimenin her harfi için elde edilen bir benzerlik değeridir. Her bir benzerlik değeri birbirinden bağımsız ve eş ağırlıklara sahiptir. Birbirinden bağımsız bu değerlerin tümünün ortalaması iki kelime arasındaki benzerlik değerini vermektedir.

5.2.2.1. Bulanıklaştırma Fonksiyonu

Mesafe : Girilen isimdeki harfin, VT'deki isimdeki bulunduğu konuma uzaklığı
n : Girilen isimdeki harf sayısı



Şekil 5.3. Üyelik Fonksiyonu

5.2.2.1.1. Üyelik Fonksiyonları

HB fonksiyonunun denklemleri:

$$\begin{cases} -n \le x \le -\frac{n}{2}; y = -\frac{2}{n}x - 1 \\ \frac{n}{2} \le x \le n; y = \frac{2}{n}x - 1 \end{cases}$$

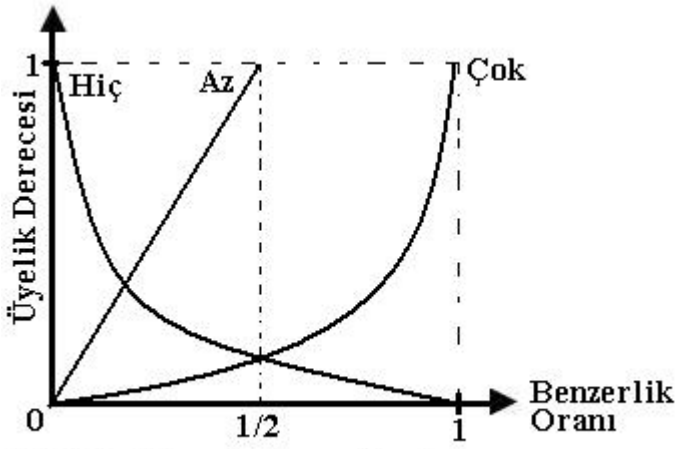
AB fonksiyonun denklemleri:
$$\begin{cases} -n \le x \le -\frac{n}{2}; y = \frac{2}{n}x + 2 \\ -\frac{n}{2} \le x \le 0; y = -\frac{2}{n}x \\ 0 \le x \le \frac{n}{2}; y = \frac{2}{n}x \\ \frac{n}{2} \le x \le n; y = -\frac{2}{n}x + 2 \end{cases}$$

ÇB fonksiyonunun denklemleri:
$$\begin{cases} -\frac{n}{2} \le x \le 0; y = \frac{2}{n}x + 1 \\ 0 \le x \le \frac{n}{2}; y = -\frac{2}{n}x + 1 \end{cases}$$

Bu denklemlerde Y uzaklığı, X üyelik derecesini ifade etmektedir.

Üyelik fonksiyonu olarak Bulanık Mantık Uygulamalarında en sık ve yaygın olarak kullanılan "Örtüşmeli Üçgen" fonksiyonu seçilmiştir. Ayrıca mesafenin negatif ve pozitif değeri için aynı sonucu almamız gerekir. Bu sebepten dolayı seçtiğimiz fonksiyonun simetrik olmalı. Örtüşmeli üçgen fonksiyonu da istediğimiz referans noktasına göre simetrik bir fonksiyondur.

5.2.2.2. Bulanık Çıkarım Motoru



Şekil 5.4. Durulaştırma Fonksiyonu

5.2.2.2.1. Bulanık Çıkarım Motoru Fonksiyonları:

Y= üyelik derecesi, X=benzerlik oranı

Çok fonksiyonunun denklemi : $X^2 + (Y-1)^2 = 1$

Az fonksiyonunun denklemi : $Y — 2X = 0$

Hiç fonksiyonunun denklemi : $(X-1)^2 + (Y-1)^2 = 1$

Bu fonksiyonların seçilmesinin sebebi, doğrusal olmayan **Çok** ve **Hiç** fonksiyonları benzerliğin az olduğu kelimeleri 0'a yakınsatmak ve benzerliğin fazla olduğu kelimeleri ise 1'e yakınsatmasıdır. Benzerliğin az olduğu **Az** fonksiyonunda ise yakınsama işlemi sıfır ya da bir'e doğru olabileceğinden, fonksiyonun buna direk etkisini azaltmak için lineer seçilmiştir. Yani bu fonksiyon benzerliği sıfır ya da bir'e doğru itelememektedir. Bu şekilde az benzeyen kelime ile çok benzeyen kelime arasındaki fark lineer fonksiyonlara göre daha çok açılacaktır ve yorum yapmayı kolaylaştıracaktır. Durulaştırma fonksiyonunda hedeflenen program kullanıcısının karar vermesini kolaylaştıracak sonuç vermektir. Yani iki kelimenin benzerlik oranının ortalama değerlerde olması hedeflenmektedir. Ara değerler yukarı ya da aşağı itelenerek kullanıcıya daha net sonuçlar verecektir.

5.2.2.3. Bulanık Kural Tablosu:

Eğer üyelik derecesi **ÇB** üyelik fonksiyonundan geliyor ise; durulaştırma işlemi için **Çok** fonksiyonu kullanılır.

Eğer üyelik derecesi **AB** üyelik fonksiyonundan geliyor ise; durulaştırma işlemi için **Az** fonksiyonu kullanılır.

Eğer üyelik derecesi **HB** üyelik fonksiyonundan geliyor ise; durulaştırma işlemi için **Hiç** fonksiyonu kullanılır.

Kuralları sözel olarak açıklayacak olursak;

Eğer harflerimiz birbirine çok yakın ise, bu durumda benzerlik fazla olacaktır ve üyelik fonksiyonlarından ÇB'ye üye olacaktır. Bu durumda durulaştırma fonksiyonu olarak Çok fonksiyonu kullanılacaktır. Eğer harflerin yakınlığı çok yakın değil ama uzak ta değil ise, bu durumda benzerlik az olacaktır ve AB üyelik fonksiyonunda bulunacaktır. Bu durumda durulaştırma fonksiyonu olarak Az kullanılacaktır. İki harfin uzaklığı çok ise, HB üyelik fonksiyonundan gelen değer dikkate alınır ve bu durumda Hiç fonksiyonu durulaştırma işlemi için kullanılır.

5.2.2.4. Durulaştırıcı

Bulanık mantık algoritmasının durulaştırıcı adımında, durulaştırma metotlarından "Ağırlıklı Ortalama Yöntemi" kullanılmıştır. Bu yöntem en çok kabul gören yöntemlerden biridir. Ayrıca konumuz isimlerin benzerlik oranını bulmak olduğu için ve algoritmada isimlerin harflerinden birisinin diğerinden daha önemli olması gibi bir tutum olmadığı için yani kelimelerdeki tüm harfler eşit ağırlıklara sahip oldukları için durulaştırma işleminde ağırlık ortalama yöntemi kullanılması en mantıklı ve sonucu en sağlıklı durulaştırma işlemi olarak görülmüştür.

$$Z^* = \frac{\sum ü}{ü(n)}$$

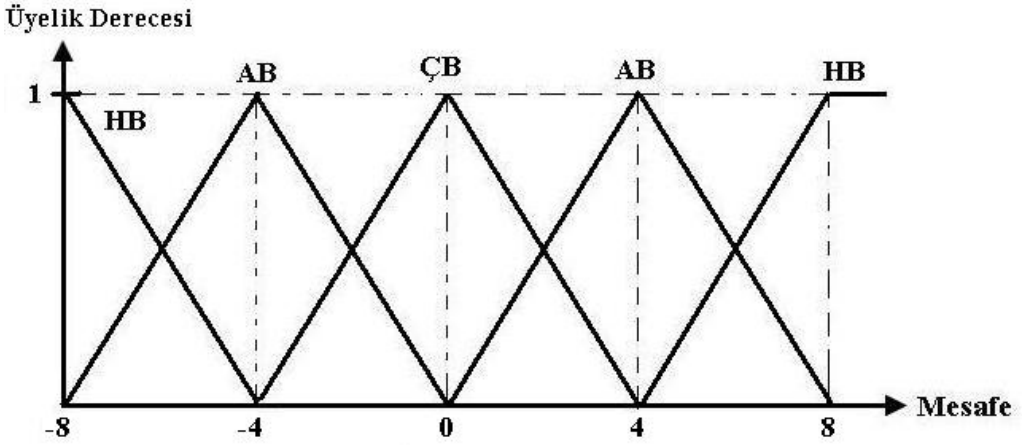
(Z^* = Durulaştırılmış Değer, Kelimenin Benzerlik Oranı, ü=her harfin benzerlik oranı, ü(n)=harf sayısı)

5.2.3. Algoritmanın Bir Örnekle Açıklaması

Veri tabanımızda bulunan isimlerden bit tanesi "TALABANİ" olsun. Kullanıcı bu ismi yazarken "ATBALANE" olarak yanlış yazmış kabul edelim. Algoritmamızı bu iki kelimenin benzerliğini bulamak üzere uygular isek;

n = 8 olacaktır.

n'nin 8 değeri için üyelik fonksiyonumuz;



Şekil 5.6. n=8 için oluşan üyelik fonksiyonu

"ATBALANE" Kelimesinin(KW) her harfi için algoritmamız ayrı ayrı uygulanır.

"A" için;

KW deki ilk "A"nın VT'deki kelimemiz içindeki en yakın "A" ya olan uzaklığı -1 birimdir.

u = 1 – 2 = -1;

1 2 3 4 5 6 7 8

T A L A B A N İ

1-2 = -1

A T B A L A N E

1 2 3 4 5 6 7 8

Üyelik fonksiyonlarımızdan mesafe(x)'in -1 değeri için elde edeceğimiz üyelik dereceleri;

AB fonksiyonu için; $\ddot{u}_{(A)} = 0,25$

ÇB fonksiyonu için; $\ddot{u}_{(A)} = 0,75$

Bulanık kural tabanımızdaki;

- Eğer bulanık giriş değeri AB fonksiyonundan geliyor ise çıkış değeri **Az** fonksiyonu…

ve

- Eğer bulanık giriş değeri ÇB fonksiyonundan geliyor ise çıkış değeri **Çok** fonksiyonu…

kurallarını uygularsak;

A harfi için; AB fonksiyonundaki üyelik derecesi = 0,25
ÇB fonksiyonundaki üyelik derecesi = 0,75

Az fonksiyonunun Y (üyelik derecesi) = 0,25 için benzerlik oranı = 0,125.

Çok fonksiyonunun Y (üyelik derecesi) = 0,75 için benzerlik oranı = 0,968 Buradan $\ddot{u}_{(A)} = 0,968$ bulunur.

T harfi için; AB fonksiyonundaki üyelik derecesi = 0,25
ÇB fonksiyonundaki üyelik derecesi = 0,75

Az fonksiyonunun Y (üyelik derecesi) = 0,25 için benzerlik oranı = 0,125.

Çok fonksiyonunun Y (üyelik derecesi) = 0,75 için benzerlik oranı = 0,968 Buradan $\ddot{u}_{(T)} = 0,968$ bulunur

B harfi için; AB fonksiyonundaki üyelik derecesi = 0,5
ÇB fonksiyonundaki üyelik derecesi = 0,5

Az fonksiyonunun Y (üyelik derecesi) = 0,5 için benzerlik oranı = 0,225.

Çok fonksiyonunun Y (üyelik derecesi) = 0,5 için benzerlik oranı = 0,866 Buradan $\ddot{u}_{(B)} = 0,866$

A harfi için; AB fonksiyonundaki üyelik derecesi = 0
ÇB fonksiyonundaki üyelik derecesi = 1

Az fonksiyonunun Y (üyelik derecesi) = 0 için benzerlik oranı = 0.

Çok fonksiyonunun Y (üyelik derecesi) = 1 için benzerlik oranı = 1 Buradan $\ddot{u}_{(A)} = 1$

L harfi için; AB fonksiyonundaki üyelik derecesi = 0,5

ÇB fonksiyonundaki üyelik derecesi = 0,5

Az fonksiyonunun Y (üyelik derecesi) = 0,5 için benzerlik oranı = 0,225.

Çok fonksiyonunun Y (üyelik derecesi) = 0,5 için benzerlik oranı = 0,866 Buradan $\ddot{u}_{(L)} = 0,866$

A harfi için; AB fonksiyonundaki üyelik derecesi = 0

ÇB fonksiyonundaki üyelik derecesi = 1

Az fonksiyonunun Y (üyelik derecesi) = 0 için benzerlik oranı = 0.

Çok fonksiyonunun Y (üyelik derecesi) = 0 için benzerlik oranı = 1 Buradan $\ddot{u}_{(A)} = 1$

N harfi için; AB fonksiyonundaki üyelik derecesi = 0

ÇB fonksiyonundaki üyelik derecesi = 1

Az fonksiyonunun Y (üyelik derecesi) = 0 için benzerlik oranı = 0.

Çok fonksiyonunun Y (üyelik derecesi) = 1 için benzerlik oranı = 1 Buradan $\ddot{u}_{(N)} = 1$

E harfi için; AB fonksiyonundaki üyelik derecesi = 0

ÇB fonksiyonundaki üyelik derecesi = 0

Az fonksiyonunun Y (üyelik derecesi) = 0 için benzerlik oranı = 0.

Çok fonksiyonunun Y (üyelik derecesi) = 0 için benzerlik oranı = 0 Buradan $\ddot{u}_{(E)} = 0$

Bulunacaktır.

Durulaştırma fonksiyonumuzu kullanırsak;

$Z^* = (0,968 + 0,968 + 0,866 + 1 + 0,866 + 1 + 1 + 0) / 8 =$ **0,8335 bulunur.**

Sonuç olarak VT'deki "TALABANİ" isminin, kullanıcı tarafından girilen KW="ATBALANE" kelimesine olan benzerliği 0,8335'dır denilebilir.

5.2.4. Algoritmanın Özel Durumları

5.2.4.1. Kullanıcı tarafından girilen kelimenin veri tabanından karşılaştırılan kelimeden uzun olması durumu

Örneğin kullanıcı girdi olarak "ERDOĞAN" girmiş olsun, bu kelime veri tabanından gelen "ERCAN" kelimesi ile karşılaştırılıyor olsun.

Girilen "ERDOĞAN" kelimesi 7 karakterlidir. Karşılaştırılan "ERCAN" ise 5 karakterlidir. "ERDOĞAN" kelimesindeki 'N' harfi 7. sıradadır, "ERCAN" da ise 'N' harfi 5. sıradadır. Ancak bu harf iki kelimede de son sıradadır. Bu durumda bu harfin iki kelime üzerindeki yeri aslında aynıdır.

Bu durum oluştuğunda sistemin esas aldığı kelime kullanıcı tarafından girilen kelime olduğundan veri tabanından gelen karşılaştırma kelimesi kullanıcının girdiği kelimenin boyutuna göre sündürülür. Verdiğimiz örneğe uyarlayacak olursak, "ERCAN" kelimesi "ERDOĞAN" kelimesinin boyutuna göre sündürülür. 5 harfin konumları 7 harfle eşleştirilebilecek hale getirilir.

7 harfli "ERDOĞAN" kelimesinde birbiri ardına gelen harflerin uzaklığı 1 birim olarak hesaplanırken, 5 harfli "ERCAN" kelimesinde birbiri ardına gelen harflerin uzaklığı 1,5 birim olarak alınmaktadır. Bu hesaplama şu şekilde yapılmaktadır:

- Kullanıcı tarafından girilen kelimenin uzunluğu L1 olsun.
- Veri tabanından gelen karşılaştırılacak olan kelimenin uzunluğu ise L2 olsun.
- Veri tabanından gelen kelimenin harfleri arasıdaki uzaklığı ise X birim olsun.

L1 > L2 ve L2,L1>2 için;

$$X = \frac{L1 - 2}{L2 - 2}$$

Bu durumda X sıfırdan büyük bir değer çıkacaktır…



Şekil 5.7 Aynı uzunlukta olmayan kelimeler için daraltma işlemi

5.2.4.2. Kullanıcı tarafından girilen kelimenin veri tabanından karşılaştırılan kelimeden kısa olması durumu

Bir önceki örneği kullanacak olursak, bu sefer kullanıcı girdi olarak "ERCAN" girmiş olsun, bu kelime veri tabanından gelen "ERDOĞAN" kelimesi ile karşılaştırılıyor olsun.

Girilen "ERCAN" kelimesi 5 karakterlidir. Karşılaştırılan "ERDOĞAN" ise 7 karakterlidir. "ERCAN" kelimesindeki 'N' harfi 5. sıradadır, "ERDOĞAN" da ise 'N' harfi 7. sıradadır. Ancak bu harf iki kelimede de son sıradadır. Bu durumda bu harfin iki kelime üzerindeki yeri aslında aynıdır.

Bu durum oluştuğunda sistemin esas aldığı kelime kullanıcı tarafından girilen kelime olduğundan veri tabanından gelen karşılaştırma kelimesi kullanıcının girdiği kelimenin boyutuna göre daraltılır. Verdiğimiz örneğe uyarlayacak olursak, "ERDOĞAN" kelimesi "ERCAN" kelimesinin boyutuna göre daraltılır. 7 harfin konumları 5 harfle eşleştirilebilecek hale getirilir.

5 harfli “ERCAN” kelimesinde birbiri ardına gelen harflerin uzaklığı 1 birim olarak hesaplanırken, 7 harfli “ERDOĞAN” kelimesinde birbiri ardına gelen harflerin uzaklığı 0,66 birim olarak alınmaktadır. Bu hesaplama şu şekilde yapılmaktadır:

- Kullanıcı tarafından girilen kelimenin uzunluğu L1 olsun.
- Veri tabanından gelen karşılaştırılacak olan kelimenin uzunluğu ise L2 olsun.
- Veri tabanından gelen kelimenin harfleri arasıdaki uzaklığı ise X birim olsun.

Yukarıdaki formülü aynı şekilde uygulayabiliriz.

L1 > L2 ve L1,L2>2 için;

Bu durumda X sıfırdan küçük bir değer çıkacaktır…

$$X = \frac{L1 - 2}{L2 - 2}$$



Şekil 5.8 Aynı uzunlukta olmayan kelimeler için sündürme işlemi

5.2.5. Algoritmanın Performans Analizi

Algoritma tek başına, birliktelik kural madenciliği içinde kullanılmadan performans testi yapılmıştır. Test için algoritma Turbo C++ editöründe yazılmıştır. Kullanılan bilgisayar; Microsoft Windows XP Professional with Service Pack 2 işletim istemi yüklü olan Intel Pentium III 863Mhz işlemcili ve 512 MB belleği olan bir makinedir. Testler sırasında sistemde Turbo C++ programından başka hiçbir program çalışmamıştır.

Test kütüklerinin özellikleri:

Test için kullanılan dosyalar diske 100 Bin, 200 Bin, 400 Bin, 600 Bin ve 800 Bin kayıt içerecek şekilde kaydedilmiştir. Bu boyutların üzerindeki testlerde ise bu dosyalar birbiri ardına eklenerek kullanılmıştır.

Bu kütüklerde geçen kelimeler;

- saddam
- laden
- erdogan
- mahmut
- talabani
- ahmet
- mehmet
- mukaddes
- nurullah
- serdar

Yukarıdaki 10 isim birbiri ardına tekrarlanarak kütüklere yazılmıştır. Bu değişik harf sayısına sahip 10 adet ismin ortala harf miktarı;

(6+5+7+6+8+5+6+8+8+6)/10=6,5 dir.

Test isimlerinin özellikleri:

Yukarıda açıklanan test kütüklerinde denenmek üzere 5 adet isim seçilmiştir. Bu isimler değişik karakter uzunlarına sahiptir. En kısa isim 3 harfli “ALİ” dir. Diğer isimler, 5 harfli “AHMET”, 7 harfli

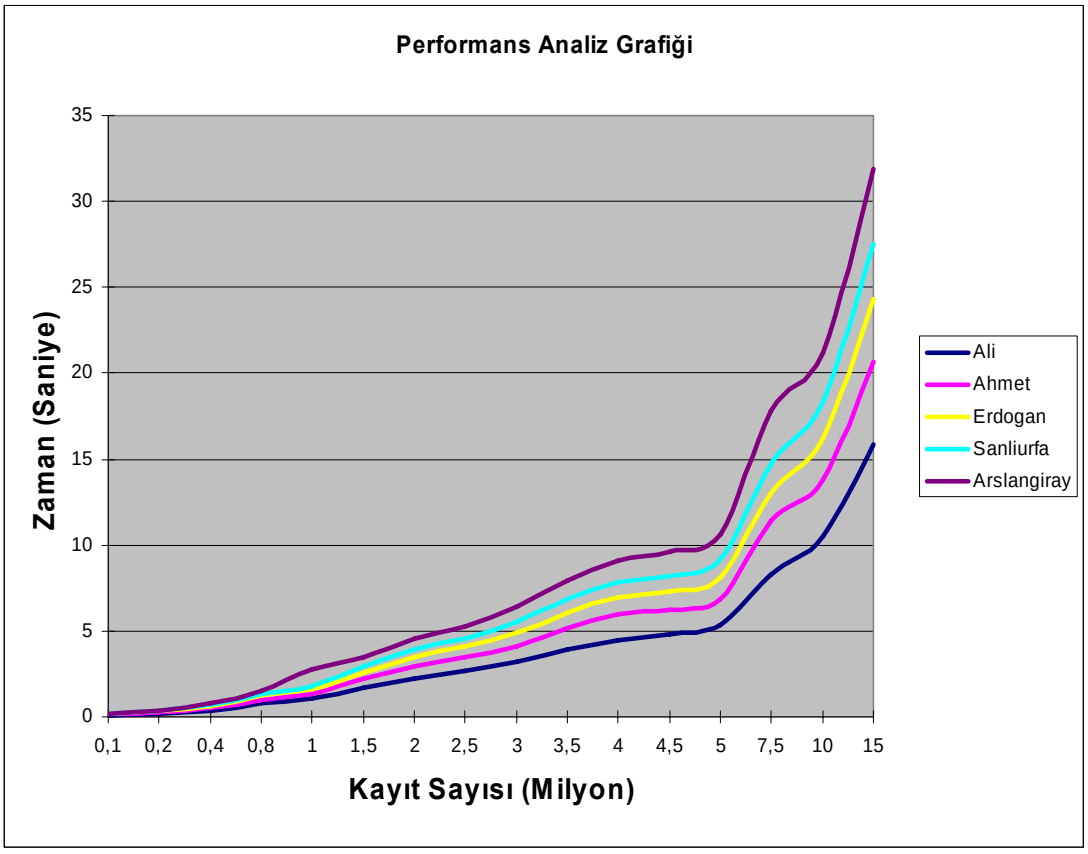
"ERDOGAN", 9 harfli "SANLIURFA" ve 11 harfli "ARSLANGIRAY" dır.

Yukarıda belirtilen kıstaslarda algoritma, ortalama karakter uzunluğu 6,5 karakter olan 100bin, 200bin, 400bin, 800bin, 1milyon, 1,5milyon, 2milyon, 2,5milyon, 3milyon, 3,5milyon, 4milyon, 4,5milyon, 5milyon, 7,5milyon, 10milyon ve 15milyon kayıt için, 3, 5, 7, 9 ve 11 karakter uzunluğundaki isimler girilerek test edilmiştir. Test sonuçları bir tablo şekline getirilerek Tablo 5.1'de verilmiştir.

Karşılaştırılan isim / **Kayıt Sayısı**	Ali (3 harf)	Ahmet (5 Harf)	Erdogan (7 Harf)	Sanliurfa (9 Harf)	ArslanGiray (11Harf)
100 Bin	0,11 sn	0,17 sn	0,16 sn	0,17 sn	0,22 sn
200 Bin	0,22 sn	0,27 sn	0,33 sn	0,39 sn	0,38 sn
400 Bin	0,38 sn	0,55 sn	0,66 sn	0,72 sn	0,82 sn
800 Bin	0,82 sn	0,99 sn	1,21 sn	1,32 sn	1,48 sn
1 Milyon	1,10 sn	1,38 sn	1,64 sn	1,82 sn	2,80 sn
1,5 Milyon	1,65 sn	2,25 sn	2,58 sn	2,97 sn	3,46 sn
2 Milyon	2,20 sn	2,96 sn	3,46 sn	3,90 sn	4,56 sn
2,5 Milyon	2,69 sn	3,46 sn	4,10 sn	4,56 sn	5,27 sn
3 Milyon	3,19 sn	4,12 sn	4,89 sn	5,49 sn	6,37 sn
3,5 Milyon	3,90 sn	5,16 sn	6,10 sn	6,86 sn	7,97 sn
4 Milyon	4,45 sn	5,93 sn	6,92 sn	7,86 sn	9,11 sn
4,5 Milyon	4,77 sn	6,21 sn	7,31 sn	8,23 sn	9,62 sn
5 Milyon	5,33 sn	6,87 sn	8,13 sn	9,17 sn	10,60 sn
7,5 Milyon	8,24 sn	11,4 sn	12,96 sn	14,67 sn	17,80 sn
10 Milyon	10,55 sn	13,79 sn	16,25 sn	18,35 sn	21,20 sn
15 Milyon	15,82 sn	20,65 sn	24,33 sn	27,52 sn	31,91 sn

Tablo 5.1 Performans analiz tablosu

Yapılan testlerin grafiği Grafik 5.1'de verilmiştir.



Grafik 5.1 Performans analiz grafiği

5.2.6. Algoritmanın Lineer Algoritma ile Karşılaştırması

5.2.6.1. Lineer algoritmanın açıklanması

Uygulanan algoritma temelde iki kelimenin birbirine olan benzerlik oranını yüzde olarak vermeye dayalı bir algoritmadır. Bu benzerlik oranı bulunurken temel kural iki kelimede de ortak olan harflerin bulundukları kelimelerdeki konumlarını ele alınmaktadır.

Uygulanan bulanık sistem araştırdığımız iki kelimenin benzerlik oranını bulanık mantık kurallarına dayandırılarak, bulanık mantık kuralları çerçevesinde bulunmaktadır. Uyguladığımız sistemi bulanık olmayan, yani lineer bir sistem olarak

ta tasarlayabiliriz. İki kelimede de ortak olan harflerin konumları arasındaki fark bir tablo yardımıyla ele alınabilir. Örneğin ele alınan harf iki kelimede de 1. sırada ise yani kelimelerdeki konumları arasındaki fark sıfır ise, bu kelime için 100 puan, eğer kelimelerdeki konumları arasındaki fark 1 ise 85 puan, 2 ise 70 puan gibi bir puanlama tablosu oluşturulabilir. Sonra iki kelime arasındaki benzerlik oranı, elde edilen puanların alınabilecek maksimum puana oranı hesaplanarak bulunabilir.

5.2.6.2. Bulanık sistemin lineer sisteme göre avantajları

Araştırdığımız benzerlik oranını bulmak için bulanık bir sistem kullanmak lineer bir sistem kullanmaktan çok daha avantajlıdır. Öncelikle lineer sistem kelimelerin boyutları analiz etmeden puan tablosundan elde edilen puanların vereceği benzerlik oranına göre bir sonuç verecektir. Ancak bulanık sistemde kelimelerin boyutları analiz edilmekte ve buna göre bir sonuç elde edilmektedir. Örnek olarak "ahmet" kelimesi ile "metin" kelimesi arasındaki karşılaştırma ile "orhan" kelimesi ve "erdogan" kelimesi arasındaki benzerlik farklı ele alınmalıdır. İlk iki kelimedeki harf sayıları eşit ama diğer iki kelimedeki harf sayıları birbirlerine eşit değildir. Yani boyutları aynı olan iki kelime arasındaki benzerlik hesaplanırken bir harfin iki kelimedeki konumları birbirlerine özdeş iken, kelime boyutları farklı olunca bir harfin iki kelimedeki konumları birbirlerine özdeş değildir. Bulanık sistem bu farklığı ele alabilirken lineer sistem bu farklılıkları ele alamamaktadır.

Harf sayısı 3 olan iki kelimenin kıyaslanması ile harf sayısı 11 olan iki kelimenin kıyaslanması da farklı olmalıdır. 3 harfli kelimelerdeki bir birimlik konum farkı, 11 harfli kelimelerdeki bir birimlik konum farkı ile aynı değildir. Lineer sistemde bu farklar çok belirgin bir şekilde görülemezken, bulanık sistemde durum farklıdır. Bu sistemde her kelime girildiğinde üyelik fonksiyonu girilen kelimedeki harf sayısına göre yeniden şekillenmekte, sınırları o kelimeye özgü bir şekilde düzenlenmektedir. Yani 5 harfli bir kelime ile 6 harfli bir kelimenin üyelik fonksiyonları birbirlerinden farklıdır. Durum böyle olunca 5 harfli bir kelimedeki

harfin bulunduğu 2. konum ile 6 harfli bir kelimedeki harfin bulunduğu 2. konum birbirlerinden farklıdır.

Lineer sistemde benzerlik oranı bulunurken puan tablosundan elde edilen değer, alınabilecek maksimum puana bölünerek bulunmaktadır. Bu durumda kullanıcıya 0 – 100 aralığında belli başlı değerler verilecektir. Ancak bulanık sistemde durum farklıdır. Bulanık çıkarım motoru fonksiyonlarımız ara değerleri kullanıcıya minimum oranda göstermeyi hedefleyecek biçimde hazırlanmıştır. Çok benzer durumlarda benzerlik değeri yukarıya doğru, benzerliğin az olduğu durumlarda ise aşağıya doğru itilecek şekilde hazırlanan bulanık çıkarım motoru fonksiyonları arada kalan benzerlik değerlerini elemeyi ve kullanıcıya daha kesin sonuçlar vermeyi hedeflemiştir.

6. SONUÇLAR

6.1. Çalışmanın Bulanık Mantık Kısmının Sonuçları

Apriori algoritması kullanılarak yapılan "Birliktelik Kural Madenciliği" çalışmasının kişi-kişi birlikteliğinin bulunmasında yapılan yazım hatalarını elemek amaçlı uygulanan bulanık mantık çalışması son derece önemlidir. Bu çalışmanın sonuçlarına göre "büyük-küçük harf" duyarlı bir karşılaştırma yapan klasik karşılaştırma algoritmasına göre "bulanık mantık" algoritması kullanılarak yapılan karşılaştırma algoritması çok önemli bir üstünlük sağlamıştır. Ayıca "bulanık mantık" algoritması uygulanarak gerçekleştirilen algoritma, lineer olarak gerçekleştirilen benzerlik algoritmalarına da üstünlük sağlamıştır. Lineer bir algoritma ile bulanık mantık ile yapılan algoritmanın birkaç ismin karşılaştırmasına göre sonuçları aşağıdaki tablo 6.1'de verilmiştir.

V.T. deki isim	Girilen İsim	Lineer Algoritma	Bulanık Mantık
SADDAM	TASDAM	0.76	0.79
MEHMET	METHEM	0.80	0.86
TALABANİ	TABALANİ	0.92	0.95
ERDOGAN	EDROGEN	0.83	0.92

Tablo 6.1. Bulanık mantık ile Lineer algoritmanın Karşılaştırma Sonuçları

6.2. Çalışmanın Veri Madenciliği Kısmının Sonuçları

Elde edilen görsel ve metinsel verinin saklanması, işlenmesi ve sorgulanmasını sağlayacak bir saklama mekanizmasının gereği hissedilmektedir. Bu

nedenle, hem daha güvenli ve verimli olması için hem de gerekli sorguların mümkün olan en hızlı bir şekilde yapılması için bir veri tabanının tasarlanması ve oluşturulmasına karar verilmiştir. Bu amaçla MySQL Veri Tabanı Yöneticisi Sistemi kullanılarak metinsel veri tabanı oluşturulmaktadır. Bu veri tabanı sayesinde Hypertext Preprocessor (PHP) script dilini kullanarak geliştirilmiş yazılımla web üzerinden de bir arayüzle sorgulama yapılması sağlanmaktadır.

İstihbarat açısından önemli kişiler hakkındaki bilgilerin toplanması için, Işık Üniversitesi Öğretim Üyesi ve Milli İstihbarat Teşkilatı Eski Müsteşarı Sönmez Köksal'ın hazırlamış olduğu Terör Örgütleri ve bunların ilişkilerini anlatan Veri Tabanı ([6]) öncelikli kaynak olarak seçilmiştir. Bu kaynak, örgütlerin kurucuları ve önemli elemanları ile ilgili ilk bilgilerin elde edilmesi için kullanılmıştır.

Bir sonraki aşama verilerin internet kullanılarak zenginleştirilmesidir. Işık Üniversitesi Terör Örgütleri Veri Tabanından elde edilen önemli kişi isimleri, Google Arama Motoru (http://www.google.com) aracılığı ile İnternet üzerindeki haber içerikli metinsel (haber metni, resimler altındaki açıklayıcı bilgiler, vs.) ve görsel (haber ile ilgili resimler) verilere ulaşmak için kullanılmıştır. Bu işlem Google Arama Motorunun 'grafikler' seçeneğini kullanarak kişi isimleri üzerine arama yapma ve bulunan sayfaların tümünün (resim, yazı, vs.) bilgisayarın sabit diskine kopyalanması aşamalarını içermektedir. Bu amaçla sed, wget gibi hazır programların yanısıra, elde edilen web sayfalarını ayrıştıran, C Programlama diliyle yazılmış geliştirilen fonksyon ve rutinler kullanılmıştır.

Metinsel bilgi haberin çok önemli bir parçasıdır. Bu sebeple metin üzerinde detaylı bir inceleme yapılması ve metin içinde geçen kişi adlarının, yer ve zaman öğelerinin çıkarılması ve diğer kelimelerden ayrıştırılması büyük önem taşımaktadır. Bu bilgiler şimdiye kadar kısmen ve elle ayrıştırılmıştır.

Daha sonra ayrıştırılmış olan bu anahtar kelimeler item olarak kabul edilerek her bir haberin frequent itemset'leri oluşturulmuş ve Apriori algoritmasının first candidate itemsets olarak algoritmaya sunulmuştur. Apriori algoritması 4-frequent itemset'e kadar çalışmaktadır.

Tablo 6.2'de güven ve destek değerleri ile kiş-yer, kişi-kişi, kişi-olay sorgulamalarından bir bölüm görülmektedir. Burada gerçek kişi adları yerine semboller kullanılmıştır.

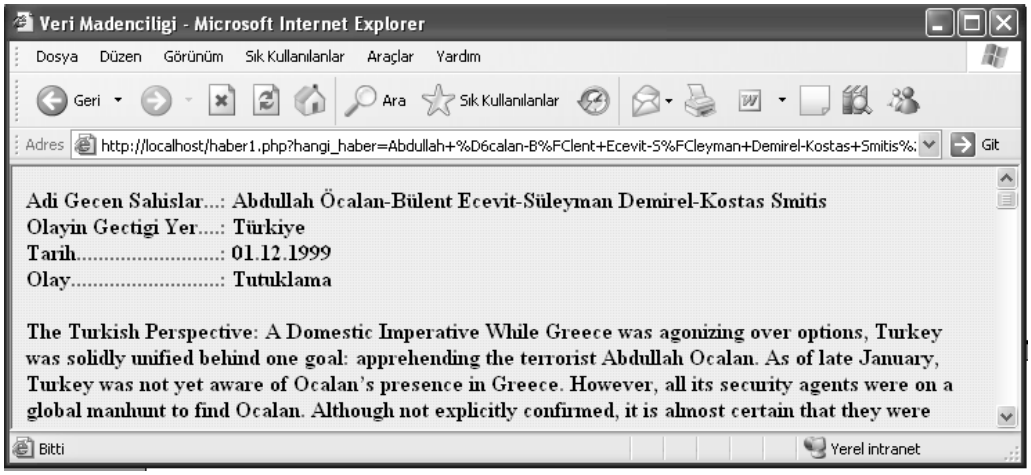
Tablo 6.2 Destek ve Güven Değeri İle kiş-yer, kişi-kişi, kişi-olay Birliktelik Kurallarından Bazıları

A	B	Support (A⇒B)	Confidence (A⇒B)	The Founded Other items
Abdullah	Irak	0.4321	0.102	Bülent Ecevit, Türkiye, 2002, Yorum, Terör, Tayyip Erdoğan, ...
Kemalpir, Öcalan	Türkiye	0.7435	0.698	Terör, Örgüt, Şehit, 2001, 1999, 1985, 1995, Şemdin sakık, Talabani, ...
Kürt Sorunu	Barzani	0.3281	0.508	Saddam Hüseyin, Usame Bin Laden, Talabani, Irak, Türkiye, Tayyip Erdoğan,. .
Laden	Bush	0. 281	0. 859	A.B.D., Bombalama, Irak, Saldırı, Arama, Elebaşı, İkiz Kule, suçlama, terör, ...

Sorgulamaya imkan veren PHP arayüzünden bir görüntü de Şekil 6.1'de sunulmuştur. Sorgulama sonucu ilgili habere ulaşıldığında elde edilen ekran görüntüsü de Şekil 6.2'de verilmiştir.



Şekil 6.1. The PHP interface of the application.



Şekil 6.2. reached the news.

KAYNAKLAR

[1] Cios, K. J. Pedrycz, W. Swiniarski, R. W., (1998), Data Mining Methods for Knowledge Discovery, Boston: Kluwer Academic Publisher.

[2] Delaney, K., (2000), Inside Microsoft SQL Server 2000, Washington: Micrpspft Pres.

[3] IBM Research Division, (1998), Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications, Sun Rose, CA: Sunita Sarawagi, Shiby Thomas, Rakesh Agrewal.

[4] Robert Grossman, (1999), Data Mining Research.

[5] Tukey, J, (1973), Exploraty Data Analysis.

[6] Alan Rae, Data Mining Student Notes – The Queen's University of Belfast.

[7] Habib Youssef, Sadiq M. Sait, Hussain Ali, (2002), Fuzzy Simulated Evolution for VLSI Cell Placement.

[8] Yun-Teng Roan, Pei-Yin Chen, (2000), A Fuzzy Search Algorithm for the Estimation of Motion Vectors.

[9] Osei Adjei, Li Chen, Heng-Da Cheng, Don H. Cooley, Rei J. Cheng, Xander Twombly, (2001), A Fuzzy Search Method for Rough Sets in Data Mining.

[10] Yen-Cheng Tseng, Tsung-Ying Sun, (2005), Adaptive Fuzzy Search Algorithm for Improving the Efficiency of Block Motion Estimation.

[11] Zekai Şen, (2001), Bulanık Mantık ve Modelleme İlkeleri,.

[12] Chun-Hsien Wu, Yeou-Jiunn Chen, (2001), Multi-Keyword Spotting of Telephone Speech Using a Fuzzy Search Algorithm and Keyword-Driven Two-Level CBMS.

[13] Paul Horng-Jyh Wu, Jin-Cheon Na, Christopher S.G. Khoo, (2004), NLP Versus IR Approaches to Fuzzy Name Searching in Digital Libraries.

[14] M. Nikravesh, V. Loia, B. Azvine, (2002), Fuzzy Logic and the Internet (FLINT): Internet, World Wide Web, and Search Engines.

[15] Baykal N., (2003) Veri Tabanı ve Veri Madenciliği, Tıp Bilişimi Güz Okulu, Antalya.

[16] Frawley, W. J., et al., 1991. Knowledge Discovery in Databases: An Overview. In: Knowledge Discovery in Database. Edited by Piatetsky-shapiro, G., et al., AAAI/MIT Press, Menlo Park, CA

[17] Cios, K.J., Teresinska, A., Konieczna, S., Potocka, J., Sharma, S., (2000) Diagnosing Myocardial Perfusion from PECT Bull's-eye Maps - A Knowledge Discovery Approach, IEEE Engineering in Medicine and Biology Magazine, Special issue on Medical Data Mining and Knowledge Discovery.

[18] David J. Hand, (1998), The American Statistician.
[19] Megaputer Intelligence Inc., (2000)
[20] Nariman A, (2005), Designing a Framework Of Intelligent Information Process On Dentistry Administration Data.:FASS
[21] Tapan S., Zhao D., (2004), Artificial intelligence and data mining techniquies in electricity market forecasting.
[22] Sorin D., (2000),Data Analysis Tools for DNA Microarray data, BioDiscovery Inc. and Computer Science Department, Wayne State University.
[23] Koyuncu E., (2004), Yeni Matematiksel Kod; Bulanık Mantık, Teknolojist, ITU IEEE
[24] Erkan Ü., Eyüp S., Emre Ç., Harun U., Ahmet A., (2005), Web Basın Verilerine Apriori Algoritması Uygulanarak Düzenli Birliktelik Kurallarının Bulunması: 4rd International Advanced Technologies Symposium September.