

**EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ  
(YÜKSEK LİSANS TEZİ)**

**REGRESYON ANALİZİNDE  
'SOFT COMPUTING' TEKNİKLERİ**

**Gülbahar EZEL**

**İstatistik Anabilim Dalı  
Bilim Dalı Kodu: 4060101  
Sunuş Tarihi: 20 / 06 / 2006**

**Tez Danışmanı: Prof. Dr. Efendi NASİBOV**

**Bornova-İzmir**



Sayın Gülbahar EZEL tarafından YÜKSEK LİSANS tezi olarak sunulan **“Regresyon Analizinde ‘Soft Computing’ Teknikleri”** başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve .../.../2006 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı:

.....

Raportör Üye:

.....

Üye:

.....



## **TEŐEKKÜR**

Bu alıŐma sűresince gűrűŐlerinden yararlandıĐım sayın Prof. Dr. Efendi Nasibov'a, bana her tűrlű kolaylıĐı saĐlayan ve ilgilerini esirgemeyen İstatistik bűlűmű elemanlarına ve hep yanımda olduklarını hissettiĐim aileme teŐekkűrű bir bor bilirim.



## ÖZET

### REGRESYON ANALİZİNDE SOFT COMPUTING TEKNİKLERİ

EZEL, Gülbahar

Yüksek Lisans Tezi, İstatistik Bölümü

Tez Yöneticisi: Prof. Dr. Efendi NASİBOV

Haziran 2006, 152 sayfa

Bu tezde Soft Computing Teknolojilerinden olan bulanık mantık, genetik algoritmalar, yapay sinir ağları ve yapay zeka teknikleri incelenerek literatürdeki değişik bulanık en küçük kareler regresyon modellerine değinilmiş ve bu modellerin uygulanmasıyla karşılaşılabilecek sakıncalı durumlar incelenmiştir.

Switching regresyon incelenerek klasik regresyondan ne gibi farklılıklara sahip olduğuna değinilmiştir. Bulanık yapay sinir ağlarının bulanık regresyonda kullanımını incelenerek, bir bulanık uyarlanabilir sinir ağı algoritmasının (FAN), Borland C++ Builder 6.0 programlama dilinde programı yapılmıştır.

**Anahtar sözcükler:** Soft computing, Bulanık en küçük kareler, Switching regresyon, Sinir ağları, Bulanık mantık, Genetik algoritmalar.





## ABSTRACT

### SOFT COMPUTING TECNICS IN REGRESSION ANALYSIS

EZEL, Gülbahar

MSc in Statistics Department

Supervisor: Prof. Dr. Efendi NASİBOV

June 2006, 152 pages

In this thesis, by investigating Fuzzy logic, genetic algorithms, neural networks and artificial intelligence in Soft Computing Tecnologies, different kinds of Fuzzy least squares regression models have been mentioned and some inconvenient situations possible to face in use of these models have been evaulated.

Switching Regression models have been investigated and their differences from classical regression have been mentioned. By investigating, Fuzzy Neural Networks in Fuzzy Regression , the programme of a Fuzzy Adaptive Networks algorithm (FAN) has been made with Borland C++ builder 6.0 programme language.

**Keywords:** Soft Computing, Fuzzy least squares, Switching regression, neural networks, Fuzzy logic, genetic algorithms.



## İÇİNDEKİLER

|   | Sayfa      |
|---|------------|
| <b>TEŞEKKÜR</b> .....                                 | <b>V</b>   |
| <b>ÖZET</b> .....                                     | <b>VII</b> |
| <b>ABSTRACT</b> .....                                 | <b>IX</b>  |
| <b>ŞEKİLLER DİZİNİ</b> .....                          | <b>XV</b>  |
| <b>ÇİZELGELER DİZİNİ</b> .....                        | <b>XIX</b> |
| <b>SİMGELER VE KISALTMALAR DİZİNİ</b> .....           | <b>XXI</b> |
| <b>1. GİRİŞ</b> .....                                 | <b>1</b>   |
| <b>2. SOFT COMPUTİNG TEKNOLOJİLERİ</b> .....          | <b>4</b>   |
| 2.1 Bulanık Mantık Ve Bulanık Kümeler Teorisi .....   | 8          |
| 2.2 Yapay Sinir Ağları .....                          | 25         |
| 2.3 Genetik Algoritmalar .....                        | 30         |
| 2.4 Yapay Zeka Teknikleri .....                       | 36         |
| <b>3. BELİRSİZLİK VE BULANIK SAYILAR</b> .....        | <b>42</b>  |
| 3.1 Bulanık Sayıların Toplanması Ve Çıkarılması ..... | 46         |
| 3.2 Bulanık Sayıların Çarpılması ve Bölünmesi.....    | 47         |
| 3.3 Durulaştırma İşlemleri .....                      | 48         |

## İÇİNDEKİLER (Devamı)

Sayfa

|  |           |
|--|-----------|
| <b>4. BULANIK REGRESYON MODELLERİ .....</b>  | <b>56</b> |
| 4.1 Klasik Ve Bulanık Regresyon Modelleri Arasındaki Farklılıklar .....  | 59        |
| 4.2 Bulanık Uzaklık Ve Farklılık Yöntemleri .....  | 60        |
| 4.3 Diamond' un bulanık regresyon modeli.....  | 61        |
| 4.4 Kim Ve Bishu'nun bulanık regresyon modeli .....  | 64        |
| 4.5 Tanaka' nın bulanık regresyon modeli.....  | 66        |
| 4.6 Nasrabadi ve Nasrabadi' nın bulanık regresyon modeli .....   | 67        |
| 4.7 Wang'ın bulanık regresyon modeli.....  | 68        |
| 4.8 Nasibov' un WABL uzaklık tanımlaması ve bulanık regresyon modeli.....  | 69        |
| 4.8.1. $R(x)=L(x)=\max(0,1- x ^s)$ kalıp fonksiyonuna sahip LR tipi bulanık sayılar için ND1'e dayanan bulanık en küçük kareler regresyon analizi..... | 73        |
| 4.8.2. $R(x)=L(x)=e^{- x ^s}$ Kalıp fonksiyonuna sahip LR tip bulanık sayılar için ND1'e dayanan bulanık en küçük kareler regresyon modeli .....       | 84        |
| 4.8.3. $R(x)=L(x)=1/(1+ x ^s)$ Kalıp fonksiyonuna sahip LR tip bulanık sayılar için ND1'e dayanan bulanık en küçük kareler regresyon modeli .....      | 89        |

## İÇİNDEKİLER (Devamı)

|  | <b>Sayfa</b> |
|--|--------------|
| 4.8.4. $R(x)=L(x)=\max(0,1- x ^s)$ kalıp fonksiyonuna sahip bulanık sayılar için ND2'ye dayanan bulanık en küçük mutlak sapmalar regresyon analizi ..... | 91           |
| 4.8.5. $R(x)=L(x)=e^{- x ^s}$ Kalıp fonksiyonuna sahip bulanık sayılar için ND2'ye dayanan bulanık en küçük mutlak sapmalar regresyon modeli .....       | 95           |
| 4.8.6. $R(x)=L(x)=1/(1+ x ^s)$ Kalıp fonksiyonuna sahip bulanık sayılar için ND2'ye dayanan bulanık en küçük mutlak sapmalar regresyon modeli .....      | 96           |
| 4.8.7. WABL determinasyon katsayısı .....  | 97           |
| <br>   |              |
| <b>5. BULANIK YAPAY SİNİR AĞLARI YAKLAŞIMININ BULANIK REGRESYON ANALİZİNDE KULLANIMI .....</b>   | <b>99</b>    |
| <br>   |              |
| <b>6. SWITCHİNG REGRESYON .....</b>  | <b>109</b>   |
| <br>   |              |
| <b>7. UYGULAMA .....</b>   | <b>115</b>   |
| <br>   |              |
| <b>8. SONUÇ VE TARTIŞMA .....</b>  | <b>126</b>   |
| <br>   |              |
| <b>KAYNAKLAR .....</b>   | <b>127</b>   |

**İÇİNDEKİLER (Devamı)**

|                                     | <b>Sayfa</b> |
|-------------------------------------|--------------|
| <b>EKLER.....</b>                   | <b>134</b>   |
| <b>EK 1 (FAN Program kodu).....</b> | <b>135</b>   |
| <b>ÖZGEÇMİŞ.....</b>                | <b>152</b>   |

## ŞEKİLLER DİZİNİ

| <b><u>Şekil</u></b> |   | <b><u>Sayfa</u></b> |
|---------------------|---|---------------------|
| Şekil 2.1           | Bir arabirim olarak yumuşak hesaplama .....                         | 6                   |
| Şekil 2.2           | Yumuşak hesaplamanın bileşenleri .....                              | 7                   |
| Şekil 2.3           | Bulanıklaştırma-durulaştırma birimli bulanık sistem .....           | 20                  |
| Şekil 2.4           | Bulanık ve ve veya işlemleri için Mamdani tipi çıkarım sistemi..... | 23                  |
| Şekil 2.5           | İleri sürümlü bir sinir ağı modeli.....                             | 29                  |
| Şekil 2.6           | Genetik algoritmaların genel akış şeması.....                       | 32                  |
| Şekil 2.7           | Bir uzman sisteminin blok yapısı .....                              | 37                  |
| Şekil 3.1           | Klasik küme teorisi.....  | 44                  |
| Şekil 3.2           | Bulanık küme teorisi.....   | 44                  |
| Şekil 3.3           | İki bulanık kümenin toplamı .....                                   | 46                  |
| Şekil 3.4(a)        | İki bulanık kümenin birleşimi.....                                  | 49                  |
| Şekil 3.4(b)        | İki bulanık kümenin kesişimi .....                                  | 49                  |
| Şekil 3.5           | En büyük üyelik yöntemi durulaştırması.....                         | 50                  |
| Şekil 3.6           | Ağırlıklı ortalama yöntemi durulaştırması .....                     | 51                  |
| Şekil 3.7           | Ortalama en büyük üyelik yöntemi durulaştırması.....                | 52                  |
| Şekil 3.8           | Toplamların merkezi yöntemi durulaştırması.....                     | 53                  |
| Şekil 4.1           | U ve v' nın üyelik fonksiyonu .....                                 | 63                  |

## ŞEKİLLER DİZİNİ (Devamı)

| <u>Şekil</u>   | <u>Sayfa</u> |
|--|--------------|
| Şekil 4.2 Bağımlı değişkenin tahmin ile gözlem değerlerinin üyeliği.....   | 64           |
| Şekil 4.3 $Y$ ve $\hat{Y}$ arasındaki farklılık dereceleri aynı olan üyelikler.....  | 65           |
| Şekil 4.4(a) $R(x)=L(x)=\max(0.1- x ^s)$ kalıp fonksiyonuna sahip bulanık sayıların üyelik fonksiyonları( $s=1/2$ ) .....  | 83           |
| Şekil 4.4(b) $R(x)=L(x)=\max(0.1- x ^s)$ kalıp fonksiyonuna sahip bulanık sayıların üyelik fonksiyonları( $s=1$ ) .....    | 83           |
| Şekil 4.4(c) $R(x)=L(x)=\max(0.1- x ^s)$ kalıp fonksiyonuna sahip bulanık sayıların üyelik fonksiyonları( $s=2$ ) .....    | 83           |
| Şekil 4.5(a) $R(x)=L(x)=e^{- x ^s}$ kalıp fonksiyonuna sahip bulanık sayıların üyelik fonksiyonları( $s=1/2$ ) .....       | 88           |
| Şekil 4.5(b) $R(x)=L(x)=e^{- x ^s}$ kalıp fonksiyonuna sahip bulanık sayıların üyelik fonksiyonları( $s=1$ ) .....         | 88           |
| Şekil 4.5(c) $R(x)=L(x)=e^{- x ^s}$ kalıp fonksiyonuna sahip bulanık sayıların üyelik fonksiyonları( $s=2$ ) .....         | 88           |
| Şekil 4.6(a) $R(x)=L(x)=1/(1+ x ^s)$ tipi kalıp fonksiyonuna sahip bulanık sayıların üyelik fonksiyonları( $s=1/2$ ) ..... | 90           |



## **ŞEKİLLER DİZİNİ (Devamı)**

| <b><u>Şekil</u></b>  | <b><u>Sayfa</u></b> |
|--|---------------------|
| Şekil 4.6(b) $R(x)=L(x)=1/(1+ x ^s)$ tipi kalıp fonksiyonuna sahip<br>bulanık sayıların üyelik fonksiyonları (s=1) ..... | 90                  |
| Şekil 4.6(c) $R(x)=L(x)=1/(1+ x ^s)$ tipi kalıp fonksiyonuna sahip<br>bulanık sayıların üyelik fonksiyonları (s=2) ..... | 90                  |
| Şekil 5.1 Beş tabakalı bir bulanık sinir ağı yapısı .....  | 102                 |
| Şekil 6.1 Switching regresyon modeli .....   | 110                 |
| Şekil 7.1: FAN Programının Başlangıç Formu.....  | 123                 |



## ÇİZELGELER DİZİNİ

| <b><u>Çizelge</u></b>  | <b><u>Sayfa</u></b> |
|--|---------------------|
| Çizelge 7.1 Önsel parametrelerin varsayılan ilk değerleri.....       | 118                 |
| Çizelge 7.2 Örneklem 1'in uygulama sonuçları .....                   | 119                 |
| Çizelge 7.3 Örneklem 2 'nin uygulama sonuçları .....                 | 120                 |
| Çizelge 7.4 Örneklem 3 'ün uygulama sonuçları .....                  | 121                 |
| Çizelge 7.5 Denemeden sonraki öncül parametre değerleri .....        | 122                 |
| Çizelge 7.6 FAN İçin Denemenin Hesaplama Sonuçları .....             | 122                 |
| Çizelge 7.7: Önsel parametrelerin varsayılan ilk değerleri.....      | 124                 |
| Çizelge 7.8: Örneklemin girdi ve hedef çıktı değerleri.....          | 125                 |
| Çizelge 7.9: İterasyon sonundaki en küçük hata kareler toplamı ..... | 125                 |



## SİMGELER VE KISALTMALAR

### Simgeler

### Açıklama

### Kısaltmalar

### Açıklama

|       |  |
|-------|--|
| FAN   | Fuzzy Adaptive Network (Bulanık uyarlanabilir sinir ağı) |
| CPU   | Bilgisayarın hesaplama zamanı                            |
| WABL  | Seviyelere dayanan ağırlıklandırılmış ortalama gösterim  |
| OWA   | Sıralı ağırlıklı ortalama                                |
| ND1   | Nasibov'un 1. uzaklık tanımı                             |
| ND2   | Nasibov'un 2. uzaklık tanımı                             |
| FRBFN | bulanık radyal tabanlı sinir ağı                         |

## 1.GİRİŞ

Günümüzün en güncel teknolojisi olarak Soft Computing(Yumuşak hesaplama, Esnek hesaplama) Teknolojileri birçok alanda başarıyla kullanılmaktadır.

Bu alanlardan biri de istatistik alanıdır. İstatistiğin en önemli konularından biri olan regresyon analizi, Soft Computing Teknolojileri ele alınarak farklı açılardan araştırılmaktadır. Örneğin; Diamond(1987), Tanaka(1982), Kim ve Bishu(1998), Wang(2000), Nasrabadi(2005), Nasibov vd.(2005) ve daha birçok araştırmacı çalışmalarında bulanık küme teorisini kullanarak regresyon analizini incelemişlerdir.

Tanaka (1982) bulanık bilgi içeren problemlerle ilgili ilk bulanık regresyon analizini ortaya atmıştır. Tanaka' nın lineer programlama tekniği kullanarak oluşturduğu ilk modelinde, girdi ve çıktı değişkenlerinin bulanık olmadığı fakat sistem bilgisinin bulanık olduğu varsayılmaktadır. Ayrıca Tanaka'nın amaç fonksiyonu bağımlı değişkenin tahmin değerinin yayılmasına dayanmaktadır.

Diamond (1987) üçgen bulanık sayılar üzerinde metrik tanımlaması yaparak bulanık en küçük kareler regresyon analizini öne sürmüştür. Diomand'ın amaç fonksiyonu bağımlı değişkenin gözlem ile tahmin değeri arasındaki uzaklığın minimizasyonuna dayanmaktadır. Daha sonra araştırmacılar tarafından gerek Tanaka' nın modeline benzer programlamaya dayalı matematiksel modeller gerekse Diomand'un

modeline benzer bulanık en küçük kareler regresyon modelleri ortaya atılmıştır.

Kim ve Bishu(1998) bağımlı değişkenin tahmin ve gözlem üyelik değerleri arasındaki farkın minimizasyonuna dayanarak bulanık en küçük kareler regresyon fonksiyonunu önermişlerdir.

Wang(2000) 'ın modeli de Tanaka' nın modeline hesaplama karmaşıklığı yönünden benzerdir. Ancak Wang' ın modelinde daha dar bir regresyon aralığı elde edilmektedir.

Nasrabadi (2005) bulanık çıktı ve bulanık olmayan girdi ile bulanık regresyon modellerinin parametrelerini tahminlemek için matematiksel bir programlama modeli geliştirmişlerdir. Model gözlem ve tahmin değerleri arasındaki toplam farklılığın karesini minimize etmeye dayanmaktadır.

Nasibov(2002, 2003, 2005) bulanık sayıların ortalama genişlik ve seviyelere dayanan ağırlıklandırılmış ortalama(WABL) gösterimini kullanarak ND1 ve ND2 olarak adlandırılan iki uzaklık tanımlamış ve ND1' e dayanan bulanık en küçük kareler regresyon analizi ve ND2'ye dayanan bulanık en küçük mutlak sapmalar regresyon modellerini ortaya atmıştır. Ayrıca yeni bir determinasyon katsayısı olan WABL determinasyon katsayısını tanımlamıştır.

Klasik regresyon analizinde kullanılan determinasyon katsayısı bu çalışmada ortaya atılan WABL determinasyon katsayısının özel bir halidir.

Bölüm 2’de Soft Computing (esnek hesaplama) teknolojileri tanımlandı ve esnek hesaplamamanın bileşenlerinden olan bulanık mantık ve bulanık kümeler teorisi, yapay sinir ağları, genetik algoritmalar ve yapay zeka tekniklerine değinildi.

Bölüm 3’de belirsizlik, bulanık küme ve bulanık sayı tanımlandı. Bulanık sayıların toplanması, çıkarılması, çarpılması, bölünmesi ve durulaştırma işlemlerinden en sık kullanılanlarına değinildi.

Bölüm 4’de klasik ve bulanık regresyon arasındaki farklılıklara değinilerek ne gibi durumlarda bulanık regresyon kullanımının uygun olacağı belirtilmiştir. Literatürdeki değişik bulanık uzaklık tanımlamaları ve bu uzaklıklara dayanılarak oluşturulmuş bulanık regresyon yöntemlerine değinildi.

Bölüm 5’de bulanık yapay sinir ağları yaklaşımının bulanık regresyonda kullanımı üzerine Cheng ve Lee (1999) ‘nın FAN (Fuzzy Adaptive Network) bulanık sinir ağı algoritmasına değinilmiştir. Bölüm 6’ da switching regresyonun tanımı yapılarak ne gibi durumlarda klasik ne gibi durumlarda switching regresyonun kullanımının uygun olacağına değinilmiştir.

Bölüm 7’de bölüm 5’de verilen bulanık uyarlanabilir sinir ağı (FAN) algoritmasının programı yapılmıştır.



## 2. SOFT COMPUTİNG TEKNİKLERİ

Son on yılı aşkın bir zamanda birçok metodolojinin birleşimi olan Soft Computing (Esnek Hesaplama, Yumuşak Hesaplama) önemli bir araştırma alanı olmuştur. Esnek hesaplama kavramı akıllı sistemlerin temelini oluşturmaktadır. Gerçek dünya problemlerinin değişik alanlarında kullanılan bütün yapay zeka sistemleri, şimdiye kadar bilgisayarlarla geleneksel hesaplama teknikleri kullanılarak gerçekleştirilmiştir. Ancak son zamanlarda geleneksel hesaplama tekniklerinin etkinlik, belirsizlik ve öğrenme gibi birçok sınırlamalarını ortadan kaldıran yeni “akıllı yöntemler” kullanılmaktadır. Hesaplama teknikleri olarak kullanılan Bulanık Mantık, Yapay Sinir Ağları, Olasılıksal Hesaplama ve Genetik Algoritmalar gibi akıllı yöntemlerin birleşimi “Yumuşak Hesaplama” veya “Esnek Hesaplama” olarak adlandırılmaktadır. Hesaplama temelli akıllı sistemler oluşturmak için yeni bir yaklaşım olan esnek hesaplama, insan bilgi ve tecrübesi ile mantık işleyişini etkin olarak birleştirerek matematiksel modellenmesi zor olan sistemlerde daha iyi bir performans için değişen ortam koşullarına uyum göstermeyi hedefler. Geleneksel hesaplamadan farklı olarak esnek hesaplama ile insanın muhakeme, sezgi ve düşüncelerinin gerçekleştirilebilme üstünlüğü kullanılmaya çalışılmaktadır. Bunun için esnek hesaplamanın amacı; insanın karar verme modelini kullanarak başarılı, basit, gerçekleştirilebilir ve düşük maliyetli çözümlerle belirsizlikleri ortadan kaldırmaktır. (Karay, De Silva, 2004).

Bir sistemin akıllı bir sistem olarak kabul edilmesi için aşağıdaki karakteristik ve yeteneklerin bir veya birkaçına sahip olması gerekmektedir.

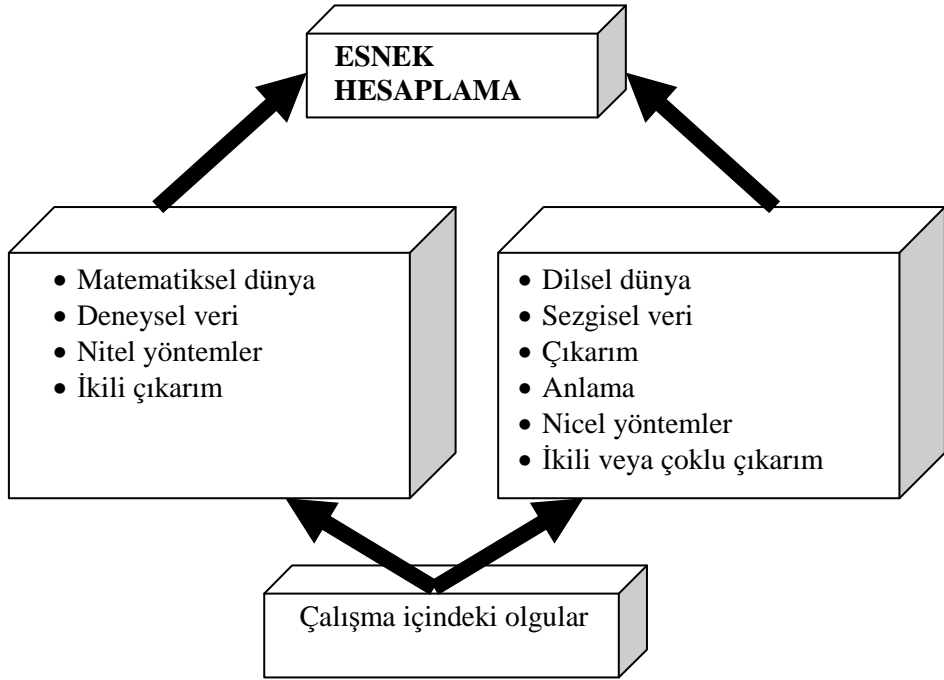
- Sezgisel algı
- Örüntü tanıma
- Öğrenme ve bilgi toplama
- Eksik bilgiden çıkarım yapma
- Nitel veya yaklaşık bilgiden çıkarım yapma
- Bilinmeyen durumlarla ilgilenme yeteneği
- Yeni durumlara uyarlanabilirlik
- Tümevarımsal akıl yürütme
- Ortak algı
- Duyguların gösterimi
- Yeni olaylar oluşturabilme

Yukarıdaki özelliklerin ilk 5 kısmında önemli ilerlemeler yapılmış ve yapılmaktadır. Akıllı makinelerin bütün bu kabiliyetlere özellikle de son üç yeteneğe sahip olması kolay değildir (Jang, 1997).

Esnek hesaplama akıllı ve bilgi tabanlı sistemlerin önemli bir dalını oluşturmaktadır. Esnek hesaplamanın temel özellikleri ;

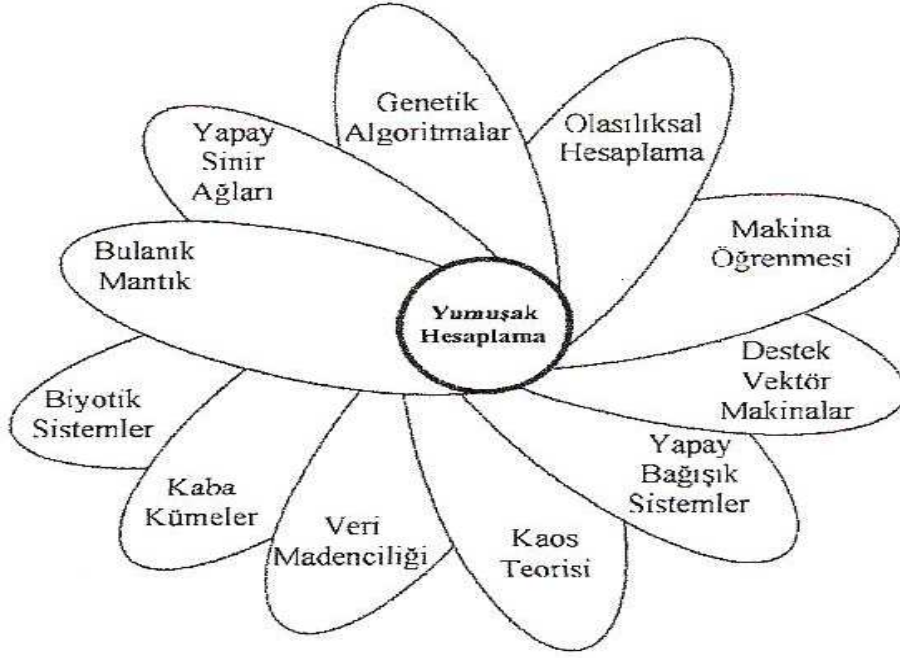
- 1) Akıllı sistemlerin analiz ve tasarımına yöneliktir.
- 2) Belirsizlik ve eksikliğin olduğu bir ortamda karar verebilmek için insan yeteneğinin kullanılabilmesini amaçlamaktadır.
- 3) Esnek hesaplamanın bileşenleri tamamlayıcıdır.
- 4) Esnek hesaplama insanın doğal zekası taklit edilerek oluşturulacak yeni tekniklerin eklenmesine açık bir çatıdır. Esnek hesaplamanın özel bir yönü Şekil 2.1'de görüldüğü gibi matematiksel

çalışma ile dilsel güç arasında bir önemli kullanıcı arabirim olarak yer almasıdır.



**Şekil 2.1:** Bir arabirim olarak esnek hesaplama

Bulanık hesaplama, sinirsel hesaplama, evrimsel hesaplama ve olasılıksal hesaplamanın çekirdek bileşenlerini oluşturduğu esnek hesaplamanın günümüzdeki bileşenleri aşağıdaki şekil 2.2’de gösterilmektedir.



**Şekil 2.2:** Yumuşak hesaplamanın bileşenleri

Sinirsel hesaplama olarak da tanımlanabilen yapay sinir ağları çok sınırlı bir duyarlılıkta biyolojik bilgi işleme mekanizması sağlayabilir. Bulanık mantık, belirsiz ve kesin olamayan bilgi için bir temel sağlar. Evrimsel algoritmalar ise, optimizasyon algoritmaları için kullanılmaktadır.

Burada dikkat edilmesi gereken bir nokta yumuşak hesaplama kavramının her zaman gelişmeye açık bir kavram olduğudur. Yumuşak hesaplama teknikleri kullanılarak akıllı sistemlerin oluşturulmasına yönelik literatürde birçok çalışma bulunmaktadır. Genellikle birçok kaynakta “Soft computing” ya da diğer bir deyişle “Esnek Hesaplama” teknikleri bulanık mantık ve bulanık küme teorisi, yapay sinir ağları, genetik algoritmalar ve yapay zeka başlıkları altında toplanabilmektedir (Karaköse, Akın, 2005).

## 2.1. Bulanık Mantık Ve Bulanık Kümeler Teorisi

Bir parmağın ucundaki bir cetveli dengelemenin ne denli zor olduğu herkes tarafından bilinmektedir. Helikopterin uçurulması da bu tür bir denge kurma yeteneğini gerektirmektedir. Bu beceriyi insanlar zamanla kazanabilirler ama makinelere bunu öğretmek çok zordur. Örneğin bir odanın belirli bir sıcaklıkta tutulması, iklimlendirici pervanesinin devrini artırıp azaltarak gerçekleştirilebilir. Oda sıcaklığı belirli bir değere erişince, pervaneyi çeviren elektrik motoruna elektrik akımı verilir ve motor döner; sıcaklık belirli bir değer altına inince akım kesilir ve motor durur. Sonuçta odanın ısısı istenen sıcaklık civarında bir salınım yapar. Makineler "azıcık", "çok", "eh işte", "kararınca" gibi terimleri anlamazlar. Bu yüzden makinelere belirli büyüklükler tam olarak tanıtılır. Matematiksel olarak makineler için bir parametrenin değeri ya vardır ya da yoktur. Yani makineler, Aristoteles'den bu yana gelen bir mantık ile çalışırlar. Ancak altmışlı yıllarda ilk kez "bulanık mantık" kavramı ortaya atılmıştır. Buna göre var ya da yok yerine, bir miktar vardan, bir miktar da yoktan oluşan bir mantıktan söz edilmiştir. İşte bu yeni mantık ve uzman sistemler yardımı ile makineler, zamanımızda çok farklı bir biçimde çalışmaktadır. Makinelere yüklenen kontrol işlevlerinde insanın düşünme tarzı örnek alınmaya başlanmıştır. İnsanın sinir sistemini örnek alan bu uzman sistemlerin sayesinde de makineler eğitilmiş ve işletmede ortaya çıkabilecek bütün ayrıntılar makinelere tanıtılmıştır. Buna göre çalışan iklimlendirici aygıt oda sıcaklığına göre motor devrini ayarlayabilmektedir. Bu tür kontroller ile nükleer santraller, ütü, çamaşır makinesi, Japonya'daki Sandai kentinin metrosu çalıştırılmaktadır. Metroda bulanık mantık kullanarak % 10 daha az

enerji tüketmenin yanı sıra, trenlerin makinist olmaksızın, durması gereken yerin t3 cm civarında durması da sağlanmaktadır.

Pek çok kültür karmaşık akıl yürütme sistemleri kullanırken, genel olarak birbirinden bağımsız olarak üç kültürde Eski Çin, Eski Hindistan ve Eski Yunan'da akıl yürütme yöntemlerinin açık olarak analiz edildiği kabul edilmektedir. Tam tarihler belirgin olmasa da her üç toplumda da MÖ 4. yüzyılda mantığın ortaya çıktığı düşünülmektedir.

Çin'de Konfüçyüz'ün çağdaşı olan ve mohist okulunun kurucusu kabul edilen Üstad Mo doğru muhakeme ve doğru sonuç çıkarma ile ilgili bazı kuralları değerlendirmiştir. Özellikle Mohizm'den türeyen okullardan biri olan Mantıkçılar biçimsel mantığın ilk bilimsel araştırmalarını yapan bazı bilginlerce kurulmuştur. Daha sonra Kin hanedanını takiben bu araştırma çizgisi Çin'de Budistlerce Hint felsefesinin sokuluşuna kadar unutulmuştur. Ayrıca Taoizm'de geçen Tao terimi, sonsuz, bedensiz, cisimsel olmayan şey, akıl, akıl yolu, ilke, düzen gibi anlamları da içerir. Bu anlamlarıyla Tao mantık teriminin kökenindeki Logos kavramı ile örtüşür.

Aristoteles'ten (MÖ 322-MÖ 294) iki yüz yıl kadar önce Gautama Buddha (Siddharta) dünyadaki her şeyin aynı zamanda biraz doğru, biraz yanlış olabileceği çelişkisi ile dolu olduğunu fark etmiştir. Bununla beraber doğru ve yanlış zıtlarının ötesinde bir üçüncü bölgenin varlığını gösteren, bulanık mantığın ne olması gerektiğini kurmayı bekleyen Platon olmuştur.

Eski Yunan'da Organon olarak bilinen Aristoteles'in çalışmaları mantık olarak daha önce olmayan bir disiplinin doğuşunu sağlamıştır.

Elealı Zenon (MÖ 490-430) akıl bilgisine ve kavramsal düşünceye dayanarak aklın ilkelerini kaba bir biçimde ortaya koymuştur. Parmenides (MÖ 500'ler), kesin ve zorunlu varlık bilgisini, değişmeyen, oluş halinde bulunmayan, hep sabit kalanın bilgisi olarak tanımlamıştır. Zenon'da hocası Parmenides'in tezlerini ünlü paradoksları ile desteklemiştir. Zenon bu paradoksları geliştirirken mantıkta dolaylı kanıtlama denilen bir kanıtlama türünü de geliştirmiş oluyordu.

Duyuların göreceliğine karşın aklın evrensel ve tümeli vereceğini öne süren sofistlere karşı bireyselliği ve göreceliği reddeden Socrates, (MÖ 469-MÖ 399), bilgiye ve aklın ortaya koyduğu zihinsel, soyut, kavramsal, tümel bilgiye önem vermiştir. Socrates ayrıca düşünce tarihinde kavram konusunu ilk defa açıkça ele alıp tanımlayan filozof olmuştur. Socrates tekil örnekler arasındaki ortak özelliklerden hareketle kavramları tanımlamak ile tümevarımsal mantığında geliştiricileri arasında yer almıştır.

Platon (MÖ 427-MÖ 347)'a göre ise bir kavramın tekilliğine asla tümevarımsal bir yol izlenerek varılamaz. Deney dünyası tekil ve duyuusal örnekler dünyasıdır.

Aristoteles mantığında temel buluş kıyaslardır. Aristoteles kıyaslarla elde edilen çıkarımın içeriğinden bağımsız bir elde yöntemi olduğunu, akıl yürütmenin kendi başına bir biçimi olduğunu, bu biçimin sözel ifadelere tam olarak uymadığını ve akıl yürütmenin geçerliliğinin sözel biçimler tarafından değil, düşünme biçimleri tarafından belirlendiğini görmüş ve böylece dil ile düşünme arasındaki farkı da ortaya koymuştur.

Modern matematiğin başarısı Aristoteles ve öncesindeki matematikçilere bağlıdır. Onun çalışmalarında sonradan gelen

matematikçilerce düşüncenin kanunu olarak da tanımlanan mantığın teorisinin az ve öz bir şekilde ifade edildiği keşfedilmiştir. Bunlardan birisi olan üçüncünün olmazlığı bir şeyin ya doğru ya da yanlış olabileceğini, ya A ya da A-değil olabileceğini, bu ya da bu-değil olabileceğini, ifade etmektedir. Her hesaplama ya da cümle doğru ya da yanlıştır veya 1 veya 0 doğruluk derecesine sahiptir. Bu önermeye oldukça güçlü itirazlar olmuştur. Çoklu mantık içinde sadece bir tanesini geçerli kılmanın sonucunda Aristoteles tarafından sistematik bir biçimde tanımlanarak ileri sürülen Aristoteles mantığı, Klasik mantık, İkili mantık, siyah-beyaz mantığı olarak ta adlandırılabilen mantık ortaya çıkmaktadır.

Ortaçağda hristyanlığın etkisi ile Aristoteles mantığı, bilimde şekil ve öz arasındaki ilişki gerçekliğin kavramsal temsili olarak ön plana atılarak stokastik felsefe kurulmuştur. St. Augustine (354-430) bilgi, insan, ahlak, zaman ve tarih konularındaki çalışmaları ile ön plana çıkmıştır. Bu dönemin otoritesi olan St. Augustine'in bilgi konusundaki ilgisi diğer kilise otoriteleri tarafından paylaşılmamıştır. Severinus Boethius (470-525) Aristoteles ve Porphyry'nin çalışmalarını tercüme etmiş ve yorumlamalarıyla beraber Batı Avrupa'da bu 6-13. yüzyıllarda eğitimde temel kaynaklardan biri olmuştur. Bu dönemde gramer, diyalektik (yani mantık), retorik, geometri, aritmetik, astronomi ve müzik olarak yüksek eğitimde verilen yedi özgür sanattan biridir. Bu dönemde Aristoteles ve onun eserlerini yorumlayan Farabi (870-950), İbn-i Sina (980-1037), İbn-i Rüşd (1126-1189), İslam dünyasında meşşai okulu denilen rasyonalist felsefenin mensuplarındandırlar. Bunların çalışmaları ile Aristoteles'in eserleri latinceye tercüme edilmiş ve felsefi çalışmalara karşı büyük bir ilgi başlatılmıştır. Ortaçağ skolastikleri bu eserler sayesinde mantık bilimini sistematik bir biçimde kurgulamışlardır. 12-14. yüzyıllarda Batı'daki mantık çalışmaları içerisinde Albertus Magnus



(1193-1280), Aquinolu Thomas (1225-1274), Duns Scotus (1265-1308) ve Ockhamlı William (1285-1347) klasik mantığın rönesansa kadar uzanmasını sağlayan Ortaçağ Batı düşünürlerindedir. 15-16. yüzyıllar skolastisizm'in yıkılışına şahitlik eder ve 1620'de Aristoteles mantığının pek çok temel kavramına Francis Bacon (1561-1626) Novum Organum adlı eseriyle saldırır.

Erken modern dönem (1600-1850)'de büyük mantıkçı Gottfried Wilhelm von Leibnitz (1646-1716), çalışmalarında büyük oranda Aristoteles mantığına bağlı kalmış ve düzenlemelerin genel teorisi, ideal dil planlamaları ve genel bilim yöntemi üzerine de çalışmıştır. Leibnitz mantığın önermelerin içeriğinden tamamen bağımsız çıkarımlar yapılmasını mümkün kılan evrensel bir dil haline getirilmesi konusunda çalışmalar yapmış ve sembolik mantığı ilk defa fikir olarak ortaya atmıştır. Alman filozof Immanuel Kant (1724-1804), felsefe ve mantık konularında derin etkileri olan çalışmalar yapmıştır. Kant, Hume'un nedensellik ve bilinçten bağımsız bir dünyanın bilinemezliği tezlerinden etkilenmiştir. Kant'a göre; tüm bilgiler iki kaynaktan doğar. Bunlar dış gerçeklik ve öznel bilinçtir. Bilincin anlayış ve kavrama gücü doğuştan getirilmiş uzay, zaman, nedensellik gibi bazı yargı formlarını da kullanarak duyu verilerini sınırlayıp düzenleyerek belli bir biçime sokar. Kant'ın bir de görelilik anlayışı vardır. Bu sofistlerin görecelik anlayışından farklı olarak insan dünyasında genel-geçer, kesin bilgilerin bulunduğunu kabul eder.

Modern dönem(1850'den sonrası) mantık üzerinde büyük çalışmaların yapıldığı dönemlerdir. Bu dönem mantığın sembolik bir şekilde ifadesini mümkün kılan çalışmalarla doludur. Augustus De Morgan (1806-1871) hala kullanılan ve adını taşıyan teorileri bulmuştur.George Boole

(1815-1864) sembolik mantığın kurucusudur. Mantıktaki ilk büyük birleştirme teorisini Boole, 1847'de Mantığın Matematiksel Analizi ve 1854'de Düşüncenin Kanunları adlı kitaplarında gerçek fonksiyonel mantık ile aritmetiği birleştirerek gerçekleştirmiştir. Böylece Boolean cebiri de denilen, gerçeklik değerlerini temsil etmek üzere sayıların kullanıldığı bir aritmetik ortaya çıkmıştır. İngiliz mantıkçı John Venn (1834-1923), kıyas değerlendirmede bir araç olarak dairesel şekillerin kullanımı yöntemini geliştirmiştir.

1900'lerin ilk yıllarında, Jan Lukasiewicz (1878-1956) iki değerli Aristo mantığına karşı bir öneride bulunmuştur. Açıkladığı üç değerli mantık en uygun olarak "belki" tanımı ile tercüme edilebilir ve doğru ile yanlış arasında bir değere sahiptir. Donald Erwin Knuth, Lukasiewicz'den aldığı üç değerli mantığı,  $[0, 1, 2]$  tam sayı aralığı yerine  $[-1, 0, 1]$  aralığını kullanarak ifade etmiştir. Bu alternatif yaygınlaşmamış, önemsenmemiş ve karanlıkta kalmıştır. Ayrıca Arend Heyting (1898-1980), de çok değerli mantığı genişletmiştir. Heyting sezgisel mantığın da kurucusudur (Baykal ve Beyan, 2004).

Modern anlamda bulanık mantığın kurucusu sayılan Zadeh, öznel ve bulanık insan düşüncesini temsil eden veriyi işlemede alışlageldik bilgisayar mantığının yetersiz kaldığını gözlemlemiş ve 1965'de bulanık kümeler çalışmasını yayınlamıştır. Teorisi üyelik fonksiyonunu  $[0.0, 0.1]$  gerçek sayı aralığında tanımlamıştır. Böylece bulanık mantık tüm dünyaya açıklanmıştır.

Bulanık mantık (Fuzzy Logic) kavramı ilk kez 1965 yılında California Berkeley Üniversitesinden Prof. Lotfi A.Zadeh'in bu konu

üzerinde ilk makalelerini yayınlamasıyla duyulmuştur. Bu tarihten sonra önemi gittikçe artarak günümüze kadar gelen bulanık mantık, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilmesi için kurulmuş katı bir matematik düzen olarak tanımlanabilmektedir. Bilindiği gibi istatistikte ve olasılık kuramında, belirsizliklerle değil kesinliklerle çalışılır ama insanın yaşadığı ortam daha çok belirsizliklerle doludur. Bu yüzden insanoğlunun sonuç çıkarabilme yeteneğini anlayabilmek için belirsizliklerle çalışmak gereklidir.

Bulanık mantık ile klasik mantık arasındaki temel fark bilinen anlamda matematiğin sadece aşırı uç değerlerine izin vermesidir. Klasik matematiksel yöntemlerle karmaşık sistemleri modellemek ve kontrol etmek işte bu yüzden zordur, çünkü veriler tam olmalıdır. Bulanık mantık kişiyi bu zorunluluktan kurtarmakta ve daha niteliksel bir tanımlama olanağı sağlamaktadır. Bir kişi için 38, 5 yaşında demektense sadece orta yaşlı demek birçok uygulama için yeterli bir veri olmaktadır. Böylece azımsanamayacak ölçüde bir bilgi indirgenmesi söz konusu olacak ve matematiksel bir tanımlama yerine daha kolay anlaşılabilen niteliksel bir tanımlama yapılabilecektir. Bulanık mantıkta Fuzzy (bulanık) kümeleri kadar önemli bir diğer kavramda linguistik değişken kavramıdır. Linguistik değişken sıcak veya soğuk gibi kelimeler ve ifadelerle tanımlanabilen değişkenlerdir. Bir linguistik değişkenin değerleri Fuzzy (bulanık) kümeleri ile ifade edilmektedir. Örneğin oda sıcaklığı linguistik değişken için; sıcak, soğuk ve çok sıcak ifadelerini alabilmektedir. Bu üç ifadenin her biri ayrı ayrı Fuzzy (bulanık) kümeleri ile modellenmektedir. Bulanık mantığın uygulama alanları çok geniştir. Sağladığı en büyük fayda ise insana özgü tecrübe ile öğrenme olayının kolayca modellenebilmesi ve belirsiz kavramların bile matematiksel olarak ifade edilebilmesine olanak

tanımasıdır. Bu nedenle lineer olmayan sistemlere yaklaşım yapabilmek için özellikle uygundur.

Bulanık mantık dar anlamda klasik iki değerli mantığın genelleştirilmiş halidir. Geniş anlamda ise, bulanık kümeleri kullanan bütün teorileri ve teknolojileri ifade eder. Bulanık mantığın ardındaki temel fikir, bir önermenin doğruluğunun, önermelerle, kesin yanlış ve kesin doğru arasındaki sonsuz sayıda doğruluk değerlerini içeren bir kümedeki değerler, ya da sayısal olarak  $[0, 1]$  gerçel sayı aralığı ile ilişkilendiren bir fonksiyon olarak kabulüdür. Bu, Zadeh'in bulanık kümeler üzerindeki ilk çalışmasının bir sonucudur. Bulanık mantık yaklaşık akıl yürütmeye dayanmaktadır.

Bulanık mantığın geçerliliği kesin değil, fakat yaklaşık olan çıkarım kurallarına sahip oluşu ayırt edici özellikleridir. Bulanık mantığı diğer mantık sistemlerinden ayıran önemli özelliklerinden birisi, üçüncünün olmazlığı ilkesi ve çelişmezlik ilkesi olarak adlandırılan ve diğer mantık sistemleri için oldukça önemli olan, hatta temel kural denebilecek iki özelliğin, bulanık mantık için geçerli olmamasıdır. Bulanık mantıkta bir önerme hem doğru hem yanlış olamaz diye bir şey yoktur. Bu durum doğruluğun çok değerli oluşundan ve bu çerçevede bağlaçlarına yüklenen anlamdan kaynaklanmaktadır (Baykal ve Beyan, 2004).

Bulanık mantığın en geçerli olduğu iki durumdan ilki, incelenen olayın çok karmaşık olması ve bununla ilgili yeterli bilginin bulunmaması durumunda kişilerin görüş ve değer yargılarına değer verilmesi, ikincisi ise insan kavrayış ve yargısına gerek duyan hallerdir.

Bulanık mantık ile günlük konuşma dilinde geçen sözel belirsizlikleri modelleme ve hesap yapılırken işin içine katma imkanı

bulunabilir. İnsanlar sözel verilerle konuşarak anlaşır. Bulanık sistemlerin asıl değerlendirileceği alan bu tür bilgilerin bulunması halinde çözümlenmelere gitmek için nasıl düşünüleceğidir. Bulanık mantıkla herhangi bir problemin yaklaşık olarak modellenmesi ve matematiksel olarak karmaşık olmayacak çözümlerle denetim altına alınmasına çalışılmaktadır.

Bulanık mantık, (fuzzy logic), adından anlaşılabilir gibi mantık kurallarının esnek ve bulanık bir şekilde uygulanmasıdır. Klasik mantıkta, "doğru" ve "yanlış" ya da "1" ve "0"lar vardır, oysa bulanık mantıkta, ikisinin arasında bir yerde olan önermeler ve ifadeler izin verilebilir ki, gerçek hayata bakıldığında hemen hemen hiçbir şey kesinlikle doğru veya kesinlikle yanlış değildir. Gerçek hayatta önermeler genelde kısmen doğru veya belli bir olasılıkla doğru şeklinde değerlendirilir. Bulanık mantığa da zaten klasik mantığın gerçek dünya problemleri için yeterli olmadığı durumlar dolayısıyla ihtiyaç duyulmuştur.

Bulanık mantık sisteminde, bir ifade tamamen yanlış ise klasik mantıkta olduğu gibi 0 değerindedir, yok eğer tamamen doğru ise 1 değerindedir. Bunların dışında tüm ifadeler 0'dan büyük 1 den küçük reel değerler alırlar. Yani değeri 0.32 olan bir ifadenin anlamı %32 doğru %68 yanlış demektir.

Bulanık mantığın da klasik mantıkta olduğu gibi operatörleri vardır, örneğin and, or, not vb. And işlemi, genelde çarpma olarak ifade edilmektedir veya not işlemi de birden çıkarma şeklinde ifade edilmektedir. Bunlar;

$$\text{AND: } A=0.2 \quad B=0.8 \Rightarrow A \text{ and } B = (A)*(B)=0.2*0.8=0.16 \quad (2.1.1)$$

$$\text{NOT: } A=0.4 \Rightarrow \text{not } A=1-(A)=1-0.4=0.6 \quad (2.1.2)$$

şeklinde örneklenebilir. Ancak bunlar en basit yaklaşımlardır.

Günümüzde bulanık mantık konusunda yapılan araştırmalar Japonya'da oldukça fazladır. Özellikle fuzzy process controller olarak isimlendirilen özel amaçlı bulanık mantık mikroişlemci yongasının üretilmesine çalışılmaktadır. Bu teknoloji fotoğraf makineleri, çamaşır makineleri, klimalar ve otomatik iletim hatları gibi uygulamalarda kullanılmaktadır. Bundan başka bulanık mantık, uzay araştırmaları ve havacılık endüstrisinde de kullanılmaktadır.

Bulanık mantığın tercih edilmesinin nedenleri arasında,

- 1) Bulanık mantığın anlaşılması kolaydır.
- 2) Bulanık mantığın dayandığı matematiksel teori basittir.
- 3) Bulanık mantığı çekici kılan şey yaklaşımının doğallığı ve kompleks ya da karmaşıklıktan uzak olmasıdır.
- 4) Bulanık mantık esnektir.
- 5) Bulanık mantıkta eksik ya da yetersiz verilerle işlemler yapılabilmektedir.
- 6) Bulanık mantık karmaşık lineer olmayan fonksiyonları modelleyebilmektedir.
- 7) Bulanık mantık sıradan insanların günlük işlerinde kullandığı dili kullanmaktadır. Bu da bulanık mantığın en büyük avantajlarından biridir (Yılmaz ve Arslan, 2005).

Küme kuramı aritmetiğin ve mantığın temellerini oluşturur, hatta matematik ve biçimsel nedenselleştirmenin en önemli kısmını oluşturduğu

söylenbilir. Temel küme işlemleri ile temel mantıksal işlemler arasında sıkı bir bağ olduğu bilinmektedir. Diğer matematiksel teorilerden farklı olarak, küme teorisi kendisinden önce bazı fikirler ileri sürülmüşse de, George Cantor'un çalışmalarıyla ortaya atılmıştır. Cantor çalışmalarında küme kavramı yerine birleşim (aggregate) terimini kullanarak kümeyi; sezgi ya da düşüncelerin bir ürünü olarak belirli ve iyi tanımlanmış nesnelerin oluşturduğu topluluk olarak tanımlamıştır.

Bulanık küme kavramı, 1960'ların ortasında Zadeh'in, klasik sistemin özellikle insanları içeren karmaşık sistemlerde yetersiz kalmasından dolayı, niteliklerin ikili üyelik fonksiyonuyla ifade edildiği klasik kümeler yerine, dereceli üyelik fonksiyonuyla ifade edildiği bulanık kümeler tanımlamasını önermiştir.

Bulanık kuramının merkez kavramı bulanık kümelerdir. Küme kavramı kulağa biraz matematiksel gelebilir ama anlaşılması kolaydır. Örneğin orta yaş kavramını inceleyecek olursak, bu kavramın sınırlarının kişiden kişiye değişiklik gösterdiğini görürüz. Kesin sınırlar söz konusu olmadığı için kavramı matematiksel olarak da kolayca formüle edemeyiz. Ama genel olarak 35 ile 55 yaşları orta yaşlılık sınırları olarak düşünülebilir. Bulanık bir küme kendi aitlik fonksiyonu ile açık olarak temsil edilebilir. Aitlik fonksiyonu 0 ile 1 arasındaki her değeri alabilmektedir. Böyle bir aitlik fonksiyonu ile kesinlikle ait veya kesinlikle ait değil arasında istenilen incelikte ayarlama yapmak mümkün olabilmektedir.

Bulanık modeller oluştururken değişik formlarda üyelik fonksiyonları seçilebilir. Yaygın olarak kullanılan üyelik fonksiyonları olarak üçgen, yamuk, Gauss eğrisi, sigmoid fonksiyonu vb.

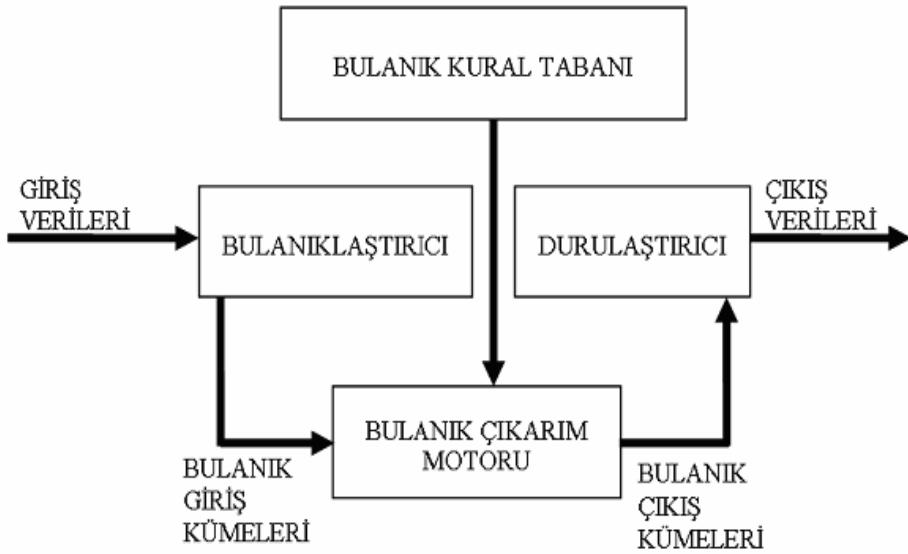
kullanılmaktadır. Bulanık sistemler genel olarak, mevcut verilerden seçilen girdi değişkenlerinden çıktı değişkenlerinin elde edilmesini sağlamak amacıyla bulanık küme ilkelerini kullanan sistemlerdir. Bulanık sistemlerin en büyük avantajı insan deneyimlerinin ve sözel verilerin bulanık modele katılması ile çözüme ulaşılmasıdır. Bulanık model (Bulanık Çıkarım Sistemi), bulanık Eğer-İse kuralları adı verilen bulanık kurallara dayanan sistemlerdir. Bulanık modelin temeli, bulanık Eğer-İse kurallarından anlaşılacağı gibi öncül ve soncul kısımlardan oluşmaktadır. Öncül kısımda sonuca sebep olan giriş değişkenleri ve bunlar arasındaki mantıksal ilişkiler, sonuç kısmında ise, bu giriş değişkenlerine bağlı olarak ortaya çıkan sonuç değişkenleri yer alır. Genel olarak bulanık kurallar aşağıdaki formdadır;

$$\text{Kural 1: Eğer } x = A_1 \text{ ve } y = B_1 \text{ ise } z = N_1, \quad (2.1.3)$$

$$\text{Kural 2: Eğer } x = A_2 \text{ ve } y = B_2 \text{ ise } z = N_2. \quad (2.1.4)$$

Burada  $x$  ve  $y$  öncül kısımdaki girdi değişkenlerince tanımlanan koşulları  $z$  ise soncul kısımdaki çıktı değişkenlerince tanımlanan sonuçları ifade eder. Şekil 2.3' de genel bir bulanık model sistemin yapısı gösterilmektedir.





**Şekil 2.3:** Bulanıklaştırma – durulaştırma birimli bulanık sistem

1) Genel Bilgi Tabanı Birimi: İncelenecek olayın etkilendiği girdi değişkenlerini ve bunlar hakkındaki tüm bilgileri içermektedir. Genel veri tabanı denmesinin sebebi buradaki bilgilerin sayısal ve/veya sözel olabilmesidir.

2) Bulanıklaştırıcı: Sayısal girdi değerlerini sözel olarak nitelendirilmiş bulanık kümelerdeki üyelik derecelerine atayan bir işlemci olarak tanımlanmaktadır.

3) Bulanık Kural Tabanı Birimi: Veri tabanındaki girişleri çıkış değişkenlerine bağlayan mantıksal Eğer-İse türünde yazılabilen kuralların tümünü içermektedir. Bu kuralların yazılmasında sadece girdi verileri ile çıktılar arasında olabilecek tüm bulanık küme bağlantıları düşünülmektedir. Böylece her bir kural girdi uzayının bir parçasını çıktı uzayına mantıksal

olarak bağlamaktadır. İşte bu bağlamların tümü kural tabanını oluşturmaktadır.

4) Bulanık Çıkarım Motoru Birimi: Bulanık kural tabanında giriş ve çıkış bulanık kümeleri arasında kurulmuş olan parça ilişkilerinin hepsini bir arada toplayarak sistemin bir çıkışlı davranmasını temin eden işlemler topluluğunu içeren bir mekanizma olarak tanımlanabilmektedir. Bu motor, her bir kuralın çıkarımlarını bir araya toplayarak tüm sistemin girdiler altında nasıl bir çıktı vereceğinin belirlenmesini sağlamaktadır.

5) Durulaştırıcı: Bulanık işlemler sonucu elde edilen bulanık çıkarım sonuçlarını keskin sayısal çıkış değerlerine dönüştürmektedir.

6) Çıktı Birimi: Bilgi ve bulanık kural tabanlarının bulanık çıkarım motoru vasıtasıyla etkileşimi sonunda elde edilen çıktı değerlerinin topluluğunu belirtmektedir (Yılmaz ve Arslan, 2005) .

Bulanık mantık ile yaygın olarak kullanılan başlıca modellemeler, Mamdani ve Takagi-Sugeno tipi bulanık modellemelerdir.

Mamdani tipi bulanık modelin oluşturulması kolay olup, insan davranışlarına çok uygun olmaktadır. Bu nedenle çok yaygın bir kullanıma sahiptir ve diğer bulanık mantık modellerin temelini oluşturmaktadır. İlk defa bir buhar motorunun insan tecrübelerinden elde edilen sözel kontrol kuralları yardımıyla kontrolü amacıyla kullanılmıştır (Mamdani and Assilian, 1975).

Bu modelde hem girdi değişkenleri hem de çıktı değişkeni kapalı formdaki üyelik fonksiyonları ile ifade edilmektedir (Akyılmaz, 2005).

a) Girdilerin bulanıklaştırılması: Öncül kısımdaki bütün bulanık ifadeleri kullanarak girdi değişkenlerine ait 0 ile 1 arasında değişen üyelik derecelerinin belirlenmesi.

b) Bulanık mantık işlemlerini kullanarak kural ağırlıklarının belirlenmesi.

c) Bulanık küme mantıksal işlemcilerin (ve, veya) uygulanması.

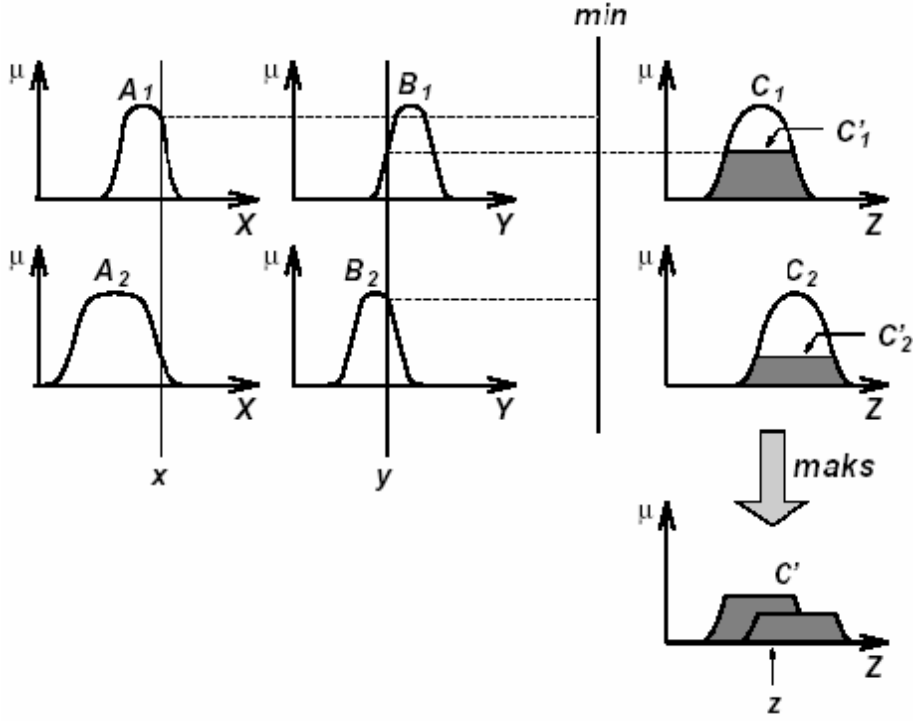
d) Sonuçların toplanması: Her bir kuralın çıktısını temsil eden bulanık kümelerin birleştirilmesi.

e) Durulaştırma: Tek bir sayıya dönüştürülmüş toplam bulanık küme sonuçlarının durulaştırılması.

Şekil 2.4'de x ve y gibi iki sayısal değişkenden oluşan iki kurallı bir Mamdani tipi bulanık modelde z çıkış değerinin  $c_i$  bulanık küme fonksiyonlarından nasıl hesaplandığı gösterilmektedir.

$$\text{Kural 1: Eğer } x=A_1 \text{ VE } y=B_1 \text{ ise } z=C_1 \quad (2.1.5)$$

$$\text{Kural 2: Eğer } x=A_2 \text{ VE } y=B_2 \text{ ise } z=C_2 \quad (2.1.6)$$



**Şekil 2.4:** Bulanık VE ve VEYA işlemleri için sırasıyla minimizasyon ve maksimizasyon operatörlerini kullanan Mamdani tipi bulanık çıkarım sistemi (Akyılmaz, 2005).

Mamdani bulanık modelinin, oluşturulmasının basit olması, diğer bulanık mantık modellerin temelini oluşturması ve insan davranış ve duyularına uygun olması gibi avantajları olduğu söylenebilmektedir.

Takagi-Sugeno bulanık mantık modellemesi, Mamdani bulanık mantık modellemesinin özel bir hali olarak tanımlanabilmektedir. Girdi değişkenlerinin bulanıklaştırılması ve bulanık mantık işlemleri Mamdani modeli ile tamamen aynıdır ancak iki modelleme arasında çıktı üyelik fonksiyonlarında fark bulunmaktadır. Sugeno tipi bulanık modellemede çıktı üyelik fonksiyonları sadece lineer ya da sabittir. Çıktı üyelik fonksiyonları sabit olduğu zaman, 0.derece, 1.derece doğru denklemi

şeklinde olduğu zaman ise birinci derece Sugeno bulanık model olarak adlandırılmaktadır. Sugeno tipi bulanık model Mamdani tipi bulanık modelden daha karmaşık ve gösterim açısından daha elverişli olduğu söylenmektedir.

Birinci derece Sugeno bulanık model,

$$\text{Eğer } x=A \text{ ve } y=B, \text{ ise } z=f(x, y)=px+qy+r(c) \quad (2.1.7)$$

şeklinde yazılabilir. Burada A ve B, x ve y üyelik fonksiyonları için tanımlanmış öncül kısımdaki bulanık kümeler, p, q, r(c) ise soncul parametrelerdir. Böylece her bir kural için bir çıktı değeri elde edilmektedir. Bulanık küme mantıksal işlemleri (ve, veya) basit toplama ve çarpma işlemleridir.

Sugeno tipi bulanık modelin avantajları ve dezavantajları bulunmaktadır. Hesaplama işlemlerinin rahat yapılabilmesi, lineer olmayan sistemlerin kontrol edilmesi için lineer tekniklerin kullanılabilmesi, Optimizasyon ve uyarlanabilir tekniklerle birlikte iyi çalışarak çıktı parametrelerini optimize ederek sonuçları iyileştirmesi gibi avantajları bulunmaktadır. Buna karşın, Yüksek derecedeki tipi kullanıldığında oldukça karmaşık bir yapıya sahip olması, insan sezgilerine çok uygun olmaması dezavantajlarındandır. Girdi ve alt küme sayılarının artması verilerin eğitilmesini zorlaştıracığından sonuçların elde edilmesi için belirlenmesi gereken soncul parametrelerin sayısının artması da bir dezavantajdır (Yılmaz ve Arslan, 2005) .

## 2.2. Yapay Sinir Ağları (YSA)

Yapay Sinir Ağı kavramı, beynin çalışma ilkelerinin sayısal bilgisayarlar üzerinde taklit edilmesi fikri ile ortaya atılmış ve ilk çalışmalar beyni oluşturan biyolojik hücrelerin ya da literatürdeki ismiyle nöronların matematiksel olarak modellenmesi üzerinde yaygınlaşmıştır. Yapay sinir ağlarının kullanıldığı önemli alanlardan biriside geleceği tahmindir. Yapay sinir ağları (YSA), veriler arasındaki bilinmeyen ve fark edilmesi güç ilişkileri ortaya çıkarabilmektedir. YSA, doğrusal olmadıkları için doğrusal modellerin uygulanamayacağı doğrusal olmayan durumlarda daha avantajlı olabilmektedir (Zang, 1998).

Yapay sinir ağları; girdi ve çıktı değişkenleri arasındaki herhangi bir ön bilgiye ihtiyaç duymadan, herhangi bir varsayımda bulunmadan, doğrusal olmayan modellemeyi sağlayabilmektedir (Kaastra, 1996).

Ağa, girdi bilgileri ve bu girdilere karşılık gelen çıktı bilgileri verilmekte ve ağın girdi-çıkıtı arasındaki ilişkiyi öğrenmesi sağlanmakta, böylece ağın eğitimi gerçekleştirilmektedir. Öğreticili Öğrenme denilen bu yöntem genelde tercih edilen bir yöntemdir (Haykin, 1999).

Sinir ağlarını çekici kılan özellikler,

1)Sistemin paralelliği ve toplumsal işlevin yapısal olarak dağılımıdır. Diğer bir deyişle birçok nöron eş zamanlı olarak çalışır ve karmaşık bir işlev çok sayıda küçük nöron aktivitesinin bir araya gelmesinden oluşur.

2)Genelleme yeteneđi, diđer bir deyişle ađ yapısının, eđitim esnasında kullanılan nümerik bilgilerden eşleştirmeyi betimleyen kaba özellikleri çıkarsaması ve böylelikle eđitim sırasında kullanılmayan girdiler içinde anlamlı yanıtlar üretilebilmesidir.

3)Ađ fonksiyonunun nonlineer oluşudur. Yapı üzerinde dağılmış belli tipteki nonlineer alt birimler özellikle, istenen eşleştirmenin denetim ya da tanılama işlemlerinde olduğu gibi nonlineer olması durumunda işlevin doğru biçimde yerine getirilebilmesini matematiksel olarak olası kılarlar. Burada işlemin doğru olması için yapısal bir esneklik gereklidir ve bu da ađ parametrelerinin başarımı arttıracak ya da maliyeti azaltacak şekilde deđiştirilmesiyle sağlanır.

4)Sayısal ortamda tasarlanan sinir ađı yaklaşımlarının tüm devre gerçekleştirilebilirliklerinin olmasıdır.

Literatürde öğrenme stratejisi iki tanedir. Bunlar “Öđreticili Ve Öđreticisiz Öğrenme” olarak adlandırılmaktadır. İki yaklaşım arasındaki temel fark; istenen çıkış deđerlerinin mevcut olup olmamasıdır. Eđer bir eđitici, sistem çıkışlarının istenen deđerlerini temin ediyorsa, bu tip öğrenme birinci grupta yer alır ve “Öđreticili Öğrenme” olarak bilinmektedir. Tasarım koşulları istenen deđerlerin temin edilmesine müsaade etmiyorsa bu tip öğrenme ikinci grupta yer alır ve “Öđreticisiz Öğrenme ” olarak bilinmektedir. Öđreticisiz öğrenme algoritmaları daha çok, sistemin geçmişte karşı karşıya kaldığı veri kümesinin içerdiği istatistiksel bilgilerin çıkarsanmasını amaçlar. Böylelikle çok elemanlı veri kümeleri içerisinde deneyim yoluyla bilgi genelleştirilmesi yapılabilir.

Eğer ağ üzerinde bilgi akışı sürekli ileri doğru ise bu yapıya sahip ağ modelleri “İleri Sürümlü Ağ” olarak adlandırılmaktadır. Ağ yapısında geri besleme bağlantıları varsa bu tipteki sistemlere “Geri Beslemeli Ağ” adı verilmektedir.

Öğrenme algoritmaları, öğrenme denen olguyu matematiğin kuralları ile ölçülebilir büyüklüklere dönüştürerek bir başarımlı ölçütünün oluşturulmasına ve bu ölçütün zaman içerisinde artırılmasını ya da bir maliyet ölçüsünün oluşturulmasına ve bu ölçütün zaman içerisinde azaltılmasını sağlayacak parametre değişikliklerinin hesaplanmasına dayanırlar. Burada önemli bir nokta parametre güncelleme işleminin zamanlamasıdır. Öğreticili öğrenme yaklaşımında parametre güncelleme işlemi, normal çalışma esnasında, anlık gözlemlerden elde edilen bilgi ile yapılıyorsa bu “Eş Zamanlı Öğrenme ” olarak adlandırılmaktadır. Eğer sinir ağı daha önceden belirlenen bir giriş /çıkış eşleştirmesini gerçekleştirmeye çalışıyorsa bu da “Zamandan Bağımsız Öğrenme” olarak adlandırılmaktadır.

Parametre güncelleme işlemi için literatürde sıklıkla iki yöntemden bahsedilmektedir.

1)Toplumsal Güncelleme: Ağ parametrelerinin, eğitim çiftlerinin tamamının ağ üzerinden geçirilip her bir geçişte hesaplanan değişim miktarlarının toplamı ile güncellenmesi olarak açıklanabilmektedir.

2)Bireysel Güncelleme: Her bir eğitim çifti için hesaplanan değişim miktarının o anda uygulanması olarak açıklanabilir.

Geri yayılım ağları (Back Propagation Networks-BPN) sıklıkla kullanılan bir ağ yapısıdır. Standart geri yayılım algoritması, ağ



ağırlıklarının, performans fonksiyonunun negatif gradyanı yönünde ilerlediği gradyan iniş algoritmasıdır. Birçok çeşidi olan geri yayılım algoritması, gradyan iniş ve Newton metodu gibi standart optimizasyon tekniklerine dayanmaktadır.

Geri yayılım algoritmasının tanınması, birbirinden bağımsız olarak ilk defa Werbos ve daha sonra Rumelhart ve arkadaşlarının geri yayılım algoritmalarını yeniden keşfetmeleri ile sağlanmıştır. Geri yayılım algoritması en çok kullanılan öğreticili öğrenme algoritmasıdır.

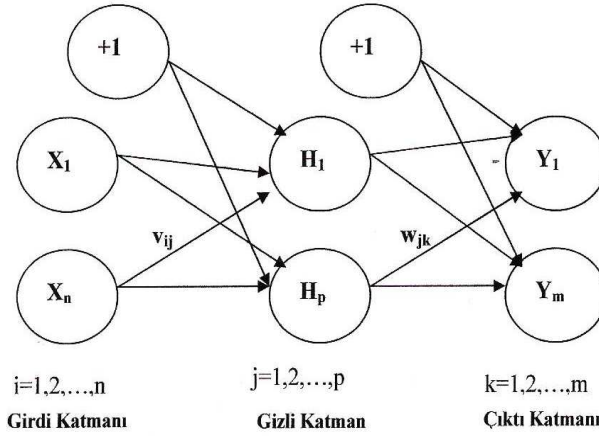
İleri beslemeli ağırlar girdiden çıktıya doğru tek yönlü bilgi akışına izin vermektedir. Bu geri beslemelerin olmadığı anlamına gelmektedir. Tipik bir ileri beslemeli yapay sinir ağı, girdi katmanı, genellikle bir veya iki ara katman (gizli katman) ve çıktı katmanından oluşmaktadır. Her katmanda ilgilenilen probleme göre değişen sayıda nöronlar bulunmaktadır (Zhang, 1998).

Şekil 2.5'de tek gizli katmanlı ileri beslemeli bir yapay sinir ağı modeli görülmektedir. Girdi katmanında n, gizli katmanda p ve çıktı katmanında m adet nöron bulunmaktadır. Her bir katmandaki nöronlar arasındaki bağlantıların ağırlıklarının düzenlenmesi ile ağın eğitimi gerçekleştirilir. Ağırlıkların düzenlenmesi işlemi,

$$E = \frac{1}{2} \sum_{k=1}^m (y_k - t_k)^2 \quad (2.2.1)$$

hata fonksiyonunun minimize edilmesi ile sağlanmaktadır. Hata fonksiyonundaki,  $y_k$ , ağın ürettiği çıktıyı,  $t_k$ , arzu edilen çıktı değerini

göstermektedir.  $\frac{1}{2}$  sabit bir katsayıdır ve fonksiyon türevini kolaylaştırmak için eklenmektedir. Geri yayılım algoritması ismini, çıktı katmanında oluşan hatayı minimize etmek için geriye doğru ağırlıkların düzenlenmesi işleminden alınmaktadır (Haykin, 1999).



**Şekil 2.5:** ileri sürümlü bir sinir ağı modeli

Radyal tabanlı fonksiyon ağları (Radyal Basis function Networks-RBFN), çok boyutlu uzayda eğri uydurma yaklaşımıdır. RBFN'nin eğitimi, çok boyutlu uzayda eğitim verilerine en uygun bir yüzeyi bulma problemidir. RBFN, yapı olarak BPN ile benzemektedir. Girdi, ara ve gizli katmanlardan oluşmaktadır. Ara katmandaki radyal tabanlı aktivasyon fonksiyonu (gaus, üstel) girdileri dönüştürmeyi sağlar. Çıktı katmanında doğrusal bir aktivasyon fonksiyonu vardır (Kartalopoulos, 1996).

RBFN'nin eğitimi BPN'nin eğitiminden daha az bir zaman gerektirmektedir. Ancak, BPN kadar doğru sonuçlar verebilmesi için daha fazla veriye ihtiyaç duymaktadır.

### 2.3. Genetik Algoritmalar

Genetik algoritmalar, doğal seleksiyon ve doğal genetik mekanizmasına dayanan araştırma algoritmalarıdır. Sayısal optimizasyon yöntemlerinden olan genetik algoritmalar, evrimsel hesaplama tekniğinin bir parçasını oluşturmakta ve geleneksel yöntemlerle çözümü zor veya hemen hemen imkansız olan problemlerin çözümünde kullanılabilir. Genetik algoritmalar, deneysel çalışmalarda optimizasyon aşamasında, endüstriyel uygulamalarda ve sınıflandırılmalarda uygulama alanı bulunmaktadır. Mühendislik alanında en çok optimizasyon amaçlı kullanılmakta olup, diğer klasik yöntemlere göre daha iyi sonuç vermektedir.

Genetik algoritmalar, bazı doğal olayları modelleyen stokastik algoritmalarıdır. Bu algoritmalar biyolojik birimin işleyiş biçimini taklit eder. Genetik algoritmalar doğal seleksiyon ve doğal genetik mekanizmasına dayanan sayısal optimizasyon algoritmalarıdır.

Evrimsel hesaplama ilk olarak I.Rechenberg tarafından “Evrimsel Stratejileri” eserinde tanıtılmıştır (Rechenberg, 1973). İlk olarak John Holland evrim sürecini bir bilgisayar yardımıyla kullanarak, genetik algoritmaları oluşturmuştur. 1992 yılında John Koza, genetik algoritmayı kullanarak genetik programlamayı geliştirmiştir.

Literatürdeki çalışmalarda araştırma alanı geniş ve karmaşık ise, konuyla ilgili bilgi az veya eldeki bilgi araştırma alanını daraltmada yeterli değilse, matematiksel analiz elde edilemiyorsa veya geleneksel araştırma

metotları ile başarısız olunmuş veya iyi sonuç elde edilememişse; genetik algoritmalarından faydalandığı görülmüştür.

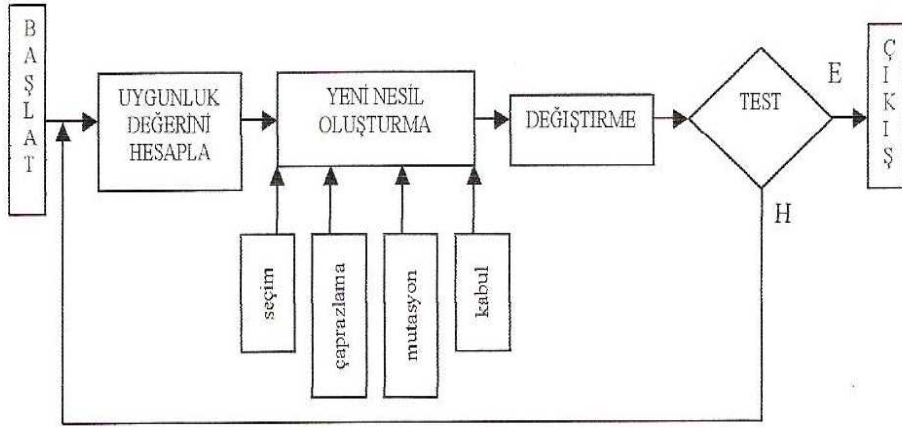
Genetik algoritmalar oldukça genel prensiplerle Şekil 2.6'da akış şemasında görüldüğü gibi çalışmaktadır.

Öncelikle ele alınan bir problem için bir rasgele  $n$  kromozomlu populasyon oluşturulur. Daha sonra populasyondaki her bir kromozom için  $f(x)$  uygunluk fonksiyonu hesaplanır. Yeni bir populasyon oluşuncaya kadar aşağıdaki adımlar tekrar edilir:

[Seleksiyon]: Uygunluk durumuna göre populasyondan iki tane kromozom çaprazlanmak (crossover) amacıyla seçilir (çoğullama-reproduction). Burada uygunluk derecesi yüksek olanın seçilme şansı yüksek olur.

[Çaprazlama]: Seçilmiş olan ebeveyn kromozomlar, çaprazlama oranına göre yeni bireyler oluşturmak üzere çaprazlanırlar. Eğer çaprazlama uygulanmazsa bireyler atalarının tamamen kopyası olacaktır.

[Mutasyon]: Kromozom üzerindeki bazı dizilerin (DNA dizilerinin) yerleri ile oynanarak belirli mutasyon oranına göre değişiklikler yapılır. Yeni populasyon kabul edildikten sonra oluşturulan yeni populasyon eskileriyle yer değiştirilir. Hedeflenen uygunluk değerine ulaşıldığında program durdurulur ve populasyondaki en iyi çözüm alınır.



Şekil 2.6:Genetik algoritmaların genel akış şeması

Genetik algoritmalarda kromozomlarla bir başlangıç popülasyonu rasgele oluşturulmaktadır. Burada popülasyon genişliğinin belirlenmesi gerekmektedir. Büyük popülasyonlarda, çözüm uzayı iyi örneklendiği için aramanın etkinliği artmakta ve karşılık olarak arama süresi uzamaktadır. Küçük popülasyonlarda ise, çözüm uzayını yeterli örnekleymeme ve zamansız yakınsama oluşabilmektedir.

Genetik algoritmanın her çevriminde yığındaki dizilerin bir değerlendirme fonksiyonu yardımıyla uygunluk değeri hesaplanmaktadır. Uygunluk fonksiyonu, kromozomları problemin parametreleri haline getirmekte ve bunlara göre hesaplama yapmaktadır. Genellikle genetik algoritmaların başarısı bu fonksiyonun verimli olmasına bağlıdır.

Genetik algortmada bağımsız parametrelerin kromozomlar içinde kodlanması gerekmektedir. Bilinen iki çeşit kodlama yöntemi bulunmaktadır.

1. İkili Kodlama: En yaygın olarak kullanılan yöntemdir. Sayılar ikilik sisteme göre sıfırlar ve birlerden oluşmaktadır.

2. Permütasyonlu Kodlama: sıralama problemlerinde kullanılmaktadır. Her kromozom dizideki bir sırayı temsil eden sayılardan oluşmaktadır.

Çoğullama (Reproduction), başlangıç popülasyonu oluşturulduktan sonra yeni popülasyonun oluşturulması için seçim yöntemine karar verilmesi gerekmektedir ve uyumu iyi olanların (bireylerin) seçilmesi olasılığı yüksek olmaktadır. Seçim yöntemi olarak literatürde birçok yöntem bulunmaktadır. En yaygın olarak kullanılanları rulet çemberi, turnuva ve elitist seçim yöntemleridir.

Rulet çemberi seçim yönteminde tüm bireylerin uygunluk değerleri bir çizelgede yazılmakta ve bu değerlerin toplamı alınmaktadır. Tüm bireylerin uygunluk değerleri toplama bölünerek (0, 1) aralığında sayılar elde edilmektedir. Daha sonra sayıların hepsi bir çizelgede toplanmakta ve bu sayılar birbirine eklenerek rasgele olarak bir sayıya doğru ilerlemeye çalışılmaktadır. Bu sayıya ulaşıldığında son eklenen sayının ait olduğu çözüm seçilmiş olur.

Elitist seçim yönteminde, popülasyonun en iyi bir bireyi korunup, popülasyonun geri kalan elemanlarını uyum orantılı seçim yöntemlerinden birini kullanarak yeni bireyler ile değiştirilmektedir. Burada amaç, en iyi uyum değerine sahip bireyin, genetik operatörler kullanıldığında kaybolmasını önlemektir.

Turnuva seçim yönteminde yerine koyarak ya da koymadan rasgele t adet birey seçilmekte ve bu büyüklüğe turnuva genişliği adı verilmektedir. Bu gruptaki en iyi birey yeni populusyona kopyalanmaktadır. Bu işlem kullanıcı tarafından önceden kararlaştırılan çevrim sayısı kadar tekrarlanmaktadır.

Seçim yönteminden sonra yeni çözümler elde etmek için çaprazlama ve mutasyon adı verilen iki genetik operatör uygulanmaktadır. Çaprazlama, genetik algoritma uygulamalarında en önemli operatördür. Çaprazlamada bireylerdeki iyi özellikleri birleştirerek daha iyi çözümlerin oluşması beklenmektedir. Ele alınan probleme göre seçilen tek nokta, iki nokta, çok nokta, üniform çaprazlamalar bulunmaktadır. Mutasyon operatörü ise, oluşan yeni çözümlerin önceki çözümü kopyalamasını önlemek için ve daha hızlı ulaşmak için kullanılmaktadır. Ele alınan probleme bağlı olarak seçilen ters çevirme, ekleme, yer değişikliği, karşılıklı değişim mutasyon operatörleri bulunmaktadır (Bolat vd., 2004).

Genetik algoritmaları cazip kılan özelliklerinin bazıları şunlardır:

1)Öğrenme: Genetik algoritma global arama tekniklerinde yaygınca kullanılmakta ve en iyisi olarak bilinmektedir. Mevcut performans ölçütlerini kullanarak verilen bir arama uzayında arama uzayını genişletme ve en iyiyi arama özelliklerini kullanma yeteneğine sahip bulunmaktadır. Bu özelliklerini çaprazlama, mutasyon ve üretim gibi genetik operatörlerle kullanarak öğrenme yeteneğine sahiptir.

2)Genetik kod yapısı: Genetik algoritma doğrudan parametrelerle değil, kodlanmış parametre dizisiyle çalışmaktadır. Bu, kullanıcının

problemleri bir deęişken optimizasyon problemi gibi çözmeye imkan vermektedir.

3)Çözümlerin optimallięi: Pek çok gerçek hayat problemlerinin Çok Modellilik (Multimodal) ve doğrusal olmama gibi özellikleri vardır. Geleneksel arama teknikleri böyle arama uzaylarında yetersiz kalmaktadır. Genetik algoritma ise, böyle karmaşık arama uzaylarında optimale yakın çözümler bulma yeteneęine sahiptir. Genetik algoritma, mühendislik, bilim, ekonomi gibi çok deęişik alanlardaki problemler için gürbüz (Robust) bir optimizasyon aracı olarak son yıllarda büyük bir önem kazanmıştır.

Genetik algoritmanın uygulama alanlarından bazıları; haberleşme şebekeleri tasarımı, elektronik devre dizaynı, gaz boruları şebekeleri optimizasyonu, görüntü ve ses tanıma, veri tabanı sorgulama optimizasyonu, uçak tasarımı, fiziksel sistemlerin kontrolü, gezgin satıcı problemlerinin çözümü, ulaşım problemleri, optimal kontrol problemleridir .

Genetik algoritmayı, dięer Evrim Algoritması (EA) türlerinden farklı kılan özellikleri;

- 1) Eşeyli üreme yöntemini,
- 2) Mutasyon ve çaprazlama operatörlerini,
- 3) Stokastik veya deterministik seçim yöntemlerini,
- 4) Problemin çözümü için problemin kendisi yerine kodlanmış bir dizisini,

kullanması olarak sayılabilmektedir.

Bu şekilde genetik algoritmalar, dięer evrim algoritması türlerinden daha esnek bir yapı sağlamaktadır.



## 2.4. Yapay Zeka Teknikleri

Yapay zeka, insan tarafından yapıldığında zeka olarak adlandırılan davranışların (akıllı davranışların) makine tarafından da yapılmasıdır. Yapay zeka'nın insan aklının nasıl çalıştığını gösteren bir kuram olduğu da söylenebilir. Yapay zeka'nın amacı, makineleri daha akıllı hale getirmek, zekanın ne olduğunu anlamak ve makineleri daha faydalı hale getirmektir.

(Tektaş vd.)' e göre, Yapay zeka'nın amacı, insanın zekasını bilgisayarlar aracılığı ile taklit etmek, bu anlamda belli bir ölçüde bilgisayarlara öğrenme yeteneği kazandırabilmektir. Bu şekilde yapay zeka çoğunlukla insanın düşünme yeteneğini, beynin çalışma modelini veya doğanın biyolojik evrimini modellemeye çalışan yöntemlerden oluşur.

Bilgisayar teknolojisindeki gelişime paralel olarak özellikle son yirmi yılda ulaşım alanında karşımıza çıkan ve sezgisel olarak çözülebilen ya da çözülmesi matematik teknikler ile mümkün olmayan gerçek hayat problemlerini (Çizelgeleme, Gezgin Satıcı vb.) çözmeye yönelik ileri teknikler Yapay zeka teknikleri olarak adlandırılır. Bunların başlıcaları:

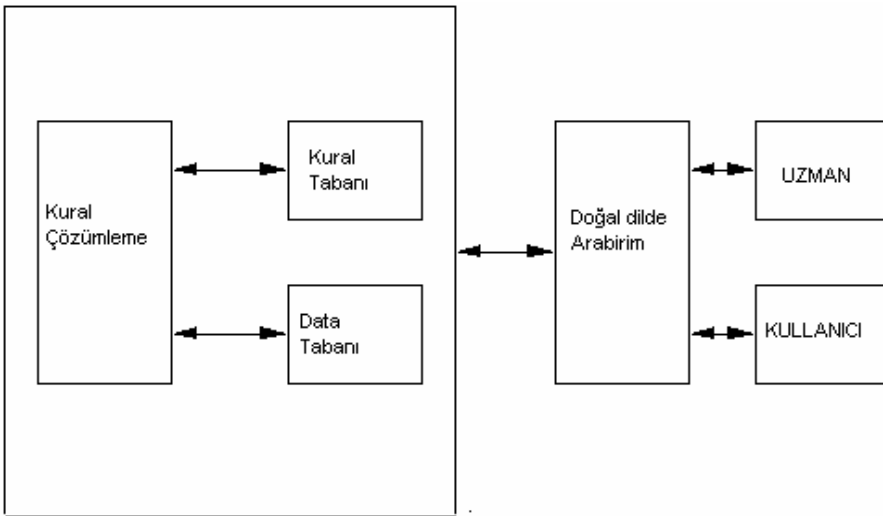
- 1) Bilgi tabanlı uzman sistem yaklaşımı
- 2) Yapay sinir ağları yaklaşımı
- 3) Bulanık mantık yaklaşımı
- 4) Geleneksel olmayan optimizasyon teknikleri
  - i) Genetik algoritma
  - ii) Benzetilmiş tavlama (simulated annealing)
  - iii) Hybrid algoritmalar

gibi sınıflandırılabilir.

Yapay zekanın en önemli uygulama alanlarından biri Uzman Sistemler'dir. Uzman sistemler kısaca, bir kural tabanlı sistem olarak nitelendirilebilmektedir. Burada oluşturulan kurallar bir uzmanın görüşü veya tecrübesine dayandırılarak oluşturulmaktadır. Oluşturulan bu kurallardan, insanın neden sonuç ilişkisine bağlı kalarak bir sonuca varması gibi mantıksal işlemler sonucunda bir çıkarım yapılmaktadır. Bu tip sistemler belli bir alanda uzman olan kişilerin uzmanlıklarına dayanarak çözüm aramaktadır.

Uzman sistemler hem makine hem de insan müdahalesine ihtiyaç duyan uygulamalarda kullanılmaktadır. Tıp, finansal planlama, bilgisayar konfigürasyonu, gerçek zamanlı sistemler, trafik yönetimi ve kontrolü, sigortacılık bazı uygulama alanlarıdır. Bir uzman sistem üç parçadan oluşmaktadır. Bunlar; Kural Tabanı, Data Tabanı ve Kural Çözümleyicidir.

Bu şekilde oluşturulmuş bir Uzman Sistemim blok yapısı Şekil 2.7'de görülmektedir.



Şekil 2.7: Bir Uzman Sistemin Blok Yapısı

Kural tabanında, kural çözümleyicinin sonuca varmak için kullanacağı kural kümesi bulunmaktadır. Pek çok uygulamada Kural tabanı Eğer-İse (if-then) kural yapısında tutulmaktadır. Bu kural kümesinde birkaç kural olabileceği gibi birkaç bin kural da olabilmektedir. Data tabanında probleme özgü olgular tutulmaktadır. Bu olgular Eğer-İse kural yapısındaki koşulların değerleridir. Kural çözümleyici ise kural tabanındaki kurallara bakarak yeni kurallar veya sonuçlar üretmektedir. Böyle bir sistemi doğal bir dil arabirimi kullanarak kullanıcının akıllı bir sistem üzerinden bir uzman ile bağlantı kurması sağlanabilmektedir.

Uzman sistemler, çözümü kesin belli olmayan problem tipleri için bazı algoritmik olmayan uzmanlıkları somutlaştıran bilgi uygulamalarıdır.

Uzman sistemlerin elemanları ve arayüzleri:

Bilgi tabanı (knowledge base): Uzmanın bildirilerinden oluşmakta olup eğer-ise kurallarıyla yapılandırılmaktadır.

Çalışma alanı (working space): Problem çözümünde gerekli özel bilgileri bulundurmaktadır

Çıkarım birimi (inference engine): Bilgi tabanından ve problemin özel verilerinden gelen tavsiyelerle değişen sistemin merkezindeki kod olarak tanımlanabilir.

Uzman sistem dizaynının anlaşılması için; sistem ile birbirini etkileyen kişisel rollerinde anlaşılmasına ihtiyaç vardır. Bunlar:

Ana uzman (domain expert): Problemin çözüm yolunu tespit eden kişi veya kişiler.

2) Bilgi Mühendisi (knowledge engineer): Uzmanın bilgisini çözerek uzman sistemin kullanabileceği şekle dönüştüren kişi.

3) Kullanıcı (user): Uzman tarafından verilen bilgileri problem çözümünde kullanacak kişi.

#### Uzman Sistemlerin Özellikleri:

1) Geriye zincirleme (backward chaning): Eğer-ise kuralları kullanılarak alt amaçlardan bir amaca varılır.

2) Belirsizlik ile işleme (coping with uncertainty): Sistemin yeteneği, tam bilinmeyen kurallar ve verilere verdiği cevaplar ile muhakeme edilir.

3) İleri zincirleme (forward chaning): Başlangıç verilerinden eğer-ise kuralları kullanılarak problem çözümüne gidilir.

4) Veri temsili (data resapantation): Sistemde (erişilebilir ve depolanabilir) probleme özel veriler.

5) Kullanıcı arayüzeyi (user interface): Sistem kullanılarak kolayca oluşturulan kod parçaları.

Bulanık mantık da bir kural tabanlı sistem olarak düşünülebilmektedir. Fakat burada nitelendirmeler, uzman sistemlerden farklı olarak kesin olmamaktadır. Bir insanın günlük hayatta yaptığı nitelendirmelerin çoğunluğu da kesin olarak nitelendirilememektedir. Bulanık mantık bu şekilde kural tabanının günlük hayatta kullanılan kesin olmayan kelimelerle oluşturulmasına imkân sağlamaktadır.

Beynin öğrenme kapasitesi, nöronlar ve bu nöronların birbiriyle olan bağlantısına bağlıdır. Yapay sinir ağları adından da anlaşıldığı gibi beynin çok basit bir nöron modelinin benzetimi olarak düşünülmektedir. Bu şekilde

öğrenilen ağ ile öğrenme olayı gerçekleşir. Yapay sinir ağları (YSA), özellikle öğrenme üzerine odaklanmıştır ve lineer olmayan sistemlerde veya sisteme ait bilginin tam olmadığı ya da hatalı olduğu sistemlerde çözüme ulaşmak için uygun bir sistemdir. Yapay sinir ağlarının en önemli dezavantajı ise, var olan bir uzman bilgisinin problem çözümüne aktarılmasındaki zorluktur. Yapay sinir ağlarının kullanım alanları, kontrol ve sistem tanımlama, görüntü ve ses tanıma, tahmin ve kestirim, arıza analizi, tıp, haberleşme, trafik, üretim yönetimi olarak sayılabilmektedir.

Genetik algoritma ise, doğanın kullanmış olduğu biyolojik evrim modelini benzeterek çözüm arayan bir optimal arama algoritmasıdır. Genetik algoritma (GA) süreci, doğal evrime benzetildiğinden dolayı, Üreme (Reproduction), Çaprazlama (Crossover), Mutasyon (Mutation) gibi doğal evrimde kullanılan operatörleri içermektedir. Üreme, uygunluk değerlerine bakılarak stokastik yöntemlerle seçilen bireylerden yeni bir popülasyon oluşturma işlemidir. Bu işlem, ilerleyen jenerasyonlarda daha yüksek uygunluk değerlerine sahip bireylerin oluşmasına neden olmaktadır. Bu nedenle bu işleme en uygunun hayatta kaldığı test (survival of the fittest) adı verilmektedir.

Çaprazlama, çoğunlukla rasgele olarak seçilen iki bireyin kromozomları çaprazlanarak gerçekleştirilmektedir. Bu işlemde, bireylerin kromozomunu oluşturan dizilerin değişik kısımlar yer değiştirerek yeni döl üretimi sağlanmaktadır. Bu döl, popülasyonda daha az uygunluk değerine sahip “zayıf” bireylerin yerine konabilmektedir. Çaprazlama, genetik algoritmada en önemli operatördür ve jenerasyonda yeni çözümlerinin üretiminden sorumludur.

Mutasyon, bireyin kromozomunu oluřturan dizideki tek bir elemanın deęerinin rasgele olarak deęiřmesidir. Mutasyon, özümün alt optimal noktalara takılmasını önleyen ve ok düşük olasılık deęeri ile uygulanan operatördür.

### 3. BELİRSİZLİK VE BULANIK SAYILAR

Her insan günlük hayatında kesin olarak bilinmeyen, bazen de önceden sanki kesinmiş gibi düşünülen ama sonuçta kesinlik arz etmeyen durumlarla karşılaşır. Gerçek dünya karmaşıktır. Bu karmaşıklık genel olarak belirsizlik, kesin düşünceden yoksunluk ve karar verilemeyeşten kaynaklanır. İnsan tarafından geliştirilmiş olan bilgisayarlar, bu tür belirsizlikleri işleyemezler ve çalışmaları için sayısal bilgiler gereklidir. Genel olarak değişik biçimlerde ortaya çıkan karmaşıklık ve belirsizlik gibi tam ve kesin olmayan bilgi kaynaklarına bulanık (Fuzzy) kaynaklar denir. Zadeh tarafından gerçek dünya sorunları ne kadar yakından incelemeye alınırsa, çözümün daha da bulanık hale geleceği ifade edilmiştir.

Bulanık ilkeler hakkındaki ilk bilgiler Azerbaycan asıllı Lütfü Askerzade tarafından ortaya atılmasına rağmen bu fikirler batı dünyasında şüphe ile karşılanmış ve yoğun tenkit almıştır. Ancak 1970'li yıllardan sonra, doğuda özellikle Japonya da bulanık mantık, sistem ve kavramlarına önem verilmiştir. Batıdaki gecikmenin ana sebebi batı kültürünün temelindeki ikili mantık yani Aristo mantığının yatması ve olaylara evet-hayır, 0- 1 gibi ikili esaslarda yaklaşılmasıdır.

Bazı araştırmacılar bulanıklık fikrini benimseyerek bu konuda çalışmayı teşvik etmiş, ama büyük bir çoğunlukta karşı görüşte olmuştur. Bu ikinci gurup, Fuzzification yani bulanıklaştırmanın kesin olan bilimsel ilkelere uymadığını ve hatta bilime karşı geldiğini ileriye sürmüşlerdir. Özellikle ihtimaller Teorisi ve istatistik gibi zaten belirsizliği konu edinen daha da belirsiz bilim dalları bulunduğundan bu konularda çalışan

arařtırmacılar, bulanık sistemlere açık biçimde karşı çıkmıřlardır (Ően, 2000).

Bulanık kavram ve sistemlerin dűnyanın deęiřik arařtırma merkezlerinde dikkat kazanması 1975 yılında Mamdani ve Assilion tarafından yapılan gerçek bir kontrol uygulaması ile olmuřtur.

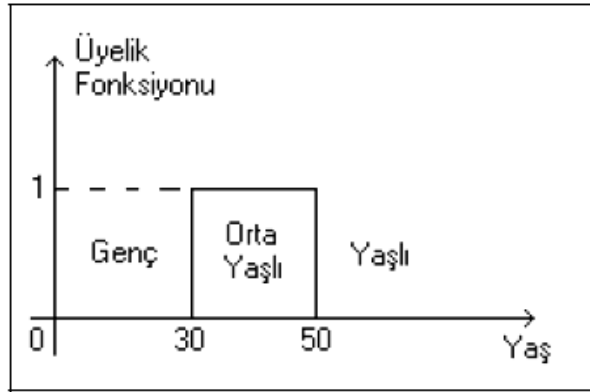
Deęiřik belirsizlik durumları arasında sözel olanlarının mükemmel olmayan bilgi içerięi olması açasından daima yaklařıklık ve bulanıklık içerdieęi anlařılmaktadır. Bu gibi belirsizlik durumlarında en uygun yöntem bilim esasının küme elamanlarına deęiřik üyelik derecelerinin verilmesi ile olacaęı Lütfü Askerzade tarafından 1965 yılında belirtilmiřtir. Aristo mantıęına göre insanlar ya uzundur ya da deęildir. Hâlbuki Zadeh'e göre uzun boyluluęun deęiřik dereceleri vardır. Uzun boylulardan biri gerçek uzun boylu olarak alınırsa, ondan biraz daha uzun veya kısa olanlar uzun boylu deęil diye dıřlanamazlar. Esas alınan uzun boyluluęun altında ve üstündeki boylar o kadar kuvvetli olmasa bile, uzun boyluluęa ait olma derecesi biraz daha az olmakla birlikte, yinede uzun boylular kümesine girmektedir. Böylelikle dűnyadaki tüm insanlar kümesindeki insanların teker teker boy açasından birer uzunluk üyelik derecelerinin bulunduęu söylenebilir.

İřte bu řekilde:0 ile 1 arasındaki deęiřimin her bir öęe için deęerine üyelik derecesi, bunun bir alt kümesi içindeki deęiřimine ise üyelik fonksiyonu adı verilir (Ően, 2001).

Klasik (geleneksel) mantıkla, herhangi bir nesne o kümeye ya aittir ya da deęildir. Bu kısaca 0 ve 1 mantıęıdır. Őekil 3.1'de görüldüęü gibi 0-

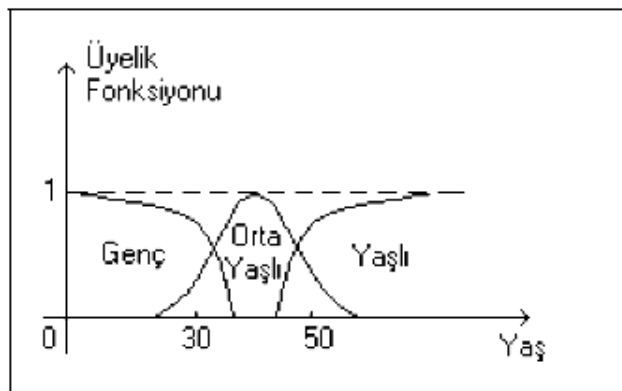


30 yaş arasındaki kişiler genç, 30- 50 yaş arasındaki kişiler orta yaşlı ve 50 yaşın üstündeki kişiler ise yaşlı sınıfına girmektedir.



Şekil 3.1: Klasik küme teorisi

Bu kurallara göre 31 yaşındaki bir kişi orta yaşlı sayılırken 29 yaşındaki bir kişi ise genç sayılmaktadır. Bu durumu bulanık mantık açısından incelersek Şekil 3.2'de görüldüğü gibi 30 yaşındaki bir kişi belli bir oranda orta yaşlı sayılırken belli bir oranda ise, genç sayılmaktadır. Bulanık mantıkta klasik mantıktaki gibi 0 ve 1 mantığı yoktur, günlük yaşantıda kullandığımız daha esnek bir yaklaşım vardır (Tuncer, 1999).



Şekil 3. 2: Bulanık küme teorisi

Klasik küme teorisindeki kesişim, birleşim ve tümlleme gibi işlemler bulanık küme teorisinde de mevcuttur. A ve B kümeleri, X evrensel kümesi içerisinde tanımlanmış iki keskin küme olduğu kabul edilirse;

Tanım: İki ayrı bulanık kümenin birbirine eşit olabilmesi için

$$\forall x \in X : \mu_A(x) = \mu_B(x) \quad (3.1)$$

şartının sağlanması gerekmektedir.

Tanım: A kümesinin b kümesinin alt kümesi olabilmesi için

$$\forall x \in X : \mu_A(x) \leq \mu_B(x) \quad (3.2)$$

şartının sağlanması gerekmektedir.

A ve B kümeleri X evrensel kümesi içerisinde tanımlanmış iki keskin küme olduğunda:

$\forall x \in X$  İçin  $A \cap B$ 'nin üyelik fonksiyonu  $\mu_{A \cap B}(x)$ :

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \quad (3.3)$$

olarak yazılabilmektedir.

$\forall x \in X$  için  $A \cup B$ 'nin üyelik fonksiyonu  $\mu_{A \cup B}(x)$ :

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \quad (3.4)$$

olarak tanımlanmaktadır.

$\forall x \in X$  İçin  $\mu_A^l(x)$  'in üyelik fonksiyonu:

$$\mu_A^l(x) = 1 - \mu_A(x)$$

olarak yazılmaktadır (Kasko, 1992; Yen and Langari, 1998).

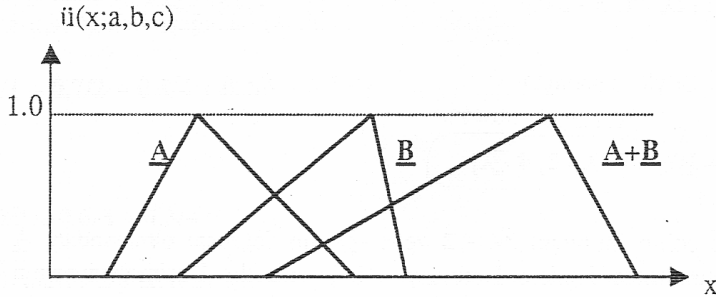
### 3.1. Bulanık Sayıların Toplanması Ve Çıkarılması

A ve B bulanık alt kümelerinin  $\lambda$  seviyelerinde kesimleri,

$A_\lambda = [a_\lambda^-, a_\lambda^+]$  ve  $B = [b_\lambda^-, b_\lambda^+]$  olarak ifade edilirse; bu iki bulanık alt kümenin toplamı

$$(\underline{A} + \underline{B})_\lambda = [a_\lambda^- + b_\lambda^-, a_\lambda^+ + b_\lambda^+] \quad (3.1.1)$$

şeklinde hesaplanabilmektedir. Bu toplamın grafik gösterimi şekil 3.3'de gösterilmiştir.



Şekil 3.3: İki bulanık kümenin toplamı

Benzer olarak iki bulanık alt kümenin birbirinden çıkarılması aşağıdaki gibi tanımlanmaktadır.

$$((\underline{A} - \underline{B})_\lambda = [EK(a_\lambda^- - b_\lambda^-, a_\lambda^+ - b_\lambda^+), EB(a_\lambda^- - b_\lambda^-, a_\lambda^+ - b_\lambda^+)]) \quad (3.1.2)$$

Bu işlemler her  $\lambda$  seviyesi için geçerli olmaktadır (Şen, 2001).

### 3.2. Bulanık Sayıların Çarpılması Ve Bölünmesi

İki bulanık sayının çarpım ve bölüm işlemlerinin anlaşılması için öncelikle aralık sayılarının bu işlemler karşısında nasıl olacağını bilmesi gerekli olmaktadır. Bir aralık sayısı sadece alt ve üst sınır değerleri ile belirlenmektedir. Bir başka deyişle, bu alt ve üst sınırlar arasında bulunan bütün sayıların üyelik değerlerinin 1'e eşit olduğu kabul edilirse, yukarıda söylenen üçgen veya yamuk yaklaşık sayıları yerine dikdörtgen bir sayı elde edilir. Aralık sayılardan iki tanesi  $A=[a, b]$  ve  $B=[c, d]$  olsun.

Bunların çarpımı tanım olarak,

$$A.B = [EK(ac, ad, bc, bd), EB(ac, ad, bc, bd)] \quad (3.2.1)$$

şeklinde yazılabilmektedir. Örneğin;  $A=[-3, 4]$  ve  $B[1, 9]$  ise,

$$A.B=[EK(-3, , -27, 4, 36), EB(-3, -27, 4, 36)]=[-27, 36]$$

elde edilir. Kural olarak, ilk sayının önce alt sınırı, sonraki sayının alt ve üst sınırları ile çarpılarak iki sayı ve benzer olarak, ilk sayının üst sınırı ile yine sonraki sayının alt ve üst sınırları çarpılarak elde edilen diğer iki sayı önceki iki sayı ile bir arada düşünülürse, çarpım işleminin sonucunu içeren dört ögeli bir küme elde edilir.

Bu küme öğelerinin en küçüğü çarpımın alt sınırını, en büyüğü ise üst sınırını göstermektedir.

Buna benzer olarak aynı iki sayının bölmesi,

$$A/B = [a,b]/[c,d]=[EK(a/c,a/d,b/c,b/d), EB(a/c,a/d,b/c,b/d)] \quad (3.2.2)$$

şeklinde ifade edilmekte olup, buradan bölen sayının sıfıra eşit olmaması gerekliliği çıkmaktadır, yani  $c \neq 0$  ve  $d \neq 0$  olması gereklidir.

Örnek olarak yine  $A[-3, 4]$ ,  $B=[1, 9]$  sayılarını bölmek istersek,

$$A/B=[EK(-3/1, -3/9, 4/1, 4/9), EB(-3/1, -3/9, 4/1, 4/9)]=[-3, 4]$$

şeklinde elde edilmektedir.

$\lambda$  seviyesinde kesilmiş iki bulanık sayı,  $A_\lambda = [a_\lambda^-, a_\lambda^+]$  ve  $B_\lambda = [b_\lambda^-, b_\lambda^+]$  ise, bunların

$\lambda$  kesim seviyesindeki çarpımları;

$$(A.B)_\lambda=[EK(a_\lambda^-b_\lambda^-, a_\lambda^-b_\lambda^+, a_\lambda^+b_\lambda^-, a_\lambda^+b_\lambda^+), EB(a_\lambda^-b_\lambda^-, a_\lambda^-b_\lambda^+, a_\lambda^+b_\lambda^-, a_\lambda^+b_\lambda^+)] \quad (3.2.3)$$

şeklinde yazılabilmektedir. Bölme işlemi ise,

$$(A/B)_\lambda=[EK(a_\lambda^-/b_\lambda^-, a_\lambda^-/b_\lambda^+, a_\lambda^+/b_\lambda^-, a_\lambda^+/b_\lambda^+),$$

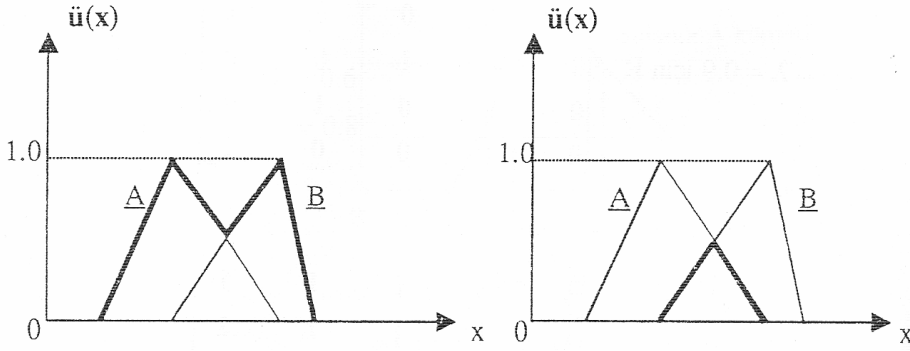
$$EB(a_\lambda^-/b_\lambda^-, a_\lambda^-/b_\lambda^+, a_\lambda^+/b_\lambda^-, a_\lambda^+/b_\lambda^+)] \quad (3.2.4)$$

olarak tanımlanmaktadır (Şen, 2001).

### 3.3. Durulaştırma İşlemleri

Bir bulanık küme işlemi sonundaki bulanık kümenin tek sayı haline dönüştürülmesi gerekebilmektedir. Bu bulanıklaştırma işleminin aksi olan durulaştırma işlemi ile yapılmaktadır. Yapılan işlemler sonrasında bulanık

sonuçlardan biri şekil 3.4 (a)'deki gibi yamuk, diğerinin ise, şekil 3.4 (b)'deki gibi üçgen şeklinde olduğu varsayıldığında, bunların ikisinin birleşimi ile yapılan son işlem sonrası bulanık çıkarım kümeleri elde edilmektedir.



**Şekil: 3.4:** İki bulanık kümenin (a) birleşimi, (b) kesişimi

Şekil 3.4'de iki tane bulanık kümenin birleşimi sonucunda elde edilen bulanık çıkarım gösterilmiştir. Literatürde değişik durulaştırma işlemleri bulunmaktadır. Hangi durulaştırma işleminin kullanılacağına ise ele alınan problemin yapısına göre araştırmacı karar vermelidir.

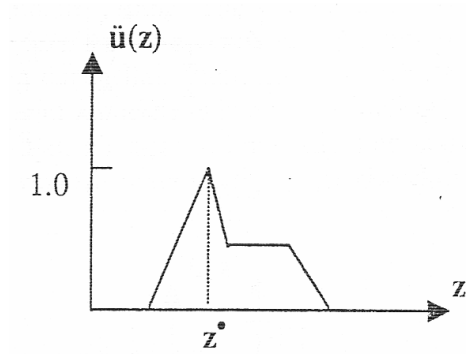
Belli başlı durulaştırma işlemleri:

- 1) En büyük Üyelik İlkesi
- 2) Sentroid (Ağırlık Merkezi) Yöntemi
- 3) Ağırlıklı Ortalama Yöntemi
- 4) Ortalama En Büyük Üyelik
- 5) Toplamların Merkezi
- 6) En Büyük Alanın Merkezi
- 7) En büyük ilk veya son üyelik derecesi
- 8) OWA (Sıralı ağırlıklı Ortalama)

### 9) WABL (Weighted average Based on the levels) Yöntemi

Bunların her birinde yapılan işlemler farklılaşmaktadır.

En büyük üyelik ilkesi yönteminin bir diğer adı Yükseklik Yöntemidir. Bu yöntem unimodal yani bir tepe noktası olan bulanık sayılarda geçerli olmaktadır. Bulanık sayının temsilci eleman olarak adlandırılan gösterim değeri olarak en büyük üyelik değerine sahip eleman seçilmektedir.



Şekil 3.5: En büyük üyelik yöntemi durulaştırması

Sentroid (Ağırlık merkezi) yönteminde,

Kesikli olarak,

$$a = \frac{\sum_i \mu_A(x_i) x_i}{\sum_i \mu_A(x_i)} \quad (3.3.1)$$

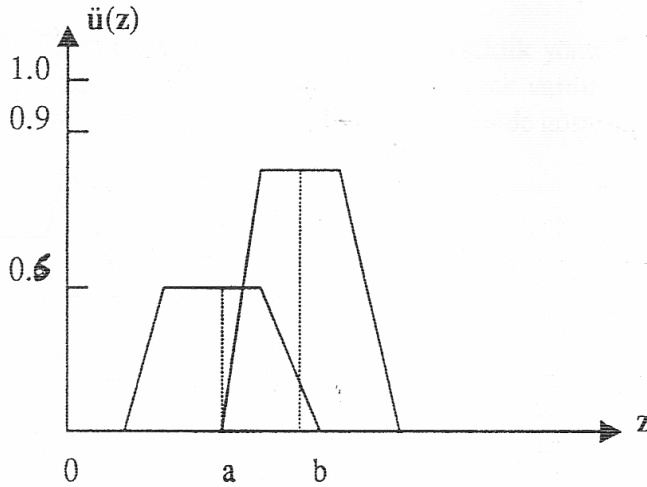
Sürekli olarak,

$$a = \frac{\int \mu_A(x) \cdot x dx}{\int \mu_A(x) dx} \quad (3.3.2)$$

olarak temsilci deęer hesaplanmaktadır.

Aęırlıklı ortalama yöntemi, k sayıda bulanık sayı için bir ortak deęer oluşturmak için kullanılmaktadır.

$$a = \frac{\sum_{k=1}^n \mu(a_k) a_k}{\sum_{k=1}^n \mu(a_k)} \quad (3.3.3)$$



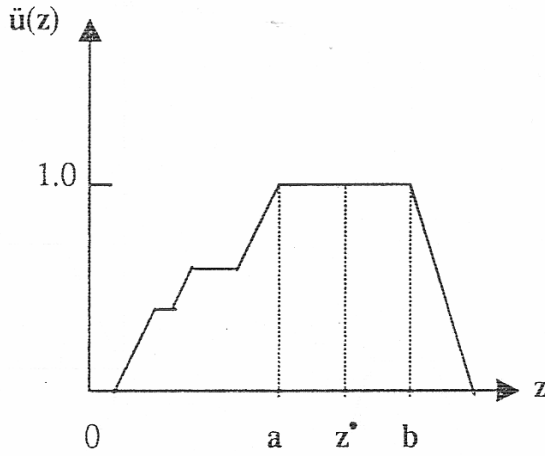
Şekil 3.6: Aęırlıklı ortalama yöntemi durulaştırması

Ortalama En Büyük Üyelik Yöntemi aynı zamanda en büyüklerin ortası diye de bilinmektedir. Bu bakımdan birinci durulaştırma ilkesine çok



yakın bir yöntemdir. Ancak burada en büyük üyelik birden fazla olabilir. En büyük üyelik derecesine sahip olanların ortalaması alınmaktadır.

$$a = \frac{a_L + a_R}{2} \quad (3.3.4)$$

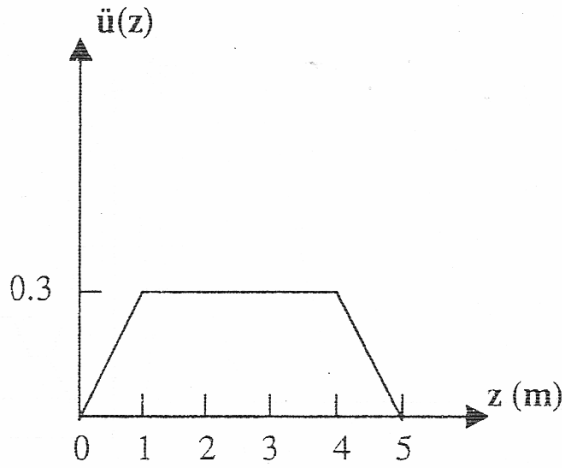


Şekil 3.7: Ortalama en büyük üyelik durulaştırması

Toplamların merkezi yönteminde, iki bulanık kümenin birleşimi yerine onların cebirsel toplamları kullanılmaktadır. Bu yöntemin bir kusuru örtüşen kısımların iki defa toplama girmesidir (Şen, 2001).

$$a = \frac{\int \sum_{i=1}^k \mu_i(x) x dx}{\int \sum_{i=1}^k \mu_i(x) dx} \quad (3.3.5)$$

Bu yöntem ağırlıklı ortalama durulaştırmasına benzemektedir ancak, toplamların merkezi yönteminde ağırlıklar ilgili üyelik fonksiyonlarının alanlarıdır. Ortalama ağırlıklar yönteminde ise, bu üyelik derecesidir (Şen, 2001).



Şekil 3.8: Toplamların merkezi yöntemi durulaştırması

En büyük alanın merkezi yönteminde, grafik olarak alanı en büyük olan eleman temsilci değer olarak seçilmektedir.

En büyük ilk veya son üyelik derecesi yönteminde ya en büyük üyelik derecesine sahip eleman ya da en küçük üyelik derecesine sahip eleman temsilci eleman olarak seçilmektedir.

OWA(Sıralı Ağırlıklı Ortalama) Yönteminde,  $a_1, a_2, \dots, a_n$  değerlerine ortak bir a temsilcisi bulmak için,

$$a = \text{OWA} (a_1, a_2, \dots, a_n) \quad (3.3.6)$$

Önce bu değerler sıralı olarak dizilirler,  $b_1 \leq b_2 \leq \dots \leq b_n$  şekline dönüşürler.

Burada her birine  $a_1, a_2, \dots, a_n$  değerlerine bağlı olmayarak sadece sıralamada i.yerde olan elemanın ağırlığı  $W_i$  olacak şekilde 0 ile 1 arasında ağırlıklar verilir.

$$W_1 \leq W_2 \leq \dots \leq W_n, \sum_{i=1}^n W_i = 1, W_i \in [0,1] \quad (3.3.7)$$

Sonra sıralı ağırlıklı ortalama eşitlik 3.3.8'deki gibi hesaplanmaktadır.

$$\text{OWA}(a_1, a_2, \dots, a_n) = \sum_{i=1}^n W_i b_i \quad (3.3.8)$$

Burada  $W_i$  ağırlıkları ortalama stratejiyi belirlemektedir. Yani,

$$\text{Eğer } W=(1,0,\dots,0) \text{ olursa } = \text{OWA}(a_1, a_2, \dots, a_n) = \min(a_i) \quad (3.3.9)$$

$$\text{Eğer } W=(0,0,\dots,1) \text{ olursa } = \text{OWA}(a_1, a_2, \dots, a_n) = \max(a_i) \quad (3.3.10)$$

$$\text{Eğer } W=\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right) \text{ olursa } = \text{OWA}(a_1, a_2, \dots, a_n) = \frac{\sum_{i=1}^n a_i}{n} \quad (3.3.11)$$

WABL Yönteminde ise, bulanık sayının ortalama gösterim değeri

$$\text{WABL}(A) = \int_0^1 (c_L \cdot L_A(\alpha) + c_R \cdot R_A(\alpha)) p(\alpha) d\alpha \quad (3.3.12)$$

$c_L + c_R = 1$  ve  $c_L, c_R \geq 0$  koşulları aranır.

$c_L, c_R$  bulanık sayının uygun sol veya sağ tarafının ağırlığını belirtmektedir. Yani optimistik-pessimistik stratejilerini belirtmektedir.

$P(\alpha)$ , seviye kümelerinin ağırlıklarının yoğunluk fonksiyonudur ve

$$\int_0^1 p(\alpha) d\alpha = 1 ; \quad P(\alpha) \geq 0 \quad (3.3.13)$$

şartını sağlamalıdır. Tamamen karar vericinin stratejisine bağlı bir yöntemdir. WABL yöntemi seviyelere dayalı yöntemler içinde en evrenseldir ve eklemeli toplamalı bir özelliğe sahiptir. WABL yönteminde yukarı ve aşağı önem  $P(\alpha)$  ile, sağa ve sola önem ise,  $c_L, c_R$  ile karar verici tarafından ayarlanmaktadır (Nasibov, 2002).

$$WABL(A+B) = WABL(A) + WABL(B) \quad (3.3.14)$$

#### 4. BULANIK REGRESYON MODELLERİ

Regresyon analizi aralarında sebep sonuç ilişkisi bulunan iki veya daha fazla değişken arasındaki ilişkiyi, o konuyla ilgili tahminler ya da kestirimler yapabilmek amacıyla regresyon modeli olarak adlandırılan matematiksel bir model ile tanımlanan bir istatistik analiz tekniğidir. Ancak günümüzde sistemler oldukça karmaşık yapıda olduklarından dolayı sistem bilgilerinin elde edilmesinde kesin sayısal değerler her zaman elde edilemeyebilir. Bunun yanı sıra araştırmalarda kişi fikir ve düşüncelerinden, sistemin çevresindeki sınırlı çevrede gözükken çeşitli kesin olmayan faktörlerden dolayı da klasik regresyon analizinin kullanımı uygun olmayabilir.

1982 yılında bulanık regresyon yöntemi ilk olarak Tanaka tarafından ortaya atılmıştır. Tanaka kesin olmayanlığın belirsizlik çeşidini ele almak için bir lineer programlama tekniği kullanarak ilk bulanık regresyon modelini oluşturmuştur. 1988'de Diamond klasik en küçük kareler regresyon analizine benzer şekilde bulanık en küçük kareler regresyon yöntemini ileri sürmüştür. Daha sonrada birçok araştırmacılar tarafından gerek Tanaka'nın metoduna benzer programlamaya dayalı matematiksel modeller gerekse Diamond'un bulanık en küçük kareler metoduna benzer, regresyon modelleri ortaya atılmıştır.

Yang, Liu (2003) interaktif bulanık doğrusal regresyon modelleri için gürültü azaltıcı bir algoritma önermişlerdir. Bu algoritmalar sapanlara karşı duyarlı algoritmalarlardır. Tran, Duckstein (2002) klasik ve bulanık regresyon modellerinin sırasıyla merkezi eğilim ve olabilirlik özelliklerini

birleştiren çok amaçlı bir bulanık regresyon modeli sunmuşlardır. Wu, Tseng (2002); en küçük kareler yaklaşımı ile bulanık parametre tahminli bulanık regresyon modeli kurmuşlardır. Hajati, Bector, Smimou (2004); sadece bağımsız değişkenlerin bulanık, hem bağımlı değişkenin hem de bağımsız değişkenin bulanık olduğu durumlar için hesaplanan yeni bir yöntem önermişlerdir. Nasrabadi (2003) bulanık veya bulanık olmayan bağımlı değişken, bulanık veya bulanık olmayan bağımsız değişkenli bulanık doğrusal regresyon modelini ele alarak matematiksel programlama tekniğine dayalı bir tahminleme yöntemi geliştirmiştir. Wang, Tsaur (2000), Tanaka'nın bulanık olmayan bağımsız değişken ve bulanık bağımlı değişkenli problemin çözümü için geliştirdiği modeli değiştirerek yeni bir bulanık en küçük kareler yöntemi ortaya atmışlardır. Xu ve Li (2001), bulanık sayılar üzerinde bir uzaklık tanımlaması yaparak, bulanık çok değişkenli doğrusal regresyon modelini geliştirmişlerdir. Ayrıca tahmin ile gözlem değeri arasındaki uyumun ölçüsünü veren bir de endeks tanımlamışlardır. Chang (2001), ağırlıklandırılmış bulanık aritmetiğin yeni bir tanımını ve iyi bilinen en küçük kareler uyum kriterini kullanan yeni bir model önermiştir.

Tanaka'nın matematiksel yaklaşımını ve onu örnek olarak geliştirilmiş modellerde amaç fonksiyon bağımlı değişkenin tahmin değerlerinin yayılmalarının minimizasyonundan oluşmaktadır. En küçük kareler yönteminde ise amaç fonksiyon bağımlı değişkenin gözlem ile tahmin değeri arasındaki uzaklığın minimizasyonundan oluşmaktadır.

Tanaka'nın bulanık regresyon yöntemi bazı eleştiriler almıştır, bu eleştirilerden bazıları şunlardır:

1)Tanaka'nın modeli bağımsız değişkenlerin bulanık olmadığı ve bağımlı ve bağımsız değişkenler arasındaki ilişkinin doğrusal olduğu durumlar için oluşturulmuştur. Ancak bağımsız değişkenler her zaman bulanık olmayan sayılardan oluşmayabilir ve bağımlı ve bağımsız değişkenler arasındaki ilişki her zaman doğrusal olmayabilir.

2)Sapanlara karşı oldukça duyarlı bir yöntemdir.

3)Elde edilen regresyon aralığı ile ilgili uygun bir yorum bulunmamaktadır.

4)Tahminleme konusu tam olarak irdelenmemiştir.

5)Değişken sayısının artması modelin çoklu bağlantı durumuna meyilli olmasına neden olur (Wang, Tsaur, 2000).

6)Modelde gözlem sayısı arttıkça tahmin edilen bağımlı değişkenin yayılması daha geniş hale gelir.bu durum ise daha çok gözlem daha iyi tahmin sağlar önsezisine aykırıdır.

7)Bağımsız değişkenin büyüklüğü arttıkça gözlemlenen bağımlı değişkenin yayılması azalsa bile bağımlı değişkenin tahmin değerinin yayılması daha geniş hale gelir (D'urso, 2003).

Buna karşın klasik en küçük karelere benzediğinden dolayı bulanık en küçük kareler yöntemi daha az eleştiri almıştır.

#### 4.1. Klasik ve Bulanık Regresyon Modelleri Arasındaki Farklılıklar

Gerek günlük hayatın karmaşıklığından dolayı gerekse gelişen teknoloji sebebiyle günümüzde karşılaştığımız ve birçok problemi çözmekte kullandığımız veriler her zaman kesin olarak toplanamazlar. Örneğin bir nehrin su seviyesi akıntıdan dolayı ölçülemediğinde su seviyesi için “30 metre dolaylarında” denilebilir. Bu; “30 metre dolaylarında” sözcüğü; “30 bulanık sayısı” olarak algılanmalıdır. Örneğin, herhangi bir kümenin içindeki elemanların “A” ile tanımlandığını, küme dışının ise “B” ile tanımlandığını düşünürsek küme sınırındaki elemanların hangi tanıma gireceğini söyleyemeyiz. Bu ve buna benzer bulanık veri gözlenen durumlarda bulanık modellerin kullanılması daha uygun olacaktır. Zadeh’in bulanık kümeleri ortaya atmasından bu yana literatürde bulanık bilgi içeren bir çok modeller yerini almıştır (Wu, 2002).

Bulanık regresyonun bazı durumlarda klasik regresyona göre daha iyi sonuçlar vereceği önceki çalışmalarda gösterilmiştir. Bu durumlar;

- 1)Veri seti istatistiksel regresyon analizi için yeterli değilse,
  - 2)İstatistiksel dağılış varsayımları sağlanmıyorsa,
  - 3)Regresyon modelinin temsilciliği zayıf ise,
  - 4)İnsan yargısı varsa (Girdi veya çıktılar bulanık sayı ise),
  - 5)Hata terimleri model bilgisinin belirsizliği ile ilişkili ise,
- (Özelkan ve Duckstein, 2000).

Balistik, psikometri, ergonomi, güvenilirlik pazarı, yönetim ve mühendislik gibi birçok alanda bulanık bilgiyi dikkate alan uygulamalar vardır. Gerçekte gözlemlerin tam olarak bilinmediği veya kesin olarak



sınıflanamadığı birçok durum vardır, bu yüzden onların yaklaşık bir tanımlaması yapılır veya onlara karşılık gelen aralıklar verilir. Örneğin, bir video terminalinde karakter büyüklüğünün okuma yeteneği üzerindeki etkisi araştırılırken deneysel çıktı olan okuma yeteneği, yaşa, görüş açısına, çevreye, kişinin ne kadar yorgun olduğuna bağlıdır. Bu faktörlerden bazıları kesin olarak tanımlanamazlar, bu tip çıktıların en iyi tanımı bulanık çıktı tipi ile yapılır (D'urso ve Gastaldi, 2000).

İstatistiksel regresyon analizi olasılık teorisine dayanarak girdi ve çıktılar arasındaki ilişkiyi ortaya koymada oldukça sık kullanılan bir tekniktir. Modeller bulanık olmayan ilişkiyi ortaya çıkarmasının yanında rasgele değişken olarak adlandırılan gerçek çıktı ve model tahmini arasındaki sapmayı verir. Bu durum istatistiksel bakış açısından regresyon modelin “oluş frekansı“ kavramını yansıttığı şeklinde görülmektedir (Tanaka ve Guo, 1999).

İstatistiksel regresyon analizinde, gözlem ve tahmin değerleri arasındaki sapmaların rasgele hatalar olduğu varsayımı yapılır. Böylece istatistikler teknikler regresyon analizinde hem tahminlemede hem de çıkarımda kullanılır. Bu hatalar bazen de sistem bilgisinin yetersizliğinden veya belirsiz verilerden kaynaklanır. Regresyon modelinin bu tipindeki kesin olmayanlık rasgelelikten daha çok bulanıklık gösterir (Yang ve Liu, 2003).

## **4.2. Bulanık Uzaklık Ve Farklılık Yöntemleri**

Literatürde iki nicelik arasındaki uzaklığı veren birçok yöntem bulunmaktadır. Bir kısmı bulanık sayının kapladığı alan değerlerini

kullanırken diğ er bir kısmı da nokta deęerlerini kullanmaktadır. Bu uzaklıklar kullanılarak literatürde deęişik bulanık en küçük kareler regresyon yöntemleri önerilmiştir.

### 4.3. Diomand'un Bulanık Regresyon Modeli

$R^n$  üzerinde bütün normal, üst yarı sürekli bulanık içbükey kümelerin bir sınıfı  $\mathcal{E}^n$  olsun. Öyle kümeler vardır ki, seviye kümeleri aralıklara yakın, en az bir noktasının üyelik deęeri 1 ve tanımlı olduęu aralıkta 0 deęildir. Bu tip kümeler taşıyıcı fonksiyonları ile karakterize edilirler.

$$u \in \mathcal{E}^n \text{ için, } S_u(\alpha, p) = \text{Sup}\{\langle p, a \rangle : a \in [u^\alpha], p \in S^{n-1}\} \quad (4.3.1)$$

$$[u^\alpha] = \{x \in R^n : u(x) \geq \alpha\}, 0 < \alpha < 1 \quad (4.3.2)$$

$S^{n-1}; R^n$  de  $\|x\|$  küresel yüzeydir. Buna göre  $P_2$  metrik tanımlaması şöyledir:

$$P_2(u, v) = \left( \int_0^1 \int_{S^{n-1}} (S_u(\alpha, p) - S_v(\alpha, p))^2 \mu(d_p) d\alpha \right)^{1/2} \quad (4.3.3)$$

$\mu(d_p); S^{n-1}$  üzerinde birim ölçüdür.

$L, R = R^+ \text{ --- } [0, 1]$  soldan sürekli artmayan fonksiyonlardır ve  $L(0) = R(0) = 1$ 'dir.

$$U(x) = L([m-x]/l), x: m \text{ ise};$$

$$R([x-m]/r), x>m \text{ ise;} \quad (4.3.4)$$

Şeklinde sol ve sağ yayılmaları sırasıyla 1,  $r>0$  ve tepe noktası  $m$  olan bir bulanık sayıdır.  $(a, c)_L$  notasyonu  $L$ 'nin çift olduğu bulanık bir sayıyı göstermektedir.  $D_2, T$  üzerinde bir  $L_2$  metriğidir ve  $P_2$  metriğine denktir.  $(T, D_2)$  tam metrik uzaylardır. Diomand; aşağıda verilen uzaklık tanımlamasını yapmıştır.

$$D_2(u, v)^2 = (u^- - v^-)^2 + (u^m - v^m)^2 + (u^+ - v^+)^2 \quad (4.3.5)$$

Burada  $u^\Delta = (u^-, u^m, u^+)$  ve  $v^\Delta = (v^-, v^m, v^+)$  üçgen bulanık sayılardır,  $u^-$  ve  $v^-$  sırasıyla  $u$  ve  $v$ 'nin sol uç değerlerini,  $u^+$  ve  $v^+$  sırasıyla  $u$  ve  $v$ 'nin sağ uç değerlerini göstermektedir (Slowinski, 1998).

Diomand yukarıdaki üçgen bulanık sayıların merkez, sağ ve sol değerlerini kullanarak bir bulanık uzaklık tanımlamış ve bu uzaklığa dayanan bir bulanık en küçük kareler regresyon modeli önermiştir. Ancak Diamond'un bu uzaklık tanımını üçgen bulanık sayılara dayandırarak tanımlamasından dolayı diğer tip bulanık sayılar için aynı işlemlerin yapılması bazı kötü sonuçlar doğurabilir. Örneğin, aşağıdaki üyelik fonksiyonlarına sahip üçgen ve karesel iki bulanık sayıyı ele alalım.

$$\mu_v(x) = 1 - \left( \frac{v^m - x}{v^m - v^-} \right), \quad v^- \leq x \leq v^m$$

$$\left\{ \begin{array}{l} 1 - \left( \frac{x - v^m}{v^+ - v^m} \right), \quad v^m \leq x \leq v^+ \end{array} \right.$$

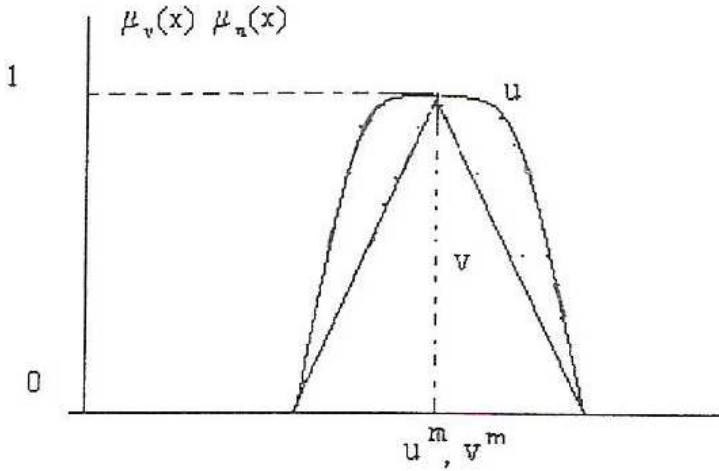
$$0, \text{ aksi halde} \quad (4.3.6)$$

$$\mu_u(x) = \begin{cases} 1 - \left( \frac{u^m - x}{u^m - u^-} \right)^{1/2}, & u^- \leq x \leq u^m \\ 1 - \left( \frac{x - u^m}{u^+ - u^m} \right)^{1/2}, & u^m \leq x \leq u^+ \end{cases}$$

$$1 - \left( \frac{x - u^m}{u^+ - u^m} \right)^{1/2}, \quad u^m \leq x \leq u^+$$

0, aksi halde (4.3.7)

Şekil 4.1'de görüldüğü gibi bulanık sayıların merkez, sağ ve sol uç değerleri aynı olduğunda iki bulanık sayı arasında  $\alpha = 0$  seviyesi haricinde bazı farklılıkların olduğu görülür. Bu durumda ise  $\alpha = 0$  seviyesine dayandırılarak yapılacak olan uzaklık tanımlaması bu farklılıkları dikkate almayacağından yanlış sonuçlar verebilir.



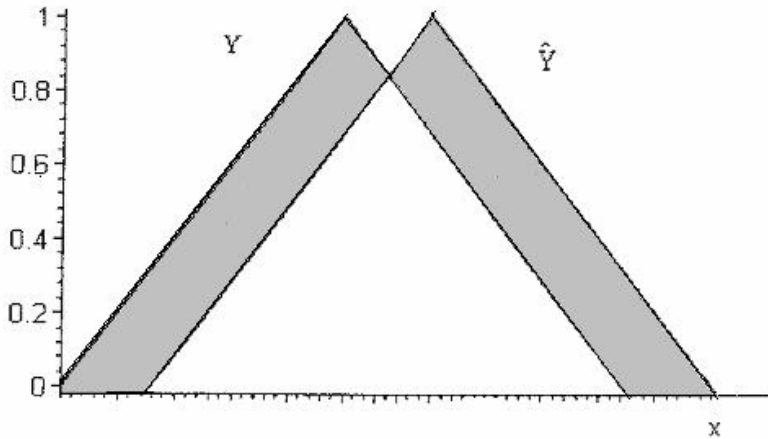
Şekil 4.1: u ve v'nin üyelik fonksiyonları

#### 4.4. Kim Ve Bishu'nun Bulanık Regresyon Modeli

Kim ve Bishu bulanık sayıların tanımlamasına dayanarak FMLS (Bulanık üyelik fonksiyon en küçük kareler) regresyon modelini önermişlerdir. Modelde regresyon fonksiyonu şöyledir:

$$\begin{aligned}
 Y &= (y - |L^{-1}(a)| \times e, y, y + |L^{-1}(a)| \times e) \\
 &= (w_1, q_1, l_1) x_1 + (w_2, q_2, l_2) x_2 + \dots + (w_m, q_m, l_m) x_m \\
 &= \left( \sum_{i=1}^m w_i x_i, \sum_{i=1}^m q_i x_i, \sum_{i=1}^m l_i x_i \right) \quad (4.4.1)
 \end{aligned}$$

Bağımlı değişkenin gözlem ile tahmin değerleri arasındaki üyelik değerleri farklılığına ilişkin bir örnek şekil 4.2'de verilmektedir.



**Şekil 4.2:** Bağımlı değişkenin tahmin ile gözlem değerlerinin üyelik değerleri

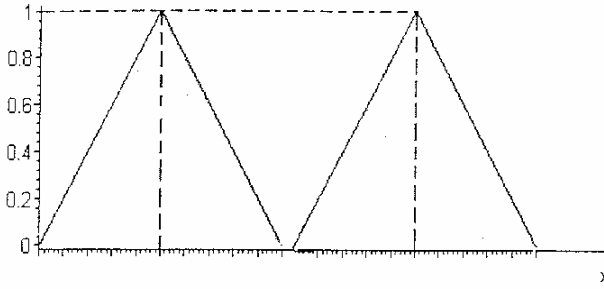
İki üyelik değeri arasındaki farkların toplamı aşağıdaki gibi hesaplanır.

$$D = \int_{S_y \cup S_{\hat{y}}} |\mu_y(x) - \mu_{\hat{y}}(x)| dx \quad (4.4.2)$$

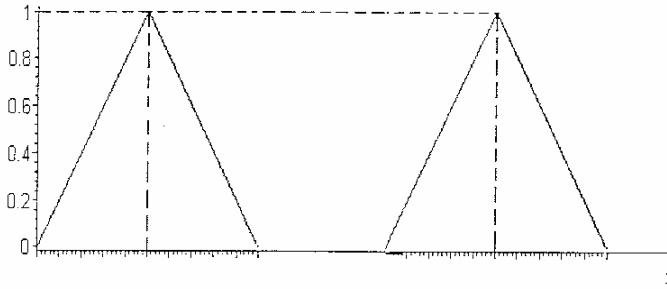
Burada D iki üyelik fonksiyonu arasındaki üyelik değerlerinin farklılığını göstermektedir.  $S_y$ ,  $S_{\hat{y}}$  sırasıyla  $\mu_y(x)$  ve  $\mu_{\hat{y}}(x)$ 'in taşıyıcılarıdır (Kim ve Bishu, 1998).

$S_y = [a, b]$ ,  $S_{\hat{y}} = [c, d]$  olsun. Burada a, b, c, d sırasıyla y ve  $\hat{y}$ 'nin  $\alpha = 0$  seviyesindeki minimum ve maximum nokta değerlerini gösterebilir.  $[a, b] \not\subset [c, d]$  ve  $[c, d] \not\subset [a, b]$  olduğu varsayalım. A, b, c, d'nin farklı değerleri için y ve  $\hat{y}$  üçgen bulanık sayıları Şekil 4.3'de gösterildiği gibi üyelik fonksiyonlarına sahip olabilir.

(a)



(b)



Şekil 4.3: Y ve  $\hat{Y}$  arasındaki farklılık dereceleri aynı olan üyelik değerleri

Şekil 4.3’de gösterilen  $y$  ve  $\hat{y}$  arasındaki farkların toplamı aynı olduğu halde, iki üyelik fonksiyonunun merkez değerlerinin birbirinden farklı olduğu görülmektedir. Dolayısıyla bu gibi durumlarda yanlış sonuçlara varılabilir.

#### 4.5. Tanaka’nın Bulanık Regresyon Modeli

Tanaka  $m$  bağımsız değişkenli, bağımlı değişkeni simetrik bir bulanık sayı olan bağımsız değişkenleri ise crisp olan bir model geliştirmiştir. İlgizlem için bağımsız değişken vektörü  $X_i = (X_{i0}, X_{i1}, \dots, X_{in})^T$ ,  $\tilde{Y}_i = (\bar{y}_i, e_i)_L$  tahminlenen bulanık bağımlı değişkendir. Amaç  $\tilde{Y}_i = \tilde{A}_0 X_{i0} + \tilde{A}_1 X_{i1} + \dots + \tilde{A}_n X_{in} = \tilde{A} X_i$  şeklinde bir (FLS) bulanık lineer regresyon modeli tahmin etmektir. Modelde  $\tilde{A} = (\tilde{A}_0, \tilde{A}_1, \dots, \tilde{A}_n)$  bulanık katsayıların bir vektörüdür ve  $\tilde{A}_j = (a_j, \alpha_j)_L$ ;  $j=0, \dots, n$  simetrik bir bulanık sayıdır. Tanaka tarafından geliştirilen modelde bulanık parametreler belirli şartlara göre tahminlenir. Bu şartlardan birisi belli bir  $h$  seviyesine,  $0 \leq h \leq 1$  olmak üzere karar verilmesidir. Tanaka ve Watada, belli bir  $h$  seviyesi için tahmin değerlerinin toplam yayılmasını minimize ederken bulanık parametrelere karar vermek için bir lineer programlama problemi (Min.problemi) geliştirmişlerdir.

$$\text{Min} \sum_{i=1}^m \sum_{j=0}^n \alpha_j |X_{ij}|$$

$$\text{s.t.} \sum_{j=0}^n a_j X_{ij} + |L^{-1}(h)| \sum_{j=0}^n \alpha_j |X_{ij}| \geq \bar{y}_i + |L^{-1}(h)| e_i, \quad i = 1, 2, \dots, m$$

$$\sum_{j=0}^n a_j X_{ij} + |L^{-1}(h)| \sum_{j=0}^n \alpha_j |X_{ij}| \geq -\bar{y}_i + |L^{-1}(h)| e_i, \quad i = 1, 2, \dots, m$$

$$a_j \in R, \quad \alpha_j \geq 0, \quad j = 0, 1, \dots, n \quad (4.5.1)$$

Tanaka'nın modelinin avantajı onun programlama ve hesaplama kolaylığıdır. Buna karşın modelin dezavantajları da vardır. Bunlardan birisi modelin sapanlara karşı oldukça hassas olmasıdır. Tanaka'nın metodunun hesaplama karmaşıklığı  $O(m^2n)$ 'dir. Diomand'ın hesaplama karmaşıklığına baktığımızda  $O(n^4m^2)$ ; Tanaka'nın modelinin hesaplama kolaylığı açısından Diomand'un modeline göre daha avantajlı olduğu söylenebilmektedir (Wang, Tsaur, 2000).

#### 4.6. Nasrabadi Ve Nasrabadi'nin Bulanık Regresyon Modeli

Bulanık çıktı ve crisp girdi ile bulanık regresyon modellerinin parametrelerini tahminlemek için matematiksel bir programlama modeli geliştirmişlerdir. Model gözlem ve tahmin değerleri arasındaki toplam farklılığın karesini minimize etmeye dayanmaktadır. Modelde,  $\tilde{A}_i$ ;  $i=0, 1, 2, \dots, n$  simetrik bulanık sayılardır. Amaç fonksiyon gözlem değerlerinin yayılmaları ve tahmin regresyonunun yayılması arasındaki farklılığın toplam karesini minimize etmektir. FLR modelinin uyum derecesi, belli bir eşik değeri olan  $h$ 'a eşit yada büyüktür. ( $0 \leq h \leq 1$ ) Matematiksel model:

$$\text{Minimize } Z(h) = \sum_{i=1}^m (\alpha |X_i| - e_i)^2$$



$$\sum_{j=0}^n a_j X_{ij} + |L^{-1}(h)| \sum_{j=0}^n \alpha_j |X_{ij}| \geq \bar{y}_i - |L^{-1}(h)| e_i, \quad i = 1, 2, \dots, m$$

ve

$$\sum_{j=0}^n a_j X_{ij} - |L^{-1}(h)| \sum_{j=0}^n \alpha_j |X_{ij}| \leq \bar{y}_i + |L^{-1}(h)| e_i, \quad i = 1, 2, \dots, m. \quad (4.6.1)$$

şeklinde ifade edilmektedir(Nasrabadi and Nasrabadi, 2005).

#### 4.7. Wang'ın Bulanık Regresyon Modeli

Wang(2000), crisp girdi ve bulanık çıktılı verilerle Tanaka'nın modelini biraz değiştirip yeni bir metot önermiştir.

$$\text{Min. } \sum_{j=0}^N \left( \sum_{i=1}^m c_j X_{ij} - e_i \right)^2$$

$$\text{s.t. } \sum_{j=0}^N \alpha_j X_{ij} + (1-h)^{1/2} \sum_{j=0}^N c_j |X_{ij}| \geq \bar{y}_i + (1-h)^{1/2} e_i$$

$$- \sum_{j=0}^N \alpha_j X_{ij} + (1-h)^{1/2} \sum_{j=0}^N c_j |X_{ij}| \geq -\bar{y}_i + (1-h)^{1/2} e_i$$

$$c_j \geq 0, \quad \alpha_j \in R, \quad X_{i0} = 1, \quad 0 \leq h \leq 1 \quad (4.7.1)$$

şeklinde ifade edilen modelin amaç fonksiyonu toplam bulanıklığı minimize etmektir. Wang'ın modelinin avantajı Tanaka'nın modeline göre daha dar

bir tahmin aralığı elde edilmesine karşın hesap karmaşıklığı Tanaka'nın modelinin hesap karmaşıklığıyla aynıdır;  $O(M^2N)$ .

#### 4.8. Nasibov'un Wabl Yöntemine Dayanan Uzaklık Tanımlaması Ve Bulanık Regresyon Modeli

Bulanık parametrelili problemlerin çözülmesi söz konusu olduğunda bulanık küme değerlerinin nokta değerinin belirlenmesi bir çözüm olacaktır. Dolayısıyla, böyle bir nokta değerinin mümkün olduğunca bulanık kümenin birçok karakteristiğini bünyesinde barındırması gereklidir. Bulanık değerlerin nokta değerlerinin oluşturulmasında minmax, ağırlık merkezi, üyelik fonksiyonunun maksimum değerini kullanan yöntem, Nasibov'un bulanık sayıların dekompozisyon gösterimine dayanan yaklaşımı gibi yöntemler kullanılır. Nasibov'un bu yöntemi her bir seviyeler kümesini kendi ağırlığı ile işler ve bulanık sayının sağ ve sol taraflarının önemlendirilmesini olanaklı hale getirir. Bulanık sayının ortalama gösterimini bulmak için karar verici tarafından yönlendirilen karar verme stratejisi, bulanık sayının taraflarının önem katsayılarına ve seviye kümesinin ağırlıklandırılma işlemine dayanır.

$F(E)$  bulanık uzayını gösterebilir. Herhangi bir  $A \in F(E)$  bulanık sayısının aşağıdaki LR gösterimi ile ifade edildiği varsayalım.

$$A = \bigcup_{\alpha \in [0,1]} (\alpha, A_\alpha) \quad (4.8.1)$$

Burada  $\forall \alpha \in [0,1]: A^\alpha = [L_A(\alpha), R_A(\alpha)] \subset (-\infty, \infty)$  'dır.

$L: [0, 1] \rightarrow (-\infty, \infty)$  monoton olarak azalmayan sürekli sol fonksiyon,

$R:[0, 1] \rightarrow (-\infty, \infty)$  monoton olarak artmayan sürekli sol fonksiyondur. Bir başka deyişle,  $L(a) = \mu_{\uparrow}^{-1}(\alpha)$ ,  $R(a) = \mu_{\downarrow}^{-1}(\alpha)$  'dır. Burada  $\mu_{\uparrow}^{-1}(\alpha)$ ,  $\mu_{\downarrow}^{-1}(\alpha)$   $\mu(t)$  fonksiyonunun sırasıyla artan ve azalan taraflarının ters fonksiyonlarıdır.

$\sup_{t \in E} \mu_A(t) = 1$  gibi bütün bulanık sayıların normalize edildiği varsayalım. Derecelerin öneminin dağılım yoğunluk fonksiyonu aşağıdaki normalizasyon koşulunu sağlayan negatif olmayan bir fonksiyondur.

$$P:[0, 1] \rightarrow [0, \infty).$$

$$\int_0^1 p(\alpha) d\alpha = 1 \quad (4.8.2)$$

**Tanım 1:** A bulanık sayısının seviyelere dayanan ağırlıklandırılmış ortalama gösterimi (WABL)

$$I(A) = c_L I_L(A) + c_R I_R(A) \quad (4.8.3)$$

şeklindedir. Burada  $c_L \geq 0, c_R \geq 0, c_L + c_R = 1$  ve

$$I_L(A) = \int_0^1 L_A(\alpha) p(\alpha) d\alpha \quad (4.8.4)$$

$$I_R(A) = \int_0^1 R_A(\alpha) p(\alpha) d\alpha \quad (4.8.5)$$

$c_R$  ve  $c_L$  katsayıları sırasıyla bulanık sayı üzerinde yapılan işlemlerde sol ve sağ tarafın önem derecesini göstermektedir.

**Tanım 2:** A bulanık sayısının ortalama genişliği,

$$D(A) = \int_0^1 (R_A(\alpha) - L_A(\alpha)) p(\alpha) d\alpha = I_R(A) - I_L(A) \quad (4.8.6)$$

şeklindedir. Burada  $I_L(a)$  ve  $I_R(a)$  sırasıyla yukarıda tanımlandığı gibidir. Burada aşağıdaki özellikler sağlanır.

$$1) \forall A \in F(E) = I(A) \in \text{supp}A;$$

2)  $\forall A \in F(E) = D(A) \geq 0$ . Eğer  $\forall \alpha \in (0,1] : p(\alpha) > 0$  ise, o zaman  $D(A) = 0 \Leftrightarrow \text{sup}A = a, a \in E$ ;

$$3) \forall A, B \in F(E) : A \subset B \Rightarrow D(A) \leq D(B).$$

**Tanım 3:** Nasibov'un 1. uzaklık tanımı

$$ND1(A, B) = \sqrt{[I(A) - I(B)]^2 + [D(A) - D(B)]^2} \quad (4.8.7)$$

şeklindedir.

Bu uzaklık tanımında,

$$\int_0^1 g(x) d(x) = \int_0^1 f(x) d(x)$$

şeklinde integralleri eşit olan iki fonksiyonun birbirine yaklaşık olarak eşit olduğu varsayılmaktadır ve bu ilişki  $\approx$  gibi gösterilir.

ND1 şu özellikleri sağlar.

- 1)  $ND1(A, B) \geq 0$ ,
- 2)  $A \approx B \Leftrightarrow ND1(A, B) = 0$ ,
- 3)  $ND1(A, B) + ND1(B, C) \geq ND1(A, C)$ ,
- 4)  $ND1(A, B) = ND1(B, A)$ .

**Tanım 4:** Nasibov'un 2.uzaklık tanımı,  $\int_0^1 g(x)d(x) = \int_0^1 f(x)d(x)$

varsayımı altında

$$ND2(A, B) = |I(A) - I(B)| + |D(A) - D(B)| \quad (4.8.8)$$

şeklindedir. Bu uzaklık tanımı aşağıdaki özellikleri sağlar.

- 1)  $ND2(A, B) \geq 0$ ,
- 2)  $A \approx B \Leftrightarrow ND2(A, B) = 0$ ,
- 3)  $ND2(A, B) + ND2(B, C) \geq ND2(A, C)$ ,
- 4)  $ND2(A, B) = ND2(B, A)$ .

Nasibov bu uzaklık tanımlarına dayanarak ND1'e dayanan bulanık en küçük kareler regresyon analizi ve ND2'ye dayanan bulanık en küçük mutlak sapmalar regresyon analizi modellerini ortaya atmıştır.

**4.8.1  $R(x)=L(x)=\max(0,1-|x|^s)$  kalıp fonksiyonuna sahip LR tipi bulanık sayılar için ND1'e dayanan bulanık en küçük kareler regresyon analizi**

Nasibov ve Bekçi (2004)'e göre, bulanık regresyon fonksiyonunun modelinin katsayılarının üyelik fonksiyonunun,

$$\mu_{A_j}(a) = \begin{cases} 1 - \left( \frac{A_j^M - a}{A_j^M - A_j^-} \right)^S, & A_j^- \leq a \leq A_j^M \\ 1 - \left( \frac{a - A_j^M}{A_j^+ - A_j^M} \right)^S, & A_j^M \leq a \leq A_j^+ \\ 0, & \text{aksihalde} \end{cases} \quad (4.8.1.1)$$

olduğu varsayalım.

Burada S bulanık sayının sağ ve sol şeklini veren 0'dan büyük bir sayıdır,  $A_j^M$  merkez,  $A_j^-$  ve  $A_j^+$  ise sırasıyla bulanık sayının sol ve sağ tarafının uç noktalarını göstermektedir. Buna göre model şöyle yazılabilir ( $i=1, \dots, N$ )

$$Y_i = (A_0^-, A_0^M, A_0^+)(+)(A_1^-, A_1^M, A_1^+)(\times) X_{i1}$$

$$(+)\dots\dots(+)(A_n^-, A_n^M, A_n^+)(\times) X_{in} (+)\epsilon_i$$

$$X_{ij} \geq 0 \text{ bulanık olamayan bir sayı, } Y_i$$

$$\mu_{Y_i}(y) = \begin{cases} 1 - \left( \frac{Y_i^M - y}{Y_i^M - Y_i^-} \right)^s, & Y_i^- \leq y \leq Y_i^M \\ \end{cases}$$

$$\begin{cases} 1 - \left( \frac{y - Y_i^M}{Y_i^+ - Y_i^M} \right)^s, & Y_i^M \leq y \leq Y_i^+ \end{cases}$$

0, aksi halde (4.8.1.2)

üyelik fonksiyonuna sahip olsun. Bağımlı değişkenin gözlem değeri  $Y_i$  'nin aşağıdaki gibi tanımlandığı varsayalım.

$$Y_i = (Y_i^-, Y_i^M, Y_i^+)$$

Burada  $Y_i^M$  merkez,  $Y_i^- = \inf\{y \mid \mu_{Y_i}(y) \geq 0\}$ ,  
 $Y_i^+ = \sup\{y \mid \mu_{Y_i}(y) \geq 0\}$ 'dir.

Buna göre bağımlı değişkenin gözlem değeri  $Y_i$ ,

$$Y_i = (Y_i^M, d_i^L, d_i^R)$$

şeklinde tekrar yazılır. Burada  $d_i^L = Y_i^M - Y_i^-$  ve  $d_i^R = Y_i^+ - Y_i^M$  bulanık bağımlı değişkenin gözlem değerinin sırasıyla sol ve sağ yayılmalarıdır.

Bağımlı değişkenin tahmin değeri  $Y_i^*$  'in aşağıdaki gibi tanımlandığı varsayalım.

$$Y_i^* = \left( \sum_{j=0}^n (\xi_j X_{ij} - \delta_j^L X_{ij}), \sum_{j=0}^n \xi_j X_{ij}, \sum_{j=0}^n (\xi_j X_{ij} + \delta_j^R X_{ij}) \right)$$

Burada  $\sum_{j=0}^n \xi_j X_{ij}$  merkez,  $\sum_{j=0}^n (\xi_j X_{ij} - \delta_j^L X_{ij})$  ve  $\sum_{j=0}^n (\xi_j X_{ij} + \delta_j^R X_{ij})$

ise  $\alpha = 0$  kesitinde sırasıyla en küçük ve en büyük nokta değerleridir,  $X_{i0} = 1$ 'dir.

Bağımlı değişkenin tahmin değeri  $Y_i^*$ ,

$Y_i^* = (\sum_{j=0}^n \xi_j X_{ij}, \sum_{j=0}^n \delta_j^L X_{ij}, \sum_{j=0}^n \delta_j^R X_{ij})$  şeklinde tekrar yazılabilir. Burada

$\sum_{j=0}^n \delta_j^L X_{ij}$ ,  $\sum_{j=0}^n \delta_j^R X_{ij}$  tahminlenen bulanık bağımlı değişkenin sırasıyla sol ve

sağ yayılmalarıdır,  $i=1, \dots, N, j=0, \dots, n$ .

$\sum_{j=0}^n \xi_j X_{ij}$ ,  $\sum_{j=0}^n (\xi_j X_{ij} - \delta_j^L X_{ij})$ ,  $\sum_{j=0}^n (\xi_j X_{ij} + \delta_j^R X_{ij})$  sırasıyla  $m_i$ ,  $c_i$ ,

$d_i$  sembolleri ile gösterilirse  $Y_i^*$ 'ın üyelik fonksiyonu aşağıdaki gibi tanımlanır.

$$\mu_{Y_i^*}(y) = \begin{cases} 1 - \left( \frac{m_i - y}{m_i - c_i} \right)^s, & c_i \leq y \leq m_i \\ \\ \\ \end{cases}$$

$$= 1 - \left( \frac{y - d_i}{d_i - m_i} \right)^s, m_i \leq y \leq d_i$$

$$= 0, \text{ aksi halde} \quad (4.8.1.3)$$

$\varepsilon_i$ ; bağımlı değişkenin gözlem ile tahmin değeri arasındaki bulanık uzaklığı verir,



$$\varepsilon_i = Y_i(-)Y_i^*$$

şeklinde tanımlanır ve bulanık hata olarak adlandırılır. Klasik regresyona benzer şekilde en küçük kareler çözümünü bulmak için, bulanık hataların karelerinin toplamının minimize edilmesi gerekir.

$$\text{Min} \sum_{i=1}^N \varepsilon_i^2 = \sum_{i=1}^N (Y_i(-)Y_i^*)^2$$

Burada N veri sayısını ifade etmektedir. Bağımlı değişkenin gözlem ile tahmin değeri arasındaki uzaklığın hesaplanmasında ND1 uzaklığı dikkate alınarak minimizasyon problemi aşağıdaki gibi tanımlanır.

$$\text{Min} \left\{ \sum_{i=1}^N ([I(Y_i) - I(Y_i^*)]^2 + [D(Y_i) - D(Y_i^*)]^2) \right\}$$

Burada  $I(Y_i)$  ve  $I(Y_i^*)$ , bağımlı değişkenin sırasıyla gözlem ve tahmin değerlerinin ortalama gösterim değerleridir.  $D(Y_i)$  ve  $D(Y_i^*)$  ise bağımlı değişkenin sırasıyla gözlem ve tahmin değerlerinin genişlik değerleridir.

Derecelerin önem yoğunluk dağılış fonksiyonu aşağıdaki normalizasyon koşulunu sağlar. Bu normalizasyon koşulunu dikkate alarak  $p(a)$ ,

$$P(a) = (k+1)a^k$$

şeklinde tanımlanır. Burada k bir doğal sayıdır.  $p(a)$ , karar vericinin a seviyeler kümesi üzerinde çeşitli şekilde ağırlıklandırma yapmasını olanaklı hale getirir.

Nasibov E., derecelerin önem dağılışı yoğunluk fonksiyonlarının seçiminde aşağıdaki temel fonksiyonları önermektedir.

1) Bütün  $\alpha$  seviye kümelerine eşit önem verildiğini gösteren  $p(\alpha)=1$  ( $k=0$ ),

2) Üyelik derecesi artışı oranında ilgili  $\alpha$  seviye kümelerine ağırlıklandırma yapıldığını belirten  $p(\alpha)=2\alpha$  ( $k=1$ ),

3) Üyelik derecesi arttıkça  $\alpha$  seviye kümelerinin öneminin karesel olarak arttığını gösteren  $p(\alpha)=3\alpha^2$  ( $k=2$ ).

$\mu_{Y_i}^*(y)$  eşitliği kullanılarak  $L_{Y_i}(\alpha)$  ve  $R_{Y_i}(\alpha)$  sırasıyla,

$$L_{Y_i}(\alpha) = Y_i^m - (1 - \alpha)^{1/S} d_i^L$$

$$R_{Y_i}(\alpha) = Y_i^m + (1 - \alpha)^{1/S} d_i^R$$

şeklinde yazılır. Buradan,  $I(Y_i)$  aşağıdaki gibi elde edilir.

$$\begin{aligned} I(Y_i) &= c_L \int_0^1 [Y_i^m - (1 - \alpha)^{1/S} d_i^L] (1 + k) \alpha^k d\alpha \\ &+ c_R \int_0^1 [Y_i^m + (1 - \alpha)^{1/S} d_i^R] (1 + k) \alpha^k d\alpha \\ &= c_L Y_i^m - c_L (1 + k) d_i^L \int_0^1 \alpha^k (1 - \alpha)^{1/S} d\alpha \end{aligned}$$

$$+ c_R Y_i^m + c_R (1+k) d_i^R \int_0^1 \alpha^k (1-\alpha)^{1/s} d\alpha$$

Burada,  $\int_0^1 \alpha^k (1-\alpha)^{1/s}$  integrali  $B(1+k, 1+\frac{1}{s})$  şeklinde gösterilebilen

bir beta fonksiyonudur. Bu beta fonksiyonu da gamma fonksiyonu kullanılarak,

$$B(1+k, 1+\frac{1}{s}) = \frac{\Gamma(1+k)\Gamma(1+\frac{1}{s})}{\Gamma(2+k+\frac{1}{s})}$$

şeklinde bulunur. Gamma fonksiyonu,

$$\Gamma(b) = \int_0^{\infty} y^{b-1} e^{-y} dy$$

şeklinde tanımlanan bir fonksiyondur.

Beta fonksiyonu kullanılarak  $I(Y_i)$ ,

$$I(Y_i) = Y_i^m + \frac{(c_R d_i^R - c_L d_i^L) \Gamma[1 + \frac{1}{s}]}{\Gamma[2 + k + \frac{1}{s}]}$$

şeklinde elde edilir. Burada  $(1+k)\Gamma(1+k) = \Gamma(2+k)$  eşitliği kullanılmıştır.

Bağımlı değişkenin tahmin değerinin ortalama gösterimi,

$$L_{Y_i}^* = \sum_{j=0}^n [\xi_j X_{ij} - (1-\alpha)^{1/S} \delta_j^L X_{ij}]$$

$$R_{Y_i}^* = \sum_{j=0}^n [\xi_j X_{ij} + (1-\alpha)^{1/S} \delta_j^R X_{ij}]$$

tanımlarıyla bulunur,  $i=1, \dots, N$ ,  $X_{i1}=0$ .

Derecelerin önem dağılışı fonksiyonundaki normalizasyon koşulu dikkate alınarak yazılırsa  $I(Y_i^*)$ ,

$$I(Y_i^*) = c_L \int_0^1 \left[ \sum_{j=0}^n (\xi_j X_{ij} - (1-\alpha)^{1/S} \delta_j^L X_{ij}) \right] (k+1) \alpha^k d\alpha$$

$$+ c_R \int_0^1 \left[ \sum_{j=0}^n (\xi_j X_{ij} + (1-\alpha)^{1/S} \delta_j^R X_{ij}) \right] (k+1) \alpha^k d\alpha$$

$$= c_L \sum_{j=0}^n \xi_j X_{ij} - c_L (1+k) \int_0^1 \left[ \sum_{j=0}^n (\delta_j^L X_{ij}) (1-\alpha)^{1/S} \right] \alpha^k d\alpha$$

$$+ c_R \sum_{j=0}^n \xi_j X_{ij} + c_R (1+k) \int_0^1 \left[ \sum_{j=0}^n (\delta_j^R X_{ij}) (1-\alpha)^{1/S} \right] \alpha^k d\alpha$$

$$= c_L \sum_{j=0}^n \xi_j X_{ij} - c_L (1+k) \sum_{j=0}^n \delta_j^L X_{ij} \int_0^1 (1-\alpha)^{1/S} \alpha^k d\alpha$$

$$= c_R \sum_{j=0}^n \xi_j X_{ij} + c_R (1+k) \sum_{j=0}^n \delta_j^R X_{ij} \int_0^1 (1-\alpha)^{1/S} \alpha^k d\alpha$$

şeklinde eşitlik elde edilir. Beta ve gamma fonksiyonları kullanılarak  $I(Y_i^*)$ ,

$$I(Y_i^*) = \sum_{j=0}^n \xi_j X_{ij} + \frac{\sum_{j=0}^n (c_R \delta_j^R X_{ij} - c_L \delta_j^L X_{ij}) \Gamma[2+k] \Gamma[1 + \frac{1}{s}]}{\Gamma[2+k + \frac{1}{s}]}$$

elde edilir. Bağımlı değişkenin gözlem ve tahmin değerlerinin ortalama genişlikleri olan  $D(Y_i)$  ve  $D(Y_i^*)$  sırasıyla aşağıdaki gibi yazılabilir.

$$D(Y_i) = \int_0^1 (R_{Y_i}(\alpha) - L_{Y_i}(\alpha)) p(\alpha) d\alpha = I_R(Y_i) - I_L(Y_i)$$

$$D(Y_i^*) = \int_0^1 (R_{Y_i^*}(\alpha) - L_{Y_i^*}(\alpha)) p(\alpha) d\alpha = I_R(Y_i^*) - I_L(Y_i^*)$$

şeklinde tanımlanır. Burada  $p(\alpha)$  şekil'de tanımlandığı gibi ve  $k=0,1,\dots$ 'dir.

Buradan

$$\begin{aligned} D(Y_i) \quad D(Y_i) &= \int_0^1 (Y_i^m + (1-\alpha)^{1/S} d_i^R - (Y_i^m - (1-\alpha)^{1/S} d_i^L))(1+k)\alpha^k d\alpha \\ &= \int_0^1 ((1-\alpha)^{1/S} d_i^R + (1-\alpha)^{1/S} d_i^L)(1+k)\alpha^k d\alpha \\ &= d_i^R (1+k) \int_0^1 (1-\alpha)^{1/S} \alpha^k d\alpha + d_i^L (1+k) \int_0^1 (1-\alpha)^{1/S} \alpha^k d\alpha \end{aligned}$$

şeklinde elde edilir. Gamma ve beta fonksiyonları kullanılarak  $D(Y_i)$  aşağıdaki gibi elde edilebilir.

$$D(Y_i) = \frac{(d_i^R + d_i^L)\Gamma[2+k]\Gamma[1+\frac{1}{s}]}{\Gamma[2+k+\frac{1}{s}]}$$

$$\begin{aligned} D(Y_i^*) &= \int_0^1 l \sum_{j=0}^n (\xi_j X_{ij} + (1-\alpha)^{1/s} \delta_j^R X_{ij}) - \sum_{j=0}^n (\xi_j X_{ij} - (1-\alpha)^{1/s} \delta_j^L X_{ij}) (1+k) \alpha^k d\alpha \\ &+ \int_0^1 [\sum_{j=0}^n ((1-\alpha)^{1/s} \delta_j^R X_{ij} + (1-\alpha)^{1/s} \delta_j^L X_{ij})] (1+k) \alpha^k d\alpha \\ &= (1+k) \sum_{j=0}^n \delta_j^R X_{ij} \int_0^1 (1-\alpha)^{1/s} \alpha^k d\alpha + (1+k) \sum_{j=0}^n \delta_j^L X_{ij} \int_0^1 (1-\alpha)^{1/s} \alpha^k d\alpha \end{aligned}$$

$D(Y_i)$  'ye benzer şekilde elde edilir.

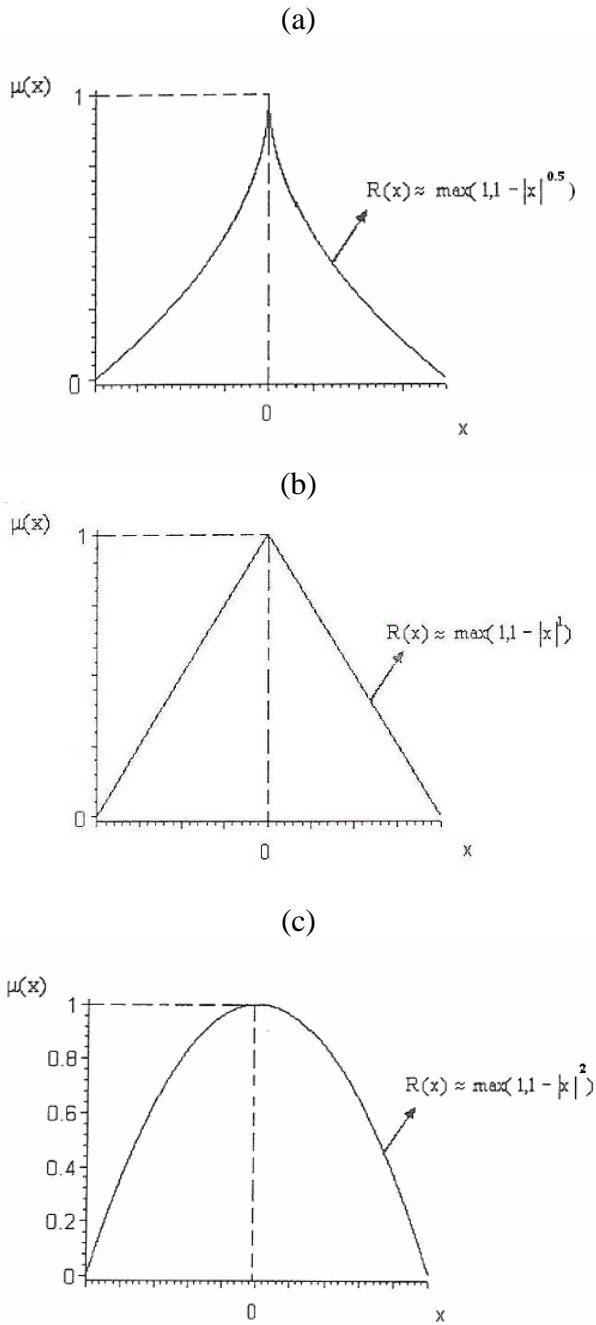
Bağımlı değişkenin gözlem ile tahmin değeri arasındaki uzaklığın hesaplanmasında ND1 dikkate alınarak, bulanık en küçük kareler regresyon parametreleri aşağıdaki minimizasyon problemi ile elde edilir.

Min

$$\begin{aligned} &\left\{ \sum_{i=1}^n [(Y_i^m - \sum_{j=0}^n \xi_j X_{ij} + \Phi(k,s) \sum_{j=0}^n [c_R (d_i^R - \delta_j^R X_{ij}) + c_L (-d_i^L + \delta_j^L X_{ij})])^2 \right. \\ &\left. + (\Phi(k,s) \sum_{j=0}^n ((d_i^R - \delta_j^R X_{ij}) + (d_i^L - \delta_j^L X_{ij}))^2] \right\} \end{aligned}$$

$$\text{Burada } \Phi(k, s) = \frac{\Gamma[2+k]\Gamma[1+\frac{1}{s}]}{\Gamma[2+k+\frac{1}{s}]} \text{ 'dır.}$$

Bu minimizasyon probleminde  $\delta_j^R \geq 0, \delta_j^L \geq 0, \xi_j \geq 0$  koşulları aranır. Buradaki  $s$ , üyelik fonksiyonunun şeklini veren bir sayıdır ( $s \geq 0$ ).  $R(x)=L(x)=\max(0, 1-|x|^s)$  kalıp fonksiyonuna sahip bulanık sayının üyelik fonksiyonunun değişik  $s$  değerlerinde aldığı biçimler şekil 4.4(a), 4.4(b), 4.4(c)'de gösterildiği gibidir.



**ŞEKİL 4.4:**  $R(x)=L(x)=\max(0, 1-|x|^s)$  kalıbı fonksiyonuna sahip bulanık sayıların üyelik fonksiyonları (a)  $s=1/2$ , (b)  $s=1$ , (c)  $s=2$ .



**4.8.2  $R(x)=L(x)=e^{-|x|^s} e^{-|x|^s}$  Kalıp fonksiyonuna sahip LR tip bulanık sayılar için ND1'e dayanan bulanık en küçük kareler regresyon modeli**

Bulanık regresyon modelinin kat sayılarının üyelik fonksiyonunun

$$\text{Exp} \left[ -\left( \frac{A_j^m - a}{A_j^m - A_j^-} \right)^s \right], A_j^- \leq a \leq A_j^m$$

$$\mu_{A_j}(a) = \left\{ \text{Exp} \left[ -\left( \frac{a - A_j^m}{A_j^+ - A_j^m} \right)^s \right], A_j^m \leq a \leq A_j^+ \right.$$

0, aksihalde

olduğu varsayılmaktadır. Burada  $A_j^m$  merkez,  $A_j^-, A_j^+$  sırasıyla sol ve sağ tarafın uç noktalarıdır. Bu üyelik fonksiyonuna sahip katsayıları olan model

$$Y_i = (A_0^-, A_0^M, A_0^+)(+)(A_1^-, A_1^M, A_1^+)(\times)X_{i1}$$

$$(+)\dots\dots(+)(A_n^-, A_n^M, A_n^+)(\times)X_{in}(+)\varepsilon_i$$

'dir.

$X_{ij} \geq 0$  bulanık olmayan bir sayı ve  $Y_i$  aşağıdaki üyelik fonksiyonuna sahip olduğu varsayılışın.

$$\text{Exp} \left[ -\left( \frac{Y_i^m - y}{Y_i^m - Y_i^-} \right)^s \right], Y_i^- \leq y \leq Y_i^m$$

$$\mu_{Y_i}^*(y) = \left\{ \text{Exp} \left[ - \left( \frac{Y_i^m - y}{Y_i^m - Y_i^-} \right)^S \right], Y_i^- \leq y \leq Y_i^+ \right.$$

0, aksihalde

$L(x) = e^{-|x|^S}$  kalıp fonksiyonuna sahip bağımlı değişkenin gözlem ve tahmin değerlerinin WABL ve ortalama aralık değerlerini bulmak için gerekli olan üyelik fonksiyonlarının sol ve sağ taraflarının ters fonksiyonları,

$$L_{Y_i}(\alpha) = (Y_i^m - (In \frac{1}{\alpha})^{1/S} d_i^L)$$

$$R_{Y_i}(\alpha) = (Y_i^m + (In \frac{1}{\alpha})^{1/S} d_i^R)$$

$$L_{Y_i}^*(\alpha) = \sum_{j=0}^n (\xi_j X_{ij} - (In \frac{1}{\alpha})^{1/S} \delta_j^L X_{ij})$$

$$R_{Y_i}^*(\alpha) = \sum_{j=0}^n (\xi_j X_{ij} + (In \frac{1}{\alpha})^{1/S} \delta_j^R X_{ij})$$

şeklindedir.  $i=1, \dots, N, X_{i1}=0$ .

Bu eşitliklerden yararlanarak  $I(Y_i)$  ve  $I(Y_i^*)$  aşağıdaki gibi elde edilmektedir.

$$I(Y_i) = c_L \int_0^1 [Y_i^m - (In \frac{1}{\alpha})^{1/S} d_i^L] (1+k) \alpha^k d\alpha +$$

$$\begin{aligned}
& c_R \int_0^1 [Y_i^m + (\ln \frac{1}{\alpha})^{1/S} d_i^R] (1+k) \alpha^k d\alpha \\
&= c_L Y_i^m - c_L (1+k) d_i^L \int_0^1 (\ln \frac{1}{\alpha})^{1/S} \alpha^k d\alpha \\
&+ c_R Y_i^m + c_R (1+k) d_i^R \int_0^1 (\ln \frac{1}{\alpha})^{1/S} \alpha^k d\alpha \\
&= Y_i^m + (c_R d_i^R - c_L d_i^L) (1+k)^{-1/S} \Gamma[1 + \frac{1}{S}] \\
& I(Y_i^*) = \sum_{j=0}^n \xi_j X_{ij} + \sum_{j=0}^n (c_R \delta_j^R X_{ij} - c_L \delta_j^L X_{ij}) (1+k)^{-1/S} \Gamma[1 + \frac{1}{S}]
\end{aligned}$$

şeklinde bulunur.

$D(Y_i)$  ve  $D(Y_i^*)$  sırasıyla;

$$D(Y_i) = (d_i^R + d_i^L) (1+k)^{-1/S} \Gamma[1 + \frac{1}{S}]$$

$$D(Y_i^*) = \sum_{j=0}^n (\delta_j^R X_{ij} + \delta_j^L X_{ij}) (1+k)^{-1/S} \Gamma[1 + \frac{1}{S}] \text{ 'dir.}$$

Bağımlı değişkenin gözlem ile tahmin değeri arasındaki uzaklığın hesaplanmasında ND1 dikkate alınarak, bulanık en küçük kareler regresyon parametreleri aşağıdaki minimizasyon problemi ile elde edilir.

Min

$$\left\{ \sum_{i=1}^n \left[ \left( \sum_{j=0}^n Y_i^m - \xi_j X_{ij} + \Psi(s) [c_R (d_i^R - \delta_j^R X_{ij}) + c_L (-d_i^L + \delta_j^L X_{ij})] \right) (1+k)^{-1/s} \right]^2 \right.$$

$$\left. + (\Psi(s) \sum_{j=0}^n [(d_i^R - \delta_j^R X_{ij}) + (d_i^L - \delta_j^L X_{ij}) + (d_i^L - \delta_j^L X_{ij})] (1+k)^{-1/s} )^2 \right]$$

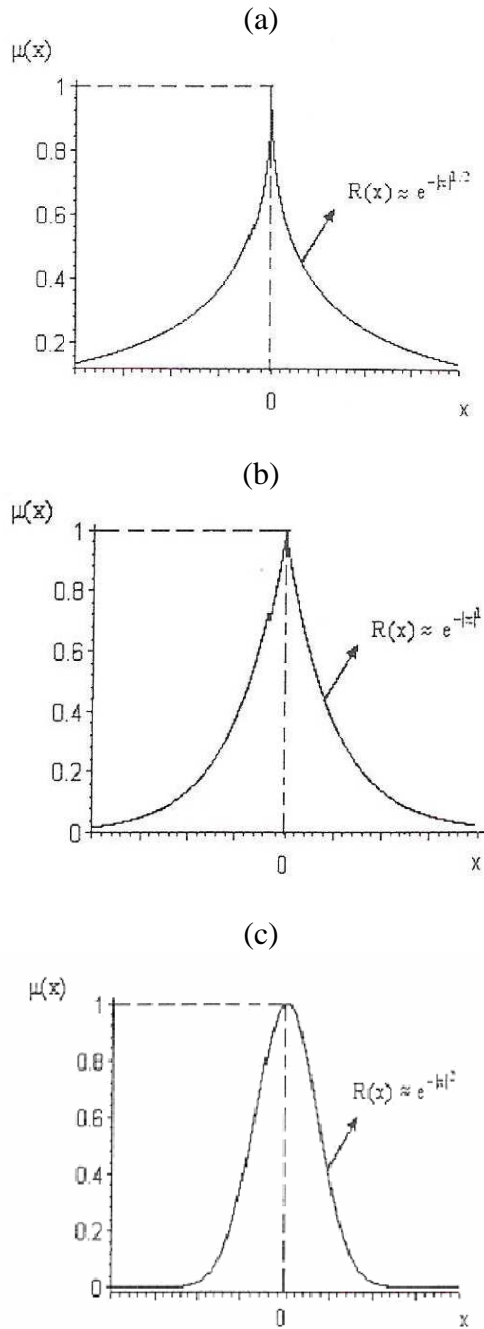
$$\Psi(s) = \Gamma\left[1 + \frac{1}{s}\right], \delta_j^R \geq 0, \delta_j^L \geq 0, \xi_j \geq 0 \text{ 'dır.}$$

S üyelik fonksiyonunun şeklini veren bir sayıdır ve  $s \geq 0$  'dır. Farklı k ve s değerlerine göre minimizasyon problemi farklı şekilde gösterilmektedir. Örneğin k=2 ve s=1 için minimizasyon problemi,

$$\left\{ \sum_{i=1}^N \left[ \left( \sum_{j=0}^n (Y_i^m - \xi_j x_{ij} + \frac{1}{3} [c_R (d_i^R - \delta_j^R x_{ij}) + c_L (-d_i^L + \delta_j^L x_{ij})]) \right) \right]^2 \right.$$

$$\left. + \left( \frac{1}{3} \sum_{j=0}^n [(d_i^R - \delta_j^R x_{ij}) + (d_i^L - \delta_j^L x_{ij})] \right)^2 \right\}$$

şeklinde gösterilebilmektedir. Bu minimizasyon probleminde s üyelik değerinin şeklini veren bir sayıdır. Farklı s değerleri için LR tipi bulanık sayının üyelik fonksiyon şekilleri Şekil 4.5(a), (b), (c)'de gösterilmiştir.



**ŞEKİL 4.5:**  $R(x)=L(x)=e^{-|x|^s}$  tipi kalıp fonksiyonuna sahip bulanık sayıların üyelik fonksiyonları (a)  $s=1/2$ , (b)  $s=1$ , (c)  $s=2$ .

### 4.8.3 $R(x)=L(x)=1/(1+|x|^s)$ Kalıp fonksiyonuna sahip LR tip bulanık sayılar için ND1'e dayanan bulanık en küçük kareler regresyon modeli

Nasibov, ayrıca farklı kalıp fonksiyonlara ait bulanık sayılar içinde ND1'e dayanan bulanık en küçük kareler regresyon modelleri geliştirmiştir.  $R(x)=L(x)=1/(1+|x|^s)$  kalıp fonksiyonuna sahip LR tipi bulanık sayılar için ND1'e dayanan bulanık en küçük kareler regresyon modeli parametreleri aşağıdaki minimizasyon problemi ile elde edilir.

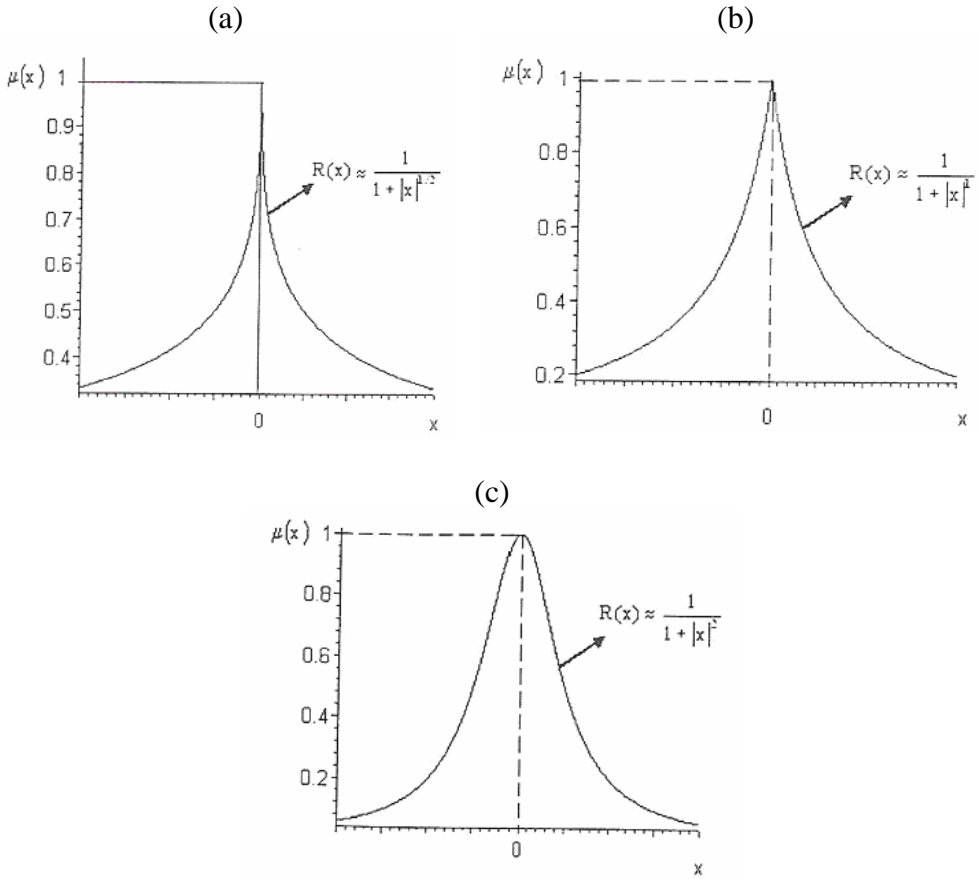
$$\begin{aligned} & \text{Min} \left( \sum_{i=1}^N \left[ \left( \sum_{j=0}^n (Y_i^m - \xi_j X_{ij}) \right. \right. \right. \\ & \left. \left. \left. + \Theta(k, s) [c_R (d_i^R - \delta_j^R X_{ij}) + c_L (-d_i^L + \delta_j^L X_{ij})] \right) \right]^2 \right. \\ & \left. + \Theta(k, s) \left( \sum_{j=0}^n [(d_i^R - \delta_j^R X_{ij}) + (d_i^L - \delta_j^L X_{ij})]^2 \right) \right) \\ & \Theta(k, s) = \frac{\Gamma[1+k - \frac{1}{s}] \Gamma[1 + \frac{1}{s}]}{\Gamma[1+k]} \text{ 'dir. } \delta_j^R \geq 0, \delta_j^L \geq 0, \xi_j \geq 0 \text{ koşulları} \end{aligned}$$

aranır. Problemin çalışabilmesi için  $1+k - \frac{1}{s} > 0$  varsayımının sağlanması gerekmektedir. Farklı k ve s değerlerine göre minimizasyon problemi farklı şekilde gösterilmektedir. Örneğin k=2 ve s=1 için minimizasyon problemi,

$$\left\{ \sum_{i=1}^N \left( \sum_{j=0}^n (Y_i^m - \xi_j x_{ij} + [c_R (d_i^R - \delta_j^R x_{ij}) + c_L (-d_i^L + \delta_j^L x_{ij})]) \right)^2 \right.$$

$$+ \sum_{i=1}^N \left( \sum_{j=0}^n [(d_i^R - \delta_j^R x_{ij}) + (d_i^L - \delta_j^L x_{ij})] \right)^2 \}$$

şeklinde gösterilebilmektedir. Bu minimizasyon probleminde  $s$  üyelik değerinin şeklini veren bir sayıdır. Farklı  $s$  değerleri için LR tipi bulanık sayının üyelik fonksiyon şekilleri Şekil 4.6(a), (b), (c)'de gösterilmiştir.



**ŞEKİL 4.6:**  $R(x)=L(x)=1/(1+|x|^s)$  tipi kalıp fonksiyonuna sahip bulanık sayıların üyelik fonksiyonları (a)  $s=1/2$ , (b)  $s=1$ , (c)  $s=2$ .

#### 4.8.4 $R(x)=L(x)=\max(0,1-|x|^s)$ kalıp fonksiyonuna sahip bulanık sayılar için ND2'ye dayanan bulanık en küçük mutlak sapmalar regresyon analizi

ND2'ye dayanan bulanık en küçük mutlak sapmalar regresyon modelinde çözüm  $\varepsilon_i = Y_i \{-\hat{Y}_i\}$ 'da tanımlanan bulanık hataların mutlak değerinin toplamalarının minimizasyonundan elde edilmektedir.

$$\text{Min} \sum_{i=1}^N |\varepsilon_i| = \sum_{i=1}^N |Y_i - \hat{Y}_i|$$

N=örneklem ölçümü

Nasibov'un ND2 uzaklık tanımlaması kullanılarak (yukarıdaki)'de yer alan minimizasyon problemi,

$$\text{Min} \left\{ \sum_{i=1}^N (|I(Y_i) - I(Y_i^*)| + |D(Y_i) - D(Y_i^*)|) \right\}$$

şeklinde yazılabilmektedir. Buradaki  $I(Y_i)$  ve  $I(Y_i^*)$ , sırasıyla bağımlı değişkenin gözlem ve tahmin değerlerinin seviyelere dayalı ortalama gösterim değerleridir.  $D(Y_i)$  ve  $D(Y_i^*)$  ise bağımlı değişkenin sırasıyla gözlem ve tahmin değerlerinin ortalama genişlik değerleridir.

Regresyon katsayısı ve hata terimi bulanık olan bulanık regresyon modeli,



$$Y_i = (A_0^-, A_0^M, A_0^+)(+)(A_1^-, A_1^M, A_1^+)(\times)X_{i1}$$

$$(+).....(+)(A_n^-, A_n^M, A_n^+)(\times)X_{in} (+)\varepsilon_i$$

$X_{ij} \geq 0$  bulanık olamayan bir sayı,  $Y_i$

$$\mu_{Y_i}(y) = \begin{cases} 1 - \left( \frac{Y_i^M - y}{Y_i^M - Y_i^-} \right)^s, & Y_i^- \leq y \leq Y_i^M \end{cases}$$

$$\begin{cases} 1 - \left( \frac{y - Y_i^M}{Y_i^+ - Y_i^M} \right)^s, & Y_i^M \leq y \leq Y_i^+ \end{cases}$$

0, aksihalde

Bulanık regresyon fonksiyonunun modelinin katsayılarının üyelik fonksiyonunun,

$$\mu_{A_j}(a) = \begin{cases} 1 - \left( \frac{A_j^M - a}{A_j^M - A_j^-} \right)^s, & A_j^- \leq a \leq A_j^M \end{cases}$$

$$1 - \left( \frac{a - A_j^M}{A_j^+ - A_j^M} \right)^s, \quad A_j^M \leq a \leq A_j^+$$

0, aksihalde .

olduğu varsayılmaktadır. Ayrıca  $I(Y_i)$ ,  $I(Y_i^*)$ ,  $D(Y_i)$ ,  $D(Y_i^*)$  ;

$$I(Y_i) = Y_i^m + \frac{(c_R d_i^R - c_L d_i^L) \Gamma[1 + \frac{1}{s}]}{\Gamma[2 + k + \frac{1}{s}]},$$

$$I(Y_i^*) = \sum_{j=0}^n \xi_j X_{ij} + \frac{\sum_{j=0}^n (c_R \delta_j^R X_{ij} - c_L \delta_j^L X_{ij}) \Gamma[2 + k] \Gamma[1 + \frac{1}{s}]}{\Gamma[2 + k + \frac{1}{s}]},$$

$$D(Y_i) = \frac{(d_i^R + d_i^L) \Gamma[2 + k] \Gamma[1 + \frac{1}{s}]}{\Gamma[2 + k + \frac{1}{s}]},$$

$$D(Y_i^*) = (1 + k) \sum_{j=0}^n \delta_j^R X_{ij} \int_0^1 (1 - \alpha)^{1/s} \alpha^k d\alpha +$$

$$(1 + k) \sum_{j=0}^n \delta_j^L X_{ij} \int_0^1 (1 - \alpha)^{1/s} \alpha^k d\alpha$$

şeklinde ifade edilmektedir.

Bağımlı değişkenin gözlem ve tahmin değeri arasındaki uzaklığın hesaplanmasında ND2 kullanılarak, bulanık en küçük sapmalar regresyon modelinin parametreleri aşağıdaki minimizasyon problemi ile elde edilmektedir.

$$\text{Min} \left\{ \sum_{i=1}^N (|Y_i^m - \sum_{j=0}^n \xi_j x_{ij} +$$

$$\Phi(k, s) \sum_{j=0}^n c_R (d_i^R - \delta_j^R x_{ij}) + c_L (-d_i^L + \delta_j^L x_{ij}) \Big| + |\Phi(k, s) \sum_{j=0}^n [(d_i^R - \delta_j^R x_{ij}) + (d_i^L - \delta_j^L x_{ij})] \Big| \Big\}$$

$$\text{Burada } \Phi(k, s) = \frac{\Gamma[2+k]\Gamma[1+\frac{1}{s}]}{\Gamma[2+k+\frac{1}{s}]}, \text{ dır.}$$

Bu minimizasyon probleminde  $\delta_j^R, \delta_j^L, \xi_j \geq 0$  koşulları sağlanmalıdır

Bu minimizasyon probleminde s, üyelik fonksiyonun şeklini veren bir sayıdır ve k karar vericinin  $\alpha$  -seviyeler kümesi üzerinde çeşitli şekilde ağırlıklandırma yapmasını olanaklı hale getirmektedir. Nasibov (2002), derecelerin önem dağılışı fonksiyonlarının seçimi sırasında aşağıdaki temel fonksiyonları önermektedir.

1. Bütün  $\alpha$  -seviye kümesine eşit önem verildiğini gösteren  $p(\alpha)=1$  (k=0) fonksiyonu.

2. Üyelik derecesi artışı oranında ilgili  $\alpha$  -seviye kümelerine ağırlıklandırma yapıldığını belirten  $p(\alpha)=2\alpha$  (k=1) fonksiyonu.

3. Üyelik derecesi arttıkça  $\alpha$  -seviye kümelerinin öneminin karesel olarak arttığını gösteren  $p(\alpha)=3\alpha^2$  (k=2) fonksiyonu.

Farklı s ve k değerleri için minimizasyon problemi de farklı olmaktadır. Örneğin k=2 ve s=1 için minimizasyon problemi,

$$\left\{ \sum_{i=1}^N (|Y_i^m - \sum_{j=0}^n \xi_j x_{ij}| + \frac{1}{4} \sum_{j=0}^n [c_R (d_i^R - \delta_j^R x_{ij}) + c_L (-d_i^L + \delta_j^L x_{ij})]) \right\} \\ + \left| \frac{1}{4} \sum_{j=0}^n [(d_i^R - \delta_j^R x_{ij}) + (d_i^L - \delta_j^L x_{ij})] \right\}$$

şeklinde gösterilebilmektedir.

#### 4.8.5 $R(x)=L(x)=e^{-|x|^s}$ Kalıp fonksiyonuna sahip bulanık sayılar için ND2'ye dayanan bulanık en küçük mutlak sapmalar regresyon modeli

Bu kalıptaki regresyon modeli,

$$\text{Min} \left\{ \sum_{i=1}^N \left| \left( \sum_{j=0}^n [Y_i^m - \xi_j x_{ij}] \right. \right. \right. \\ \left. \left. \left. + \psi(s) [c_R (d_i^R - \delta_j^R x_{ij}) + c_L (-d_i^L + \delta_j^L x_{ij})] (1+k)^{\frac{1}{s}} \right| \right. \\ \left. \left. \left. + \left| \psi(s) \sum_{j=0}^n [(d_i^R - \delta_j^R x_{ij}) + (d_i^L - \delta_j^L x_{ij})] (1+k)^{\frac{1}{s}} \right| \right\}$$

Burada,

$$\psi(s) = \Gamma\left[1 + \frac{1}{s}\right] \text{ 'dir.}$$

Bu minimizasyon probleminde  $\delta_j^R, \delta_j^L, \xi_j \geq 0$  koşulları sağlanmalıdır. S üyelik fonksiyonunun şeklini veren bir sayıdır. Farklı s ve k

değerleri için minimizasyon problemi de farklı olmaktadır. Örneğin  $k=2$  ve  $s=1$  için minimizasyon problemi,

$$\left\{ \sum_{i=1}^N \left| \left( \sum_{j=0}^n [Y_i^m - \xi_j x_{ij}] + \frac{1}{3} [c_R (d_i^R - \delta_j^R x_{ij}) + c_L (-d_i^L + \delta_j^L x_{ij})] \right) \right| \right. \\ \left. + \frac{1}{3} \sum_{j=0}^n [(d_i^R - \delta_j^R x_{ij}) + (d_i^L - \delta_j^L x_{ij})] \right\}$$

şeklinde gösterilebilmektedir.

#### 4.8.6 $R(x)=L(x)= 1/(1+|x|^s)$ Kalıp fonksiyonuna sahip bulanık sayılar için ND2'ye dayanan bulanık en küçük mutlak sapmalar regresyon modeli

$R(x)=L(x)=1/(1+|x|^s)$  kalıp fonksiyonuna sahip bulanık sayılar için ND2'ye dayanan bulanık en küçük mutlak sapmalar regresyon modelinin parametreleri aşağıdaki minimizasyon problemi ile elde edilir.

$$\text{Min} \left\{ \sum_{i=1}^N \left| \left( \sum_{j=0}^n (Y_i^m - \xi_j x_{ij}) \right. \right. \right. \\ \left. \left. + \Theta(k, s) [c_R (d_i^R - \delta_j^R x_{ij}) + c_L (-d_i^L + \delta_j^L x_{ij})] \right) \right| \\ \left. + \Theta(k, s) \sum_{j=0}^n [(d_i^R - \delta_j^R x_{ij}) + (d_i^L - \delta_j^L x_{ij})] \right\}$$

$$\text{Burada } \Theta(k, s) = \frac{\Gamma[1+k-\frac{1}{s}]\Gamma[1+\frac{1}{s}]}{\Gamma[1+k]}, \text{ dir.}$$

Bu minimizasyon probleminde,  $\delta_j^R \geq 0$ ,  $\delta_j^L \geq 0$ ,  $\xi_j \geq 0$  koşulları aranır. Problemin çalışabilmesi için  $1 + k - \frac{1}{s} > 0$  varsayımının sağlanması gerekmektedir. Farklı k ve s değerlerine göre minimizasyon problemi farklı biçimde ifade edilmektedir. Örneğin k=2 ve s=1 için minimizasyon problemi,

$$\left\{ \sum_{i=1}^N \left( \left| \sum_{j=0}^n (Y_i^m - \xi_j x_{ij} + [c_R(d_i^R - \delta_j^R x_{ij}) + c_L(-d_i^L + \delta_j^L x_{ij})]) \right| \right. \right. \\ \left. \left. + \left| \sum_{j=0}^n [(d_i^R - \delta_j^R x_{ij}) + (d_i^L - \delta_j^L x_{ij})] \right| \right) \right\}$$

şeklinde gösterilebilmektedir.

#### 4.8.7 WABL determinasyon katsayısı

Klasik regresyon analizinde  $R^2$  sembolü ile gösterilen determinasyon katsayısı doğrusal regresyon modelinin verileri ne ölçüde yansıttığının bir ifadesi olarak tanımlanmaktadır. Veri noktalarının hepsinin regresyon modelinin üzerine düşmesi halinde determinasyon katsayısı 1 olarak bulunacaktır. Regresyon modeli veri noktalarından ne kadar uzak olursa, determinasyon katsayısının değeri de o kadar 0'a doğru yaklaşmaktadır. Bu nedenle doğrusal durumlar için determinasyon katsayısının regresyon modelinin geçerliliğinin bir ölçüsü olduğu söylenebilmektedir.

$Y$ ; bağımlı değişkenin gözlem değeri ve  $\hat{Y}$ ; bağımlı değişkenin tahmin değeri,  $\bar{Y}$ ; ise bağımlı değişkenin gözlem değerinin ortalaması olarak gösterildiğinde, klasik regresyon modelinde

$$\sum_{i=1}^N (Y_i - \bar{Y})^2 = \sum_{i=1}^N (\hat{Y}_i - \bar{Y})^2 + \sum_{i=1}^N (Y_i - \hat{Y}_i)^2$$

eşitliği yazılabilmektedir. Burada  $\sum_{i=1}^N (Y_i - \hat{Y}_i)^2$  hata kareler toplamını ifade ederken, N, de gözlem sayısını göstermektedir. Determinasyon katsayısı ise bu eşitlikten yararlanılarak

$$R^2 = 1 - \frac{\sum_{i=1}^N (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^N (Y_i - \bar{Y})^2}$$

formülüyle ifade edilebilmektedir.

Nasibov, ND1 olarak tanımladığı birinci uzaklığını kullanarak bulanık en küçük kareler regresyon analizi için yeni bir determinasyon katsayısı olan WABL determinasyon katsayısını önermiştir. Bulanık sayıların WABL, ortalama genişlik değerleri ve ND1 (Nasibov'un birinci uzaklığı) kullanılarak bu yeni determinasyon katsayısı,

$$R^2 \approx 1 - \frac{\sum_{i=1}^N ([I(Y_i) - I(Y_i^*)]^2 + [D(Y_i) - D(Y_i^*)]^2)}{\sum_{i=1}^N ([I(Y_i) - \sum_{i=1}^N I(Y_i) / N]^2 + [D(Y_i) - \sum_{i=1}^N D(Y_i) / N]^2)}$$

şeklinde formülle gösterilmiştir.

## **5. BULANIK YAPAY SİNİR AĞLARI YAKLAŞIMININ BULANIK REGRESYON ANALİZİNDE KULLANIMI**

Yapay sinir ağları (Neural Network) non-parametrik regresyon analizinde çoğu araştırmacının başvurduğu bir yöntemdir. Kullanılan bağlantı ağırlıkları aşağıdaki gibi sınıflandırılabilir.

- 1) Bulanık olmayan ağırlıklar (nonfuzzy)
- 2) Simetrik bulanık ağırlıklar
- 3) Simetrik olmayan bulanık ağırlıklar

Bu yaklaşımların yanında nonparametrik regresyon analizi için yapay sinir ağlarının bulanık çıkarım sistemi ile güçlü bir kombinasyon kurması gerekmektedir.

Literatürde bulanık yapay sinir ağları konusunda araştırma bulunmaktadır. Cheng ve Lee, bir bulanık radyal temel fonksiyon ağı (FRBFN) adı ile bulanık regresyon analizi için bir metot önermişlerdir (Cheng and Lee, 1997).

Jang ve Sun, bulanık olmayan radyal temelli fonksiyon ağının fonksiyonel olarak bulanık çıkarım sistemine eşit olduğunu göstermişlerdir (Jang and Sun, 1995).

Ayrıca Cheng ve Lee, bulanık sinir ağı (FAN) adı altında yeni bir model önermişlerdir (Cheng and Lee, 1999).

Bu modelde (FAN) bulanık sinir ağı; iki kısımdan oluşmaktadır.



1) Öncül parametrelerin tanımlanması: Bu tanımlama geride üretilmiş algoritmalar kullanılarak uygulanmaktadır.

2) Sonsal parametrelerin tanımlanması: Bu tanımlama, olasılıksal bir lineer sistemi çözmek vasıtasıyla uygulanmaktadır.

Bulanık çıkarsama sistemi, bulanık küme teorisi, bulanık eğer-ise kuralları ve bulanık muhakemeye dayalı kullanışlı bir hesaplama yapısı oluşturmaktadır .Regresyon modelinin bilinmeyen parametrelerinin tahmininde kullanılabilen bulanık sinir ağları bulanık eğer-ise kuralları ve bulanık çıkarsama sistemine dayanmaktadır. Farklı dağılımlardan gelen bulanık girdilere bir regresyon doğrusu tahmin etmek sözkonusu olduğunda genelde Sugeno bulanık çıkarım sistemi kullanılabilir. Sugeno bulanık çıkarım sisteminin bulanık kuralı ;

$$R^l = \text{Eğer} ( X_1 = F_1^l \text{ ise ve } X_2 = F_2^l \text{ ise } \dots \text{ve } X_p = F_p^l \text{ ise})$$

$$Y = Y^l = c_0^l + c_1^l x_1 + \dots + c_p^l x_p \quad 5.1)$$

eşitliği ile ifade edilmektedir.

Burada  $F_i^l$  bulanık kümeyi,  $Y^l$ ,  $R^l$  kuralına göre sistem çıktısını ifade etmektedir ( $l=1, \dots, m$ ). Bu bulanık eğer-ise kuralları kümesinden sonuç çıkarmak için bir çıkarım mekanizmasına ihtiyaç duyulmaktadır ki bu da bulanık muhakeme mantığı olarak adlandırılmaktadır. Bu çıkarım mekanizmaları tüm kurallardan toplam bilgiye dayalı sonuç sağlar. Sugeno bulanık çıkarım sisteminde  $c_i^L$  bulanık sayı varsayılarak regresyon analizinde bulanık çıktı elde edilebilecektir.

Bulanık kurallara göre elde edilen modellerin ağırlıklandırılmış ortalaması, Sugeno bulanık çıkarım sisteminin çıktısıdır ve farklı sınıflardan gelen veriler için ortak regresyon modeli bu ağırlıklandırılmış ortalama ile ifade edilebilmektedir.

$$\hat{Y} = \frac{\sum_{l=1}^m W^l Y^l}{\sum_{l=1}^m W^l} \quad (5.2)$$

$W^l$  ağırlığı, önerilen  $Y = Y^l$  'nün gerçek değerini göstermektedir ve aşağıdaki gibi tanımlanmaktadır.

$$W^l = \prod_{i=1}^p \mu_{F_i}^l(x_i) \quad (5.3)$$

Örnek olarak beş tabakalı bir (FAN) bulanık sinir ağını vermek istediğimizde varsayalımki bulanık çıkarım sistemi aşağıdaki 4 kuralı içersin:

$R^1$  = eğer ( $X_1$  küçük ise ve  $X_2$  düşük ise) sonra

$$Y = Y^1 = c_0^1 + c_1^1 x_1 + c_2^1 x_2$$

$R^2$  = eğer ( $X_1$  küçük ise ve  $X_2$  yüksek ise) sonra

$$Y = Y^2 = c_0^2 + c_1^2 x_1 + c_2^2 x_2$$

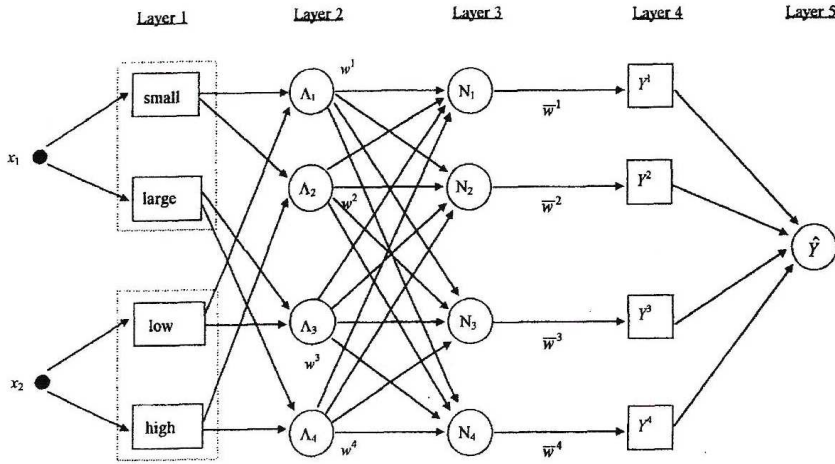
$R^3$  = eğer ( $X_1$  büyük ise ve  $X_2$  düşük ise) sonra

$$Y = Y^3 = c_0^3 + c_1^3 x_1 + c_2^3 x_2$$

$R^4 =$  eğer ( $X_1$  büyük ise ve  $X_2$  yüksek ise) sonra

$$Y = Y^4 = c_0^4 + c_1^4 x_1 + c_2^4 x_2$$

Bu çıkarım mekanizmasında iki boyutlu girdi mevcuttur.



Şekil 5.1: Beş tabakalı bir bulanık sinir ağı yapısı

Beş tabakadan oluşan sistemin her bir tabakasında yapılan işlemler aşağıda tanımlanmaktadır.

**Tabaka 1:** Sistem iki düzeyli iki değişkenden meydana geldiği için 4 bulanık kurala sahiptir. Bu kurallara ilişkin bulanık kümeler  $F_1$ ,  $F_2$ ,  $F_3$ ,  $F_4$  ile gösterildiğinde tabakadaki h.sinirin çıktısı:

$$F_{1,h} = \mu_{F_h}(x_1) \quad h = 1,2 \text{ için.} \quad (5.4)$$

$\mu_{F_h}$  ;  $F_h$  'ın üyelik fonksiyonudur.  $F_h$  için farklı üyelik fonksiyonları tanımlanabilmektedir. Burada verilerin parametre seti  $\{V_h, c_h\}$  olan normal dağılımdan geldiği düşünüldüğünde üyelik fonksiyonları aşağıdaki gibi tanımlanabilir.

$$\mu_{F_h}(x_1) = e \times p\left[-\left(\frac{x_1 - v_h}{\sigma_h}\right)^2\right], h=1, 2 \text{ için.}$$

$$\mu_{F_h}(x_2) = e \times p\left[-\left(\frac{x_2 - v_h}{\sigma_h}\right)^2\right], h=3, 4 \text{ için.} \quad (5.5)$$

Bu tabakadaki  $\{V_h, c_h\}$  parametre seti önsel parametreleri göstermektedir.

**Tabaka 2:** Bu tabakadaki her sinir sabitlenmiş sinir olmakla beraber  $\wedge_1$  sembolü ile gösterilmektedir. Birinci tabakadan gelen girdi sinyallerine sahiplerdir ve  $\wedge_1$  ; bu girdi sinyallerinin çarpımı şeklinde tanımlanmaktadır. Bu tabakaya ilişkin sinir fonksiyonları,

$$f_{2,1} = W^1 = \mu_{F_1}(x_1) \cdot \mu_{F_3}(x_2)$$

$$f_{2,2} = W^2 = \mu_{F_1}(x_1) \cdot \mu_{F_4}(x_2)$$

$$f_{2,3} = W^3 = \mu_{F_2}(x_1) \cdot \mu_{F_3}(x_2)$$

$$f_{2,4} = W^4 = \mu_{F_2}(x_1) \cdot \mu_{F_4}(x_2)$$

olarak ifade edilebilmektedir.

**Tabaka 3:** Bu tabakadaki sınırlar  $N_l$ ,  $l = 1,2,3,4$ . ile etiketlenmiştir ve 2. tabakada olduğu gibi sabit sınırlardır. Bu tabakanın çıktısı 2. tabakanın çıktısının bir normalizasyonudur.

$$f_{3,l} = \bar{W}^l = \frac{W^l}{\sum_{i=1}^m W^i} ; l = 1,2,3,4. \quad (5.6)$$

**Tabaka 4:** Bu tabakanın çıktı sinyalleri de bir fonksiyona bağlıdır ve  $Y^l$ , bulanık eğer-ise kurallarının sonuç kısmıdır ve

$$Y^l = c_0^l + c_1^l x_1 + c_2^l x_2$$

ile verilmektedir.  $c_i^l$ , bulanık sayılardır ve sonsal parametreleri” göstermektedirler.

$$f_{4,l} = \bar{W}^l Y^l \quad l = 1,2,3,4. \quad (5.7)$$

**Tabaka 5:** Bu tabakadaki tek sınır sabitlemiş sinirdir ve gelen sinyallerin tümünün toplamı olarak hesaplanmaktadır.

$$f_{5,1} = \hat{Y} = \sum_{l=1}^4 \bar{W}^l Y^l \quad (5.8)$$

Verilen girdi – çıktı veri çiftleri arasındaki sınır ağının eğitimi hata ölçütüne dayanmaktadır. Ağdan tahmin edilen tahmin ile hedeflenen çıktı arasındaki fark hata olarak tanımlandığında, ağ bu hata ölçütünü en küçük yapacak şekilde model oluşturabilmek üzere eğitilmektedir. Eğitim hata ölçütü, önceden belirlenen bir değerden küçük olduğunda sonlanmaktadır. Tahmin ile hedeflenen çıktı arasındaki farklılık her bir gözlem için

$$\varepsilon_k = Y_k \{-\} \hat{Y}_k$$

şeklinde hesaplanabilmektedir.

$Y_k$  ; k.hedef çıktı ve  $\hat{Y}_k$  ; k.girdi vektörü için ağ çıktısını göstermektedir.

Öncül ve sonsal parametrelerin belirlenmesi eğitimi için kullanılan farklı metotlar bulunmaktadır. Çünkü, sinir ağı ile parametre tahmini, hata ölçütünün en küçüklenmesi prensibine dayanmaktadır. Hatası en küçük olan tahmine ulaşmak önsel parametrelerin doğru belirlenmesine ve güncellenmesine ve ayrıca sonsal parametrelere bağlıdır.

$$\begin{aligned} \hat{Y} &= \bar{W}^l (c_0^l + c_1^l x_1 + \dots + c_p^l x_p) + \bar{W}^2 (c_0^2 + c_1^2 x_1 + \dots + c_p^2 x_p) + \\ &+ \dots + \bar{W}^m (c_0^m + c_1^m x_1 + \dots + c_p^m x_p) \\ &= c_0^l \bar{W}^l + c_1^l (\bar{W}^l x_1) + \dots + c_p^l (\bar{W}^l x_p) + \dots + c_p^m (\bar{W}^m x_p) \end{aligned} \quad (5.9)$$

Eğer  $\bar{W}^l$  bilirse eşitlik(5.9) ;

$$Y = A_0 + A_1 x_1 + A_2 x_2 + \dots + A_p x_p \quad (5.10)$$

eşitliği ile aynı olmaktadır.

Bu lineer eşitlik Tanaka'nın lineer programlama tekniği ile çözümlenmekte ve  $c_i^l; i = 0,1,\dots,p; l = 1,2,\dots,m$ . sonsal parametreleri aşağıda verilen bu modelle bulunmaktadır.

$$\text{Min} \sum_k^N \sum_{l=1}^m \sum_{i=0}^p \bar{W}^l b_i^l x_{ik}$$

$$\text{s.t. } b_i^l \geq 0, \quad i = 0, \dots, p, \quad l = 1, \dots, m;$$

$$\sum_{l=1}^m \sum_{i=0}^p \bar{W}^l a_i^l x_{ik} + (1-\alpha) \sum_{l=1}^m \sum_{i=0}^p \bar{W}^l b_i^l x_{ik} \geq y_k + (1-\alpha)e_k,$$

$$-\sum_{l=1}^m \sum_{i=0}^p \bar{W}^l a_i^l x_{ik} + (1-\alpha) \sum_{l=1}^m \sum_{i=0}^p \bar{W}^l b_i^l x_{ik} \geq -y_k + (1-\alpha)e_k,$$

$$k = 1, \dots, N. \quad (5.11)$$

Eşitlik(5.11)'in kısıtları,

$$[Y_k]_\alpha \subseteq [\hat{Y}_k]_\alpha \quad k = 1, \dots, N. \quad (5.12)$$

şartını sağlamak zorundadır. Buradaki  $\hat{Y}_k$ ;

k.gözlem  $Y_k$  'nın tahmini,  $[\cdot]_\alpha$  ise,  $\alpha$  seviye kümesini göstermektedir.

Öncül parametrelerin güncellenmesi is genellikle geri yayılım algoritmaları kullanılarak yapılmaktadır. Elde edilen parametre setinin süreç içerisinde güncellenmesi

$$\varepsilon_{5,1} = \frac{\partial \varepsilon_k^2}{\partial y_k}, \quad \varepsilon_{r,1} = \sum_{h=1}^{\mu_r+1} \varepsilon_{r+1,h} \frac{\partial f_{r+1,h}}{\partial f_{r,l}} \quad (5.13)$$

biçiminde tanımlanan geri yayılım hatalarına bağlıdır ve

$$\Delta\rho = -\eta \frac{\partial \varepsilon}{\partial \rho} \quad (5.14)$$

eşitliği ile yapılmaktadır.

Buradaki  $\rho$ , önsel parametre setini,  $\eta$  ise, 0 ve 1 arasında değer alabilen ve karar verici tarafından belirlenen öğrenme oranını ifade etmektedir.

Burada bahsedilen bütün bu eşitliklerin bir algoritma (FAN) ile gösterilmektedir.

Adım 1: İlk olarak  $\alpha$  değerini ayarla ve önsel parametre setine karar ver  $\{V_{i,j}; c_{i,j}\}$ .

Adım 2: Sonsal parametre seti  $\{c_i^l\}$  tanımla .

Adım 3: Hata ölçütü E'yi hesapla .

$$E = \frac{1}{N} \left( \sum_{k=1}^N (y_k - \hat{y}_k)^2 + (e_k - \hat{e}_k)^2 \right) \quad (5.15)$$

Eğer E, önceden belirlenmiş küçük bir hata değerinden küçükse ; dur, değilse, adım 3'e git.

Adım 3: Geri yayılım hatalarını hesapla, önsel parametre setini geri yayılım algoritmasını kullanarak güncelle.

Adım 4: Adım 1'e git.



Cheng ve Lee, bir örnek üzerinde FAN ile FRBFN algoritmalarını karşılaştırmışlardır.Sonuçta ise, FAN'ın fonksiyon tahminleme yeteneğinin FRBFN'den daha iyi olduğunu göstermişlerdir.Ayrıca FAN'ın, bir noktaya yakınsama hızının FRBFN'ye göre daha yüksek olduğunu göstermişlerdir (Cheng and Lee, 1999).

## 6. SWITCHING REGRESYON MODELİ

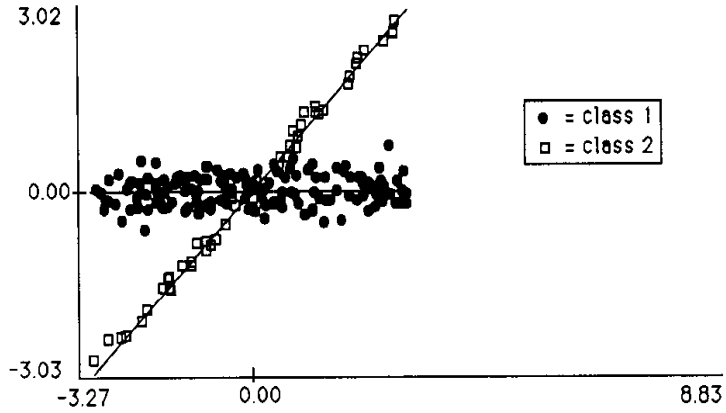
Klasik regresyon analizinde verilerin tek bir sınıftan geldiği düşünülür ve böylece bağımsız veriler  $x \in R^p$  ile bağımlı veri  $Y \in R^l$  arasında basit fonksiyonel bir ilişkinin olduğu varsayılmakta ve genel model,

$$Y = f(x) + \varepsilon \quad (6.1)$$

şeklinde ifade edilmektedir. Veri kümesi klasik regresyondakinin aksine, birbirinden farklı dağılımlara sahip iki veya daha fazla sınıftan alınan gözlemlerin bir araya getirilmesiyle meydana gelmiş olabilmektedir. Bu durumda,  $c$  sınıf sayısını göstermek üzere, her farklı sınıf bir  $f_i$  fonksiyonuyla ve rasgele hata  $e_i$  ile tanımlandığında switching regresyon modeli olarak tanımlanan model;

$$Y_i = f_i(x) + \varepsilon_i \quad 1 \leq i \leq c \quad (6.2)$$

şeklinde ifade edilmektedir. Şekil 6.1 'de görüldüğü gibi siyah ve beyazlarla gösterilen noktaları aynı regresyon doğrusuyla temsil etmemiz olanaksızdır. Bu yüzden siyah noktaların bir sınıftan, beyaz noktaların ise başka bir sınıftan geldiği düşünülerek iki farklı sınıf ile switching regresyon modeli kurularak bu noktalar temsil edilebilir.



Şekil 6.1: Switching Regresyon Modeli (c=2 için)

Gözlem kümesindeki verilerin birden fazla sınıftan oluşması ve verilerden bir kısmının sınıflara ait olma durumunun kesinlik taşımadığı yerlerde bulanıklık kavramı ile karşılaşılmaktadır.

Bu durumda eşitlik 6.2’de verilen modeldeki c farklı sınıf için üyelik fonksiyonları tanımlanarak bu fonksiyonlar, gözlemlerin tümünü yansıtması sağlanarak,

$$\hat{Y} = \frac{\sum_{i=1}^N w_i y_i}{\sum_{i=1}^c w_i} \quad (6.3)$$

modelinde  $w_i$  ağırlıklarının yerine koyulmaktadır.

Regresyon analizinde bir veri kümesi için gözlemlerin tek bir sınıftan geldiği düşünülür ve bağımlı ve bağımsız değişken arasındaki basit fonksiyonel ilişki eşitlik (6.1)’de verildiği gibidir. Veri kümesi birbirinden farklı dağılımlara sahip birden fazla sınıftan elde edilen gözlemlerin bir

araya getirilmesiyle elde edilmiş olabilir. Bu durumda  $c$  sınıf sayısını göstermek üzere eşitlik (6.2) yazılabilmektedir ve bu model switching regresyon modeli olarak tanımlanmaktadır (Wolkenhauer, ).

Wolkenhauer'a göre,  $m_j = (x_j, y_j)$  verisi için bir switching regresyon modeli aşağıdaki gibi yazılabilmektedir.

$$y_i = f_i(x; \theta_i) + \varepsilon_i \quad 1 \leq i \leq c \quad (6.4)$$

İstatistiksel bir çerçevede  $\theta^*$  nın optimal tahmini  $\varepsilon_i$  rasgele değişkenlerinin dağılımı hakkındaki varsayımlara bağlıdır. Genellikle  $\varepsilon_i$  'lerin, pdf  $p(\varepsilon, \sigma, c)$  ile normal dağılımdan geldiği varsayılmaktadır (0 ortalama ve  $\sigma$  bilinmeyen standart sapma ile).

$$p(\varepsilon, \sigma, c) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\varepsilon-\eta)^2}{2\sigma^2}} \quad (6.5)$$

Örneğin  $c$  sınıf sayısını göstermek üzere  $c=2$  ve  $l=2$  için,

$$y = f_1(x; \theta_1) + \varepsilon_1 \quad (6.6)$$

$$y = f_2(x; \theta_2) + \varepsilon_2 \quad (6.7)$$

Her bir  $m_j = (x_j, y_j)$  'nin  $i$  tane modelden  $\Pr(i)$  olasılığı ile meydana geldiği varsayılmaktadır.

$$\sum_{i=1}^c \Pr(i) = 1 \quad (6.8)$$

Verinin Log-likelihood fonksiyonu,

$$\mathcal{L}(\theta; M) = \ln \Pr(M | \theta)$$

$$= \sum_{j=1}^d \ln \sum_{i=1}^c \Pr(i) \cdot p(m_j | i, \theta_i)$$

$$= \sum_{j=1}^d \ln [\Pr(i) \cdot p(y_j - \theta_{11}x_j - \theta_{12} | 0, \sigma_1)$$

$$+ ((1 - \Pr(i)) \cdot p(y_j - \theta_{21}x_j - \theta_{22} | 0, \sigma_2))] \quad (6.9)$$

şeklinde yazılmaktadır. Bu  $M = \{m_j\}$  nin  $c$  alt modele bölünmesi ve regresyonla  $\theta_i$ 'nin bulunmasıyla çözümlenmektedir.

Hathaway ve Bezdek'e göre;

Bulanık switching regresyon modellerinde  $M$  bulanık olarak bölünmektedir.  $\theta_1, \theta_2, \dots, \theta_c$  bulanık  $c$ -ortalamalar algoritması (FCM)'yi tasarlamakla birlikte tahminlenmektedir. Her bir  $m_j$  bir bulanık etikete atanmaktadır ( $u_{ij} \in U$ ). Hata kriteri ise,

$$E_{ij}[\theta_i] = \|y_j - f_i(x_j; \theta_i)\|^2 \quad (6.10)$$

Amaç fonksiyonları:

$$E_w[U, \{\theta_i\}] = \sum_{i=1}^c \sum_{j=1}^d u_{ij}^w \cdot E_{ij}[\theta_i] \quad (6.11)$$

Eğer regresyon fonksiyonları  $f_i(x; \theta_i)$   $\theta_i$  parametrelerinde lineerse, parametreler ağırlıklandırılmış en küçük karelerin bir çözümü gibi elde edilebilmektedir:

$$\theta_i = [X_e^T W_i X_e]^{-1} X_e^T W_i Y \quad (6.12)$$

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_d^T \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ y_d \end{bmatrix}$$

$$W = \begin{bmatrix} u_{i1} 0 \dots \dots \dots 0 \\ 0 u_{i2} \dots \dots \dots 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \dots \dots \dots u_{id} \end{bmatrix}$$

ve  $X_e = [X, 1]$ .

Fuzzy c-regresyon algoritmasının adımları:

Hazırlık:

1)  $c$ 'yi  $2 \leq c < d$  arasında olmak şartı ile belirle.

2) termination toleransını seç  $\delta > 0$ .

3) w'yi  $1 \leq w < \infty$  arasında olmak üzere belirle.

4)  $U^{(0)} \in \mu_{fc}$  başlangıç değerini rasgele olarak seç.

L=1, 2, ....için tekrarla:

Adım 1: Aşağıdaki hata fonksiyonunu global olarak minimize eden  $\theta_i^{(l)}$  model parametrelerini hesapla.

$$E_w[U, \{\theta_i\}] = \sum_{i=1}^c \sum_{j=1}^d u_{ij}^w \cdot E_{ij}[\theta_i]$$

Adım 2: U'yu  $E_{ij} = E_{ij}[\theta_i^{(l-1)}]$  ile güncelle ve

$$u_{ij}^l = \frac{1}{\sum_{k=1}^c \left( \frac{E_{ij}}{E_{kj}} \right)^{\frac{2}{w-1}}}$$

eşitliğini sağla, eğer  $E_{ij} > 0$  ise  $1 \leq i \leq c$  için.

Ve aksi takdirde  $u_{ij} = 0$ , eğer  $E_{ij} > 0$  ve  $u_{ij} \in [0,1]$  ( $u_{1j} + \dots + u_{cj} = 1$ ) ile.

Algoritmayı  $\|U^{(l)} - U^{(l-1)}\| < \delta$  olana kadar çalıştır (Wolkenhauer).

## 7. UYGULAMA

Bu bölümde 5. bölümde verilen FAN (Fuzzy Adaptive Network) adı altındaki bulanık sinir ağı algoritmasının programı yazılmıştır. Program C++ Builder 6.0 programlama dili ile yazılmıştır ve program kodları Ek 1’de verilmiştir.

Cheng ve Lee’nın aşağıdaki örneği kullanılarak aynı sonuçlar elde edilmiş ve böylece programın çalıştığı gösterilmiştir. İstenen her türlü veriyi kullanarak program çalıştırılabilmektedir.

Deneme verilerini genelleştirmede aşağıdaki üç yapay fonksiyon kullanılmıştır.

$$f_1(x_1, x_2) = 24.23r^2(0.75 - r^2) + 5,$$

$$r^2 = (x_1/10 - 0.5)^2 + (x_2/10 - 0.5)^2. \quad (7.1)$$

$$f_2(x_1, x_2) = 1.3356(1.5(1 - \frac{x_1}{10}) + e^{\frac{x_1}{5}} \sin(3\pi(\frac{x_1}{10} - 0.6)^2) + e^{3(x_2/10 - 0.5)} \sin(4\pi(\frac{x_2}{10} - 0.9)^2)) + 4, \quad (7.2)$$

(Hwang, 1994).

$$f_3(x_1, x_2) = 10 \cdot \frac{\sin(2(x_1 - 5))}{2(x_1 - 5)} \cdot \frac{\sin(2(x_2 - 5))}{2(x_2 - 5)} + 5, \quad (7.3)$$



(Jang, 1993).

30 çift girdi-çıkıktı verileri her bir fonksiyondan oluşturulmuştur. Bu veriler örneklem 1, örneklem 2, örneklem 3, olarak f fonksiyonlarından elde edilmiştir. Her bir veri seti girdi  $(x_{k1}, x_{k2})$ ,  $k=1, \dots, 30$  olmak üzere uniform rasgele şekilde  $[0, 10]^2$  oluşturulmuştur.

Bulanık hedef çıktı  $y_k$  'nın merkezi:

$$y_k = f_j = (x_{k1}, x_{k2}), k=1, \dots, 30 ; j=1, 2, 3 \text{ için} \quad (7.4)$$

Buna karşılık hedefin yayılması:

$$e_k = \frac{1}{4} y_k, k=1, \dots, 30 \text{ için} \quad (7.5)$$

Bu verilere ve üç fonksiyona dayanarak bulanık çıkarım sistemi aşağıdaki gibi formüle edilmiştir.

$R^1 =$ Eğer  $(x_1 \text{ small1(küçük1)})$  ve  $x_2 \text{ small2(küçük2)}$ )ise sonra;

$$Y = Y^1 = c_0^1 + c_1^1 x_1 + c_2^1 x_2 \quad (7.6)$$

$R^2 =$ Eğer  $(x_1 \text{ small1(küçük1)})$ ve  $x_2 \text{ medium2(orta2)}$ )ise sonra;

$$Y = Y^2 = c_0^2 + c_1^2 x_1 + c_2^2 x_2 \quad (7.7)$$

$R^3 =$ Eğer  $(x_1 \text{ small1(küçük1)})$ ve  $x_2 \text{ large2(büyük2)}$ )ise sonra;

$$Y = Y^3 = c_0^3 + c_1^3 x_1 + c_2^3 x_2 \quad (7.8)$$

$R^4 =$  Eğer ( $x_1$  medium1(orta1))ve  $x_2$  small2(küçük2))ise sonra;

$$Y = Y^4 = c_0^4 + c_1^4 x_1 + c_2^4 x_2 \quad (7.9)$$

$R^5 =$  Eğer ( $x_1$  medium1(orta1))ve  $x_2$  medium2(orta2))ise sonra;

$$Y = Y^5 = c_0^5 + c_1^5 x_1 + c_2^5 x_2 \quad (7.10)$$

$R^6 =$  Eğer ( $x_1$  medium1(orta1))ve  $x_2$  large2(büyük2))ise sonra;

$$Y = Y^6 = c_0^6 + c_1^6 x_1 + c_2^6 x_2 \quad (7.11)$$

$R^7 =$  Eğer ( $x_1$  large1(büyük1))ve  $x_2$  small2(küçük2))ise sonra;

$$Y = Y^7 = c_0^7 + c_1^7 x_1 + c_2^7 x_2 \quad (7.12)$$

$R^8 =$  Eğer ( $x_1$  large1(büyük1))ve  $x_2$  medium2(orta2))ise sonra;

$$Y = Y^8 = c_0^8 + c_1^8 x_1 + c_2^8 x_2 \quad (7.13)$$

$R^9 =$  Eğer ( $x_1$  large1(büyük1))ve  $x_2$  large2(büyük2))ise sonra;

$$Y = Y^9 = c_0^9 + c_1^9 x_1 + c_2^9 x_2 \quad (7.14)$$

Her bir girdi değişkeni Çizelge7.1'deki gibi üç kısma bölünmüştür.

**Çizelge 7.1:** Önsel parametrelerin varsayılan ilk değerleri

|         | Örneklem 1 |     | Örneklem 2 |     | Örneklem 3 |     |
|---------|------------|-----|------------|-----|------------|-----|
|         | $v$        | $c$ | $v$        | $c$ | $v$        | $c$ |
| Small 1 | 1          | 5   | 2          | 5   | 2          | 5   |
| Medium1 | 5          | 5   | 5          | 5   | 5          | 5   |
| Large1  | 9          | 5   | 8          | 5   | 9          | 5   |
| Small 2 | 1          | 5   | 2          | 5   | 1          | 5   |
| Medium2 | 5          | 5   | 5          | 5   | 5          | 5   |
| Large2  | 6          | 5   | 8          | 5   | 9          | 5   |

Bu çıkarım sistemi  $x_1$  ve  $x_2$  olmak üzere 2 girdi düğümü olan ve 5 tabakadan oluşan bir (FAN) bulanık sinir ağı ile gösterilmiştir. Bu ağ yapısı Şekil 5.1’de verilmiştir. Tabaka 1’deki düğümler  $x_1$  ve  $x_2$  girdilerine karşılık gelen her üç düğümlerle 2 alt gruba bölünmüştür. Geriye kalan tüm tabakalar 9 düğüme sahiptir. Tabaka 1 ve 4’teki düğümlerin fonksiyonları sırasıyla önsel ve sonsal parametre setlerindeki parametreleri göstermektedir. Diğer tabakalardaki düğümler karma düğümlerdir. Sonuç olarak tabaka 5; tek bir çıktı ile tabaka 4’ün tüm çıktılarının toplamıdır.

Tabaka 1’de düğüm fonksiyonu  $v$  ve  $c$  parametreleri ile gaussian üyelik fonksiyonu ile tanımlanmıştır. Burada  $v$  merkez,  $c$  ise genişliği göstermektedir. Bu merkezler ve genişlikler için varsayılan ilk değerler Çizelge 7.1’de listelenmiştir.

Örneklem 1, 2 ve 3’ü kullanarak ağ çıktıları, hedef çıktılar ve tahminlenen çıktılar 3 farklı veri seti için Çizelge 7.2, Çizelge 7.3, Çizelge 7.4’de listelenmiştir.

Çizelge 7.2: Örneklem 1'in uygulama sonuçları

| Girdi değerleri |       | Hedef çıktıları |       | Ağ çıktıları |           |
|-----------------|-------|-----------------|-------|--------------|-----------|
| $x_1$           | $x_2$ | $y$             | $e$   | $\hat{y}$    | $\hat{e}$ |
| 6.628           | 6.609 | 11.680          | 3.628 | 11.795       | 3.756     |
| 9.297           | 7.181 | 16.962          | 4.840 | 16.962       | 4.840     |
| 3.178           | 7.145 | 7.734           | 2.468 | 7.997        | 2.760     |
| 3.553           | 4.261 | 7.481           | 2.510 | 7.500        | 2.531     |
| 1.194           | 6.210 | 3.926           | 1.236 | 4.361        | 1.720     |
| 3.922           | 8.984 | 9.707           | 3.140 | 9.707        | 3.140     |
| 2.639           | 5.727 | 7.278           | 1.848 | 6.984        | 2.307     |
| 9.736           | 8.843 | 17.832          | 4.482 | 17.680       | 4.651     |
| 3.711           | 3.831 | 6.990           | 1.990 | 7.256        | 2.547     |
| 2.830           | 3.631 | 5.800           | 2.237 | 5.800        | 2.237     |
| 0.891           | 7.150 | 3.714           | 1.408 | 3.885        | 1.598     |
| 6.349           | 2.660 | 10.115          | 2.877 | 9.815        | 3.250     |
| 7.188           | 1.798 | 11.475          | 3.353 | 11.475       | 3.353     |
| 6.349           | 2.394 | 9.576           | 3.227 | 9.576        | 3.227     |
| 1.300           | 4.852 | 4.117           | 1.725 | 4.182        | 1.797     |
| 0.516           | 5.976 | 3.249           | 1.074 | 2.890        | 1.472     |
| 7.579           | 3.559 | 12.373          | 3.232 | 12.707       | 3.603     |
| 0.725           | 2.129 | 4.132           | 1.459 | 4.132        | 1.459     |
| 1.071           | 7.482 | 4.463           | 1.700 | 4.463        | 1.700     |
| 8.931           | 7.540 | 15.267          | 4.221 | 15.369       | 4.530     |
| 7.106           | 2.084 | 11.245          | 2.984 | 11.108       | 3.368     |
| 5.248           | 6.667 | 10.117          | 3.214 | 9.788        | 3.579     |
| 2.967           | 8.828 | 6.784           | 2.273 | 7.186        | 2.720     |
| 3.609           | 8.893 | 8.979           | 2.867 | 8.871        | 3.024     |
| 3.919           | 8.990 | 9.386           | 2.552 | 9.668        | 3.137     |
| 5.696           | 9.632 | 12.078          | 3.095 | 12.013       | 3.167     |
| 3.161           | 7.127 | 8.085           | 2.618 | 7.968        | 2.748     |
| 5.802           | 0.217 | 8.913           | 2.819 | 8.913        | 2.819     |
| 4.486           | 9.433 | 9.621           | 3.067 | 9.710        | 3.166     |
| 0.807           | 9.837 | 4.860           | 1.576 | 4.860        | 1.576     |

Çizelge 7.3: Örneklem 2'nin uygulama sonuçları

| Girdi değerleri |       | Hedef çıktıları |       | Ağ çıktıları |           |
|-----------------|-------|-----------------|-------|--------------|-----------|
| $x_1$           | $x_2$ | $y$             | $e$   | $\hat{y}$    | $\hat{e}$ |
| 8.486           | 0.674 | 8.301           | 2.075 | 8.242        | 2.140     |
| 3.613           | 9.075 | 7.535           | 1.884 | 7.535        | 1.884     |
| 4.104           | 4.537 | 5.182           | 1.296 | 5.200        | 1.421     |
| 0.379           | 2.430 | 8.187           | 2.047 | 8.187        | 2.047     |
| 8.839           | 5.784 | 7.219           | 1.805 | 7.219        | 1.805     |
| 5.998           | 1.443 | 7.029           | 1.757 | 7.029        | 1.757     |
| 7.971           | 4.760 | 6.423           | 1.606 | 6.408        | 1.631     |
| 1.081           | 4.159 | 7.295           | 1.824 | 7.295        | 1.824     |
| 8.395           | 2.577 | 7.429           | 1.857 | 7.429        | 1.857     |
| 2.009           | 1.523 | 7.751           | 1.938 | 7.738        | 1.952     |
| 8.278           | 5.040 | 6.674           | 1.668 | 6.664        | 1.681     |
| 0.473           | 5.337 | 7.717           | 1.929 | 7.729        | 1.964     |
| 5.365           | 1.823 | 6.605           | 1.651 | 6.616        | 1.672     |
| 6.386           | 4.522 | 5.379           | 1.345 | 5.333        | 1.431     |
| 8.106           | 3.923 | 6.681           | 1.670 | 6.663        | 1.690     |
| 8.345           | 1.463 | 7.946           | 1.987 | 7.936        | 1.998     |
| 3.206           | 9.349 | 7.835           | 1.959 | 7.904        | 2.036     |
| 4.795           | 1.650 | 6.740           | 1.685 | 6.752        | 1.698     |
| 2.017           | 3.649 | 6.670           | 1.668 | 6.701        | 1.702     |
| 9.915           | 4.733 | 7.981           | 1.995 | 7.959        | 2.020     |
| 2.672           | 3.086 | 6.451           | 1.613 | 6.481        | 1.660     |
| 9.035           | 8.424 | 8.190           | 2.047 | 8.170        | 2.068     |
| 3.285           | 1.113 | 7.491           | 1.873 | 7.491        | 1.873     |
| 5.397           | 2.195 | 6.302           | 1.576 | 6.312        | 1.615     |
| 3.082           | 4.163 | 5.749           | 1.437 | 5.810        | 1.535     |
| 0.953           | 8.301 | 8.155           | 2.039 | 8.201        | 2.398     |
| 1.747           | 0.926 | 8.150           | 2.037 | 8.196        | 2.088     |
| 6.550           | 5.012 | 5.422           | 1.356 | 5.376        | 1.429     |
| 8.583           | 9.516 | 8.364           | 2.091 | 8.364        | 2.091     |
| 5.200           | 5.181 | 5.013           | 1.253 | 4.967        | 1.362     |

Çizelge 7.4: Örneklem 3'ün uygulama sonuçları

| Girdi değerleri |       | Hedef çıktıları |       | Ağ çıktıları |           |
|-----------------|-------|-----------------|-------|--------------|-----------|
| $x_1$           | $x_2$ | $y$             | $e$   | $\hat{y}$    | $\hat{e}$ |
| 6.859           | 9.688 | 4.992           | 1.248 | 5.030        | 1.290     |
| 7.215           | 0.617 | 4.849           | 1.212 | 5.126        | 1.520     |
| 3.199           | 2.898 | 5.256           | 1.314 | 4.776        | 1.847     |
| 0.260           | 9.151 | 4.994           | 1.248 | 4.994        | 1.248     |
| 5.528           | 7.114 | 3.275           | 0.819 | 3.428        | 1.605     |
| 3.539           | 2.766 | 4.837           | 1.209 | 4.572        | 1.814     |
| 9.476           | 8.443 | 5.042           | 1.260 | 5.042        | 1.260     |
| 5.968           | 9.446 | 5.276           | 1.319 | 5.276        | 1.319     |
| 7.562           | 2.678 | 5.384           | 1.346 | 5.353        | 1.546     |
| 5.103           | 3.324 | 4.379           | 1.095 | 3.830        | 1.754     |
| 2.802           | 6.101 | 4.208           | 1.052 | 5.042        | 1.978     |
| 2.831           | 6.492 | 4.887           | 1.222 | 5.094        | 1.933     |
| 8.287           | 6.081 | 5.167           | 1.292 | 5.203        | 1.332     |
| 5.479           | 2.999 | 3.382           | 0.845 | 4.140        | 1.687     |
| 0.423           | 0.885 | 5.033           | 1.258 | 5.033        | 1.258     |
| 6.134           | 2.824 | 4.273           | 1.068 | 4.679        | 1.616     |
| 8.976           | 1.165 | 5.160           | 1.290 | 4.743        | 1.753     |
| 2.316           | 7.121 | 5.310           | 1.328 | 5.851        | 1.838     |
| 0.080           | 2.127 | 5.036           | 1.259 | 5.223        | 1.467     |
| 2.937           | 1.300 | 4.755           | 1.189 | 5.074        | 1.543     |
| 5.408           | 1.343 | 6.047           | 1.512 | 6.047        | 1.512     |
| 6.512           | 6.041 | 5.163           | 1.291 | 4.930        | 1.550     |
| 3.504           | 5.534 | 5.409           | 1.352 | 4.839        | 1.986     |
| 9.319           | 1.516 | 5.075           | 1.269 | 5.531        | 1.776     |
| 6.879           | 2.906 | 5.318           | 1.330 | 5.104        | 1.568     |
| 6.793           | 0.142 | 5.035           | 1.259 | 4.858        | 1.456     |
| 8.325           | 2.432 | 4.904           | 1.226 | 5.218        | 1.575     |
| 0.539           | 8.211 | 5.012           | 1.253 | 5.221        | 1.486     |
| 1.544           | 6.900 | 4.863           | 1.216 | 4.301        | 1.841     |
| 9.298           | 2.566 | 4.826           | 1.207 | 4.437        | 1.640     |

Uygulamadan sonraki 3 veri seti için öncül parametre değerleri Çizelge 7.5'te gösterilmektedir. Yakınsama ölçümleri, hata ölçümleri ve hesaplama zamanı (CPU TIME) üç veri seti için de Çizelge 7.6'da listelenmektedir.

**Çizelge 7.5:** Denemeden sonraki öncül parametre değerleri

|         | Örneklem 1 |      | Örneklem 2 |      | Örneklem 3 |      |
|---------|------------|------|------------|------|------------|------|
|         | v          | c    | v          | c    | v          | c    |
| Small 1 | 1001       | 5001 | 1701       | 4735 | 2005       | 4968 |
| Medium1 | 4994       | 4998 | 5043       | 5110 | 4997       | 5036 |
| Large1  | 9002       | 4995 | 7772       | 5039 | 8963       | 5031 |
| Small 2 | 1005       | 5004 | 2277       | 4945 | 1073       | 5068 |
| Medium2 | 5003       | 4998 | 5849       | 4975 | 4923       | 5038 |
| Large2  | 6003       | 4998 | 7461       | 5491 | 9032       | 4995 |

**Çizelge 7.6:** FAN İçin Denemenin Hesaplama Sonuçları

|          | Örneklem 1 | Örneklem 2 | Örneklem 3 |
|----------|------------|------------|------------|
|          | 500        | 159        | 174        |
| E        | 0.0148     | 0.0992     | 0.3469     |
| APE      | 4 %        | 15 %       | 32 %       |
| CPU Time | 297        | 70         | 85         |

Buradaki E Eşitlik (5.15)'de verilmiş olan hata ölçütüdür. CPU Time, deneme verilerinin bilgisayar tarafından hesaplanma performansını göstermektedir ve dakika cinsinden ifade edilmektedir. APE ortalama hata yüzdesini göstermektedir ve aşağıdaki gibi tanımlanmaktadır.

$$APE = \frac{1}{N} \sum_{k=1}^N \frac{(y_k^u - \hat{y}_k^u + \hat{y}_k^L - y_k^L)}{y_k^u - y_k^L} \cdot 100\% \quad (7.15)$$

Burada  $y_k^u, y_k^L [y_k]_{\alpha}$  'nın sırasıyla üst ve alt limitlerini,  $\hat{y}_k^u, \hat{y}_k^L$  ise sırasıyla  $[\hat{y}_k]_{\alpha}$  'nın üst ve alt limitlerini göstermektedir.

$$y_k^u = y_k + (1 - \alpha)e_k, \quad (7.16)$$

$$y_k^L = y_k - (1 - \alpha)e_k, \quad (7.17)$$

$$\hat{y}_k^u = \hat{y}_k + (1 - \alpha)\hat{e}_k, \quad (7.18)$$

$$\hat{y}_k^L = \hat{y}_k - (1 - \alpha)\hat{e}_k. \quad (7.19)$$

Sinir ağını oluşturmak için Borland C++ builder tasarımı altında program tasarlanmıştır (Şekil 7.1).

The screenshot shows a window titled 'Form1' with a 'FAN' button. The window is divided into two sections, X1 and X2. X1 has input fields for SMALL (V=1.5), LARGE (V=5.0), and Sig (5.0). X2 has input fields for SMALL (V=1.0), LARGE (V=6.0), and Sig (5.0). Below these fields is a text area containing the following output:

```
( 0.000, 0.000) ( 0.000, 0.000) ( 0.000, 0.000)
( 0.000, 0.000) ( 4.758, 0.000) ( -2.227, 5.293)
( 0.000, 0.000) ( 0.000, 0.000) ( 0.000, 0.000)
( 0.000, 0.000) ( 0.000, 0.000) ( 0.000, 0.000)
 3.922  8.984  9.707  -0.738
x1=3.32199993138545
x2=8.98400020599365
y=9.70699977874756
e=9.14000010490417
yFAN=9.737694621086121
eFAN=2.0721340179443
mu[0][0]=0.7922
mu[0][1]=0.8531
mu[1][0]=0.4091
mu[1][1]=0.9544
w[0]=0.3541
w[1]=0.7561
w[2]=0.3490
w[3]=0.8142
w[0]=0.1445
w[1]=0.3370
w[2]=0.1556
w[3]=0.3629
Amac vektörü:
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
Kısıtlar:
 19.494  19.382  8.968  8.736  4.544  11.277  8.202  0.072  0.479  0.477  0.169  1.117  1.114
Kısıtlar matrisi:
 0.144  0.958  0.955  0.337  2.234  2.227  0.156  1.031  1.028  0.363  2.406  2.399  -0.144
```

Şekil 7.1: FAN Programının Başlangıç Formu.

Programımızda kullanılan önsel parametrelerin varsayılan ilk değerleri Çizelge 7.7' de verilmiştir. Ayrıca örneklemin girdi değerleri ve hedeflenen çıktı değerleri de Çizelge 7.8' de verilmiştir. Sistemimiz dört bulanık kurala sahiptir. Bu kurallar,



$R^1$  =Eğer ( $x_1$  small1(küçük1) ve  $x_2$  small2(küçük2))ise sonra;

$$Y = Y^1 = c_0^1 + c_1^1 x_1 + c_2^1 x_2$$

$R^2$  =Eğer( $x_1$  small1(küçük1)ve  $x_2$  large2(büyük2))ise sonra;

$$Y = Y^2 = c_0^2 + c_1^2 x_1 + c_2^2 x_2$$

$R^3$  =Eğer( $x_1$  large1(büyük1)ve  $x_2$  small2(küçük2))ise sonra;

$$Y = Y^3 = c_0^3 + c_1^3 x_1 + c_2^3 x_2$$

$R^4$  = Eğer( $x_1$  large1(büyük1) ve  $x_2$  large2(büyük2))ise sonra;

$$Y = Y^4 = c_0^4 + c_1^4 x_1 + c_2^4 x_2$$

**Çizelge 7.7:** önsel parametrelerin varsayılan ilk değerleri

|         | Örneklem |     |
|---------|----------|-----|
|         | $v$      | $c$ |
| Small 1 | 1.5      | 5   |
| Large1  | 5        | 6   |
| Small 2 | 1        | 5   |
| Large2  | 6        | 5   |

**Çizelge 7.8:** Örneklemin girdi ve hedef çıktı değerleri

| Girdi değerleri |       | Hedef çıktılar |       |
|-----------------|-------|----------------|-------|
| $x_1$           | $x_2$ | $y$            | $e$   |
| 6.628           | 6.609 | 11.680         | 3.628 |
| 9.297           | 7.181 | 16.962         | 4.840 |
| 3.178           | 7.145 | 7.734          | 2.468 |
| 3.553           | 4.261 | 7.481          | 2.510 |
| 1.194           | 6.210 | 3.926          | 1.236 |
| 3.922           | 8.984 | 9.707          | 3.140 |
| 2.639           | 5.727 | 7.278          | 1.848 |

Verilen girdi değerleri için 10 iterasyon yapılmış ve bulunan en küçük hata kareler toplamı Çizelge 7.9'da her bir iterasyon için gösterilmiştir.

**Çizelge 7.9:** İterasyon sonundaki en küçük hata kareler toplamı

| İterasyon    | E    |
|--------------|------|
| 1. iterasyon | 5.89 |
| 2. iterasyon | 2.14 |
| 3. iterasyon | 1.08 |
| 4. iterasyon | 1.31 |
| 5. iterasyon | 2.14 |
| 6. iterasyon | 1.67 |
| 7. iterasyon | 1.79 |
| 8. iterasyon | 3.93 |
| 9. iterasyon | 1.36 |
| 10.iterasyon | 2.73 |

## 8. SONUÇ VE TARTIŞMA

Bu çalışmada güncel teknikler olan Soft Computing tekniklerinin regresyon analizinde kullanılması üzerine araştırmalar yapılmıştır. Uygulama kısmı olarak Sinir ağları kullanılarak regresyon modelinde en küçük hata kareler toplamı gitgide küçülen bir ağ yapısı ayarlanmıştır ve bunun Borland C++ Builder 6.0 sisteminde bilgisayar yazılımı oluşturulmuştur. Programın kodları Ek 1' de verilmiştir.

Çalışmada yer alan iterasyon sonuçlarından da görüldüğü üzere en küçük hata kareler toplamı  $E'$  nin değeri her bir iterasyonda giderek küçülme eğilimi göstermektedir. Ancak az iterasyon yapıldığından dolayı sabitlenmeye ulaşılmamıştır. Daha çok iterasyonda sinir ağlarının yapısı gereği  $E'$  nin küçük değerlerinde sabitlenmeye ulaşılır.

## KAYNAKLAR DİZİNİ

- Baykal, N., Beyan, T.,** 2004, Bulanık Mantık İlke ve Temelleri, Bıçaklar Kitabevi Yayın, 413s.
- Bekçi, S.,** 2004, Klasik regresyon modelinin geçerli olmadığı durumlarda Bulanık regresyon yaklaşımı, Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, 160 s.
- Bolat, B., Erol, K.O., İmrak, E.,** 2004, Genetic Algorithms In Engineering Applications And The Function of Operators, Mühendislik Ve Fen Bilimleri Dergisi 4:264-271.
- Chang P.T., LEE, E.S.,** 1994, Fuzzy least absolute deviations regression and the conflict trends in Fuzzy parameters, Computers Mathematical Application, 28 (5) : 89-101.
- Chang., P.T.,** 1997, Fuzzy seasonality forecasting, Fuzzy Sets and Systems, 90 (1) : 1-10.
- Chang, Y.H.O.,** 2001, Hybrid Fuzzy least-squares regression analysis and its reliability measures, Fuzzy Sets and Systems, 119 (2) :247-257.
- Chang, Y.H.O., Ayyub, B.M.,** 2001, Fuzzy regression methods- a comparative assessment, Fuzzy Sets and Systems, 119 (2) : 187-203.
- Cheng, C.B., LEE, E.S.,** 1999, Applying Fuzzy adaptive Network to Fuzzy regression analysis, Computers and Mathematics with Applications 38 : 123-140.

**KAYNAKLAR DİZİNİ (Devamı)**

- Dalkılıç, T.E., Apaydın, A.,** Regresyon analizinde girdilerin bulanık olması durumunda uyarlamalı ağ yaklaşımı ile parametre tahmini, Ankara Üniversitesi fen fakültesi İstatistik Bölümü, [terbay@science.ankara.edu.tr](mailto:terbay@science.ankara.edu.tr), 6s.
- Dijkman J.G., Lange S.I.D.,** 1983, Fuzzy numbers, Journal of mathematical Analysis and Applications 92 : 301-341.
- Diomand, E.,** 1987, Fuzzy least squares fitting of several Fuzzy variables, Analysis of Information , CRC Press, Tokyo, 329-331.
- Diomand P., Kloeden P.,** 1990, Metric spaces of Fuzzy sets, Fuzzy Sets and Systems 35 : 241-249.
- D'urso, P.,** 2003, Linear regression analysis for Fuzzy/crisp input and Fuzzy/crisp output data, Computational Statistics & Data Analysis, 42 (1-2) : 47-72.
- Hathaway, R.J., Bezdek, J.C.,** 1993, Switching regression models and Fuzzy clustering, IEEE Transactions on Fuzzy systems Vol:1 no:3 195-204.
- Haykin, S.,** 1999, Neural Networks: A comprehensive Foundation, Perenctice Hall, New Jersey.
- Hwang, J.N., Lay, S.R., Maechler, M., Martin, R.d., Schimert,J.,** 1994, Regression modelling in back propagation and projection pursuit learning, IEEE Transactions on Neural Networks 5 (3), 342-353.

**KAYNAKLAR DİZİNİ (Devamı)**

**Ishibuchi, H., Kwon, K., Tanaka, H.,** 1995, A Learning algorithm of fuzzy neural networks with triangular fuzzy weights, Fuzzy Sets and Systems 71, 277-293.

**Jang, J.S. R., Sun, C.T., Mizutani, E.,** 1997, A Computational Approach to Learning and Machine Intelligence, Neuro-Fuzzy and Soft Computing , Prentice Hall.

**Kaastra, I., Boyd, M.,** 1996, “Designing a Neural Network for Forecasting Financial and Economic Time Series”, Neurocomputing, Cilt 10, 215-236.

**Karaköse, M., Akın,E.,** 2005, Akıllı Sistem Tasarımı İçin Yumuşak Hesaplama Çatısı, 11.Elektrik Elektronik Bilgisayar Mühendisliği Ulusal Kongresi, 4 s.

**Karay, F.O., De Silva, C.,** 2004, Soft Computing and Intelligent Systems Design: Theory, Tools and Applications, Pearson Addison wesley, England.

**Kartalopoulos, S.V.,**1996, Understanding Neural Network and Fuzzy Logic, IEEE Press, Newyork.

**Kim, B., Bishu R.R.,** 1998, Evaluation of Fuzzy linear regression models by comparing membership functions, Fuzzy Sets and Systems, 100 (1-3) : 343-352.

**KAYNAKLAR DİZİNİ (Devamı)**

**Ma, M., Kandel, A., Friedman, M.,** 2000, A new approach for defuzzification, Fuzzy Sets and Systems, 111 : 351-356.

**Ma, M., Kandel, A., Friedman, M.,** 2002, Correction to “A new approach for defuzzification, Fuzzy Sets and Systems 128 : 133-134.

**Mizumoto M., Tanaka K.,** 1979, Some properties of Fuzzy numbers, In Gupto, M.M, and R.K.

**Modarres, M.,** 2005, Fuzzy lineer regression models with least square errors, (163) 977 -989.

**Nasibov, E.,** 2000, Metody obrabotki nechetkoi informatsii v zadachakh prinyatiya reshenii, Baku, Elm.

**Nasibov, E.,** 2002, Certain integral characteristic of Fuzzy numbers and a visual interactive method for choosing the strategy of their calculation, Journal of Computer and Systems Sciences International, 41 No : 4 : 584-590.

**Nasibov, E.,** 2003, Aggregations of Fuzzy values in linear programming tasks, Automatic Control & Computer Sciences , 37 No : 2 : 1-11.

**KAYNAKLAR DİZİNİ (Devamı)**

**Nasibov, E., Şenol, Ş.,** 2005, Seviyelere bağlı ağırlıklı uzaklığa dayanan yeni bir bulanık regresyon modeli, 4. İstatistik Kongresi Bildiriler Kitabı 53-58.

**Nasrabadi, M.M., Nasrabadi E.,** 2003, Applied Mathematics and Computation, In Press.

**Olaf Wolkenhauer,** Fuzzy clustering and switching regression, UMIST ControlSystemsCentre,  
[www.csc.umist.ac.uk/people/wolkenhauer.htm](http://www.csc.umist.ac.uk/people/wolkenhauer.htm), 2-13.

**Özelkan, E.C., Duckstein L.,** 2000, Multi-objective Fuzzy regression : a general framework, Computers & Operations Research 27 : 635-652.

**Rechenberg, I.,** 1973, Evolutionstrategie (Evolution strategy), Fromman-Holzboog, Stuttgart.

**Şen, Z.,** 2001, Bulanık (Fuzzy) Mantık ve Modelleme İlkeleri, Bilge Kültür Sanat, 172 s.

**Tanaka H., Uejima S., and Asai K.,** 1982, Linear regression analysis with Fuzzy model, IEEE Transaction on system, Man and Cybernetics, 12 : 903-907.

**Tanaka H., Guo P.,** 1999, physica-Verlag Heidelberg, Newyork, 183 p.



**KAYNAKLAR DİZİNİ (Devamı)**

- Tektaş, M., Akbaş, A., Topuz, V.,** Yapay zeka tekniklerinin trafik kontrolünde kullanılması üzerine bir inceleme, Marmara Üniversitesi Teknik Bilimler MYO, [tektaş@marmara.edu.tr](mailto:tektaş@marmara.edu.tr) .
- Tuncer, S.,** 1999, “Değişken Hızlı Sürücü Sistemleri İçin Fuzzy Denetleyici Yeni Bir Algoritmanın Geliştirilmesi ve Uygulanması” , F. Ü. ,Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- Wang, H.F., Tsauro, R.C.,** 2000, Insight of Fuzzy regression model, Fuzzy Sets and Systems, 112 (3) : 355-369.
- Wang, H.f., Tsauro, R.C.,** 2000, Resolution of Fuzzy regression model, European journal of operational Research 126 (3) :637-650.
- Wu, B., Tseng, N.F.,** 2002, A new approach to Fuzzy regression models with application to business cycle analysis, Fuzzy Sets and Systems, 130 (1): 33-42.
- Yang, M.S., Liu, H.H.,** 2003, Fuzzy least squares algorithms for interactive Fuzzy linear regression models, Fuzzy Sets and Systems, 135 (2) : 305-316.
- Yen, J., Langari, R.,** 1998, Fuzzy Logic Intelligence Control and Information, Prentice Hall, s.93-102.
- Zadeh, L.A.,** 1965, Fuzzy sets, Information and Control, Volume 8 (1965) 338-353.

**KAYNAKLAR DİZİNİ (Devamı)**

**Zadeh, L.A.**, 1968, Fuzzy algorithms, Informath and Control, 12 : 94-102,  
1: 3-28.

**Zadeh, L.A.**, 1975, The concept of a linquistik variable and its application  
to approximate reasoning I, Information Sciences, 8:199-251.

**Zadeh, L.A.**, 1975, The concept of a linquistik variable and its application  
to approximate reasoning II, Information Sciences, 8:43-80.

**Zadeh, L.A.**, 1975, The concept of a linquistik variable and its application  
to approximate reasoning III, Information Sciences, 8:301-357.

**Zhang,G.,Patuwo, B.E., Hu, M.Y.**,1998, “Forecasting With Artificial  
Neural Networks:The State of The Art ”, Inter.Journal of  
Forecasting, Cilt 14, 35-62.

**EKLER**

Ek1 FAN Program Kodu

**EK 1 FAN program kodu**

```

#include <vcl.h>
#pragma hdrstop
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "UFAN.h"
#define N 10
#define m 4
#define p 2
#define NM 200
typedef float m50_50[NM][NM];
typedef float *m50;
int b[NM];
int bazaj;
void zameshenie(int M1,int N1,int ki,int kj,m50_50 a);
void optima(float aaa[m][p+1],float bbb[m][p+1],float v[2][2], float
sig[2][2],float w[m]);
void backpro(float x1,float x2,float E,float yk[m],float v[2][2],float
sig[2][2],float ww[m]);
m50 simplex(int M1,int N1,m50 f,m50 ogr,m50_50 aa,m50 x);
m50 simplex2(int M1,int N1,int baza,m50 f,m50 ogr,m50_50 aa,m50_50
a,m50 x);
float x[p+1][N]={{1,1,1,1,1,1,1,1,1,1},//1,1,1,1,1},//1,1,1,1,1},

```

```

{6.628,7.297,3.178,3.553,1.194,3.922,2.639,5.375,2.554,8.769},//1.432,6.2
31,9.123,3.333,2.871,3.590,1.037,7.943,5.955},

{6.609,7.181,7.145,4.261,6.210,8.984,5.727,5.453,2.467,3.271}};//4.109,4.
298,1.093,2.984,9.453,7.233,3.873,8.222,7.988}};

float
y[N]={11.680,16.962,7.734,7.481,3.926,9.707,7.278,12.324,9.764,4.260};//
8.098,6.573,2.321,7.987,4.345,7.546,9.212,4.988,2.320,6.434};

float
e[N]={3.628,4.840,2.468,2.510,1.236,3.140,1.848,1.194,3.922,2.639};//5.37
5,2.467,3.271,4.109,4.298,1.093,5.654,8.223,1.234,3.100};

AnsiString kural[m]={ "00","01","10","11"};

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}

//-----

void __fastcall TForm1::Button3Click(TObject *Sender)
{
int M1,N1;

```

```
m50 f,ogr,x;
```

```
m50_50 aa;
```

```
f=new float[NM]; ogr=new float[NM]; x=new float[2*NM];
```

```
M1=3;
```

```
N1=7;
```

```
f[0]=4; f[1]=1; f[2]=3; f[3]=5; f[4]=0; f[5]=0; f[6]=0;
```

```
ogr[0]=20; ogr[1]=12; ogr[2]=30;
```

```
aa[0][0]= 3; aa[0][1]=1; aa[0][2]=1; aa[0][3]=6; aa[0][4]=1; aa[0][5]=0;
```

```
aa[0][6]=0;
```

```
aa[1][0]= 1; aa[1][1]=1; aa[1][2]=1; aa[1][3]=1; aa[1][4]=0; aa[1][5]=1;
```

```
aa[1][6]=0;
```

```
aa[2][0]= 2; aa[2][1]=1; aa[2][2]=4; aa[2][3]=8; aa[2][4]=0; aa[2][5]=0;
```

```
aa[2][6]=1;
```

```
simplex(M1,N1,f,ogr,aa,x);
```

```
}
```

```
//-----
```

```
m50 simplex(int M1,int N1,m50 f,m50 ogr,m50_50 aa,m50 x)
```

```
{
```

```
//m50 x=new float[2*NM];
```

```
static m50_50 a;
```

```

//Dvuxetapnoye reshenie

//Form1->Memo1->Lines->Add(" ETAP I:");

simplex2(M1,N1,1,f,ogr,aa,a,x);

if(a[1][0]<0){
    //Form1->Memo1->Lines->Add("Hell yoxdur!");
    return NULL;
}
else {
l4:for(int i=0; i<M1; i++)
    if(b[i]>=N1){

        if(fabs(a[i+2][0])<0.001){
            for(int j=1; j<bazaj; j++)
                if(a[i+2][j]!=0){ zameshenie(M1,N1,i,j,a); goto l4;}
            for(int ii=i+2; ii<M1+2; ii++)
                for(int j=0; j<M1+N1+1; j++)a[ii][j]=a[ii+1][j];
            M1=M1-1;
        }
    }

    simplex2(M1,N1,0,f,ogr,aa,a,x);
}
return x;
}

```

```
//-----
```

```
void zameshenie(int M1,int N1,int ki,int kj,m50_50 a){
float c;
int i1,i2;
c=a[ki+2][kj+1];
for(i2=0; i2<bazaj; i2++)
  a[ki+2][i2]=a[ki+2][i2]/c;
for(i1=0; i1<M1+2; i1++){
  if(i1==ki+2)continue;
  c=a[i1][kj+1];
  for(i2=0; i2<bazaj; i2++)
    a[i1][i2]=a[i1][i2]-a[ki+2][i2]*c;
}
b[ki]=kj;
}
```

```
//-----
```

```
m50 simplex2(int M1,int N1,int baza,m50 f,m50 ogr,m50_50 aa,m50_50
a,m50 x)
{

bool opt=false, neogr=false;
float tt,c;
int ki,kj;
```



```

char s[NM+2][NM*10];
char s1[NM*10],s2[NM*10];

bazaj=N+1;
//formirovanie dvuxetapnoy simplex-tablisi a[M+2][M+N+1]
if(baza==1){
    bazaj=M1+N1+1;
    for(int i=0; i<M1; i++)b[i]=i+N1;
    a[0][0]=a[1][0]=0;
    for(int j=1; j<N1+1; j++)a[0][j]=-f[j-1];
    for(int j=N+1; j<M1+N1+1; j++)a[0][j]=0;
    for(int j=1; j<N1+1; j++)a[1][j]=0;
    for(int j=N+1; j<M1+N1+1; j++)a[1][j]=-1;
    for(int i=2; i<M1+2; i++)a[i][0]=ogr[i-2];
    for(int i=2; i<M1+2; i++)
        for(int j=1; j<N1+1; j++)a[i][j]=aa[i-2][j-1];
    for(int i=2; i<M1+2; i++)
        for(int j=N1+1; j<N1+M1+1; j++)a[i][j]==((i-2)==(j-N-1))?1:0;

    for(int i=2; i<M1+2; i++)
        for(int j=0; j<bazaj; j++)a[baza][j]=a[baza][j]-a[i][j];
}
///  

for(int i1=0; i1<M1+2; i1++) strcpy(s[i1],"");
strcpy(s2,"");

Form1->Memo1->Lines->Add("  A matrиси:");

```

```

for(int i1=0; i1<M1+2; i1++){
    for(int i2=0; i2<bazaj; i2++){
        sprintf(s1,"%8.3f ",a[i1][i2]);
        strcat(s[i1],s1);
    }
    Form1->Memo1->Lines->Add(s[i1]);
}
Form1->Memo1->Lines->Add("  Bazis deyiskenler:");
for(int i1=0; i1<M1; i1++){
    sprintf(s1,"%3d ",b[i1]);
    strcat(s2,s1);
}
Form1->Memo1->Lines->Add(s2);
Form1->Memo1->Lines->Add("=====");
/**/
while((!opt)&&(!neogr)){
    kj=-1; c=999999;
    for(int j=1; j<bazaj; j++)
        if((a[baza][j]<0)&&(a[baza][j]<c)){kj=j-1; c=a[baza][j];}
    if(kj===-1){opt=true; goto l1;}
    else{
        opt=false; tt=999999; ki=-1;
        for(int i=2; i<M1+2; i++)
            if((a[i][kj+1]>0)&&(a[i][0]/a[i][kj+1]<tt)){
                ki=i-2;tt=a[i][0]/a[i][kj+1];
            }
        if(ki>=0){

```

```

        zameshenie(M1,N1,ki,kj,a);
    goto l1;
}
else{neogr=true; goto l1;}
}

l1:
}

for(int i1=0; i1<N1+M1; i1++)x[i1]=0;
for(int i1=0; i1<M1; i1++)x[b[i1]]=a[i1+2][0];

return x;
}

//-----

void weights(float x1,float x2, float v[2][2], float sig[2][2], float w[m]){
float mu[2][2];

mu[0][0]=exp(-pow((x1-v[0][0])/sig[0][0],2));
mu[0][1]=exp(-pow((x1-v[0][1])/sig[0][1],2));
//mu[0][2]=exp(-pow((x2-v[0][2])/sig[0][2],2));
mu[1][0]=exp(-pow((x2-v[1][0])/sig[1][0],2));
mu[1][1]=exp(-pow((x2-v[1][1])/sig[1][1],2));

```

```

for(int i=0;i<2;i++)
  for(int j=0;j<2;j++){
    AnsiString s="mu["+IntToStr(i)+"]["+IntToStr(j)+"]=";
    Form1->Memo1->Lines->Add(s+FloatToStrF(mu[i][j],ffFixed,7,4));
  }
float sum=0;

```

```

for(int i=0;i<m;i++){
  int i0=kural[i].SubString(1,1).ToInt();
  AnsiString jj0;
  jj0=kural[i].SubString(2,1);
  int j0=jj0.ToInt();
  w[i]=mu[0][i0]*mu[1][j0];
  sum+=w[i];
  AnsiString s="w["+IntToStr(i)+"]=";
  Form1->Memo1->Lines->Add(s+FloatToStrF(w[i],ffFixed,7,4));
}

```

```

for(int i=0;i<m;i++){
  w[i]=w[i]/sum;
  AnsiString s="w["+IntToStr(i)+"]=";
  Form1->Memo1->Lines->Add(s+FloatToStrF(w[i],ffFixed,7,4));
}
}

```

```
//-----
```

```
void optima(float aaa[m][p+1],float bbb[m][p+1], float w[m]){
```

```

float f[m][p+1];
float g1a[N][m][p+1],g1b[N][m][p+1],g1c[N];
float g2a[N][m][p+1],g2b[N][m][p+1],g2c[N];

float alfa=0.5;

for(int l=0;l<m;l++){
  for(int i=0;i<p+1;i++){
    f[l][i]=0;
    for(int k=0;k<N;k++) f[l][i]+=w[l]*x[i][k];
  }
for(int k=0;k<N;k++){
  g1c[k]=y[k]+(1-alfa)*e[k];
  g2c[k]=-y[k]+(1-alfa)*e[k];
  for(int l=0;l<m;l++){
    for(int i=0;i<p+1;i++){
      g1a[k][l][i]=w[l]*x[i][k];
      g1b[k][l][i]=(1-alfa)*w[l]*x[i][k];
      g2a[k][l][i]=-w[l]*x[i][k];
      g2b[k][l][i]=(1-alfa)*w[l]*x[i][k];
    }
  }
}
m50_50 aa;
m50 ff,xx;
m50 ogr;
ff=new float[NM]; ogr=new float[NM],xx=new float[NM];
for(int k=0;k<2*N;k++)

```

```
for(int j=0;j<3*m*(p+1)+2*N;j++) aa[k][j]=0;
```

```
for(int i=0;i<3*m*(p+1)+2*N;i++)ff[i]=0;
```

```
for(int l=0;l<m;l++)
```

```
for(int i=0;i<p+1;i++)ff[2*m*(p+1)+l*(p+1)+i]=f[l][i];
```

```
for(int i=0;i<N;i++)ogr[i]=g1c[i];
```

```
for(int i=N;i<2*N;i++)ogr[i]=g2c[i];
```

```
for(int k=0;k<N;k++){
```

```
for(int l=0;l<m;l++)
```

```
for(int i=0;i<p+1;i++){
```

```
aa[k][l*(p+1)+i]=g1a[k][l][i];
```

```
aa[k][m*(p+1)+l*(p+1)+i]=-g1a[k][l][i];
```

```
aa[k][2*m*(p+1)+l*(p+1)+i]=g1b[k][l][i];
```

```
}
```

```
aa[k][3*m*(p+1)+k]=-1;
```

```
}
```

```
for(int k=N;k<2*N;k++){
```

```
for(int l=0;l<m;l++)
```

```
for(int i=0;i<p+1;i++){
```

```
aa[k][l*(p+1)+i]=g2a[k-N][l][i];
```

```
aa[k][m*(p+1)+l*(p+1)+i]=-g2a[k-N][l][i];
```

```
aa[k][2*m*(p+1)+l*(p+1)+i]=g2b[k-N][l][i];
```

```
}
```

```
aa[k][3*m*(p+1)+k]=-1;
```

```

}
char s[2000],s1[100];

int N1=3*(p+1)*m+2*N;
int M1=2*N;
simplex(M1,N1,ff,ogr,aa,xx);

for(int l=0;l<m;l++)
  for(int i=0;i<p+1;i++){
    aaa[l][i]=xx[l*(p+1)+i]-xx[m*(p+1)+l*(p+1)+i];
    bbb[l][i]=xx[2*m*(p+1)+l*(p+1)+i];
  }

Form1->Memo1->Lines->Add("---- c[i][l] katsayilari----");
for(int l=0;l<m;l++){
  strcpy(s,"");
  for(int i=0;i<p+1;i++){
    sprintf(s1,"%8.3f,%8.3f ",aaa[l][i],bbb[l][i]);
    strcat(s,s1);
  }
  Form1->Memo1->Lines->Add(s);
}

}

//-----

```

```

void __fastcall TForm1::Button1Click(TObject *Sender)
{

float w[N][m],ww[m];
float x1,x2;
float v[2][2],sig[2][2];
float aaa[m][p+1],bbb[m][p+1];
float yy[N],ee[N],yk[m];
v[0][0]=Edit1->Text.ToDouble();
v[0][1]=Edit3->Text.ToDouble();
//v[0][2]=Edit3->Text.ToDouble();
v[1][0]=Edit5->Text.ToDouble();
v[1][1]=Edit7->Text.ToDouble();
//v[1][2]=Edit7->Text.ToDouble();
sig[0][0]=Edit2->Text.ToDouble();
sig[0][1]=Edit4->Text.ToDouble();
//sig[0][2]=Edit4->Text.ToDouble();
sig[1][0]=Edit6->Text.ToDouble();
sig[1][1]=Edit8->Text.ToDouble();
//sig[1][2]=Edit8->Text.ToDouble();
int kk=0;
do{
Form1->Memo1->Lines->Add("Iterasyon "+IntToStr(++kk));
Form1->Memo1->Lines->Add("  x1    x2    y    yFAN");
for(int k=0;k<N;k++){
x1=x[1][k];          //Edit9->Text.ToDouble();
x2=x[2][k];          //Edit10->Text.ToDouble();

```



```

weights(x1,x2,v,sig,ww);
optima(aaa,bbb,ww);
yy[k]=0; ee[k]=0;
for(int l=0;l<m;l++){
    yk[l]=0;
    for(int i=0;i<p+1;i++){
        yk[l]+=aaa[l][i]*x[i][k];
        yy[k]+=ww[l]*aaa[l][i]*x[i][k];
        ee[k]+=ww[l]*bbb[l][i]*x[i][k];
    }
}
char s1[100];
sprintf(s1,"%8.3f %8.3f %8.3f %8.3f",x1,x2,y[k],yy[k]);
Form1->Memo1->Lines->Add(s1);
/**
Form1->Memo1->Lines->Add("x1="+FloatToStr(x1));
Form1->Memo1->Lines->Add("x2="+FloatToStr(x2));
Form1->Memo1->Lines->Add("y="+FloatToStr(y[k]));
Form1->Memo1->Lines->Add("e="+FloatToStr(e[k]));
Form1->Memo1->Lines->Add("yFAN="+FloatToStr(yy[k]));
Form1->Memo1->Lines->Add("eFAN="+FloatToStr(ee[k]));

/**/
    backpro(x1,x2,y[k]-yy[k],yk,v,sig,ww);
}
float E=0,eps=2;
for(int k=0;k<N;k++)E+=pow((y[k]-yy[k]),2)+pow((e[k]-ee[k]),2);

```

```

E=E/N;
Form1->Memo1->Lines->Add("====E="+FloatToStr(E));
Form1->Memo1->Lines->Add("");
// if(E<eps)break;
}while(kk<3);

}
//-----

void backpro(float x1,float x2,float E,float yk[m],float v[2][2],float
sig[2][2],float ww[m]){
float e[5][4],mu[2][2];
float dv0,dv1,dv2,dv3;
float dsig0,dsig1,dsig2,dsig3;
float eta=0.1;
mu[0][0]=exp(-pow((x1-v[0][0])/sig[0][0],2));
mu[0][1]=exp(-pow((x1-v[0][1])/sig[0][1],2));
// mu[0][2]=exp(-pow((x2-v[0][2])/sig[0][2],2));
mu[1][0]=exp(-pow((x2-v[1][0])/sig[1][0],2));
mu[1][1]=exp(-pow((x2-v[1][1])/sig[1][1],2));
// mu[1][2]=exp(-pow((x2-v[1][2])/sig[1][2],2));
e[4][1]=-2*E; // (y[k]-yy[k]);
for(int l=0;l<m;l++)e[3][l]=e[4][1];
for(int l=0;l<m;l++)e[2][l]=e[3][l]*yk[l];
float ss=0;
for(int l=0;l<m;l++)ss+=ww[l];

```

```

for(int l=0;l<m;l++){
    e[1][l]=e[2][l]/ss;
    for(int ll=0;ll<m;ll++)e[1][ll]-=e[2][ll]*ww[ll]/(ss*ss);
}

```

```

e[0][0]=e[1][0]*mu[1][0]+e[1][1]*mu[1][1];
e[0][1]=e[1][2]*mu[1][0]+e[1][3]*mu[1][1];
e[0][2]=e[1][0]*mu[0][0]+e[1][2]*mu[0][1];
e[0][3]=e[1][1]*mu[0][0]+e[1][3]*mu[0][1];

```

```

dv0=2*eta*e[0][0]*exp(-pow((x1-v[0][0])/sig[0][0],2))*(x1-
v[0][0])/pow(sig[0][0],2);

```

```

dv1=2*eta*e[0][1]*exp(-pow((x1-v[0][1])/sig[0][1],2))*(x1-
v[0][1])/pow(sig[0][1],2);

```

```

dv2=2*eta*e[0][2]*exp(-pow((x2-v[1][0])/sig[1][0],2))*(x2-
v[1][0])/pow(sig[1][0],2);

```

```

dv3=2*eta*e[0][3]*exp(-pow((x2-v[1][1])/sig[1][1],2))*(x2-
v[1][1])/pow(sig[1][1],2);

```

```

dsig0=2*eta*e[0][0]*exp(-pow((x1-v[0][0])/sig[0][0],2))*pow((x1-
v[0][0]),2)/pow(sig[0][0],3);

```

```

dsig1=2*eta*e[0][1]*exp(-pow((x1-v[0][1])/sig[0][1],2))*pow((x1-
v[0][1]),2)/pow(sig[0][1],3);

```

```

dsig2=2*eta*e[0][2]*exp(-pow((x2-v[1][0])/sig[1][0],2))*pow((x2-
v[1][0]),2)/pow(sig[1][0],3);

```

```

dsig3=2*eta*e[0][3]*exp(-pow((x2-v[1][1])/sig[1][1],2))*pow((x2-
v[1][1]),2)/pow(sig[1][1],3);

```

```
v[0][0]+=dv0; v[0][1]+=dv1; v[1][0]+=dv2; v[1][1]+=dv3;  
sig[0][0]+=dsig0; sig[0][1]+=dsig1; sig[1][0]+=dsig2; sig[1][1]+=dsig3;
```

```
}
```

```
//-----
```

## ÖZGEÇMİŞ

12/08/1980 tarihinde İzmir’de doğdum. İlk ve orta öğrenimimi Karşıyaka Cumhuriyet ilkokulu ve Şirinyer Otuz Ağustos ortaokulunda yaptım. Şirinyer lisesinin fen matematik bölümünden mezun olduktan sonra 1998 yılında Ege Üniversitesi İstatistik Bölümünü kazandım ve 2003 yılında mezun oldum. Mezun olduktan sonra İstatistik Bölümünde İstatistik Teorisi anabilim dalında yüksek lisansa başladım.