

**UZMAN SİSTEMLERDE
ÖRÜNTÜ TANIMA:**

***YAPAY SİNİR AĞLARI
GENETİK ALGORİTMALAR
BULANIK MANTIK
MAKİNE ÖĞRENMESİ***

Tamer Ölmez, Zümray Dokur

**UZMAN SİSTEMLERDE
ÖRÜNTÜ TANIMA:**

***YAPAY SİNİR AĞLARI
GENETİK ALGORİTMALAR
BULANIK MANTIK
MAKİNE ÖĞRENMESİ***

Prof. Dr. Tamer Ölmez, Prof.Dr. Zümray Dokur Ölmez

**İ.T.Ü. Elektrik-Elektronik Fakültesi,
Elektronik ve Haberleşme Mühendisliği Bölümü**

T.C.
İSTANBUL TEKNİK ÜNİVERSİTESİ
REKTÖRLÜĞÜ
Sayı :

1. Baskı 2009

“Her hakkı mahfuzdur. Kitabın sahibi İ.T.Ü. Rektörlüğü’dür. Bu kitabın tamamı veya bir kısmı İ.T.Ü. Rektörlüğü’nün yazılı izni alınmaksızın mekanik veya elektronik bir şekilde çoğaltılamaz veya kopyalanamaz.”

İÇİNDEKİLER

BİRİNCİ BÖLÜM	1
TEMEL KAVRAMLAR	1
1.1 KARAR VERME MEKANİZMASI	2
1.2 NORMALİZASYON İŞLEMİ	3
1.3 ÖZNİTELİK ÇIKARTMA İŞLEMİ	4
1.4 BOYUT İNDİRGEME VE VERİ ANALİZİ	5
1.5 FANTOM ÖZNİTELİK UZAYI	6
1.6 KARAR FONKSİYONLARI	8
1.7 SINIFLAYICI YAPILARI VE EĞİTİM	11
1.8 MAKİNE ÖĞRENMESİ	13
1.9 PERFORMANS KRİTERLERİ VE TASARIM	14
PROBLEMLER:	15
İKİNCİ BÖLÜM	17
DÖNÜŞÜM İŞLEMLERİ	17
2.1 İŞARETLER VE SİSTEMLER	18
2.1.1 İki Boyutlu İşaretler	18
2.1.2 Sistemlerde Lineer ve Öteleme İle Değişmezlik	20
2.1.3 İki Boyutlu Sistemler	22
2.1.4 Nicemleme ve Çözünürlük	24
2.2 ZAMAN SERİLERİNDE LİNEER ÖNGÖRÜ	24
2.2.1 AR (Autoregressive) Modeli	29
2.2.1.1 AR Parametrelerinin Kestirimi	30
2.2.2 Kayan Ortalama (MA) Modeli	32
2.2.2.1 İteratif Parametre Kestirimi	33
2.2.3 ARMA Modeli	34
2.2.3.1 Parametre Kestirimi – Direkt Yöntem	34
2.3 FOURİER DÖNÜŞÜMÜ	36
2.4 DALGACIK DÖNÜŞÜMÜ	39
2.4.1 Kısa-Zaman Fourier Dönüşümü	39
2.4.2 Sürekli Dalgacık Dönüşümü	43
2.4.3 Ayırık Dalgacık Dönüşümü	45
2.5 AYRIK KOSİNÜS DÖNÜŞÜMÜ	51
2.6 HADAMARD DÖNÜŞÜMÜ	55
2.7 GÖRÜNTÜ ANALİZ TEKNİKLERİ	57
2.7.1 Şekilsel Özelliklere Dayalı Analizler	58
2.7.1.1 Hough Dönüşümü	64
2.7.1.2 Zincir Kodlama	66

2.7.1.3 Moment Özellikleri	67
2.7.1.4 Şekilsel Özellikler	70
2.7.2 Dokusal Özelliklere Dayalı Analizler	72
2.7.2.1 Gri Seviye Dağılımı.....	72
2.7.2.2 Komşu Benek Genlik Dağılımı	74
2.7.2.3 Komşuluk Gri-Ton Fark Matrisi	74
2.7.2.4 Eş-Oluş Matrisi.....	77
PROBLEMLER:	82
ÜÇÜNCÜ BÖLÜM	85
BOYUT İNDİRGEME VE VERİ ANALİZİ	85
3.1 FANTOM ÖZNİTELİK UZAYI.....	85
3.2 ENTROPİ METODLARI.....	88
3.2.1 Entropinin Minimumlaştırılması	89
3.3 FISHER LİNEER AYRIMI	93
3.3.1 M Sınıflı Durum için Fisher Lineer Ayrımı	97
3.4 KARHUNEN-LOEVE GENİŞLETMESİ (KLG).....	98
3.4.1 Karhunen-Loeve Dönüşümü	99
3.5 TEKİL DEĞER AYRIŞTIRMASI (TDA).....	102
3.6 ÖZNİTELİKLERİN DOĞRUDAN SEÇİLEREK SIRALANMASI.....	106
3.6.1 Dinamik Programlama Metodları.....	106
PROBLEMLER:	113
DÖRDÜNCÜ BÖLÜM	115
SINIFLAYICILAR	115
4.1 KARAR FONKSİYONLARI.....	115
4.1.1 Lineer Karar Fonksiyonları	117
4.1.2 Genelleştirilmiş Karar Fonksiyonları	122
4.2 BAYES KARAR TEORİSİ.....	125
4.3 ÖBEKLEŞME KRİTERİ	135
4.3.1. Basit Bir Öbek Araştırma Algoritması	135
4.3.2 Maksimin-Mesafe Algoritması.....	137
4.3.3 k-Ortalamalar Algoritması.....	138
4.4 K En Yakın Komşu Sınıflayıcısı	141
PROBLEMLER:	143
BEŞİNCİ BÖLÜM	145
YAPAY SİNİR AĞLARI	145
5.1 YAPAY SİNİR AĞLARININ TANIMI	145
5.2 ÖZNİTELİK UZAYININ YAPAY SİNİR AĞLARI İLE	149
BÖLMELENMESİ.....	149
5.2.1 ‘Perceptron’ın Öznitelik Uzakını Bölmelemesi.....	149

5.3 YAPAY SİNİR AĞLARINDA ÖĞRENME	153
5.4 DENETİMLİ YAPAY SİNİR AĞLARI	153
5.4.1 Çok-Katmanlı Ağ	154
5.4.1.1 Çok-Katmanlı Ağın Yapısı	154
5.4.1.2 Çok-Katmanlı Ağın Öznitelik Uzayını Bölmelemesi	155
5.4.1.3 Çok-Katmanlı Ağın Eğitimi	159
5.4.2 Kısıtlı ‘Coulomb’ Enerji Ağı	162
5.4.2.1 RCE Ağının Yapısı	162
5.4.2.2 RCE Ağının Öznitelik Uzayını Bölmelemesi	163
5.4.2.3 RCE Ağının Eğitimi	164
5.4.3 Büyü ve Öğren Ağı	166
5.4.3.1 GAL Ağının Yapısı	166
5.4.3.2 GAL Ağının Öznitelik Uzayını Bölmelemesi	166
5.4.3.3 GAL Ağının Eğitimi	168
5.5 DENETİMSİZ YAPAY SİNİR AĞLARI	169
5.5.1 Kohonen Ağı	170
5.5.1.1 Kohonen Ağının Yapısı	170
5.5.1.2 Kohonen Ağının Öznitelik Uzayını Bölmelemesi	170
5.5.1.3 Kohonen Ağının Eğitimi	171
5.5.2 Artımsal Öz-Düzenlemeli Harita	173
5.5.2.1 ArÖz Ağının Öznitelik Uzayını Bölmelemesi	173
5.5.2.2 ArÖz Ağının Yapısı	173
5.5.2.3 ArÖz Ağının Eğitimi	174
5.6 EĞİTİMSİZ YAPAY SİNİR AĞLARI	176
5.6.1 Ayrık Hopfield Ağı	176
ALTINCI BÖLÜM	183
GENETİK ALGORİTMALAR	183
6.1 TEMEL GENETİK İŞLEMLER	183
6.2 UYUMLULUK FONKSİYONUNUN MAKSİMUM	186
DEĞERE YAKINSAMASI	186
6.3 BENZERLİK ŞEMASI	188
6.3.1 Şema Teoremi	189
6.3.2 Yapısal Blok Hipotezi	194
6.4 AMAÇ-UYUMLULUK FONKSİYONU DÖNÜŞÜMÜ	195
6.5 UYUMLULUK FONKSİYONUNUN ÖLÇEKLENMESİ	196
6.6 DİZİLERİN KODLANMASI	197
6.7 UYUMLULUK FONKSİYONU KISITLARI	199
6.8 YSA’LARIN GENETİK ALGORİTMALAR İLE EĞİTİMİ	206
6.8.1 Çok-katmanlı Ağın Genetik Algoritmalar ile Eğitilmesi	208
6.8.1.1 GetÇKA ile Öznitelik Uzayının Bölmelenmesi	208
6.8.1.2 GetÇKA’nın Yapısı	210

6.8.1.3 GetÇKA'nın Eğitimi	211
6.8.2 Yarışmacı Ağların Genetik Algoritmalar ile Eğitimi	213
6.8.2.1 GetYar Ağı ile Öznitelik Uzunluğunun Bölmelenmesi.....	215
6.8.2.2 GetYar Ağına Yapısı.....	215
6.8.2.3 GetYar Ağına Eğitimi	216
PROBLEMLER:	220
YEDİNCİ BÖLÜM	223
BULANIK MANTIK	223
7.1 BULANIK KÜME TEORİSİ	223
7.1.1 Üyelik Fonksiyonları	225
7.1.2 Bulanık Çıkarım	226
7.1.3 Bulanık Denetim.....	227
7.1.3.1. Bulanık Mantık Denetleyicinin Yapısı	228
7.2 BULANIK SINIFLAYICILARDA EĞİTİM	233
7.2.1 Bulanık k-Ortalamlar Algoritması	233
7.2.2 Bulanık Kohonen Öbekleşme Algoritması	235
7.3 BULANIK SINIFLAMA KARARININ ÜRETİLMESİ	237
7.3.1 Bulanık K-En Yakın Komşu Sınıflayıcısı	237
7.3.2 Bulanık En Yakın Prototip Sınıflayıcısı	239
PROBLEMLER:	241
SEKİZİNCİ BÖLÜM	243
MAKİNE ÖĞRENMESİ	243
8.1 KAVRAMLARI ÖĞRENME	244
8.1.1 Hipotezleri Genelden Özele Sıralama	245
8.1.2 Maksimal Özel Hipotezi Bulma ("Find-S").....	246
8.1.3 Aday Eleme Algoritması	247
8.2 DURUM UZAYINDA ARAŞTIRMA.....	253
8.2.1 Terminoloji ve Örnek Problem.....	253
8.2.2 Derinlik Öncelikli ("Depth-First") Araştırma	256
8.2.3 Genişlik Öncelikli ("Breadth-First") Araştırma	257
8.2.4 En İyiyi Öncelikle ("Best-First") Araştırma.....	258
8.3 KURALLAR KÜMESİNİ ÖĞRENME	258
8.3.1 Adım Adım Kapsama Algoritması	259
8.3.2 Bir_Kural_Öğren Algoritması	260
8.3.3 Birinci Mertebeden Kuralları Öğrenme.....	262
8.4 KARAR AĞACINI ÖĞRENME	266
8.4.1 Örnekler Üzerinde Entropi Ölçüsünün Kullanımı	268
8.4.2 Tümevarımsal Karar Ağacı (ID3) Algoritması	270
8.5 TAKVİYELİ ("REINFORCEMENT") ÖĞRENME	273
8.5.1 Tecrübenin Oluşturulması	273

8.5.2 Q Fonksiyonu	275
8.5.3 Q Öğrenmesi.....	277
8.6 KEŞİF YOLUYLA ÖĞRENME.....	278
8.6.1 Durum Vektörleri ile İlişkinin Keşfedilmesi	279
8.6.2 Graf Yoluyla İlişkinin Keşfedilmesi	281
8.7 BENZERLİKLERİ ÖĞRENME	284
8.7.1 Benzerlikleri Öğrenmede Temel Tanımlar	284
8.7.2 Benzerlikleri Öğrenme Algoritması	287
8.8 PROLOG PROGRAMLAMA	291
8.8.1 Prolog Programlarda girişi Verileri	291
8.8.2 Prolog Derleyicisinin Programları Çalıştırması	292
8.8.3 Prolog'ta Önemli Program Yazım Şekilleri	294
PROBLEMLER:	298
KAYNAKLAR.....	301
CEVAP ANAHTARI.....	305
İNDEKS.....	325

SİMGE LİSTESİ

$\bar{x} = (x_1, x_2, \dots, x_n)$	İndirgenmiş öznitelik vektörü
$\bar{f} = (f_1, f_2, \dots, x_n)$	Öznitelik vektörü
$\bar{x} = (x_1, x_2, \dots, x_n)$	Sınıflayıcıların ve yapay sinir ağlarının girişi
$u(t)$	Bir boyutta giriş işareti
$v(t), y(t)$	Bir boyutta çıkış işaretleri
$K(\cdot), K_i$	Karar yüzeyinin ifadesi
S_i, s_i	i-inci sınıfın etiketleri
D, d	Mesafe ifadeleri
$I(n_1, n_2), I(x, y)$	İki boyutlu görüntü dizileri
$FI(k_1, k_2)$	İki boyutlu görüntü dizisinin Fourier katsayıları
$h(n_1, n_2)$	İki boyutta süzgeç transfer fonksiyonu
$h(k)$	Tek boyutta süzgeç transfer fonksiyonu
$m_{p,q}$	İki boyutta moment katsayıları
$p(\bar{f} s_i)$	Koşullu olasılık yoğunluğu
$p(\bar{x})$	\bar{x} vektörünün olasılık yoğunluğu
C_f	Giriş vektörlerinin Kovaryans matrisi
C_x	Dönüşüm sonucu oluşan vektörlerin Kovaryans matrisi
λ_i	Özdeğerler
\bar{u}_i	λ_i özdeğerinin özvektörü
A_f	Giriş vektörlerine ait sınıflar-arası saçılım matrisi
w_{ij}	Sınıflayıcı veya yapay sinir ağlarında düğüm ağırlıkları
\bar{w}_i	Düğüm ağırlık vektörü
\S	Şema
μ_F	Bulanık kümede üyelik fonksiyonu
μ_{ij}	Üyelik derecesi
H	Hipotez
du	Sembolik mantıkta bir durum
L	Literal
$\bar{O} = (\bar{o}_1, \bar{o}_2, \dots, \bar{o}_n)$	Özellik vektörü

ŞEKİL VE TABLO LİSTESİ

- ŞEKİL 1.1 Karar verme mekanizması.
- ŞEKİL 1.2 İki-boyutlu örnek öznitelik uzayı ve sınıfların dağılımı.
- ŞEKİL 1.3 Karar verme mekanizmasının bileşenleri.
- ŞEKİL 1.4 İki-boyutlu fantom öznitelik uzayında sınıfların dağılımı.
- ŞEKİL 1.5 İki-boyutlu fantom uzaydan elde edilen alt-küme.
- ŞEKİL 1.6 Doğrularla sınıf sınır eğrisini oluşturma.
- ŞEKİL 1.7 Minimum mesafe ölçüsü ile sınıf sınırlarının oluşturulması.
- ŞEKİL 1.8 İki-boyutlu fantom uzayın eşitlik (1.6) kullanılarak sınıflanması.
- ŞEKİL 1.9 Parametrelerin eğitimle değişimi. Noktalı çizgiler, önceki iterasyonlarda bulunan hatalı sınır doğrularını göstermektedir.
- ŞEKİL 2.1 $I(m, n)$ dizisinin iki boyutta gösterilimi.
- ŞEKİL 2.2 $\delta(m, n)$ darbesi.
- ŞEKİL 2.3 $I(m, n)$ ve $h(m, n)$ dizilerinin konvolüsyonu [1].
- ŞEKİL 2.4 $I(m, n)$ dizisi üzerinde nicemlemenin etkisi :a) 6 ikili/benek, b) 4 ikili/benek, c) 2 ikili/benek ve d) 1 ikili/benek.
- ŞEKİL 2.5 Görüntü çözünürlüğünün etkisi: a) 256×256 benek/çerçeve büyüklüğü, b) 128×128 benek/çerçeve büyüklüğü, c) 64×64 benek/çerçeve büyüklüğü ve d) 32×32 benek/çerçeve büyüklüğü.
- ŞEKİL 2.6 Transfer fonksiyonu modeli.
- ŞEKİL 2.7 ARMA (autoregressive moving average) modeli.
- ŞEKİL 2.8 AR ve MA modelleri.
- ŞEKİL 2.9 Frekans spektrumu.
- ŞEKİL 2.10 Çeşitli görüntülere ait frekans spektrumları
- ŞEKİL 2.11 Orijinal $u(t)$ işareti.
- ŞEKİL 2.12 $u(t)$ 'nin frekans spektrumu.
- ŞEKİL 2.13 Durağan olmayan bir işaret.
- ŞEKİL 2.14 2.13'deki işaretin frekans spektrumu.
- ŞEKİL 2.15 2.13'deki durağan olmayan işaretin kısa-zaman Fourier dönüşümü [5].
- ŞEKİL 2.16 2.13'deki işaretin sürekli dalgacık dönüşümü [5].
- ŞEKİL 2.17 Dalgacık ağacı. $h(n)$, alçak geçiren; $g(n)$ ise yüksek geçiren filtreleri göstermektedir.
- ŞEKİL 2.18 (a) Daubechies-2 dalgacık fonksiyonu, (b) Ölçek fonksiyonu.
- ŞEKİL 2.19 Daubechies-2 dalgacıklarına ait filtrelerin frekans spektrumları.
- ŞEKİL 2.20 Normal EKG vurusuna ait dalgacık düzlemi.
- ŞEKİL 2.21 EKG İşaretine ait dalgacık katsayıları.
- ŞEKİL 2.22 Dalgacık dönüşümünün iki boyutlu görüntüler üzerinde kullanımı. $a^{i,yt}$, i. seviye yatay ayrıntı; $a^{i,di}$, i. seviye dikey ayrıntı; $a^{i,kö}$, i. seviye köşegen ayrıntı; y^i ise, i. seviye yaklaşıklık katsayı matrisini göstermektedir.
- ŞEKİL 2.23 (a) Orjinal görüntü, (b) Orjinal görüntüye ait dalgacık katsayıları.

ŞEKİL 2.24 Aynalama işlemi ($N=4$).

ŞEKİL 2.25 Görüntü için aynalama işlemi ($N_1=3$, $N_2=4$) [1].

ŞEKİL 2.26 a) Orjinal görüntü, b) Orjinal görüntüye ait ayırık kosünüs katsayıları.

ŞEKİL 2.27 2-boyutlu kenar belirleme sistemi [1]

ŞEKİL 2.28 (a) $\partial I(x,y)/\partial x$ 'nin $I(m,n)*h_x(m,n)$ ile (b) $\partial I(x,y)/\partial y$ 'nin $I(m,n)*h_y(m,n)$ ile yaklaşımı [1].

ŞEKİL 2.29 Yöne bağımlı kenar belirleyiciler ile elde edilen kenar haritaları

(a) Orijinal CT görüntüsü (b) Yatay kenar belirleyici sonucu (c) Dikey kenar belirleyici sonucu (d) Yönden bağımsız kenar belirleyici sonucu.

ŞEKİL 2.30 Hough dönüşümü ile oluşan parametre uzayı.

ŞEKİL 2.31 Cisme ait eğim ve kenar bilgileri [7].

ŞEKİL 2.32 Eğim aralıkları ve örnek bir zincir kodunun oluşturulması [7].

ŞEKİL 2.33 Temel eksen ile koordinat eksenleri arasındaki açı [7].

ŞEKİL 2.34 Dikdörtgensel alan bilgisinin çıkartılması [7].

ŞEKİL 2.35 Dokusal özelliğe sahip görüntü.

ŞEKİL 2.36 Bir ve iki komşuluklu alt pencereler.

ŞEKİL 2.37 Komşu beneklerin gri-tonları ile vektör elemanlarının oluşturulması.

ŞEKİL 2.38 Seçilen benek etrafında çeşitli açılar için yönlerin gösterilimi.

ŞEKİL 2.39 a) N tipi, b) L tipi, c) R tipi ve d) P tipi vurulara ait EKG kayıtları.

ŞEKİL 2.40 X-ışınlı CT kafa kesit görüntüsü.

ŞEKİL 3.1 Karar verme mekanizması.

ŞEKİL 3.2 İki-boyutlu bir fantom öznitelik uzayında örnek sınıf dağılımı.

ŞEKİL 3.3 Karmaşık bir sınıf dağılımına sahip iki-boyutlu fantom öznitelik uzayı.

ŞEKİL 3.4 Entropi kriteriyle sıkıştırma ve sınıflama.

ŞEKİL 3.5 Örnek problem için sınıf bölgelerinin gösterilimi.

ŞEKİL 3.6 Örnek dağılımda sınıflara ait vektörlerin gösterilimi.

ŞEKİL 3.7 Örnek dağılımda bir boyutun genişletilmesi.

ŞEKİL 4.1 İki sınıflı dağılım için basit bir karar fonksiyonu.

ŞEKİL 4.2 Durum 1 için karar bölgeleri [13].

ŞEKİL 4.3 Durum 2 için karar bölgeleri [13].

ŞEKİL 4.4 Durum 3 için karar bölgeleri [13].

ŞEKİL 4.5 Ayırım fonksiyonları cinsinden sınıflama.

ŞEKİL 4.6 Örnek veri kümesi.

ŞEKİL 4.7 Öbek araştırmasında eşik değerinin seçiminin etkileri.

ŞEKİL 4.8 (a) Maksimin-mesafe algoritmasını açıklamak için kullanılan örnek görüntüler, (b) Örnek ve öbek merkezi tabloları

ŞEKİL 4.9 Örnek öznitelik vektörleri.

ŞEKİL 4.10 k-NN sınıflayıcıda komşuluğun etkisi.

ŞEKİL 5.1 İşlem elemanının yapısı [21].

ŞEKİL 5.2 Yapay sinir ağı [21].

ŞEKİL 5.3 Altı katmanlı bir yapay sinir ağı [21].

ŞEKİL 5.4 (a) 'Perceptron'un yapısı, (b) 'Perceptron'un eğitimi.

ŞEKİL 5.5 Giriş vektörünün konumu ve eşitlik (5.2)'de tanımlanan düzlem.

ŞEKİL 5.6 Kırpıcı fonksiyonunun iki-boyutlu öznitelik uzayında karara etkisi.

ŞEKİL 5.7 'Sigmoid' fonksiyonunun karara etkisi.

ŞEKİL 5.8 'Kernel' fonksiyonunun karara etkisi.

- ŞEKİL 5.9 Eğitim yöntemlerine göre gruplandırılan bazı yapay sinir ağları.
- ŞEKİL 5.10 Çok-katmanlı ağın yapısı.
- ŞEKİL 5.11 İki girişli ($n_0=2$), üç çıkışlı ($n_c=3$), üç katmandan oluşan ÇKA.
- ŞEKİL 5.12 Üç adet doğru ile bölmelenmiş iki-boyutlu örnek öznitelik uzayı.
- ŞEKİL 5.13 Fantom uzayın ÇKA ile sınıflanma sonucu.
- ŞEKİL 5.14 RCE ağının yapısı.
- ŞEKİL 5.15 Hiper küreler ile sınıfların temsil edilmesi.
- ŞEKİL 5.16 Fantom uzayın RCE ile sınıflanma sonucu.
- ŞEKİL 5.17 GAL ağının yapısı [23].
- ŞEKİL 5.18 Düğümle yardımcıyla sınıf sınırlarının oluşturulması.
- ŞEKİL 5.19 Fantom uzayın GAL ile sınıflanma sonucu.
- ŞEKİL 5.20 Kohonen ağının yapısı.
- ŞEKİL 5.21 Fantom uzayın Kohonen ağı ile sınıflanma sonucu.
- ŞEKİL 5.22 ArÖz ağının yapısı.
- ŞEKİL 5.23 Hopfield ağı.
- ŞEKİL 5.24 (a) Sekiz adet örnek örüntü ($M=8$), (b)gürültülü '3' girişi için çıkış örüntüleri.
- ŞEKİL 5.25 Dört bit'lik A/D çevirici iç yapısı [27].
- ŞEKİL 6.1 Uyumluluk fonksiyonunu gösteren rulet tekerleği [28].
- ŞEKİL 6.2 Dizilerin çaprazlanması.
- ŞEKİL 6.3 Uyumluluk fonksiyonunun ölçeklenmesi.
- ŞEKİL 6.4 Uyumluluk fonksiyonunun negatif değerler üretmesi.
- ŞEKİL 6.5 (a) El radyografi görüntüsü, (b) Türdeş olmayan ışık dağılımının üç boyutlu görünümü [31].
- ŞEKİL 6.6 Noktasal bir kaynaktan çıkan x-ışınının yakın mesafede bir röntgen filmine gelişinin üç boyutlu gösterimi [31].
- ŞEKİL 6.7 (a) Varyans görüntüsü, (b) bölge büyütme sonucunda elde edilen arka plan, (c) arka plandan örneklenmiş noktalar.
- ŞEKİL 6.8 Genetik havuz.
- ŞEKİL 6.9 (a) Orijinal X-ışını görüntüsünün arka planı maskelenmiş üç boyutlu görünümü, (b) ışık dağılımı düzeltilmiş görüntünün arka planı maskelenmiş üç boyutlu görünümü (c) orijinal görüntünün histogramı, (d) ışık dağılımı düzeltilmiş görüntünün histogramı [31].
- ŞEKİL 6.10 GetÇKA'nın yapısı.
- ŞEKİL 6.11 GetÇKA'nın eğitiminde kullanılan genetik havuz.
- ŞEKİL 6.12 (a) ÇKA'nın sınıflama sonucu, (b) GetÇKA'nın sınıflama sonucu.
- ŞEKİL 6.13 Düğümle yardımcıyla sınıf sınırlarının oluşturulması.
- ŞEKİL 6.14 GetYar ağının yapısı.
- ŞEKİL 6.15 GetYar ağının eğitiminde kullanılan genetik havuz.
- ŞEKİL 6.16 (a) Kohonen ağının sınıflama sonucu, (b) GAL ağının sınıflama sonucu, (c) GetYar ağının sınıflama sonucu.
- ŞEKİL 7.1 Duru H kümesine ait üyelik fonksiyonu.
- ŞEKİL 7.2 Çeşitli formlarda üyelik fonksiyonları (işlevleri).
- ŞEKİL 7.3 Yedi ayrı etiketli üyelik fonksiyonları.
- ŞEKİL 7.4 İki araba arasındaki uzaklık, arkadaki arabanın hızı ve frenleme şiddeti için üyelik fonksiyonları [33, 34].

- ŞEKİL 7.5 Araba fren şiddetinin denetimi örneğinde fren şiddetinin bulunması [33, 34].
- ŞEKİL 7.6 Bir bulanık denetleyicinin temel yapısı.
- ŞEKİL 7.7 Hata (e) ve hata değişiminin (ce) dilsel değerlerinin üyelik fonksiyonları [33, 34].
- ŞEKİL 7.8 Denetim çıktısının dilsel değerlerinin üyelik fonksiyonları [33, 34].
- ŞEKİL 7.9 Bulanık denetleyici kuralları [33, 34].
- ŞEKİL 7.10 Kohonen öbekleşme ağı.
- ŞEKİL 8.1 Kutu probleminin du_0 başlangıç durumuna göre çözümü [40].
- ŞEKİL 8.2 Su oyununun eğlenceli olduğu günleri temsil eden karar ağacı [41].
- ŞEKİL 8.3 (a)Durum uzayında oluşan hareket için anlık ödül ($o(du_t, m_t)$) değerleri, (b) orantılı toplamsal ödül (O^*) değerleri, (c) $Q(du_t, m_t)$ değerleri [41].
- ŞEKİL 8.4 Basınç, hacim ve sıcaklık verileri üzerinde “genişlik öncelikli araştırma” algoritması ile elde edilen araştırma sonuçları [40].
- ŞEKİL 8.5 Öznitelikler arasındaki ilişkiyi temsil eden graflar. Kesik çizgiler maksimum klikleri göstermektedir [40].
- ŞEKİL 8.6 Benzerlikler yoluyla öğrenmede kullanılan bilgi domenleri [40].
- ŞEKİL 8.7 Temel ve Hedef bilgi domenlerindeki durumlar ve bu durumların temsili [40].
- ŞEKİL 8.8 Aile prolog programı [42].
- ŞEKİL 8.9 Prolog derleyicisinin araştırması [42].
- ŞEKİL 8.10 Listelerin ağaç gösterilimi.
- ŞEKİL 8.11 (A) kuralının derleyici tarafından sorgulanması [42].
- ŞEKİL p.1 “fu_mous.txt” dosyasına ait ilk satırlar.
- ŞEKİL p.2 Eğitim kümesi içinden rastgele seçilen vektörler.
- ŞEKİL p.3 Normalize edilmiş (a) N tipi, (b) L tipi, (c) R tipi ve (d) P tipi vurulara ait kümeler.
- ŞEKİL p.4 EKG vurularına ait öznitelik vektörleri.
- ŞEKİL p.5 Diverjans değerinin boyut sayısına göre değişimi.
- ŞEKİL p.6 Diverjans değerinin boyut sayısına göre değişimi.
- ŞEKİL p.7 Fantom uzayın k-NN sınıflayıcı ile sınıflandırılması.
- ŞEKİL p.8 CT görüntüsünün k-NN sınıflayıcı ile sınıflandırılması.
- ŞEKİL p.9 ÇKA’nın eğitiminde iterasyona bağlı hata grafiği.
- ŞEKİL p.10 Fantom uzayın ÇKA ile sınıflandırılması.
- ŞEKİL p.11 ÇKA’nın eğitiminde iterasyona bağlı hata grafiği.
- ŞEKİL p.12 ÇKA’nın eğitiminde iterasyona bağlı hata grafiği.
- ŞEKİL p.13 CT görüntüsünün ÇKA ile sınıflandırılması.
- ŞEKİL p.14 Ağın parametrelerinin genetik havuzda konumlanması.
- ŞEKİL p.15 Fantom uzayın ağ ile sınıflandırılması.
- ŞEKİL p.16 Ağın parametrelerinin genetik havuzda konumlanması.
- ŞEKİL p.17 Ağın parametrelerinin genetik havuzda konumlanması.
- ŞEKİL p.18 CT görüntüsünün ağ ile sınıflandırılması.
- ŞEKİL p.19 Fantom uzayın K-en yakın komşu sınıflayıcı ile sınıflandırılması.
- ŞEKİL p.20 Fantom uzayın K-en yakın komşu sınıflayıcı ile 9 sınıfa bölütlenmesi.
- ŞEKİL p.21 CT görüntüsünün bulanık K-en yakın sınıflayıcı ile sınıflandırılması.
- ŞEKİL p.22 En kısa mesafeden hedefe at’ın getirilmesi.

TABLO 2.1 Cisim tanım tablosu.

TABLO 4.1 n ve r değişkenlerine göre KA_w 'nin değişimi [13].

TABLO 5.1 ÇKA ile başarımlar sonuçları.

TABLO 5.2 RCE'nin başarımlar sonuçları.

TABLO 5.3 GAL ile başarımlar sonuçları.

TABLO 5.4 Başarımlar sonuçları.

TABLO 6.1 Örnek uyumluluk fonksiyonu [28].

TABLO 6.2 Genetik işlem sonuçları [28].

TABLO 6.3 Havuzda şema sayısı [28] (a) Kopyalamadan önce, (b) kopyalamadan sonra.

TABLO 6.4 İkili olan ve olmayan alfabede kodlamalar.

TABLO 6.5 İkili olan ve olmayan dizi topluluklarının karşılaştırılması [28].

TABLO 6.6 Örnek öznelik uzayında ÇKA ve GetÇKA'nın başarımlar sonuçları.

TABLO 6.7 Örnek öznelik uzayında GAL, Kohonen ve GetYar'ın başarımlar sonuçları.

TABLO 8.1 Pozitif ve negatif örnekler kümesi [41].

TABLO 8.2. Su oyunlarının eğlenceli olduğu günler [41].

ÖNSÖZ

Yapay zeka destekli örüntü (“pattern”) tanıma konusu, teknolojik gelişmelerle paralel geniş bir uygulama alanı bulmaktadır. Bu konudaki çalışmalar; tıpta vücut üzerinden alınan biyolojik işaretlerin analizinde, cep telefonlarında veya bilgisayarlarda ses tanıma düzeneklerinde, yüz veya parmak izi tanıyan güvenlik sistemlerinde, görerek objeleri tanıyan ve işlem yapan robot veya otomatlarda, uzaktan algılanan görüntülerin analizinde, savunma amaçlı oluşturulan düzeneklerde sıkça gözlenmektedir.

Bu kitapta yapay sinir ağları, genetik algoritmalar, bulanık mantık ve makine öğrenmesi gibi yapay zeka yaklaşımlarının örüntü tanıma konusunda kullanımları ele alınmaktadır. Kitap içindeki her bölümde örüntü tanıma işleminde oluşturulacak karar mekanizmasına ait gerekli adımlar ayrıntılı olarak incelenmektedir.

Kitabın birinci bölümden sekizinci bölüme kadar bölüm sonlarında üç farklı örnek üzerinde MATLAB ortamında çözülmesi istenen problemler bulunmaktadır : (i) İki-boyutlu fantom uzay, (ii) dört farklı EKG vurusu, (iii) bilgisayarlı tomografi görüntüsünde dört farklı doku. Sekizinci bölümün sonunda ise PROLOG programlama ile gerçekleştirilmesi istenen problemler bulunmaktadır. “Cevap Anahtarı” bölümünde problemlerin çözümleri ve oluşturulan çıktıların isimleri verilmektedir. Tüm programlar ve üretilen sonuçlar, kitapla birlikte verilen CD içinde bulunmaktadır.

Birinci bölümde örüntü uzayı, bu uzaydaki örüntüler, örüntülerin vektörel temsili (özniteliklerin belirlenmesi), vektörlerin normalize edilmesi, vektörlerin öznitelik uzayında dağılımı (sınıf tanımı), karar fonksiyonları (sınıflayıcı yapısı) ve makine eğitimi gibi temel kavramlar hakkında bilgiler verilecektir. Öznitelik çıkartma işlemi için çok sayıda yöntem bulunmaktadır ve bu alana sürekli yeni geliştirilen yöntemler katılmaktadır. İkinci bölümde genel karakteristiğe sahip bazı öznitelik çıkartma yöntemleri anlatılacaktır. Üçüncü bölümde, kategorileri tanımlama ve kategoriler-arası ayırcılığı vurgulama yeteneklerine göre özniteliklerin sıralanması konusu incelenmektedir. Literatürde sıkça kullanılmakta olan boyut indirgeme yöntemleri bu bölümde ele alınacaktır. Dördüncü bölümde, karar fonksiyonları, Bayes sınıflayıcı ve çeşitli öbek araştırma yöntemleri anlatılmaktadır. Beşinci bölümde, yapay sinir ağlarının sınıflama amacı ile kullanımı anlatılmaktadır. Bu bölümde öznitelik uzayında doğrusal olmayan sınıf dağılımlarının yapay sinir ağları ile temsil edilmesi incelenmektedir. Altıncı bölümde, öncelikle temel genetik işlemlerin bilgisayar ortamına yapay uyarlaması ve bu işlemlerin optimizasyon içindeki rolleri açıklanmaktadır. Bölümün sonuna doğru yapay sinir ağlarının genetik algoritmalar ile eğitimi

konusunda bilgi verilmektedir. Yedinci bölümde, önce bulanık küme teorisi incelenecek ve duru küme teorisiyle arasındaki ilişkiler gösterilecektir. İlerleyen alt bölümlerde sınıflayıcılar için iki farklı öğrenme algoritması açıklanacaktır: Bulanık öbek-ortalama ve bulanık Kohonen öbekleme algoritması. Bölümün sonunda ise bulanık en yakın prototip ve bulanık K-en yakın komşu sınıflayıcıları anlatılacaktır. Sekizinci bölümde sembolik süreçler ve yapay zeka algoritmaları üzerinde durulmaktadır. Yapay zeka sistemlerinde kavramları öğrenme, durum uzayında araştırma, kurallar kümesini öğrenme, karar ağacını öğrenme, takviyeli öğrenme, keşif yoluyla ve benzerlik yoluyla öğrenme konularında bilgi verilmektedir. Ayrıca bölüm sonunda, incelenen kavramların pratik gerçekleşmesi için PROLOG programlama ile ilgili temel bilgiler verilmektedir.

Bu kitabın, örüntü tanıma konusuna ilgi duyan okuyucularımıza faydalı olmasını diler, kendilerinden gelecek öneri ve eleştiriler için şimdiden teşekkür ederiz.

Bilimsel çalışmalarımızın şekillenmesi ve olgunlaşmasında önemli katkıları bulunan değerli hocalarımız Prof. Dr. Ertuğrul Yazgan ile Doç. Dr. Mehmet Korürek'e ve araştırma görevlileri yük. müh. Ayhan Yüksel ile yük. müh. Zafer İşcan'a teşekkür ederiz.

Üzerimizde büyük emekleri olan, daimi ilham kaynağımız sevgili ailelerimize de en içten teşekkürlerimizi sunarız.

İstanbul, 2009

Tamer Ölmez ve Zümray Dokur Ölmez

BİRİNCİ BÖLÜM

TEMEL KAVRAMLAR

Bu bölümde örüntü (*pattern*) uzayı, bu uzaydaki örüntüler, örüntülerin vektörel temsili (özniteliklerin belirlenmesi), vektörlerin normalize edilmesi, vektörlerin öznitelik uzayında dağılımı (sınıf tanımı), karar fonksiyonları (sınıflayıcı yapısı) ve makine eğitimi gibi temel kavramlar hakkında bilgiler verilecektir.

Karar verme amacıyla bilgisayarda geliştirilen algoritmalarından çevremizde gelişen tüm olaylara insan gibi cevap vermesini bekleyemeyiz. Örüntü tanıma, eğitim ve tanıma işlemine başlamadan önce örüntülerin doğasına bağlı olarak çevremizde gelişen olayları gruplandırmamız gerekir. Ses işaretinden konuşmacıları tanıma, konuşulan kelimeleri tanıma, kamera karşısındaki objeleri tanıma işlemlerinin her biri ayrı bir konu olarak ele alınacaktır. Karar verme amacıyla geliştirilen algoritmalar, genellikle seçilen bir konu içindeki örüntüler hakkında karar üretecektir. Örüntü uzayı; tanıma işleminde seçilen konuyu (amaç), bu uzaydaki örüntüler ise varlığı sorgulanan bilgileri temsil etmektedir.

Örüntü uzayını ve örüntüleri birlikte açıklamak amacıyla birkaç örnek verelim: Konuşma tanıma işlemi, “açıl, kapan” gibi komutlar; biyolojik işaretlerin tanınması işlemi, normal ve hastalık belirtisi içeren işaretler; yürüten hat üzerindeki cisimlerin kamera ile tanınması işlemi, “somun, vida” gibi cisimler; tarayıcı ile oluşturulan döküman görüntüsünden karakter tanıma işlemi, rakamlar ve harfler; örüntü uzayı ve bu uzaydaki örüntüleri temsil etmektedir. Dikkat edilecek olursa örnek olarak verilen dört konuda da amaç farklıdır.

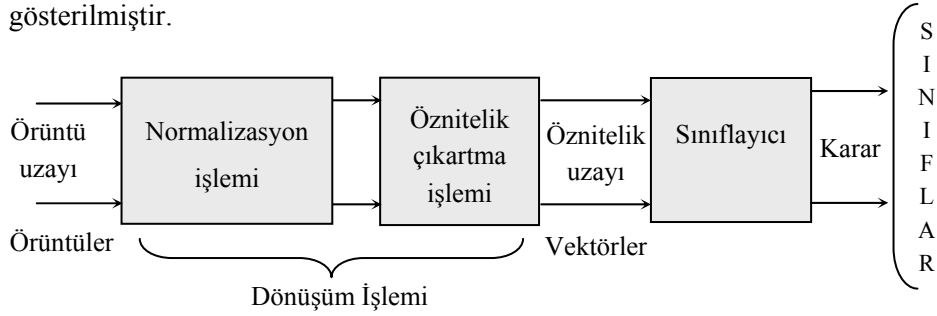
Bilgisayarların örüntüleri algılayabilmesi için bu uzaydaki her bir örüntünün, bilgisayarın işleyebileceği bir vektörel form ile temsil edilmesi gerekmektedir. İncelenen her konu için vektörler, farklı bir dönüşüm yöntemi kullanılarak oluşturulacaktır. Örüntüler, dönüşüm yapıldıktan sonra bilgisayarın işleyebileceği forma gelmektedir. Dönüşüm işlemini, kategorileri birbirinden ayıran ve kendi kategorisini en iyi temsil eden özniteliklerin bulunması olarak düşünebiliriz.

Örnek olarak verilen dört tanıma işleminde de öncelikle kategori (sınıf) sayısı belirlenir. Bu sayı, bilgisayarın örüntüleri kaç gruba ayıracağını

göstermektedir. Tarayıcı ile alınan dökümanlar sadece rakamlar içeriyorsa, kategori sayısı 10 (0,1...9) olarak belirlenecektir. Döküman içinde 10'dan fazla karakter bulursa bile görüntüdeki tüm işaretler 10 gruba ayrılacaktır. Yürüyen hat üzerinde ayırt edilecek cisim sayısı, kategori sayısı olarak tanımlanacaktır. Tüm işlemlerde kategori sayısı, tasarımcının ihtiyacına göre belirlenmektedir. Kategori sayısının artması bilgisayarın örüntüleri algılama gücünün artmasına yol açar. Ancak, kategori sayısı ile doğru algılama performansı arasında genellikle ters bir ilişki vardır. Kategori sayısını artırma, doğru algılama performansını düşürür.

1.1 KARAR VERME MEKANİZMASI

Giriş bölümünde anlatılan kavramlar, aşağıda bloklar şeklinde gösterilmiştir.



ŞEKİL 1.1 Karar verme mekanizması.

Karar verme mekanizmasını temel olarak üç blok şeklinde düşünebiliriz: Normalizasyon işlemi, öznitelik çıkartma işlemi ve sınıflama işlemi. Bazı durumlarda karar verme mekanizmasına sembolik süreçlerle de destek verilebilmektedir. Bu durumda elde edilen sınıflar/kategoriler sembollerle temsil edilir ve sembollerle ilgili gerçekleştirilecek süreçler karar verme mekanizmasına ilave edilir. Kitap içinde örüntü tanıma konusu ağırlıklı olarak incelendiği için, aksi belirtilmediği sürece karar verme mekanizması ile şekil 1.1'de gösterilen bloklar ifade edilmektedir.

Şekil 1.1'de gösterilen dönüşüm işlemi, normalizasyon ve öznitelik çıkartma işlemlerinin birleşimi olarak ele alınmaktadır. Şekil 1.1'de görülen normalizasyon ve öznitelik çıkartma işlemlerinin sırası bazı durumlarda yer değiştirebilmektedir.

Genellikle karar verme mekanizmasına ait parametrelerin belirlenmesinde az sayıda vektör içeren bir alt küme kullanılmaktadır. Bu nedenle öznitelik uzayında sınıfların dağılımının matematiksel ifadesini tam olarak çıkartmak mümkün değildir ve bulunması için de bir uğraş verilmez. Alt küme içinde vektörlerin sayısı ne kadar fazla ise öznitelik uzayındaki dağılımın temsili o kadar iyi olacaktır. Karar verme mekanizmasının

genelleme yeteneği, alt küme dışındaki vektörler (örüntüler) için Şekil 1.1'deki bloğun doğru cevap verme sayısı olarak tanımlanmaktadır. Karar verme mekanizmasının genelleme yeteneğini artırmak için genellikle alt kümenin büyütülmesi yolu tercih edilmektedir.

Örüntü uzayına ait her örüntü, öznitelik uzayında bir noktaya karşılık gelir ve bir vektör ile temsil edilir. Öznitelik uzayındaki vektörlerin sayısı, uzay boyutuna ve her bir boyutun nicemlenme miktarına bağlıdır. n -boyutlu uzayda her bir boyut q adet ayrık değer ile temsil edilirse, q^n adet vektör bulunmaktadır. 10-boyutlu uzayda her bir boyut 256 ayrık değer ile tanımlanırsa, öznitelik uzayı 256^{10} adet vektör içerecektir. Genellikle sınıflayıcı tasarımında seçilen alt kümedeki vektörlerin sayısı, 1000 veya 10000'ler mertebesinde. Eğitim kümesinin küçük seçilmesi yapılan çalışmanın istatistiksel olarak anlamını düşürmektedir. Bu durumda genelleme yeteneği, dönüşüm işleminin (normalizasyon + öznitelik çıkartma) seçimine direkt bağlı olacaktır.

1.2 NORMALİZASYON İŞLEMİ

Örüntülere ait bilginin, öznitelik uzayındaki vektörlere direkt olarak aktarılmasını engelleyen üç sebep vardır: (i) Örüntüleri temsil eden vektör elemanlarının sayısının değişimi (öznitelik uzayı boyutu), (ii) elemanların vektör içindeki pozisyonun değişmesi (elemanların vektör içinde kayması) ve (iii) aynı kategoriden örüntüler için vektör elemanlarının genliğinin öznitelik uzayında büyük bir değişim göstermesi. Vektör elemanları üzerindeki bu bilgi taşımayan anlamsız etkiler, sınıflamadan önce normalizasyon işlemi ile ayarlanmalı ve sabitlenmelidir.

Bir örnek üzerinde problemi ortaya çıkartalım. Konu olarak EKG (elektrokardiyogram) vurularının (normal veya hastalık belirtisi taşıyan işaretler şeklinde) sınıflandırılması işlemi seçilmiş olsun. Vurular iki kısma ayrılacağı için kategori sayısı iki olacaktır. Her vuru, bir örneği temsil eder ve normal ya da anormal (hastalık belirtisi) bilgisini taşır. Öznitelik çıkartma yöntemi olarak EKG vurularının sabit bir pencere içindeki (pencere, sadece bir vuruya ait verileri içerecek şekilde seçilsin) direkt genlik değerlerinin kullanıldığını düşünürsek, öznitelik vektörünün elemanları EKG işaretinin zaman içindeki ayrık genlik değerlerinden meydana gelecektir. Vektörlere aktarılan bilgiyi bozan anlamsız etkiler -yukarıda belirtilen sıra ile- aşağıdaki şekilde maddelenebilir.

Atım sayısı (dolayısıyla, bir EKG periyodunun süresi) yaşa, cinsiyete ve kişinin o andaki fiziksel durumuna bağlı olarak bir değişim göstermektedir. Örnekleme frekansını sabit düşünelim. Bu durumda bir vuru periyodu içindeki veri sayısı her denek için farklı olacaktır (öznitelik vektörü boyutunun değişimi).

Vektörün ilk elemanını oluşturmak için bir yol belirlenmelidir. Aksi takdirde her oluşturulan vektör için vektör elemanları farklı genlik değerleri alacak, elemanların vektör içindeki pozisyonu kayacaktır.

EKG vurularının genlik değerleri yaşa ve cinsiyete göre değişmektedir. EKG vurularının genlikleri bazı deneklerde -benzer sağlık durumunda olmalarına rağmen- düşük olurken, bazılarında ise son derece yüksek olabilmektedir. Bu durum vektör elemanları üzerinde yanıltıcı bir etki olarak gözlenecektir.

Öznitelik çıkartma işlemi son derece basit olmasına karşın vektörler üzerinde yukarda maddelenen yanıltıcı (sınıfları ayırıcı bilgi taşımayan) etkiler oluşmaktadır. Bu etkiler, normalizasyon işlemi ile öznitelik vektörü üzerinden kaldırılmalıdır.

Sınıflayıcı yapılarında öznitelik uzayının boyutu sabit olduğu için vektör elemanlarının sayısı önceden uygun bir değere getirilmelidir. Vektör elemanlarının kendi içinde sırası önemli değildir; ancak elemanların diğer vektörlerdeki elemanlara göre sırası aynı olmalıdır. Genellikle öznitelik uzayında sınıf dağılımına ait anlamlı bilgi, vektör elemanlarının birbirlerine göre genlik dağılımında saklıdır.

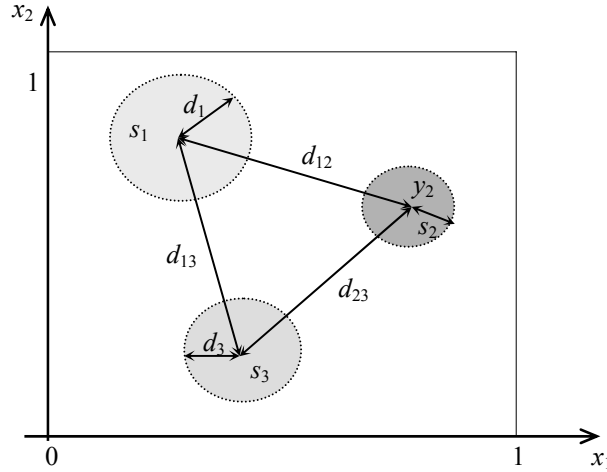
Bazı durumlarda alt kümeye ait tüm vektörlerin bazı elemanlarının diğer elemanlara göre çok küçük değerler aldığı gözlemlenebilir. Bir vektör elemanının tüm vektörler için küçük değerler alması, o elemanın vektörden çıkartılması anlamına gelmemelidir. Boyut azaltmak için bu elemanı tüm vektörlerden çıkartmak yerine, bu elemanın değeri, sabit bir katsayı ile çarpılarak diğer elemanlarla aynı seviyeye getirilmelidir. Öznitelik uzayındaki sınıf dağılımı, alt kümedeki tüm vektörlerin sabit bir K katsayısı ile çarpılması ile değiştirilmiş olmamaktadır. Aynı kural vektörlerin tüm elemanları yerine bazı elemanlarına uygulandığı zaman da geçerlidir. Bu kural ile vektör elemanlarının kendi içinde normalize edilebilmesi sağlanmaktadır. Ancak vektör elemanlarının normalizasyonunda, tüm vektörler aynı işlem içinde bulunmalıdır. Oluşturulan alt küme içindeki vektörlerin bazıları bir K_1 katsayısı ile, diğerleri de farklı bir K_2 sabit katsayısı ile çarpılırsa, yeni oluşuma ait dağılım doğal olarak orjinal kümeye ait dağılımdan farklı olacaktır. Normalizasyon işleminin bu özelliği veri ve boyut analizi bölümünde ayrıntılı olarak incelenecektir.

1.3 ÖZNİTELİK ÇIKARTMA İŞLEMİ

Özniteliklerin çıkartılması işlemi, isminden de anlaşılacağı üzere sadece bir kategoriye (sınıfa) özgü olan ve diğer kategorilerde bulunmayan karakteristik bilgilerin tespit edilmesi işlemidir. Kategorileri birbirlerinden ayıran doğru özniteliklerin tespit edilmesi, sınıflama performansının artması üzerinde olumlu bir etki oluşturacaktır. Seçilen her konu için doğru özniteliklerin bulunmasına yönelik matematiksel bir yöntem yoktur ve çoğu

durumda öznitelikler tasarımcının sezgisine bağlı olarak tespit edilir. Alt kümedeki vektör sayısının küçük oluşu, özellikle karar verme mekanizmasını olumsuz yönde etkilemektedir. Bu etki özniteliklerin doğru belirlenmesi ile bir ölçüde ortadan kaldırılır.

Özniteliklerin belirlenmesi kavramını, iki-boyutlu üç kategoriden oluşan bir örnek öznitelik uzayında (*fantom uzay*) inceleyelim. Özniteliklerin doğru belirlenmesinden sonra, fantom uzayda vektörlerin dağılımı Şekil 1.2'ye benzemektedir. Farklı kategorilerin birbirlerinden mümkün olduğunca uzakta olduğu ve aynı kategoriye ait vektörlerin bir arada bulunduğu gözlenir. Öznitelik çıkartma işleminin başarımını; d_1, d_2, d_3 'ün küçük değerlerde, d_{12}, d_{13} ve d_{23} 'ün de büyük değerlerde oluşu belirleyecektir. Böyle bir sınıf dağılımına sahip fantom uzay için lineer olmayan yapıya sahip sınıflayıcılar kullanmak gereksizdir.



ŞEKİL 1.2 İki-boyutlu örnek öznitelik uzayı ve sınıfların dağılımı.

1.4 BOYUT İNDİRGEME VE VERİ ANALİZİ

Kullanılacak öznitelik çıkartma yöntemi genellikle tasarımcının sezgisine bağlı olarak tespit edilir. Kategorilerin yeterince iyi temsil edilebilmesi için uzay boyutu mümkün olduğunca büyük tutulur. Uzay boyutunun büyümesi dağılımın daha kolay temsil edilebilmesini sağlayacaktır. Ancak öznitelik uzayındaki vektörlerin bazı elemanları (bazı öznitelikler) anlamlı bilgi taşımayacaktır. Anlamsız öznitelikler sınıflayıcı yapısının aşırı büyümesine sebep olacaktır. Bu özniteliklerin, dağılımın yapısı incelenerek ayıklanması (böylece vektör boyutunun küçültülmesi) gerekmektedir.

Öznitelik uzayında sınıfların saçılımını ve sınıflar-arası uzaklığı incelemek üzere değişik yöntemler önerilmektedir. Bu yöntemler içinde

diverjans analizinin sıkça kullanıldığı gözlenmektedir (Bölüm 3’e bakınız). Eşitlik (1.1)’de diverjans hesabında kullanılan denklem verilmiştir.

$$Diverjans = \text{tr}((C^{a-1})^{-1} A^{a-1}) \quad (1.1)$$

Burada C^{a-1} matrisi, $(a-1)$ boyutlu öznitelik vektörlerinin sınıf içi saçılımını; A^{a-1} matrisi ise vektörlerin sınıflar arası saçılmalarını temsil etmektedir.

Diverjans değeri, seçilen öznitelikler ile oluşturulan vektörlerin saçılımı hakkında bir bilgi vermektedir. Diverjans değeri tek başına bir anlam içermez, bu işlem incelenecek yöntemleri karşılaştırma amacıyla kullanılmalıdır. Küçük diverjans değerleri, vektörlerin öznitelik uzayında saçıldığıнын, büyük değerler ise vektörlerin bu uzayda öbekleştiğinin bir göstergesidir. Vektörlerin öznitelik uzayında saçılması, sınıf sınırlarının doğru belirlenmesini engelleyecek ve sınıflama performansının düşmesine sebep olacaktır. Bu nedenle vektörlerin sınıf-içi dağılımını küçük, sınıflar-arası uzaklığını büyük tutacak, yani diverjans değerini büyük yapacak en iyi öznitelikler araştırılır.

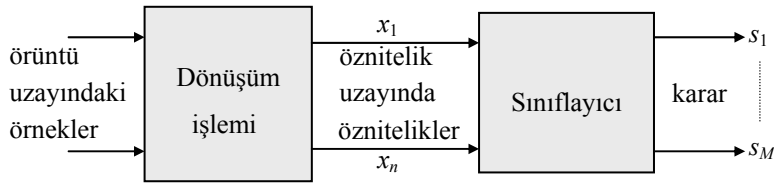
Dinamik programlama ile vektörlerin boyutu 1’den n sayısına kadar adım adım artırılır. Her adımda en iyi öznitelikler diverjans hesaplanarak bulunur. Vektör boyutunu artırmak diverjans değerini büyütecektir. Dolayısıyla büyük diverjans değerine sahip dağılımlar için daha iyi sınıflama başarıımı elde edilecektir. Ancak, boyut arttığı için hesap yükünde de (sınıflayıcının bilgisayarda bir algoritma şeklinde oluşturulduğu düşünüldüğü zaman) bir artış gözlenecektir.

1.5 FANTOM ÖZNİTELİK UZAYI

Bu alt bölümde yeni geliştirilen sınıflayıcı yapılarını incelemek için test amaçlı iki-boyutlu bir fantom uzay önerilmiştir. Örnek öznitelik uzayının matematiksel ifadesi aşağıdaki gibi tanımlanabilir:

$$F(x_1, x_2, \dots, x_n) = \{s_1, s_2, \dots, s_M \mid s_i \in Z, \quad x_i \in R\} \quad (1.2)$$

Bu ifadelerde M kategori (sınıf) sayısını, n öznitelik uzayının boyutunu, s_i sınıfları birbirinden ayıran sınıf etiketini, \bar{x} ise n -boyutlu öznitelik vektörünü ifade etmektedir. Şekil 1.3’te eşitlik (1.2)’deki parametrelerin karar verme mekanizmasındaki yeri gösterilmiştir.



ŞEKİL 1.3 Karar verme mekanizmasının bileşenleri.

Doğal olarak $F(x_1, x_2)$ fonksiyonu, iki boyutta bir görüntü oluşturacaktır. Test için iki-boyutlu bir öznitelik uzayının seçilmesi, fonksiyonun dağılımının incelenmesine ve sınıflama sonucunun görsel olarak karşılaştırılmasına olanak sağlayacaktır.

Fantom uzay, eşitlik (1.2)'deki formül gereği herhangi bir fonksiyon olabilir. Ancak sınıflayıcı yapılarının yeteneğini araştırmak için bu fonksiyon üzerinde bir ölçüde sınırlamalar getirilecektir. Örnek olarak bir $F(x_1, x_2)$ fonksiyonunu üç sınıf ($M=3$) için inceleyelim.

$$F(x_1, x_2) = \{s_1, s_2, s_3 \mid s_1, s_2, s_3 \in Z; x_1, x_2 \in R\} \quad (1.3)$$

Yukardaki fonksiyonun Şekil 1.2'deki gibi bir sınıf dağılımına sahip olduğunu düşünürsek öznitelik çıkartma işleminin başarımını, d_1 , d_2 , d_3 'ün küçük değerlerde d_{12} , d_{13} ve d_{23} 'ün büyük değerlerde oluşu belirleyecektir. Böyle bir sınıf dağılımında, sınırları temsil etmek için lineer yapıya sahip olmayan sınıflayıcılar kullanmak gereksizdir.

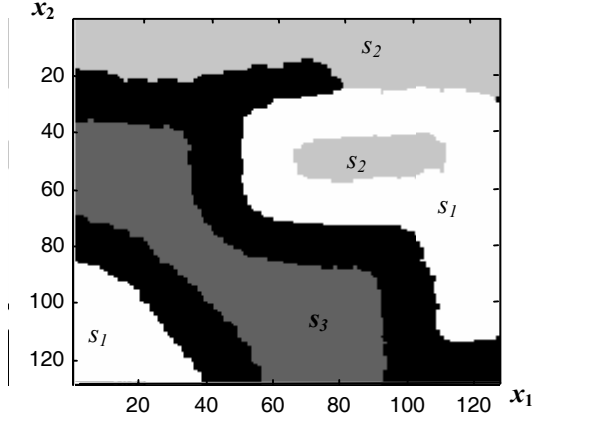
Yeteri kadar örüntü (vektör) her zaman elde edilemediği için (dolayısıyla eğitim kümesinde bulunan vektör sayısı azdır) sınıf dağılımını ifade eden bir fonksiyon tam olarak bulunamaz. Bazı durumlarda öznitelik uzayında, aynı sınıftan farklı bölgelerde öbekleşmeler hatta, sınıfların birbirlerinin içine girdiği karmaşık dağılımlar oluşabilmektedir.

Şekil 1.4'te üç sınıflı bir fantom uzay ($F(x_1, x_2)$ fonksiyonu) tanımlanmıştır. (x_1, x_2) , öznitelik uzayındaki vektörlere karşılık gelmektedir. Her vektör, fantom uzayda bir nokta ile gösterilmektedir. Bu uzayda, herbir sınıf etiketi farklı bir renge karşı düşürülmüş ve vektörler, ait oldukları sınıflara özgü beyaz, açık-gri ve koyu-gri renkler ile görselleştirilmiştir. Siyah renkle ifade edilen bölgeler hiçbir sınıfa dahil değildir. Fantom uzayda siyah bölgelerin bulunması sınıflayıcıların yükünü bir ölçüde hafifletmektedir. Sınıf sınırları tanımlanırken bu bölgeler için karar sınıflayıcılara bırakılır, bir zorlama/koşul getirilmez. Öznitelik uzayındaki tüm noktaların tanımlı olması, sınıf sınırlarının belirlenmesinde sınıflayıcılara hiç bir kolaylık vermeyen en kötü durumu temsil etmektedir. Doğada incelenen çoğu olayda boş, tanımsız bölgeler de mevcuttur.

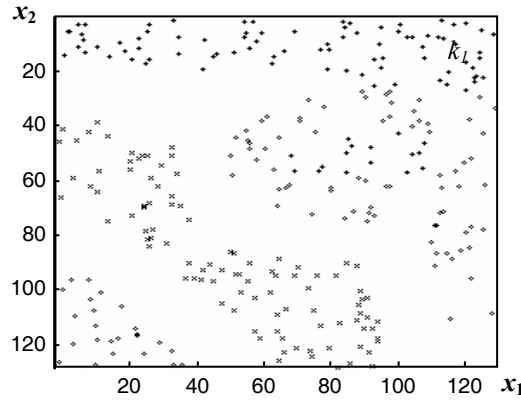
Bazı sınıfların fantom uzayda birbirlerine bitişik olması veya aynı sınıfa ait vektörlerin farklı yerlerde öbekleşmeler göstermesi, öznitelik çıkartma işleminin başarılı olamaması durumunu içeren bir örnek teşkil etmektedir. Şekil 1.2 ve 1.4 için bulunan diverjans değerleri sırasıyla 27.83 ve 0.49'dur. Küçük diverjans değeri (0.49), fantom öznitelik uzayında vektörlerin saçıldığının sayısal bir göstergesidir.

Fantom uzay sayesinde sınıflayıcı yapılarını oluşturan eğitim algoritmalarının stratejisi, yapıların bu uzayı bölütleme yetenekleri, geliştirilen eğitim algoritmalarının problemsiz işlerliği ve yapıların genelleme özelliği görsel olarak analiz edilecektir. Şekil 1.4'teki iki-boyutlu uzaydan elde edilen

alt-küme Şekil 1.5'te gösterilmektedir. Her sınıf için 100 olmak üzere, alt-küme toplam 300 vektörden meydana gelmiştir.



ŞEKİL 1.4 İki-boyutlu fantom öznelik uzayında sınıfların dağılımı.



ŞEKİL 1.5 İki-boyutlu fantom uzaydan elde edilen alt-küme.

1.6 KARAR FONKSİYONLARI

Öznelik uzayında sınıfları birbirinden ayırmak için sınıf sınırlarını temsil edebilecek fonksiyonlar tanımlanmaktadır. Karar fonksiyonlarını oluşturmak için mesafe ölçüleri kullanılmaktadır. Literatürde en bilinen mesafe ölçüsü aşağıda gösterilmektedir:

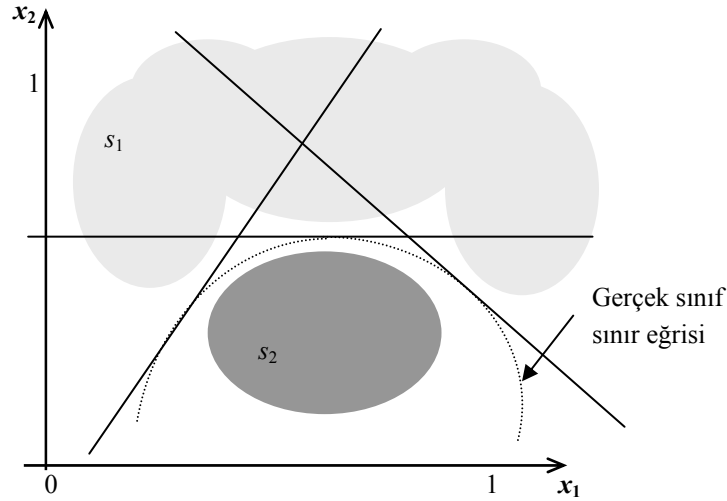
$$d(\bar{x}) = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n \quad (1.4)$$

\bar{x} , n -boyutlu öznitelik uzayında giriş vektörünü temsil etmektedir. $d(\bar{x})$ fonksiyonu, $d(\bar{x})=0$ 'da n -boyutlu uzayda bir *hiper-düzlem* tanımlamaktadır.

Hiper-düzlem öznitelik uzayını iki parçaya bölmektedir. \bar{x} vektörü bu iki parçadan biri içinde ise düzleme olan uzaklığına bağlı olarak $d(\bar{x})$ fonksiyonunun değeri işareti ile birlikte değişmektedir. $d(\bar{x})$ fonksiyonu, aşağıdaki özelliklere sahip bir fonksiyondan geçirilirse öznitelik uzayı iki parçaya etiketlenerek ayrılacaktır.

$$K(\bar{x}) = T(d(\bar{x})) = \begin{cases} +1 & d(\bar{x}) \geq 0 \\ -1 & \text{aksi takdirde} \end{cases} \quad (1.5)$$

Karar fonksiyonu ($K(\bar{x})$) ile birbirlerinden uzakta öbekleşmiş iki sınıfa ait vektörler kolaylıkla ayrılabilir. Ancak Şekil 1.4'teki gibi bir dağılımda karar sınırları eğrilerle tanımlanabilmektedir, sadece bir doğru kullanmak sınırı oluşturmak için yeterli olmayacaktır. Bu durumda sınıf sınırının oluşturulmasına birden fazla lineer karar fonksiyonu kullanılarak yaklaşılabacaktır. Bu yaklaşım, sınıflayıcı yapılarında katmanlaşmaya doğru gidilmesine sebep olmaktadır. Şekil 1.6'da iki-boyutlu fantom uzayda bir sınır eğrisini oluşturmak için doğruların birleştirilme şekli gösterilmektedir. Şekilde sınıf sınır eğrisi, doğru parçalarının birleşimi ile elde edilmiştir.



ŞEKİL 1.6 Doğrularla sınıf sınır eğrisini oluşturma.

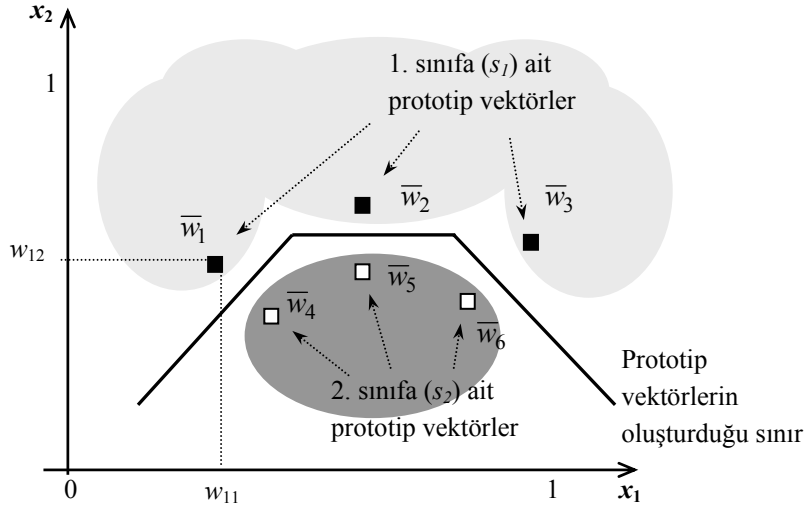
Eğrisel bir karaktere sahip sınıf sınırlarını (çok-boyutlu uzayda sınıf yüzeylerini) temsil etmek için doğru parçalarının birleştirilmesi yaklaşımı

(sınır, fonksiyonları tanımlı basit eğri parçaları ile de oluşturulabilir) tek çözüm değildir. (1.4) eşitliği aynı zamanda mesafe ölçüsü olarak tanımlanmaktadır. Literatürde çok sayıda mesafe ölçüsü kullanılmaktadır. Eşitlik (1.6)'da sıkça kullanılan *minimum mesafe ölçüsünün* tanımı verilmiştir. Bu ölçü ile öznitelik uzayındaki sınıf sınır eğrileri, doğrularla yapılan parça parça yaklaşıma benzemektedir.

$$d_j(\bar{x}) = \sum_{i=1}^n (x_i - w_{ji})^2 \quad (1.6)$$

$$\text{Min} = \min_{j=1, \dots, L} (d_j(\bar{x}))$$

Eşitlik (1.6)'da L , prototip vektörlerin (yapay sinir ağlarında düğümlerin) sayısını; w_{ji} , j -inci prototip vektörün i -inci ağırlığını göstermektedir. Eşitlik (1.4)'te ağırlıklar ve giriş vektörünün elemanları arasında lineer bir ilişki vardır. (1.6) eşitliğinde ise ağırlıklar ile giriş vektörü elemanları arasında ikinci dereceden bir ilişki vardır. Eşitlik (1.4)'te sınır, mesafe ölçüsünün sıfıra eşit değerlerinde, eşitlik (1.6)'da ise sınır yüzeyi (mesafe ölçüsünde karşılaştırma işlemi bulunduğu için) farklı sınıftan en yakın iki prototip vektöre eşit mesafede ortaya çıkmaktadır. Şekil 1.7'de *minimum mesafe* kullanımı ile sınıf sınırlarının oluşturulması iki-boyutlu, iki sınıflı fantom uzayda gösterilmektedir.



ŞEKİL 1.7 Minimum mesafe ölçüsü ile sınıf sınırlarının oluşturulması.

Yukarıda sınıf sınırlarının (çok-boyutlu uzayda sınıf yüzeylerinin) belirlenmesine yönelik verilen iki açıklama, literatürde *yapay sinir ağları* ile oluşturulan sınıflayıcı yapılarına temel örnek teşkil etmektedir. Sınıf sınırlarını bulma işlemi, genel olarak eğri (çok boyutlu uzayda yüzey) uydurma problemine dönüştürülebilir. Farklı mesafe ölçüleri ile değişik sınıflayıcı yapıları (yeni yapay sinir ağları) elde etmek mümkündür (Bölüm 4'e bakınız).

1.7 SINIFLAYICI YAPILARI VE EĞİTİM

Alt-kümedeki vektörlerin öznitelik uzayında Şekil 1.5'te gösterildiği gibi dağıldığını düşünelim. Sınıflayıcıların görevi bu dağılımı en iyi temsil edecek şekilde sınıf sınırlarını oluşturmaktır. Şekil 1.2'deki fantom uzaydan da görüleceği üzere özniteliklerin doğru belirlenmesi durumunda sınıflayıcıların sınırları belirleme görevi kolayca gerçekleştirilebilir. Ancak Şekil 1.4'teki dağılıma ait sınıf sınırları, bir önceki fantom uzayda olduğu gibi kolayca belirlenemez. Sınıflara ait özniteliklerin iyi belirlenememesi durumunda doğru sınıflama performansı, sınıflayıcıların öznitelik uzayındaki dağılımları temsil edebilme yeteneklerine bağlıdır. Örneğin, Şekil 1.4'teki dağılım lineer sınıflayıcı yapıları ile temsil edilemez.

Sınıflayıcı yapısını karar fonksiyonları belirlemektedir. Yapı, karar fonksiyonlarının fiziksel olarak gerçekleştirilmesinden ibarettir. Sınıflayıcı yapısını tanımlayan karar fonksiyonlarının parametreleri, genellikle alt kümedeki vektörler kullanılarak belirlenmektedir. Bu kitap kapsamında parametreler, alt kümeden rastgele alınan vektörler ile iteratif olarak belirlenecektir.

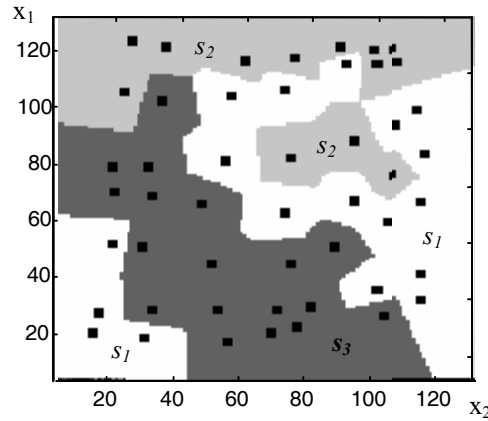
Eşitlik (1.6)'daki ölçü (karar fonksiyonu) kullanılarak Şekil 1.4'teki fantom uzay sınıflanmış, sonuç Şekil 1.8'de gösterilmiştir. Şekildeki siyah noktalar prototip vektörlerin bu uzaydaki konumlarını göstermektedir. Şekil 1.4'teki siyah bölgeler için sınıf kararları sınıflayıcının tercihi bırakılmıştır. Sınıflayıcı, Şekil 1.4'teki sınıf dağılım bilgisini prototip vektörlerin ağırlık değerleri ile saklamaktadır.

Sınıflayıcı yapısını karar fonksiyonları belirlediği için makine eğitimi, bu fonksiyona ait parametrelerin bulunması şeklinde tanımlayabiliriz. Eğitim algoritması için bir amaç fonksiyonu belirlenir ve bu amaç fonksiyonunu sağlayacak şekilde karar fonksiyonunun parametreleri adım adım değiştirilir. Şekil 1.9'da amaç fonksiyonu '*sınıflama hatasını minimize etmek*' şeklinde tanımlanırsa, eğitim ile karar fonksiyonunun parametreleri iki sınıfı birbirinden ayıracak duruma gelene kadar adım adım değiştirilerek çözüme ulaşılır.

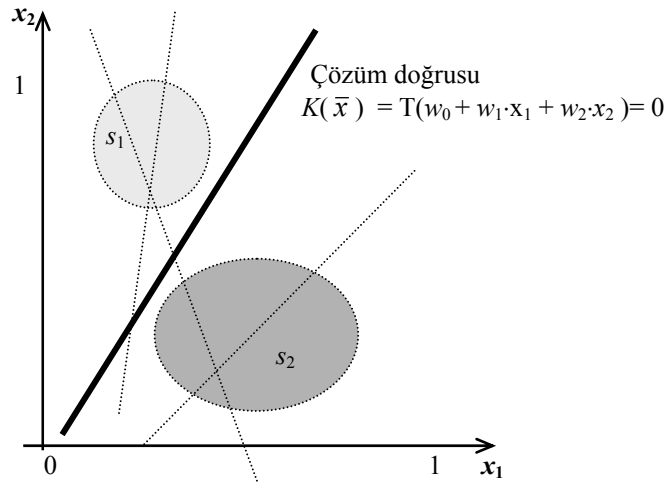
Şekil 1.9'daki dağılım için karar fonksiyonu olarak aşağıdaki ifade kullanılmıştır.

$$K(\bar{x}) = T(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2) = 0 \quad (1.7)$$

Eğitimin basit olarak anlaşılması için yine iki sınıflı iki-boyutlu bir fantom uzay örnek olarak verilmiştir. Eğitim işleminde w_0 , w_1 ve w_2 parametrelerinin amaç fonksiyonu için en uygun değerleri (kalın çizgi) bulunur. En uzun süreli eğitimi, bu üç değişkene tek tek değer vermek olarak tanımlayabiliriz. n tane parametrenin herbirinin alabileceği değer sayısı q ise q^n tane rastgele değer vererek arama gerçekleştirilebilir. Geliştirilen her yeni eğitim algoritması bu arama adımından daha az sayıda adım ile sonuca ulaşmalıdır.



ŞEKİL 1.8 İki-boyutlu fantom uzayın eşitlik (1.6) kullanılarak sınıflanması.



ŞEKİL 1.9 Parametrelerin eğitimle değişimi. Noktalı çizgiler, önceki iterasyonlarda bulunan hatalı sınır doğrularını göstermektedir.

1.8 MAKİNE ÖĞRENMESİ

Örüntü tanıma işlemlerinde temel olarak genellikle üç blok kullanılmaktadır. Kitap içersinde bu üç blok, karar verme mekanizması olarak ifade edilmektedir. Bazı durumlarda karar üretmek için bu üç blok yeterli olmamaktadır.

Kalp hastalıklarını inceleyen bir sistemi bu duruma örnek olarak verilebiliriz. Kalp rahatsızlıklarını inceleyen sistem için çok sayıda parametre/durum göz önüne alınmalıdır. Elektrokardiyogram, kalbin elektriksel faaliyetleri hakkında bilgi vermektedir. Ekokardiyogram veya fonokardiyogram ise kalbin mekanik hareketleri hakkında bilgi vermektedir. Hastanın kilosu, yaşı, cinsiyeti, sigara veya alkol kullanma alışkanlığı, daha önceden geçirdiği bazı hastalıklar, ailesinde gözlenen kalp hastalıkları, kan yağı ve şekeri ile ilgili bilgiler de kalp rahatsızlığının belirlenmesinde büyük bir öneme sahiptir. Dikkat edilirse, her bilgi farklı bir domenden gelmektedir. Karar verme mekanizmaları kullanılarak elektrokardiyogram incelenir ve her normal veya anormal şekle sahip vuru bir sembolle ifade edilir. Daha sonra tespit edilen anormal vuruların tipleri ve sıklığı bir veri tabana kaydedilir. Aynı şekilde karar verme mekanizmaları kullanılarak eko görüntüler incelenir ve önemli damar hemo-dinamikleri ile kapakcık hareketleri sembollerle ifade edilip aynı veri tabanına kaydedilir. Kalbin anlık elektriksel ve mekanik hareketleri dışında hastaya ait diğer bilgiler (hastanın fizyolojik, alışkanlıkları, ailesinde ve daha önceden geçirdiği hastalıklar hakkında bilgiler) de aynı veri tabanına yerleştirilir. Son olarak uzman sistem oluşturulan bu veri tabanını ve sembolik mantıktaki bazı süreçleri kullanarak hasta hakkında bir karar üretir.

Kamera ile satranç tahtasını gözleyip oyun oynayan uzman sistemini de yine bu duruma bir örnek olarak gösterebiliriz. Uzman sistem önce kamera ile satranç tahtasının görüntüsünü elde eder. Daha sonra bu görüntü üzerinde karar verme mekanizmaları kullanılarak taşların tipleri ve bu taşların tahta üzerindeki konumları tespit edilir (örüntü tanıma süreçleriyle). Bu bilgiler, sembolik olarak bir veri tabanında ifade edilir. Uzman sistem daha sonra oluşturulan veri tabanı ve bazı sembolik süreçleri kullanarak kendi hamlesini gerçekleştirir.

Düşünme yeteneğimiz incelendiğinde dış dünyadan aldığımız verileri işleyip sembollerle tanımladığımız (örüntü tanıma süreçleri) ve bu semboller ile bilgi birikimimizi kullanarak olaylar üzerinde bir karar oluşturduğumuz bilinmektedir. Bu bağlamda kitabın yedinci bölümde bir bakıma sembolik mantığa bir geçiş sağlanmakta karar verme süreçlerinde sembollerin kullanılmasına örnekler verilmektedir. Kitabın sekizinci bölümünde ise sembolik mantıkta kullanılan bazı önemli süreçler ve öğrenme algoritmaları anlatılmaktadır.

1.9 PERFORMANS KRİTERLERİ VE TASARIM

Karar verme mekanizmasına ait blokların tasarımında dikkate alınacak birkaç özellik aşağıda verilmiştir.

- i. Öğrenme süreçlerinde kullanılan parametre veya ağırlıkların iteratif olarak belirlenmesi
- ii. Karar verme mekanizmasındaki yapıların eğitim sırasında ihtiyaca göre büyümesi
- iii. Eğitimin kısa sürede gerçekleştirilebilmesi
- iv. Eğitim sonrasında sınıflayıcının az sayıda prototip vektörü ile gerçekleştirilmesi
- v. Karar verme mekanizmasının yüksek genelleme özelliği (yeni vektörleri doğru sınıflama performansı)
- vi. Tasarlanan sistemin fiziksel olarak kolaylıkla gerçekleştirilebilmesi

Seçilen bir örüntü tanıma işleminde karar verme mekanizmasının geliştirilmesinde aşağıdaki işlemlerin takip edilmesi tasarımcıya fayda sağlayacaktır.

Örüntü uzayı için öncelikle kategori sayısı belirlenir, daha sonra kategorileri birbirlerinden en iyi ayıran öznitelikler çıkartılır. Öznitelik çıkartma işleminin seçimi için literatürde belirli bir yöntem yoktur, seçim tasarımcının tecrübesine bağlıdır. Tasarımcı öznitelik vektörlerinin saçılmamasını dikkate alacak şekilde bir yöntem belirlemelidir. Seçimde örnek olması açısından, insanın doğal olarak bu ayrımı nasıl gerçekleştirdiği dikkate alınmalıdır. Yeni özniteliklerin ayırım gücünü bulmak ve diğer bilinen öznitelik çıkartma yöntemleri ile karşılaştırmak için diverjans analizi kullanılabilir. Bu analiz, aynı zamanda incelenen örüntüler için en iyi özniteliklerin bulunmasını da (boyut azaltma işlemi) sağlayacaktır.

Araştırma yeni sınıflayıcı yapısı tasarlamaya yönelikse, sınıflayıcı gerçek dünya problemini çözmede kullanılmadan önce fantom uzayda denenmelidir. Bu deneme ile hem gerçekleştirilen programın yazılım olarak doğruluğu test edilir hem de eğitimin ve yapının diğer yöntemlerle görsel olarak karşılaştırılması sağlanır. Son olarak, fantom uzayda test edilen yeni sınıflayıcı yapısı ya da eğitim algoritması benzetim ortamında gerçek örüntüler için kullanılır.

Gerçek-zaman uygulamaları için geliştirilecek yöntemler, öncelikle MATLAB veya LABVIEW gibi işaret işleme kütüphanesi zengin benzetim araçlarında incelenmeli daha sonra algoritma C veya makine kodunda yazılarak mikro-denetleyici üzerinde çalıştırılmalıdır. Gerçek zaman uygulaması olarak düşünülen yöntemler için algoritma geliştirmede özellikle eğitim/sınıflama zamanı ve bellek ihtiyacı dikkate alınmalıdır.

PROBLEMLER:

Bu alt bölümdeki bilgisayar ödevleri, fantom uzay için eğitim kümesi oluşturulmasına yöneliktir.

1.1) Fantom uzayda sınıf sayısı ve sınıflara ait vektörlerin dağılımı araştırıcının seçimine bırakılmıştır. Sınıflara ait vektörlerin birbirlerine yaklaştığı, sınıf sınırlarının iç içe geçtiği durumu ele almak doğru sınıf sınırlarını tespit etmede sınıflayıcıları zorlayıcı bir etken olacaktır. Bu nedenle fantom uzay için sınıfların öbekleşme karakteri gösterdiği dağılımları oluşturmak yerine; bu uzay için konveks dağılımları içeren daha karmaşık vektör dağılımlarının oluşturulması tercih edilmelidir (örneğin Şekil 1.4).

a) Resim oluşturma veya boyama programları kullanılarak iki-boyutlu ve üç sınıf içeren bir fantom uzayı temsil edecek “jpeg” veya “bmp” uzantısında bir dosya oluşturun. Vektörlerin $x1$ ve $x2$ elemanlarının değerlerinin [0-255] aralığında değişmesi istenmektedir.

b) Örnek fantom uzay içinden seçilen öznitelik vektörleri ile bir eğitim kümesi oluşturulacaktır. Bilgisayarın faresi yardımıyla her sınıftan 20 vektör içeren bir eğitim kümesi oluşturulacaktır.

c) Fare kullanılmadan rastgele örnekleme ile her sınıftan 100 vektör içerecek şekilde bir eğitim kümesi oluşturulacaktır.

Eğitim kümesi bilgisayarda “txt” uzantılı bir dosya şeklinde saklanmalıdır (vektör elemanları “text” okuyucu yardımıyla kolaylıkla gözlemlenecektir). Eğitim kümesindeki her vektör, elemanları ve etiketi tek bir satırda bulunacak şekilde gösterilmelidir. Önceden oluşturulan örnek fantom uzayın ismi, benzetim programına dışardan parametre olarak verilecektir.

KAYNAKLAR

- [1] JAE S. L. (1990): *Two-Dimensional Signal and Image Processing*, Prentice-Hall International Inc. .
- [2] DİMİTRIS G. M., VINAY K. I., STEPHEN M. K. (2000): *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modelling, Adaptive Filtering and Array Processing*, Mc-Hill.
- [3] JONATHAN Y. S. (2000): *Digital Signal Processing*, John-Wiley & Sons. Inc. .
- [4] AKRAM A., MICHAEL U. (1996): *Wavelets in Medicine and Biology*, CRC Press. .
- [5] POLİKAR R. (2001): *The wavelet tutorial*, <http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html>, Rowan University .
- [6] RAO K. R., YIP P. (1990): *Discrete Cosine Transform: Algorithms, Advantage, Application*, Academic Press. Inc. .
- [7] ANİL K. J. (1989): *Fundamentals of Digital Image Processing*, Prentice Hall Inc. .
- [8] ROBERTO M. C. J., LUCIANO F. C. (2000): *Shape Analysis and Classification Theory and Practice*, CRC Press LLC .
- [9] AMADASUN M. AND KING R. (1989): *Textural features corresponding to textural properties*. IEEE Trans Sys Man Cyber, 19(5), 1264-1274.
- [10] HARALICK R. M., SHANMUGAN K. AND DINSTEN I. (1973): *Texture Feature for Image Classification*, IEEE Tran. on Systems Man and Cybernetics, 3, 610-621.
- [11] COHEN A. (1986): *Biomedical Signal Processing*, vol. II, CRC Press Inc. .
- [12] DEVIJVER P. A., KITTLER J. (1982): *Pattern Recognition: A Statistical Approach*, Prentice-Hall .
- [13] TOU J. T., GONZALES R. C. (1974): *Pattern Recognition Principle*, Addison-Wesley Pub. Comp.
- [14] GEOFFREY J. M. (1992): *Discriminant Analysis and Statistical Pattern Recognition*, John-Wiley & Sons. Inc. .
- [15] Duda R. O., Hart P. E., Stork D. G. (2001): *Pattern Classification*, John Wiley & Sons, New York.
- [16] KAUFMAN L. AND ROUSSEEUW P.J. (1990): *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, New York.

- [17] DUDA R. O. AND HART P. E. (1973): *Pattern Classification and Scene Analysis*, John-Wiley & Sons. Inc. .
- [18] HELVE A., DOMINIQUE V., BETTY E. (1999): *Neural Networks*, Sape Publication .
- [19] LAURENE V. F. (1994): *Fundamentals of Neural Networks*, Prentice-Hall.
- [20] MAKHOUL J., JARAUDI A. E., SCHWARTZ R. (1989): *Formation of Disconnected Region with a Single Hidden Layer*, IJCNN International Conference on Neural Networks, Cilt 1, 192-202.
- [21] NIELSON R. H. (1989): *Neurocomputing*, Addison-Wesley Publishing Company.
- [22] REILLY D. L., COOPER L. N., ELBOUM C. (1982): *Neural Model for Category Learning*, Biological Cybernetics, Cilt 45, 35-41.
- [23] ALPAYDIN E. (1990): *Neural Models of Incremental Supervized and Unsupervised Learning*, Ds. Thesis, Ecole Polytechnique Federale De Lausanne, Switzerland.
- [24] LIPPMAN R. P. (1987): *An Introduction to Computing with Neural Nets*, IEEE ASSP Magazine, 4-22.
- [25] KURNAZ M.N., DOKUR Z., ÖLMEZ T. (2001): *Segmentation of Ultrasound Images by Using an Incremental Self-Organized Map*, 23rd Annual Int. Conference of the IEEE-EMBS, Istanbul.
- [26] PANDYA A. S., MACY R. B. (1995): *Pattern Recognition with Neural Networks in C++*, CRC Press.
- [27] LEE B.W. and SHEU B.J. (1988): *An Investigation of Local Minima of Hopfield Network for Optimization Circuits*, IEEE International Conference on Neural Networks, I45-I51.
- [28] GOLDBERG D. E. (1989): *Genetic Algorithms in Search Optimization and Machine Learning*, Addison Wesley Pub. Comp. Inc. .
- [29] SANKAR K. P., PAUL P. W. (1996): *Genetic Algorithms for Pattern Recognition*, CRC Press.
- [30] WOLFGANG B., PETER N., FRANK D. F., ROBERT E. K. (1998): *Genetic Programming: An Introduction*, Morgan Kaufmann Publishers.
- [31] YUKSEL, A., DOKUR, Z., KORUREK, M. AND OLMEZ T. (2008): *Modeling of Non-Uniform Intensity Distribution of X-Ray Source in Radiographic Images*, ISCIS08. Makale Numarası: 185.
- [32] DOKUR Z. (2000): *Yapay Sinir Ağları ve Genetic Algoritmalar Kullanılarak EKG Vurularının Sınıflandırılması*, Doktora Tezi, İTÜ Fen Bilimleri Enstitüsü.
- [33] KAYNAK, O. ve ARMAĞAN, G. (1992): *Bulanık Denetim ve Endüstriyel Uygulamaları*, Tübitak MAM, 35-55.
- [34] SUGENO M. (1985) : *An Introductory Survey of Fuzzy Control*, Information Science, Cilt 36, 59-83.

- [35] WITOLD P. (1989): *Fuzzy Control and Fuzzy Systems*, John-Wiley & Sons. Inc. .
- [36] STAMATIOS V. K. (1995): *Understanding Neural Networks and Fuzzy Logic : Basic Concepts and Applications*, Wiley-IEEE Press.
- [37] BEZDEK J. C. (1973): *Fuzzy Mathematics in Pattern Classification*, Ph.D. Dissertation, Cornell University, Ithaca, NY.
- [38] BEZDEK J. C. (1992): *Computing with Uncertainty*, IEEE Communication Magazine, 24-36.
- [39] KELLER J. M., GRAY M. R. AND GIVENS J. A. (1985): *A Fuzzy K-Nearest Neighbor Algorithm*, IEEE Tran. on Systems Man and Cybernetics, Cilt SMC-15, numara: 4.
- [40] YUICHIRO A. (1992): *Pattern Recogniton and Machine Learning*, Academic Press, INC. .
- [41] MICHELL M. T. (1997): *Machine Learning*, The MacGraw-Hill Companies, INC. .
- [42] BRATKO I. (2001): *Prolog Programming for Artificial Intelligence*, Addison-Wesley.

CEVAP ANAHTARI

Birinci bölümden sekizinci bölüme kadar olan bölüm sonu sorularının cevapları, MATLAB (uyarlama 7.1) ortamında koşan benzetim programları ile gerçekleştirilir. Benzetim programlarından gerçek zamanda sonuçlar üretmesi beklenmez. MATLAB gibi ortamlar araştırmacıya, geliştirilen algoritmaların sonuçlarını karşılaştırma ve yeni yöntemler araştırma imkanı sunmaktadır. Bölüm sonu sorularında üç farklı işaret için ayrı ayrı analizler yapılmaktadır: i) Fantom uzay, ii) EKG vuruları ve iii) CT görüntüsü. Fantom uzay, geliştirilen algoritmaların mantıksal hatalarını tespit etmek, iki boyuttan yüksek öznelik boyutuna kolaylıkla geçilmesini sağlamak ve eğitimin başarısını test etmek için kullanılmaktadır. İşaretleri doğada tek boyutlu işaretler veya görüntüler şeklinde iki farklı tipte gözlemlemekteyiz. Bu nedenle bölüm ödevlerinde iki farklı tipte işaret örnek olarak analiz edilmektedir. Dördüncü bölümden sekizinci bölüme kadar bölüm içi cevaplarda seçilen sınıflayıcı yapıları ve eğitim süreçleri aynıdır; ancak seçilen işaretler farklıdır (dolayısıyla sadece öznelik vektörlerinin belirlenişi farklıdır). Bu yaklaşım, ödevlerdeki sınıflayıcı yapılarını ve eğitim süreçlerini gerçekleyen yazılımları okuyucunun daha kolay inceleyebilmesine olanak sağlamaktadır. Ayrıca algoritmaların/programların rahat anlaşılması için benzetim programı içinde fazlaca değişken ve özel fonksiyon kullanılmamıştır.

Sekizinci bölümün bölüm sonu soruları, SWI-PROLOG (uyarlama 5.2.10) ortamında koşan benzetim programları ile gerçekleştirilir. Sembolik mantıkta gerçekleştirilecek süreçlerin prolog ortamında koşturulması daha uygundur. Bölüm içinde tanıtılan kavramları açıklayan örneklerin yapıları da, prolog derleyicilerinde biçimsel hata vermeyecek şekilde düzenlenmiştir.

- 1.a)İki boyutlu fantom uzay, üç sınıf içermektedir. Fantom uzay, farklı formatta “G3” ve “G3.jpeg” isimli iki dosya içinde bulunmaktadır. Benzetim programının kaynak kodu, “fu_oznelik.m” dosyasında saklanmıştır.
- 1.b)Benzetim programı imleci, fare ile fantom uzay üzerinde istenilen pozisyona konumlandırılır. İstenilen konuma ait öznelik vektörü, farenin sol düğmesine basılarak eğitim kümesi içine yazılır. Eğitim kümesi “fu_mous.txt” dosyası içinde saklanmaktadır. Şekil p.1’de “fu_mous.txt”

dosyası içinde saklanan iki boyutlu öznitelik vektörlerinin formatı gösterilmektedir.

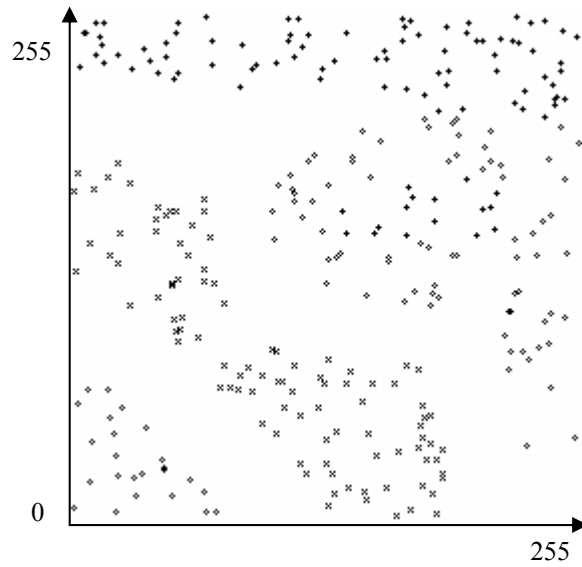
```

18  94  50
54  101 50
30  135 50
76  146 50
65  171 50
111 172 50
102 211 50
.....
39  16  100

```

ŞEKİL p.1 “fu_mous.txt” dosyasına ait ilk satırlar.

- 1.c) Benzetim programı, her sınıftan rastgele 100 öznitelik vektörü olmak üzere üç sınıf için toplam 300 vektör içeren bir eğitim kümesi oluşturur. Eğitim kümesi, “fu_rand.txt” dosyası içinde saklanmaktadır. Benzetim programı, öznitelik vektörlerinin rastgele dağılımını göstermek için “Ornek_kume.jpeg” isiminde bir görüntü dosyası oluşturur. Şekil p.2’de “Ornek_kume.jpeg” dosyası gösterilmektedir.



ŞEKİL p.2 Eğitim kümesi içinden rastgele seçilen vektörler.

2.1) Benzetim programının kaynak kodu, “Normalize.m” dosyasında saklanmıştır. Program öncelikle X_tipi_vurular.txt (X=N, L, R ve P) dosyasından EKG işaretinin genliklerini okur ve bir dizi içine yazar. Program daha sonra ardışıl süzgeçler kullanarak vuru dizisindeki tüm EKG’lerin R tepelerini tespit eder ve bu tepeleri farklı bir dizide işaretleyerek saklar. Programın diğer bölümü, R tepesi işaretlerini ve sabit bir gecikmeyi dikkate alarak kestirilen R tepesi etrafından (-64 , +64) toplam 128 genlik değeri olarak EKG işaret vektörünü oluşturur.

Şekil p.3’te R tepesi etrafında 128 genliğe sahip EKG vuruları topluca gösterilmektedir. Şekil p.3’teki her vektör, gerçek kayıttaki bir EKG vurusuna karşılık gelmektedir. Bu işlem ile R tepesinin vektörlerin ortasında yer alması sağlanmıştır. Daha sonra vektörlerin ortalamasını sıfır ve gücünü 1 değerine normalize edecek şekilde işlemler gerçekleştirir. 40 adet öznitelik vektörü, zamanda ve genlikte normalize edilmiş tüm vuruları temsil etmek üzere vektör elemanları ve etiketi bir satırda olacak şekilde “X-küme.txt (X=N, L, R ve P)” uzantılı bir dosyada saklanır.

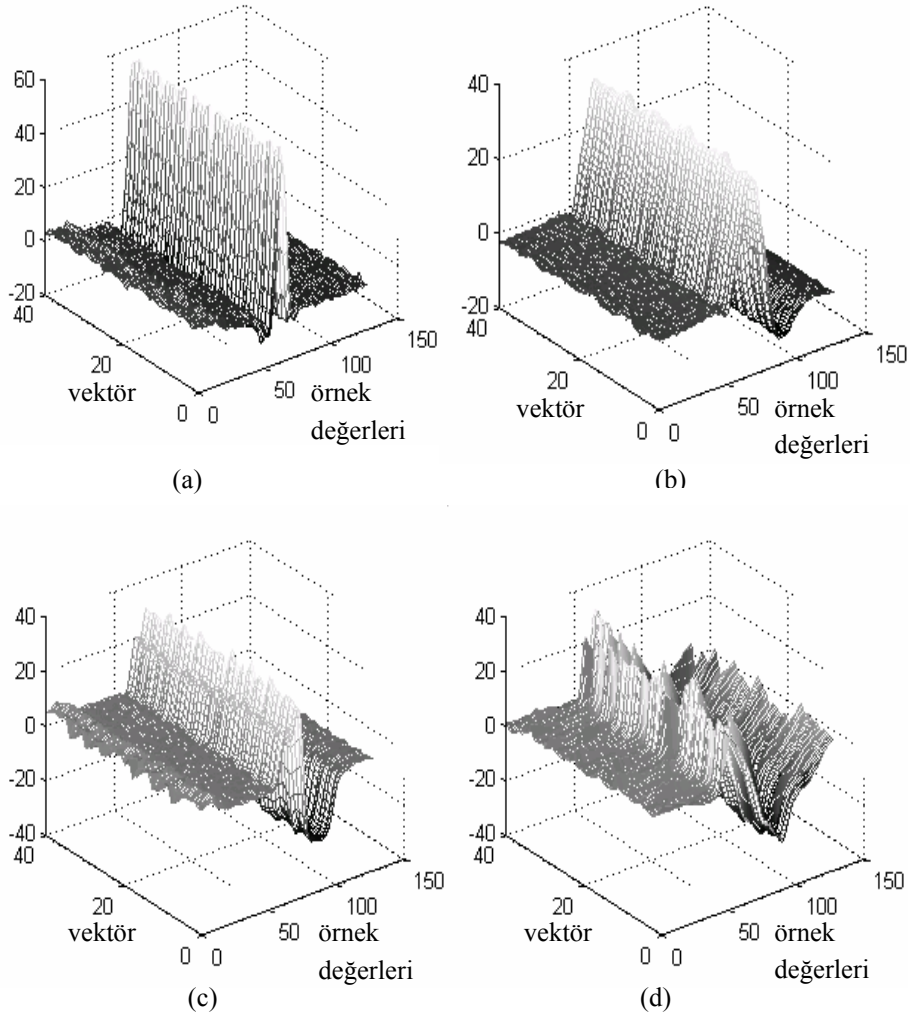
2.2) Benzetim programının kaynak kodu, “oz_frekans.m” dosyasında saklanmıştır. “X-küme.txt” dosyasından normalize edilmiş vuruları temsil eden EKG vektörleri okunur ve bir diziye yazılır. Her vektöre Fourier dönüşümü uygulanır. Fourier katsayılarının modülü hesaplanır ve bu değerler, öznitelik vektörü elemanları olarak farklı bir dizide saklanır. Tüm EKG vuru vektörleri için hesaplama tamamlandığında öznitelik vektörleri, vektör elemanları ve etiketi bir satırda olacak şekilde “X_oz_frekans.txt (X=N, L, R ve P)” uzantılı bir dosyada saklanır.

2.3) Benzetim programının kaynak kodu, “kume_olustur.m” dosyasında saklanmıştır. “X_oz_frekans.txt (X=N, L, R ve P)” uzantılı dört dosya okunur ve üç dizi oluşturulur.

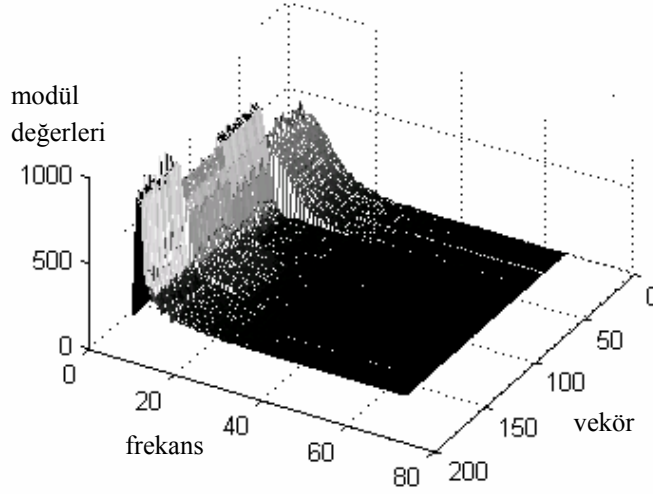
64 modül değerinden oluşan dört sınıfa ait 160 adet öznitelik vektörü, vektör elemanları ve etiketi bir satırda olacak şekilde “oz_fre_toplam.txt” uzantılı bir dosyada saklanır. Tüm EKG vurularına ait öznitelik vektörleri, Şekil p.4’te gösterilmektedir. 64 modül değerinden oluşan dört sınıfa ait 80 adet öznitelik vektörü eğitim kümesini temsil etmek üzere, vektör elemanları ve etiketi bir satırda olacak şekilde “oz_fre_egitim.txt” uzantılı bir dosyada saklanır. 64 modül değerinden oluşan dört sınıfa ait 80 adet öznitelik vektörü test kümesini temsil etmek üzere, vektör elemanları ve etiketi bir satırda olacak şekilde “oz_fre_test.txt” uzantılı bir dosyada saklanır.

2.4) Benzetim programının kaynak kodu, “oz_ct.m” dosyasında saklanmıştır. Eğitim kümesi dört sınıf içerecek şekilde “ct.jpeg” görüntüsü üzerinde fare

kullanılarak oluşturulur. Fare yardımıyla işaretlenen beneğin kendisi ve bir komşuluğundaki benek şiddetleri öznitelik vektörünün elemanlarını oluşturmaktadır. Dört sınıfa ait 40 adet öznitelik vektörü eğitim kümesini temsil etmek üzere, vektör elemanları ve etiketi bir satırda olacak şekilde “oz_ct.txt” uzantılı bir dosyada saklanır. Aynı şekilde eğitim kümesinden farklı bir test kümesi de oluşturulabilmektedir.



ŞEKİL p.3 (a) N tipi, (b) L tipi, (c) R tipi ve (d) P tipi vurulara ait kümeler.

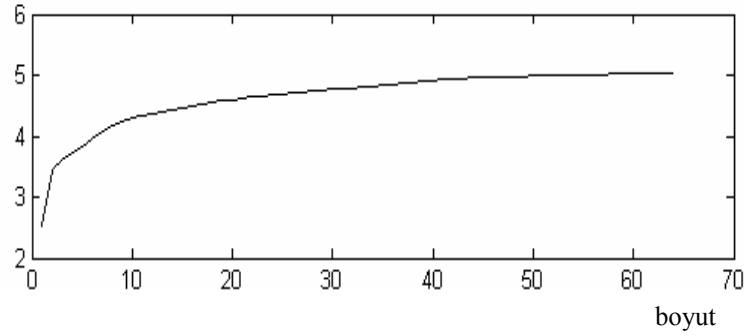


ŞEKİL p.4 EKG vurularına ait öznitelik vektörleri.

- 3.1) Benzetim programının kaynak kodu, “Boyut_azalt.m” dosyasında saklanmıştır. “oz_fre_toplam.txt” dosyası içindeki tüm öznitelik vektörleri dikkate alınarak, vektörlerin boyutu 1’den 64’e kadar adım adım genişletilir. Genişletme işleminde daha anlamlı öznitelikler (diverjans değeri dikkate alınarak) ilk sıralarda olacak şekilde bir sıralama gerçekleşir. Öznitelik vektörü boyutu ve diverjans değerlerinin logaritması, Şekil p.5’te gösterilmiştir. Eğride ilk kırılma noktası olan 32, indirgenmiş boyut değeri olarak seçilmiştir. Sıralama ve indirgenen vektör boyutu dikkate alınarak düzenlenen yeni öznitelik vektörleri, vektör elemanları ve etiketi bir satırda olacak şekilde “Boyut_frekans.txt” uzantılı bir dosyada saklanır. Özniteliklerin anlam sırası, öznitelik numarası ve diverjans değeri bir satırda gösterilecek şekilde “fre_siralama.txt” dosyasında saklanır.
- 3.2) Benzetim programının kaynak kodu, “Boyut_azalt.m” dosyasında saklanmıştır. “oz_ct.txt” dosyası içindeki tüm öznitelik vektörleri dikkate alınarak, vektörlerin boyutu 1’den 9’a kadar adım adım genişletilir. Genişletme işleminde daha anlamlı öznitelikler (diverjans değeri dikkate alınarak) ilk sıralarda olacak şekilde bir sıralama gerçekleşir. Öznitelik vektörü boyutu ve diverjans değerlerinin logaritması, Şekil p.6’da gösterilmiştir. Eğride ilk kırılma noktası olan 4, indirgenmiş boyut değeri olarak seçilmiştir. Sıralama ve indirgenen vektör boyutu dikkate alınarak düzenlenen yeni öznitelik vektörleri, vektör elemanları ve etiketi bir satırda olacak şekilde “Boyut_ct.txt” uzantılı bir dosyada saklanır.

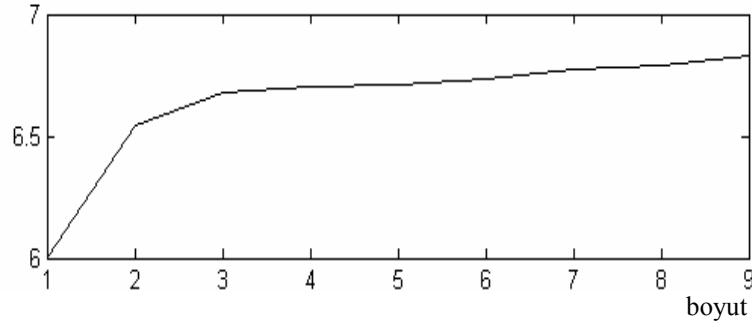
Özniteliklerin anlam sırası, öznitelik numarası ve diverjans değeri bir satırda gösterilecek şekilde “ct_sıralama.txt” dosyasında saklanır.

log (diverjans)



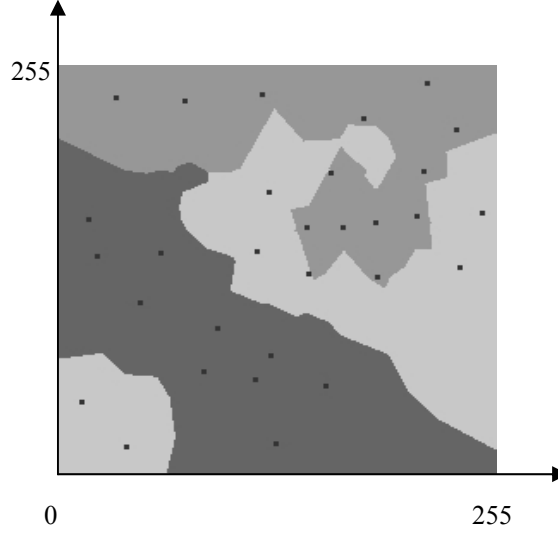
ŞEKİL p.5 Diverjans değerinin boyut sayısına göre değişimi.

log (diverjans)



ŞEKİL p.6 Diverjans değerinin boyut sayısına göre değişimi.

4.1) Sınıflayıcı olarak K-en yakın komşu (literatürde K-NN olarak bilinmektedir) sınıflayıcısı seçilmiştir. K değeri 3 olarak seçilmiştir. Benzetim programının kaynak kodu, “fu_KNN_sınıflama.m” dosyasında saklanmıştır. Fantom uzaya ait öznitelik vektörleri, “fu_mous.txt” dosyasından okunur. Eğitim kümesindeki tüm öznitelik vektörleri ağın düğümleri olarak seçilir. Dolayısıyla sınıflayıcıya ait parametreler karmaşık işlemler yapılmadan kolaylıkla bulunmuştur. Test için özel bir küme oluşturulmamıştır. Fantom uzaydaki tüm vektörler, K-NN sınıflayıcı tarafından sınıflandırılmıştır. Vektörlere ait etiketler, vektörlerin konumlarına göre ekranda görüntülenmiştir. Sonuç görüntü, “fu_KNN_sınıflama.jpeg” dosyasında saklanmıştır. Şekil p.7’de K-NN sınıflayıcının ürettiği kararlar görüntülenmektedir.

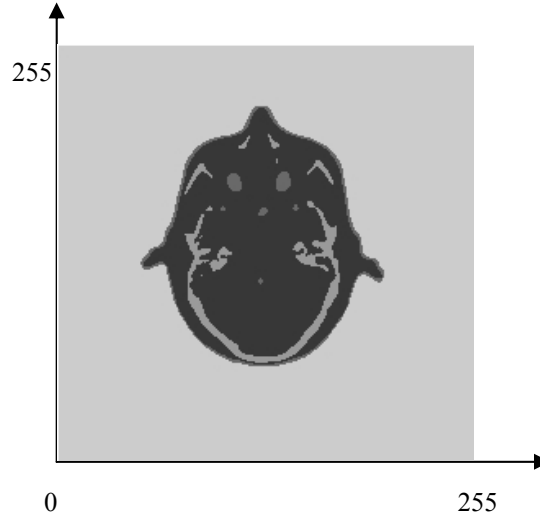


ŞEKİL p.7 Fantom uzayın KNN sınıflayıcı ile sınıflandırılması.

4.2) Sınıflayıcı olarak K-en yakın komşu (K-NN) sınıflayıcısı seçilmiştir. K değeri 3 olarak seçilmiştir. Benzetim programının kaynak kodu, “fre_KNN_sınıflama.m” dosyasında saklanmıştır. Eğitim kümesine ait öznitelik vektörleri, “oz_fre_egitim.txt” dosyasından okunur. İngirgenmiş boyuta ait bilgiler “fre_siralama.txt” dosyasından okunur. Eğitim kümesindeki tüm öznitelik vektörleri, ağın düğümleri olarak seçilir. Dolayısıyla sınıflayıcıya ait parametreler karmaşık işlemler yapılmadan kolaylıkla bulunmuştur. Test kümesine ait tüm vektörler, “oz_fre_test.txt” dosyasından okunur. Test kümesine ait tüm vektörler, K-NN sınıflayıcı tarafından sınıflandırılmış ve %100 başarı elde edilmiştir.

4.3) Sınıflayıcı olarak K-en yakın komşu (K-NN) sınıflayıcısı seçilmiştir. K değeri 3 olarak seçilmiştir. Benzetim programının kaynak kodu, “ct_KNN_sınıflama.m” dosyasında saklanmıştır. Eğitim kümesine ait öznitelik vektörleri, “oz_ct.txt” dosyasından okunur. İngirgenmiş boyuta ait bilgiler “ct_siralama.txt” dosyasından okunur. Eğitim kümesindeki tüm öznitelik vektörleri, ağın düğümleri olarak seçilir. Dolayısıyla sınıflayıcıya ait parametreler karmaşık işlemler yapılmadan kolaylıkla bulunmuştur.

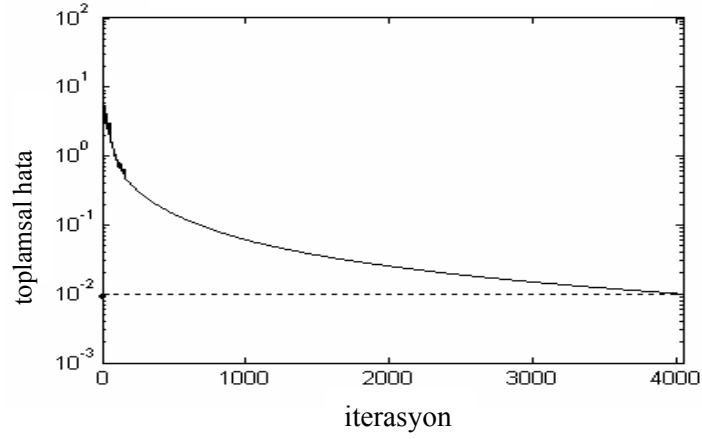
Test için farklı bir küme oluşturulmamıştır. CT görüntüsündeki tüm benekler için K-NN sınıflayıcı karar (etiket) üretir ve benekler için üretilen etiketler ekranda görüntülenir. Sonuç görüntü, “ct_KNN_sınıflama.jpeg” isimli bir dosyada saklanmıştır. CT görüntüsündeki beneklerin K-NN sınıflayıcı tarafından sınıflandırılması Şekil p.8’de gösterilmiştir.



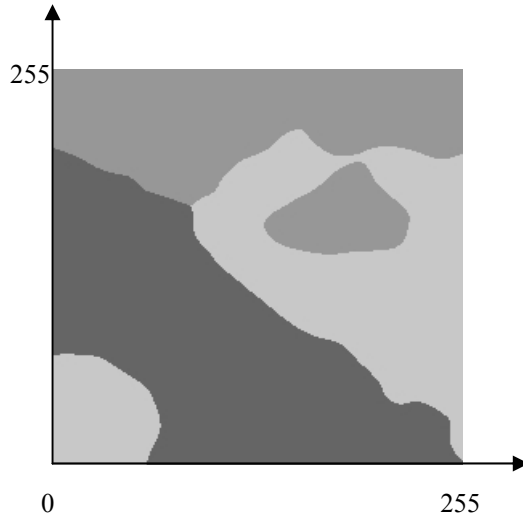
ŞEKİL p.8 CT görüntüsünün KNN sınıflayıcı ile sınıflandırılması.

5.1) Sınıflayıcı olarak literatürde sıkça kullanılan çok-katmanlı ağ (ÇKA) seçilmiştir. Benzetim programının kaynak kodu, “fu_MLP_sınıflama.m” dosyasında saklanmıştır. Fantom uzaya ait öznitelik vektörleri, “fu_mous.txt” dosyasından okunur. P dizisi, giriş öznitelik vektörleri kümesini; T dizisi, arzu edilen çıkış değerleri kümesini temsil etmektedir. Ağın topolojisi, 2-170-180-3 olarak bulunmuştur. Bu topoloji, çok sayıda denemeden sonra hatayı en aza indirgeyen yapı olarak tespit edilmiştir. Eğitim aşamaları, Şekil p.9’de grafik şeklinde gösterilmiştir. Düğümler (W1 B1 W2 B2 W3 B3) “kayit.mat” dosyasında saklanmıştır. ÇKA’nın fantom uzayını sınıflama sonucu Şekil p.10’da gösterilmektedir. Sonuç görüntü, “fu_MLP_sınıflama.jpeg” dosyasında saklanmıştır.

5.2) Sınıflayıcı olarak literatürde sıkça kullanılan çok-katmanlı ağ (ÇKA) seçilmiştir. Benzetim programının kaynak kodu, “fre_MLP_sınıflama.m” dosyasında saklanmıştır. EKG vurularına ait öznitelik vektörleri, “oz_fre_egitim.txt” dosyasından okunur. P dizisi, giriş öznitelik vektörleri kümesini; T dizisi, arzu edilen çıkış değerleri kümesini temsil etmektedir. Ağın topolojisi, 32-150-180-4 olarak bulunmuştur. Bu topoloji, çok sayıda denemeden sonra hatayı en aza indirgeyen yapı olarak tespit edilmiştir. Eğitim aşamaları, Şekil p11’de grafik şeklinde gösterilmiştir. Düğümler (W1 B1 W2 B2 W3 B3) “kayit.mat” dosyasında saklanmıştır. Test kümesi “oz_fre_test.txt” dosyasından okunur.



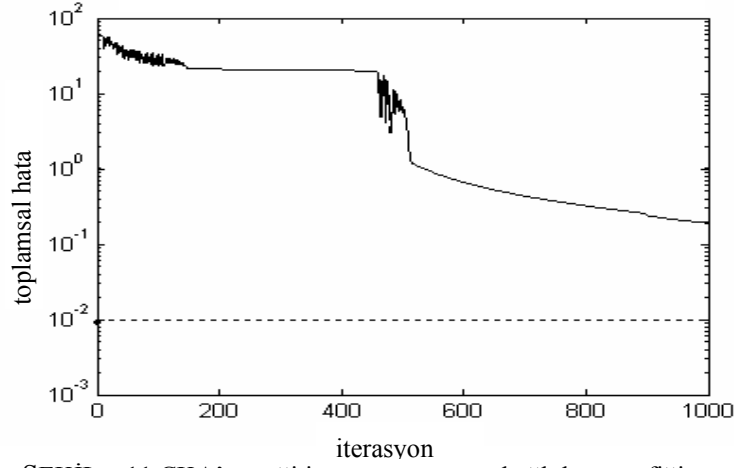
ŞEKİL p.9 ÇKA'nın eğitiminde iterasyona bağlı hata grafiği.



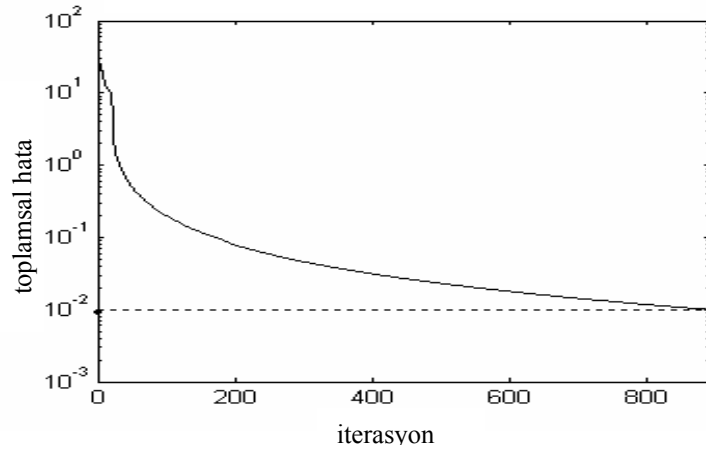
ŞEKİL p.10 Fantom uzayın ÇKA ile sınıflandırılması.

5.3) Sınıflayıcı olarak literatürde sıkça kullanılan çok-katmanlı ağ (ÇKA) seçilmiştir. Benzetim programının kaynak kodu, "ct_MLP_sınıflama.m" dosyasında saklanmıştır. CT görüntüsüne ait öznelik vektörleri, "oz_ct.txt" dosyasından okunur. P dizisi, giriş öznelik vektörleri kümesini; T dizisi, arzu edilen çıkış değerleri kümesini temsil etmektedir. Ağın topolojisi, 4-220-230-4 olarak bulunmuştur. Bu topoloji, çok sayıda denemeden sonra hatayı en aza indirgeyen yapı olarak tespit edilmiştir. Eğitim aşamaları, Şekil p12'de grafik şeklinde gösterilmiştir. Düğümler

(W1 B1 W2 B2 W3 B3) “kayit.mat” dosyasında saklanmıştır. ÇKA’nın CT görüntüsündeki benekleri sınıflama sonucu Şekil p13’te gösterilmektedir. Sonuç görüntü, “ct_MLP_sınıflama.jpeg” dosyasında saklanmıştır.



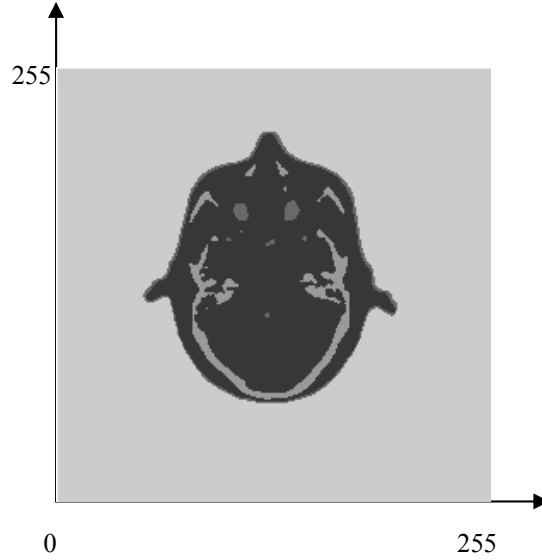
ŞEKİL p.11 ÇKA’nın eğitiminde iterasyona bağlı hata grafiği.



ŞEKİL p.12 ÇKA’nın eğitiminde iterasyona bağlı hata grafiği.

6.1) Sınıflayıcı olarak kullanılmak üzere yarışmacı bir ağ yapısı seçilmiştir. Benzetim programının kaynak kodu, “fu_ga_sınıflama.m” dosyasında saklanmıştır. Fantom uzaya ait öznelilik vektörleri, “fu_mous.txt” dosyasından okunur. Ağın topolojisi, 2-30-3 olarak seçilmiştir. Bu topoloji, çok sayıda denemeden sonra hatayı en aza indirgeyen yapı olarak tespit edilmiştir. Genetik havuzun yapısı Şekil p14’te gösterilmiştir.

Havuzdaki her dizi tek bir ağa karşılık gelmektedir. Havuz 16 dizi içermektedir. Ağ, 30 düğümden meydana gelmektedir. Ağın her düğümü 2 parametre içermektedir.



ŞEKİL p.13 CT görüntüsünün ÇKA ile sınıflandırılması.

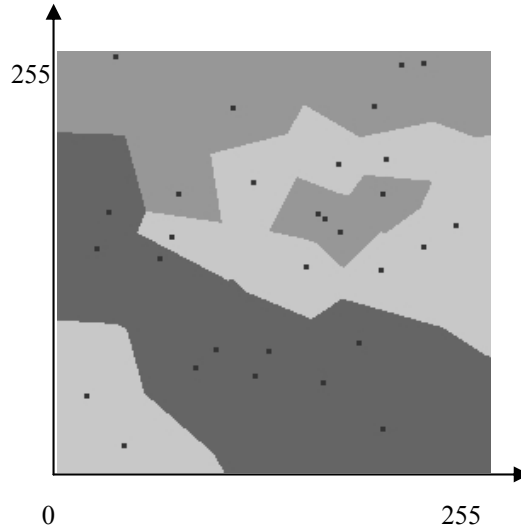
1.sınıf		2.sınıf		3.sınıf		3.sınıf	
$w_{1,1}^1$	$w_{1,2}^1$	$w_{2,1}^1$	$w_{2,2}^1$	$w_{3,1}^1$	$w_{3,2}^1$	$w_{30,1}^1$	$w_{30,2}^1$
.....
.....
15. dizi								
$w_{1,1}^{16}$	$w_{1,2}^{16}$	$w_{2,1}^{16}$	$w_{2,2}^{16}$	$w_{3,1}^{16}$	$w_{3,2}^{16}$	$w_{30,1}^{16}$	$w_{30,2}^{16}$

ŞEKİL p.14 Ağın parametrelerinin genetik havuzda konumlanması.

Aynı sınıfa ait düğümlerin çıkışı ortak bir sınıf düğümüne (OR işlemini gerçekleyecek şekilde) bağlanır. w_{jk}^i , i -inci dizinin j -inci düğümünün k -ıncı parametresini temsil etmektedir. Düğümler “fu_ga_dugum1.mat” dosyasında saklanmıştır. Ağın fantom uzayı sınıflama sonucu Şekil p15’te gösterilmektedir. Sonuç görüntü, “fu_ga_sınıflama.jpeg” dosyasında saklanmıştır.

- 6.2) Sınıflayıcı olarak kullanılmak üzere yarışmacı bir ağ yapısı seçilmiştir. Benzetim programının kaynak kodu, “fre_ga_sınıflama.m” dosyasında

saklanmıştır. EKG vurularına ait öznitelik vektörleri, “oz_fre_egitim.txt” dosyasından okunur. Ağın topolojisi, 32-20-4 olarak seçilmiştir. Havuzdaki her dizi bir ağa karşılık gelmektedir. Havuz 16 dizi içermektedir. Ağ, 20 düğümden meydana gelmektedir. Ağın her düğümü 32 parametre içermektedir.



ŞEKİL p.15 Fantom uzayın ağ ile sınıflandırılması.

Aynı sınıfa ait düğümlerin çıkışı ortak bir sınıf düğümüne (OR işlemini gerçekleyecek şekilde) bağlanır. w_{jk}^i , i -inci dizinin j -inci düğümünün k -ıncı parametresini temsil etmektedir. Bu topoloji, birçok denemeden sonra hatayı en aza indirgeyen yapı olarak tespit edilmiştir. Genetik havuzun yapısı Şekil p16’da gösterilmiştir. Düğümler “fre_ga_dugum1.mat” dosyasında saklanmıştır. Test kümesi “oz_fre_test.txt” dosyasından okunur.

6.3) Sınıflayıcı olarak kullanılmak üzere yarışmacı bir ağ yapısı seçilmiştir. Benzetim programının kaynak kodu, “ct_ga_sınıflama.m” dosyasında saklanmıştır. CT görüntüsüne ait öznitelik vektörleri, “oz_ct.txt” dosyasından okunur. Ağın topolojisi 4-20-4 olarak seçilmiştir. Havuzdaki her dizi, tek bir ağa karşılık gelmektedir. Havuz 16 dizi içermektedir. Ağ, 20 düğümden meydana gelmektedir. Ağın her düğümü 4 parametre içermektedir.

Aynı sınıfa ait düğümlerin çıkışı ortak bir sınıf düğümüne (Or işlemini gerçekleyecek şekilde) bağlanır. w_{jk}^i , i -inci dizinin j -inci düğümünün k -ıncı parametresini temsil etmektedir. Bu topoloji, birçok denemeden sonra

hatayı en aza indirgeyen yapı olarak tespit edilmiştir. Genetik havuzun yapısı Şekil p17’de gösterilmiştir. Düzgümler “ct_ga_dugum1.mat” dosyasında saklanmıştır. Ağın CT görüntüsündeki benekleri sınıflama sonucu Şekil p18’de gösterilmektedir. Sonuç görüntü, “ct_ga_sınıflama.jpeg” dosyasında saklanmıştır.

1.sınıf	...	3.sınıf	4.sınıf	...	4.sınıf
$w^1_{1,1} \dots w^1_{1,32}$...	$w^1_{3,1} \dots w^1_{3,32}$	$w^1_{4,1} \dots w^1_{4,32}$...	$w^1_{20,1} \dots w^1_{20,32}$
.....
.....
15. dizi					
$w^{16}_{1,1} \dots w^{16}_{1,32}$...	$w^{16}_{3,1} \dots w^{16}_{3,32}$	$w^{16}_{4,1} \dots w^{16}_{4,32}$...	$w^{16}_{20,1} \dots w^{16}_{20,32}$

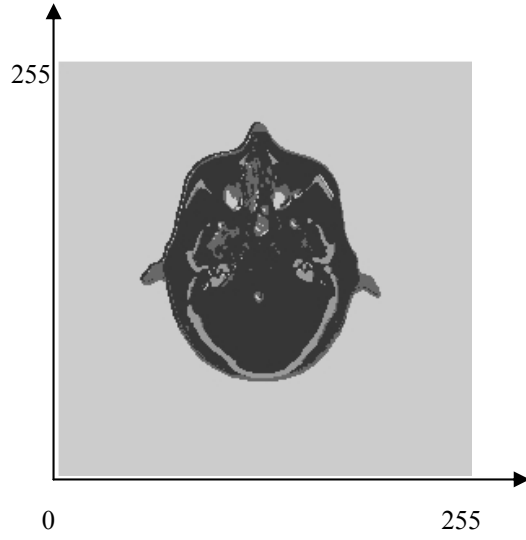
ŞEKİL p.16 Ağın parametrelerinin genetik havuzda konumlanması.

1.sınıf	...	3.sınıf	4.sınıf	...	4.sınıf
$w^1_{1,1} \dots w^1_{1,4}$...	$w^1_{3,1} \dots w^1_{3,4}$	$w^1_{4,1} \dots w^1_{4,4}$...	$w^1_{20,1} \dots w^1_{20,4}$
.....
.....
15. dizi					
$w^{16}_{1,1} \dots w^{16}_{1,4}$...	$w^{16}_{3,1} \dots w^{16}_{3,4}$	$w^{16}_{4,1} \dots w^{16}_{4,4}$...	$w^{16}_{20,1} \dots w^{16}_{20,4}$

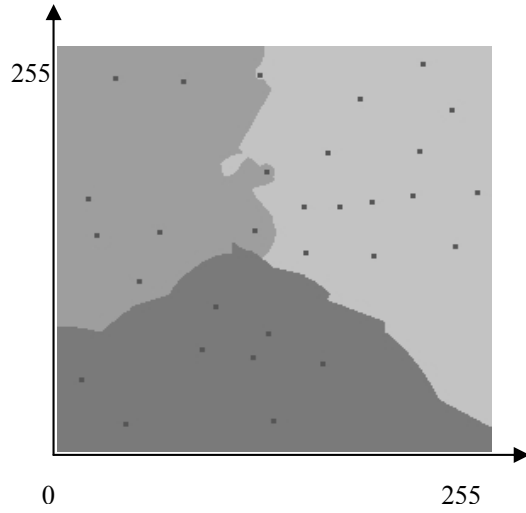
ŞEKİL p.17 Ağın parametrelerinin genetik havuzda konumlanması.

7.1)Sınıflayıcı olarak literatürde sıkça kullanılan bulanık K-en yakın komşu sınıflayıcı seçilmiştir. K değeri 3 olarak seçilmiştir. Benzetim programının kaynak kodu, “fu_fuzzy_sınıflama.m” dosyasında saklanmıştır. Fantom uzaya ait öznitelik vektörleri, “fu_mous.txt” dosyasından okunur. Ağ, 30 düğümden meydana gelmektedir. Ağın her düğümü 2 parametre içermektedir.

Aynı sınıfa ait düğümlerin çıkışı ortak bir sınıf düğümüne (OR işlemini gerçekleyecek şekilde) bağlanır. Ağın fantom uzayını sınıflama sonucu Şekil p19’da gösterilmektedir. Sonuç görüntü, “fu_fuzzy_sınıflama.jpeg” dosyasında saklanmıştır. Şekil p19’da üç sınıf için üç öbek merkezi seçilmiştir. Bu nedenle fantom öznitelik uzayı üç sınıfa ayrılmıştır. Aynı programda sınıf sayısı 9 seçilerek Şekil p.20’deki sonuç elde edilmiştir. Bu analiz, saçılma oluştuğunda sınıf sayısı kadar öbek merkezinin fantom uzaydaki sınıf dağılımını temsil edemediğini göstermektedir.



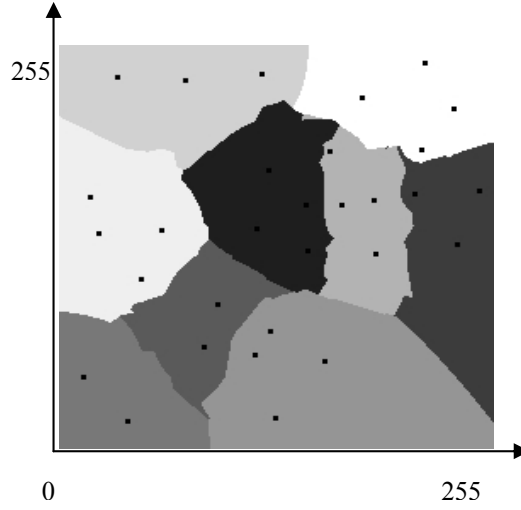
ŞEKİL p.18 CT görüntüsünün ağ ile sınıflandırılması.



ŞEKİL p.19 Fantom uzayın K-en yakın komşu sınıflayıcı ile sınıflandırılması.

7.2)Sınıflayıcı olarak literatürde sıkça kullanılan bulanık K-en yakın sınıflayıcı seçilmiştir. K değeri 3 olarak seçilmiştir. Benzetim programının kaynak kodu, “fre_fuzzy_sınıflama.m” dosyasında saklanmıştır. EKG vurularına ait öznitelik vektörleri, “oz_fre_egitim.txt” dosyasından okunur. Ağ, 20

düğümünden meydana gelmektedir. Ağın her düğümü 32 parametre içermektedir.



ŞEKİL p.20 Fantom uzayın K-en yakın komşu sınıflayıcı ile 9 sınıfa bölütlenmesi.

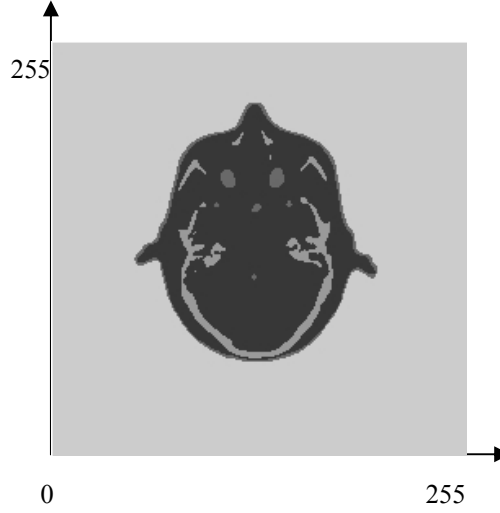
Aynı sınıfa ait düğümlerin çıkışı ortak bir sınıf düğümüne (OR işlemini gerçekleyecek şekilde) bağlanır. Düğümler “fre_fuzzy_dugum1.mat” dosyasında saklanmıştır. Test kümesi “oz_fre_test.txt” dosyasından okunur.

7.3)Sınıflayıcı olarak literatürde sıkça kullanılan bulanık K-en yakın sınıflayıcı seçilmiştir. K değeri 3 olarak seçilmiştir. Benzetim programının kaynak kodu, “ct_fuzzy_sınıflama.m” dosyasında saklanmıştır. CT görüntüsüne ait öznelilik vektörleri, “oz_ct.txt” dosyasından okunur. Ağ, 20 düğümünden meydana gelmektedir. Ağın her düğümü 4 parametre içermektedir.

Aynı sınıfa ait düğümlerin çıkışı ortak bir sınıf düğümüne (OR işlemini gerçekleyecek şekilde) bağlanır. Ağın CT görüntüsündeki benekleri sınıflama sonucu Şekil p21’de gösterilmektedir. Sonuç görüntü “ct_fuzzy_sınıflama.jpeg” dosyasında saklanmıştır.

8.1.a)“Aday Eleme” algoritmasının prolog kaynak kodu, “AdayYokEt.pl” dosyasında bulunmaktadır.

b)Bu ödevde kalp krizi risk faktörünün belirlenmesi konusunda bir çalışma yapılmış ve konu ile ilgili gerçeklerden oluşan bir veri dosyası



ŞEKİL p.21 CT görüntüsünün bulanık K-en yakın sınıflayıcı ile sınıflandırılması.

oluşturulmuştur. Veriler, sırasıyla sigara kullanımı, stres, cinsiyet, boy, kilo ve ten rengine sahip özelliklerden meydana gelmektedir. Veriler, prolog kaynak kodlu “HastalikRiski.pl” dosyasında bulunmaktadır. Aşağıda bu dosyanın içeriği gösterilmiştir.

% -----Örnek sayısı
n(7).

% -----Örnekler:

```
ornek(1, p, [icer ,var ,kadin ,orta ,sisman ,esmer ]).
ornek(2, n, [icmez ,yok ,erkek ,kisa ,orta ,sarisin ]).
ornek(3, n, [icmez ,yok ,kadin ,uzun ,zayif ,kumral ]).
ornek(4, p, [icer ,var ,erkek ,uzun ,sisman ,sarisin ]).
ornek(5, p, [icer ,var ,kadin ,kisa ,sisman ,kumral ]).
ornek(6, n, [icmez ,var ,kadin ,kisa ,orta ,esmer ]).
ornek(7, n, [icer ,yok ,erkek ,uzun ,orta ,sarisin ]).
```

% -----Her bir kavramın alabileceği değerler:

```
sigara(icer).
sigara(icmez).
sigara(x).
```

```
stress(var).
stress(yok).
stress(x).
```

```
cinsiyet(erkek).
```

```
cinsiyet(kadin).
cinsiyet(x).
```

```
boy(kisa).
boy(orta).
boy(uzun).
boy(x).
```

```
vucut(zayif).
vucut(sisman).
vucut(orta).
vucut(x).
```

```
ten(esmer).
ten(kumral).
ten(sarisin).
ten(x).
```

```
% -----Hipotez yapısı:
hipotez([Si,St,C,B,V,T]) :-
```

```
    sigara(Si),
    stress(St),
    cinsiyet(C),
    boy(B),
    vucut(V),
    ten(T).
```

c)“Soru_8_1.doc” dosyasında prolog kaynak kodunun kullanımı anlatılmaktadır. Prolog derleyicisine veri dosyası (HastalikRiski.pl) ve program kaynak kodu (AdayYokEt.pl) yüklenir. Derleyici konsolundan “basla.” komutu girilir. Program, G ve S kümelerinin aşama aşama içeriğini ve daha sonra uyarlama uzayının yapısını ekranda gösterir.

8.2.a)“Derinlik Öncelikli Araştırma” algoritmasının prolog kaynak kodu, “IlkDerinAra.pl” dosyasında bulunmaktadır. Aşağıda bu dosyanın içeriği gösterilmiştir.

```
% Derinlik Öncelikli Araştırma Algoritması
% (X1, Y1) den (X2, Y2) ye olacak olan hamle olası bir hamlemi?
```

```
uye(X,[X|_]).
uye(X,_[Liste]) :- uye(X,Liste).
```

```
basla:-
```

```

baslangic(S0),
bitis(S1),
Z = [S0],
ilerle(S0,S1,Z).

```

```

ilerle(S1,S1,Z) :-
    write('su sekilde bir yol bulundu:'),nl,write(Z),nl.

```

```

ilerle(Sa,S1,Z) :-
    sonraki(Sa,Sb),
    not(uye(Sb,Z)),
    append(Z,[Sb],Z1),
    % write(Sb),nl,      % ziyaret ettigi tum durumlari gormek için bu satiri acin
    ilerle(Sb,S1,Z1).

```

b) Problemi, karelerden oluşmuş bir tahta üzerinde bir konumdan diğer konuma önceden ziyaret edilmiş bir kareyi tekrar ziyaret etmeden satrançtaki at hamlelerini kullanarak ulaşma olarak tanımlayalım. Veriler, prolog kaynak kodlu “AtHareketi.pl” dosyasında bulunmaktadır. Aşağıda bu dosyanın içeriği gösterilmiştir.

```

%-----At hareketlerine ait veriler
tablo([a,b,c,d],[1,2,3,4]).

```

```

baslangic([a,1]).
bitis([d,4]).

```

```

durum([X,Y]):-
    tablo(TX,TY),
    member(X,TX),
    member(Y,TY).
%-----satranc tahtasindaki atin hamleleri
sonraki(S0,S1):-
    tablo(TX,TY),
    [X,Y]=S0,
    durum([X2,Y2]),
    (
        (nextto(X,X2,TX),nextto(Y,Z,TY),nextto(Z,Y2,TY));
        (nextto(X2,X,TX),nextto(Y,Z,TY),nextto(Z,Y2,TY));
        (nextto(X,Z,TX),nextto(Z,X2,TX),nextto(Y2,Y,TY));
        (nextto(X2,Z,TX),nextto(Z,X,TX),nextto(Y2,Y,TY));
        (nextto(X2,X,TX),nextto(Y2,Z,TY),nextto(Z,Y,TY));
        (nextto(X,X2,TX),nextto(Y2,Z,TY),nextto(Z,Y,TY));
        (nextto(X,Z,TX),nextto(Z,X2,TX),nextto(Y,Y2,TY));
        (nextto(X2,Z,TX),nextto(Z,X,TX),nextto(Y,Y2,TY))
    ),

```

$S1=[X2,Y2]$.

“Soru_8_2.doc” dosyasında prolog kaynak kodunun kullanımı anlatılmaktadır. Prolog derleyicisine veri dosyası (AtHareketi.pl) ve program kaynak kodu (IlkDeriniArastirma.pl) yüklenir. Derleyici konsolundan “basla.” komutu girilir. Program, başlangıç konumundan hedefe kadar olan ilerlemeyi sembollerle ekranda gösterir. Başlangıç konumu [a,1] olarak veri dosyasında tanımlıdır.

8.3.a)“FOIL” algoritmasının prolog kaynak kodu, “Foil.pl” dosyasında bulunmaktadır.

b)Bu ödevde, aile veri yapısı içinde kardeş kavramının tanımı çıkartılacaktır. Veriler, prolog kaynak kodlu “Aile.pl” ve “Kardes.pl” dosyalarında bulunmaktadır.

c)“Soru_8_3.doc” dosyasında prolog kaynak kodunun kullanımı anlatılmaktadır. Prolog derleyicisine veri dosyaları (Aile.pl ve Kardes.pl) ve program kaynak kodu (Foil.pl) yüklenir. Derleyici konsolundan “foil(kardes/2).” komutu girilir. Program, veri tabanını inceleyerek kardeş kavramını ifade eden bir koşul cümlesini ekranda gösterir.

8.4.a)“Id3” algoritmasının prolog kaynak kodu, “Id3.pl” dosyasında bulunmaktadır.

b)Bu ödevde, denizle ilgili veriler üzerinde bir veri tabanı oluşturulmuştur. Veriler, prolog kaynak kodlu “Varliklar.pl” dosyasında bulunmaktadır. Aşağıda bu dosyanın içeriği gösterilmiştir.

```
example(1, balik, [motorlumu=hayir, canlimi=evet, nerede_gider=denizde]).
example(2, gemi, [motorlumu=evet, canlimi=hayir, nerede_gider=denizde]).
example(3, at, [motorlumu=hayir, canlimi=evet, nerede_gider=karada ]).
example(4, kus, [motorlumu=hayir, canlimi=evet, nerede_gider=havada ]).
example(5, ucak, [motorlumu=evet, canlimi=hayir, nerede_gider=havada ]).
example(6, araba, [motorlumu=evet, canlimi=hayir, nerede_gider=karada ]).
example(7, bisiklet, [motorlumu=hayir, canlimi=hayir, nerede_gider=karada ]).
```

c)“Soru_8_4.doc” dosyasında prolog kaynak kodunun kullanımı anlatılmaktadır. Prolog derleyicisine veri dosyaları (varliklar.pl) ve program kaynak kodu (id3.pl) yüklenir. Derleyici konsolundan “id3.” komutu girilir. Konsoldan girilen “showtree.” komutu ile program, veri tabanını inceleyerek oluşturduğu ağacı ekranda gösterir.

8.5.a)“Q Öğrenmesi” algoritmasının prolog kaynak kodu, “QOgren.pl” dosyasında bulunmaktadır.

b) Bu ödevde, ödev 8.2’de ele alınan at hareketi incelenmektedir. Ödev 8.2’de bir başlangıç konumundan hedef konumuna atın ulaşması için bir yol optimizasyonu yapılmamaktadır. Ödevin bu şikkında at hareketi için en uygun yolun belirlenmesi üzerinde durulacaktır. Veriler, prolog kaynak kodlu “AtHareketi.pl” dosyasında bulunmaktadır.

Prolog konsol ekranına “qOgren.”komutunu girilir. Her oyun, rastgele bir başlangıç konumundan başlar ve hedef duruma ulaşana kadar sürer. Q Öğrenme algoritması her oyunda rastgele başlangıç durumlarından veri dosyasında belirtilen hedef (bitiş) durumuna ulaşana kadar hamleler üretecek ve her oyunun sonunda hamle değerlerini içeren Q tablosunu güncelleyecektir. QOgren.pl dosyasında belirtilen 100 sayıda oyun tekrarlandıktan sonra, hamle değerlerini (Q tablosunu) görmek için “hamleleriGoster.” komutu girilir.

c) Herhangi bir durumdan hedeflenen duruma ulaşmak için konsol ekranından örneğin “qOyna([a,1]).” komutu yazılır. Algoritma, [a,1] konumundan veri tabanında belirtilen sabit hedef konuma ([d,4]) olan en kısa yolu hamle değerleri ile (Şekil p.22) birlikte aşağıdaki şekilde belirtecektir:

[a, 1]durumundan[b, 3]durumuna deger,90
[b, 3]durumundan[d, 4]durumuna deger,100
hedefe_ulasildi
Yes.

	a	b	c	d
4				
3				
2				
1				

ŞEKİL p.22 En kısa mesafeden hedefe at’ın getirilmesi.

İNDEKS

A

Açık bölge, 158
Aday eleme, 247, 248, 298
Aday_Hipotezler, 260, 261, 262
Adım Adım Kapsama, 259, 263
Alt-küme, 8
Alt-örnekleme, 46
Ana bileşenlerin analizi, 99
Ana dalgacık, 43
AND işlemi, 157
Ani değişkenlik, 74, 76
AR modeli, 28, 29
ARMA modeli, 27, 34
ARMAX modeli, 27
Ayrım fonksiyonları, 128
Ayrım fonksiyonu, 129
Ayrım kriteri, 106
Ayrım kriterleri, 106
Ayrık dalgacık dönüşümü, 45
Ayrık Hopfield ağı, 176
Ayrık kosinüs dönüşümü, 51
Ayrılabilirlik, 22
Ayrıntı katsayıları, 48

B

Başarım indeksi, 135
Bayes kuralı, 126
Belirsizliğin ölçüsü, 89, 268
Bellek ihyacı, 14
Benzerlik oranı, 128
Benzerlik ölçüsü, 135
Benzerlikler şeması, 188
Benzerlikler yoluyla öğrenme, 287
Benzerlikleri Öğrenme, 284
Benzetim araçlarında, 14

Best-First Search, 258
Bilgi domeni, 284, 290
Bilgi kazancı, 269, 270, 272
Bilgi miktarı, 88
Bir Kural Öğren, 263
Boyut indirgeme, 85
Bölütleme işlemi, 58
Breadth-First Search, 257
Bulandırıcı, 228
Bulanık c-ortalama, 233
Bulanık Çıkarım, 226
Bulanık Denetim, 227
Bulanık K-NN sınıflayıcı, 237
Bulanık Kohonen öbekleme, 235
Bulanık küme, 223
Bulanık mantık, 224
Bulanık sınıflayıcılar, 223
Büyük ve öğren ağı, 154,166

Ç

Çağrışımsal bellek, 176
Çaprazlama pozisyonu, 185
Çaprazlama, 183
Çıkarım motoru, 228
Çıkış katmanı, 155
ÇKA, 154
Çok katmanlı ağ, 154
Çözünürlük, 46
Çözünürlük problemi, 39

D

Dalgacık analizi, 43
Dalgacık dönüşümü, 43
Dalgacık düzlemi, 49
Dalgacık fonksiyonu, 46
Darbe dizisi, 19

Daubechies dalgacıkları, 44
 Denetimli öğrenme, 153
 Denetimsiz öğrenme, 153, 169
 Depth-First Search, 256
 Derinlik Öncelikli Araştırma, 256
 Deterministik sistem, 27
 Dinamik programlama, 106, 107
 Diverjans, 6, 108
 Diverjans analizi, 14
 Dizilerin Kodlanması, 197
 Doğrunun kesişmesi, 157
 Doku kuvveti, 74, 77
 Dokusal Özellikler, 72
 Dönüşüm
 işlemi, 3, 17
 Durağan işaretlerin, 37
 Durağan olmayan, 39, 43
 Duru küme, 223
 Durulandırıcı, 229
 Düşüm, 154
 Düzenlenme uyumluluğu, 255

E

Eğitim algoritması, 11, 159, 164
 Eğitim işlemi, 148
 Eğitim kümesi, 148, 161
 Eğitim zamanı, 88
 En İyiyi Öncelikle Araştırılması, 258
 En yakın model sınıflayıcı, 239
 Enerji yoğunlaştırma, 52
 Enformasyon miktarı, 88
 Entropi, 88, 268, 269, 272
 Entropinin minimumlaştırılması, 90
 Eşikleme, 62
 Eş-Oluş matrisi, 77
 Etiketleme, 170
 Etiketleme işlemi, 174
 Euclidean mesafesi, 131

F

Faktör analizi, 99
 Fantom uzay, 5, 6
 Find-S, 246, 247
 Fisher dönüşüm vektörü, 95
 Fisher lineer ayrımı, 93

FOIL, 263, 264, 265, 266, 298
 Fourier Dönüşümü, 36

G

Genelden Özele Sıralama, 245
 Genelleme
 yeteneği, 3
 Genelleme özelliği, 88, 162
 Genelleştirilmiş δ -kuralı, 159
 Genelleştirilmiş karar fonksiyonları, 122
 Genel-özel ilişkisi, 246
 Genetik algoritmalar, 183
 Genetik havuz, 185, 189, 197
 Genişlik öncelikli araştırma, 257
 Gerçek-zaman, 14
 Geri izleme, 296
 GetÇKA, 208
 GetKoh, 214
 Görüntü, 17
 Görüntü çözünürlüğünün etkisi, 24
 Gözlenen işaret, 27
 Graf, 281, 282
 Gürültü modeli, 27

H

Hadamard dönüşümü, 55
 Hiper düzlemler, 157, 163
 Hiper küre, 162
 Hiper küre yarıçapları, 164
 Hiper küreleri, 163
 Hiper-düzlem, 9
 Hiperkuadratik yüzeyler, 130
 Hipotez, 245, 260, 261, 262
 Hopfield ağı, 176
 Horn koşulu, 253
 Hough dönüşümü, 64

I

ID3 algoritması, 270, 271, 272, 273

İ

İçerik adreslemeli bellek, 176, 178

İrilik, 74, 75
 İşaret çözünürlüğü, 46
 İşlem elemanı, 146

K

Kapalı bölge, 158
 Karar ağacı, 266, 267, 268, 269
 Karar Ağacını Öğrenme, 266
 Karar eşiği, 128
 Karar fonksiyonu, 8, 116, 117
 Karar kuralları, 115
 Karar sınırları, 120
 Karar verme
 mekanizmasını, 2
 Karar yüzeyleri, 115
 Kararlılık problemi, 176
 Karhunen-Loeve dönüşümü, 99
 Karmaşık dağılımlar, 7, 86
 Karmaşıklık, 74, 76
 Kategori, 1
 Katman, 146
 Kavramları Öğrenme, 244
 Kenar bulma işlemi, 59
 Kenar inceltme, 59
 Kernel fonksiyonları, 152
 Keşif yoluyla öğrenme, 278
 Kırpıcı, 148
 Kırpıcı fonksiyonu, 151
 Kısa-zaman Fourier, 41
 Kısıtlı 'Coulomb' enerji, 162
 Kısıtlı Coulomb enerji ağı, 154
 Klik, 281, 282, 283
 Kohonen Ağı, 170
 Komşuluk Gri-Ton Fark Matrisi, 74
 Kontrast, 74, 76
 Konvolüsyon, 22
 Kopyalama, 183
 K-ortalamalar algoritması, 138
 Koşul, 253
 Koşullu olasılık, 125
 Koşullu risk fonksiyonu, 128
 Kovaryans matrisi, 90
 Kural tabanı, 229
 Kurallar Kümesini Öğrenme, 258

L

Lagrange çarpanları, 101
 Learn One Rule, 260
 Lineer ayrılabilen sınıflar, 121
 Lineer karar fonksiyonları, 115
 Lineer karar fonksiyonu, 117
 Lineer sınıflayıcı, 11
 Lineerlik, 21
 Literal, 253, 254, 259, 262, 264

M

MA modeli, 28
 MA modeli, 32
 Mahalanobis mesafesi, 131
 Maksimal kısmi eşleşme, 286
 Maksimal klik, 282
 maksimum-mesafe algoritması, 137
 Maksimum klik, 282
 Maliyetin minimizasyonu, 195
 Mesafe
 ölçüleri, 8
 Mikro-denetleyici, 14
 Minimal özellikleri, 248, 251
 MLP, 154
 Model parametreleri, 24
 Moment, 73
 Moment Özellikleri, 67
 Mutasyon, 183

N

Nicemlemenin etkisi, 24
 Normalizasyon
 işlemi, 2
 Nyquist kriteri, 37

O

Optimizasyon problemi, 178
 OR işlemi, 157

Ö

Öbek merkezi, 137, 138
 Öbekleşme kriteri, 135
 Öğrenme algoritması, 148

Öğrenme işlemi, 148
 Ölçek değeri, 44
 Ölçek fonksiyonu, 46
 Ölçek parametresi, 44
 Ölçekleme, 43
 Ön-olasılık bilgisi, 126
 Örnekleme frekansı, 37
 Örüntü
 uzayı, 1
 Örüntü onarma, 178
 Örüntüler, 1
 Öteleme ile değişmezlik, 21
 Öteleme terimi, 43
 Özdeğer, 90, 91
 Özdeğerler, 101
 Öz-düzenlemeli harita, 173
 Özdeşlemler, 103
 Özelden genelle, 260
 Öznitelik
 çıkartma, 2
 Öznitelik Uzayını Bölmeleme, 155,
 163, 166, 170, 173
 Özniteliklerin
 çıkartılması, 4
 Özvektör, 90
 Özvektörler, 101

P

Parametre vektörü, 117
 Penaltı yöntemi, 200
 Perceptron, 147
 Pozitif ve negatif örnekler, 244, 245
 Prolog, 258, 259, 291, 292, 293, 294,
 295, 296, 298, 303
 Prototip vektörler, 141

Q

Q Fonksiyonu, 275
 Q Öğrenmesi, 277, 278

R

Rayleigh Oranı, 95
 RCE, 162

S

Sabit çözünürlük, 44
 Saçılım matrisi, 106, 135
 Saklı katman, 155
 Salt eylemci, 153
 Sequential Covering, 259
 Servo denetleyici, 230
 Sınıf
 dağılımı, 4
 Sınıf sınırlarını, 8
 Sınıf üyelikleri, 238
 Sınıf-içi, 6
 Sınıflama
 işlemi, 2
 Sınıflama başarımı, 116
 Sınıflama performansı, 11
 Sınıflama zamanı, 14
 Sınıflar-arası, 6
 Sınıflayıcı
 yapısını, 11
 Sigmoid fonksiyonu, 151
 Spesific to general, 260
 Sürekli dalgacık
 dönüşümü, 44

Ş

Şekilsel Özellikler, 70
 Şema sırası, 192
 Şema tanımlayıcı uzunluğu, 192
 Şema teoremi, 189, 192

T

Takviyeli öğrenme, 273, 274
 Tek boyutlu, 17
 Tekil değer, 102
 Temel eksen açısı, 69
 Terim, 253, 295
 Topluluk büyüklüğü, 187
 Transfer fonksiyonu, 146

U

Uniter matris, 102
 Uyarlama uzayı, 248

Uyumluluk deęerine, 183
Uyumluluk fonksiyonu, 183

Ü

Üst-örnekleme, 46
Üyelik deęerleri, 234
Üyelik fonksiyonları, 225
Üyelik fonksiyonu, 224

V

Vektör boyutunun
küçültülmesi, 5
Veri tabanı, 228

Y

Yaklaşıklik katsayıları, 48
Yapay sinir aęları, 145
Yapay zeka, 243, 253
Yapısal bloklar, 195
Yarar maksimizasyonu, 195
Yerel bellek, 146
Yönlü graf, 146
Yüklemler, 253, 264

Z

Zincir Kodlama, 66

Yapay zeka destekli örüntü tanıma konusu, teknolojik gelişmelerle paralel geniş bir uygulama alanı bulmaktadır. Bu konudaki çalışmalar; tıpta vücut üzerinden alınan biyolojik işaretlerin analizinde, cep telefonlarında veya bilgisayarlarda ses tanıma düzeneklerinde, yüz veya parmak izi tanıyan güvenlik sistemlerinde, görerek objeleri tanıyan ve işlem yapan robot veya otomatlarda, uzaktan algılanan görüntülerin analizinde, savunma amaçlı oluşturulan düzeneklerde sıkça gözlenmektedir.

Bu kitapta yapay sinir ağları, genetik algoritmalar, bulanık mantık ve makine öğrenmesi gibi yapay zeka yaklaşımlarının örüntü tanıma konusunda kullanımları ele alınmaktadır. Kitap içindeki her bölümde örüntü tanıma işleminde oluşturulacak karar mekanizmasına ait gerekli adımlar ayrıntılı olarak incelenmektedir.

Kitabın birinci bölümden sekizinci bölüme kadar bölüm sonlarında üç farklı örnek üzerinde MATLAB ortamında çözülmesi istenen problemler bulunmaktadır: (i) İki-boyutlu fantom uzay, (ii) EKG verisi, (iii) bilgisayarlı tomografi. Sekizinci bölümün sonunda ise PROLOG programlama ile gerçekleştirilmesi istenen problemler bulunmaktadır. 'Cevap Anahtarı' bölümünde problemlerin çözümleri ve oluşturulan çıktıların isimleri verilmektedir. Tüm programlar ve üretilen sonuçlar, kitapla birlikte verilen CD içinde bulunmaktadır.



Tamer Ölmez 1964 yılında İstanbul'da doğdu. Orta öğrenimini 1981 yılında Çavuşoğlu koleji'nde tamamladı. Lisans öğrenimini 1985 yılında İ.T.Ü. Elektronik ve Haberleşme Mühendisliğinde, yüksek lisans öğrenimini 1988 yılında İ.T.Ü. Fen Bilimleri Enstitüsü Bilgisayar ve Kontrol Mühendisliğinde tamamladı. 1988-1991 yılları arasında TELETAS'ın araştırma ve geliştirme bölümünde işaret işleme konusunda proje mühendisi olarak çalışmıştır. 1991 yılında altı ay TÜBİTAK'ta uzman yardımcısı olarak çalışmıştır. Aynı yıl İ.T.Ü. Elektronik Anabilim Dalına araştırma görevlisi olarak göreve başlamıştır. 1995 yılında İ.T.Ü. Fen Bilimleri Enstitüsü, Biyomedikal Mühendisliği Programında doktora çalışmasını tamamlamıştır. 1997 yılında Elektronik dalında Doçent ünvanı almıştır. 2003 yılından beri aynı üniversitenin Elektronik Anabilim Dalında Profesör olarak görev yapmaktadır. Biyolojik işaretlerin işlenmesi, görüntü işleme, tıbbi bilişim, yapay sinir ağları, genetik algoritmalar, örüntü tanıma, makine öğrenmesi, bilgisayarla görü, mikro-işlemci ve sayısal

işaret işlemci destekli sistem tasarımı, veri toplama sistemleri tasarımı, sanal gerçeklik, çoğul ortam uygulamaları konularında çalışmalarını sürdürmektedir.



Zümray Dokur Ölmez, lisans öğrenimini 1992 yılında İTÜ Elektronik ve Haberleşme Mühendisliği Bölümünde, yüksek lisans öğrenimini 1995 yılında İ.T.Ü. Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, Biyomedikal Mühendisliği Programında tamamladı. 1992 yılında İ.T.Ü. Elektronik Anabilim Dalında araştırma görevlisi olarak göreve başlamıştır. Doktora çalışmasını 2000 yılında İ.T.Ü. Elektronik ve Haberleşme Mühendisliği Anabilim Dalı, Biyomedikal Mühendisliği Programında tamamlamıştır. 2003 yılında Elektrik-Elektronik Mühendisliği Bilim Alanında Doçent ünvan ve yetkisi almıştır. 2009 yılından beri Elektronik Anabilim Dalında Profesör olarak görev yapmaktadır. Medikal enstrümantasyon, biyolojik işaretlerin analizi ve tanınması, medikal görüntülerin analizi, bölütlenmesi, yapay sinir ağ modellerinin geliştirilmesi, genetik eğitim, bulanık mantık ve bulanık sınıflayıcılar konularında çalışmalarını sürdürmektedir.

ISBN: