

TURKCEDEKI SESLİ HARFLERİN ÇOKLU AYIRMA  
ANALİZİNE GÖRE SINIFLANDIRILMASI

Umit KUNKÇU

Yüksek Lisans Tezi

Elektrik Elektronik Mühendisliği

Anabilim Dalı

1992

Anadolu Üniversitesi  
Merkez Kütüphane

TURKÇEDEKİ SESLİ HARFLERİN ÇOKLU AYIRMA ANALİZİNE GÖRE  
SINIFLANDIRILMASI

Ümit KÜNKÜ

Anadolu Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Yönetmeliği Uyarınca  
Elektrik Elektronik Mühendisliği Anabilim Dalı  
Elektronik Bilim Dalında  
YÜKSEK LİSANS TEZİ  
Olarak Hazırlanmıştır.

Danışman : Prof. Dr. Atila BARKANA

EYLÜL - 1992

Ümit KÜNKÜ'nün YÜKSEK LİSANS tezi olarak hazırladığı " Türkçe-  
deki Sesli Harflerin Çoklu Ayırma Analizine Göre Sınıflandırılması "   
başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili   
maddeleri uyarınca değerlendirilerek kabul edilmiştir.

17.9.1992

Üye: Prof. Dr. Atila BARKANA

Üye: Prof. Dr. Atalay BARKANA

Üye: Y. Doç. Dr. Osman PARLAKTUNA

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 17 EYLÜL 1992 gün  
ve 323 - 6 sayılı kararıyla onaylanmıştır.

Prof. Dr. Rüstem KAYA  
Enstitü Müdürü

## İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET .....	iv
SUMMARY .....	v
TEŞEKKÜR .....	vi
ÇİZELGELER DİZİNİ .....	vii
1. GİRİŞ .....	1
2. ÇOKLU AYIRMA ANALİZİ TEKNİĞİ .....	2
2.1 Ayırma Analizinin Özellikleri ve Temel Kavramlar .....	2
2.2 Ayırma Fonksiyonlarının ve Katsayılarının Elde Edilmesi .....	3
2.3 Düzeltilmiş Katsayıların Elde Edilmesi .....	9
2.3.1 Standart olmayan katsayıların elde edilmesi .....	9
2.3.2 Katsayıların standartlaştırılması .....	10
2.4 Ayırma Fonksiyonlarının Sayısı ve Önem Sıraları .....	10
3. SINIFLANDIRMA ANALİZİ .....	12
3.1 Sınıflandırma Kuralları .....	12
3.1.1 İki grup için geliştirilen sınıflandırma metodu ...	12
3.1.2 İkiden fazla grup olması halinde sınıflandırma metodu .....	14
3.2 Uzaklık Fonksiyonlarına Dayalı Sınıflandırma .....	16
3.2.1 İki grup için sınıflandırma kuralları .....	16
3.2.2 İkiden fazla grup için sınıflandırma kuralları ....	17
3.3 Kanonik Vektör Yaklaşımına Göre Sınıflandırma .....	18
4. ÇOKLU AYIRMA ANALİZİNİN PARAMETRİK MODELLERİN ÖNEM DERECELERİNİ BULMADA KULLANILMASI .....	20
4.1 Parametrik Modellerin Nisbi Önemlerinin Elde Edilmesi ...	21
4.2 Sıfır Geçiş Sayıları, Enerji Seviyeleri ve Maksimum Genlikler Arasındaki Önem Derecelerinin İncelenmesi .....	21
4.3 Lpc, Lar ve Cepstral Parametreleri Arasındaki Önem Derecelerinin İncelenmesi .....	22

## İÇİNDEKİLER (devam)

### Sayfa

4.4 Otokorelasyon, Lpc Parametreleri ve Formant Frekansları Arasındaki Önem Derecelerinin İncelenmesi .....	23
5. ÇOKLU AYIRMA ANALİZİNİN ÜNLÜLERİ SINIFLANDIRMADA KULLANILMASI .....	33
6. SONUÇLAR .....	35
7. KAYNAKLAR DİZİNİ .....	37

### EKLER

1. Çoklu ayırma analizi programı
2. Sesli harflerin parametrelerini hesaplayan program.
3. Salih EREN ve Ümit KÜNKÜ'den elde edilen LPC - Cepstral - LAR parametreleri için en önemli ayırma fonksiyonları
4. Salih EREN ve Ümit KÜNKÜ'den elde edilen LPC parametreleri için en önemli ayırma fonksiyonları
5. Salih EREN ve Ümit KÜNKÜ'den elde edilen Cepstral parametreleri için en önemli ayırma fonksiyonları
6. Salih EREN ve Ümit KÜNKÜ'den elde edilen LAR parametreleri için en önemli ayırma fonksiyonları
7. Salih EREN ve Ümit KÜNKÜ'den elde edilen Otokorelasyon - LPC - Formant parametreleri için en önemli ayırma fonksiyonları
8. Salih EREN ve Ümit KÜNKÜ'den elde edilen Otokorelasyon parametreleri için en önemli ayırma fonksiyonları
9. Ümit KÜNKÜ'den elde edilen LAR parametrelerine göre sekiz ünlü için sınıflandırma
10. Ümit KÜNKÜ'den elde edilen LPC parametrelerine göre sekiz ünlü için sınıflandırma
11. Ümit KÜNKÜ'den elde edilen Otokorelasyon parametrelerine göre sekiz ünlü için sınıflandırma

**ÖZET**

Bu tez çalışmasında, Türkçedeki ünlülerin çoklu ayırma analizi ile tanınması gerçekleştirilmiştir. Öncelikle 5 bayan ve 9 erkek konuşmacıdan alınan ses örneklerinin çeşitli parametreleri elde edilmiştir. Daha sonra elde edilen parametreler çoklu ayırma analizinde kullanılarak grupları ayırmada en önemli parametreler bulunmuştur. Bu parametreler kullanılarak hangi gruba ait olduğu bilinmeyen bir gözlem vektörü çoklu ayırma analizi yardımıyla, Türkçedeki ünlü gruplarından birine sınıflandırılmaktadır.

Anahtar Sözcükler :

Çoklu ayırma analizi  
Ses parametreleri  
Türkçe ünlüler

## SUMMARY

In this thesis, Turkish vowels are recognized by using multiple discriminant analysis. First of all, using voice samples taken from 5 females and 9 males, various parameters of vowels in Turkish Alphabet were obtained. Then, using these parameters in multiple discriminant analysis, the parameters which discriminate groups better than the others are obtained. By using these parameters and multiple discriminant analysis, we can classify an unknown vectors of vowels into Turkish vowel groups.

### Key Words :

Multiple dicriminant analysis

Voice parameters

Turkish vowels

**TESEKKUR**

Çalıřmalarımda bana yardımcı olan danıřmanım Prof. Dr. Atila BARKANA'ya, Prof. Dr. Atalay BARKANA'ya . Arř. Gör. Salih EREN'e diđer tüm mesai arkadaşlarıma teřekkür ederim.



## ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
4.1 Sıfır Geçiş Sayıları, Enerji Seviyeleri ve Maksimum Genlikler için en önemli ayırma fonksiyonu katsayıları .....	25
4.2 LPC, Cepstral ve LAR parametreleri için en önemli ayırma fonksiyonu katsayıları .....	26
4.3 LPC parametreleri için en önemli ayırma fonksiyonu katsayıları .....	27
4.4 Cepstral parametreleri için en önemli ayırma fonksiyonu katsayıları .....	28
4.5 LAR parametreleri için en önemli ayırma fonksiyonu katsayıları .....	29
4.6 Otokorelasyon, LPC ve Formant Frekansları için en önemli ayırma fonksiyonu katsayıları .....	30
4.7 Otokorelasyonlar ve LPC'ler için en önemli ayırma fonksiyonu katsayıları .....	31
4.8 Otokorelasyonlar için en önemli ayırma fonksiyonu katsayıları .....	32

## 1. GİRİŞ

Günümüzde konuşmayı tanıma üzerinde yapılan çalışmalar bilgisayar teknolojisinden yararlanma olanaklarının artmasıyla ayrı bir hız kazanmıştır. Özellikle konuşmayı tanıma çalışmalarında hızlı ve kapsamlı bilgisayarlarla çalışmak hem zaman kaybını önlemekte hem de uzun analiz süreleri kısaltmaktadır.

Ses bilgisinden elde edilen parametrik modeller araştırmacılar tarafından sıklıkla kullanılmaktadır. Ancak araştırmalarımız da kullandığımız parametrik modeller , özellikle ses bilgisinin sınıflandırılması durumunda yeterli sonuç vermeyebilir.

Kullanılan parametrik modellerin sınıflandırma olayında, grupları ayırmada ne derece önemli olduğu çoklu ayırma analizi kullanılarak saptanabilir.

Bu tez çalışması dört bölümden oluşmaktadır. İlk iki bölümde ayırma ve sınıflandırma analizi ile ilgili teoremler verilmektedir. Diğer bölümlerde de ayırma analizinin parametrelerin önem derecelerinin bulunmasında kullanımı ve sınıflandırma analizi konuları işlenmiştir.

## 2. ÇOKLU AYIRMA ANALİZİ TEKNİĞİ

Çoklu Ayırma Analizi ( Discriminant Analysis ), iki grup için uygulanabilen ayırma analizi tekniğinin genelleştirilmiş şekli olup, birden fazla değişkene göre üç veya daha çok grup arasındaki farklılıkları araştırarak çok değişkenli istatistik bir tekniktir. Ayırma analizinde en önemli konu gruplar arasındaki farklılıkların belirlenmesinde kullanılacak olan ayırma fonksiyonlarının elde edilerek yorumlanmasıdır. Ayırma analizi elemanların gruplara sınıflandırılması işlemine temel oluşturduğundan, istatistik bilimi açısından elde edilen ayırma fonksiyonlarına " karar fonksiyonu " adı da verilmektedir. İzleyen kısımlarda ayırma analizinin uygulamayla ilgili kuramsal esasları verilecektir.

### 2.1 Ayırma Analizinin Özellikleri ve Temel Kavramlar

Ayırma analizinde elemanlar analizinin temel birimleri olup iki ya da daha fazla karşılıklı ayırık grupların elemanları olmalıdır. Diğer bir deyişle, elemanlar analiz öncesi gruplara atanmalı ve gruplar, her eleman sadece ve sadece bir gruba ait olacak şekilde oluşturulmalıdır.

Bilindiği gibi ayırma analizine konu olan elemanların çeşitli ortak özellikleri vardır. Grupları ayırmada kullanılan bu özelliklere "ayırıcı değişkenler" denir. Ayırma analizinde aritmetik ortalama, varyans v.b. ölçülerin hesaplanabilmesi için ayırıcı değişkenlerin eşit aralıklı ölçekte ölçülmesi gereklidir. Genelde ayırıcı değişkenlerin sayısı ile ilgili bir sınırlama yoktur. Ancak bu sayı en çok tüm gruplardaki eleman sayısından iki eksik olmalıdır (Overall and Klett, 1972). Bununla birlikte ayırıcı değişkenler için bazı sınırlamalar sözkonusudur. Bunlardan birincisi, herhangi bir ayırıcı değişken diğer ayırıcı değişkenlerin doğrusal bileşimi olmamalıdır. Aynı şekilde aralarında tam korelasyon olan iki değişken, aynı şeyi açıklayacağından birlikte kullanılmaları sakıncalıdır.

Çoğu uygulamalarda gerekli olan diğer bir varsayım da varyans-kovaryans matrislerinin her grup için eşit veya yaklaşık eşit ol-

duğudur. Grupların varyans-kovaryans matrislerinin eşit varsayılması bu fonksiyonları hesaplamada kullanılan formüllerin elde edilmesini kolaylaştırmaktadır.

Ayırma analizindeki en önemli varsayımlardan birisi de, her grubun çok değişkenli normal dağılım gösteren bir yığından çekilmiş olduğunun kabul edilmesidir.

Çoklu ayırma analizi için yapılan varsayımlar ve kullanılan gösterimler aşağıdaki gibi özetlenebilir (Çakmak, 1986).

a) Kullanılacak gösterimler:

- g = Grupların sayısı,
- p = Ayırıcı değişkenlerin sayısı,
- $n_k$  = k'inci gruptaki elemanların sayısı,
- N = Tüm gruptaki elemanların sayısı.

b) Varsayımlar:

- (1) İki veya daha fazla grup :  $g \geq 2$
- (2) Her grup için enaz iki eleman :  $n_k \geq 2$
- (3) Ayırıcı değişkenlerin sayısı :  $0 < p < (N-2)$
- (4) Ayırıcı değişkenler eşit aralıkta ölçülmüştür.
- (5) Hiçbir ayırıcı değişken diğer ayırıcı değişkenlerin doğrusal bileşimi olamaz.
- (6) Her grup için varyans - kovaryans matrisleri eşit veya yaklaşık eşittir.

## 2.2 Ayırma Fonksiyonlarının ve Katsayılarının Elde Edilmesi

Ayırma analizinin temel problemlerinden birisi grupları ayırma bakımından en güçlü ayırma değişkenlerinin neler olduğudur. Ayırma fonksiyonları genel olarak, P tane ayırıcı değişkenin doğrusal bileşimi olup, aşağıdaki gibi gösterilebilir.

$$f_{km} = V_1X_{1km} + V_2X_{2km} + \dots + V_pX_{pkm} \quad (2.1)$$

$f_{km}$  : Ayırma fonksiyonunun k'inci gruptaki m'inci eleman için değeri;

$X_{ijkm}$  : i'inci ayırıcı değişkenin, k'inci gruptaki m'inci eleman için değeri ve

$V_i$  : Ayırma fonksiyonun katsayılarıdır.

Genel şekli yukarıda gösterilen bu fonksiyonda  $V_i$  katsayıları gruplar arası farklılıkları maksimum yapacak şekilde elde edilirler. Bu işlemler önce birinci ayırma fonksiyonu, daha sonra diğer ayırma fonksiyonları için yapılır. Analizde elde edebileceğimiz ayırma fonksiyonlarının sayısı grup sayısından bir eksik veya ayırıcı değişken sayısına eşit olacaktır. Grupların sayısı  $g$ , ayırıcı değişkenlerin sayısı  $p$  ve ayırma fonksiyonlarının sayısı  $r$  ile gösterilirse, bunun matematiksel ifadesi;

$$\max\{ r \} = \min\{ g-1, p \} \quad (2.2)$$

olur (Lachenbruch, 1975).

Bireyler arasındaki farklılıkların derecesini ölçmek için bazı istatistik yöntemler geliştirilmiştir. Ayırıcı değişkenler arasındaki iç ilişkileri belirlemede yalnızca grup ortalamaları ve standart sapmaları incelemek yeterli olmamaktadır. Bu durumda öncelikle,

- a) grup içindeki elemanların, o grubun aritmetik ortalamasından sapmaları,
- b) elemanların genel ortalamadan sapmaları ve
- c) grup ortalamalarının genel ortalamadan sapmaları

incelenmelidir.

Ayırıcı değişken sayısını başlangıçta bir olarak aldığımızda elemanların ait oldukları grubun ortalamasından sapmaları  $(X_{ijkm} - \bar{X}_k)$ , elemanların genel ortalamadan sapmaları  $(X_{ijkm} - \bar{X})$  ve grup ortalamalarının genel ortalamadan sapmaları  $(\bar{X}_k - \bar{X})$  olarak gösterildiğinde, aralarında aşağıdaki gibi bir ilişki olduğu görülür.

$$(X_{ijkm} - \bar{X}) = (\bar{X}_k - \bar{X}) + (X_{ijkm} - \bar{X}_k) \quad (2.3)$$

Eşitliğin her iki tarafında karesini alırsak;

$$[(X_{ijkm} - \bar{X})]^2 = [(\bar{X}_k - \bar{X}) + (X_{ijkm} - \bar{X}_k)]^2$$

$$(X_{ikm} - \bar{X})^2 = (\bar{X}_k - \bar{X})^2 + (X_{ikm} - \bar{X}_k)^2 + 2(\bar{X}_k - \bar{X})(X_{ikm} - \bar{X}_k)$$

yukarıdaki terimlerin tümünü  $m=1,2,\dots,n_k$  üzerinden toplarsak,

$$\begin{aligned} \sum_{m=1}^{n_k} (X_{ikm} - \bar{X})^2 &= \sum_{m=1}^{n_k} (\bar{X}_k - \bar{X})^2 + \sum_{m=1}^{n_k} (X_{ikm} - \bar{X}_k)^2 \\ &+ 2(\bar{X}_k - \bar{X}) \sum_{m=1}^{n_k} (X_{ikm} - \bar{X}_k) \end{aligned}$$

burada  $(X_{ikm} - \bar{X}_k)$  yani, elemanların ait oldukları grup ortalamalarından sapmaları toplamı, her zaman sıfır olacaktır. Böylece,

$$\sum_{m=1}^{n_k} (X_{ikm} - \bar{X})^2 = n_k (\bar{X}_k - \bar{X})^2 + \sum_{m=1}^{n_k} (X_{ikm} - \bar{X}_k)^2 + 0 \quad (2.4)$$

elde edilir (Anderson and Sclove, 1978). Yukarıdaki ifade genel olarak şöyle yazılabilir:

$$\begin{array}{ccc} \text{Toplam kareler} & = & \text{Gruplar arası kareler} + \text{Grup içi kareler} \\ \text{toplama} & & \text{toplama} \quad \quad \quad \text{toplama} \end{array}$$

veya

$$T = B + W \quad (2.5)$$

Ayrıcı değişken sayısı birden fazla olduğunda, yukarıdaki denklemdeki her bir terim, boyutu ayrıcı değişken sayısına eşit olan simetrik karematrislere dönüşür. Bu durumda T matrisinin elemanları aşağıdaki formüle göre hesaplanabilir (Çakmak, 1986).

$$t_{ij} = \sum_{k=1}^g \sum_{m=1}^{n_k} (X_{ikm} - \bar{X}_i)(X_{jkm} - \bar{X}_j) \quad (2.6)$$

$i=1,2,\dots,p$   
 $j=1,2,\dots,p$

Burada:

- $g$  = grupların sayısı,
- $n_k$  = k'inci gruptaki elemanların sayısı,

$X_{ikm}$  = i'inci ayırıcı değişkenin, k'inci gruptaki m'inci birey için değeri,

$\bar{X}_i$  = i'inci ayırıcı değişkenin tüm bireyler için ortalama değeri olan genel ortalama olmaktadır.

Yukarıdaki formülde  $i = j$  olduğunda T matrisinin ana köşegeni üzerindeki elemanlar elde edilir. Bu elemanlar, ilgili değişkenlerin genel ortalamadan sapmalarının karelerini göstermektedir.

Eğer, T matrisinin her elemanı  $(N - 1)$ 'e bölünürse toplam kovaryans matrisi elde edilir. Bu matrisin ana köşegeni üzerindeki elemanlar varyansları, diğer elemanlar ise kovaryansları göstermektedir. Söz konusu kovaryanslar iki değişkenin birlikte ne kadar değiştiğinin bir ölçüsü olmaktadır.

Eğer gruplar birbirinden farklı ise, diğer bir deyişle grup ortalamaları çakışık değilse, gruplar içindeki varyans - kovaryans, toplam varyans - kovaryanstan daha küçük olacaktır. Bu, gruplar içi kareler toplamı olarak adlandırılan W matrisi ile ölçülür. W matrisinde sapmalar, T matrisinde olduğu gibi, genel ortalamadan değil, elemanın ait olduğu grup ortalamasından ölçülmektedir. Bu matrisin elemanları aşağıdaki formüle göre hesaplanabilir (Lachenbruch, 1975).

$$w_{ij} = \sum_{k=1}^g \sum_{m=1}^{n_k} (X_{ikm} - \bar{X}_{ik})(X_{jkm} - \bar{X}_{jk}) \quad \begin{matrix} i=1,2,\dots,p \\ j=1,2,\dots,p \end{matrix} \quad (2.7)$$

Burada;

$g$  = grupların sayısı,

$n_k$  = k'inci gruptaki elemanların sayısı,

$\bar{X}_{ik}$  = i'inci değişkenin k'inci gruptaki elemanlar için ortalama değeri,

$X_{ikm}$  = i'inci değişkenin k'inci gruptaki m'inci eleman için değeri olmaktadır.

W matrisinin her elemanı  $(N - g)$ 'ye bölünürse grup kovaryans matrislerinin ağırlıklı bir ortalaması olan gruplar içi kovaryans matrisi elde edilir.

$$W_C = \frac{W}{N - g} \quad (2.8)$$

T ve W matrislerinde kullanılan grup ortalamaları ve genel grup ortalaması aşağıdaki formüle göre hesaplanmaktadır.

$$\bar{X}_{ik} = \frac{1}{n_k} \sum_{m=1}^{n_k} X_{ikm} \quad \begin{array}{l} i=1,2,\dots,p \\ k=1,2,\dots,g \end{array} \quad (2.9)$$

ve

$$\bar{X}_i = \frac{1}{N} \sum_{k=1}^g \sum_{m=1}^{n_k} X_{ikm} \quad i=1,2,\dots,p \quad (2.10)$$

Burada;

$\bar{X}_{ik}$  : i'inci ayırıcı değişkenin k'inci gruptaki bireyler için değeri ve

$\bar{X}_i$  : i'inci değişkenin tüm bireyler için ortalama değeri.

Grup merkezlerinin çakışık olmadığı durumlarda W matrisinin elemanlarının T matrisine karşı gelen elemanlardan daha küçük olduğu yukarıda açıklanmıştır. Bu iki matris arasındaki fark ( $B = T - W$ ) " gruplar arası kareler toplamı " adı verilen B matrisi ile ölçülür. B matrisinin her elemanı ( $g - 1$ )'e bölünürse gruplar arası kovaryans matrisi elde edilir.

$$B_C = \frac{B}{g - 1} \quad (2.11)$$

Fisher'e (1936) göre,  $W_C$  ve  $B_C$  matrisleri gruplar içi ve gruplar arası ilişkilere ait temel bilgileri içerdiklerinden gruplar arası ayırımın yapılmasında kullanılan ölçüt veya fonksiyonda bu matrislerin varyans miktarlarının birbirlerine olan oransal durumlarını dikkate almak gerekir (Lachenbruch, 1975).

Bu durumda analizde kullanılacak ölçüt,

$$\phi = \frac{\underline{V}' B_C \underline{V}}{\underline{V}' W_C \underline{V}} \quad (2.12)$$



şeklinde ifade edilen fonksiyonu maksimum yapan değerdir. Başka bir deyişle, gruplar arası varyansın gruplar içi varyansa oranının maksimum yapılmasıdır. Burada,

$$\underline{V}' = (V_1, V_2, \dots, V_p) \quad (2.13)$$

şeklinde tanımlanan katsayılar vektörüdür ve Lagrange yöntemine göre,  $\underline{V}'W_C\underline{V} = 1$  kısıtlaması ile  $\Phi$ 'nin sütun vektörü olan  $\underline{V}$  ye göre kısmi türevi alınıp sıfır vektörüne eşitlenirse,

$$\frac{d\Phi}{d\underline{V}} = \frac{2[(B_C\underline{V})(\underline{V}'W_C\underline{V}) - (\underline{V}'B_C\underline{V})(W_C\underline{V})]}{(\underline{V}'W_C\underline{V})^2} = 0$$

pay ve paydası  $\underline{V}'W_C\underline{V}$  'ye bölünüp, (2.10) nolu tanım ve  $\underline{V}'W_C\underline{V} = 1$  kısıtlaması kullanılırsa yukarıdaki eşitlik,

$$B_C\underline{V} - \Phi W_C\underline{V} = 0$$

şeklini alır. Bu eşitlik  $W_C^{-1}$  ile soldan çarpılırsa,

$$W_C^{-1}B_C\underline{V} - \Phi W_C^{-1}W_C\underline{V} = 0$$

$$(W_C^{-1}B_C - \Phi I)\underline{V} = 0 \quad (2.14)$$

elde edilir (Overal and Klett, 1972).

$$\left| W_C^{-1}B_C - \Phi I \right| = 0 \quad (2.15)$$

denkleminin çözümü özdeğer olarak adlandırılan  $\Phi$ 'nin köklerini, bulunan her  $\Phi$  değeri için  $(W_C^{-1}B_C - \Phi I)\underline{V} = 0$  denkleminin çözümü ise aranan  $\underline{V}$  vektörünü verir.  $\Phi$ 'nin  $i$ 'inci kökü için bulunan vektör  $\underline{V}_i$  ile gösterilirse,  $i$ 'inci ayırma fonksiyonu,

$$f_i = \underline{V}_i' X \quad (2.16)$$

şeklinde ifade edilir.

özdeğer ve özvektörlerini bilgisayar kullanılarak çözebilmek için Choleski ve Jacobi algoritmaları kullanılabilir. Jacobi metodu bir simetrik matrisin bütün özdeğer ve özvektörlerinin bulunabilmesi için kullanılan bir metottur. En sık kullanılan metotlardan biri olan Jacobi metodunda amaç, bir seri ortogonal dönüşümler yardımıyla simetrik bir kare matrisi köşegen matrisi haline dönüştürmektir. Bu durum-

da matrisin köşegeni üzerindeki elemanlar özdeğerleri verir. Özvektörleri bulmak içinde benzer işlemler yapılır.

Jacobi metodunu kullanabilmek için özdeğerleri bulunacak olan matrisin simetrik bir matris olması gerekmektedir. Ancak  $W_C^{-1}$  ve  $B_C$  matrisleri simetrik olsalar bile bunların çarpımı  $W_C^{-1}B_C$  matrisi simetrik değildir. Bu durumda  $W_C^{-1}B_C$  matrisini simetrik bir matrise dönüştürecek olan Choleski metodu kullanılmaktadır (Press and Flanery, 1988).

Analizden sonra elde edilen özvektörler " istenilen ayırma fonksiyonunun standart olmayan ham katsayıları " olarak tanımlanır.

### 2.3 Düzeltilmiş Katsayıların Elde Edilmesi

Yukarıda elde edilen ve ham katsayılar olarak adlandırılan katsayıların yorumlanabilir olmaması yüzünden yeni katsayıların elde edilmesi gerekir. Katsayıların, dolayısıyla ayırma fonksiyonlarının yorumunu kolaylaştırmak için izleyen kısımlarda bazı dönüşümler yapmak suretiyle yeni katsayılar elde edilecektir.

#### 2.3.1 Standart olmayan katsayıların elde edilmesi

Bu katsayılar, özgün verilerin standart hale getirilmeden aşağıdaki dönüşüm formüllerinin kullanılmasıyla bulunabilir.

$$u_i = v_i \sqrt{N - g} \quad \text{ve} \quad u_0 = - \sum_{i=1}^p u_i X_i \quad (2.17)$$

dönüşüm formüllerinden ayırma fonksiyonu,

$$f_{km} = u_0 + u_1 X_{1km} + u_2 X_{2km} + \dots + u_p X_{pkm} \quad (2.18)$$

şeklını alır.

Katsayılarda yapılan düzeltme işlemi ne gruplar arası ayırım miktarını ne de grupların nisbi durumlarını değiştirir. Ancak, ayırma fonksiyonlarının eksen merkezi, tüm ayırma fonksiyonları eksenlerinin sıfır değerini aldığı noktaya kayar. Sistemin merkezi olarak kabul edilen bu nokta sayesinde elemanların sistem merkezine göre hangi konumda oldukları belirlenebilir (Çakmak, 1986).

### 2.3.2 Katsayıların standartlaştırılması

Standart olmayan katsayıların mutlak değeri ve işareti ilgili değişkenin ayırma gücünü gösterir. Eğer bir değişkenin nisbi önemi hakkında bilgi sahibi olmak isteniyorsa, standart katsayıların bulunması gereklidir. Bu katsayılar özgün verilerin standart hale getirilerek (2.17) nolu formül yardımıyla u katsayılarına çevrilmesiyle elde edilebilir. Ancak, özgün verilerin standart hale getirilip katsayıların tekrar hesaplanması yerine aşağıdaki dönüşüm yapılarak standart katsayılar elde edilebilir.

$$C_i = u_i \sqrt{\frac{w_{ii}}{N - g}} = V_i \sqrt{w_{ii}} \quad (2.19)$$

Burada;

$w_{ii}$  : i değişkeni için grup içi kareler toplamı.

$N$  : tüm gruptaki eleman sayısı.

$g$  : grup sayısı

olarak alınmaktadır.

Yukarıdaki formülle bulunan standart katsayılar hangi değişkenlerin fonksiyon değerlerini belirlemeye katkısının daha çok olacağı hakkında fikir verir. Bu işlem standart katsayıların büyüklüklerinin (mutlak değerlerinin) incelenmesiyle yapılır. Daha büyük değerdeki katsayı ilgili değişkenin katkısının, başka bir değişle öneminin daha büyük olacağını gösterir. Uygulamada bu katsayılar genellikle sadece yorum yapmak amacıyla, hangi değişkenlerin hangi fonksiyona ne kadar katkısı olacağını belirlemek amacıyla kullanılmaktadır (Çakmak, 1986).

### 2.4 Ayırma Fonksiyonlarının Sayısı ve Önem Sıraları

Daha önceki kısımlarda ayırma analizinden elde edebilecek maksimum ayırma fonksiyonu sayısının grup sayısından bir eksik veya ayırıcı değişken sayısına eşit olacağını belirtmiştik. (2.15) denkleminin çözümü  $\Phi$  olarak adlandırılan özdeğerleri vermekte olup muhtemel çözüm sayısı ayırıcı değişkenlerin sayısı kadar olmaktadır. Ancak bu çözümlerin bazıları ihmal edilebilir, bazılarının da istatistiksel önemi olmayabilir. Bu bakımdan elde edilecek özdeğerler ya pozitif ya da

sıfır değerli olacak ve dolayısıyla elde edilecek ayırma fonksiyonu sayısı da sıfırdan farklı özdeğer sayısına eşit olacaktır.

Böylece elde edilen özdeğerler büyükten küçüğe doğru sıralandığında en büyük özdeğere karşılık gelen ayırma fonksiyonu en fazla ayırma gücüne, ikinci özdeğerden elde edilen ayırma fonksiyonu birinciden bağımsız ve ikinci dereceden ayırım gücüne sahip olacaktır. İzleyen ayırma fonksiyonları için de aynı şey söylenebilir. Ayırma fonksiyonlarının tümü dikkate alındığında gruplar arası ayırım yeterli olmayabilir. Buna rağmen en azından ayırma fonksiyonlarının kendi aralarındaki önem sıraları ortaya çıkmış olur.

Fonksiyonların birbirlerine göre ne derecede önemli olduğunu ve oransal ayırım güçlerini belirlemek için oransal yüzdelerine bakılır. Bunun için tüm özdeğerler toplanarak toplam ayırma gücü bulunur. Daha sonra her özdeğer bulunan sayıya bölünmek suretiyle her bir fonksiyonun toplam ayırma gücünün yüzde kaçına sahip oldukları, başka bir deyişle her bir fonksiyonun genel varyasyonu açıklama yüzdelerinin ne kadar olduğu belirlenir. Bunun matematiksel ifadesi ;

$$f_i = \frac{\phi_i}{\sum_{i=1}^r \phi_i} * 100 \quad (2.20)$$

şeklinde yazılabilir. Burada r özdeğer sayısını göstermektedir.

### 3. SINIFLANDIRMA ANALIZI

Sınıflandırma, hangi gruba veya yığına ait olduğu kesin olarak bilinmeyen bir elemanın üzerinde yapılan gözleme dayanarak, en büyük olasılıkla hangi grup veya yığına atanacağını tahmin edilmesidir. Bock'un (1975) önerisi doğrultusunda, elemanın grup veya yığınlara sınıflandırma işleminin yapılabilmesi için gerekli bazı temel varsayımlar şu şekilde sıralanabilir (Çakmak, 1986).

- a) Elemanın analiz öncesi belirlenen sonlu sayıdaki grup veya yığınlardan birisine ait olduğu bilinmektedir.
- b) Grup veya yığınların nisbi büyüklükleri bilinmekte veya tahmin edilmektedir.
- c) Elemanlar üzerinde p tane özellik bakımından yapılan gözlem vektörü bilinmektedir.
- d) Her bir grup veya yığına ait olasılık yoğunluk fonksiyonları bilinmekte veya tahmin edilmektedir.

Problem, bu bilgilerden yararlanarak, elemanları en az hata ile grup veya yığınlara sınıflandıracak istatistik karar fonksiyonlarının elde edilmesidir. İzleyen kesimlerde ilk olarak genel sınıflandırma kuralı verilecek, daha sonra da sınıflandırma ile ilgili diğer yöntemler üzerinde durulacaktır.

#### 3.1 Sınıflandırma Kuralları

Yukarıda da belirtildiği gibi, sınıflandırma işleminde en önemli olayın elemanların en az hata ile grup veya yığınlara sınıflandırılması olduğundan, herhangi bir eleman üzerinde yapılan sınıflandırmada elemanın yanlış bir yığına sınıflandırma olasılığının minimum olması gereklidir. Bu durumu daha iyi açıklayabilmek amacıyla önce sadece iki yığının, daha sonra da ikiden fazla yığının sözkonusu olduğu durum ele alınarak konuya açıklık getirilmeye çalışılacaktır.

##### 3.1.1 İki grup için geliştirilen sınıflandırma metodu

Üzerinde  $X' = (X_1, X_2, \dots, X_p)$  gözlemlerinin yapıldığı bir eleman p boyutlu uzayda bir noktayı;  $G_1$  ve  $G_2$  yığınları;  $f_1(X)$  ve  $f_2(X)$  bu

yığınlaraya ait yoğunluk fonksiyonlarını gösterebilirsin. Ayrıca, ele alınan elemanın  $G_1$  yığınınından gelmiş olma olasılığı  $q_1$ ,  $G_2$  yığınınından gelmiş olma olasılığı  $q_2$  olsun. Ön olasılıklar olarak adlandırılan  $q_1$  ve  $q_2$ 'nin bilindiğini varsayarak suretiyle yanlış sınıflandırma olasılıklarını hesaplayabilmek için,  $p$  boyutlu uzayda  $X$ ' gözlemlerine karşı gelen tüm noktaları içine alan bölgeyi  $R$  ile gösterip, bu bölgeyi  $R_1$  ve  $R_2$  gibi iki bölgeye ayıracak öyle bir ölçüt bulunacaktır ki,  $X$ ' noktası  $R_1$  bölgesine düştüğünde  $G_1$  yığına,  $R_2$  bölgesine düştüğünde  $G_2$  yığına sınıflandırılınsın.

Gerçekte  $G_1$  yığına ait olan bir elemanı  $G_2$  yığına yanlış sınıflandırma olasılığı,

$$P(2,1/R) = P_1 = \int_{R_2} f_1(X) dX \quad (3.1)$$

ve gerçekte  $G_2$  yığına ait olan bir elemanı  $G_1$  yığına yanlış sınıflandırma olasılığı,

$$P(1,2/R) = P_2 = \int_{R_1} f_2(X) dX \quad (3.2)$$

olmaktadır (Lachenbruch, 1975).

Daha önce belirtildiği gibi,  $G_1$  ve  $G_2$  yığınlarına ilişkin ön olasılıkları dikkate almak suretiyle,  $G_1$ 'den bir eleman alma ve bu elemanı doğru olarak sınıflandırma olasılığı  $q_1 P(1,1/R)$ , yanlış sınıflandırma olasılığı ise  $q_1 P(2,1/R)$ 'dir. Benzer şekilde  $G_2$ 'den alınan bir bireyin doğru sınıflandırma olasılığı  $q_2 P(2,2/R)$  ve yanlış sınıflandırma olasılığı  $q_2 P(1,2/R)$  bulunur (Çakmak, 1986).

Buna göre toplam yanlış sınıflandırma olasılığı

$$\begin{aligned} T(R, f) &= q_1 \int_{R_2} f_1(X) dX + q_2 \int_{R_1} f_2(X) dX \\ &= q_1 \left[ 1 - \int_{R_1} f_1(X) d(X) \right] + q_2 \int_{R_1} f_2(X) d(X) \\ &= q_1 + \int_{R_1} [q_2 f_2(X) - q_1 f_1(X)] d(X) \end{aligned} \quad (3.3)$$

ifadesinin minimum yapılması suretiyle optimum sınıflandırma kuralı bulunabilir (Lachenbruch, 1975). Bunun için eğer  $R_1$  bölgesini göz önünde bulundurursak,  $R_1$  bölgesine düşen elemanlar için,

$$q_2 f_2(X) - q_1 f_1(X) < 0 \quad (3.4)$$

olduğunda, toplam yanlış sınıflandırma olasılığı minimum olacaktır. Buna göre sınıflandırma kuralı aşağıdaki gibi yazılabilir. Herhangi bir eleman için,

$$\frac{f_1(X)}{f_2(X)} > \frac{q_2}{q_1} \quad (3.5)$$

koşulu sağlanıyorsa, eleman  $G_1$  yığına aksine halde  $G_2$  yığına sınıflandırılır (Lachenbruch, 1975).

### 3.1.2 İkiden fazla grup olması halinde sınıflandırma metodu

Daha önce sadece iki yığın söz konusu olduğu durumlar için elde edilen sınıflandırma kuralları,  $k$  tane yığın söz konusu olduğu durumlar için genelleştirilebilir.  $G_1, G_2, \dots, G_k$  yığınları;  $f_1(X), f_2(X), f_3(X), \dots, f_k(X)$  yığına ait yoğunluk fonksiyonlarını gösterebilir. Elemanları bu yığına sınıflandırmak amacıyla  $p$  boyutlu uzay, karşılıklı ayrı (mutually exclusive) ve bütüne tamamlayan  $R_1, R_2, \dots, R_k$  bölgelerine ayrılır. Buna göre  $R_g$  bölgesine düşen bir elemanın  $G_g$  yığına geldiği söylenebilir. Ayrıca  $G_g$  yığına ait olan elemanın yanlışlıkla  $G_h$  yığına sınıflandırılmasının maliyeti  $C_{hg}$  olsun. Eğer  $G_g$  yığına ait olduğu bilinen bir eleman yanlışlıkla  $G_h$  yığına sınıflandırılma olasılığı,

$$P_{hg} = P(h, g/R) = \int_{R_h} f_g(X) d(X) \quad (3.6)$$

olur. Yığına ilişkin ön olasılıklar sırasıyla  $q_1, q_2, \dots, q_k$  olarak gösterilirse, yanlış sınıflandırmadan beklenen kayıp,

$$\sum_{g=1}^k q_g \left\{ \sum_{g=1, g \neq h}^k C_{hg} P_{hg} \right\} \quad (3.7)$$

olur.  $R_1, R_2, \dots, R_k$  bölgelerini belirlemek için bu ifadenin minimum yapılması gerekir.

Elemana ait gözlem vektörünün elemanları  $(X_1, X_2, \dots, X_p)$  olarak verildiğine göre, elemanlar koşullu olasılıklarının daha yüksek olduğu yığınlara sınıflandırılırsa yukarıdaki ifade minimum yapılmış olur. Herhangi bir elemanın  $G_g$  yığınınından gelmiş olmasının koşullu olasılığı,

$$\frac{q_g f_g(X)}{\sum_{i=1}^k q_i f_i(X)} \quad g=1, 2, \dots, k \quad (3.8)$$

dir. Eğer bu eleman yanlışlıkla  $G_h$  yığına sınıflandırılmışsa,

$$\text{Beklenen Kayıp} = \frac{\sum_{g=1, g \neq h}^k q_g f_g(X) C_{hg}}{\sum_{i=1}^k q_i f_i(X)} \quad (3.9)$$

olur. Bu durumda beklenen kaybı minimize etmek için (3.9)'nın payındaki ifadenin minimize edilmesi gereklidir. Yani,

$$\sum_{g=1, g \neq h}^k q_g f_g(X) C_{hg} \quad (3.10)$$

ifadesi minimum yapılarak,

$$\frac{f_g(X)}{f_h(X)} > \frac{C_{gh} q_h}{C_{hg} q_g} \quad h=1, 2, \dots, k \text{ ve } g \neq h \quad (3.11)$$

sınıflandırma kuralı elde edilir. Buna göre yukarıdaki koşul sağlanıyorsa, eleman  $G_g$  yığına sınıflandırılır. Yanlış sınıflandırma maliyetleri tüm yığınlar için  $C_{hg}=1$  olduğu kabul edilirse,

$$\frac{f_g(X)}{f_h(X)} > \frac{q_h}{q_g} \quad h=1, 2, \dots, k \text{ ve } g \neq h \quad (3.12)$$



kuralı elde edilir ve eleman  $G_g$  yığına sınıflandırılır (Lachenbruch, 1975).

### 3.2 Uzaklık Fonksiyonlarına Dayalı Sınıflandırma

Bundan önceki kısımda Bayes yaklaşımıyla elde edilen genel sınıflandırma kurallarında, yığına ilişkin ön olasılıkları ve yanlış sınıflandırma maliyetlerini esas alarak sınıflandırma hatalarının, başka bir deyişle beklenen kaybın, minimum yapılması suretiyle sınıflandırma kuralları belirlenmişti. Bu kısımda ise, üzerinde  $p$  sayıda özellik bakımından gözlem yapılan elemanın gözlem vektörü ile grupların ortalama vektörleri arasındaki uzaklık esas alınarak sınıflandırma kuralları incelenecektir.

Mahalanobis, varyans-kovaryans matrisleri  $\Sigma$  eşit, ortalama vektörleri  $\mu_1$  ve  $\mu_2$  olan ve aynı değişken kümesiyle tanımlanan çok değişkenli iki yığın için genelleştirilmiş uzaklık fonksiyonunu,

$$D^2 = (\mu_1 - \mu_2)' \Sigma^{-1} (\mu_1 - \mu_2) \quad (3.13)$$

şeklinde tanımlamıştır (Lachenbruch, 1975). Bu bağıntıdan yararlanarak sınıflandırılması istenen eleman ile  $g$ 'inci grup arasındaki uzaklık,

$$D_{g}^2 = (\bar{X}_1 - \mu_2)' \Sigma_g^{-1} (\bar{X}_1 - \mu_2) \quad (3.14)$$

şeklinde yazılabilir. Uzaklık fonksiyonlarına dayalı sınıflandırma önce iki grubun söz konusu olduğu durumlar için incelenecek, daha sonra da  $k$  grup için genelleştirilecektir.

#### 3.2.1 İki grup için sınıflandırma kuralları

Çok değişkenli normal dağılım gösteren  $G_1$  ve  $G_2$  yığına ilişkin varyans-kovaryans matrisi  $\Sigma$  ve ortalama vektörleri  $\mu_1$  ve  $\mu_2$ , sınıflandırılması istenen elemanın  $p$  elemanlı gözlem vektörü  $X$  ve yığına ilişkin ön olasılıklar  $q_1$  ve  $q_2$  olsun. Ayrıca yanlış sınıflandırma maliyetlerinin birbirine eşit olduğu kabul edilirse, yığın parametrelerinin bilindiği ve bilinmediği durumlar için sınıflandırma kuralları bulunabilir.

Cok deęişkenli normal daęılım gösteren yığınlara ilişkin yoğunluk fonksiyonları,

$$f_i(X) = (2\pi)^{-p/2} \Sigma^{-1/2} \exp\left\{\frac{-1}{2}(X - \mu_i)' \Sigma^{-1}(X - \mu_i)\right\} \quad (3.15)$$

i=1,2,...,k

şeklinde gösterilebilir (Lachenbruch, 1975). Burada X, p elemanlı gözlem vektörünü,  $\Sigma$  yığınlara ilişkin varyans-kovaryans matrisini ve  $\mu_i$  ise yığınlara ilişkin ortalamalar vektörünü göstermektedir.

Yığın parametrelerinin bilindięi durumlar için yığınlara ilişkin yoğunluk fonksiyonlarını (3.5) denkleminde yerine koyarak her iki tarafında doğal logaritması alınır ve gerekli işlemler yapılırsa aşağıdaki sınıflandırma kuralı elde edilir.

$$(X - \mu_1)' \Sigma^{-1}(X - \mu_1) < (X - \mu_2)' \Sigma^{-1}(X - \mu_2) - 2\ln(q_2/q_1) \quad (3.16)$$

Buna göre sınıflandırılması istenen eleman için (3.16)'daki koşullar sağlanıyorsa birey  $G_1$  yığına aksine halde  $G_2$  yığına sınıflandırılır.

Yığın parametrelerinin bilinmedięi durumlar içinse, sınıflandırma işlemi yapılırken çoęu uygulamalarda yığın parametreleri olan  $\Sigma$ ,  $\mu_1$  ve  $\mu_2$  bilinmedikleri için bunların yerine belirlenen örnek istatistikleri yığın parametreleri yerine kullanılır. Burada  $\Sigma$  yerine gruplar içi varyans-kovaryans matrisi  $W_C$ ,  $\mu_1$  ve  $\mu_2$  yerine ise sırasıyla birinci ve ikinci yığınlardan alınan örneklerin ortalama vektörleri  $\bar{X}_1$  ve  $\bar{X}_2$  kullanılacaktır.  $W_C$  matrisi şu şekilde tanımlanmıştır.

$$W_C = \frac{W}{(N - g)} \quad (3.17)$$

Buna göre (3.16) nolu denklem yeniden düzenlenirse,

$$(X - X_1)' W_C^{-1}(X - X_1) < (X - X_2)' W_C^{-1}(X - X_2) - \ln(q_2/q_1) \quad (3.18)$$

elde edilir.

### 3.2.2 İkiden fazla grup için sınıflandırma kuralları

Bundan önceki kısımda sadece iki grubun söz konusu olduęu durum-

lar için elde edilen sınıflandırma kurallarını k grubun sözkonusu olduğu durumlar için genelleştirmek mümkündür.

Yığınlara ilişkin varyans-kovaryans matrislerinin eşit varsayıldığı durumda  $\Sigma$  yerine  $W_C$  matrisi,  $\mu_g$  ve  $\mu_h$  yerine ise örnek ortalama vektörleri  $X_g$  ve  $X_h$  kullanılırsa,

$$(X - X_g)' W_C^{-1} (X - X_g) < (X - X_h)' W_C^{-1} (X - X_h) - 2 \ln \left( \frac{q_h}{q_g} \right) \quad (3.19)$$

$$h=1,2,\dots,k; g \neq h$$

sınıflandırma kuralı elde edilir. Buna göre sınıflandırılması istenen eleman için yukarıdaki denklem hangi g değeri için sağlanıyorsa, eleman  $G_g$  grubuna sınıflandırılır (Çakmak, 1986).

### 3.3 Kanonik Vektör Yaklaşımına Göre Sınıflandırma

Daha önceki kısımlarda da incelenen p tane ayırıcı değişkenin doğrusal bileşimininden oluşan ayırma fonksiyonu,

$$f = V'X, \quad (3.20)$$

olarak gösterilmektedir. İki yığının olması durumunda; yığın ortalamaları  $\mu_1$  ve  $\mu_2$ 'yi sırasıyla ayırma fonksiyonundaki ayırıcı değişkenler yerine kullandığımızda ayırma fonksiyonlarının ilgili yığınlardaki ortalamaları elde edilir. Yani, birinci yığın için ayırma fonksiyonunun ortalama değeri

$$f_1 = V'\mu_1 \quad (3.21)$$

ve ikinci yığın için ayırma fonksiyonunun ortalama değeri

$$f_2 = V'\mu_2 \quad (3.22)$$

Fisher'e (1936) göre sınıflandırılması istenen herhangi bir eleman için ayırma fonksiyonundan elde edilen değer, grup ortalamasından elde edilen değerden sapması ne kadar küçükse eleman bu gruba sınıflandırılır. Bu durum aşağıdaki sınıflandırma kuralı ile gösterilebilir (Lachenbruch, 1975).

Herhangi bir eleman için,

$$\left| f_x - f_1 \right| < \left| f_x - f_2 \right| \quad (3.23)$$

koşulu sağlanıyorsa eleman birinci gruba sınıflandırılır.

İkiden fazla yığın olması durumunda grup sayısından bir eksik veya ayırıcı değişken sayısı kadar özdeğer elde edileceğinden, ayırma fonksiyonu sayısı sıfırdan farklı pozitif özdeğer sayısına eşit olacaktır. Bu durumda elde edilen tüm ayırma fonksiyonları sınıflandırma işleminde kullanılabilir. Elde edilen ayırma fonksiyonları sayısını  $r$  ile gösterirsek iki yığın için önerilen sınıflandırma koşulu aşağıdaki forma dönüşür.

Herhangi bir eleman için,

$$\sum_{l=1}^r [V_l'(X - \mu_j)]^2 = \min_j \sum_{l=1}^r [V_l'(X - \mu_j)]^2 \quad (3.24)$$

$j=1,2,\dots,g$  ve  $j \neq i$

veya daha basit olarak,

$$\sum_{l=1}^r [V_l'(X - \bar{X}_i)]^2, \quad i=1,2,\dots,g \quad (3.25)$$

ifadesi hangi  $i$  değeri için minimum değer veriyorsa eleman o gruba sınıflandırılır (Lachenbruch, 1975). Burada  $g$  grup sayısıdır ve yığın ortalaması  $\mu_j$  yerine örnek değerlerinden elde edilen örnek ortalaması  $\bar{X}_i$  kullanılmaktadır.

#### 4. ÇOKLU AYIRMA ANALİZİNİN PARAMETRİK MODELLERİN ÖNEM DERECELERİNİ BULMADA KULLANILMASI

Bilindiği gibi ses sinyallerinin incelenebilmesi için birtakım parametrik modeller kullanılmaktadır. Bunlar,

- a) Ortalama enerji seviyeleri
- b) Sıfır geçiş sayıları
- c) Maksimum genlikler
- c) Otokorelasyonlar
- d) Doğrusal öngörücü katsayıları (LPC)
- e) Logaritmik alan oranları (LAR)
- f) Cepstral'ler
- g) Formant frekansları

ve benzerleri olabilir.

Şimdiye kadar yapılan konuşmayı tanıma çalışmalarında yukarıda gösterilen parametrik modellerin bir veya birkaçı sıklıkla kullanılmıştır. Ancak hangi parametrik modelin konuşmayı tanımada diğerlerine göre daha verimli sonuçlar verdiğinin araştırılması için, çalışmalardan elde edilen sonuçların titizlikle incelenmesi gereklidir. Bununla birlikte, çoklu ayırma analizi kullanılarak parametrik modellerin konuşmayı tanımada ne derecede önemli oldukları bulunabilir. Bunun için öncelikle gruplar oluşturulmalıdır. Grupları oluşturan elemanlar yukarıdaki parametrik modellerden biri veya birkaçı ile temsil edildikten sonra çoklu ayırma analizi gerçekleştirilir. Burada parametrik modellerin herbir terimi ayırma analizindeki ayırıcı değişkenlere karşılık gelmektedir.

Analiz sonucunda elde edilen ayırma fonksiyonlarından grupları en iyi ayıran ya da diğer bir değişle en büyük özdeğerden elde edilen ayırma fonksiyonunun katsayıları büyükten küçüğe doğru sıralandıklarında, ilgili ayırıcı değişkenlerin önem sıraları da ortaya çıkmaktadır. Yani, hangi ayırıcı değişkenin katsayısı diğerlerine göre mutlak değerce daha büyükse, ilgili ayırıcı değişken grupları diğer ayırıcı değişkenlere göre daha iyi ayırmaktadır.

Daha önce parametrik modellerin herbir terimini ayırma fonksiyonundaki ayırıcı değişkenlere karşılık seçtiğimizden, analiz sonucunda ayırıcı değişkenler arasında elde ettiğimiz önem sırası aynı zamanda parametrik modellerin birbirlerine göre önem sırasınıda vermektedir.

#### 4.1 Parametrik Modellerin Nisbi Önemlerinin Elde Edilmesi

Parametrik modellerin nisbi önemlerini elde edebilmek için, öncelikle yukarıda açıklanan parametrelerin hesaplanması gereklidir. Bunun için 9 erkek 5 bayandan alınan sekiz adet ünlüye ait ses sinyalleri 103 ms'lik frame aralıklarıyla örneklenmiş ve depolanmıştır. İzleyen kısımlarda 128 örnek'lik parçalar 1 frame olarak adlandırılacaktır. Daha sonraki aşamalarda her bir frame için yukarıda açıklanan parametrik modeller hesaplanmıştır.

#### 4.2 Sıfır Geçiş Sayıları, Enerji Seviyeleri ve Maksimum Genlikler Arasındaki Önem Derecelerinin İncelenmesi

Analizin ilk aşamasında, parametrik modellerden enerji seviyeleri, sıfır geçiş sayıları ve maksimum genlikler arasındaki önem dereceleri belirlenmeye çalışılmıştır. Bu üç parametrenin analiz için, birlikte seçilmesinin en önemli nedeni, üçünün de her frame için bir parametre vermeleri yüzündendir. Yani her frame için sıfır geçiş sayısı, enerji seviyesi ve maksimum genliğin herbiri tek bir değerden ibarettir. Böylece, ayırıcı değişkenler olarak sıfır geçiş sayıları, enerji seviyeleri ve maksimum genlikler seçildiğinde, ünlüler grupları, frame'ler elemanları oluşturmaktadır. Ayrıca ayırıcı değişkenler kendi aralarında normalize edilmişlerdir. Normalizasyon işlemi, herbir ayırıcı değişkenin grup içindeki ortalamasına bölünmesiyle gerçekleştirilmektedir.

Coklu ayırma analizi uygulandıktan sonra grupları ayırmada diğerlerinden daha etkili olan ayırma fonksiyonu katsayıları mutlak değerce büyükten küçüğe doğru sıralandığında hemen herkes için aşağıdaki önem sırası ortaya çıkmaktadır.

- 1) Enerji seviyeleri
- 2) Sıfır geçiş sayıları
- 3) Maksimum genlikler

Incelemelerimizde enerji seviyeleri ile sıfır geiş sayılarının önem dereceleri arasındaki fark hemen herkes için küçük çıkmıştır. Ayırma fonksiyonu katsayıları incelendiğinde enerji seviyeleri ve sıfır geiş sayıları grupları ayırmada maksimum genliklerden daha önemli olduğu görülmektedir (bkz. çizelge 4.1).

Bu durumda grupları ayırmada enerji seviyeleri ve sıfır geiş sayılarının yeterli olabileceği ancak maksimum genliklerin grupları ayırmada yetersiz ve hatta önemsiz olduğu sonucuna varılabilir.

#### 4.3 LPC, Cepstral ve LAR Parametreleri Arasındaki Önem Derecelerinin İncelenmesi

Analizin ikinci aşamasında, parametrik modellerden LPC, Cepstral ve LAR arasındaki önem dereceleri elde edilmeye çalışılmıştır. Analizden önce ses örneklerinin herbir frame'i için 12 şer adet LPC, Cepstral ve LAR parametreleri elde edilmiştir. Daha sonra bu parametrelerin her birine bir ayırıcı deęişken tahsis edilerek analiz işlemine geçilmiştir. Toplam ayırıcı deęişken sayısı 36 olmaktadır. Bu şekilde bir ile on ikinci ayırıcı deęişkenler LPC'leri, on üç ile yirmi dördüncü ayırıcı deęişkenler Cepstralleri ve yirmi beşinci ile otuz altıncı ayırıcı deęişkenler LAR'ları temsil etmektedir.

Analiz işlemine başlarken, ayırıcı deęişkenler olarak yukarıda belirtildiği gibi toplam 36 adet LPC, Cepstral ve LAR parametreleri; grup olarak 8 adet ünlü harf ve bu grupları oluşturan elemanlar olarak da ünlülere ait frame'ler alınmıştır ve ayırıcı deęişkenlere normalizasyon uygulanmıştır.

Analiz sonucunda en büyük özdeęerden elde edilen ayırma fonksiyonu katsayıları incelendiğinde, genelde LPC parametrelerinin dięer parametrelere göre grupları ayırmada daha etkili olduğu görülmüştür. Özellikle LPC parametrelerinden 3, 2 ve 8 nolu parametrelerinin ilk sıraları paylaştığı gözlenmektedir. Ancak bu sıralama bazı kişiler için farklılık gösterebilmektedir (bkz. çizelge 4.2).

İkinci dereceden önemli parametreler genelde Logaritmik Alan Oranları olarak gözlenmiştir. Ancak LPC'lerden farklı olarak LAR parametre bloğunun ancak küçük bir bölümü LPC'lerden sonraki sıraları

paylaşmaktadır. Diğer LAR parametreleri ise Cepstrallerden bile önem siz bulunabilmektedir. Bazı kişiler için LAR parametrelerinin büyük bir bölümü Cepstraller den daha önemsiz çıkabilmektedir.

Sonuç olarak LPC parametrelerinin tümünün hemen herkes için grupları ayırmada önemli olduğunu, ancak LAR ve Cepstral parametrelerinin grupları ayırmada yetersiz kaldıklarını söyleyebiliriz.

Bu çalışmaya ek olarak LPC, Cepstral ve LAR parametreleri, grupları ayırma bakımından ayrı ayrı incelenmiştir. Sekiz ünlü grubu için LPC parametreleri kullanıldığında, çoklu ayırma analizinden elde edilen sonuçlara göre, grupları ayırmada genelde üçüncü, ikinci, dördüncü ve dokuzuncu LPC parametreleri önemli çıkmaktadır (bkz. çizelge 4.3).

Grupları ayırmada sadece Cepstral parametreleri kullanıldığında, analiz sonucuna göre genelde üçüncü, altıncı ve beşinci parametreler önemli çıkmaktadır (bkz.çizelge 4.4).

LAR parametreleri grupları ayırmada kullanıldığında, altıncı ve beşinci parametreler diğerlerine göre daha etkili olduğu görülmektedir (bkz.çizelge 4.5).

#### 4.4 Otokorelasyon, Lpc Parametreleri ve Formant Frekansları Arasındaki Önem Derecelerinin İncelenmesi

Analizin üçüncü aşaması olarak Otokorelasyon, LPC parametreleri ve Formant frekansları arasındaki önem dereceleri belirlenmeye çalışılmıştır. Analizden önce tüm ünlüler için Otokorelasyonlar, LPC parametreleri ve Formant frekansları hesaplanmıştır. Formant frekansları, Lpc parametreleri kullanılarak oluşturulan frekans spektrumundan elde edilmişlerdir.

Otokorelasyonlar ve Lpc parametrelerinin çoklu ayırma analizine uygulanmasında her hangi bir sorun çıkmamıştır. Çünkü, seslilere ait her bir frame için sabit sayıda parametre üretilmektedir. Ancak on iki Lpc parametresi için oluşturulan frekans spektrumunda altı veya daha az miktarda Format frekansı elde edilmektedir. Yani her bir frame için elde edilen formant sayısı farklı olabilmektedir. Bu olumsuz durumu ortadan kaldırmak için ilk üç formant frekansı göz önünde bulundurul-



muştur. Diğer bir deyişle, bazı frameler için üçden fazla formant elde edilse bile, bunların ilk üçü alınmıştır.

Analiz sırasında karşımıza çıkan diğer bir sorun ise, özellikle otokorelasyon parametrelerinin nümerik değerlerinin, LPC ve formantlara göre çok büyük olmasıdır. Bu sorun parametrelerin herbirinin kendi grup ortalamasına bölünmesiyle normalize edilerek ortadan kaldırılmıştır.

Analiz sonucunda, en büyük özdeğerden elde edilen ayırma fonksiyonu katsayıları incelendiğinde, Otokorelasyon ve LPC parametrelerinin hemen herkes için ilk sıraları paylaştıkları söylenebilir. Genelde Otokorelasyon parametreleri LPC parametrelerinden daha önemli çıktığı gözlenmiştir. Formant frekansları ise hemen herkes için düşük önem sıralarını paylaşmıştır. Ancak bazı kişiler için formant frekanslarından biri veya bir kaç üst sıralara yükselmektedir.

Sonuç olarak LPC ve Otokorelasyon parametrelerinin grupları ayırmada formant frekanslarından daha etkili oldukları söylenebilir. (bkz.çizelge 4.6 ve çizelge 4.7).

Benzer şekilde, grupları ayırmada sadece Otokorelasyonlar kullanıldığında, analiz sonucunda ikinci ve üçüncü parametrelerin diğerlerine göre daha önemli oldukları gözlenmiştir (bkz.çizelge 4.8).

SAHISLAR	1.DERECE DEN ONEMLI AYIRMA FONKSIYONUNUN EN ONEMLI KATSAYILARI			2.DERECE DEN ONEMLI AYIRMA FONKSIYONUNUN EN ONEMLI KATSAYILARI		
	1.DERECE ONEMLI KATSAYI	2.DERECE ONEMLI KATSAYI	3.DERECE ONEMLI KATSAYI	1.DERECE ONEMLI KATSAYI	2.DERECE ONEMLI KATSAYI	3.DERECE ONEMLI KATSAYI
SALIH	2	1	3	1	3	2
SELCUK	2	1	3	1	3	2
METIN D.	2	1	3	1	3	2
UMIT	2	1	3	1	3	2
MURAT	2	1	3	1	3	2
MUAMMER	2	1	3	1	3	2
OMER	2	1	3	1	3	2
MURAT O.	2	1	3	1	3	2
METIN	2	1	3	1	3	2
NAZAN	2	1	3	1	3	2
IDIL	2	1	3	1	3	2
ZEYNEP	2	1	3	1	3	2
GULAY	2	1	3	1	3	2
MUNEVER	2	1	3	1	3	2

Cizelge 4.1. Sıfır Geçiş Sayıları, Enerji seviyeleri ve Maksimum Genlikler için en önemli ayırma fonksiyonu katsayıları

SAHISLAR	1.DERECE DEN ONEMLI AYIRMA FONKSIYONUNUN EN ONEMLI KATSAYILARI			2.DERECE DEN ONEMLI AYIRMA FONKSIYONUNUN EN ONEMLI KATSAYILARI		
	1.DERECE ONEMLI KATSAYI	2.DERECE ONEMLI KATSAYI	3.DERECE ONEMLI KATSAYI	1.DERECE ONEMLI KATSAYI	2.DERECE ONEMLI KATSAYI	3.DERECE ONEMLI KATSAYI
SALIH	7	6	9	3	2	1
SELCUK	10	9	11	5	3	2
METIN D.	3	4	5	3	6	2
UMIT	2	3	9	8	9	7
MUAMMER	9	10	2	3	2	4
OMER	3	4	2	4	3	7
METIN	5	4	3	9	6	8
NAZAN	2	3	8	2	9	8
IDIL	9	6	8	5	3	6
ZEYNEP	8	7	9	8	7	9
GULAY	8	9	10	2	9	8
MUNEVER	7	6	2	2	7	8

Cizelge 4.2. LPC, Cepstral ve LAR parametreleri için en önemli ayırma fonksiyonu katsayıları

SAHISLAR	1.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI			2.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI		
	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI
SALIH	6	7	5	3	2	1
SELCUK	10	11	9	3	5	4
METİN D.	3	4	2	3	6	2
UMIT	2	3	1	2	3	4
MURAT	5	4	6	5	6	4
MUAMMER	9	8	5	3	2	4
ÖMER	3	4	2	2	10	3
MURAT Ö.	4	3	5	5	4	6
METİN	5	4	3	2	3	1
NAZAN	2	3	1	2	9	3
IDİL	2	9	10	9	2	10
ZEYNEP	8	9	7	8	7	9
GULAY	8	9	10	8	7	6
MUNEVER	3	2	4	2	3	1

Çizelge 4.3. LPC parametreleri için en önemli ayırma fonksiyonu katsayıları

SAHISLAR	1.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI			2.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI		
	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI
SALIH	5	6	4	4	3	5
SELCUK	6	7	5	3	12	2
METIN D.	6	7	5	4	3	5
UMIT	3	6	4	3	10	2
MURAT	3	5	13	4	13	3
MUAMMER	3	4	11	4	5	3
OMER	3	5	6	10	5	3
MURAT O.	3	5	10	13	3	4
METIN	6	5	8	3	4	9
NAZAN	3	5	6	7	4	10
IDIL	7	4	9	7	9	10
ZEYNEP	5	6	4	8	7	3
GULAY	2	3	6	10	4	2
MUNEVER	5	3	7	3	4	7

Çizelge 4.4. Cepstral parametreleri için en önemli ayırma fonksiyonu katsayıları

SAHISLAR	1.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI			2.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI		
	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI
SALIH	7	6	10	1	3	4
SELCUK	1	10	5	3	4	1
METİN D.	1	4	3	1	3	2
UMIT	2	5	6	1	2	4
MURAT	6	5	10	1	3	4
MUAMMER	5	6	2	1	7	2
OMER	6	3	4	1	4	3
MURAT O.	6	5	1	1	3	4
METİN	3	4	1	1	3	2
NAZAN	4	5	2	1	2	5
IDİL	2	3	1	1	2	7
ZEYNEP	4	5	7	1	2	3
GULAY	2	7	5	1	8	5
MUNEVER	2	3	9	1	2	4

Çizelge 4.5. LAR parametreleri için en önemli ayırma fonksiyonu katsayıları

SAHISLAR	1.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI			2.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI		
	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI
SALIH	2	19	20	15	16	14
SELCUK	15	24	16	2	1	10
METIN D.	9	2	3	2	3	1
UMIT	2	1	3	11	10	4
MUAMMER	2	3	1	2	1	3
OMER	2	1	4	2	3	1
METIN	2	3	10	2	1	10
NAZAN	2	3	1	2	3	1
IDIL	1	2	4	1	4	2
ZEYNEP	2	1	3	2	3	10
GULAY	2	3	9	2	3	1
MUNEVVER	2	1	5	10	9	2

Çizelge 4.6. Otokorelasyon, LPC ve Formant Frekansları için en önemli ayırma fonksiyonu katsayıları

SAHISLAR	1.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI			2.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI		
	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI
SALIH	2	19	20	16	15	14
SELCUK	2	10	1	16	17	18
METIN D.	2	3	9	9	11	2
UMIT	2	1	3	4	11	10
MUAMMER	2	3	1	2	1	3
OMER	4	1	2	2	1	3
METIN	2	3	10	2	10	11
NAZAN	2	3	1	2	3	1
IDIL	1	2	4	4	5	9
ZEYNEP	2	1	3	2	3	10
GULAY	10	9	4	2	3	1
MUNEVVER	2	1	5	2	10	9

Çizelge 4.7. Otokorelasyonlar ve LPC'ler için en önemli ayırma fonksiyonu katsayıları



SAHISLAR	1.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI			2.DERECE DEN ÖNEMLİ AYIRMA FONKSİYONUNUN EN ÖNEMLİ KATSAYILARI		
	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI	1.DERECE ÖNEMLİ KATSAYI	2.DERECE ÖNEMLİ KATSAYI	3.DERECE ÖNEMLİ KATSAYI
SALIH	2	3	7	2	10	3
SELCUK	2	3	1	2	1	3
METIN D.	3	8	2	3	8	2
UMIT	2	1	9	12	11	2
MURAT	2	3	1	2	1	3
MUAMMER	3	1	9	12	2	11
OMER	2	1	3	2	1	3
MURAT O.	2	3	1	2	1	3
METIN	2	9	12	2	3	9
NAZAN	2	3	1	2	3	1
IDIL	2	3	1	4	3	2
ZEYNEP	10	9	11	9	3	8
GULAY	3	2	12	2	3	9
MUNEVER	2	3	9	10	9	3

Çizelge 4.8. Otokorelasyonlar için en önemli ayırma fonksiyonu katsayıları

## 5. ÇOKLU AYIRMA ANALİZİNİN ÜNLÜLERİ SINIFLANDIRMADA KULLANILMASI

Sınıflandırma analizi başlığı altında, Bayes sınıflandırma metodu, uzaklık fonksiyonlarına dayalı sınıflandırma ve Kanonik vektör yaklaşımına göre sınıflandırma metotları incelenmiştir. Burada ilk iki metot ön olasılıkları dikkate alarak sınıflandırma yapmaktadır.

Bu tez çalışmasında, ön olasılıkların sınıflandırma öncesi belirlenmesindeki zorluklar yüzünden, kanonik vektör yaklaşımı ünlüleri sınıflandırmak için kullanılmıştır.

Üçüncü bölümde incelediğimiz kanonik vektör yaklaşımına göre sınıflandırma metodu, hangi ünlü grubuna ait olduğu bilinmeyen bir gözlem vektörünü minimum hata ile sınıflandırmada kullanılabilir.

Sınıflandırma işlemine geçmeden önce gözlem vektörü olarak hangi parametrik modelin kullanılacağına karar vermek gereklidir. Bunun için dördüncü bölümde elde edilen sonuçlara göre LPC, otokorelasyon, LAR, cepstral parametreleri kullanılarak tek bir şahıs için ünlüler sınıflandırılmaya çalışılmıştır.

Bir erkek konuşmacıdan, her bir ünlü harfi dört kez söylemesi istenmiştir. Bu şekilde birden çok örnekle çalışılarak sınıflandırma hatasının minimuma indirilmesi amaçlanmıştır. Konuşmacının verdiği ilk örnek grubu şablon (template) olarak kullanılarak bu şablonun ve diğer üç örnek grubunun ayrı ayrı otokorelasyon, LPC, LAR ve cepstral parametreleri bulunmuştur. Konuşmacının verdiği ilk örnek grubuna, her bir parametre için çoklu ayırma analizine uygulanmıştır. Böylelikle, her bir parametre için ayırma fonksiyonları elde edilmiştir.

Bu aşamadan sonra, öncelikle otokorelasyonlar için elde edilen ayırma fonksiyonlarının katsayıları kullanılarak, kanonik vektör yaklaşımına göre sınıflandırma metodu, konuşmacıdan alınan ünlüleri birer gruba sınıflandırmak için kullanılmıştır. Sınıflandırma her bir frame için ayrı ayrı yapıldığından, ünlüyü oluşturan frameler hangi grupta daha fazla yığılma gösteriyorsa ünlü o gruba sınıflandırılmaktadır.

Diğer parametreler için de aynı şekilde sınıflandırma gerçekleştirilir. Ancak doğaldır ki aynı gözlem vektörleri için değişik para-

metreler kullanılarak yapılan sınıflandırma farklı tanıma oranları vermektedir.

Elde edilen sonuçlara göre, LPC ve otokorelasyon parametreleri ile yapılan sınıflandırmada diğerlerine göre daha iyi sonuçlar elde edilmektedir. Ancak burada tamamen kişiye bağımlı bir tanıma söz konusudur.

## 6. SONUÇLAR

Bu tez çalışmasında, daha çok istatistiki uygulamalarda kullanılan çoklu ayırma analizi tekniğinin konuşmayı tanıma konusunda çeşitli uygulamaları gerçekleştirilmiştir.

Öncelikle ses bilgisinden elde edilen parametrik modellerin önem dereceleri elde edilmeye çalışılmıştır. Analizlerden elde edilen sonuçlar şu şekilde sıralanabilir:

1) Maksimum genlikler, enerji seviyeleri ve sıfır geçiş sayıları arasında yapılan önem sıralamasında, ilk sıraları enerji seviyeleri ve sıfır geçiş sayıları paylaşmaktadır. Önem dereceleri açısından ikisinin arasındaki fark yok denecek kadar azdır. Ancak maksimum genlikler grupları ayırmada önemsiz olarak gözlenmiştir.

2) LPC, cepstral ve LAR parametreleri arasında yapılan önem sıralamasında, LPC parametreleri ilk sıraları paylaşmaktadır. Grupları ayırmadaki başarıları açısından LAR parametreleri ikinci sıradadır. Ancak bir çok kişi için cepstral parametreleri bazı durumlarda üst sıralara yerleşebilmektedir. Bu nedenle LPC parametrelerinin grupları ayırmada başarılı olabileceğini, LAR ve cepstral parametrelerinin tek başlarına kullanıldıklarında grupları ayırmada başarısız olacaklarını söyleyebiliriz.

3) Otokorelasyon, LPC ve formant frekansları arasında yapılan önem sıralamasında, ilk sıraları otokorelasyonlar ve LPC'ler paylaşmaktadır. Formant frekanslarının grupları ayırmada genelde başarısız oldukları görülmüştür.

Formant frekanslarının grupları ayırmadaki başarısızlığı oldukça şaşırtıcıdır. Ancak oniki LPC katsayısından oluşturulan frekans spektrumuna bakıldığında, bazı ünlüler için özellikle üçüncü veya ikinci formant frekansının diğer formantlara karıştığı gözlenmiştir. Ünlü harfler periyodik sinyaller olduklarından frekans spektrumlarından elde edilen formantlar genelde aynı bölgelerde çıkması beklenmektedir. Ancak LPC'lerden elde edilen frekans spektrumlarında elde edilecek formant sayısı analizde kullanılan LPC parametre sayısına bağlıdır. Bu durumda LPC parametre sayısı çoğaltılırsa, bundan elde

edilen spektrumda, bazen birbirlerine karışan formantlar yakalanabilir.

Çoklu ayırma analizinin sınıflandırma işleminde kullanılması durumunda otokorelasyon ve LPC parametreleri ile modellenen ünlüler diğerlerine göre daha başarılı şekilde sınıflandırılmıştır. Sınıflandırma yalnızca kişiye bağımlı özellikler taşımaktadır. Kişiden bağımsız tanıma konusunda herhangi bir çalışma yapılmamıştır.

**KAYNAKLAR DİZİNİ**

- Anderson, T.W. and Sclove, S.L., 1978, An introduction to the Statistical Analysis of Data, Houghton Mifflin Com., Boston, 704 p.
- Çakmak, Z., 1986, Çoklu Ayırma ve Sınıflandırma Analizi, Doktora Tezi, A.Ü. Sosyal Bilimler Enstitüsü, Eskişehir.
- Lachenbruch, P.A, 1975, Discriminant Analysis, Hafner Press, London.
- Overal, J.E. and Klett, C.J., 1972, Applied Multivariate Analysis, McGraw-Hill Book Comp., New York.
- William, H.P. and Brain P.F., 1988, Numerical Recipes in C, Cambridge University Press, U.S.A.
- Yurtcan, E., 1992, Türkçe Alfabenin Discriminant Analizi Yöntemi ile Sınıflandırılması, Diploma Tezi, A.Ü. Müh. Mim. Fak. Elektrik Elektronik Müh. Böl., Eskişehir.

E K L E R

## EK 1 : Çoklu ayırma analizi programı

```
program discriminant_alanysis;
uses crt,dos;
type
  type_def=(integer_type,real_type);
  tworeal=^arr1;
  onereal=^arr2;
  threereal=^arr3;
  oneint=^arr4;
  two=^arr5;
  arr1=array[1..31,1..31] of real;
  arr2=array[1..31] of real;
  arr3=array[1..31,1..8,1..35] of real;
  arr4=array[1..31] of integer;
  arr5=array[1..31,0..31] of real;
  dz1=array[0..500] of integer;
  dz2=string[80];
const
  epsi=1E-6;
  type size=array[0..1] of integer=(2,6);
  type
    integer_array=array[1..1,1..1] of integer;
    real_array=array[1..1,1..1] of real;
    point_int=^integer_array;
    point_rl =^real_array;
var
  x:threereal;
  x_gen_ort,eigval:onereal;
  x_kis_ort:tworeal;
  nvar,nlat,ngps,nobs,p,g,nt,z,y:integer;
  s1:tworeal;
  n:oneint;
  t,w,c,r,b:tworeal;
  bt:two;
  filename:dz2;
  vecnum,a,xg,length,sj,k1,p1,i1:integer;
  j1:real;
  ek,filen,kom,files:array[1..17] of string;
  ek2:string;
  i,k,j:integer;
procedure new_array(var pointer;element_type:type_def;array_size:integer);
var
  any_pointer:^integer absolute pointer;
begin
  getmem(any_pointer,type_size[integer(element_type)]*array_size);
end;
procedure dispose_array(var pointer;element_type:type_def;array_size:
  integer);
var
  any_pointer:^integer absolute pointer;
begin
  freemem(any_pointer,type_size[integer(element_type)]*array_size);
end;
{ Ses parametreleri ayırıcı değişkenler olarak dosyadan okunuyor}
procedure read_frm1(var x:threereal;filename:dz2;var length:integer;
  k:integer);
label ret,se1,se2,se3,se4,se5;
var
  dptr:text;
  isim1,isim2,isim3,isim4,frm_no1:string[80];
  b:string[1];
  a,c,d:byte;
```



```

code,i,m:integer;
isim5:string[5];
begin
assign(dptra,'c:\umit\doc\'+filename);
  {$I-}
  reset(dptra);
  if ioresult<>0 then
    begin
      write(' Un able to open ',filename);
      halt;
    end;
  {$I+}
  readln(dptra,isim1);
  readln(dptra,isim2);
while not eof(dptra) do
  begin
    readln(dptra,isim4);
    if (isim4='AUTOCORRELATIONS:') then goto ret;
  end;
ret:
  length:=1;

se3:  for i:=1 to 13 do
      begin
        readln(dptra,isim4);
        if (isim4>'LPCs:') then
          begin
            val(isim4,x^[i,k,length],code);
            goto sel;
          end;
        goto se2;
      end;
sel:
  end;
  length:=length+1;
  goto se3;
se2:  length:=length-1;
{
  for m:=1 to length do
  for i:=1 to 12 do
    begin
      readln(dptra,isim4);
      val(isim4,x^[i,k,m],code);
    end;}
{ close(dptra);
assign(dptra,'c:\umit\doc\'+ek2+'.lpf');
reset(dptra);
  length:=0;
  while not eof(dptra) do
    begin
      readln(dptra,isim4);
      isim5:=isim4;
      if (isim5='Frame') then begin writeln(isim5);
      i:=26;length:=length+1;goto se5; end;
      if (isim5<>'Frame') then
        begin
          val(isim4,x^[i,k,length],code);
          i:=i+1;
        end;
      se5: end; }
    close(dptra);
  end;

{ Grup içi ortalamalarla gruplar arası ortalama hesaplanıyor. }
procedure ort(var x kis ort:tworeal;var x_gen_ort:onereal;x:threereal;
              p,g,nE:integer;n:oneint);
var
ort,genort:real;
i,k,m:integer;

```

```

begin
  for i:=1 to p do
    begin
      genort:=0;
      for k:=1 to g do
        begin
          ort:=0;
          for m:=1 to n[k] do
            ort:=ort+x[i,k,m];
            x_kis_ort[i,k]:=ort/n[k];
            genort:=genort+ort;
          end;
          x_gen_ort[i]:=genort/nt;
        end;
      end;
    end;
  { WC-1BC matrisinin özdeğer ve özvektörleri hesaplanıyor. }
  procedure jacob(var c,r:tworeal;b:tworeal;p:integer;epsi:real);
  var
    v,e,d:arr2;
    vo,u,uf,alfa,beta,h,s,f:real;
    l,j,k,l,m,p1,q,n:integer;
  label l1;
  label l2;
  label l3;
  label l4;
  label l5;
  begin
    vo:=0;
    n:=p;
    for i:=1 to n do
      begin
        for j:=1 to n do
          begin
            if (i=j) then goto l1;
            vo:=vo+c[i,j]*c[i,j];
          l1: end;
        end;
        u:=sqrt(vo)/n;
        uf:=epsi*u;
        { for i:=1 to n do
          begin
            for j:=1 to n do
              write(c[i,j], ' ');
              writeln(' ');
            end;}

        for i:=1 to n do
          begin
            for j:=1 to n do
              begin
                r[i,j]:=0;
                if (i=j) then r[i,j]:=1;
              end;
            end;
          end;
        15: for l:=1 to n-1 do
          begin
            for m:=l+1 to n do
              begin
                if (abs(c[l,m])<u) then goto l2;
                p1:=l; q:=m;
                for i:=1 to n do
                  begin
                    v[i]:=c[p1,i];
                    d[i]:=c[i,p1];
                    e[i]:=r[i,p1];
                  end;
                f:=c[p1,p1];
                alfa:=0.5*(c[p1,p1]-c[q,q]);
                beta:=sqrt(c[p1,q]*c[p1,q]+alfa*alfa);
              end;
            end;
          end;
        end;
      end;
    end;
  end;

```

```

h:=sqrt(0.5+0.5*abs(alfa)/beta);
s:=-0.5*alfa*c^[p1,q]/(beta*h*abs(alfa));
for j:=1 to n do
begin
  if (j=p1) then goto l3;
  if (j=q) then goto l3;
  c^[p1,j]:=h*c^[p1,j]-s*c^[q,j];
  c^[q,j]:=s*v[j]+h*c^[q,j];
  c^[j,p1]:=h*c^[j,p1]-s*c^[j,q];
  c^[j,q]:=s*d[j]+h*c^[j,q];
13:  r^[j,p1]:=h*r^[j,p1]-s*r^[j,q];
  r^[j,q]:=s*e[j]+h*r^[j,q];
end;
c^[p1,p1]:=h*h*c^[p1,p1]+s*s*c^[q,q]-2*h*s*c^[p1,q];
c^[q,q]:=s*s*f+h*h*c^[q,q]+2*h*s*c^[p1,q];
c^[p1,q]:=0; c^[q,p1]:=0;
12: end;
end;
if (u<=uf) then goto l4;
u:=u/n;
goto l5;
14: end;

```

{  $W_C^{-1}B_C$  'ye Choleski parçalama metodu uygulanıyor. }

procedure choles(var c,b:tworeal;w:tworeal;p:integer);

```

var
  l,ll:arr1;
  i,j,k,n,m,s:integer;
topl:=0;
begin
  n:=p;
  for i:=1 to n do
  begin
    for j:=1 to n do
    begin
      ll[i,j]:=0;
      ll[i,j]:=0;
    end;
  end;
  topl:=0;
  l[[1,1]]:=sqrt(w[[1,1]]);
  for i:=2 to n do
  begin
    for j:=1 to i-1 do
    begin
      topl:=0;
      for k:=1 to j-1 do
      begin
        topl:=topl+l[i,k]*l[j,k];
      end;
      l[i,j]:=(w[i,j]-topl)/l[j,j];
    end;
    j:=j+1;
    topl:=0;
    for k:=1 to i-1 do
    begin
      topl:=topl+sqrt(l[i,k]);
    end;
    l[[i,i]]:=sqrt(w[i,i]-topl);
    for j:=i to n do
    begin
      l[i-1,j]:=0;
    end;
  end;
  {for m:=1 to n do
  begin
    for s:=1 to n do
      write(l[m,s]);
      writeln('');
    end;
  }
end;

```

```

    writeln('');)
end;
ll[1,1]:=1/ll[1,1];
for i:=2 to n do
begin
  ll[i,i]:=1/ll[i,i];
  for j:=i-1 downto 1 do
  begin
    topl:=0;
    for k:=j+1 to i do
      topl:=topl+ll[i,k]*l[k,j];
    ll[i,j]:=-topl/ll[j,j];
  end;
end;
for i:=1 to n do
begin
  for j:=1 to n do
  begin
    c^[i,j]:=0;
    for k:=1 to n do
      c^[i,j]:=c^[i,j]+ll[i,k]*b[k,j];
    end;
  end;
end;
for i:=1 to n do
begin
  for j:=1 to n do
  begin
    c^[i,j]:=c^[i,j];
  end;
end;
for i:=1 to n do
begin
  for j:=1 to n do
  begin
    c^[i,j]:=0;
    for k:=1 to n do
      c^[i,j]:=c^[i,j]+ll[i,k]*b[k,j];
    end;
  end;
end;
{ for m:=1 to n do
begin
  for s:=1 to n do
    write(c^[m,s], ' ');
  writeln('');
end; writeln('');}
end;

{ özdegerler elde ediliyor. }

procedure eigen_val(var eigval: onereal; c: tworeal; var r: tworeal;
p: integer);

var
  i, ara, j, k, l: integer;
  deg_var: boolean;
  s: real;
  sl: array[1..31] of real;
label l8;
begin
  l:=p;
  for i:=1 to p do
  begin
    if (c^[i,i]<=1E-6) then begin c^[i,i]:=0; l:=l-1; end;
    eigval[i]:=c^[i,i];
  end;
  vecnum:=l;
  for i:=1 to p-1 do
  begin

```

```

for j:=i+1 to p do
begin
if (eigval^[i]<eigval^[j]) then
begin
s:=eigval^[i];
eigval^[i]:=eigval^[j];
eigval^[j]:=s;
for k:=1 to p do
begin
s1[k]:=r^[k,i];
r^[k,i]:=r^[k,j];
r^[k,j]:=s1[k];
end;
end;
end;
end;
for j:=1 to p do
begin
for i:=1 to p do
begin
if (abs(c^[i,i])=eigval^[i]) and (c^[i,i]<0) then begin
eigval^[j]:=-1*eigval^[i];
goto 18;
end;
end;
end;
18: end;
end;

```

```

procedure find_lamda(i,k:integer;eigval:onereal);
var
dosya:text;
begin
for k:=1 to g do
begin
assign(dosya,'c:\umit\'+filen[sj]+'.ltp');
rewrite(dosya);
i:=0;
for i:=1 to vecnum do
begin
writeln(dosya,eigval^[i]);
end;
close(dosya);
end;
end;
end;

```

{ Özvektörler elde ediliyor. }

```

procedure eigen_vec(var r:tworeal;b:tworeal;p:integer);
var
s1:arr1;
i,j,k:integer;
begin
for i:=1 to p do
begin
for j:=1 to p do
begin
s1[i,j]:=0;
for k:=1 to p do
begin
s1[i,j]:=s1[i,j]+b^[i,k]*r^[k,j];
end;
end;
end;
end;
for i:=1 to p do
begin
for j:=1 to p do
r^[i,j]:=s1[i,j];
end;
end;
end;

```

{ Ayırıcı değişkenler önem derecelerine göre sıralanıyorlar. }

```
procedure sort(r:tworeal);
var t:real;
    j,k,s,z,m:integer;
    y:array[1..31] of integer;
    dks:text;
begin
for m:=1 to 3 do
begin
for s:=1 to p do
begin
y[s]:=s;
end;

for j:=1 to p-1 do
for k:=j+1 to p do
if (abs(r^[j,m])<abs(r^[k,m])) then begin
t:=r^[j,m];
z:=y[j];
r^[j,m]:=r^[k,m];
y[j]:=y[k];
r^[k,m]:=t;
y[k]:=z;
end;
writeln(filen[sj],'.vlp');
for j:=1 to p do
begin
writeln(j,'.sirada ',y[j],'.ayirici degisken');
end;

assign(dks,'c:\umit\for.sr');
{$I-}
{ if (sj=1) then rewrite(dks) else append(dks);}
append(dks);

if ioresult<>0 then rewrite(dks);
{$I+}
writeln(dks,filen[sj],'.vlp');
for j:=1 to p do
begin
writeln(dks,j,'.sirada ',y[j],'.ayirici degisken');
end;

close(dks);
end;
end;

procedure fonksiyon_deger;
var
k,j:integer;
dosya:text;
f:array[1..8] of real;

begin
for k:=1 to g do
begin
i[k]:=0;
begin
for j:=1 to p do
begin
f[k]:=f[k]+r^[j,1]* x_kis_ort^[j,k];
end;
end;
end;

assign(dosya,'c:\umit\'+'filen[sj]+' .fff');
rewrite(dosya);
for k:=1 to g do
begin
writeln(dosya,f[k]);
end;
end;
```

```

        close(dosya);
    end;

procedure find_eigvec(var r:tworeal);
var
dosya:text;
begin
    assign(dosya,'c:\umit\'+'+filen[sj]+'vfr');
    rewrite(dosya);
    for i:=1 to vecnum do
begin
    writeln(dosya,'sutun(',i,')');
    for j:=1 to p do
begin
    writeln(dosya,r^[j,i]);
end;
end;
end;
close(dosya);
end;

{ WC hesaplanıyor. }

procedure w_cov(var w:tworeal;p,g,nt:integer);
var
i,j:integer;
begin
for i:=1 to p do
begin
for j:=1 to p do
w^[i,j]:=w^[i,j]/(nt-g);
end;
end;
end;

{ TC hesaplanıyor. }

procedure t_cov(var t:tworeal;p,g,nt:integer);
var
i,j:integer;
begin
for i:=1 to p do
begin
for j:=1 to p do
t^[i,j]:=t^[i,j]/(nt-1);
end;
end;
end;

{ BC hesaplanıyor. }

procedure b_cov(var b:tworeal;t,w:tworeal;p,nt:integer);
var
i,j:integer;
begin
for i:=1 to p do
begin
for j:=1 to p do
b^[i,j]:=b^[i,j]/(g-1);
end;
end;
end;

procedure w_matrice(var w:tworeal;p,g:integer;n:oneint;x:threereal;
x_gen_ort:oneal;x_kis_ort:tworeal);
var
i,j,k,m:integer;
deg,deger1:real;
begin
for i:=1 to p do
begin
for j:=1 to p do
begin
deg:=0;
for k:=1 to g do

```

```

begin
  deger1:=0;
  for m:=1 to n[k] do
    begin
      deger1:=deger1+(x^[i,k,m]-x_kis_ort^[i,k])
        *(x^[j,k,m]-x_kis_ort^[j,k]);
    end;
  deg:=deg+deger1;
end;
w^[i,j]:=deg;
end;
end;
end;

```

{ T matrisi hesaplanıyor. }

```

procedure t_matrice(var t:tworeal;p,g:integer;n:oneint;x:threereal;
  x_gen_ort:oneoreal;x_kis_ort:tworeal);

```

```

var
  i,j,k,m:integer;
  deg,deger1:real;
begin
  for i:=1 to p do
    begin
      for j:=1 to p do
        begin
          deg:=0;
          for k:=1 to g do
            begin
              deger1:=0;
              for m:=1 to n[k] do
                begin
                  deger1:=deger1+(x^[i,k,m]-x_gen_ort^[i])
                    *(x^[j,k,m]-x_gen_ort^[j]);
                end;
              deg:=deg+deger1;
            end;
          t^[i,j]:=deg;
        end;
      end;
    end;
end;

```

{  $W_C^{-1}$  hesaplanıyor. }

```

procedure inv_w(var a:arr1;p:integer;w:tworeal);

```

```

var
  g:array[1..25,1..50] of real;
  i,j,k:integer;
  b,c:real;
begin
  for i:=1 to p do
    begin
      for j:=1 to p do
        begin
          g[i,j]:=w^[i,j];
          g[i,j+p]:=0;
        end;
      g[i,i+p]:=1;
    end;
    for i:=1 to p do
      begin
        b:=g[i,1];
        for j:=1 to 2*p do
          g[i,j]:=g[i,j]/b;
        for j:=1 to p do
          begin
            if (j>i) then
              begin
                c:=g[j,1];
                for k:=1 to 2*p do

```



```

        g[j,k]:=g[j,k]-c*g[i,k];
    end;
end;
end;
for i:=1 to p do
begin
    for j:=p+1 to 2*p do
        a[i,j-p]:=g[i,j];
    end;
end;
end;

procedure class(var bt:two;a:arr1;p,g:integer;x_kis_ort:tworeal);
var
i,j,k:integer;
b:real;
begin
    for k:=1 to g do
        begin
            for i:=1 to p do
                begin
                    b:=0;
                    for j:=1 to p do
                        begin
                            b:=b+a[i,j]*x_kis_ort^[j,k];
                        end;
                    bt^[k,i]:=(nt-g)*b;
                end;
            end;
            for k:=1 to g do
                begin
                    b:=0;
                    for j:=1 to p do
                        b:=b+bt^[k,j]*x_kis_ort^[j,k];
                    bt^[k,0]:=-0.5*b;
                end;
            end;
        end;
end;

procedure print_t(t:tworeal;p:integer);
var
h,v:byte;
i,j:integer;
begin
    h:=1;
    v:=1;

    for i:=1 to p do
        begin
            h:=1;
            for j:=1 to p do
                begin
                    Gotoxy(h,v);
                    write(t^[i,j]);
                    h:=h+20;
                end;
            v:=v+1;
        end;
    end;
end;

{ B matrisi hesaplanıyor. }

procedure b_matrice(var b:tworeal;t,w:tworeal;p:integer);
var
i,j:integer;
begin
    for i:=1 to p do
        begin
            for j:=1 to p do
                b^[i,j]:=t^[i,j]-w^[i,j];
            end;
        end;
    end;
end;

```

```

procedure print_w(w:tworeal;p:integer);
var
  h,v:byte;
  i,j:integer;
begin
  h:=1;
  v:=4;

  for i:=1 to p do
    begin
      h:=1;
      for j:=1 to p do

        begin
          gotoxy(h,v);
          write(w[i,j]);
          h:=h+20;
        end;
        v:=v+1;
      end;
    end;
end;

```

```

procedure print_b(b:tworeal;p:integer);
var
  h,v:byte;
  i,j:integer;
begin
  h:=1;
  v:=7;
  for i:=1 to p do
    begin
      h:=1;
      for j:=1 to p do

        begin
          gotoxy(h,v);
          write(b[i,j]);
          h:=h+20;
        end;
        v:=v+1;
      end;
    end;
end;

```

```

procedure print_a(a:arr1;p:integer);
var
  h,v:byte;
  i,j:integer;
begin
  h:=1;
  v:=10;

  for i:=1 to p do
    begin
      h:=1;
      for j:=1 to p do
        begin
          gotoxy(h,v);
          write(a[i,j]);
          h:=h+20;
        end;
        v:=v+1;
      end;
    end;
end;

```

```

procedure print_bt(bt:two;p,g:integer);
var
  h,v:byte;
  i,j:integer;
begin
  h:=1;
  v:=12;

```

```

for i:=1 to p do
begin
h:=1;
for j:=1 to p do

begin
gotoxy(h,v);
write(bt^[i,j]);
h:=h+20;
end;
v:=v+1;
end;
gotoxy(1,v+1); write(bt^[1,0]);
gotoxy(1,v+2); write(bt^[2,0]);

```

```
end;
```

```

procedure read_files(var x:threereal;var length,nt,sj:integer;
var n:oneint);

```

```
var
```

```

i,k,j:integer;
begin

```

```

ek[1]:= 'a';      ek[5]:= 'o';
ek[2]:= 'e';      ek[6]:= 'oo';
ek[3]:= 'i';      ek[7]:= 'u';
ek[4]:= 'ii';     ek[8]:= 'uu';

```

```

filen[1]:= 'salih';
filen[2]:= 'sel';
filen[3]:= 'ender';
filen[4]:= 'metind';
filen[5]:= 'umit';
filen[6]:= 'gulay';
filen[7]:= 'muam';
filen[8]:= 'omer';
filen[9]:= 'murato';
filen[10]:= 'metin';
filen[11]:= 'nazan';
filen[12]:= 'idil';
filen[13]:= 'zeyn';
filen[14]:= 'munev';

```

```

kom[1]:= '.doc';      kom[5]:= '.p';
kom[2]:= '.amp';     kom[6]:= '.l';
kom[3]:= '.int';     kom[7]:= '.c';
kom[4]:= '.a';       kom[8]:= '.f';

```

```
for i:=1 to 31 do
```

```

begin
for j:=1 to 8 do
begin
for k:=1 to 35 do
x^[i,j,k]:=0;
end;
end;
nt:=0;

```

```

for k:=1 to 8 do
begin
ek2:=filen[sj]+ek[k];
filename:=ek2+kom[1];
read_frm1(x,filename,length,k);
writeln(filename);
n^[k]:=length;
writeln(length);
nt:=nt+length;
end;

```

```
end;
```

```

procedure find_func(r:tworeal);
var
  dosya:text;
  m,j,k:integer;
  jk:real;
begin
  for k:=1 to g do
  begin
    assign(dosya,'c:\umit\doc\'+filen[sj]+ek[k]+'fal');
    rewrite(dosya);
    for m:=1 to n^[k] do
    begin
      jk:=0;
      for j:=1 to 9 do
      begin
        jk:=jk+r^[j,1]*x^[j,k,m];
      end;
      writeln(dosya,jk);
    end;
    close(dosya);
  end;
end;

```

{ Normalizasyon işlemi gerçekleştiriliyor. }

```

procedure normalization(var x:threereal);
var
  i,k,m:integer;
  top:array[1..2,1..8,1..35]of real;
begin
  for k:=1 to g do
  begin
    for m:=1 to n^[k] do
    begin
      top[1,k,m]:=0;
      top[2,k,m]:=0;
      for i:=1 to 13 do
      begin
        top[1,k,m]:=top[1,k,m]+sqr(x^[i,k,m]);
      end;
      { for i:=14 to 25 do
      begin
        top[2,k,m]:=top[2,k,m]+sqr(x^[i,k,m]);
      end; }

    end;
  end;
  for k:=1 to g do
  begin
    for m:=1 to n^[k] do
    begin
      for i:=1 to 13 do
      begin
        x^[i,k,m]:=x^[i,k,m]/sqrt(top[1,k,m]);
        {writeln(x^[i,k,m]);}
      end;
      { for i:=14 to 25 do
      begin
        x^[i,k,m]:=x^[i,k,m]/(top[2,k,m]);
      end; }
    end;
  end;
end;

```

end;

{ Sınıflandırma işlemi gerçekleştiriliyor. }

```

procedure classific;
var
  fark vec:array[1..14] of real;
  j,m,I,k,k1,i1,fa,fb:integer;

```

```

sayi:string[1];
st:string;
fc,fd,son_vec1,ka,kl:real;
index:array[1..8,1..3,1..20] of integer;
begin
  files[1]:= 'a';
  files[2]:= 'e';
  files[3]:= 'i';
  files[4]:= 'ii';
  files[5]:= 'o';
  files[6]:= 'oo';
  files[7]:= 'u';
  files[8]:= 'uu';
for l:=1 to 8 do
begin
  for j:=1 to 3 do
  begin
    str(j,sayi);
    filename:='unit'+files[l]+sayi+'.doc';
    k:=1;
    read_frm1(x,filename,length,k);
    writeln(filename);
    for m:=1 to length do
    begin
      fd:=1000000;
      for il:=1 to g do
      begin
        fc:=0;
        for fa:=1 to vecnum do
        begin
          for kl:=1 to p do
          begin
            fark_vec[kl]:=x^[kl,1,m]-x_kis_ort^[kl,il];
            end;
            son_vec1:=0;
            for fb:=1 to p do
            son_vec1:=son_vec1+r^[fb,fa]*fark_vec[fb];
            son_vec1:=son_vec1*son_vec1;
            fc:=fc+son_vec1;
            end;
            if fc<=fd then begin fd:=fc;index[l,j,m]:=il; end;
          end;
        end;
      end;
    end;
  for l:=1 to 8 do
  begin
    for j:=1 to 3 do
    begin
      kl:=0;
      for m:=1 to length do
      if index[l,j,m]=1 then kl:=kl+1;
      ka:=kl/length;
      if ka>=0.5 then st:='olumlu' else st:='olumsuz';
      writeln(l,' inci grup ',j,' inci ornek icin sonuc : ',st);
      end;
    end;
  end;
end;

```

```

{ MAIN PROGRAM }
begin
  { first group (A) }
  g:=8;
  p:=13;
  clrscr;
  new(x_kis_ort);
  new(x_gen_ort);
  new(x);
  new(eigval);

```

```

new(s1);
new(n);
new(t);
new(w);
new(c);
new(r);
new(b);
new(bt);

for sj:=5 to 5 do
begin
  read_files(x,length.nt,sj,n);
  { scaI; }
  {normalization(x);}
  ort(x_kis_ort,x_gen_ort,x,p,g,nt,n);
  w_matrice(w,p,g,n,x,x_gen_ort,x_kis_ort);
  t_matrice(t,p,g,n,x,x_gen_ort,x_kis_ort);
  b_matrice(b,t,w,p);
  b_cov(b,t,w,p,nt);
  w_cov(w,p,g,nt);
  { E_cov(t,p,g,nt);}
  clrscr;
  choles(c,b,w,p);
  jacob(c,r,b,p,epsi);
  for z:=1 to p do
  begin
    for y:=1 to p do
      write(r^[z,y], ' ');
      writeln(' ');
    end;writeln(' ');
    eigen_vec(r,b,p);
    eigen_val(eigval,c,r,p);
  classific;
  { find lamda(i,k.eigval);
  find eigvec(r);
  for z:=1 to p do
  begin
    writeln(eigval^[z]);
  end;
  find func(r);
  writeln(' ');
  for z:=1 to p do
  begin
    for y:=1 to p do
      write(r^[z,y]:1:4, ' ');
      writeln(' ');
    end;}
  { print t(t,p);
  print w(w,p);
  print b(b,p);
  }
  { sort(r);
  fonksiyon deger;
  for k:=1 to 8 do
  writeln('k=',k);
  for xg:=1 to n[k] do
  writeln('x[' ,i,k,xg, ']=' ,x^[14,k,xg]);)
  end;
  end.

```

## Ek 2 : Seseli harflerin parametrelerini hesaplayan program

```
program parameters;
uses tpcrt,
    tpstring,
    tpedit,
    graph,
    tpmenu,
    crt;
const
  Color1 : MenuColorArray = ($0E, $6f, $03, $6f, $0b, $6F);
  Color2 : MenuColorArray = ($0E, $2E, $03, $1E, $0B, $0E);
  Frame1 : FrameArray = 'Z@?YD3';
  Frame2 : FrameArray = 'IH;<M: ';
  n=128;

var
  f:text;
  fname:string[20];
  frnum,pof,inf:string[80];
  si,flx,m,zc,j,i,fn,sn,t:integer;
  data:array[0..45,0..127] of real;
  k,r:array[0..45,0..30] of real;
  lpc:array[0..45,1..30] of real;
  tot,w:real;
  zero,ints,maxa,auto,llpc,ceps,llar,alll,bool:integer;
  chh,ch:char;
  main:menu;
  key:menukey;
  mi:integer;

procedure InitMenu(var M : Menu);
begin
  M := NewMenu([], nil);
  SubMenu(29,8,24,Vertical,Frame2,Color1,'');
  MenuItem('Load Frames',1,1,1,'Read frame file from disk. ');
  MenuItem('Draw',2,1,2,'Draw the speech signal. ');
  MenuItem('Parameters',3,1,0,'Parameters of syllables. ');
  SubMenu(37,11,24,Vertical,Frame1,Color1,'');
  MenuItem('Number of ZeroCrossings',1,1,15,'
    No. of ZC in each frame. ');
  MenuItem('Intensity',2,1,5,'Intensity of frame in(dB) ');
  MenuItem('Maximum Amplitude',3,1,6,'Maximum
    absolute amplitude in frame ');
  MenuItem('Autocorrelation',4,1,7,'Autocorrelation coefficients ');
  MenuItem('LPC',5,1,8,'Linear Predictive Coefficients ');
  MenuItem('Cepstral',6,1,9,'Cepstral coefficients ');
  MenuItem('LAR',7,1,10,'LOG-AREA ratios ');
  MenuItem('All of them',8,1,25,'All parameters in one time. ');
  MenuItem('SETUP',9,1,11,'Change number of parameters ');

  PopSublevel;
  MenuItem('Quit',4,1,4,'Return to DOS. ');
  PopSublevel;

  ResetMenu(M);
end;

procedure setup;
begin
  clrscr;
```

```

write('No. of parameters (between 1 .. 12) : ');
read(m);readln;
end;

procedure initial_file;
begin
  if fname='' then exit;
  assign(f,fname+'.doc');
  {$I-}
  rewrite(f);
  {$I+}
  if ioresult<>0 then exit;
  writeln(f,pof);
  writeln(f,inf);
  close(f);
end;

procedure read_frames;
begin
  assign(f,fname+'.frm');
  {$I-}
  reset(f);
  {$I+}
  if ioresult=0 then
  begin
    fn:=0;
    zero:=0;
    ints:=0;
    maxa:=0;
    while not(eof(f)) do
    begin
      if fn=0 then
      begin
        readln(f,pof);
        readln(f,inf);
        end;
        readln(f,frnum);
        for sn:=0 to 127 do
          readln(f,data[fn,sn]);
          fn:=fn+1;
        end;
        close(f);
      end else
      begin
        clrscr;
        gotoxy(25,10);
        writeln('File Not Found');
        sound(250);
        delay(600);
        nosound;
        exit;
      end;
    end;
  end;

procedure hamming_window;
begin
  for i:=0 to fn-1 do
  begin
    for sn:=0 to 127 do
    begin
      w:=0.54-0.46*cos(2*pi*sn/(n-1));
      data[i,sn]:=w*data[i,sn];
    end;
  end;
end;

procedure preemphasis;
begin
  for i:=0 to fn-1 do
  begin
    for sn:=1 to 127 do
      data[i,sn]:=data[i,sn]-0.9375*data[i,sn-1];
  end;
end;

```



```

    end;
end;

procedure write_data(d:real);
begin
    if bool=1 then writeln(f,d:4:0) else writeln(f,d);
end;
procedure errormessage;
begin
    ClrScr;
    GoToXY(25,10);
    Writeln('You have to load frames ! ');
    Sound(500);
    Delay(500);
    NoSound;
    si:=1;
    Exit;
end;
procedure h(hh:string);
begin
    {$I-}
    append(f);
    {$I+}
    If IoResult<>0 then errormessage;
    if si=1 then exit;
    writeln(f,hh);
end;

procedure drawspeechsignal;
var
    ev, smp,gd,gm:integer;
    maxx,maxy:word;
    chh:char;
    xxl:integer;
    lxx:real;

begin
    gd:=detect;
    initgraph(gd,gm,'');
    maxx:=getmaxx;
    maxy:=getmaxy;
    setviewport(0,0,maxx,maxy,clipon);
    moveto(10,10);
    outtext(inf);

    rectangle(0,0,maxx,maxy div 2);
    moveto(0,maxy div 2);
    smp:=trunc((fn-1)*128/maxx)+1;
    xxl:=0;
    ev:=128 div smp;
    lxx:=0;
    {
        for i:=1 to fn-1 do
            for j:=0 to 127 do
                if abs(data[i,j])>lxx then lxx:=abs(data[i,j]);
            str(lxx:4:0,inf);
            moveto(20,20);

            outtext(inf); }
    moveto(0,maxy div 2);
    for i:=1 to fn-1 do
        begin
            for j:=0 to 127 do
                begin
                    lineto(trunc(xxl/smp),trunc(data[i,j])
                        div (maxy div 20)+maxy div 4 );

                    inc(xxl);
                end;
            end;
        lineto(xxl,maxy div 4);
        lineto(maxx-1,maxy div 4);

```

```

chh:=readkey;
cleardevice;
closegraph;
end;

```

```

procedure no_of_zerocrosses;

```

```

begin
  if zero=1 then begin
    clrscr;
    gotoxy(25,10);
    writeln('It's done before'+chr(7));
    exit;
  end;
  zero:=1;
  bool:=1;
  h('NUMBER OF ZEROCROSSINGS:');
  if si=1 then exit;
  for i:=0 to fn-1 do
    begin
      zc:=0;
      for sn:=0 to 126 do
        begin
          if (data[i,sn]>=0) and (data[i,sn+1]<0) then zc:=zc+1;
          if (data[i,sn]<0) and (data[i,sn+1]>=0) then zc:=zc+1;
        end;
      write_data(zc);
    end;
  bool:=0;close(f);
end;

```

```

procedure autocorrelation;

```

```

begin
  if flx>0 then h('AUTOCORRELATIONS:');
  if si=1 then exit;
  if mi=0 then
    begin
      preemphasis;
      Hamming_Window;
      mi:=1;
    end;
  for j:=0 to fn-1 do
    begin
      for i:=0 to m do
        begin
          r[j,i]:=0;
          for sn:=0 to 127-i do
            r[j,i]:=r[j,i]+data[j,sn]*data[j,sn+i];
          if flx>0 then write_data(r[j,i]);
        end;
      end;if flx>0 then close(f);
    end;
end;

```

```

procedure lpc_coefficients;

```

```

var
  e:array[0..12] of real;
  lp:array[0..44,1..12,1..12] of real;
  x:integer;
begin
  if flx> 1 then h('LPCs:');
  if si=1 then exit;
  for t:=0 to fn-1 do
    begin
      e[0]:=r[t,0];
      k[t,1]:=-r[t,1]/r[t,0];
      e[1]:=(1-sqr(k[t,1]))*e[0];
      for i:=1 to m do
        begin
          tot:=0;
          if i>1 then
            begin

```

```

        for x:=1 to i-1 do
            tot:=tot+lp[t,x,i-1]*r[t,i-x];
            k[t,i]:=-(r[t,i]+tot)/e[i-1];
        end;
        lp[t,i,i]:=k[t,i];
        if i>1 then
            begin
                for j:=1 to i-1 do
                    lp[t,j,i]:=lp[t,j,i-1]+k[t,i]*lp[t,i-j,i-1];
                e[i]:=(1-sqr(k[t,i]))*e[i-1];
            end;
        end;
        for i:=1 to m do
            begin
                lpc[t,i]:=lp[t,i,m];
                if flx<>1 then write_data(lpc[t,i])
            end;
        end;
        if flx<>1 then close(f);
    end;
end;

```

procedure cepstral;

```

var
    c:array[0..44,0..30] of real;
    kk:integer;
begin
    h('CEPSTRALS:');
    if si=1 then exit;
    for t:=0 to fn-1 do
        begin
            c[t,0]:=r[t,0];
            c[t,1]:=-lpc[t,1];
            for i:=2 to m do
                begin
                    tot:=0;
                    for kk:=1 to i-1 do
                        tot:=tot+((i-kk)/i)*c[t,i-kk]*lpc[t,kk];
                    c[t,i]:=-lpc[t,i]-tot;
                end;
            for i:=0 to m do
                write_data(c[t,i]);
            end;close(f);
        end;
    end;
end;

```

procedure lar;

```

var
    lar:array[0..44,1..30] of real;
begin
    h('LARs:');
    if si=1 then exit;
    for t:=0 to fn-1 do
        begin
            for i:=1 to m do
                begin
                    lar[t,i]:=ln((1+k[t,i])/(1-k[t,i]));
                    write_data(lar[t,i]);
                end;
            end;close(f);
        end;
    end;
end;

```

procedure intensity;

```

const
    log=2.302585093;
var
    inten:array[0..44] of real;

begin
    if ints=1 then begin
        clrscr;
        gotoxy(25,10);
        writeln('It`s done before'+chr(7));
    end;
end;

```

```

exit;
end;ints:=1;
h('INTENSITIES:');
if si=1 then exit;
for i:=0 to fn-1 do
begin
tot:=0;
for sn:=0 to 127 do
tot:=tot+sqr(data[i,sn])/n;
inten[i]:=10*ln(tot);
write_data(inten[i]);
end;
close(f);
end;

```

```

procedure maximum_amplitude;

```

```

var
max_amp:array[0..44] of real;
xdata:array[0..44,0..127] of real;
max,min,mini:real;
begin
if maxa=1 then begin
clrscr;
gotoxy(25,10);
writeln('It`s done before'+chr(7));
exit;
end;maxa:=1;
bool:=1;
h('MAXIMUM AMPLITUDE:');
if si=1 then exit;
for i:=0 to fn-1 do
begin
min:=data[i,0];
for sn:=1 to 127 do
if data[i,sn]<min then min:=data[i,sn];
if min<0 then
begin
mini:=abs(min);
for sn:=0 to 127 do
xdata[i,sn]:=data[i,sn]+mini;
end;
if min>0 then
begin
for sn:=0 to 127 do
xdata[i,sn]:=data[i,sn]-min;
end;
max:=data[i,0];
for sn:=1 to 127 do
if xdata[i,sn]>max then max:=xdata[i,sn];
max_amp[i]:=max;
write_data(max_amp[i]);
end;
bool:=0; close(f);
end;

```

```

procedure all;

```

```

begin
if si=1 then errormessage;
if si=1 then exit;
no of zerocrosses;
intensity;
maximum_amplitude;
flx:=1;
autocorrelation;flx:=0;
lpc_coefficients;
cepstral;
lar;
end;

```

```

{ Main Program Begins....}
begin

```

```

bool:=0;m:=12;flx:=0;mi:=0;si:=1;
clrscr;
initmenu(main);
repeat
key:=menuchoice(main,ch);
erasuremenu(main,false);
case key of

1: begin
gotoxy(25,10);
write('Filename.....');
read(fname);readln;
read frames;
initial_file;
si:=0;
end;
2: DrawSpeechSignal;
15: no_of_zerocrösses;
3: intensity;
6: maximum_amplitude;
7: begin
flx:=1;
autocorrelation;
end;
8: begin
if flx<>1 then
begin
autocorrelation;          {autocor.}
end;
lpc_coefficients;
end;
9: begin
if flx<>1 then
begin
autocorrelation;
flx:=1;                    {CEPSTRAL}
lpc_coefficients;
end;
cepstral;
end;
10: begin
if flx<>1 then
begin
autocorrelation;          {LAR}
flx:=1;
lpc_coefficients;
end;
lar;
end;
11: setup;
25: all;
end;
clrscr;flx:=0;
initmenu(main);
until(ch='m') and (key=4);
end.

```

Ek 3 : Salih EREN ve Umit KUNKCU'den alınan LPC - Cepstral - LAR  
parametreleri için en önemli ayırma fonksiyonları

Salih EREN

1.dereceden önemli ayırma fonksiyonu

1.sirada	7.ayirici	degisken
2.sirada	6.ayirici	degisken
3.sirada	9.ayirici	degisken
4.sirada	10.ayirici	degisken
5.sirada	5.ayirici	degisken
6.sirada	12.ayirici	degisken
7.sirada	11.ayirici	degisken
8.sirada	8.ayirici	degisken
9.sirada	2.ayirici	degisken
10.sirada	30.ayirici	degisken
11.sirada	36.ayirici	degisken
12.sirada	4.ayirici	degisken
13.sirada	31.ayirici	degisken
14.sirada	32.ayirici	degisken
15.sirada	3.ayirici	degisken
16.sirada	34.ayirici	degisken
17.sirada	14.ayirici	degisken
18.sirada	35.ayirici	degisken
19.sirada	26.ayirici	degisken
20.sirada	27.ayirici	degisken
21.sirada	17.ayirici	degisken
22.sirada	24.ayirici	degisken
23.sirada	29.ayirici	degisken
24.sirada	15.ayirici	degisken
25.sirada	18.ayirici	degisken
26.sirada	33.ayirici	degisken
27.sirada	1.ayirici	degisken
28.sirada	16.ayirici	degisken
29.sirada	20.ayirici	degisken
30.sirada	25.ayirici	degisken
31.sirada	13.ayirici	degisken
32.sirada	22.ayirici	degisken
33.sirada	19.ayirici	degisken
34.sirada	28.ayirici	degisken
35.sirada	21.ayirici	degisken
36.sirada	23.ayirici	degisken

2.dereceden önemli ayırma fonksiyonu

1.sirada	3.ayirici	degisken
2.sirada	2.ayirici	degisken
3.sirada	1.ayirici	degisken
4.sirada	4.ayirici	degisken
5.sirada	5.ayirici	degisken
6.sirada	6.ayirici	degisken
7.sirada	7.ayirici	degisken
8.sirada	36.ayirici	degisken
9.sirada	9.ayirici	degisken
10.sirada	29.ayirici	degisken
11.sirada	34.ayirici	degisken
12.sirada	35.ayirici	degisken
13.sirada	21.ayirici	degisken
14.sirada	13.ayirici	degisken
15.sirada	22.ayirici	degisken
16.sirada	8.ayirici	degisken
17.sirada	17.ayirici	degisken
18.sirada	16.ayirici	degisken
19.sirada	10.ayirici	degisken
20.sirada	31.ayirici	degisken
21.sirada	32.ayirici	degisken
22.sirada	23.ayirici	degisken
23.sirada	20.ayirici	degisken

24.sirada 15.ayirici degisken  
25.sirada 28.ayirici degisken  
26.sirada 19.ayirici degisken  
27.sirada 18.ayirici degisken  
28.sirada 24.ayirici degisken  
29.sirada 12.ayirici degisken  
30.sirada 25.ayirici degisken  
31.sirada 14.ayirici degisken  
32.sirada 30.ayirici degisken  
33.sirada 33.ayirici degisken  
34.sirada 11.ayirici degisken  
35.sirada 26.ayirici degisken  
36.sirada 27.ayirici degisken  
36.sirada 19.ayirici degisken

Umit KUNKÇU

### 1.dereceden önemli ayırma fonksiyonu

1.sirada 2.ayirici degisken  
2.sirada 3.ayirici degisken  
3.sirada 9.ayirici degisken  
4.sirada 10.ayirici degisken  
5.sirada 8.ayirici degisken  
6.sirada 11.ayirici degisken  
7.sirada 1.ayirici degisken  
8.sirada 7.ayirici degisken  
9.sirada 4.ayirici degisken  
10.sirada 12.ayirici degisken  
11.sirada 6.ayirici degisken  
12.sirada 32.ayirici degisken  
13.sirada 25.ayirici degisken  
14.sirada 5.ayirici degisken  
15.sirada 36.ayirici degisken  
16.sirada 18.ayirici degisken  
17.sirada 35.ayirici degisken  
18.sirada 29.ayirici degisken  
19.sirada 28.ayirici degisken  
20.sirada 30.ayirici degisken  
21.sirada 19.ayirici degisken  
22.sirada 16.ayirici degisken  
23.sirada 22.ayirici degisken  
24.sirada 20.ayirici degisken  
25.sirada 33.ayirici degisken  
26.sirada 26.ayirici degisken  
27.sirada 15.ayirici degisken  
28.sirada 24.ayirici degisken  
29.sirada 23.ayirici degisken  
30.sirada 27.ayirici degisken  
31.sirada 14.ayirici degisken  
32.sirada 31.ayirici degisken  
33.sirada 34.ayirici degisken  
34.sirada 13.ayirici degisken  
35.sirada 21.ayirici degisken  
36.sirada 17.ayirici degisken

### 2.dereceden önemli ayırma fonksiyonu

1.sirada 8.ayirici degisken  
2.sirada 9.ayirici degisken  
3.sirada 7.ayirici degisken  
4.sirada 10.ayirici degisken  
5.sirada 5.ayirici degisken  
6.sirada 6.ayirici degisken  
7.sirada 4.ayirici degisken  
8.sirada 11.ayirici degisken  
9.sirada 3.ayirici degisken  
10.sirada 12.ayirici degisken  
11.sirada 1.ayirici degisken  
12.sirada 29.ayirici degisken  
13.sirada 27.ayirici degisken

14.sirada	28.ayirici	degisken
15.sirada	25.ayirici	degisken
16.sirada	2.ayirici	degisken
17.sirada	34.ayirici	degisken
18.sirada	33.ayirici	degisken
19.sirada	35.ayirici	degisken
20.sirada	26.ayirici	degisken
21.sirada	23.ayirici	degisken
22.sirada	32.ayirici	degisken
23.sirada	24.ayirici	degisken
24.sirada	21.ayirici	degisken
25.sirada	18.ayirici	degisken
26.sirada	22.ayirici	degisken
27.sirada	20.ayirici	degisken
28.sirada	16.ayirici	degisken
29.sirada	19.ayirici	degisken
30.sirada	36.ayirici	degisken
31.sirada	14.ayirici	degisken
32.sirada	13.ayirici	degisken
33.sirada	31.ayirici	degisken
34.sirada	17.ayirici	degisken
35.sirada	30.ayirici	degisken
36.sirada	15.ayirici	degisken



**Ek 4: Salih EREN ve Umit KUNKÇU'den elde edilen LPC parametreleri için en önemli ayırma fonksiyonları**

**Salih EREN**

**1. dereceden önemli ayırma fonksiyonu**

1.sirada	6.ayirici	degisken
2.sirada	7.ayirici	degisken
3.sirada	5.ayirici	degisken
4.sirada	10.ayirici	degisken
5.sirada	8.ayirici	degisken
6.sirada	12.ayirici	degisken
7.sirada	9.ayirici	degisken
8.sirada	11.ayirici	degisken
9.sirada	4.ayirici	degisken
10.sirada	1.ayirici	degisken
11.sirada	3.ayirici	degisken
12.sirada	2.ayirici	degisken

**2.dereceden önemli ayırma fonksiyonu**

1.sirada	3.ayirici	degisken
2.sirada	2.ayirici	degisken
3.sirada	1.ayirici	degisken
4.sirada	4.ayirici	degisken
5.sirada	9.ayirici	degisken
6.sirada	8.ayirici	degisken
7.sirada	5.ayirici	degisken
8.sirada	10.ayirici	degisken
9.sirada	6.ayirici	degisken
10.sirada	11.ayirici	degisken
11.sirada	7.ayirici	degisken
12.sirada	12.ayirici	degisken

**Umit KUNKÇU**

**1.dereceden önemli ayırma fonksiyonu**

1.sirada	2.ayirici	degisken
2.sirada	3.ayirici	degisken
3.sirada	1.ayirici	degisken
4.sirada	9.ayirici	degisken
5.sirada	8.ayirici	degisken
6.sirada	10.ayirici	degisken
7.sirada	4.ayirici	degisken
8.sirada	11.ayirici	degisken
9.sirada	7.ayirici	degisken
10.sirada	6.ayirici	degisken
11.sirada	12.ayirici	degisken
12.sirada	5.ayirici	degisken

**2.dereceden önemli ayırma fonksiyonu**

1.sirada	2.ayirici	degisken
2.sirada	3.ayirici	degisken
3.sirada	4.ayirici	degisken
4.sirada	1.ayirici	degisken
5.sirada	5.ayirici	degisken
6.sirada	11.ayirici	degisken
7.sirada	10.ayirici	degisken
8.sirada	6.ayirici	degisken
9.sirada	9.ayirici	degisken
10.sirada	12.ayirici	degisken
11.sirada	7.ayirici	degisken
12.sirada	8.ayirici	degisken

Ek 5 : Salih EREN ve Umit KUNKÇU'den elde edilen Cepstral parametreleri için en önemli ayırma fonksiyonları

Salih EREN

1.dereceden önemli ayırma fonksiyonu

1.sirada 5.ayirici degisken  
2.sirada 6.ayirici degisken  
3.sirada 4.ayirici degisken  
4.sirada 7.ayirici degisken  
5.sirada 8.ayirici degisken  
6.sirada 13.ayirici degisken  
7.sirada 11.ayirici degisken  
8.sirada 10.ayirici degisken  
9.sirada 1.ayirici degisken  
10.sirada 12.ayirici degisken  
11.sirada 9.ayirici degisken  
12.sirada 3.ayirici degisken  
13.sirada 2.ayirici degisken

2.dereceden önemli ayırma fonksiyonu

1.sirada 4.ayirici degisken  
2.sirada 3.ayirici degisken  
3.sirada 5.ayirici degisken  
4.sirada 2.ayirici degisken  
5.sirada 1.ayirici degisken  
6.sirada 11.ayirici degisken  
7.sirada 12.ayirici degisken  
8.sirada 9.ayirici degisken  
9.sirada 8.ayirici degisken  
10.sirada 13.ayirici degisken  
11.sirada 7.ayirici degisken  
12.sirada 6.ayirici degisken  
13.sirada 10.ayirici degisken  
13.sirada 7.ayirici degisken

Umit KUNKÇU

1.dereceden önemli ayırma fonksiyonu

1.sirada 3.ayirici degisken  
2.sirada 6.ayirici degisken  
3.sirada 4.ayirici degisken  
4.sirada 7.ayirici degisken  
5.sirada 5.ayirici degisken  
6.sirada 12.ayirici degisken  
7.sirada 10.ayirici degisken  
8.sirada 2.ayirici degisken  
9.sirada 11.ayirici degisken  
10.sirada 8.ayirici degisken  
11.sirada 9.ayirici degisken  
12.sirada 1.ayirici degisken  
13.sirada 13.ayirici degisken

2.dereceden önemli ayırma fonksiyonu

1.sirada 3.ayirici degisken  
2.sirada 10.ayirici degisken  
3.sirada 2.ayirici degisken  
4.sirada 7.ayirici degisken  
5.sirada 12.ayirici degisken  
6.sirada 8.ayirici degisken  
7.sirada 9.ayirici degisken  
8.sirada 6.ayirici degisken  
9.sirada 4.ayirici degisken  
10.sirada 13.ayirici degisken  
11.sirada 1.ayirici degisken  
12.sirada 5.ayirici degisken  
13.sirada 11.ayirici degisken

**Ek 6: Salih EREN ve Ümit KUNKÇU'den elde edilen LAR parametreleri için en önemli ayırma fonksiyonları**

**Salih EREN**

**1. dereceden önemli ayırma fonksiyonu**

1.sirada	7.ayirici	degisken
2.sirada	6.ayirici	degisken
3.sirada	10.ayirici	degisken
4.sirada	9.ayirici	degisken
5.sirada	1.ayirici	degisken
6.sirada	5.ayirici	degisken
7.sirada	8.ayirici	degisken
8.sirada	2.ayirici	degisken
9.sirada	3.ayirici	degisken
10.sirada	4.ayirici	degisken
11.sirada	12.ayirici	degisken
12.sirada	11.ayirici	degisken

**2.dereceden önemli ayırma fonksiyonu**

1.sirada	1.ayirici	degisken
2.sirada	3.ayirici	degisken
3.sirada	4.ayirici	degisken
4.sirada	5.ayirici	degisken
5.sirada	2.ayirici	degisken
6.sirada	8.ayirici	degisken
7.sirada	11.ayirici	degisken
8.sirada	6.ayirici	degisken
9.sirada	12.ayirici	degisken
10.sirada	9.ayirici	degisken
11.sirada	7.ayirici	degisken
12.sirada	10.ayirici	degisken

**Ümit KUNKÇU**

**1.dereceden önemli ayırma fonksiyonu**

1.sirada	2.ayirici	degisken
2.sirada	5.ayirici	degisken
3.sirada	6.ayirici	degisken
4.sirada	3.ayirici	degisken
5.sirada	10.ayirici	degisken
6.sirada	4.ayirici	degisken
7.sirada	7.ayirici	degisken
8.sirada	9.ayirici	degisken
9.sirada	12.ayirici	degisken
10.sirada	1.ayirici	degisken
11.sirada	11.ayirici	degisken
12.sirada	8.ayirici	degisken

**2.dereceden önemli ayırma fonksiyonu**

1.sirada	1.ayirici	degisken
2.sirada	2.ayirici	degisken
3.sirada	4.ayirici	degisken
4.sirada	7.ayirici	degisken
5.sirada	3.ayirici	degisken
6.sirada	8.ayirici	degisken
7.sirada	6.ayirici	degisken
8.sirada	11.ayirici	degisken
9.sirada	12.ayirici	degisken
10.sirada	9.ayirici	degisken
11.sirada	10.ayirici	degisken
12.sirada	5.ayirici	degisken

Ek 7 : Salih EREN ve Ümit KÜNKÜ' den alınan Otokorelasyon - LPC -  
Formant parametreleri için en önemli ayırma fonksiyonları

Salih EREN

1.dereceden ayırma fonksiyonu

1.sirada	2.ayirici	degisken
2.sirada	19.ayirici	degisken
3.sirada	20.ayirici	degisken
4.sirada	18.ayirici	degisken
5.sirada	3.ayirici	degisken
6.sirada	1.ayirici	degisken
7.sirada	21.ayirici	degisken
8.sirada	4.ayirici	degisken
9.sirada	7.ayirici	degisken
10.sirada	28.ayirici	degisken
11.sirada	23.ayirici	degisken
12.sirada	13.ayirici	degisken
13.sirada	27.ayirici	degisken
14.sirada	11.ayirici	degisken
15.sirada	24.ayirici	degisken
16.sirada	17.ayirici	degisken
17.sirada	10.ayirici	degisken
18.sirada	25.ayirici	degisken
19.sirada	9.ayirici	degisken
20.sirada	16.ayirici	degisken
21.sirada	15.ayirici	degisken
22.sirada	6.ayirici	degisken
23.sirada	22.ayirici	degisken
24.sirada	8.ayirici	degisken
25.sirada	14.ayirici	degisken
26.sirada	12.ayirici	degisken
27.sirada	5.ayirici	degisken
28.sirada	26.ayirici	degisken

2.dereceden ayırma fonksiyonu

1.sirada	15.ayirici	degisken
2.sirada	16.ayirici	degisken
3.sirada	14.ayirici	degisken
4.sirada	17.ayirici	degisken
5.sirada	18.ayirici	degisken
6.sirada	3.ayirici	degisken
7.sirada	7.ayirici	degisken
8.sirada	2.ayirici	degisken
9.sirada	24.ayirici	degisken
10.sirada	19.ayirici	degisken
11.sirada	25.ayirici	degisken
12.sirada	10.ayirici	degisken
13.sirada	4.ayirici	degisken
14.sirada	21.ayirici	degisken
15.sirada	1.ayirici	degisken
16.sirada	23.ayirici	degisken
17.sirada	26.ayirici	degisken
18.sirada	11.ayirici	degisken
19.sirada	5.ayirici	degisken
20.sirada	28.ayirici	degisken
21.sirada	8.ayirici	degisken
22.sirada	20.ayirici	degisken
23.sirada	27.ayirici	degisken
24.sirada	12.ayirici	degisken
25.sirada	6.ayirici	degisken
26.sirada	9.ayirici	degisken
27.sirada	13.ayirici	degisken
28.sirada	22.ayirici	degisken

## Umit KUNKÇU

### 1.dereceden ayırma fonksiyonu

1.sirada	2.ayirici degisken
2.sirada	1.ayirici degisken
3.sirada	3.ayirici degisken
4.sirada	6.ayirici degisken
5.sirada	8.ayirici degisken
6.sirada	7.ayirici degisken
7.sirada	5.ayirici degisken
8.sirada	4.ayirici degisken
9.sirada	9.ayirici degisken
10.sirada	21.ayirici degisken
11.sirada	22.ayirici degisken
12.sirada	20.ayirici degisken
13.sirada	23.ayirici degisken
14.sirada	19.ayirici degisken
15.sirada	24.ayirici degisken
16.sirada	18.ayirici degisken
17.sirada	15.ayirici degisken
18.sirada	14.ayirici degisken
19.sirada	12.ayirici degisken
20.sirada	13.ayirici degisken
21.sirada	25.ayirici degisken
22.sirada	17.ayirici degisken
23.sirada	16.ayirici degisken
24.sirada	10.ayirici degisken
25.sirada	11.ayirici degisken
26.sirada	26.ayirici degisken
27.sirada	28.ayirici degisken
28.sirada	27.ayirici degisken

### 2.dereceden ayırma fonksiyonu

1.sirada	11.ayirici degisken
2.sirada	10.ayirici degisken
3.sirada	4.ayirici degisken
4.sirada	3.ayirici degisken
5.sirada	12.ayirici degisken
6.sirada	5.ayirici degisken
7.sirada	9.ayirici degisken
8.sirada	2.ayirici degisken
9.sirada	15.ayirici degisken
10.sirada	6.ayirici degisken
11.sirada	16.ayirici degisken
12.sirada	13.ayirici degisken
13.sirada	8.ayirici degisken
14.sirada	14.ayirici degisken
15.sirada	17.ayirici degisken
16.sirada	23.ayirici degisken
17.sirada	22.ayirici degisken
18.sirada	1.ayirici degisken
19.sirada	24.ayirici degisken
20.sirada	21.ayirici degisken
21.sirada	7.ayirici degisken
22.sirada	18.ayirici degisken
23.sirada	25.ayirici degisken
24.sirada	20.ayirici degisken
25.sirada	19.ayirici degisken
26.sirada	26.ayirici degisken
27.sirada	27.ayirici degisken
28.sirada	18.ayirici degisken

k 8: Salih EREN ve Ümit KÜNKÇÜ'den elde edilen Otokorelasyon parametreleri için en önemli ayırma fonksiyonları

Salih EREN

1. dereceden önemli ayırma fonksiyonu

1.sirada	2.ayirici	degisken
2.sirada	3.ayirici	degisken
3.sirada	7.ayirici	degisken
4.sirada	4.ayirici	degisken
5.sirada	9.ayirici	degisken
6.sirada	6.ayirici	degisken
7.sirada	10.ayirici	degisken
8.sirada	5.ayirici	degisken
9.sirada	12.ayirici	degisken
10.sirada	8.ayirici	degisken
11.sirada	13.ayirici	degisken
12.sirada	1.ayirici	degisken
13.sirada	11.ayirici	degisken

2.dereceden önemli ayırma fonksiyonu

1.sirada	2.ayirici	degisken
2.sirada	10.ayirici	degisken
3.sirada	3.ayirici	degisken
4.sirada	7.ayirici	degisken
5.sirada	13.ayirici	degisken
6.sirada	6.ayirici	degisken
7.sirada	5.ayirici	degisken
8.sirada	11.ayirici	degisken
9.sirada	9.ayirici	degisken
10.sirada	1.ayirici	degisken
11.sirada	4.ayirici	degisken
12.sirada	8.ayirici	degisken
13.sirada	12.ayirici	degisken
13.sirada	1.ayirici	degisken

Ümit KÜNKÇÜ

1.dereceden önemli ayırma fonksiyonu

1.sirada	2.ayirici	degisken
2.sirada	1.ayirici	degisken
3.sirada	9.ayirici	degisken
4.sirada	4.ayirici	degisken
5.sirada	8.ayirici	degisken
6.sirada	5.ayirici	degisken
7.sirada	12.ayirici	degisken
8.sirada	13.ayirici	degisken
9.sirada	3.ayirici	degisken
10.sirada	11.ayirici	degisken
11.sirada	7.ayirici	degisken
12.sirada	10.ayirici	degisken
13.sirada	6.ayirici	degisken

2.dereceden önemli ayırma fonksiyonu

1.sirada	12.ayirici	degisken
2.sirada	11.ayirici	degisken
3.sirada	2.ayirici	degisken
4.sirada	5.ayirici	degisken
5.sirada	10.ayirici	degisken
6.sirada	1.ayirici	degisken
7.sirada	4.ayirici	degisken
8.sirada	6.ayirici	degisken
9.sirada	13.ayirici	degisken
10.sirada	7.ayirici	degisken
11.sirada	8.ayirici	degisken
12.sirada	9.ayirici	degisken
13.sirada	3.ayirici	degisken

**Ek 9: Ümit KUNKCU'den elde edilen LAR parametrelerine göre sekiz  
ünlü için sınıflandırma**

A ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
A ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
A ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
E ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
E ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
E ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
I ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
I ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
I ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
ı ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
ı ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
ı ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
O ünlüsü ve 1. örnek için sınıflandırma sonucu	olumsuz
O ünlüsü ve 2. örnek için sınıflandırma sonucu	olumsuz
O ünlüsü ve 3. örnek için sınıflandırma sonucu	olumsuz
ö ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
ö ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
ö ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
U ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
U ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
U ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
Ü ünlüsü ve 1. örnek için sınıflandırma sonucu	olumsuz
Ü ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
Ü ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu

Ek 10: Ümit KUNKÇU'den elde edilen LPC parametrelerine göre sekiz  
ünlü için sınıflandırma

A ünlüsü ve 1. örnek için sınıflandırma sonucu olumlu  
A ünlüsü ve 2. örnek için sınıflandırma sonucu olumlu  
A ünlüsü ve 3. örnek için sınıflandırma sonucu olumlu  
E ünlüsü ve 1. örnek için sınıflandırma sonucu olumlu  
E ünlüsü ve 2. örnek için sınıflandırma sonucu olumlu  
E ünlüsü ve 3. örnek için sınıflandırma sonucu olumlu  
I ünlüsü ve 1. örnek için sınıflandırma sonucu olumlu  
I ünlüsü ve 2. örnek için sınıflandırma sonucu olumlu  
I ünlüsü ve 3. örnek için sınıflandırma sonucu olumlu  
İ ünlüsü ve 1. örnek için sınıflandırma sonucu olumlu  
İ ünlüsü ve 2. örnek için sınıflandırma sonucu olumlu  
İ ünlüsü ve 3. örnek için sınıflandırma sonucu olumlu  
O ünlüsü ve 1. örnek için sınıflandırma sonucu olumsuz  
O ünlüsü ve 2. örnek için sınıflandırma sonucu olumlu  
O ünlüsü ve 3. örnek için sınıflandırma sonucu olumlu  
Ö ünlüsü ve 1. örnek için sınıflandırma sonucu olumlu  
Ö ünlüsü ve 2. örnek için sınıflandırma sonucu olumlu  
Ö ünlüsü ve 3. örnek için sınıflandırma sonucu olumlu  
U ünlüsü ve 1. örnek için sınıflandırma sonucu olumlu  
U ünlüsü ve 2. örnek için sınıflandırma sonucu olumlu  
U ünlüsü ve 3. örnek için sınıflandırma sonucu olumlu  
Ü ünlüsü ve 1. örnek için sınıflandırma sonucu olumlu  
Ü ünlüsü ve 2. örnek için sınıflandırma sonucu olumlu  
Ü ünlüsü ve 3. örnek için sınıflandırma sonucu olumlu



**Ek 11 : Umit KUNKÇU'den elde edilen Otokorelasyon parametrelerine  
göre sekiz ünlü için sınıflandırma**

A ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
A ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
A ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
E ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
E ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
E ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
I ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
I ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
I ünlüsü ve 3. örnek için sınıflandırma sonucu	olumsuz
i ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
i ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
i ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
O ünlüsü ve 1. örnek için sınıflandırma sonucu	olumsuz
O ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
O ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
Ö ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
Ö ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
Ö ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
U ünlüsü ve 1. örnek için sınıflandırma sonucu	olumlu
U ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
U ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu
Ü ünlüsü ve 1. örnek için sınıflandırma sonucu	olumsuz
Ü ünlüsü ve 2. örnek için sınıflandırma sonucu	olumlu
Ü ünlüsü ve 3. örnek için sınıflandırma sonucu	olumlu