

ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

VEKTÖR NİCEMLEME TEKNİKLERİNE DAYALI KONUŞMACI TANIMA
ALGORİTMALARININ İNCELENMESİ

Metin UZUNÇARŞILI

ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA

2005

Her hakkı saklıdır

Yrd. Doç. Dr. H. Gökhan İLK danışmanlığında, METİN UZUNÇARŞILI tarafından hazırlanan bu çalışma 25/07/2005 tarihinde aşağıdaki jüri tarafından Elektronik Mühendisliği Anabilim Dalı'nda yüksek lisans tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. H. Faruk ÖZEK,
Ankara Üniversitesi Elektronik Mühendisliği Bölümü

İmza :

Üye : Yrd. Doç. Dr. H. Gökhan İLK,
Ankara Üniversitesi Elektronik Mühendisliği Bölümü

İmza :

Üye : Yrd. Doç. Dr. Cemal ATAKAN,
Ankara Üniversitesi İstatistik Bölümü

İmza :

Yukarıdaki sonucu onaylarım

Prof. Dr. Ülkü MEHMETOĞLU
Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

VEKTÖR NİCEMLEME TEKNİKLERİNE DAYALI KONUŞMACI TANIMA ALGORİTMALARININ İNCELENMESİ

Metin UZUNÇARŞILI

Ankara Üniversitesi

Fen Bilimleri Enstitüsü

Elektronik Mühendisliği Ana Bilim Dalı

Danışman: Yrd. Doç. Dr. H. Gökhan İLK

Konuşmacı tanıma işlemi, seslerin, konuşmacılardan dolayı sahip oldukları karakteristikler ve özellikler yardımıyla çeşitli analiz teknikleri ve yöntemleri uygulanarak bir konuşmacının diğerlerinden ayırt edilmesi işlemidir. Sesin bu karakteristik özellikleri tıpkı parmak izi ya da DNA gibi insandan insana farklılık gösterdiği için ayırt edici bir özelliğe sahiptir. Bu çalışmada; kapalı set, metinden bağımsız, vektör nicemeleme tekniklerine dayanan bir konuşmacı tanımlama sistemi tasarlanmıştır. Her bir konuşmacının kayıtlı konuşmaları kullanılarak öznitelik vektörleri çıkarılmış ve bu vektörler LBG algoritmasıyla eğitilip her bir konuşmacı için kod kitapları tasarlanmıştır. Özellikle güvenliğin ön planda olduğu sistemler için hızlı ve başarılı bir çözüm sağlamak amacıyla kullanılan sistemin performansını artırma yollarına gidilmiştir. Bu yüzden test telaffuzları ve kayıtlı konuşmacı sayısı değiştirilmiştir. Kod kitabının içerdiği vektör sayısı artırıldığında ve test işlemi kelime yerine cümle üzerinden yapıldığında sistem tanıma performansının arttığı gözlemlenmiştir. Sistem performansını ortaya koymak için çizilen alıcı işletim karakteristik eğrisi altında kalan alan hesaplandığında sistemin tatmin edici bir performans ortaya koyduğu tespit edilmiştir.

2005, 68 sayfa

ANAHTAR KELİMELER: Vektör nicemeleme, öznitelik parametreleri, kod kitabı, metinden bağımsız konuşmacı tanıma, kümelendirme analizi

ABSTRACT
Master Thesis

INVESTIGATION OF VECTOR QUANTIZATION BASED SPEAKER
IDENTIFICATION ALGORITHMS

Metin UZUNCARSILI

Ankara University

Graduate School of Natural and Applied Sciences

Department of Electronic Engineering

Supervisor: Ass. Prof. Dr. H. Gokhan ILK

Speaker identification uses various analysis techniques and methods with the assistance of characteristics and specifications of speech in order to identify speakers. These characteristic properties of speech are distinctive as DNA and finger print due to uniqueness of these characteristics for every people. In this study, vector quantization based text independent speaker identification algorithm has been designed for a closed set of speakers. Feature vectors have been extracted and a codebook has been designed by training these vectors using the LBG algorithm for each speaker. Three different methods have been examined in order to increase the system performance. First one is changing test utterance by performing tests using words instead of sentences. Second one is increasing the number of the speakers in the closed set. Third one is increasing the number of the code vectors in each speaker's codebook. The results revealed that the system performance would be increased when the system was tested using sentences instead of words and increased when the number of the code words was increased. The system performance was determined by computing the area under the ROC (Receiver Operating Characteristic) curve, which indicated satisfactory performance to be used in a speaker identification system.

2005, 68 pages

Key Words: Vector quantization, feature parameters, codebook, text-independent speaker identification, clustering analysis

TEŐEKKÜR

Çalıőmamın her aőamasında önerileri ile beni yönlendiren, ihtiyaç duyduğum her an yanımda olarak bana bilimsel danışmanlık veren sevgili danışmanım Sayın Yrd. Doç. Dr. H. Gökhan İLK'e teşekkürü bir borç bilirim.

Tez çalışması ve tezin yazımı süresince dostluklarını ve sevgilerini bir an bile esirgemeyen, bu süre boyunca bana tahammül eden tüm sevgili dostlarıma, can dostum Hakan ERTAŐ'a ve kız arkadaşım Ayőenur TOPTAN'a sonsuz teşekkürler...

Metin UZUNÇARŐILI
Ankara, Temmuz 2005

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	viii
1. GİRİŞ	1
2. KURAMSAL TEMELLER	4
2.1. Ses	4
2.2. Sesin Oluşum Mekanizması ve Fizyolojisi	4
2.3. Sesin Özellikleri	5
2.4. Konuşmacı Tanıma	6
2.5. Konuşmacı Tanıma Sistemlerinin Sınıflandırılması	7
2.6. Metinden Bağımsız Konuşmacı Tanıma Metodları	8
2.6.1. Gaussian karma modeli	8
2.6.1.1. Model tanımı ve açıklaması	8
2.6.1.2. En çok olabilirlik parametre tahmin yöntemi	11
2.6.1.3. Konuşmacı tanıma	13
2.6.2. Vektör nicemleme tekniklerine dayalı konuşmacı tanıma	13
2.6.2.1. Kümelendirme analizi	14
2.6.2.1.1. Birleştirme/bağlama (ağaç kümelendirme)	15
2.6.2.1.2. Hiyerarşik ağaç	15
2.6.2.1.3. Uzaklık ölçümü	16
2.6.2.1.3.1. Euclid uzaklığı	17
2.6.2.1.3.2. City-block (manhattan) uzaklığı	17
2.6.2.1.4. Bağlantı kuralları	17
2.6.2.1.4.1. Tek bağlantı(en yakın komşu)	18
2.6.2.1.4.2. Tam bağlantı(en uzak komşu)	18
2.6.2.1.5. K-ortalama kümelendirme	18
2.6.2.1.6. Beklenti maksimizasyon algoritması	19
2.6.2.2. Vektör nicemleme	20
2.6.2.3. Vektör nicemleme tasarımı	22
2.6.2.3.1. Linde, Buzo, Gray algoritması	24
2.6.2.4. Konuşmacı tanımda vektör nicemleme kullanımı	25
2.7. Metne Bağlı Konuşmacı Tanıma Metodları	25
2.7.1. Saklı markov model	25
2.7.2. Dinamik zaman eğrisi	27
2.7.2.1. Dinamik programlama	29
2.8. Konuşmacı Tanımda Kullanılan Parametreler	31
2.8.1. Parametrelendirme yapılması	31
2.8.1.1. Doğrusal öngörüm katsayıları	31
2.8.1.2. Cepstral katsayılar	34
2.8.1.3. Mel-ölçeklendirilmiş cepstrum katsayıları	37
2.8.1.3.1. Mel-ölçeklendirme	37
2.8.1.3.2. Mel-ölçeklendirilmiş cepstral katsayılar	39
2.8.1.3.3. Çerçeveleme	39

2.8.1.3.4. Pencereleme	39
2.8.1.3.5. FFT spektrum	40
2.8.1.3.6. Mel-spektrum	40
2.8.1.3.7. Mel-filtre bankası tasarımı	40
2.8.1.3.8. Mel-cepstrum	41
2.8.1.4. Doğrusal öngörüm katsayılarından türetilmiş cepstral katsayılar	42
2.8.1.5. Delta cepstral katsayılar	43
2.9. Alıcı İşletim Karakteristik Eğrisi(Receiver Operating Characteristic Curve)	45
3. MATERYAL ve YÖNTEM	50
4. ARAŞTIRMA BULGULARI	52
4.1. Eğitim Aşaması	53
4.1.1. Öznitelik vektörlerinin elde edilmesi	53
4.1.2. Kod kitap tasarımı	55
4.2. Test Süreci	57
5. TARTIŞMA ve SONUÇ	58
5.1. Kod Kitap Boyutunun Tanıma Performansı Üzerindeki Etkisi	58
5.2. Test Sürecinde Sistemin Cümle Yerine Kelime ile Test Edilmesinin Tanıma Performansına Etkisi	59
5.3. Konuşmacı Sayısı Artışının Tanıma Performansına Etkisi	60
KAYNAKLAR	63
EK. Konuşmacı veritabanında kayıtlı kelimeler ve cümleler	65
ÖZGEÇMİŞ	68

ŞEKİLLER DİZİNİ

Şekil 2.1. Konuşma üretim mekanizması.....	4
Şekil 2.2. Konuşmacı tanımlama	6
Şekil 2.3. Konuşmacı doğrulama blok diyagramı.....	7
Şekil 2.4. M bileşenli Gaussian yoğunluk.....	8
Şekil 2.5. Unimodal Gaussian, karışım ve vektör nicemleme modellerine ait karşılaştırma grafikleri	11
Şekil 2.6. Kümelendirme tanımı	15
Şekil 2.7. Hiyerarşik ağaç	16
Şekil 2.8. Dağılım karışımları	20
Şekil 2.9. 1-Boyutlu vektör nicemleme.....	21
Şekil 2.10. 2-Boyutlu vektör nicemleme.....	21
Şekil 2.11. LBG algoritması akış diyagramı.....	24
Şekil 2.12. Viterbi algoritmasının matris gösterimi.....	27
Şekil 2.13. “Speech” kelimesinin değişik zaman normalizasyonlarıyla karşılaştırılması	28
Şekil 2.14. Dinamik programlamanın gösterildiği noktalı düzlem.....	30
Şekil 2.15. Konuşmanın doğrusal öngörüm modeli.....	32
Şekil 2.16. Konuşma sinyali üretim modeli	33
Şekil 2.17. a seslisinin dalga şekli.....	35
Şekil 2.18. a seslinin cepstral gösterimi	35
Şekil 2.19. Cepstral katsayıların elde edilişi	36
Şekil 2.20. Mel-ölçeklendirme örneği.....	37
Şekil 2.21. 0-4 kHz arasında mel-ölçeklendirme grafiği	38
Şekil 2.22. 0-10 kHz. arasında mel-ölçeklendirme grafiği	38
Şekil 2.23. Mel-ölçeklendirilmiş cepstral katsayıların elde edilişi	39
Şekil 2.24. FFT spektrum.....	40
Şekil 2.25. Mel-filtre bankası.....	41
Şekil 2.26. Mel cepstrum	41
Şekil 2.27. Değişik ROC eğri gösterimleri	47
Şekil 2.28. İki ROC eğrisinin karşılaştırılması.....	48

Şekil 3.1. Matlab arayüzü	50
Şekil 4.1. Tasarlanan konuşmacı tanıma sistemi blok diyagramı	52
Şekil 4.2. Mel filtre bankası merkez frekans dizilim gösterimi	54
Şekil 4.3. 40 adet filtrenin frekans tepkesi	54
Şekil 4.4. Mel filtre bankası çıkış örneği	55
Şekil 5.1. Alıcı işletim karakteristik eğrisi	62

ÇİZELGELER DİZİNİ

Çizelge 3.1. Kayıttta kullanılan telefon çeşitleri	51
Çizelge 4.1. 6-bit vektör nicemleyicinin konuşmacı tanıma performansı	57
Çizelge 5.1. Değişik kod kitap boyutları için tanıma performans değişimi.....	58
Çizelge 5.2. Kelime/cümle ile test edilen sistemde tanıma performansları	60
Çizelge 5.3. 7-bit kod kitap kullanıldığında konuşmacı sayısı artışının performansa etkisi	60
Çizelge 5.4. ROC eğrisi ölçüleri	61

1. GİRİŞ

Ses içerisinde dil, kişilik, fizyoloji, ruh hali, yaş, ortam, lehçe gibi birçok bilgiyi aynı anda bulundurabilen karmaşık bir işarettir. Ses bilimciler, bu işareten istenilen bilgiyi elde etmeye çalışmakta ve elde edilen bilgiler doğrultusunda sonuçları günlük hayatta kullanmayı amaçlamaktadırlar. Konuşma sentezi, konuşmacı tanıma ve konuşma tanıma, konuşma işleme uygulamalarının en temel örnekleridir.

Bilgisayar teknolojisindeki gelişmelere paralel olarak iletişimin önemli bir unsuru olan ses yeni bir boyut kazanmıştır. Teknolojinin çok hızlı ilerlediği bu çağda bilgi güvenliğinin çok önemli olduğu bir gerçektir. Ses akustik parametreleri bilgisayar yardımıyla kolayca bulunabilmekte ve bulunan bu parametreler çeşitli programlar yardımıyla işlenip güncel uygulamalarda kullanılabilir. Sesin bilgisayar veya elektronik devre elemanlarınca tanınması sayesinde iletişim ve güvenlik kurumları tarafından verilen hizmetlerde hız, güven ve çağdaşlık kazandırılmıştır.

Konuşma işleme teknolojisindeki gelişmelerle ağır işitenler artık daha iyi işitebilmekte, sağır olanlar ise konuşmanın anında yazıya çevrilmesi ile canlı yayındaki konuşmaları anlayabilmektedir.

Konuşmacı kimliğini belirleyen etkenlerden biri kişinin organlarının anatomik yapısı diğeri kişinin bu yapıyı zaman içinde kendine özgü bir şekilde kullanmasıdır(Nolan 1983). Ses telleri, ses yolunun uzunluğu, ağız ve burun boşluklarının büyüklüğü, tam veya noksan dişler, dudak yapısı gibi anatomik durumlar birinci faktöre girer. İkinci faktör ise zamanla kazanılmış, öğrenilmiş etkenlerdir: Lehçe, telaffuz ve etnik farklılıklar gibi. Birinci grup faktörler formantlar ve temel frekans gibi sesin statik özellikleri, ikinci gruba giren faktörler formant geçişleri ve koartikülasyon gibi dinamik özellikleri belirler. İnsanların biyolojik yapısındaki bu farklılık ürettikleri sesin farklı olmasına yol açar. Bu sayede konuşma sinyalleri neredeyse parmak izi kadar güvenilir ve ayırdedici bir özellik olarak kullanılmaya başlanmıştır. Sesin bu öneminden dolayı konuşmacı

tanıma sistemleri de önem kazanmış ve günümüzde kritik uygulama alanlarında kendine yer bulmaya başlamıştır. Bu uygulamalardan bazıları: Güvenliğin ön planda olduğu yerlerde sadece belirli kişilerin seslerine cevap veren ses aktif düzenekler, güvenlik kuruluşlarında veya kriminal laboratuvarlarında şüphelenilen bir şahsın sesinin kayıtlardaki seslerle karşılaştırılması yoluna giderek sesin gerçek sahibin bulma, telefon bankacılığı. Konuşmacı tanıma uygulamalarının ilgi çekiciliği telefon ve internet üzerinden diyalog hizmetlere olan gereksinimin artışına paralel olarak artmaktadır.

Konuşmacı tanıma sistemlerinde veri olarak ses işareti alınır ve bu veriden konuşmacıyı modelleyecek, ilerideki tanıma işlemlerinde kullanılacak karakteristik şablon elde edilir. Tanıma aşamasında önceden elde edilen başvuru şablonlarıyla test için alınan konuşmadan elde edilen şablonlar karşılaştırılır. Benzerlik olduğuna karar verildiğinde konuşmacı kabul edilir aksi durumda reddedilir.

Konuşma çalışmalarının tarihi akışı içinde konuşmacı tanıma, konuşma tanımadan sonra gelmektedir. Bu nedenle konuşma tanıma tekniklerinin bazıları konuşmacı tanımada da kullanılmıştır. Daha önce kaydedilmiş sesleri tanımaya yönelik olan metne bağlı ses tanıma amacıyla Wolf 1972 yılında dinamik zaman deformasyonu yöntemini kullanmıştır(Gish et al. 1986). Soong metinden bağımsız konuşmacı tanıma için kümeleme yaklaşımı önermiştir. Bu bir vektör nicemeleme yaklaşımıdır; konuşmacı eğitim vektörlerinin kümelenişi o konuşmacıyı temsil eden model olur(Liu et al. 1996). Sınıflama da minimum nicemeleme hatası ile belirlenir. Vektör nicemeleme ile metinden bağımsız konuşmacı tanımada bir çok başarılı uygulama gerçekleştirilmiştir. Sesin bir başka modellenme şekli ise Saklı Markov Model gösterimidir. Bu modellemede konuşmanın en küçük birimleri olan dile özgü sesler bir durum olarak modellenir ve bu durumların birbirlerine geçiş olasılıkları incelenir(Rabiner et al. 1989). Saklı Markov Modellemede durumların birbirlerine geçme olasılıkları başvuru konuşmadan elde edilir ve karakteristik örüntü hesaplanır. Test konuşmasının da örüntüsü elde edildikten sonra karşılaştırma işlemi yapılır.

Sesin zamanla deęişen karakteristiklerini göz önüne almayan başka bir metinden bağımsız konuşmacı tanıma yaklaşımı da Reynolds'un getirdiđi Gaussian karma modelidir. Bu modelde Gauss bileşenlerinin spektral şekillerinin konuşmacıya bađlı olduđu gösterilmiştir(Reynolds et al. 1995). Bunlarla konuşmacı kişiliđi verimli olarak modellenabilir.

İnsanların deęişik durumlarda farklı ton ve vurguda konuşmaları, duygusal koşulların deęişmesinin sese yansması sonucu çıkarılan seslerin farklı olması ve hastalık gibi durumlarda sesin deęişmesi dikkate alındığında kullanılan konuşma ve konuşmacı tanıma sistem yöntemlerinin karmaşıklığı ve geliştirilmeye açık olduđu görülmektedir .

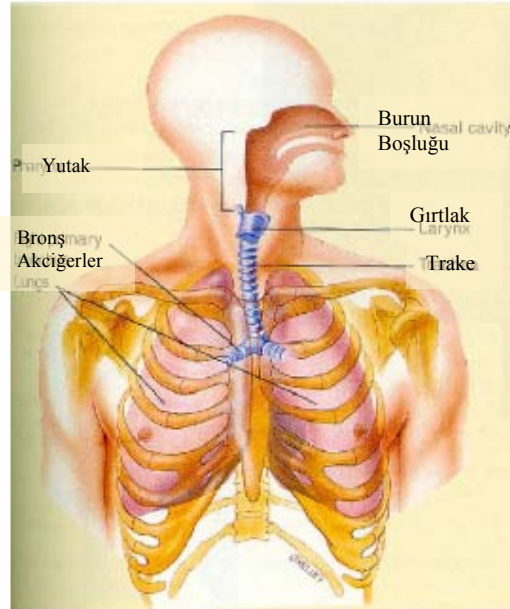
2. KURAMSAL TEMELLER

2.1. Ses

Ses, akustik bir sinyaldir. Ses oluşumu için maddesel ortamda titreşim yapabilen bir ses kaynağı ve ses dalgalarının yayılabileceği iletici bir ortama ihtiyaç vardır. Ses dalgaları; birinden diğerine ulaşan maddesel ortamların titreşimleridir. Her madde, komşu partikül üzerine kendinde bulunan hareketi nakletmek için etki yapar. Bir ortam içerisinde sesin yayılabilmesi, sesin ve ortamın özelliklerine bağlıdır. Ses dalgası bir düzlem üzerinde basınç meydana getirir ve bu da sesin fizik şiddetini ölçmeye yarar.

2.2. Sesin Oluşum Mekanizması ve Fizyolojisi

Ses, gırtlak, akciğerler, kas ve iskelet sistemi, psiko-nörolojik sistemlerin birbirleriyle koordineli olarak çalışması sonucunda oluşur. Şekil 2.1' de sesin oluşumunda etkisi olan tüm organlar ayrıntılı olarak verilmiştir.



Şekil 2.1. Konuşma üretim mekanizması

Ses oluşumu sırasında ses tellerinin titreşimi için gerekli olan havanın kaynağı akciğerlerdir. Titreşimi sağlayan temel kas diyafram olup, soluk vermeyi birincil abdominal kaslar sağlar. Psikonörojenik sistem ses oluşumu sırasında kas hareketleri arasındaki koordinasyonun düzenlenmesi ile önemli role sahiptir.

Sonuç olarak akciğerlerdeki hava karımsal kasların yardımıyla ses tellerinin arasından geçerek vızıltı sesine benzer ham sesi oluşturur. Bu ses nefes borusu ağız üzerinde ses telleri boyunca, geniz, ağız boşluğu, dil, burun ve sinüslerde rezonansa uğrar (Karamürsel 2002).

2.3. Sesin Özellikleri

Ses, maddeden oluşan bir ortamda yayılan bir titreşim dalgasıdır. Sesin oluşumu için titreşim özelliği olan bir enerji kaynağına ve çoğalıp dağılması için de elastik elemanlara sahip bir ortamın bulunmasına gerek vardır. Bu şekilde ses enerjisi, hava ortamında bulunan moleküllerin yer yer sıkışması ve gevşemesi ile ses dalgaları şeklinde iletilir.

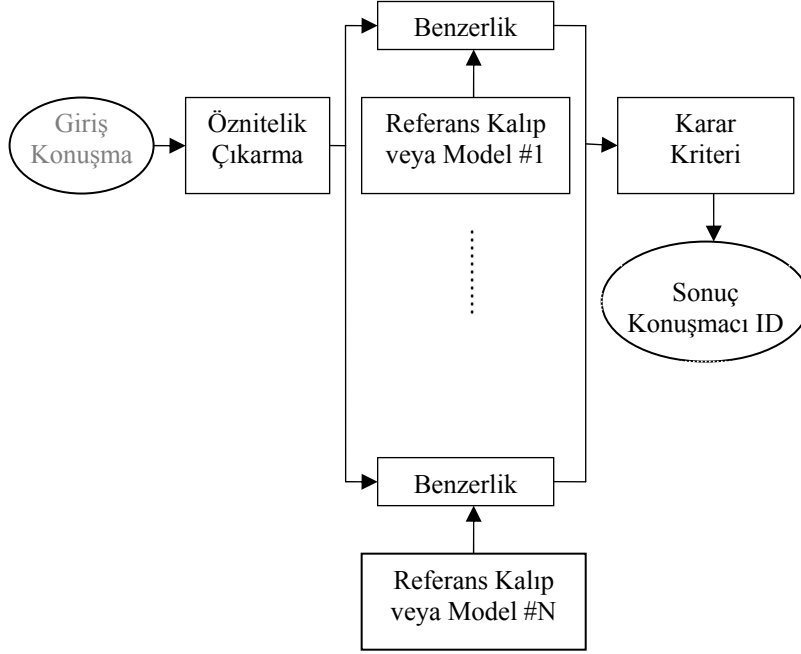
Ses Dalgası: Sabit bir genliği ve tek bir frekansı bulunan sinüzoidal bir dalgadır.

Frekans: Ses dalgaları enine ve boyuna sinüzoidal dalgalar halindedir. Birim zamanda ses dalgalarını oluşturan partiküllerin sıkışması ve gevşemesi ile oluşan tam bir dalga, sesin frekansını oluşturur. Yani saniyedeki titreşim sayısıdır. Buna temel frekans da denir. İnsan kulağı 20-20000 Hz arasındaki sesleri algılayabilir. Günlük konuşma sesleri 500-1000-2000 Hz arasındadır.

Şiddet: Ses dalgasının yayılma doğrultusundaki dik bir düzlem içinde 1cm² lik yüzeye 4 saniyede verdiği ses enerjisidir. Ölçüsü dB dir(Ergenç 1995).

2.4. Konuşmacı Tanıma

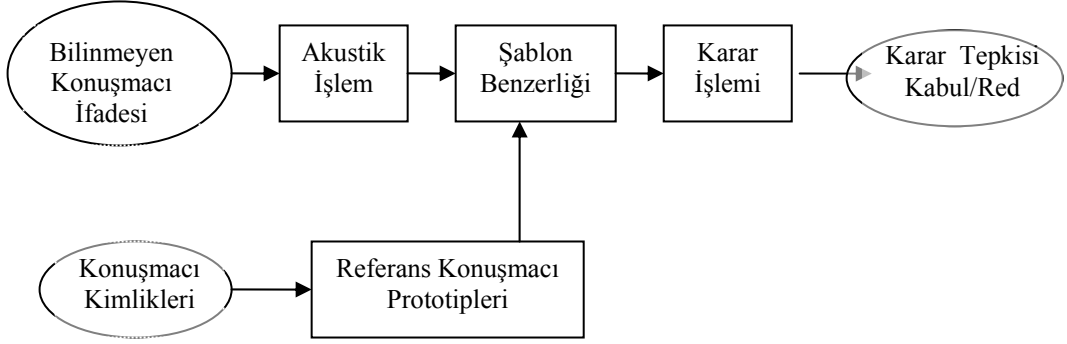
Konuşmacı tanıma, ses dalgalarının içerdiği bireysel bilgiden faydalanılarak konuşmacının otomatik tanındığı bir işlemdir. Tanımlama işlemlerinde şekil 2.2’de gösterilen basamaklar yapılır.



Şekil 2.2. Konuşmacı tanımlama

Konuşmacı tanımlama sistemlerinin kullanıldığı alanlar; telekom, sesli arama, bilgi servisleri, sesli elektronik posta, sesli yetkilendirme, bilgisayarlara uzaktan erişim, güvenlik kontrol, adli uygulamalardır.

Konuşmacı doğrulama sistemlerinin temel yapıları şekil 2.3’deki blok diyagramlarla gösterilmiştir. Konuşmacı tanımlama, kayıtlı konuşmacılar arasında konuşmacı belirleme, konuşmacı doğrulama da kabul veya red işlemidir. Sesin anahtar gibi şahsın belirlenmesinde kullanıldığı bir çok uygulama konuşmacı doğrulama olarak sınıflandırılır.



Şekil 2.3. Konuşmacı doğrulama blok diyagramı

2.5. Konuşmacı Tanıma Sistemlerinin Sınıflandırılması

Konuşmacı tanıma, tanımlama ve doğrulama olarak ikiye ayrılır. Her ikisinde de kayıt aşaması, sistemde N bilinen konuşmacının bir dizi referans kalıp veya modeli, model eşleme algoritmaları ve mantıksal karar vardır.

Konuşmacı tanıma söylenen metnin kayıtlı hangi konuşmacı tarafından söylendiğinin belirlenmesi işlemidir. Konuşmacı tanıma sistemleri kullanılan metine ve konuşmacıya göre sınıflandırılır (Rabiner et al. 1993).

Metne bağlı konuşmacı tanıma sistemleri ikiye ayrılır:

- i) Metne Bağlı Konuşmacı Tanıma: Eğitim ve test aşamasında kullanılan metin biliniyor veya aynıdır.
- ii) Metinden Bağımsız Konuşmacı Tanıma: Kullanılan eğitim ve test metinleri aynı değildir.

Konuşmacıya bağlı olarak konuşmacı tanıma sistemlerinin sınıflandırılmasında karşımıza kapalı set ve açık set olmak üzere iki durum ortaya çıkar.

- i. Kapalı Set: Bilinen N konuşmacı içinden kayıtlı konuşmacının tanınmasıdır.
- ii. Açık Set: Bilinmeyen konuşmacı için referans modelinin bulunmadığı durumdur. Tanıma işleminin test aşamasında, eğitim aşamasında oluşturulan referans şablonu bulunmayan konuşmacıyla karşılaşılabileceği için açık set olarak tasarlanan sistemlerde alternatif bir karar aşaması eklemek gerekir. Bu karar konuşmacının sisteme kayıtlı olup olmadığının bir göstergesidir.

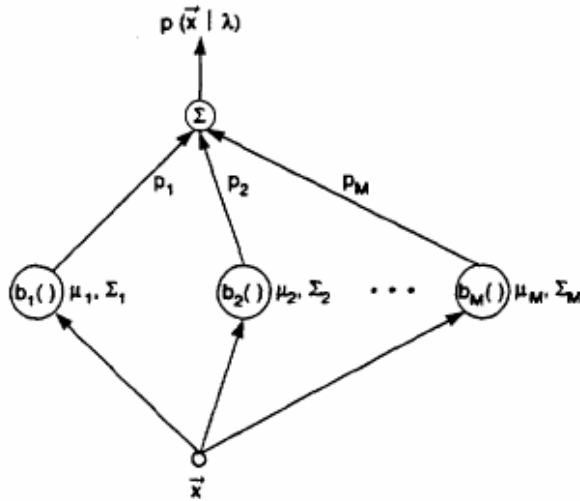
2.6. Metinden Bağımsız Konuşmacı Tanıma Metodları

Metinden bağımsız konuşmacı tanıma sistemlerinde kullanılan iki temel metod Gaussian karma modeli veya vektör nicemlemedir.

2.6.1. Gaussian karma modeli

Bu modelde Gauss bileşenlerinin spektral şekillerinin konuşmacıya bağlı olduğu gösterilmiştir.

2.6.1.1. Model tanımı ve açıklaması



Şekil 2.4. M bileşenli Gaussian yoğunluk

Gaussian karma yoğunluk fonksiyonu şekil 2.4'de gösterildiği gibi M bileşenli yoğunluk fonksiyonunun ağırlıklandırılmış toplamıdır ve 2.1 eşitliği ile gösterilir:

$$p(\vec{x} | \lambda) = \sum_{i=1}^M p_i b_i(\vec{x}) \quad (2.1)$$

Bu eşitlikte \vec{x} D-boyutlu rastgele bir vektör, $b_i(\vec{x})$, \vec{x} rastgele değişkenin i. kitle için ortak yoğunluk fonksiyonu ve p_i 'lerde karma ağırlıklarıdır ($i=1,2,\dots,M$). Her bir kitle için \vec{x} rastgele değişken dağılımı, ortalama vektörü $\vec{\mu}_i$ ve kovaryans matrisi Σ_i olan çok değişkenli Gaussian dağılımlarıyla ifade edilir. Karma ağırlıkları $\sum_{i=1}^M p_i = 1$ şartını sağlarlar. $||$ işareti, determinantı göstermek üzere, i. kitlenin ortak olasılık yoğunluk fonksiyonu 2.2 eşitliğiyle verilir.

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\vec{x} - \vec{\mu}_i)' \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i)\right\} \quad (2.2)$$

Tam bir Gaussian karma yoğunluk fonksiyonu; her bir yoğunluk fonksiyonunun ortalama vektörleri, kovaryans matrisleri ve karma ağırlıklarıyla parametrelendirilir. Bu parametrelendirmenin ortak gösterimi 2.3 eşitliği ile yapılmaktadır:

$$\lambda = \left\{ p_i, \mu_i, \Sigma_i \right\} \quad i=1,\dots,M. \quad (2.3)$$

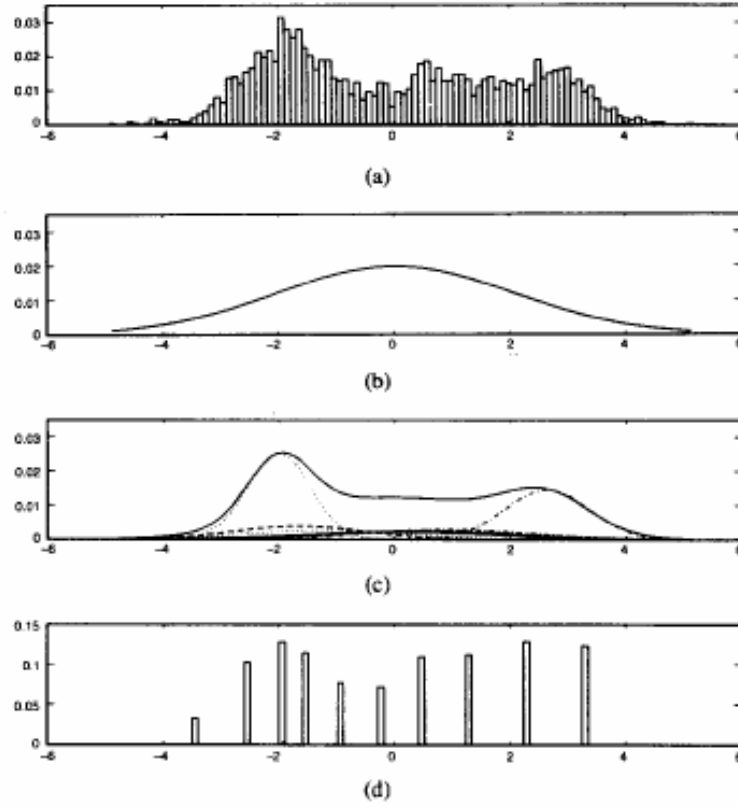
Konuşmacı tanımında her bir konuşmacı λ ile gösterilen bir Gaussian karma modeliyle ifade edilir.

Gaussian karma modelinin kovaryans matrisini seçimine göre birçok çeşidi vardır. Modelin her bir Gaussian bileşen için kovaryans matrisi(Nodal Kovaryans) olabileceği gibi, bir konuşmacının tüm Gaussian bileşenleri için bir kovaryans matrisi(Grand Kovaryans) veya tüm konuşmacılar tarafından ortak kullanılan bir kovaryans matrisi(Global Kovaryans) de olabilir.

Gaussian karma yoğunluklarını konuşmacı kimliğinin göstergesi olarak kullanmanın iki temel nedeni vardır. Bunlardan birincisi Gaussian karma modeli gibi çok modelli yoğunlukların kişisel bileşen yoğunluklarının akustik sınıf setini modelleyebilmesine dayanan sezgisel kavramdır. Bir konuşmacının sesine ilişkin akustik uzay; sesliler, nazal veya sürtüşmeli harfler gibi bazı boğaz olaylarını gösteren akustik sınıf gruplarıyla karakterize edilebilir. Bu akustik sınıflar konuşmacı kimliğini tanımlamak için gerekli olan konuşmacıya bağlı ses yolu konfigürasyonlarını yansıtır. i . akustik sınıfın spektral biçimi, i . bileşen yoğunluğu ortalaması μ_i ile ve ortalama spektral biçimdeki değişimler de kovaryans matrisi Σ_i ile gösterilebilir. Tüm test veya eğitim konuşmaları etiketlenmediği için akustik sınıflar bilinmeyen gözlem sınıfının içinde saklı olur. Bu akustik sınıflardan elde edilen öznitelik vektörlerinin gözlem yoğunlukları Gaussian karma yoğunluk fonksiyonlarını oluşturur.

Konuşmacı tanımda Gaussian karma yoğunluğu kullanmanın ikinci nedeni ise geniş örnek dağılımları gösterimlerinin, temel Gaussian fonksiyonlarının doğrusal kombinasyonları ile yapılabilmesine dayanan deneysel gözlemlerdir(Reynolds et al. 1995). Gaussian karma modelin en önemli özelliği keyfi biçimli dağılım yoğunluklarının düzgün kestirimlerini yapabilmesidir. Unimodal Gaussian konuşmacı tanımda, konuşmacı öznitelik vektörlerinin gösterimi bir ortalama vektör ve kovaryans matrisi ile yapılırken vektör nicemlemede ise ayrık karakteristik şablonlarla bu gösterim yapılmaktadır. Bazı durumlarda modelleme performansını iyileştirmek için ayrık Gaussian fonksiyonlarını kullanan Gaussian karma modeli, vektör niceme teknikleriyle karma bir model yaratmada kullanılabilir. Şekil 2.5'de Unimodal Gaussian model, Gaussian karma modeli ve vektör niceme modelleri karşılaştırılmaktadır. Şekil 2.5.a'da bir erkek konuşmacının 25 saniyelik konuşmasından elde edilen cepstral katsayıların histogramı, şekil 2.5.b'de en çok olabirlik unimodal Gaussian modeli,

şekil 2.5.c'de Gaussian karma modeli ve içerdiği 10 bileşen yoğunluğu gösterilmektedir. Şekil 2.5.d'de 10 elemanlı kod kitaba sahip vektör nicemleme ağırlık merkezlerine karşılık gelen verinin histogramı görülmektedir.



Şekil 2.5. Unimodal Gaussian, karma ve vektör nicemleme modellerine ait karşılaştırma grafikleri

Tüm olasılık yoğunluk fonksiyonu modelini oluşturmak için Gaussian bileşenler birlikte kullanılacağı için, öznelik vektörleri istatistiksel olarak bağımsız değilse kovaryans matrisinin elemanlarını hepsi gerekli değildir. Öznelik vektör elemanları arasındaki korelasyonu modellemek için diyagonal kovaryans matris elemanlarının lineer birleşiminden faydalanılabilir.

2.6.1.2. En çok olabilirlik parametre tahmin yöntemi

Konuşmacı model eğitimin amacı, verilen bir konuşmacının eğitim datasını kullanarak konuşmacı eğitim öznelik vektörlerinin dağılımına en iyi karşılık gelen Gaussian

karma model parametrelerinin, λ 'nın, tahminidir. Bu parametrelerin tahmini için birçok metod vardır. En yaygın olarak kullanılan metod, en çok olabilirlik tahmin yöntemidir(Maximum Likelihood Estimation, MLE).

En çok olabilirlik tahmin yönteminin amacı verilen eğitim datası için Gaussian karma modelinin olabilirlik fonksiyonunu maksimize eden model parametrelerini bulmaktır. T adet eğitim vektöründen oluşan $X = \left\{ \vec{x}_1, \dots, \vec{x}_T \right\}$ dizisi için Gaussian karma model olasılık fonksiyonu 2.4 eşitliğindeki gibi yazılabilir:

$$p(X|\lambda) = \prod_{t=1}^T p(\vec{x}_t|\lambda) \quad (2.4)$$

Bu ifade λ parametresinin doğrusal olmayan bir fonksiyonudur ve doğrudan maksimizasyonu mümkün değildir. Bununla birlikte, en çok olabilirlik parametre tahmini, beklenti maksimizasyon(Expectation Maximization, EM) algoritmasının özel bir durumu kullanılarak iteratif bir yolla elde edilebilir.

λ başlangıç modeli ile başlayan beklenti maksimizasyon algoritmasının temeli $p(X|\bar{\lambda}) \geq p(X|\lambda)$ olacak şekilde yeni bir $\bar{\lambda}$ modelinin kestirimine dayanır. Bir sonraki iterasyon için bulunan yeni model başlangıç modeli kabul edilir ve bir yakınsama seviyesine ulaşıncaya kadar işlem tekrarlanır. Bu Baum-Welch tekrar kestirim algoritmasıyla HMM parametrelerinin elde edilmesinde kullanılan teknikle aynıdır.

i. akustik sınıf için sonsal olasılık 2.5 eşitliği ile verilir:

$$p(i|\vec{x}_t, \lambda) = \frac{p_i b_i(\vec{x}_t)}{\sum_{k=1}^M p_k b_k(\vec{x}_t)} \quad (2.5)$$

Gaussian karma konuşmacı tanımada model eğitimindeki iki kritik faktör, M karma sayısını seçmek ve beklenti maksimizasyon algoritmasından önce model parametrelerinin kullanıma hazırlanmasıdır. Bunlar deneysel yöntemlerle en iyi şekilde belirlenebilir.

2.6.1.3. Konuşmacı tanıma

Konuşmacı tanıma için $S = \{1, 2, \dots, S\}$ şeklindeki S adet konuşmacı $\lambda_1, \lambda_2, \dots, \lambda_S$ Gaussian karma modelleriyle gösterilir. Amacımız verilen gözlem dizisi için maksimum sonsal olasılığı veren konuşmacı modelinin bulunmasıdır. Biçimsel olarak,

$$\hat{S} = \arg \max p(\lambda_k | X) = \arg \max_{1 \leq k \leq S} \frac{p(X | \lambda_k) p(\lambda_k)}{p(X)} \quad (2.6)$$

Eşitliğin ikinci tarafına Bayes kuralı kullanılarak geçilir. Konuşmacı olasılıklarının eşit olduğu kabul edilip ($p(\lambda_k) = 1/S$) ve $p(X)$ 'in de tüm konuşmacılar için aynı olduğu bilindiğinden, sınıflandırma kuralı basite indirgenebilir :

$$\hat{S} = \arg \max_{1 \leq k \leq S} p(X | \lambda_k) \quad (2.7)$$

2.6.2. Vektör nicemeleme tekniklerine dayalı konuşmacı tanıma

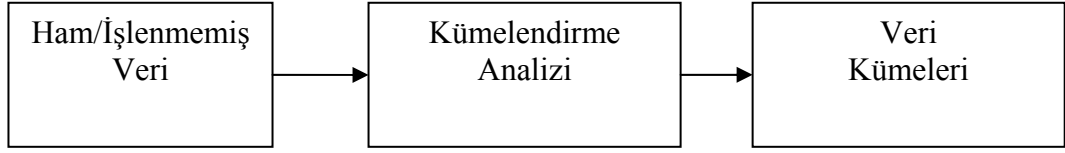
Vektör nicemeleme, verideki bilgi miktarını azaltmaya yarayan bir tür kümelendirme analiz yöntemidir.

2.6.2.1. Kümelenendirme analizi

Tryon tarafından ilk olarak 1939 yılında kullanılan kümelenendirme analizi benzer nesnelerin kendilerine ait olan kategorilere gruplandırmasını yapmak için çeşitli sayıdaki değişik algoritmaları ihtiva eder. Araştırmalarda karşılaşılan temel sorun gözlenen verinin anlamlı verilere dönüşümünün nasıl yapılacağı yani sınıflandırma ilminin geliştirilmesidir. Başka bir deyişle kümelenendirme analizi, amacı değişik nesnelere aynı gruba ait olduklarında maksimum, aksi takdirde farklı gruplara ait olduklarında da minimum ilişki verecek şekilde gruplara ayıracak keşifçi veri analiz aracıdır(Kinnunen et al. 2000). Tüm bunların ışığında kümelenendirme analizi herhangi bir açıklama ve yorumlama gerektirmeden verideki yapıları keşfetmemiz amacıyla kullanılabilir. Yani, temel olarak kümelenendirme analizi verideki yapıyı varoluş nedeni açıklamadan keşfeder.

Kümelenendirme ile günlük hayatın her alanında karşılaşmaktayız. Örneğin yiyecek satılan marketlerde aynı gruba ait et veya meyveler aynı yerlerde olacak şekilde yerleştirilir. Kümelenendirme hayatımızda üstlendiği önemli rolü anlatabilecek sayısız örnek vardır. Bir başka örnek olarak biyologların hayvanlar arasındaki farklılıkların anlamlı bir tanımını yapabilmek için hayvan türlerini düzenlemelerini verebiliriz. Biyolojide kullanılan modern sistemlere göre insanlar maymun, memeli, omurgalı ve hayvan sınıfına ait olarak tanımlanır. Ait olunan sınıfın derecesi arttıkça ilgili sınıfa benzerliğin azaldığı gözlenir. İnsanların maymunlarla, memelilerle olduğundan daha çok benzerliği vardır. Kısacası mesleğimiz ne olursa olsun karşımıza bu veya şu şekilde bir kümelenendirme analiz problemi çıkabilir.

Kümelenendirme analizi bazı benzerliklere dayanarak bileşenlerin gruplarını bulmaya çalışır. Kümelenendirme analizi aynı zamanda veri setlerinin kitle merkezini bulmak için kullanılır. Kümeyle ait olup olmadığını belirlemek için çoğu algoritma bir nokta ile kitle merkezi arasındaki uzaklığı değerlendirir. Temel olarak bir kümelenendirme algoritmasının çıkışı her bir kümedeki bileşen sayısı ile küme kitle merkezlerinin istatistiksel tanımını verir.



Şekil 2.6. Kümelenendirme tanımı

Gerçekte iyi tanımlanmış benzerlik kurallarına göre nesnelere kümelere koyan değişik kümelenendirme algoritmalarından oluşan kümelenendirme analizi tipik bir istatistiksel test değildir. Diğer bazı istatistiksel yöntemlerin aksine kümelenendirme analizleri genellikle önsel, apriori hipotezimiz olmadığı zamanlar kullanılır. Kümelenendirme analizi olası en anlamlı sonucu bulur.

2.6.2.1.1. Birleştirme/bağlama (ağaç kümelenendirme)

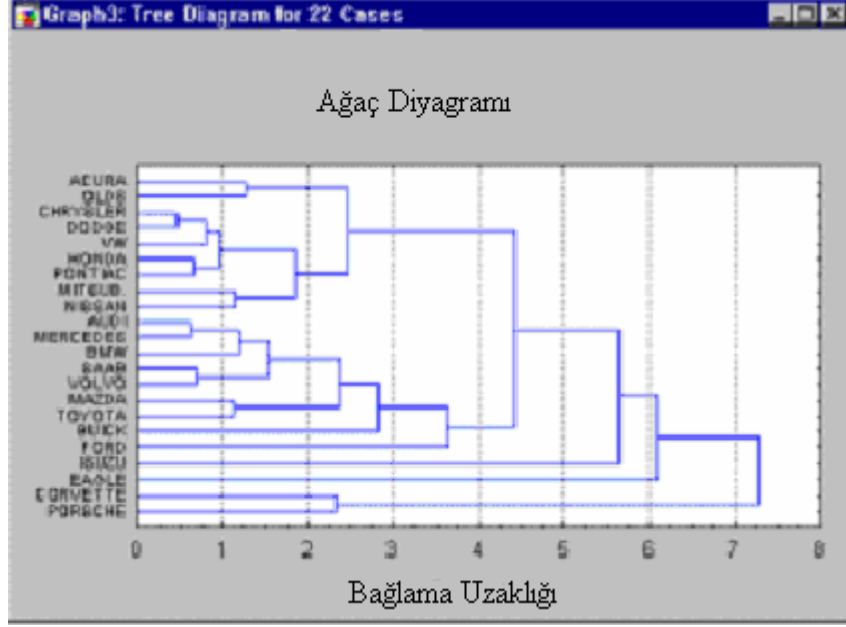
Bu algoritmanın amacı, bazı benzerlik veya uzaklık ölçümlerini kullanarak nesnelere geniş kümelere birleştirmektir. Bu çeşit kümelenendirmenin en temel örneği hiyerarşik ağaçtır.

2.6.2.1.2. Hiyerarşik ağaç

Şekil 2.7’de gösterildiği gibi çizimin sol tarafında her bir nesnenin kendisinin bir sınıf olduğunu varsayarak başladığımız yatay hiyerarşik bir ağaç çizimimiz olsun. Eşik seviyemizi iki veya daha fazla nesnenin aynı kümenin elemanı olduğunu tanımlamamıza ilişkin olarak düşürelim.

Sonuç olarak daha fazla nesneyi birbirine bağlamakta ve farklı elemanlardan oluşan daha geniş kümelere biraraya getirmekteyiz. En son basamakta tüm nesnelere birbirine bağlanmıştır. Bu çizimde yatay eksen bağlantı uzaklığını göstermektedir. Dolayısıyla yeni bir kümenin oluştuğu her bir düğüm için yeni bir küme oluşturmak amacıyla bağlanan elemanların ölçüt uzaklığını okuyabiliriz. Veri, benzer nesnelere kümelere göre açık bir yapı oluşturduğu zaman, bu yapı bağımsız hiyerarşik ağaçta bir dal olarak

yansıtılır. Bağlama yöntemiyle yapılan başarılı bir analiz sonucu kümeler ortaya çıkarılabilir ve bu dallar yorumlanabilir.



Şekil 2.7. Hiyerarşik ağaç

2.6.2.1.3. Uzaklık ölçümü

Bağlama veya ağaç kümelendirme yöntemi, kümeleri oluştururken nesnelere arasındaki farklılığı (benzerliği) ve uzaklıkları kullanır. Benzerlikler, ögeleri gruplandırma veya ayırma kriteri olarak hizmet eden kurallar bütünüdür. Bu uzaklık veya benzerlikler her biri nesnelere gruplandırmak için birer koşul veya kurala dayanmaktadır. Örneğin hazır yiyecekleri gruplandıracaksak içerdikleri kalori miktarını, fiyatlarını, kişisel tat sınıflandırmalarını hesaba katmamız gerekir. Çok boyutlu uzayda nesnelere arasındaki uzaklığı hesaplamanın en doğru yolu euclid uzaklığıdır. 2 veya 3 boyutlu uzayda bu uzaklık gerçek geometrik uzaklık olur. Bununla beraber bağlama algoritmaları hesaplanan uzaklığın gerçek uzaklık veya daha anlamlı gelen türetilmiş uzaklık olup olmadığıyla ilgilenmez; araştırmasına göre doğru yöntemin seçimi araştırmacıya bağlıdır.

2.6.2.1.3.1. Euclid uzaklığı

En çok tercih edilen uzaklık ölçüsüdür. Temel olarak çok boyutta geometrik uzaklığın karşılığıdır. 2.8 eşitliği ile hesaplanır:

$$d_E(x,y) = \sqrt{\sum_{i=1}^{\text{Vektör Boyutu}} (x_i - y_i)^2} \quad (2.8)$$

2.6.2.1.3.2. City-block (manhattan) uzaklığı

Temel olarak bu uzaklık boyutlar arasındaki ortalama uzaklıktır. Çoğu durumlarda bu uzaklık euclid uzaklığıyla aynı sonucu verir. Bununla beraber bu ölçümde, aykırı değerlerin etkisi kareleri alınmadığı için azaltılır. 2.9 eşitliği ile hesaplanır:

$$d_M(x,y) = \sum_{i=1}^{\text{Vektör Boyutu}} |x_i - y_i| \quad (2.9)$$

2.6.2.1.4. Bağlantı kuralları

İlk aşamada her bir nesne kendi kümesini temsil ettiği zaman, bu nesneler arasındaki uzaklık, uzaklık ölçüsünün seçimiyle tanımlanır. Bununla beraber çeşitli nesneler birkez bağlandığında bu yeni kümeler arasındaki uzaklığı nasıl belirleriz? Başka bir deyişle iki kümenin verimli olarak ne zaman birbirlerine bağlanacağını belirlemek için bağlantı kuralına ihtiyacımız vardır. Bunun için çeşitli olasılıklar vardır: Örneğin, iki kümedeki iki nesne birbirlerine bağlantı uzaklığından daha yakınsa iki küme birbirine bağlanabilir. Tek bağlantı olarak adlandırılan kümeler arasındaki uzaklığı belirlemek için en yakın komşu kullanılabilir(Ong et al. 1996). Alternatif olarak kümeler etrafında birbirlerinden en uzak olan komşular kullanılabilir. Bu yöntemde tam bağlantı olarak adlandırılır.

2.6.2.1.4.1. Tek bağlantı(en yakın komşu)

Bölüm 2.6.2.1.4'de bahsedildiği gibi bu yöntemde iki küme arasındaki uzaklık, farklı iki kümedeki en yakın iki nesne arasındaki uzaklıkla belirlenir. Bu kural bir anlamda nesnelere küme oluşturacak şekilde dize ve sonucunda oluşan kümeler uzun zincir oluşturmaya yönelirler.

2.6.2.1.4.2. Tam bağlantı(en uzak komşu)

Bu yöntemde, kümeler arasındaki uzaklık, farklı iki kümedeki en uzak iki nesne arasındaki uzaklıkla belirlenir. Bu yöntem nesnelere aslında ayrı kümeler oluşturacağı zaman daha iyi performans gösterir.

2.6.2.1.5. K-ortalama kümelendirme

K-ortalama, kümelendirme problemlerini çözmek için kullanılan en temel eğitimli olmayan öğretim algoritmalarından biridir. Verilen data setini belirli sayıda küme (K adet küme olduğunu varsayarsak) sınıflandırmak için temel ve basit bir yöntem kullanır. Temel düşünce her bir küme için K adet küme merkezi tanımlamaktır. Değişik yerlerdeki küme merkezleri değişik sonuçlara yol açacağı için bunları tanımlarken seçilecek yola dikkat etmek gerekir. En iyi seçim her birini birbirinden en uzak olacak şekilde seçmektir. Daha sonraki aşamada verilen veri setine ait her nokta alınır ve en yakın küme merkezine atanır. Karara bağlanmamış herhangi bir nokta kalmadığında ilk aşama tamamlanmış ve ilk kümeleme yapılmış olur. Bu noktada önceki aşamadan kalan kümelerin ağırlık merkezi olarak k adet yeni küme merkezi hesaplamamız gerekir. Bu K adet yeni küme merkezini elde ettikten sonra aynı veri seti noktaları arasında yeni kümeleme ve en yakın yeni küme merkezi hesabı yapılır. Böylece bir döngü kurulmuş olur. Bu döngünün sonucunda K adet küme merkezinin, hiçbir değişiklik olmayıncaya kadar adım adım yerlerini değiştirdiği gözlenir. Başka bir deyişle artık küme merkezleri hareket etmiyordur. Son olarak bu algoritma, hata kare fonksiyonu olarak tanımlanan amaç fonksiyonunu minimize etmeyi amaçlar. Amaç fonksiyonu, 2.10 eşitliği ile verilir.

Burada k küme sayısı, c_j j . kümenin merkezi, m toplam veri sayısı ve x_i , i . veriyi göstermektedir. $\| \cdot \|$, L2 normunu göstermektedir.

$$\sum_{j=1}^k \sum_{i=1}^m \|x_i - c_j\|^2 \quad (2.10)$$

Algoritma aşağıdaki basamaklardan oluşur:

1. Kümelendirilmiş nesnelere gösterilen uzaya K nokta yerleştirilir. Bu noktalar başlangıç kitlelerin merkezlerini gösterir.
2. Her nesne en yakın kitle merkezine atanır.
3. Tüm nesnelere atandığı zaman, K kitle merkezi konumları tekrar hesaplanır.
4. Kitle merkezi konumlarında değişim olmayıncaya kadar 2. ve 3. basamaklar tekrar edilir.

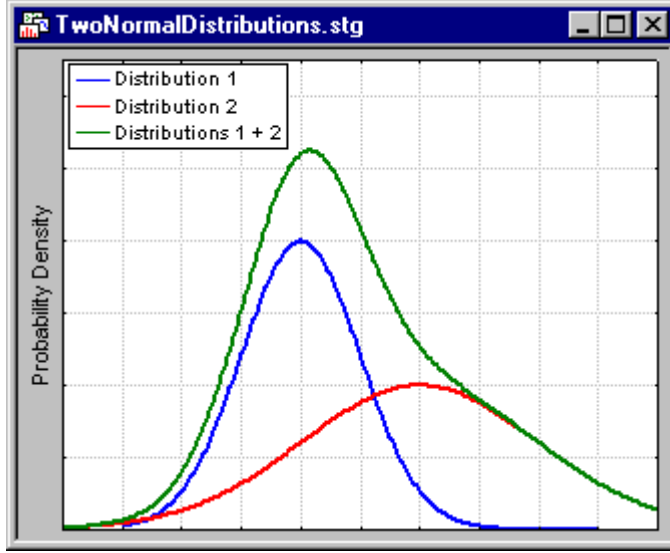
Bu yöntem sonlanacak olmasına rağmen K -ortalama algoritma, global amaç fonksiyonunla ilgili olarak optimum konfigürasyonu bulmaz. Algoritma aynı zamanda rastgele olarak seçilen başlangıç küme merkezlerine duyarlıdır. K -ortalama algoritması bu etkiyi azaltmak için bir çok kere çalıştırılabilir.

2.6.2.1.6. Beklenti maksimizasyon algoritması

Ortalamaları ve standart sapmaları farklı Gaussian dağılıma sahip iki kitleden çok sayıda örnekler aldığımızı kabul edelim.

Şekil 2.8'de, ortalamaları ve standart sapmaları farklı Gaussian dağılımların ve dağılımlardan oluşan karma dağılımın olasılık yoğunluk fonksiyonlarının grafik çizimi görülmektedir. Sadece ortalamaları ve standart sapmaları farklı iki Gaussian karma dağılımdan gözlemlerin alındığı durum ele alınacaktır. Beklenti maksimizasyon kümelendirmenin amacı, her bir kümenin ortalama ve standart sapmasını gözlemlenen data ile benzerliğini maksimize edecek şekilde tahmin etmektir. Başka bir deyişle

beklenti maksimizasyon algoritması deęişik kümelerdeki deęişik daęılımların karışımlarına dayanan gözlemlenmiş daęılım deęerlerini tahmin etmeye çalışır. Bazı bilgisayar programlarında beklenti maksimizasyon(EM) algoritması uygulanmalarının gerçekleştirilmesinde normal, log-normal ve poisson daęılımları gibi deęişik daęılımlar seçilebilir.



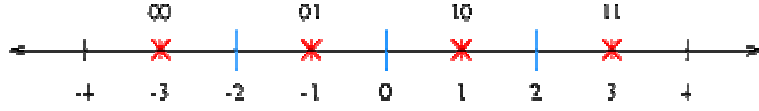
Şekil 2.8. Daęılım karışımları

Beklenti maksimizasyon kümelendirme algoritmasıyla K-ortalama algoritması kıyaslandığı zaman, K-ortalama algoritması kümeler arasındaki uzaklığı maksimum yapacak şekilde gözlemleri kümelere atar. EM algoritması gözlemleri kümelere atama hesabı yapmaz fakat gözlemler için sınıflandırma olasılığını hesaplar. Her bir gözlem belli bir olasılıkla her bir kümeye aittir. Gözlemler en büyük sınıflandırma olasılığını verdikleri kümelere atanır.

2.6.2.2. Vektör nicemleme

Vektör nicemleme verideki bilgi miktarını azaltmaya yarayan bir tür data sıkıştırma metodudur. Geçmişte çok boyutlu integrasyona ihtiyaç duyulduğu için vektör nicemleme tasarımı büyük bir problem olarak görülüyordu. 1980'lerde Linde, Buzo ve Gray eğitim dizisine dayanan bir vektör nicemleme algoritması önerdiler.

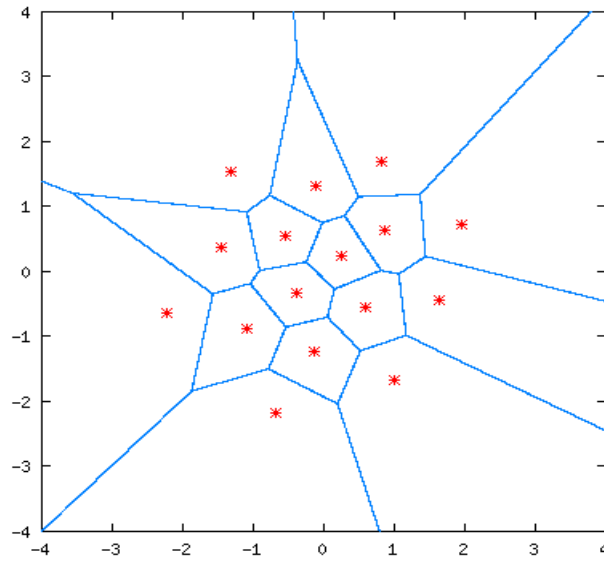
Vektör nicemleyicide aslında yapılan işlem yaklaştırmadır. 1-boyutlu bir vektör nicemleyici şekil 2.9’da gösterilmektedir.



Şekil 2.9. 1-Boyutlu vektör nicemleme

Burada -2’den küçük her sayı -3’e; -2 ile 0 arasındaki her sayı -1’e; 0 ile 2 arasındaki her sayı 1’e, 2’den büyük her sayı 3’e yaklaştırılmıştır.

Şekil 2.10’da iki boyutta bir vektör nicemleyici gösterilmektedir. İki boyutlu vektör nicemlemede, belirli bir bölgeye düşen sayı çiftleri o bölgeyi temsil eden *’a yaklaştırılmıştır. 16 bölge ve 16 yıldız vardır. *’lar kod vektörü ve *’a yaklaştırılan bölgeler kodlama bölgesi olarak adlandırılır. Kod vektörlerinin oluşturduğu gruba kod kitabı denir.



Şekil 2.10. 2-Boyutlu vektör nicemleme

2.6.2.3. Vektör nicemleme tasarımı

M adet kaynak vektöründen oluşan bir eğitim dizimizin olduğunu varsayalım.

$$T = \{x_1, x_2, \dots, x_M\} \quad (2.11)$$

Bu eğitim dizisi büyük bir veri tabanından elde edilebilir. Örneğin kaynak konuşma sinyali ise, eğitim dizisi birçok uzun telefon görüşmesi kaydedilerek elde edilebilir.

Kaynak vektörleri -k- boyutlu olsun.

$$x_m = (x_{m,1}, x_{m,2}, \dots, x_{m,k}), \quad m = 1, 2, \dots, M \quad (2.12)$$

Kod vektör sayısı N olsun.

$$C = \{c_1, c_2, \dots, c_N\} \text{ 'de kod vektörlerini gösterebiliriz.} \quad (2.13)$$

c_N kod vektörleriyle ilgili kodlama bölgesi S_N olsun ve uzayın bölümlerini gösterebiliriz.

$$P = \{S_1, S_2, \dots, S_N\} \quad (2.14)$$

Eğer x_m kod vektörü S_N bölgesi içinde ise x_m kod vektörünün yaklaşımı c_N ' dir ve $Q(x_m) = c_N, x_m \in S_N$ şeklinde gösterilir.

Hata kare bozulma ölçümü kullanıldığını varsayılırsa, ortalama ve toplam bozulma :

$$D_{\text{Ort}} = \frac{1}{M} \sum_{m=1}^M \|x_m - Q(x_m)\|^2 \quad (2.15)$$

$$\|e\|^2 = e_1^2 + e_2^2 + \dots + e_k^2 \quad (2.16)$$

Vektör nicemleme tasarım problemi şöyle ifade edilir: Verilen eğitim dizisi vektörleri(T) ve kod vektör sayısı (N) için öyle kod vektörleri ve kodlama bölgesi (S_N) bulunmalı ki D_{Ort} minimum olsun(Linde et al. 1980).

Bunun için iki kriter sağlanmalı :

i) En Yakın Komşu Koşulu

$$S_N = \left\{ x : \|x - c_n\|^2 \leq \|x - c_{n'}\|^2, \forall n' = 1, 2, \dots, N \right\} \quad (2.17)$$

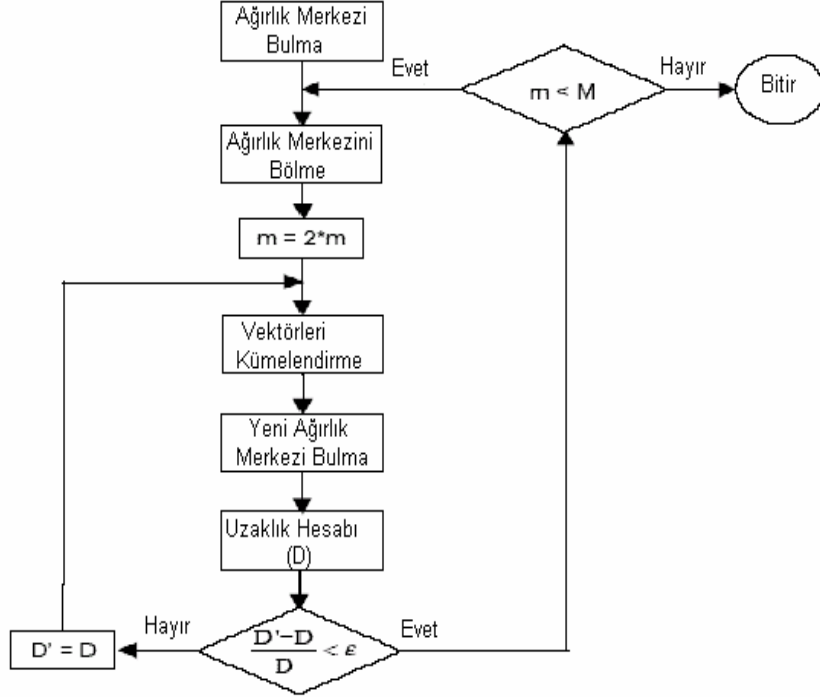
Bu şart bize S_N şifreleme bölgesinin, c_n ' e yakın olan tüm kod vektörlerini içerdiğini gösterir.

ii) Ağırlık Merkezi Koşulu

$$c_n = \frac{\sum_{x_m \in S_N} x_m}{\sum_{x_m \in S_N} 1}, n = 1, 2, \dots, N \quad (2.18)$$

Bu şartta bize c_n kod vektörünün S_N bölgesindeki tüm eğitim vektörlerinin ortalaması olduğunu gösterir.

2.6.2.3.1. Linde, Buzo ve Gray algoritması



Şekil 2.11. LBG algoritması akış diyagramı

Optimum kriterlerini (En yakın komşu ve Ağırlık merkezi) iteratif bir yolla çözen algoritmadır. Algoritmanın bir başlangıç kod kitabına ihtiyacı vardır.

- i) Başlangıç kod vektörü tüm eğitim setinin ortalaması kabul edilir.
- ii) Daha sonra kod vektörü ikiye bölünür. İteratif algoritma bu iki vektörü başlangıç kod kitabı olarak kabul edip çalışır.

$$c_{\bar{N}} = c_N * (1 - \varepsilon) \quad (2.19)$$

$$c_N^+ = c_N * (1 + \varepsilon) \quad (2.20)$$

- iii) Herbir eğitim vektörü, euclid uzaklığı kullanılarak mevcut kod kitabı içindeki en yakın olduğu kod vektörünü bulunup onun etrafında kümelendirilir.
- iv) Atanan yeni eğitim vektörleri göz önüne alınarak ağırlık merkezi güncellenir.
- v) İstenilen sayıda kod vektörü elde ediliyince ve ağırlık merkezleri değişmeyince kadar bu iterasyonlar gerçekleştirilir.

2.6.2.4. Konuşmacı tanımda vektör nicemleme kullanımı

Vektör nicemlemeye dayalı konuşmacı tanıma sistemlerinde eğitim dataları kullanılarak herbir konuşmacının öznitelik vektörleri çıkarılır. Öznitelik vektörleri kişiden kişiye değişen özelliklerdir. İnsanların gırtlak yapısından kaynaklanan ve kişileri ayırt etmeye yarayan niteliklerdir. Konuşmacı tanıma sistemlerinde doğrusal öngörüm katsayıları veya mel-ölçeklendirilmiş cepstral katsayılar kullanılarak konuşmacılar tanınmaya çalışılır. Herbir konuşmacının öznitelik vektörleri çıkarıldıktan sonra, bu vektörler kümelendirme algoritmalarından biri kullanılarak eğitilir ve herbir konuşmacı için optimum referans model yani kod kitabı tasarlanır.

Konuşmacıyı tanımak için konuşmacının test dataları kullanılır. Bu datalarında öznitelik vektörleri çıkarıldıktan sonra referans model olarak hazırlanan kod kitabı vektörleriyle en yakın uzaklığı veren kod vektörleri bulunur. Bilinmeyen konuşmacı belirlenen karar kriterine göre konuşmacılardan birine atanır(Picone et al. 1993).

2.7. Metne Bağlı Konuşmacı Tanıma Metodları

Metine bağlı konuşmacı tanıma sistemlerinde kullanılan iki temel metod Saklı Markov Model veya Dinamik Zaman Eğrisidir.

2.7.1. Saklı markov model

Saklı Markov Modeli, konuşma işleme uygulamalarında kullanılan temel yöntemlerden biridir. Rastgele bir süreç olarak kabul edilen ve bu süreç parametrelerinin iyi

tanımlanmış yöntemlerle doğru olarak tahmin edilebileceği bu modelde, çerçevelenen konuşma sinyalinin spektral özellikleri istatistiksel yöntemlerle karakterize edilir. Saklı Markov Modeli, durumların gizlendiği Markov modelidir.

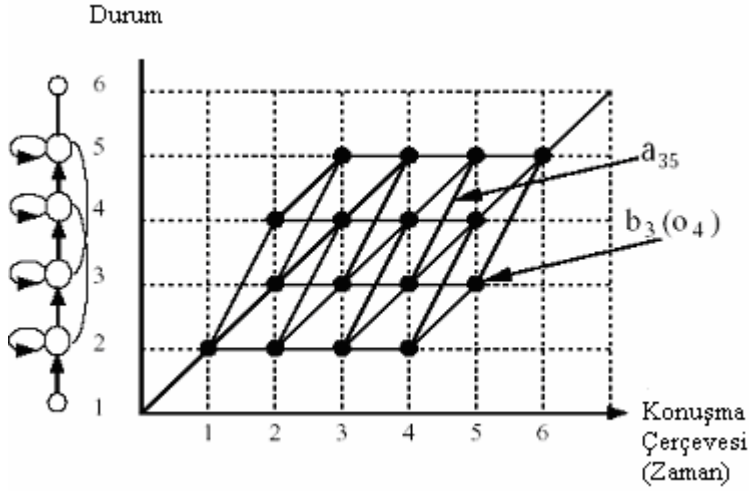
Saklı Markov Modeli, kesikli model, sürekli model veya yarı sürekli model gibi her bir duruma tahsis edilen gözlenebilir olayların sürekli, kesikli veya her ikisi birden olmasına göre sınıflandırılabilir. Durumlar, herhangi bir zaman anında her bir durumdan herhangi bir duruma geçiş içerecek biçimde ergodik olarak tasarlanabilir. Durumların diziyi sonlandırabilmesi için durum dizilişlerine göre soldan sağa veya sağdan sola tasarlanabilir model.

Tüm konuşma işleme uygulamalarında bir eğitim evresi ve bu evreyi takip eden tanıma evresi yer almaktadır. Eğitim evresi süresince temel alınacak referans modelin parametreleri tahmin edilir. Bu modelde tahmin edilecek üç parametre vardır. Bunlardan biri, t anında i . durumda ve $t+1$ anında j . durumda olma geçiş olasılıklarını gösteren a_{ij} elemanlarını içeren durum geçiş olasılığı matrisi A 'dır. $O = \{o_1, o_2, \dots, o_T\}$ olan bir gözlem dizisi tanımlandığı zaman bu vektör dizisinin her bir elemanı, konuşmacı tanıma sistemlerinde kullanılan öznitelik vektörlerini gösterir. $B = [b_j(o_t)]$ gözlem sembol olasılık dağılımı matrisidir. $b_j(o_t)$, o_t vektörünün t anında j . durumda gözlemlenmesi olasılığını gösterir. $\pi = \{\pi_i\}$ vektörü, başlangıçta i . durumda olma olasılığını belirten başlangıç durum dağılımını gösterir. Bu üç parametre, $\lambda = \{A, B, \pi\}$ olan Saklı Markov Modeli oluşturur. Bunların dışında durum sayısı N , her bir durumdaki karma sayısı M de diğer parametrelerdir. Gözlem sembol olasılığını göstermek için çeşitli yöntemler vardır ama bunlardan en çok kullanılanı sürekli olasılık yoğunluk fonksiyonlarıdır.

Eğitim sürecinin başlangıcında Saklı Markov Modelin λ parametrelerinin taslak hesabı yapılır. Saklı Markov Model temelli konuşma tanıma uygulamalarının tanıma evresinde Baum-Welch algoritması yer alır(Rabiner et al. 1989).

Şekil 2.12'de gösterildiği gibi viterbi algoritması matristeki en iyi yolu veya izi bulmak için gözönünde canlandırılabilir. Bir model ve bir gözlem dizisi verildiğinde öyle bir

durum dizisi kullanılsın ki, bu durum dizisiyle gözlem dizisinin oluşturulma olasılığı maksimum olsun. Satırlar durumları sütunlar da konuşma çerçevelerini göstermektedir. Her bir geniş nokta o anda gözlenen durumun logaritmik olasılığını ve her bir yay da durumlardan birbirlerine geçiş olasılıklarının logaritmasını göstermektedir. Herhangi bir yolun logaritmik olasılığı geçtiği noktaların ve yayların olasılıkları toplamına eşittir.



Şekil 2.12. Viterbi algoritmasının matris gösterimi

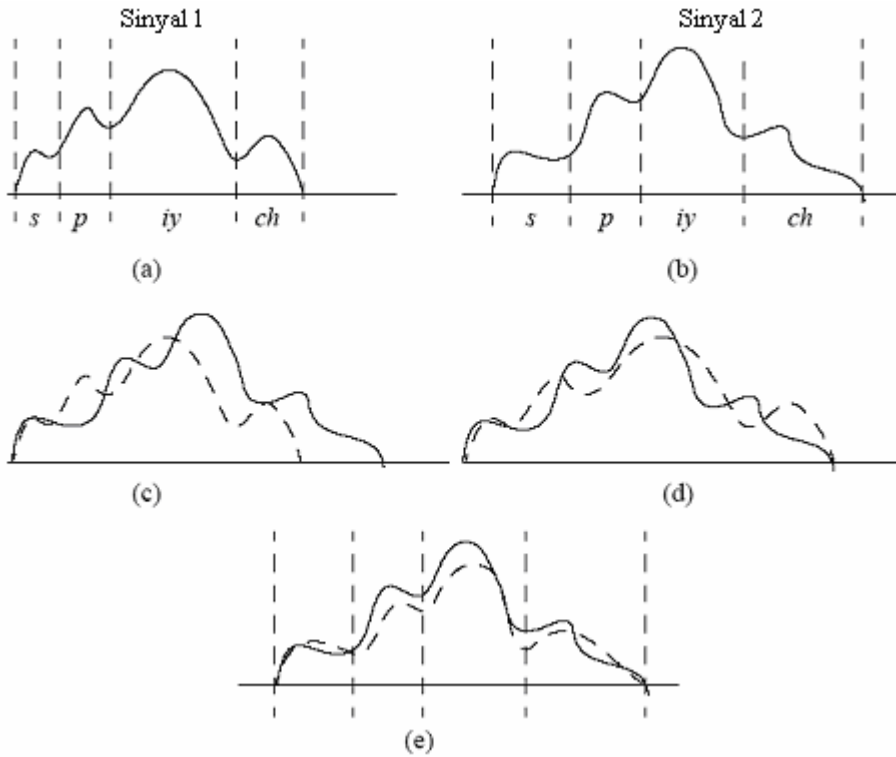
2.7.2. Dinamik zaman eğrisi

Konuşmacı aksanlarından, konuşmacıların konuşma stili gibi nedenlerden dolayı bir konuşmanın dalga şeklinde birçok değişiklikler görülebilir. Konuşmacı aynı olsa bile konuşma hızından, ses yüksekliğinden veya stres gibi nedenlerden dolayı da konuşma dalga şeklinde farklılıklar görülür. Bazı örüntüler yüksek genlik ve uzun süreye sahipken bazıları ise daha düşük genlikli ve kısa sürelidir. Şekil 2.13.a'da ve şekil 2.13.b'de "speech" kelimesinin iki tekrarı görülmektedir. Bu iki sinyal karşılaştırıldığında, birinci sinyalin s, p, ch fonemleri daha kısarken iy fonemi daha uzun olduğu görülür. Birinci sinyal noktalı olarak, ikinci sinyal ise sürekli çizilerek gösterilmiştir. Bu iki sinyali karşılaştırırken karşımıza çıkan temel sorun iki sinyal süresinin eşit olmamasıdır. Konuşma sinyalinin gösterimi öznitelik vektörleri kullanılarak yapılabilir. Birinci konuşma sinyalini $a = \{a_1, a_2, a_3, \dots, a_l\}$ ve ikinci konuşma sinyalini $r = \{r_1, r_2, r_3, \dots, r_j\}$ vektörleriyle ifade edelim. Bu iki sinyal süreleri

eşit olamayacağı için I ve J birbirinden farklı sayıları göstermektedir. Temel bir normalizasyon tekniği kısa süreli konuşma sinyalinin süresini doğrusal olarak uzun olana genişletmektir. $J < I$ olduğunu kabul edelim, a ve r sinyalleri arasındaki uzaklığı $D(r,a)$ ile gösterirsek,

$$D(r,a) = \sum d_f(r_i, a_i) \quad (2.21)$$

$d_f(r_i, a_i)$ terimi r_i ve a_i çerçeveleri arasındaki uzaklığı göstermektedir.



Şekil 2.13. “Speech” kelimesinin değişik zaman normalizasyonlarıyla karşılaştırılması

Zaman bölgesinde aynı konuşmacının telaffuz ettiği “speech” konuşma sinyal süreleri uzunluklarının karşılaştırılması şekil 2.13.c’de gösterilmektedir. Lineer normalizasyon yaparak aynı hizaya getirme işleminin sonucu şekil 2.13.d’de görülmektedir. Bu bize konuşma hızının telaffuz edilen sözcük süresiyle orantılı olduğunu ve telaffuz edilen sözcükten bağımsız olduğunu gösterir. Fakat bu konuşma sinyalinin gerçek durumunu

modellemeye yetmez. Sinyaller arasında kabul edilebilir bir eşleme sağlamak amacıyla kullanılan normalizasyon tekniğinin, $D(r,a)$ ' yi minimize edilebilmek için sinyallerden birinin süresini diğer sinyal karakterlerine göre değiştirmesi gerekmektedir. Şekil 2.13.e doğrusal olmayan normalizasyon örneğini göstermektedir. Burada birinci konuşma sinyalinde s, p, ch fonemlerinin süresinin uzatıldığını ve iy fonem süresinin kısaltıldığını görülmektedir.

2.7.2.1. Dinamik programlama

Dinamik programlama ardışık karar problemlerinin çözümünde kullanılan yöntemlerden biridir. Doğrusal olmayan eşleme dik açılarla birleşen ve kareler oluşturan düzlem üzerinde gerçekleştirilir(Zeng 2000). Uyumlandırılacak sinyaller a ve r sinyalleridir. a sinyali x eksenini boyunca r sinyali de y eksenini boyunca sıralanır. Bu düzlem üzerindeki her bir kesişim noktası bir düğüm olarak adlandırılır ve (i, j) düğüm noktası bize a sinyalinin i. çerçevesinin, r sinyalinin j. çerçevesiyle eşlendiğini gösterir. $(0,0)$ düğümü tüm yolların başladığı başlangıç düğümü olarak kabul edilir. Uyumlandırmanın sonucu, a_i ve r_j öznitelik vektörleri arasındaki $d_f(i,j)$ uzaklık ölçütüyle tanımlanır. Başlangıç düğümü için;

$$d_f(0,0) = 0 \text{ olarak tanımlanır.} \quad (2.22)$$

Yol tanımı $(i_{k-1}, j_{k-1}) \rightarrow (i_k, j_k)$ düğüm çiftlerinin zincir gibi birbirine bağlanmasıyla elde edilir. Bu bize (i_{k-1}, j_{k-1}) düğümünün (i_k, j_k) düğümüne uzatıldığını gösterir. i_k , k anındaki a sinyalini, j_k ise k anındaki r sinyalini göstermektedir. $(0,0)$ düğümünden başlayıp (i_k, j_k) düğümünde biten yol için toplam uzaklık hesabı 2.23 eşitliği ile yapılmaktadır:

$$D(i_k, j_k) = D(i_{k-1}, j_{k-1}) + d_f(i_k, j_k) \quad (2.23)$$

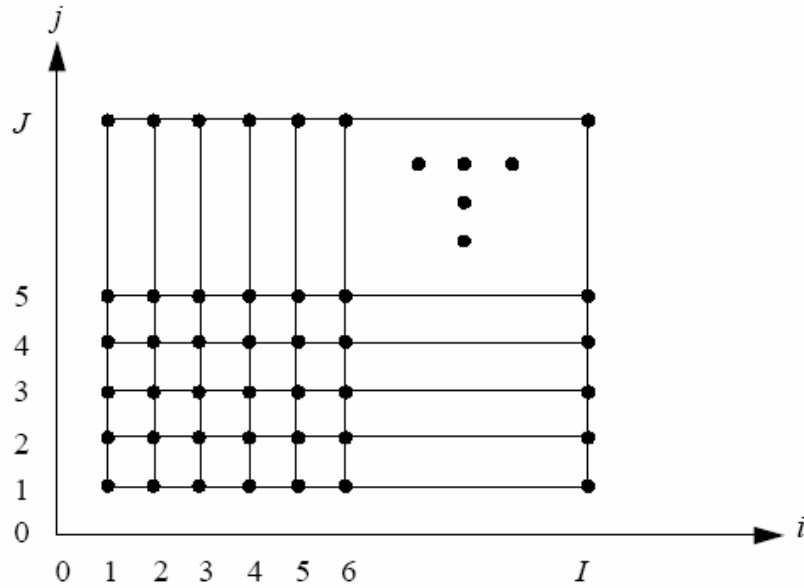
Başlangıç düğümü için $D(0,0) = 0$ olduğundan 2.24 eşitliğine geçiş yapılabilir.

$$D(i_k, j_k) = \sum_{m=0}^k d_f(i_m, j_m) \quad (2.24)$$

Problem (I,J) düğümünde bitecek toplam uzaklığı minimum yapacak (i_k, j_k) düğüm çiftlerinin bulunmasına indirgenebilir.

$$\begin{aligned} D(i_k, j_k) &= \min[D(i_{k-1}, j_{k-1})] + d_f(i_k, j_k) \\ &= \min \left[\sum_{m=0}^k d_f(i_m, j_m) \right] \end{aligned} \quad (2.25)$$

Konuşma ve konuşmacı tanıma uygulamalarında örüntü karşılaştırma metodu test sinyali ile tüm şablonlar arasındaki mesafe ölçümü ve en küçük uzaklığı veren şablonu bulma anlamına gelmektedir. Tanımlama işlemi şablonla sinyal arasındaki uzaklık ölçümü olarak basite indirgenebilir. Bu problemin zorluğu insanların konuşmalarındaki farklılıktan kaynaklanmaktadır. Konuşma aksanları, konuşma hızı gibi insandan kaynaklanan konuşmacı tanımlama uygulamalarında bu zorluğun üstesinden gelmek için test sinyali ile şablonlar arasındaki uzaklığı minimum yapmak amacıyla doğrusal olmayan dinamik programlama kullanılır. Bu dinamik programlama uygulaması dinamik zaman eğrisi olarak adlandırılır.



Şekil 2.14. Dinamik programlamanın gösterildiği noktali düzlem

2.8. Konuşmacı Tanımda Kullanılan Parametreler

Konuşmacı tanımda kullanılan parametrelere geçmeden önce konuşma sinyalinin neden parametrelendirildiğine bakalım:

- i) Konuşma sinyalinin anlamlı ve sağlam bir gösterimini yapabilmek için,
- ii) Konuşma sinyalindeki gereğinden fazla bilgiden kurtulmak için,
- iii) Tanıma performansını iyileştirmek için,
- iv) Hafıza gereksinimini ve işlemsel karışıklığı azaltmak için.

2.8.1. Parametrelendirmenin yapılması

Konuşmacı tanımda en çok kullanılan parametreler:

1. Doğrusal Öngörüm Katsayıları (LPC);
2. Cepstrum Katsayıları
 - Standart Cepstrum Katsayıları
 - LPC'den elde edilen Cepstrum Katsayıları
 - FFT temelli Mel-ölçeklendirilmiş Cepstrum Katsayıları
 - Delta(Fark) Cepstrum Katsayıları

2.8.1.1. Doğrusal öngörüm katsayıları

Doğrusal öngörüm modelinin arkasında yatan temel düşünce, n zamanda verilen s(n) konuşma örneklerinin önceki p tane konuşma örneğinden yaklaşık olarak 2.26 eşitliğindeki gibi elde edilmesidir (Deller et al. 2000).

$$s(n) = a_1s(n-1)+a_2s(n-2)+\dots+\dots+a_p s(n-p) \quad (2.26)$$

$$s(n) = \sum_{i=1}^p a_i s(n-i) \quad (2.27)$$

Konuşma analizi çerçevesindeki a_1, a_2, \dots, a_p katsayıları sabit varsayılmaktadır. $G.u(n)$ uyarıcı terimini kullanarak 2.28 eşitliğine dönüştürebiliriz:

$$s(n) = \sum_{i=1}^p a_i s(n-i) + G.u(n) \quad (2.28)$$

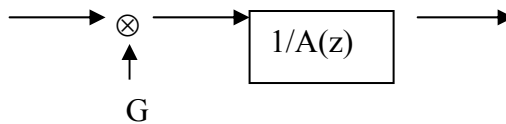
Burada $u(n)$ normalize edilmiş uyarıcı fonksiyonu G ise uyarıcı kazançtır. 2.28 eşitliğini z bölgesinde gösterirsek 2.29 eşitliğindeki ilişki elde edilir.

$$S(z) = \sum_{i=1}^p a_i z^{-i} S(z) + G.U(z) \quad (2.29)$$

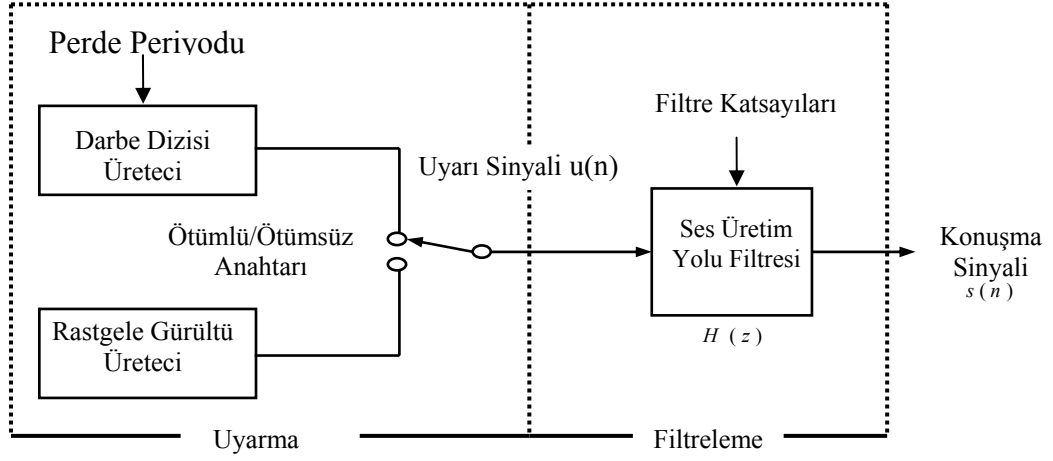
Buradan transfer fonksiyonunu çekersek,

$$H(z) = \frac{S(z)}{G.U(z)} = \frac{1}{1 - \sum_{i=1}^p a_i z^{-i}} = \frac{1}{A(z)} \text{ elde edilir.} \quad (2.30)$$

$s(n)$ konuşma sinyali üretmek için, $H(z) = \frac{1}{A(z)}$ transfer fonksiyonlu all-pole sistemine giriş olarak uygulanan ve G kazanç katsayısıyla çarpılmış, normalize edilmiş uyarıcı kaynak $u(n)$ ' nin gösterimi şekil 2.15 ile yapılabilir.



Şekil2.15. Konuşmanın doğrusal öngörüm modeli



Şekil 2.16. Konuşma sinyali üretim modeli

Konuşmacı için uyarıcı fonksiyon temelde ya periyodikmiş gibi görünen vuruş dizisi (sesli konuşma sesleri için) ya da geliş güzel olan gürültü kaynağıdır (sessiz konuşma sesleri için).

Konuşma sesleri için LPC analizine uygun yöntem şekil 2.16’de gösterilmektedir.

Burada normalize edilmiş uyarıcı kaynak $u(n)$, pozisyonu konuşmanın sesli/sessiz karakterleriyle kontrol edilen ve bu vasıta ile ya sesli sesler için uyarıcı gibi davranan sanki periyodikmiş gibi olan vuruş dizilerini ya da sessiz sesler için uyarıcı gibi davranan geliş güzel gürültü kaynağını seçen anahtarla kontrol edilir. Kaynağın G uygun kazanç konuşma sinyalinden tahmin edilir ve G katsayısıyla çarpılmış uyarıcı kaynak uygulanan konuşmanın karakteristik ses yolu parametreleriyle kontrol edilen $H(z)$ sayısal filtrenin girişine uygulanır. Böylece bu modelin parametreleri sesli/sessiz sınıflandırma, sesli sesler için perde periyodu, kazanç parametresi ve sayısal filtre katsayılarıdır (a_k). Bütün bu parametreler zamanla yavaş yavaş değişmektedirler (Kondo 1990).

2.8.1.2. Cepstral katsayılar

Sesler arasındaki farkların ve benzerliklerin nedeni insan gırtlak yapısından kaynaklanmaktadır. Cepstral katsayılar sesler arasındaki fark ve benzerlikleri ortaya koymada kullanılan yöntemlerden biridir.

Konuşma sinyali, sentez filtresinin girişine uyarı sinyalinin uygulanmasıyla elde edilir. Başka bir deyişle konuşma sinyali ses üretim yolu filtresi ve uyarı sinyalinin katlanması sonucu elde edilir. Uyarı sinyalinin dürtü tepkisini $u(n)$, ses üretim yolu filtresi dürtü tepkisi $h(n)$ ile gösterirsek, bunların katlanmasıyla elde edilecek ses sinyali $s(n)$ 'de

$$s(n) = u(n) * h(n) \text{ olur.} \quad (2.31)$$

2.31. eşitliğini frekans bölgesinde ifade edersek 2.32 eşitliğini elde ederiz.

$$S(w) = U(w) \cdot H(w) \quad (2.32)$$

Konuşma sinyalinden ses üretim yolu filtresi ve uyarı sinyalinin dürtü tepkilerini elde edebilmek için doğrusal filtreler kullanılmak istendiğinde bu çarpımsal ifadeyi her iki tarafın logaritmasını alarak toplamsal hale dönüştürebiliriz.

$$\text{Log}(S(w)) = \text{Log}(U(w) \cdot H(w)) \quad (2.33)$$

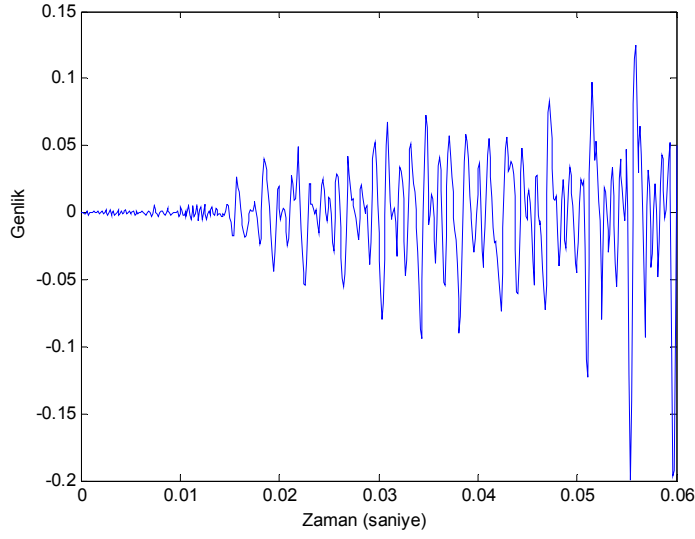
$$\text{Log}(S(w)) = \text{Log}(U(w)) + \text{Log}(H(w)) \quad (2.34)$$

Çarpımsal ifade toplam hale dönüştürüldükten sonra bu sinyallerin ters fourier dönüşümü alınarak quefrensy olarak adlandırılan bölgeye geçiş yapılmış olur.

$$\text{IFFT}(\text{Log}(S(w))) = \text{IFFT}(\text{Log}(U(w))) + \text{IFFT}(\text{Log}(H(w))) \quad (2.35)$$

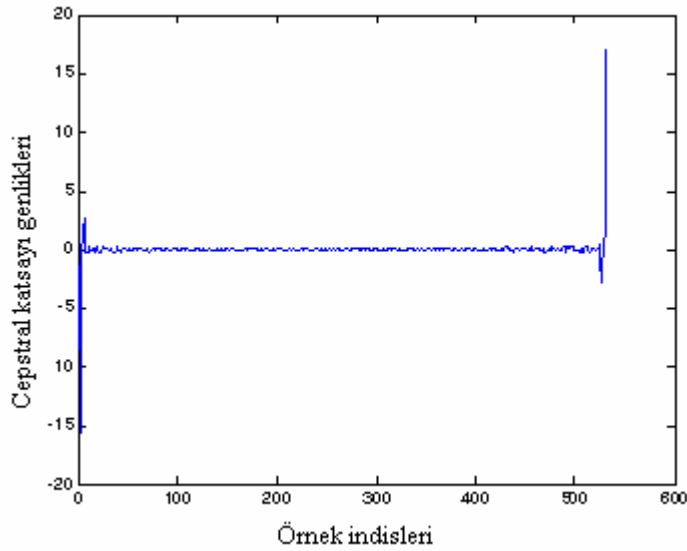
Doğrusal filtreleme yöntemleri kullanılarak bu bölgede ses üretim yolu filtre dürtü tepkisine veya uyarı sinyali dürtü tepkisine geçiş yapılabilir(Hanzo 2001).

Şekil 2.17’de “a” seslisinin zaman bölgesindeki gösterimi görülmektedir.



Şekil 2.17. a seslisinin dalga şekli

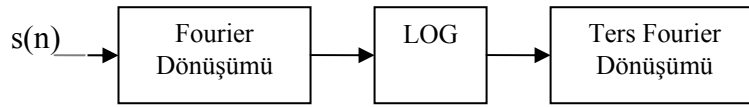
Şekil 2.17’de gösterilen konuşma sinyalinden elde edilen cepstral katsayılara ait grafik şekil 2.18’de görülmektedir.



Şekil 2.18. a seslinin cepstral gösterimi

$$s(n) = u(n) * h(n) = \text{IFFT}(\text{Log}(\text{U}(w))) + \text{IFFT}(\text{Log}(\text{H}(w))) \quad (2.36)$$

2.36 eşitliğindeki katlama operatörünün toplamsal hale dönüştüğünü görüyoruz. Cepstrum homomorf bir dönüşümdür. Konuşma sisteminde ses üretim yolu filtresi ve uyarı sinyali zamanla değiştiğinden dolayı cepstral analiz konuşma sinyalinin durağan olduğu kısa zamanlı aralıklarda yapılır. Bu işleme ait blok diyagram şekil 2.19'da verilmiştir.



Şekil 2.19. Cepstral katysayıların elde edilişi

Cepstral katsayıları elde etmek için önce pencerelenmiş ses örneğinin FFT(Fast Fourier Transform)'si alınır. FFT' si alınan ses örneğinin frekans bileşen genliklerinin logaritması alınır. En sonunda elde edilen değerin ters FFT'si alınır. Böylece pencerelenen ses sinyalini cepstral analizi yapılmış olur.

Cepstrum, sinyal bileşenlerinden gerekli bilgiyi elde etmemizi sağlar. Homomorf süreçle elde edilen cepstrum kompleks cepstrum olarak adlandırılır ve kompleks cepstrum içindeki ölçeğin reel kısmı reel cepstrum olarak adlandırılır. Belirli bir zaman zarfı sonunda konuşma araştırmalarında cepstrum öznelikleri, tanıma hızını artırdığını gösteren iki özellikten dolayı doğrusal öngörüm katsayılarının yerini almaya başlamıştır. Bunlardan birincisi “liftering” ve diğeri de ağırlıklandırma süreçlerini kullanarak doğrusal öngörüm temelli spektrumu düzleştirmesidir. “liftering”, cepstrumdaki ayrık bileşenlerin her birinin logaritmik spektrumunun tahmini elde etmek için kullanılan yararlı bir reel cepstrum sürecidir. Ağırlıklandırma, cepstral katsayılar arasındaki euclid uzaklığında toplamdaki her terimin önceden belirlenmiş ağırlıklandırma katsayıları ile çarpılması işlemidir. Sabit ağırlıklandırma kullanılırsa bu cepstral uzaklığa indirgenmiş olur. Konuşmacıya bağlı ve konuşmacıdan bağımsız tanıma deneyleri bize ağırlıklandırılmış cepstral uzaklık ölçümlerinde tanıma

performansının, cepstral fark sayısının yaklaşık olarak all-pole model derecesine eşit olduğu zaman en iyi olduğunu göstermiştir (Wong et al. 2001)

2.8.1.3. Mel-ölçeklendirilmiş cepstrum katsayıları

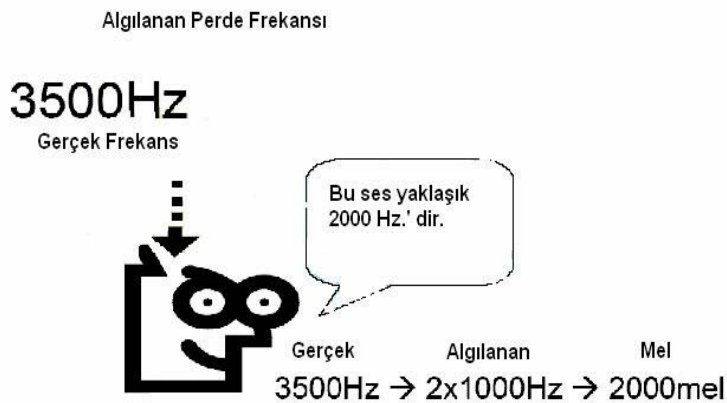
2.8.1.3.1. Mel-ölçeklendirme

Mel ölçeklendirme, insan kulağının duyma özelliğine göre oluşturulmuş algısal bir ölçeklendirmedir. İnsan kulağı frekansları lineer olarak algılayamaz. Mel algılanan ses sinyal frekansı ölçüm birimidir. Stevens ve Valkman adlı bilimadamları tarafından 1937 yılında bulunmuştur.

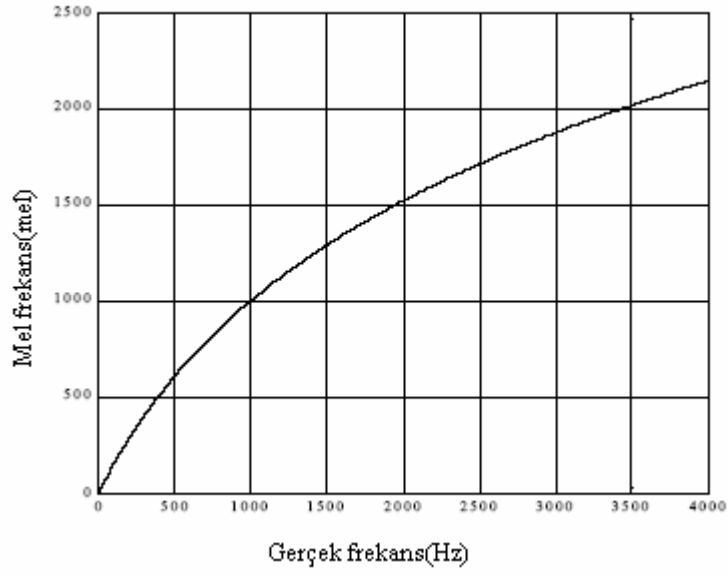
Referans olarak 1 kHz seçilmiş ve 1000 mel' e karşılık geldiği kabul edilmiştir. Dinleyicilere algıladıkları frekansı referansın 2 katı, 10 katı, 1/2' si, 1/10' u oluncaya kadar değiştirmeleri istenmiştir. Bu frekanslar 2000 mel, 10000 mel, 500 mel ve 100 mel' e karşılık gelmektedir.

Matematiksel formülasyonu (0-4 kHz. arasında):

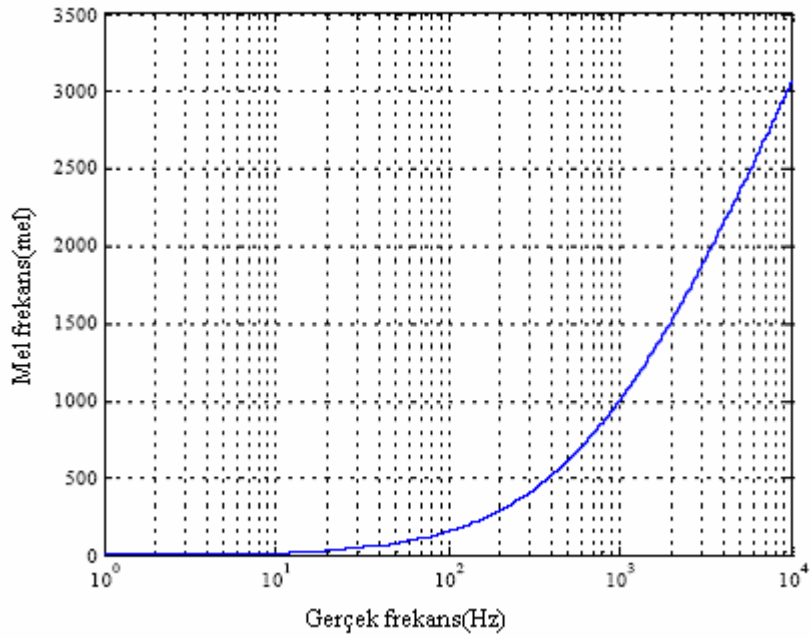
$$M(f)=2595\log_{10}(1+f/700) \quad (2.37)$$



Şekil 2.20. Mel-ölçeklendirme örneği



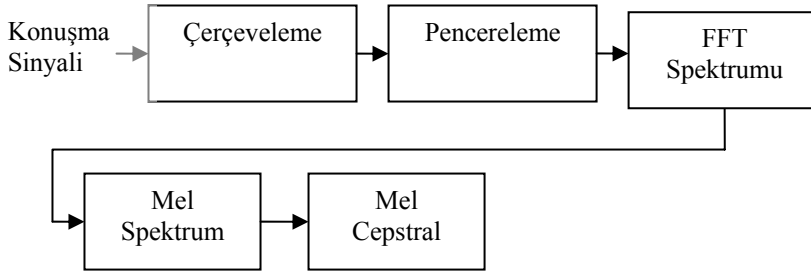
Şekil 2.21. 0-4 kHz arasında mel-ölçeklendirme grafiği



Şekil 2.22. 0-10 kHz. arasında mel-ölçeklendirme grafiği

2.8.1.3.2. Mel-ölçeklendirilmiş cepstral katsayılar

Konuşma sinyali çerçevelere ayrılıp pencerelenir. Güç spektrumu elde edilir. Mel spektrumuna geçilir. DCT (Discrete Cosine Transform, Kesikli Kosinüs Dönüşümü) alınarak cepstral katsayılar elde edilir.



Şekil 2.23. Mel-ölçeklendirilmiş cepstral katsayıların elde edilişi

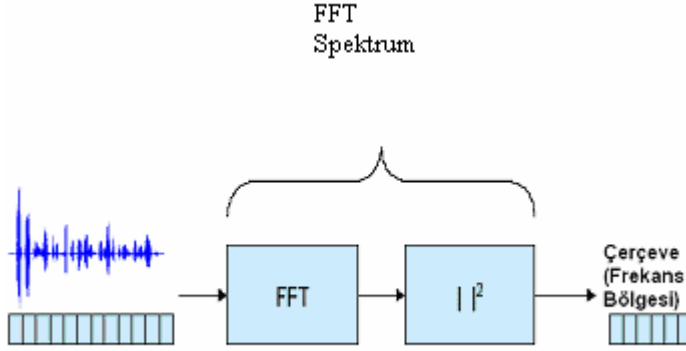
2.8.1.3.3. Çerçeveleme

Sürekli konuşma sinyali N konuşma örneği içeren çerçevelere ayrılır ve her komşu çerçeve $M < N$ şartını sağlayacak şekilde ayrılır. Birinci çerçeve N konuşma örneği içerir. İkinci çerçeve birinci çerçeveden M örnek sonra başlatılır. Birinci çerçeve ile ikinci çerçevenin $N-M$ örneği örtüşür. Benzer olarak üçüncü çerçeve, birinci çerçeveden $2M$ örnek sonra başlatılır ve $N-2M$ örnek örtüşür.

2.8.1.3.4. Pencereleme

Çerçevelemeden sonraki adım, her bir çerçeve sonundaki ve başlangıcındaki süreksizliği önlemek için yapılan pencereleme işlemidir. Bu adımda spektral bozulmayı azaltmak için sinyali, her bir frame başlangıcında ve sonunda sifra çekmek için bir pencereyle çarpılır. Kullanılabilecek pencereler Hamming, Hanning, Blackman, Dikdörtgen pencerelerdir.

2.8.1.3.5. FFT spektrum



Şekil 2.24. FFT spektrum

FFT, her çerçeveyi zaman bölgesinden frekans bölgesine taşır. FFT, kesikli fourier dönüşümün hızlı şekilde yerine getiren hızlı bir algoritmadır. $\{x(n)\}$ 'nin N örnek kümesinin FFT'si 2.38 eşitliğindeki gibi alınır:

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn / N}, n = 0,1,2,\dots,N-1 \quad (2.38)$$

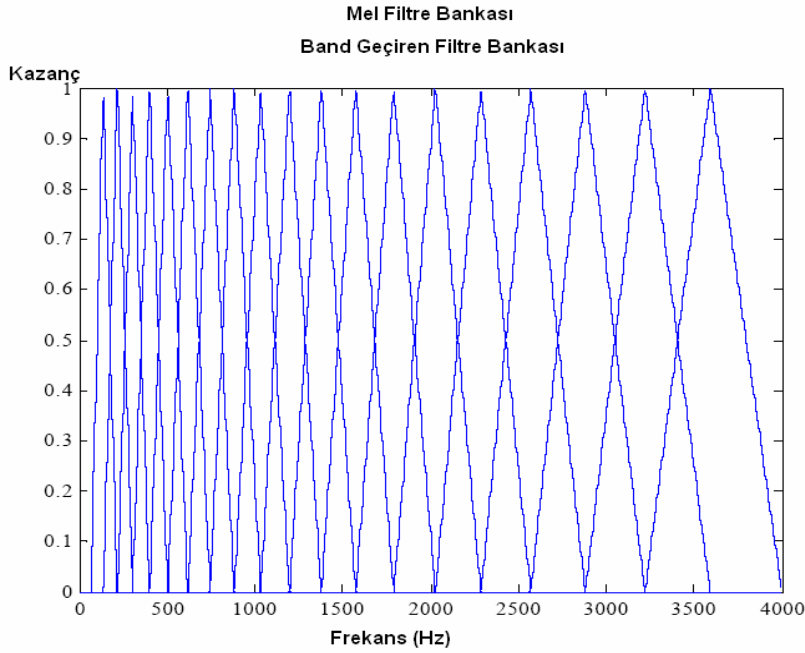
2.8.1.3.6. Mel-spektrum

Mel Spektrumunu elde etmek için FFT Spektrum çıkış sinyali Mel-Filtre Bankası adı verilen bir dizi filtreden geçirilir. Bu filtre bankasını üçgen dalgalar oluşturur. Mel filtre bankası perde harmoniklerini ve konuşma sinyalindeki gürültüyü yumuşatır ve insan kulağı tarafından duyulan perde frekansını kuvvetlendirir(Molau et al. 2001).

2.8.1.3.7. Mel-filtre bankası tasarımı

Mel-Filtre Bankasını band-geçiren üçgen dalgalar oluşturur. Elde edilmeye çalışılacak katsayı sayısı kadar üçgen filtre kullanılır. Band genişlikleri ve filtreler

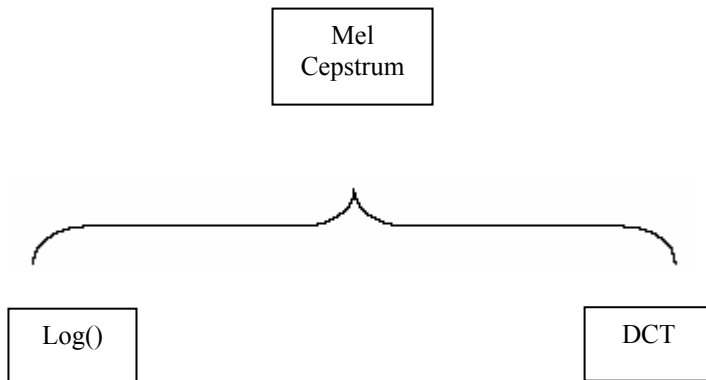
arasındaki boşluklar mel ölçeklendirmiş frekans bölgesinde sabittir. Gerçek frekans bölgesinde 1 kHz'in altında lineer ve 1 kHz'in üstünde logaritmiktir.



Şekil 2.25. Mel-filtre bankası

%50 oranında birbiri üstüne getirilen üçgen band geçiren dalgalar kullanılır. 10 Hz.'den 1000 Hz.'e lineer olarak 10 filtre atanır. Frekans her iki katına çıktığında logaritmik olarak 5 filtre bu frekans aralığına atanır. Filtre band genişliği 3 dB noktası, mevcut dalga ile bir önceki veya bir sonraki dalganın ortasına gelecek şekilde seçilir. Genellikle 20 filtre kullanılır.

2.8.1.3.8. Mel-cepstrum



Şekil 2.26. Mel cepstrum

Mel-filtre bankasından geçirilen sinyalin logaritması alındıktan sonra kesikli kosinüs dönüşümü ile tekrar zaman bölgesine geçilir. Şekil 2.26'da gösterildiği gibi elde edilen katsayılara mel frekans cepstrum katsayıları denir. Konuşma spektrumunun cepstral gösterimi verilen sinyal çerçevesi için yerel spektral özelliklerinin iyi bir gösteriminin yapılmasını sağlar. Çünkü mel spectrum katsayıları reel sayılardır, DCT kullanılarak zaman bölgesi dönüşümleri yapılabilir.

DCT dönüşüm denklemi:

$$c(i) = \sqrt{\frac{2}{L}} \sum_{m=1}^L \log(S(m)) \cos\left[\frac{\pi i}{L}(m-0.5)\right], i = 0, 1, \dots, C-1 \quad (2.39)$$

2.39 eşitliğinde kullanılan $\tilde{S}(m)$, mel spektrum çıkış sinyalini ve L' de mel filtre bankasında kullanılan band geçiren üçgen filtre sayısını gösterir.

2.8.1.4. Doğrusal öngörüm katsayılarından türetilmiş cepstral katsayılar

Konuşmadan spektral bilginin çıkarılması için kullanılan analiz yöntemlerinin içinde doğrusal öngörüm gelir. Doğrusal öngörüm analizi ses üretim yolu karakteristiklerini çözmez. Gırtlaksal karakteristikler insandan insana değiştiğinden ve hatta aynı konuşmacının aynı telaffuzu tekrar edememesinden, doğrusal öngörüm parametreleri özellikle konuşmacıdan bağımsız sistemlerde konuşmacı tanıma performansını azaltan bazı bilgileri taşır. Doğrusal öngörüm modeli, cepstral katsayıları hesaplamak için çok yararlı bir araçtır. Bölüm 2.8.1.1.'de anlatılan LP filtre katsayıları $a_i, 1 \leq i \leq p$ kullanılarak 2.40 eşitliğindeki gibi iteratif bir şekilde hesaplanabilir(Tokuda et al. 1994).

$$c_m = \begin{cases} a_m + \sum_{k=1}^{m-1} \frac{k}{m} c_k a_{m-k}, & 1 < m < p \\ \sum_{k=m}^{m-1} \frac{k}{m} c_k a_{m-k}, & m > p \end{cases} \quad (2.40)$$

Bu katsayılar doğrusal öngörümünden türetilmiş cepstral katsayılar olarak adlandırılır. Konuşma sinyalinin doğrusal öngörüm analizi gerçekleştirildikten sonra, doğrusal öngörüm parametrelerinin eşdeğerleri ortaya çıkmıştır. Konuşmacı tanıma uygulamalarının doğruluğunda parametrelerin karşılaştırması yapıldığında eğitim ve test aşamasında doğrusal öngörüm cepstrumun iyi sonuç verdiği ortaya konulmuştur. Doğrusal öngörüm cepstrumla ilgili temel problem eğitim ve test koşullarındaki uyumsuzluktan dolayı performansın kötüleşmesine yol açmasıdır; bu yüzden sağlamlığı azaltmaktadır. Doğrusal öngörüm spektrumu konuşmanın spektral zarfını tanımlayan all-pole transfer fonksiyonundan türetilir. Bu özellikle konuşmacı tanımanın başarılı olabilmesi için kritik olan formantlar hakkında bilgi verir. Gerekli olan ilk aşama doğrusal öngörüm analizinden türetilen all-pole transfer fonksiyonunu formantlara daha fazla önem veren pole-zero transfer fonksiyonuna dönüştürmektir. Pole-zero transfer fonksiyonunun cepstrumu özneliktir.

2.8.1.5. Delta cepstral katsayılar

Konuşma veya konuşmacı tanıma sistemlerinin performansı temel durağan parametrelere zaman türevlerinin eklenmesiyle başarılı bir şekilde iyileştirilebilir. Delta öznelikleri, regresyon formülüzasyonu kullanılarak 2.41 eşitliğinde olduğu gibi elde edilebilir (Wildermoth et al. 2001).

$$d_t = \frac{\sum_{w=1}^W w(c_{t+w} - c_{t-w})}{2 \sum_{w=1}^W w^2} \quad (2.41)$$

d_t , t anında güncel öznitelik veya katsayıdan bir önceki ve bir sonraki durağan parametrelere göre hesap edilir. W , mevcut bütün datayı yapılandırmak için kullanılan pencere boyutudur. Regresyon formülü geçmiş ve gelecekteki konuşma parametrelerine bağlı olduğu için konuşma dosyasının başlangıç ve bitişindeki sinyali kullanabilmek için bazı değişikliklere ihtiyaç vardır. Bu da 2.42 eşitliğinde gösterildiği gibi temel birinci derece farkların kullanılmasıyla giderilebilir.

$$\begin{aligned} d_t &= c_{t+1} - c_t, t < W \\ d_t &= c_t - c_{t-1}, t \geq W \end{aligned} \quad (2.42)$$

t , veri dosyasının toplam uzunluğunu gösterir.

Cepstral veya mel-cepstral parametrelere ek olarak güncel konuşma ve konuşmacı tanıma uygulamalarında kullanılan bir diğer parametre de delta cepstrumdur. m süresi sonundaki $s(n)$ konuşma sinyali çerçevesinin mel-cepstral öznitelik değerlerini $c(n,m)$ 'nin gösterdiğini varsayarsak, m çerçevesi için delta cepstral katsayılar 2.43 eşitliği ile hesaplanır:

$$\Delta c(n,m) = c(n,m+\delta Q) - c(n,m-\delta Q) \quad (2.43)$$

Burada Q , her bir çerçeve için pencerenin kaydırıldığı örnek sayısını gösterir. δ parametresi kestirimi düzgünleştirmek için seçilir ve genellikle bir veya iki çerçeve ileri ve geri bakacak şekilde değer alır. n 'nin göreceli olarak düşük değerlerindeki özniteliklerini içeren vektör önceki çerçeveden beri oluşan spektral değişimler hakkında bilgi edinmemizi sağlar. Delta cepstrum aynı zamanda doğrusal öngörüm temelli cepstral parametreler için de hesaplanabilir. Herhangi bir örüntü tanıma uygulaması için 8-14 cepstral katsayı türevleriyle birlikte cepstral tekniklerin uygulandığı sistemlerde kullanılır. Sinyal işleme alanlarında algoritmaların verimi dikkate alındığında bilgisayar gücü çok önemli bir yer teşkil etmektedir. $s(n)$ sinyali için birinci dereceden türev almak amacıyla en çok kullanılan yaklaşım:

$$s^*(n) = \frac{\partial}{\partial t} s(n) = s(n) - s(n-1) \quad (2.44)$$

$$s^*(n) = \frac{\partial}{\partial t} s(n) = s(n+1) - s(n) \quad (2.45)$$

$$s^*(n) = \frac{\partial}{\partial t} s(n) = \sum_{m=-N_d}^{N_d} m s(n+m) \quad (2.46)$$

2.44 ve 2.45 eşitlikleri sırasıyla geri ve ileri farklar olarak bilinir. Bu differansiyel yöntemi oluşturan sinyal çıkışı delta parametre olarak adlandırılır. İkinci derece zaman türevi, birinci derece türev alıcını çıkışına 2.46. eşitliğinin tekrar uygulanmasıyla benzer biçimde yaklaşılabılır. İkinci derece türev alıcıdan elde edilen çıkış delta-delta parametredir.

2.9. Alıcı İşletim Karakteristik Eğrisi(Receiver Operating Characteristic Curve)

Bir çift duyarlılık ve seçicilik değeri kullanmanın getirdiği dezavantajları ortadan kaldıracak bir yöntem olarak 1971 yılında Lusted tarafından geliştirilmiştir. Testlerin kendi doğruluklarını tanımlaması ve testler arasında en doğru karşılaştırma yapmaya olanak sağlaması açısından sıklıkla kullanılmaktadır. ROC(Receiver Operating Characteristic) eğrisi, değişik kesim noktalarında testin duyarlılığının (y-ekseni), testin YP(yanlış pozitif) oranına (x-ekseni) karşı noktalanması ile elde edilir. Her kesim noktasındaki DP(doğru pozitif) ve YP'e karşılık gelen noktalar birleştirilerek ROC eğrisi çizilir.

1970'li yıllardan sonra ROC eğrileri yaygın olarak tıpta kullanılmaya başlanmıştır. ROC eğrileri elektronik sinyal algılama teorisi içerisinde geliştirilmiş, ancak, tıbbi, askeri ve diğer bir çok alanda kullanım yeri bulmuştur.

ROC eğrileri yorumlayıcı performansını incelemek için iyi bir temel oluşturur. Bir durumun varlığı yada yokluğuna karar vermek için yorumlayıcının kullandığı kriter ile durumu belirlemede yorumlayıcının yeteneği (test sonucunu yorumlamada kullanılan duyuşsal ve/veya bilişsel özellikler) arasında ayırımın yapılması gerekir. ROC eğrisi bu ayırımı yapmak için kullanılabilir (Lusted, 1971).

ROC eğrisinin parametrelerinin kestiriminde Gauss modellerinin kullanımı Green ve Swets tarafından önerilmiş, Dorfman ve Alf iki değişkenli normal ROC eğrisinin parametrelerinin "en çok olabilirlik kestirimleri"ni bulan RSCORE adlı bir yazılım geliştirmişlerdir(Grey et al. 1972).

Metz, Swets ve Picket test doğruluğunun bir ölçüsü olarak ROC eğrisi altında kalan alanı tanımlayarak bu amaca yönelik ROCFIT, LABROC, CORROC ve CLABROC gibi yazılımları hazırlamışlardır(Swets et al. 1995).

ROC eğrileri olası tüm kesim noktalarını gösterirler ve her kesim noktasında değişik sonuçların (DP, DN, YP ve YN) sıklığı hakkında kestirimler yapılmasına olanak sağlarlar.

Verilen bir test için en iyi kesim noktasının saptanmasına yönelik, doğru ve yanlış kararların yarar ve maliyetlerinin hesaplanmasında kullanılırlar.

ROC eğrilerinin konuşmacı tanıma uygulamalarında kullanılan ölçüleri:

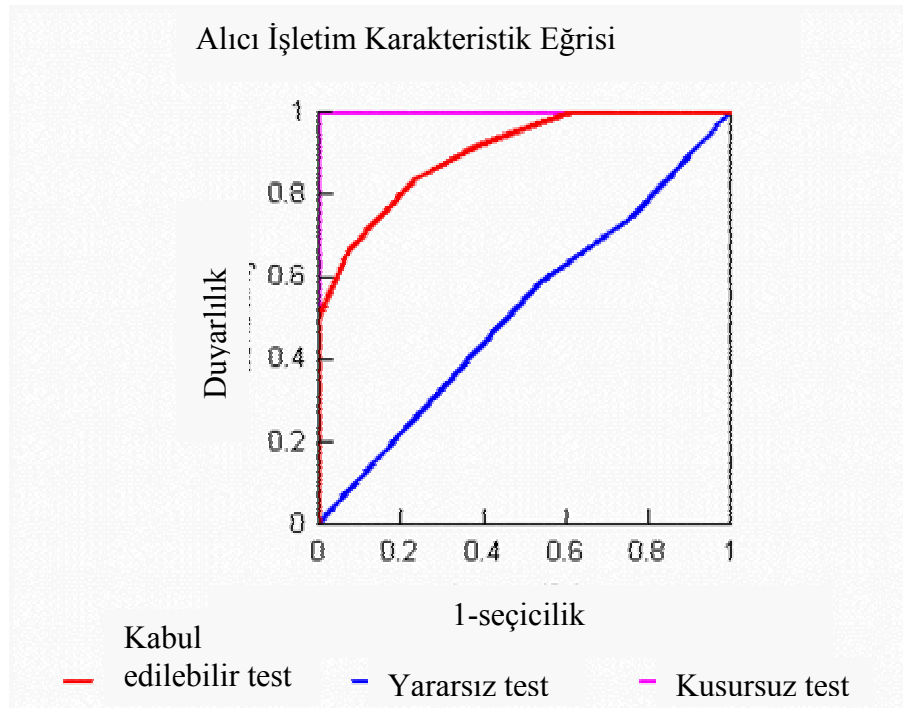
Duyarlılık (Sensitivity): Sistemde gerçekte tanınan konuşmacılar arasında tanıma işleminin pozitif sonuç verme oranı(Doğru pozitif oran).

Seçicilik (Specificity): Sistemde gerçekte tanınmayan konuşmacılar arasında tanıma işleminin negatif sonuç verme oranı(Doğru negatif oran)

Yanlış pozitif oran: Sistemin gerçekte tanıyamadığı konuşmacılar arasında tanıma işleminin yanlışlıkla pozitif sonuç verme oranı(1-Seçicilik).

Yanlış negatif oran: Gerçekte tanınan konuşmacılar arasında tanıma işleminin yanlışlıkla negatif sonuç verme oranı(1-Duyarlılık).

Bir sınıflandırıcının alıcı işletim karakteristik eğrileri seçicilikle duyarlılık arasındaki ödünleşim performansını gösterir. Şekil 2.27'de de gösterildiği gibi eğri her zaman iki noktadan geçmektedir; (0,0) ve (1,1). (0,0) noktasında sınıflandırıcı hiçbir pozitif durum bulamaz. (0,0) konumunda tüm pozitif durumlar yanlış, tüm negatif durumlar da doğru alınmıştır. (1,1) noktası ise tüm durumların pozitif sınıflandırıldığı noktadır. Bu noktada sınıflandırıcı tüm pozitif durumları doğru ve tüm negatif durumları da yanlış alır. Rastgele olarak tahmin yapan bir sınıflandırıcının alıcı işletim karakteristik eğrisi köşegen olacaktır.

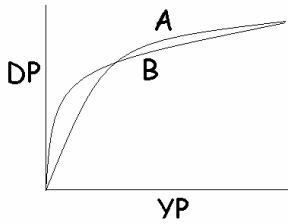


Şekil 2.27. Değişik ROC eğri gösterimleri

ROC Eğrisi altında kalan alan, testin doğruluğunu tek bir sayısal değerle özetlemek için kullanılır. En büyük “1” değerini alabilir. Pratik olarak alabileceği en küçük değer “0.50” dir.

ROC eğrisi altında kalan alan her zaman çok bilgilendirici olmayabilir. Kimi zaman, testin yüksek seçici(yüksek duyarlı) olması istendiğinde düşük seçicilik değerleri(düşük duyarlılık değerleri) ile değil, ROC eğrisinde yüksek seçiciliğe (yüksek duyarlılığa) karşılık gelen bölge ile ilgilenebiliriz.

İki test aynı ROC eğrisi altında kalan alan değerine sahip olabilir, ancak işleyişleri farklıdır.



Düşük yanlış pozitif oran (yüksek seçicilik) gerekli ise, şekil 2.28'deki eğrilerden B testi, A testine tercih edilebilir.

Şekil 2.28. İki ROC eğrisinin karşılaştırılması

ROC eğrilerinin duyarlılığını ve seçiciliğini birleştirerek kullanan diğer performans ölçüleri:

Pozitif Olabilirlik Oranı:

$$POO = \text{Duyarlılık} / (1 - \text{Seçicilik}) = DP / YP \quad (2.47)$$

Testin, her doğru pozitif sonuca karşılık kaç tane yanlış pozitif sonuç verdiğini gösterir.

DP=0,80 ve DN=0,90 olan bir test için;

$$POO = 0.80 / 0.10 = 8$$

Bu test, her 8 doğru pozitif sonuca karşılık 1 yanlış pozitif sonuç verir.

Dokuz pozitif sonucun 8'i doğru, biri yanlıştır

Negatif Olabilirlik Oranı:

$$NOO=(1-Duyarlılık)/Seçicilik=YN/DN \quad (2.48)$$

Testin, her yanlış negatif sonuca karşılık kaç tane doğru negatif sonuç verdiğini gösterir.

DP=0,80 ve DN=0,90 olan bir test için;

$$NOO=0.2/0.9=2/9$$

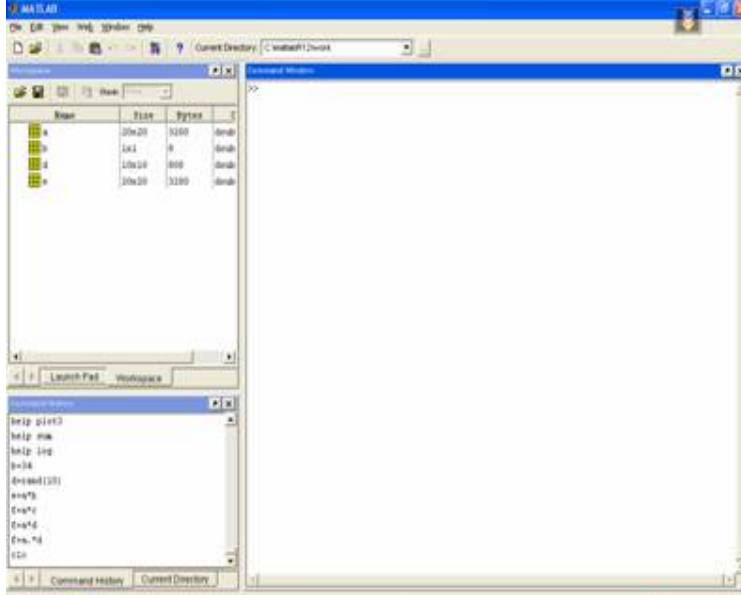
Bu test, her 2 yanlış negatif sonuca karşılık 9 doğru negatif sonuç verir.

Onbir negatif sonucun 9'u doğru, ikisi yanlıştır.

POO'nun olabildiğince büyük, NOO'nun olabildiğince küçük olması istenir.

3. MATERİYAL ve YÖNTEM

Vektör nicemleme tekniklerine dayalı, kapalı set, metinden bağımsız bir konuşmacı tanıma sisteminin tasarlandığı bu çalışma şekil 3.1’de arayüzü gösterilen Matlab ortamında kodlanmıştır. MATLAB, teknik hesaplamalar, matematiksel problemlerin



Şekil 3.1. Matlab arayüzü

elde

acquistion), veritabanı (database), süzgeç tasarımı (filter design), bulanık mantık (fuzzy logic), sistem kimliklendirme (system identification), dalgacıklar (wavelets) gibi araçları ile mükemmel bir ortam sunar.

çözümü ve analizi için tasarlanmış bir yazılım geliştirme aracıdır. MATrix LABoratory kelimesinin kısaltması olan MATLAB, adında da anlaşılacağı üzere matrisler yani diğer bir deyişle diziler ile çalışır. Özellikle mühendislik yapay sinir ağları, sayısal işaret işleme,optimizasyon

(optimization), veri etme (data

Konuşmacı tanıma sisteminde kullanılmak üzere konuşmacı seslerini kaydedecek sağlıklı bir ortam oluşturulamadığı için hazır bir veri tabanı kullanılmıştır. Bu çalışmada, telefon hattı üzerinden okuma metni şeklinde Tübitak Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü(UEKAE) tarafından toplanan TURTEL veritabanından yararlanılmıştır. Oluşturulan veritabanının içeriği toplam 373 kelime ve 15 cümleden meydana gelmektedir. Kelimeler, Türkçe’de geçen üçlü-seslerin dağılımı gözönüne alınarak seçilmiştir. Türkçe’de geçen yaklaşık 11000 üçlü-sesten sadece en sık geçen 1000 tanesinin modellenmesi ile dilin %80’lik kısmının tanınabileceği hesaplanmıştır ve bu veritabanının üçlü-ses veya tek-ses gibi alt birimlerle modellenen Türkçe konuşma tanıma sistemlerinin eğitimi için iyi bir kaynak olduğu öngörülmüştür.

Bu veritabanının içeriğinde deęişik yaşı, eğitim ve coęrafi dağılımdan kadın ve erkek konuşmacıdan toplanan eğitim kayıtları ve test kayıtları bulunmaktadır. Konuşmaların toplam kayıt süresi yaklaşık bir saattir. Telefon hattından gelen ses bilgisi 48000Hz’de DAT’a kaydedildikten sonra DATLink aracılığı ile 8000Hz (Mono) örnekleme frekansı ile bilgisayar ortamına aktarılmıştır. Kayıt çözünürlüğü 16 bit/örnek’dir. Kayıtlar üç farklı tip telefon üzerinden alınmıştır. Bu telefonların dağılımı çizelge 3.1’de verilmiştir.

Çizelge 3.1. Kayıtta kullanılan telefon çeşitleri

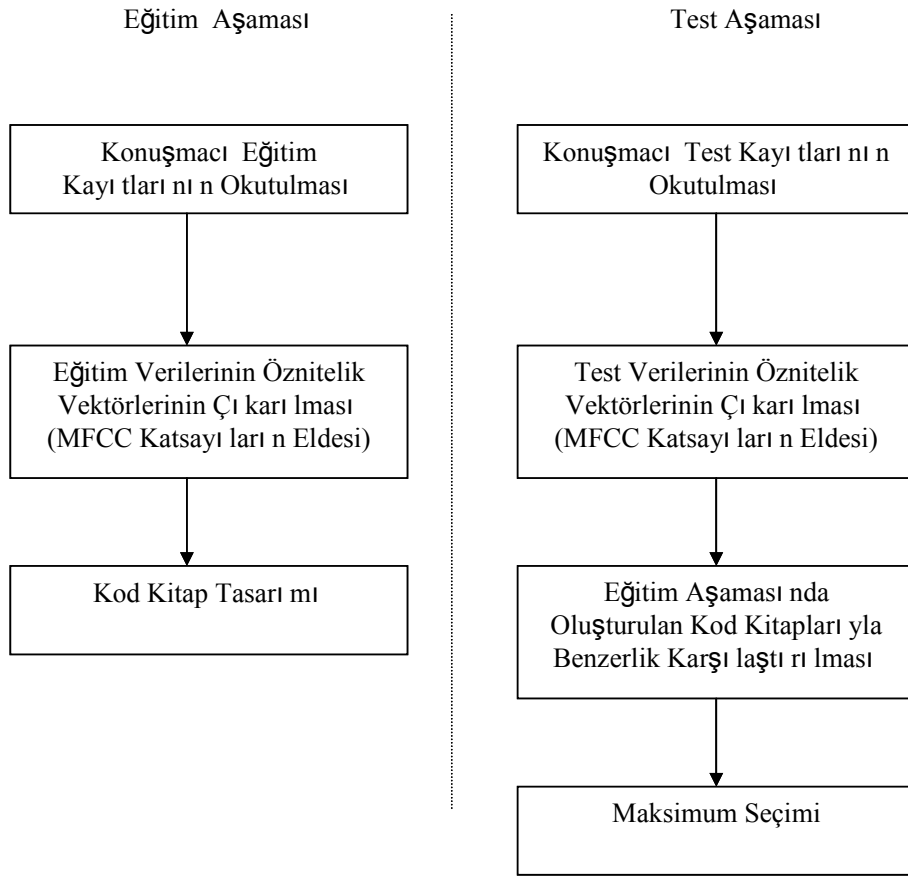
<i>Normal:</i>	%58
<i>Cep:</i>	%19
<i>Ahizesiz:</i>	%23

Eğitim aşamasında veri tabanındaki kelimelerden yararlanılmıştır. Veritabanında yer alan 373 kelime ekte verilmiştir.

Cümleler anahtar kelime yakalama uygulamalarının performansını test etmek amacıyla kayıt edilmiştir. Özellikle ‘ç’, ‘ğ’, ‘ö’, ‘ş’ ve ‘ü’ gibi Türkçe karakterlerin de geçtięi, “yağmur”, “güneş”, “rezervasyon”, “bilet”, “proje”, “fatura”, “çiçek”, “örümcek”, “asker” ve “deprem” kelimelerini içeren bir anahtar kelime seti öngörülmüştür.

4. ARAŞTIRMA BULGULARI

Vektör nicemleme tekniklerine dayalı, kapalı set, metinden bağımsız bir konuşmacı tanıma sisteminin tasarlandığı bu çalışmada eğitim ve test olmak üzere iki süreç bulunmaktadır. Eğitim süreci, konuşmacı eğitim datalarını kullanarak her bir konuşmacının öznelik vektörlerinin çıkarılmasını ve Linde, Buzo ve Gray(LBG) algoritmasıyla her bir konuşmacı için kod kitap tasarımını içerir. Test aşaması ise konuşmacı test kayıtlarının kullanarak öznelik vektörlerinin elde edilmesi ve bulunan öznelik vektörleri ile eğitim aşamasında oluşturulan her bir konuşmacının şablon kod kitap vektörleriyle karşılaştırılması sürecini içermektedir.



Şekil 4.1. Tasarlanan konuşmacı tanıma sistemi blok diyagramı

4.1. Eğitim Aşaması

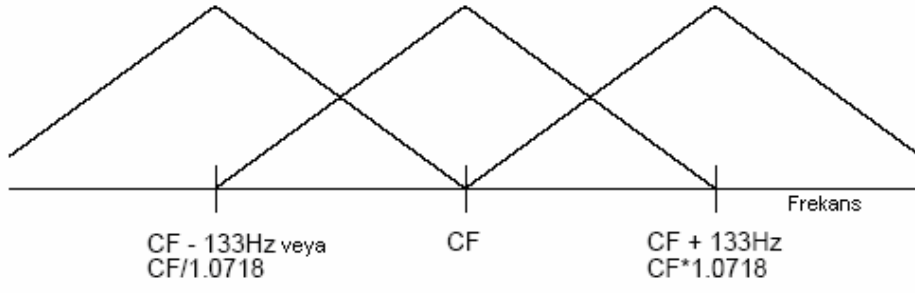
Eğitim aşamasında TURTEL veri tabanındaki kayıtlı 5 erkek ve 5 kadın konuşmacı kullanılmıştır. Bu aşamanın amacı her bir konuşmacının referans şablonunu oluşturacak kod kitaplarını bulmaktır. Bu aşamada öznitelik vektörlerinin elde edilmesi ve kod kitap tasarımı olarak ikiye ayırabilir.

4.1.1. Öznitelik vektörlerinin elde edilmesi

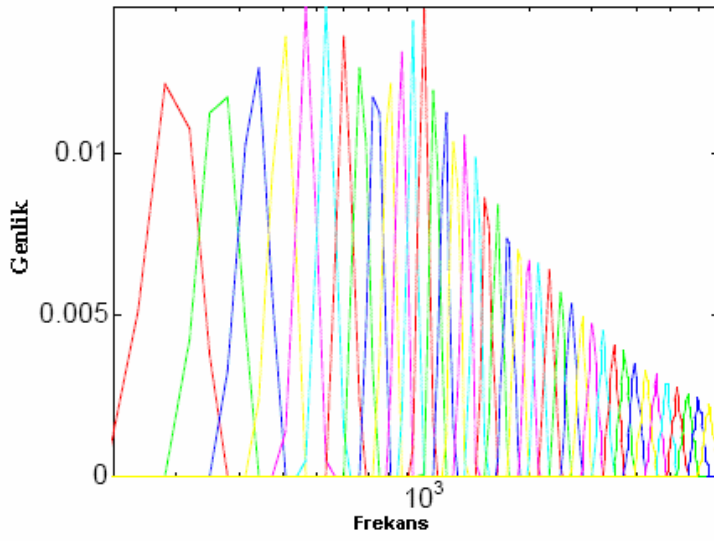
Öznitelik vektörlerinin bulunması sürecinde her bir konuşmacı için EK’te verilen kayıtlı 373 kelimededen bölüm 2.8.1.3’de anlatılan yöntemler kullanılarak mel-ölçeklendirilmiş cepstrum katsayıları elde edilmiştir. Bunun için önce konuşmacılara ait konuşma sinyalleri 2.8.1.3.2. bölümde anlatıldığı gibi 30 ms’lik çerçeveler ayrılmıştır. Pencereleme işleminde hanning pencere kullanılmıştır. Herbir pencere için 12 adet mel-ölçeklendirilmiş cepstral katsayı elde edilmiştir.

Pencerelenmiş konuşma sinyal çerçevelerinin 1024 nokta FFT’si alınarak konuşma sinyali frekans bölgesine taşınmıştır. Bir sonraki adım FFT spektrumundan mel spektruma geçmek olmuştur. Bunun için FFT spektrumunun genlik değerlerinin mutlak kare değerleri hesaplanarak güç spektrumuna geçilmiştir.

Mutlak genlik değerlerinin kareleri hesaplanan FFT noktaları band geçiren üçgen dalgalardan oluşan mel-filtre bankasından geçirilmiştir.. Burada kullanılan mel filtre bankası, 13 adet doğrusal olarak yerleştirilen filtreyi takip eden 27 adet logaritmik filtreden meydana gelmektedir. Merkez frekanslar arasında 133 Hz veya 1.0718 çarpan olacak şekilde yerleştirilmiştir.

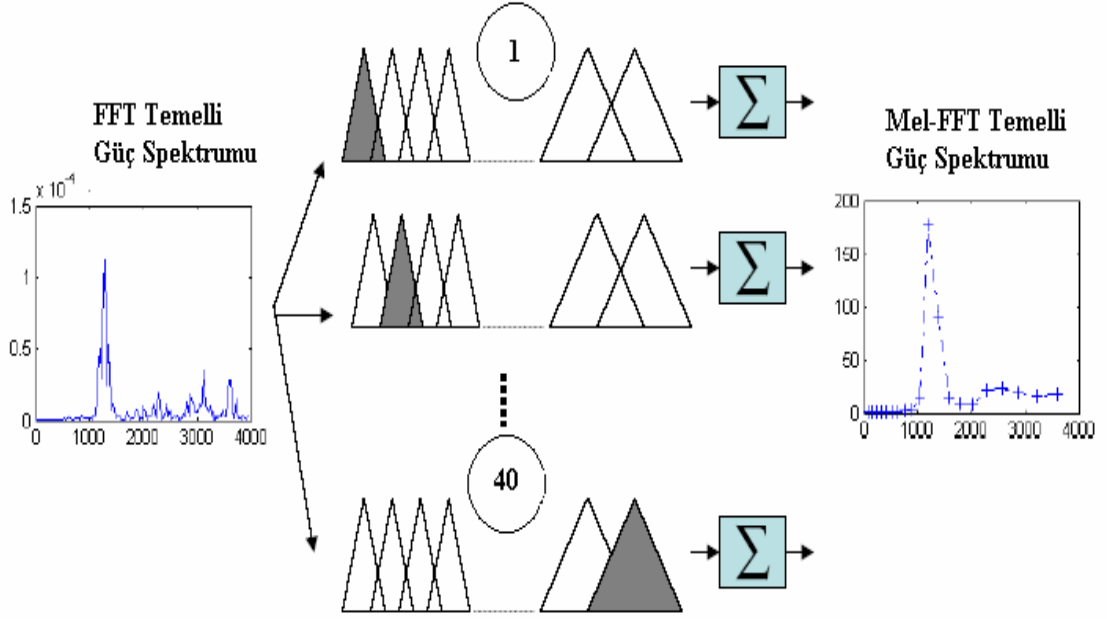


Şekil 4.2. Mel filtre bankası merkez frekans dizilim gösterimi



Şekil 4.3. 40 adet filtrenin frekans tepkisi

Güç spektrumunun her bir üçgen mel filtre ile çarpılıp toplanması sonucu şekil 4.4'te gösterilen mel spektruma geçiş yapılmış olur.



Şekil 4.4. Mel filtre bankası çıkış örneği

Mel spektrumun elde edilmesinden sonraki basamak mel cepstral katsayıların elde edilmesidir. Bunun için mel filtre bankası çıkışının logaritması alınır ve kesikli kosinüs dönüşümü ile zaman bölgesine geçilir. Elde edilen bu katsayılar mel-ölçeklendirilmiş FFT temelli cepstral katsayılardır. DCT kullanılmasının nedenlerinden biri IFFT uygulamasının DCT'ye göre kompleks aritmetik işlemler gerektirmesidir. Aynı zamanda sinyal ayna simetriği gösteren gerçel bir sinyaldir. DCT, gerçel bir sinyalde artırlıklığın ortalamasını kullanarak FFT'den daha verimli bir uygulama gerçekleştirir.

4.1.2. Kod kitap tasarımı

5 erkek ve 5 kadın konuşmacının öznitelik vektörleri kullanılarak eğitim dizi oluşturulmuştur.

$$T = \{x_1, x_2, \dots, x_{10}\} \quad (4.1)$$

4.2 eşitliğinde tüm konuşmacıların herbiri 12 boyutlu olan öznitelik vektör sayıları verilmektedir. $x_{i,j}$ gösteriminde $i=1,2,\dots,10$ 'ler konuşmacı indisini ve $j=1,2,\dots$,toplam eğitim verisi vektör sayısını göstermektedir

$$\begin{aligned}
 x_1 &= \left(\begin{array}{c} \vec{x}_{1,1}; \vec{x}_{1,2}; \dots; \vec{x}_{1,9585} \end{array} \right) \\
 x_2 &= \left(\begin{array}{c} \vec{x}_{2,1}; \vec{x}_{2,2}; \dots; \vec{x}_{2,8516} \end{array} \right) \\
 x_3 &= \left(\begin{array}{c} \vec{x}_{3,1}; \vec{x}_{3,2}; \dots; \vec{x}_{3,8090} \end{array} \right) \\
 x_4 &= \left(\begin{array}{c} \vec{x}_{4,1}; \vec{x}_{4,2}; \dots; \vec{x}_{4,8357} \end{array} \right) \\
 x_5 &= \left(\begin{array}{c} \vec{x}_{5,1}; \vec{x}_{5,2}; \dots; \vec{x}_{5,8399} \end{array} \right) \\
 x_6 &= \left(\begin{array}{c} \vec{x}_{6,1}; \vec{x}_{6,2}; \dots; \vec{x}_{6,10020} \end{array} \right) \\
 x_7 &= \left(\begin{array}{c} \vec{x}_{7,1}; \vec{x}_{7,2}; \dots; \vec{x}_{7,10830} \end{array} \right) \\
 x_8 &= \left(\begin{array}{c} \vec{x}_{8,1}; \vec{x}_{8,2}; \dots; \vec{x}_{8,10211} \end{array} \right) \\
 x_9 &= \left(\begin{array}{c} \vec{x}_{9,1}; \vec{x}_{9,2}; \dots; \vec{x}_{9,10141} \end{array} \right) \\
 x_{10} &= \left(\begin{array}{c} \vec{x}_{10,1}; \vec{x}_{10,2}; \dots; \vec{x}_{10,9506} \end{array} \right)
 \end{aligned} \tag{4.2}$$

6 bit kod kitabı ile tüm konuşmacıların ifade edilebileceği düşünüldüğünden kod vektör sayısı 64 olarak seçilmiştir.

$C = \{c_1, c_2, \dots, c_{64}\}$ 'de kod vektörlerini gösterebiliriz.

Kod kitabı tasarım aşamasında verilen eğitim dizisi vektörleri(T) ve kod vektör sayısı (N) için öyle kod vektörleri ve kodlama bölgesi bulunmalı ki D_{Ort} minimum olsun.

Bölüm 2.6.2.3.1’de açıklanan Linde, Buzo ve Gray algoritması kullanılarak herbir konuşmacı için 64 kod vektöründen oluşan 6 bit’lik kod kitabı tasarlanmıştır.

4.2. Test Süreci

Test süreci, tasarlanan vektör nicemleme tabanlı konuşmacı tanıma sisteminin öznitelik eşleme safhası olarak da adlandırılabilir. Test aşamasında konuşmacıların EK’te verilen kayıtlı cümleleri kullanılmıştır. Bu datalarında bölüm 2.8.1.3’de anlatılan yöntemlere göre öznitelik vektörleri çıkarıldıktan sonra referans model olarak hazırlanan kod kitabı vektörleriyle en yakın uzaklığı veren kod vektörleri bulunmuştur. Uzaklık kriteri olarak euclid uzaklığı kullanılmıştır. Bilinmeyen konuşmacı bu kod vektör sayısı en fazla olan konuşmacı olarak atanarak çizelge 4.1’de verilen sonuçlar elde edilmiştir.

Çizelge 4.1. 6-bit vektör nicemleyicinin konuşmacı tanıma performansı

Konuşmacı	Bilinmeyen Konuşmacı Test Sayısı	Konuşmacının Tanınma Sayısı	Başarı (%)
1. Erkek	10	10	100
2. Erkek	10	10	100
3. Erkek	10	10	100
4. Erkek	10	3	30
5. Erkek	10	10	100
1. Kadın	10	0	0
2. Kadın	10	10	100
3. Kadın	10	10	100
4. Kadın	10	10	100
5. Kadın	10	10	100

5. TARTIŞMA ve SONUÇ

6 bit olarak tasarlanan kod kitapları için bir erkek bir kadın konuşmacı tanımada problemle karşılaşmıştır. 4. erkek konuşmacı 10 testin 3'ünde tanınabilirken, 1. kadın konuşmacı tüm testlerde tanınamamıştır. Hem bu problemi giderebilmek hem de konuşmacı tanıma performansı üzerindeki etkisini görebilmek için önce kod kitap boyutu değiştirilmiştir. İkinci olarak, eğitim aşamasında kelimelerin kullanıldığı sistem, test aşamasında cümle yerine kelime ile test edilmiştir. Son olarak konuşmacı sayısı artışının konuşmacı tanıma performansı üzerindeki etkisi incelenmiştir.

5.1. Kod Kitap Boyutunun Tanıma Performansı Üzerindeki Etkisi

Tanıma performansını iyileştirmek için ilk akla gelen yöntem konuşmacı kod kitap boyutunu değiştirmektir. Bunun için 6 bit dışında 5 ve 7 bit kod kitapları da bölüm 4.1.2'de açıklanan yöntemlere göre tasarlanmıştır.

7-bit, 6-bit, 5-bit kod kitapları için konuşmacı tanıma yüzdeleri aşağıdaki gibidir:

Çizelge 5.1. Değişik kod kitap boyutları için tanıma performans değişimi

Konuşmacı \ Kod Kitap Boyutu	7-bit Başarı(%)	6-bit Başarı(%)	5-bit Başarı(%)
1.Erkek Konuşmacı	100	100	100
2.Erkek Konuşmacı	100	100	100
3.Erkek Konuşmacı	100	100	100
4.Erkek Konuşmacı	60	30	0
5.Erkek Konuşmacı	100	100	100
1.Kadın Konuşmacı	90	0	0
2.Kadın Konuşmacı	100	100	90
3.Kadın Konuşmacı	100	100	100

4.Kadın Konuşmacı	100	100	70
5.Kadın Konuşmacı	100	100	100

Konuşmacılar için tasarlanan kod kitabındaki vektör sayısı artırıldıkça konuşmacı tanıma yüzdesi düşük olan konuşmacıların tanınma yüzdesinin arttığı; konuşma yüzdesi yüksek olan konuşmacılar için ise tanınma yüzdesinin azalmadığı gözlemlenmiştir. Bunun nedeni kod kitap boyutu artırıldığında, boyutu artan nicemleyicinin eski kitle merkezlerini içermesinden kaynaklanmaktadır. Vektör nicemleyici elimizdeki veriyi sıkıştırarak en az bilgiyle ifade etmemizi sağlar. Ama bu veri kaybına sebep olacaktır. Bununla birlikte kod kitap boyutu arttıkça tasarlamak süresi buna paralel olarak artış göstermektedir. Dolayısıyla konuşmacı veya konuşma tanıma gibi önemli uygulamalarda tanıma performansının değişim göstermediği kod vektörlerinden oluşan nicemleyici bizim için optimum kabul edilebilir. Bu deneysel çalışma 10 kişilik bir veritabanının 128 kod vektöründen oluşan kod kitaplarıyla rahatlıkla ifade edebileceğini göstermektedir.

5.2. Test Sürecinde Sistemin Cümle Yerine Kelime ile Test Edilmesinin Tanıma Performansına Etkisi

Konuşmacı tanıma aşamasında kayıtlı konuşmacılar için ilk olarak test amacıyla EK'de verilen cümleler kullanılmıştır. Eğitim aşamasında kelimelerin kullanıldığı sistem, test aşamasında hızlı bir şekilde sonuca ulaşabilmek maksadıyla cümle yerine kelime ile test edilmiştir. Test kelimeleri, kayıtlı cümlelerden Goldwave adlı ses işleme programında kesilerek elde edilmiştir. Tanıma performansı çizelge 5.2'de verildiği şekilde değişmiştir(7-bit kod kitap için) :

Çizelge 5.2. Kelime/cümle ile test edilen sistemde tanıma performansları

Konuşmacı	1.Erkek	2.Erkek	3.Erkek	4.Erkek	5.Erkek	1.Kadın	2.Kadın	3.Kadın	4.Kadın	5.Kadın
Başarı(%) (Kelime)	90	90	100	20	100	90	90	100	100	100
Başarı(%) (Cümle)	100	100	100	60	100	90	100	100	100	100

4. erkek şahsın tanıma performansının, sistemin test aşamasında cümle yerine kelime ile test edilmesi durumunda düştüğü ve tanınmadığı gözlemlenmiştir. Aynı şekilde diğer konuşmacılar tanınabilse de tanıma performanslarının düştüğü gözlemlenmiştir. Bu da konuşma tanıma uygulamalarının test aşamasında kod vektörleriyle kıyaslama yapılacak test vektör sayısındaki artışın performansı arttırdığını göstermektedir. Çünkü bu çalışmadaki gibi referans model olarak hazırlanan kod kitabı vektörleriyle, en yakın uzaklığı veren kod vektör sayısı en fazla olan konuşmacının bilinmeyen konuşmacı olarak atandığı uygulamalarda konuşmacı öznelik vektör sayısındaki artış hata yapma olasılığımızı azaltacaktır.

5.3. Konuşmacı Sayısı Artışının Tanıma Performansına Etkisi

Konuşmacı sayısı artışının tanıma performansına etkisini gözlemek amacıyla 7 bit kod kitap için konuşmacı sayısı belirli aralıklarla 20'ye kadar artırıldı.

Çizelge 5.3. 7-bit kod kitap kullanıldığında konuşmacı sayısı artışının performansa etkisi

Konuşmacı Sayısı	10	13	16	20
Tanıma Performansı(%)	100	92,3	75	70

Konuşmacı sayısı arttıkça tanıma performansı düşmektedir. 10 kişi için tanıma performansı %100 iken konuşmacı sayısı 20'ye çıkarıldığında tanıma performansının %70'e düştüğü gözlemlenmiştir. Vektör nicemeleme tekniklerinin az sayıda konuşmacı içeren veritabanları için daha iyi performans verdiği görülmektedir.

Sistemimizin performansını ortaya koymak amacıyla alıcı işletim karakteristiği eğrisine başvuruldu. Şekil 5.1'de kesikli çizgilerle ROC eğrisi gösterilmektedir. Bölüm 2.9'da yapılan açıklamalara göre elde edilen ölçüler çizelge 5.4'de verilmektedir. Sistem performansını bu eğri altında kalan alanla ölçebiliriz. Bu alan 1 ise sınıflandırıcı mükemmel, 0.5'den küçükse kullanışsız, 0.5-1 arasında ise iyi olarak kabul edilir. Konuşmacı sayısının 20'ye artırıldığı durum için eğrinin alanı matlabda hesaplandığında 0.71 birim² olduğu bulunmuştur. Dolayısıyla tasarlanan sistem tatmin edici bir performans ile çalışmaktadır.

Çizelge 5.4. ROC eğrisi ölçüleri

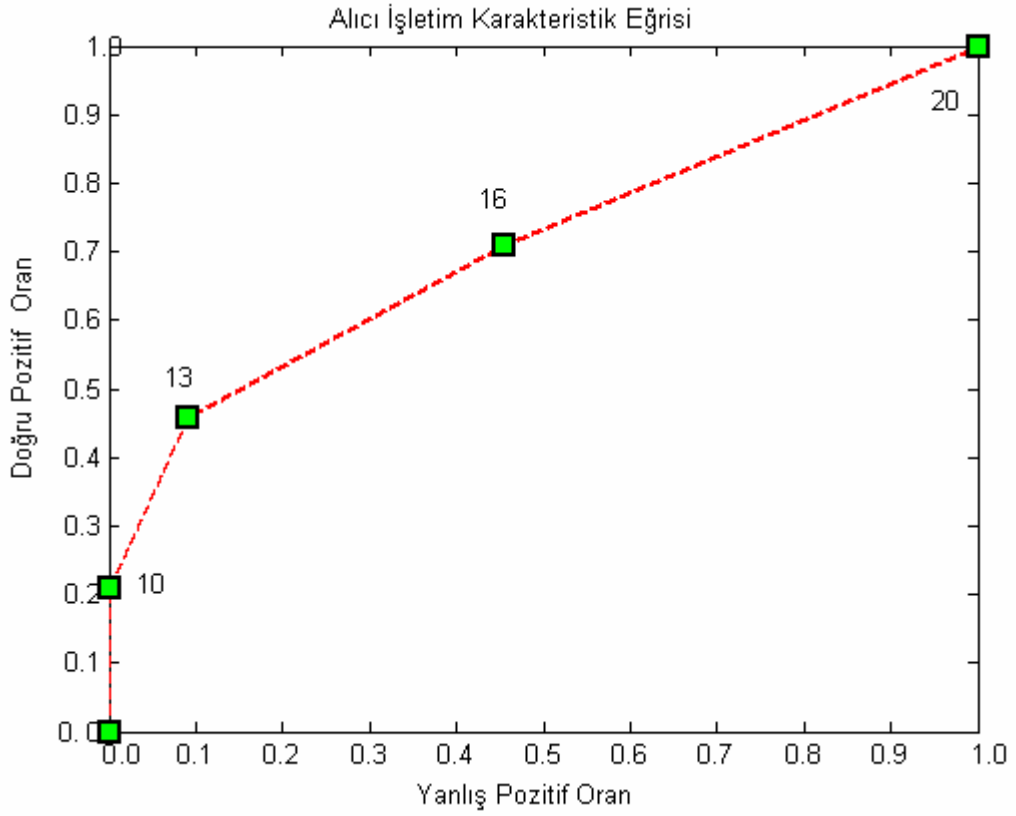
Sistem Değerlendirmesi (Tanıma Performansı)	Konuşmacı Sayısı				
	10	13	16	20	TOPLAM
Tanınan Konuşmacı Sayısı	10	12	12	14	48
Tanınamayan Konuşmacı Sayısı	0	1	4	6	11

$$\begin{aligned} DP &= 10/48 = 0.208 \\ YP &= 0/11 = 0 \end{aligned}$$

$$\begin{aligned} DP &= 22/48 = 0.458 \\ YP &= 1/11 = 0.09 \end{aligned}$$

$$\begin{aligned} DP &= 34/48 = 0.708 \\ YP &= 5/11 = 0.45 \end{aligned}$$

$$\begin{aligned} DP &= 48/48 = 1 \\ YP &= 11/11 = 1 \end{aligned}$$



Şekil 5.1. Alıcı işletim karakteristik eğrisi

Konuşma ve konuşmacı tanıma yöntemleri, ülkemizde son zamanlarda potansiyel bir araştırma dalı haline gelmiştir. Bu doğrultuda gerçekleştirilen çalışmalar henüz yeterli sayılmasa da, bu açık yapılacak çalışmalarla hızla kapatılabilir.

Bu konuda daha ileri bir çalışma olarak, daha verimli, konuşmadan bağımsız gerçek zamanda çalışan bir konuşmacı tanıma sisteminin gerçekleştirilmesi düşünülebilir. Böyle bir durumda, daha farklı örüntü kümelerine ve daha karmaşık eğitim süreçlerine ihtiyaç duyulacaktır.

KAYNAKLAR

- Deller, J.R., Hansen, H.L. and Proakis, J.R. 2000. Discrete-Time Processing of Speech Signals. Prentice Hall
- Ergenç, İ. 1995. Konuşma Dili ve Türkçenin Söyleyiş Sözlüğü/ Bir Deneme. Türk Dilleri Araştırma Dizisi:8
- Gish, M. and Wolf, J. 1986. Methods and experiments for text-independent speaker recognition over telephone channels. Speech and Signal Processing Proc. IEEE Int. Conf Acoust.; 865-868
- Grey, D.R. and Morgan, B.J.T. 1972. Some aspects of ROC curve-fitting: Normal and logistic models. [Corrections and improvements to Dorfman-Alf; minimum chi-square bilogistic]. Journal of Mathematical Psychology, Vol 9; 128-139,
- Hanzo, J., Somerville, F.C.A. and Woodard, J.P. 2001. Voice Compression and Communications. Wiley-Interscience
- Karamursel, A. 2002. Speech Therapy on Organic and Functional Speech Disorders, Evaluation of Medical and Surgery Treatment Results Using Aoustic and Spectrographic Analyses, Ankara University Head and Neck Department Doctorate Thesis, Ankara
- Kinnunen, T., Kilpelainen, T. and Franti, P. 2000. Comparansion of Clustering Algorithms in Speaker Identification
- Kondoç, A.M. 1990. Digital Speech. John Wiley & Sons, New York
- Linde Y., Buzo A. and Gray R.M. January 1980. An Algorithm for Vector Quantizer Design. IEEE Trans. On Communications, 28(1); 84-95
- Liu, L., He, J. and Palm, G. Signal Modeling For Speaker Identification
- Lusted, B. 1978. General Problems in Medical Decision Making with Comments on ROC Analysis. Semin Nucl Med., Vol 8; 299-306
- Molau, S., Pitz, M., Schluter, R. and Ney, H. 2001. Computing Mel-frequency cepstral coefficients on the power spectrum. Acoustics, Speech, and Signal Processing Proceedings, Vol 1; 73 -76

- Ong, S., Sridharan, S., Yang, C.H. and Moody, M.P. 1996. Comparison of Four Distance Measures for Long Time Text-Independent Speaker Identification. ISSPA
- Picone, J. September 1993. Signal Modeling Techniques in Speech Recognition. Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop, Vol. 81(9); 1215-1247
- Rabiner, L. and Juang, B.H. 1993. Fundamentals of Speech Recognition. Prentice Hall
- Rabiner, R. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, Vol 77(2); 257-286
- Reynold, D.A. and Rose, R.C. January 1995. Robust Text-Independent Speaker Identification Using Gaussian Speaker Models
- Swets, J.A., 1995. Signal Detection Theory and ROC Analysis in Psychology and Diagnostics: Collected Papers, Lawrence Erlbaum Associates
- Tokuda, K., Kobayashi, T. and Imai, S. April 1994. Recursive Calculation of Mel-Cepstrum from LP Coefficients
- Wildermoth, B.R., January 2001. Text-Independent Speaker Recognition Using Source Based Features. IEEE Transaction on Speech and Audio Processing, Vol 3(1); 72-83
- Wong, E. and Sridharan, S. 2001. Comparison of Linear Prediction Cepstrum Coefficients and Mel-Frequency Cepstrum Coefficients for Language Identification. Intelligent Multimedia, Video and Speech Processing Proceedings; 95 -98
- Zeng, Y. November 2000. Dynamic Time Warping Digit Recognizer

EK. Konuşmacı veri tabanında kayıtlı kelimeler ve cümleler

1	bitki	46	içen	91	çabukluk	136	nene
2	tutulmak	47	kabartma	92	ağırlık	137	gibi
3	bir	48	acılık	93	kalınca	138	yaşar
4	gösterilen	49	sürelî	94	görevliyi	139	yıl
5	nakışlarla	50	kimsenin	95	karşılmalıdır	140	dinle
6	kaza	51	testi	96	katılmış	141	yakasız
7	dervişlerin	52	boğumları	97	yarıp	142	yoluyla
8	veya	53	ilgili	98	kendi	143	varken
9	keserek	54	sonra	99	sevmeyecek	144	önce
10	yapılan	55	kadar	100	kıran	145	ambar
11	karışyorsun	56	İçin	101	yalıtkan	146	fikirli
12	bağlanarak	57	öğrenilmesi	102	matlaşmak	147	alttaki
13	getirmek	58	kaymaklı	103	el	148	mizan
14	başında	59	şaşırmak	104	türlü	149	olduğundan
15	yönteminde	60	ayrıldığı	105	sunum	150	acemileşmek
16	parçanın	61	edilme	106	şeyi	151	yazı
17	arama	62	maddeler	107	aştır	152	daha
18	sağlayan	63	indirimden	108	bekleyen	153	nokta
19	her	64	kumaştan	109	topluluğu	154	makam
20	eve	65	güneşte	110	sıkıntı	155	değer
21	hastalıklara	66	geldiğine	111	pekişti	156	keten
22	işaretlerini	67	yayınlara	112	donanım	157	onun
23	dolaplar	68	geçirmeden	113	açık	158	kaldırım
24	buğday	69	işi	114	sandım	159	çekmek
25	tahtası	70	kuruluşu	115	artıyor	160	arasındaki
26	genellikle	71	eylem	116	duyguları	161	mübalağa
27	isimlerden	72	cariye	117	çeşit	162	kültürünü
28	bağlantısını	73	düz	118	lise	163	sardı
29	dikkatli	74	merkez	119	yüklenmek	164	durumu
30	işine	75	tuvalet	120	yoksul	165	ali
31	davranışta	76	zaman	121	çıkacak	166	halka
32	bölgesinde	77	kapanmak	122	dal	167	şaştı
33	bere	78	hissetmek	123	koruyucu	168	dönem
34	konu	79	çok	124	hayat	169	alman
35	kurallarını	80	kenetlenmiş	125	silmiş	170	familyası
36	asalak	81	etkiler	126	düşünce	171	selin
37	biçimi	82	bu	127	ortaya	172	ekte
38	tazelemek	83	mesleği	128	belirti	173	özen
39	yersiz	84	lacivert	129	aksiyon	174	ermezler
40	durumuna	85	iklim	130	büyük	175	kararsızlık
41	bakımından	86	harflerle	131	azlığını	176	abartıcı
42	alnına	87	atardamar	132	kullanılır	177	tatil
43	girebilir	88	vurmasını	133	ciddiyet	178	çalı
44	sevecenlik	89	istek	134	gereksiz	179	kavuşturmak

45	anlamı	90	bulunma	135	bütün	180	merhamet
181	canlı	218	aydın	266	zarar	314	prizma
182	renk	219	sırtından	267	acelecilik	315	hakem
183	benzer	220	kuzey	268	deniz	316	kenter
184	başladı	221	ölümünden	269	kuş	317	asla
185	dilde	222	çivi	270	iyi	318	düğüm
186	kötü	223	yabansı	271	yaptıkları	319	evrim
187	savaş	224	afrika	272	söyledi	320	siyah
188	vakti	225	alfabetik	273	ilişki	321	delil
189	uykusuz	226	otsu	274	dünya	322	dede
190	doğru	227	garip	275	çaylak	323	özüt
191	imtiyazlar	228	ördek	276	ünlüler	324	sekiz
192	abide	229	raf	277	polis	325	çocuk
193	devletin	230	lekeleri	278	sera	326	demokrasi
194	anlatmak	231	sarmaşık	279	altı	327	ebe
195	görülmemiş	232	titizlikle	280	burun	328	eğri
196	mensup	233	esmer	281	hırka	329	kaçak
197	niteliği	234	akmak	282	hüner	330	hap
198	alışmamış	235	gelmek	283	surat	331	fena
199	namaz	236	ordu	284	aynı	332	imdat
200	ceviz	237	oku	285	o	333	bora
201	mutlu	238	demli	286	ani	334	boy
202	türk	239	yitimi	287	soru	335	ruj
203	incitici	240	adıyla	288	ses	336	füze
204	girdi	241	nitelikte	289	hata	337	sülük
205	abajur	242	çengi	290	pilav	338	aldatmak
206	başka	243	bunu	291	sahil	339	illet
207	sayı	244	ettiğin	292	küçük	340	proje
208	yüzde	245	bina	293	ülke	341	olmayan
209	hane	246	taş	294	sanki	342	jale
210	kin	247	söz	295	dönünce	343	jilet
211	emel	248	işiyle	296	kızıl	344	sakat
212	irade	249	ray	297	sükse	345	sıfır
213	aşı	250	yumuşak	298	saçlar	346	bir
214	uzunluğu	251	kıl	299	oynatma	347	iki
215	şan	252	kıvanç	300	türkiye	348	üç
216	insan	253	üst	301	varis	349	dört
217	oyun	254	kol	302	gamlı	350	beş
218	aydın	255	libas	303	kısa	351	altı
219	sırtından	256	ağaç	304	gerçek	352	yedi
220	kuzey	257	dizi	305	zorla	353	sekiz
221	ölümünden	258	kereste	306	kurt	354	dokuz
222	çivi	259	ancak	307	üzüm	355	on
223	yabansı	260	yapma	308	dış	356	yirmi
224	afrika	261	sattığım	309	tamam	357	otuz
214	uzunluğu	262	müdür	310	şam	358	kırk
215	şan	263	şimşek	311	istedi	359	elli
216	insan	264	gözler	312	göreceğim	360	altmış

217	oyun	265	beyin	313	yapacağım	361	yetmiş
362	seksen	365	bin	368	trilyon	371	katrilyar
363	doksan	366	milyon	369	trilyar	372	evet
364	yüz	367	milyar	370	katrilyon	373	hayır

1. Yağmurdan sonra güneş çıkarsa gökkuşağı oluşur.
2. Rezervasyon için bilet almak gerekir mi?
3. Projenin faturası bana çıkartıldı.
4. Verdiğin çiçekte örümcek var.
5. Askerler depremde kurtarma çalışmalarına katıldılar.
6. Ümit demektir yağmur çiftçiler için
7. Sinemaya bilet bulabildin mi?
8. Ayvalık'ta güneşin batışını seyretmelisin.
9. Yemek için rezervasyon yaptırdım.
10. Acı bir iz bıraktı deprem zihinlerimizde.
11. Dün manifaturacıya gittim.
12. Dünyanın en harika dokumacılarıdır örümcekler.
13. Fıkranın konusu bir çiçek.
14. Kafanda nasıl projeler var.
15. Demek büyüyünce asker olacaksın.

ÖZGEÇMİŞ

1980 yılında Denizli’de doğdu. İlk öğrenimini, orta ve lise öğrenimini Kütahya’da tamamladı. 1998 yılında girdiği Ankara Üniversitesi Elektronik Mühendisliği Bölümü’nden 2002 yılında Elektronik Mühendisi ünvanı ile mezun oldu. 2002 yılının Eylül ayında Ankara Üniversitesi Fen Bilimleri Enstitüsü’nde yüksek lisans öğrenimine başladı.

Ankara Üniversitesi Mühendislik Fakültesi Elektronik Mühendisliği Bölümü bünyesinde 2002 yılından beri araştırma görevlisi olarak görev yapmaktadır.