İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ÖRNEKLEM TABANLI GÜRBÜZ KONUŞMA TANIMA

YÜKSEK LİSANS TEZİ

Fatih AKTÜRK

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

OCAK 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ÖRNEKLEM TABANLI GÜRBÜZ KONUŞMA TANIMA

YÜKSEK LİSANS TEZİ

Fatih AKTÜRK
(504121317)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Ender Mete EKŞİOĞLU

OCAK 2015

İTÜ, Fen Bilimleri Enstitüsü'nün 504121317 numaralı Yüksek Lisans Öğrencisi **Fatih AKTÜRK**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"ÖRNEKLEM TABANLI GÜRBÜZ KONUŞMA TANIMA"** başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Doç. Dr. Ender Mete EKŞİOĞLU** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Doç. Dr. İlker BAYRAM** İstanbul Teknik Üniversitesi	.....
	Yrd. Doç. Dr. Ahmet Korhan TANÇ Kırklareli Üniversitesi	.....
		.....

Teslim Tarihi : **15 Aralık 2014**
Savunma Tarihi : **21 Ocak 2015**

Aileme ve arkadaşlarıma,

ÖNSÖZ

Tez çalışmam boyunca yardımlarını esirgemeyen tez danışmanım Sayın Doç. Dr. Ender Mete EKŞİOĞLU'na teşekkür ederim. Ayrıca bu süreçte bana destek olan aileme ve arkadaşlarıma da teşekkür ederim. Son olarak Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'na BİDEB 2210-A Yurt İçi Yüksek Lisans Burs Programı kapsamındaki desteği için teşekkürlerimi arz ederim.

Ocak 2015

Fatih AKTÜRK
(Mühendis)

İÇİNDEKİLER

Sayfa

KISALTMALAR

SMM	**:** Saklı Markov Model
HTK	**:** Hidden Markov Model Toolkit
NOMA	**:** Negatif Olmayan Matris Ayrıştırması
SA	**:** Sıkıştırılmış Algılama
KL	**:** Kullback-Leibler
İGO	**:** İşaret Gürültü Oranı

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

ÖRNEKLEM TABANLI GÜRBÜZ KONUŞMA TANIMA

ÖZET

Bu çalışmadaki amaç örneklem tabanlı yöntemlerin gürbüz konuşma tanıma konusundaki başarımını ölçmektir. Bu amaçla "Seyrek Sınıflandırma" isimli bir yöntem gerçeklenmiştir. Ayrıca elde edilen sonuçlar, yöntemin başarımını ölçmek adına klasik SMM-GKM tabanlı bir konuşma tanıma sisteminin sonuçları ile karşılaştırılmıştır.

Çalışmanın birinci bölümünde otomatik konuşma tanıma sistemleri hakkında kısa bilgi verilmiştir. Devamında ise gürültünün konuşma tanıma işlemine olan olumsuz etkisinden bahsedilmiştir ve literatürde kullanılan gürbüz konuşma tanıma yöntemlerine değinilmiştir. Bunlardan biri çok durumlu eğitim yöntemidir. Ancak bu yöntem yüksek başarım oranları elde edebilmek için çok fazla gürültü çeşidini içinde barındıran veri setleri gerektirmektedir. Bunun yanında gürültü içeren konuşmalardan eğitilen modeller temiz konuşmaları tanırken, temiz konuşmalardan eğitilen modellere göre daha kötü başarım vermektedir. Bu amaçla literatürde model dengeleme ve öznitelik iyileştirme gibi yaklaşımları kullanan yöntemler kullanılmaktadır. Bunlara ek olarak, gürbüz konuşma tanıma için, literatürdeki örneklem tabanlı konuşma tanıma yöntemlerinden biri olan Seyrek Sınıflandırma yöntemi kullanılmaktadır. Bu yöntem modeli ya da öznitelikleri güncellemek yerine, gürültülü konuşmaları gürültü ve konuşma örneklemlerinin seyrek doğrusal birleşimi cinsinden ifade ederek bir kaynak ayrıştırması yapmaktadır. Bu yolla gürültülü bir konuşmanın temiz konuşma bileşenine yakınsanmaktadır.

İkinci bölümde Markov zinciri ve SMM hakkında ayrıntılı bilgi verilmiştir.SMM'ye ait üç problem ve bu problemlere çözüm olarak kullanılan algoritmalar anlatılmıştır.

Üçüncü bölümde konuşma tanıma sisteminin bileşenleri, SMM'nin konuşma tanıma yapısında nasıl kullanıldığı, eğitim ve test işlemlerinin nasıl yapıldığı, öznitelik vektörlerinin çıkarım şeması verilmiştir.

Dördüncü bölümde örneklem tabanlı konuşma tanıma hakkında bilgi verilmiştir. Öncelikle literatürde çokça kullanılan ve Destek Vektör Makineleri, Yapay Sinir Ağları ve Saklı Markov Modelleri gibi örnekleri olan global modelleme teknikleri ve örneklem tabanlı modelleme tekniklerinin farkı üzerinde durulmuştur. Daha sonra örneklem tabanlı teknikleri konuşma tanımaya uygularken takip edilen adımlar açıklanmıştır. Bölümün devamında ise seyrek gösterimin örneklem tabanlı konuşma tanıma uygulamalarında kullanımı incelenmiştir. Son olarak ise bu çalışmada gerçeklenen örneklem tabanlı Seyrek Sınıflandırma yöntemi anlatılmıştır.

Beşinci bölümde yapılan deneyler ve sonuçları anlatılmıştır. Öncelikle eğitim ve test setlerinin nasıl elde edildiği ve hangi işlemlerden geçirildiği anlatılmıştır. Daha sonra bu setler kullanılarak HTK ile yapılan eğitim ve test işlemleri, sonrasında ise seyrek sınıflandırma yöntemi için yapılan çalışmalar anlatılmıştır.

Bu çalışmanın sonucunda Seyrek Sınıflandırma yöntemi yardımıyla bir konuşma tanıma sistemi gerçeklenmiştir. Bu yöntem HTK yardımıyla gerçeklenen klasik bir konuşma tanıma sisteminin sonuçları ile karşılaştırılmış ve Seyrek Sınıflandırma yönteminin düşük İGO değerleri için SMM-GKM tabanlı sisteme göre daha iyi başarım sağladığı görülmüştür.

EXEMPLAR BASED NOISE ROBUST SPEECH RECOGNITION

SUMMARY

This work aims to measure the achievement of exemplar based noise robust speech recognition technique which is called Sparse Classification. This technique was implemented in MATLAB environment and its results was compared to HMM-GMM system results which was implemented via HTK.

In the first chapter, we present introductory knowledge about the history of automatic speech recognition systems. While automatic speech recognition systems has become significantly accurate in clean speech conditions and for well pronounced speech, performance still degrades rapidly for conversational speech and when the speech signal is corrupted by noise. Such noises can be originated from environment, background music or other speakers. Automatic speech recognition performance drops with increasing levels of noise because, the observed acoustic features no longer match the acoustic models learned during training. To handle this shortcoming, multi-condition training approach was proposed in literature. But this approach requires large data sets containing several noise types. Even if this is achieved, multi-condition trained acoustic models cannot yield high accuracies on clean speeches like acoustic models trained from clean speeches.

The other approaches has been used in literature can be categorized into two groups: model compensation and feature enhancement. Model compensation techniques update the model parameters to reduce the mismatch between acoustic model and observed features while feature enhancement techniques update the observation due to same purpose. However, there is another method was proposed in literature to provide noise robustness called Sparse Classification. This method represents the noisy speech as sparse linear combinations of exemplars. In this context, exemplars are speech patches that spans the time interval of 25-300 ms. Sparse Classification automatically conducts a source separation because, if we collect the clean speech exemplars with their weights, we can estimate the clean speech part of a noisy speech.

In the second chapter, we review the Markov chains and the HMM structure. Also, we explain three problems of HMM and the algorithms which are used as solutions to these problems. These problems are likelihood computation, decoding and learning. For likelihood computation we are given an HMM structure and an observation sequence. We need to find this observation sequence for given HMM. Forward algorithm is the solution of this problem. Second problem is decoding. In this problem we are given an HMM structure and an observation sequence. We need to find the best hidden state sequence for this observation. Viterbi algorithm is the solution of this problem. Third and last problem is learning. We are given state alphabet and an observation sequence for this problem and we have to find the state transitions and observation density functions. Baum-Welch algorithm is the solution of this problem.

In the third chapter, we present how an HMM-GMM based system can be built and what are the main components of these systems. First, we explain the speech recognition architecture. Secondly, MFCC features and the extraction scheme is reviewed. In MFCC extraction scheme section, all steps are explained which are pre-emphasising, windowing, Discrete Fourier Transform, Mel Filter Bank and Log, Inverse Discrete Fourier Transform and extraction of delta features. Third subsection is about the acoustic likelihood computation and GMM. After that, we give language models and lexicons and how they are used in speech recognition architecture, decoding and training stages and finally evaluation metric Word Error Ratio. Also we give a connected digit recognition structure as an example to see the usages of these components.

In the fourth chapter, we provide the comparison between the global data modeling approaches like HMMs, Artificial Neural Networks and Support Vector Machines and the exemplar based approaches which are used in automatic speech recognition. Moreover, main steps that are used in exemplar based approaches are given. After that, the sparse representations with exemplar based methods and speech recognition applications of this combination are reviewed. Finally, Sparse Classification method which is implemented in this work is explained in detail.

Sparse Classification is a hybrid method which employs HMM state transition probabilities and exemplar weights for acoustic likelihoods instead of GMMs. If we have a dictionary that contains HMM state label information about every exemplar, then we can use this structure with classical Viterbi algorithm. For exemplars, we used Mel scaled magnitude energies with 23 frequency bands and 20 consecutive time frames. Hence, we have windows size of 23×20. Every time frame corresponds to 10 ms so, exemplars span time intervals length of 200 ms. Also, another approach which is used for modelling arbitrary length signals is presented which is called sliding window approach. In this approach, windows which have same length with exemplars, are shifted one frame at a time and for each window sparse representation is found via Non-Negative Matrix Factorisation. For every window, 200 iterations are run with the given update rule in Chapter 4. With this approach, for an arbitrary length speech file, sparse representations can be found.

The fifth chapter presents the experiments conducted in this work. The data set contains connected digit sequences from TIDIGITS. Noise signals obtained from NOIZEUS data set. The noise signals added to these clean connected digit sequences at different SNR values which are 20, 15, 10, 5, 0 and -5 dB for test set. There are also clean speech sets. For training set we added noises for only 20, 15, 10, 5 dB and also we have clean subset. We trained two HMM-GMM acoustic models via HTK. One of them is trained from clean speeches, and the other one is trained from multi-condition training set. These models employs word HMM structures which have 16 states and GMMs including 3 mixtures for every digit. Also, we have a silence model employing three states and GMMs including 6 mixtures for every state. We used MFCC vectors length of 39 and they contain 12 Mel cepstrum coefficients, energy feature, delta and delta delta components.

Speech recognition results were given for both acoustic models. Normally clean speech model gives very poor results especially for lower SNR values except for clean speech test sets. We also give Sparse Classification experiment results for three cases which are given in detail. Moreover we obtain the speech and noise dictionary of

exemplars. The speech part of this dictionary has been gained via forced alignment by using clean speech acoustic model and the clean version of training database. Finally we have very promising results especially for -5 dB subsets. HTK gives better results for the subsets which have higher SNR values. As a result, Sparse Classification approach has an effective aspect for highly corrupted speech by noise but it needs to be improved for speeches which have better qualities.

1. GİRİŞ

Konuşma günlük hayattaki iletişim kurma yollarının başında gelir ve bilgi edinimi adına önemli bir yoldur. Konuşma işaretini, bir insanın işleme ve anlamlandırma yeteneği ile aynı veya ondan daha iyi bir seviyede başarabilecek sistemler geliştirmek telefonun icadından bu yana önemli bir araştırma konusudur. Otomatik konuşma tanıma, konuşma işleme problemleri arasındaki en zor problemlerden biridir ve konuşma sinyalini, karşı düşen kelime dizisine çevirmeyi hedefler.

1.1 Otomatik Konuşma Tanıma Kısa Tarihçesi

Otomatik konuşma tanımanın ilk uygulamaları 1950'li yıllara kadar uzanan ayrık rakam tanıma uygulamalarıdır [1]. Geliştirilen sistem ile her bir rakam için referans bir örüntü çıkarılmış ve bu örüntülere göre bilinmeyen bir rakam tanınmaya çalışılmıştır. Rakamların örüntülerini belirlemek için formant frekansları kullanılmıştır. Benzer uygulamalar 1960'lı yıllarda da devam etmiş ancak tanıma işlemi yine rakamlar veya heceler ile sınırlı kalmıştır.

Otomatik konuşma tanıma alanında halen kullanılmakta olan Saklı Markov Model(SMM)'leri, 1970'li yıllarda aktif olarak kullanılsa da, teorik arka plan sadece alanın öncüsü olan laboratuvarlar tarafından biliniyordu. Bu konudaki metodolojinin netleşmesi 1980'li yıllara rastlamaktadır [2]. Devam eden 20 yıl boyunca SMM'ler konuşma tanıma sistemlerinde başarılı bir şekilde kullanılmıştır. Bu yıllarda konuşma tanıma sistemlerinin başarımı eskiye göre büyük oranda artmış ve geniş dağarcıklı sürekli konuşmalar içeren verileri tanıyabilen sistemler geliştirilmiştir.

Son yıllarda ise, işlemci teknolojilerinin gelişmesiyle birlikte önceden hız anlamında uygulanması mümkün olmayan yöntemler konuşma tanıma alanında kullanılmaya başlanmıştır. Bu duruma örnek olarak yapay sinir ağları [3] ve seyrek gösterim uygulamaları [4] verilebilir.

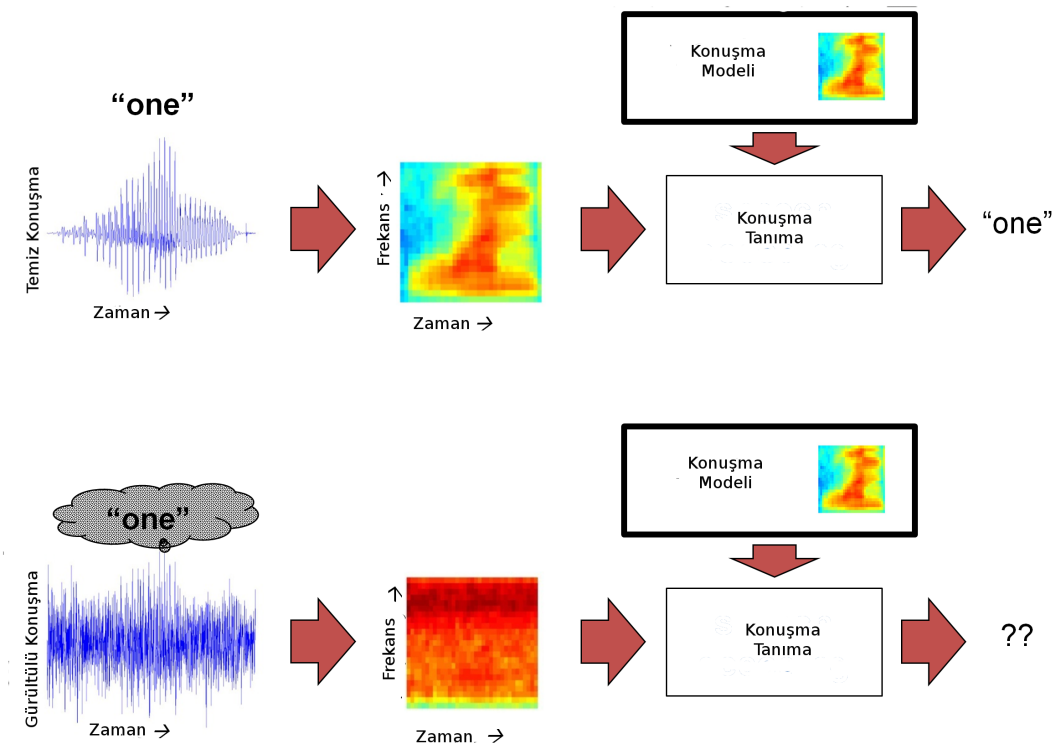
1.2 Gürbüz Konuşma Tanıma

Günümüzde kullanılan otomatik konuşma teknikleri, sessiz ortamlarda kaydedilmiş ve dikkatli bir şekilde telaffuz edilmiş konuşmaları tanımada gayet iyi başarım oranları vermektedir. Ancak karşılıklı konuşma ve arka plan gürültüsü içeren verilerde başarım ciddi anlamda düşmektedir. Bu gürültüler konuşmanın geçtiği ortamdan kaynaklanan harici sesler, arka planda çalan bir müzik veya aynı anda konuşan bir başka konuşmacıya ait sesler olabilir. Bu koşullar altında konuşma tanıma başarımının düşmesinin temel nedeni test sırasında karşılaşılan gözlem vektörleri ile eğitim sırasında öğrenilen akustik modeller arasındaki uyumunun azalmasıdır. Bu durum Şekil 1.1'de gösterilmiştir.

Şeklin üst kısmı temiz konuşma tanımayı, alt kısmı ise gürültülü konuşma tanımayı anlatmaktadır. Test sırasında gözlemlenen "one" kelimesi öncelikle akustik gücün spektrum-zaman dağılımına çevrilir. Daha sonra konuşma tanıma motoru daha önceden eğitilmiş bir akustik modeli kullanarak tanıma işlemini gerçekleştirir. Şekilden de görülebileceği gibi bu konuşmaya ait spektrum-zaman dağılımı, modelin eğitildiği verilerin sahip olduğu spektrum-zaman dağılımı ile uyuşmaktadır ve sonuç olarak konuşma doğru olarak tanınmıştır. Alt kısımda ise gözlem ve model arasında uyumsuzluk vardır ve bu durum konuşma tanıma motorunun konuşmayı doğru olarak tanımasını engellemektedir.

Bu soruna çözüm olarak çok durumlu eğitim yöntemi önerilmiştir [5]. Kısaca özetlemek gerekirse, sadece sessiz ortamda alınmış kayıtlardan oluşan konuşmalar yerine gürültülü ortamlarda kaydedilmiş konuşmalar eğitimde kullanılır ve eğitim ile test arasındaki uyumsuzluk giderilmeye çalışılır. Ayrıca eğitim verilerindeki gürültülerin hedef uygulama için anlamlı olmasına dikkat edilir.

Çok durumlu eğitim yöntemi durağan arka plan gürültüleri için etkin bir çözüm olmasına rağmen test ortamlarındaki gürültüler eğitimde karşılaşılan gürültülerden farklı olduğunda başarımı düşmektedir. Pratik uygulamalarda test sırasında karşılaşılabilecek gürültüleri tam olarak tahmin etmek mümkün olmadığından, çok durumlu eğitim bu soruna tam anlamıyla bir çözüm sunamamaktadır. Eğitim için çok fazla gürültülü veri gerektirmesinin yanında bu yöntemin diğer bir olumsuz yanı da

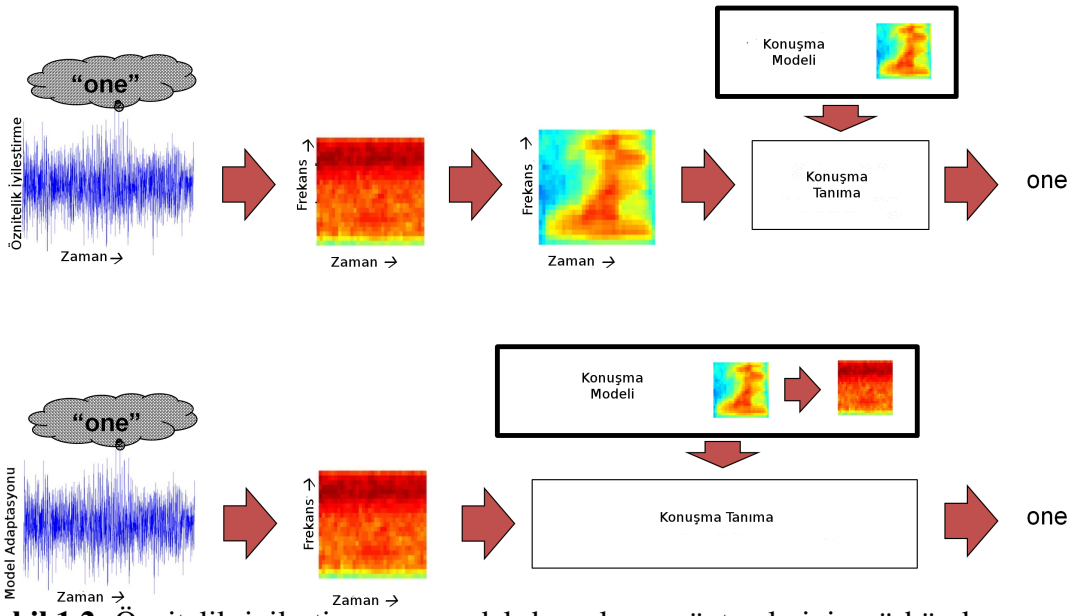


Şekil 1.1: Gürültünün konuşma tanıma işlemine olan etkisi [4].

gürültü içermeyen konuşmalarda yine gürültüsüz konuşmalardan eğitilmiş modellere nazaran daha düşük başarım vermesidir.

Literatürde önerilen diğer gürbüz konuşma tanıma teknikleri akustik model ile gözlem vektörleri arasındaki uyumsuzluğu azaltmak için, akustik modeli veya gözlem vektörlerini değiştirmeye odaklanmıştır. Akustik modeli değiştirme işlemi genel olarak model dengeleme diye tabir edilmektedir. Model dengeleme yöntemlerinde gürültü için bir kestirim yapılmakta ve bu kestirim yardımıyla akustik modeller güncellenmektedir [6, 7]. Bu güncellemedeki amaç gürültünün neden olduğu akustik enerji dağılımındaki değişikliklerin modele yansıtılmasıdır.

Gözlem vektörlerini değiştirerek uyumsuzluğu azaltmayı amaçlayan yöntemler ise literatürde öznitelik iyileştirme yöntemleri adıyla geçmektedir. Bu yöntemlere verilebilecek en basit örnek, temiz akustik gözlem vektörlerine yakınsayabilmek için yapılan ortalama ve varyans normalizasyonudur. Daha ileri yöntemler ise gürültü kestirimi yapmakta ve bu gürültünün gözlem vektörleri üzerindeki etkisini kaldırmaya çalışmaktadır. Şekil 1.2'de bu iki yaklaşımı daha anlaşılır kılmak için bu yöntemlerin tanıma işleminde nasıl kullanıldığı anlatılmıştır.



Şekil 1.2: Öznitelik iyileştirme ve model dengeleme yöntemlerinin gürbüz konuşma tanımada kullanımı [4].

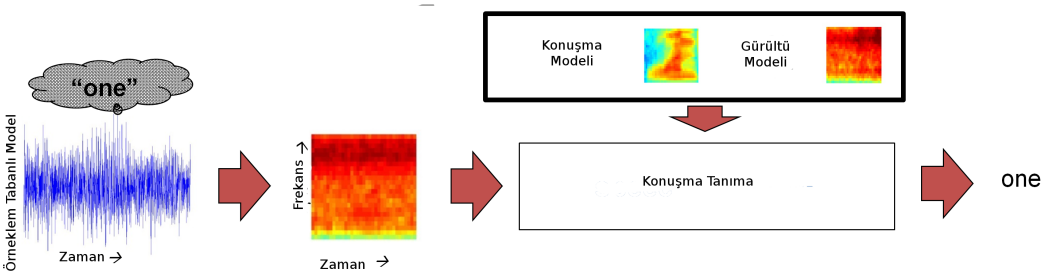
1.3 Örneklem Tabanlı Konuşma Tanıma ve Tez Planı

Örneklem tabanlı yöntemler makine öğrenmesi ve sinyal işleme alanlarındaki popüler araştırma konularındandır. Bu yöntemler ağırlıklı olarak görüntü işleme alanında yüz ve nesne tanıma uygulamaları için kullanılmıştır. Bu yöntemlere örnek olarak [8–10] çalışmaları verilebilir.

Örneklem tabanlı yöntemler ses işleme alanında içerik tabanlı ses sınıflandırma [11,12] ve ses-müzik ayrıştırma gibi uygulamalarda kullanılmıştır [13]. Fakat bu yöntemlerin konuşma tanıma alanında kullanımı ancak son yıllarda mümkün olmuştur. Artan depolama alanı ve işlemci gücüne ek olarak makine öğrenmesi algoritmalarının gelişmesi gibi etkenler, bu yöntemlerin konuşma tanıma alanında başarılı ve verimli bir şekilde kullanılmasını sağlamıştır.

Bu çalışmada gürültülü bir konuşmayı, gürültü ve temiz konuşma örneklemlerinin seyrek doğrusal birleşimi olarak modelleyen bir yöntem olan seyrek sınıflandırma [14] yöntemi gerçeklenmiştir. Seyrek sınıflandırmanın çalışma prensibi Şekil 1.3'te gösterilmiştir. Seyrek sınıflandırma yönteminin iki avantajı vardır: Bunlardan ilki yöntem gürültülü bir konuşmayı oluşturan gürültü ve konuşma örneklemlerinin seyrek doğrusal birleşimini bulduğundan, gürltülü bir konuşmadaki konuşma ve gürültü bileşenleri kestirilebilir. İkinci ve daha önemli bir avantajı ise kullanılan örneklemlerin

karşı düşen ses birimleri ile etiketlenmesi durumunda tanıma işlemi SMM'lerde kullanılan klasik kod çözme işlemi şeklinde yapılabilmektedir. Bu yaklaşımda her bir örneklemin ağırlığı, klasik SMM'lerde kullanılan durum olabilirliğine karşı düşmektedir.



Şekil 1.3: Seyrek sınıflandırma yönteminin gürbüz konuşma tanımada kullanımı [4].

Çalışmanın planı şu şekildedir: İkinci bölümde SMM hakkında teorik bilgi verilecektir. Üçüncü bölümde ise SMM ile bir akustik model eğitimi ve konuşma tanıma işleminin SMM çatısında nasıl yapılacağı anlatılacaktır. Dördüncü bölümde seyrek sınıflandırma yöntemi ve yapılan iyileştirmeler, beşinci bölümde kullanılan veri seti ve yapılan deneyler anlatılacaktır. Altıncı ve son bölümde ise bu çalışmanın sonuçları yorumlanacak ve elde edilen kazanımlar anlatılacaktır.

2. SAKLI MARKOV MODELİ

Konuşma tanıma sistemleri, bir dilde bulunan sesleri temsil eden istatistiksel modeller yardımıyla tanıma işlemini gerçekleştirmektedir. Buna ek olarak konuşma sinyali insan sesinin frekans aralığını örten spektral vektörler dizisi olarak ifade edilebilmektedir. Bu iki bilgi birleştirildiğinde ortaya şu çıkmaktadır: Konuşma tanıma sistemleri en temel anlamda aslında bir dizi sınıflandırması yapmaktadır. Bir dizi sınıflandırıcı, ilgili dizideki her bir elemana bir etiket veya bir sınıf atar. Bu şekilde düşünüldüğünde, herhangi bir konuşmaya ait spektral vektör dizisindeki her bir vektör bir ses ile etiketlenirse, o dildeki sesler istatistiksel olarak modellenebilir. Bu bilgiler ışığında, SMM'ler konuşma sinyalinin yapısını modellemek ve ilgili istatistiksel modelleri elde etmek için iyi bir çatı oluşturmaktadır.

Çalışmanın bu bölümü SMM'ler hakkında temel bilgi vermektedir. Öncelikle Markov zinciri hakkında bilgi verilecektir. Daha sonra SMM ve SMM'nin üç temel problemi irdelenecektir.

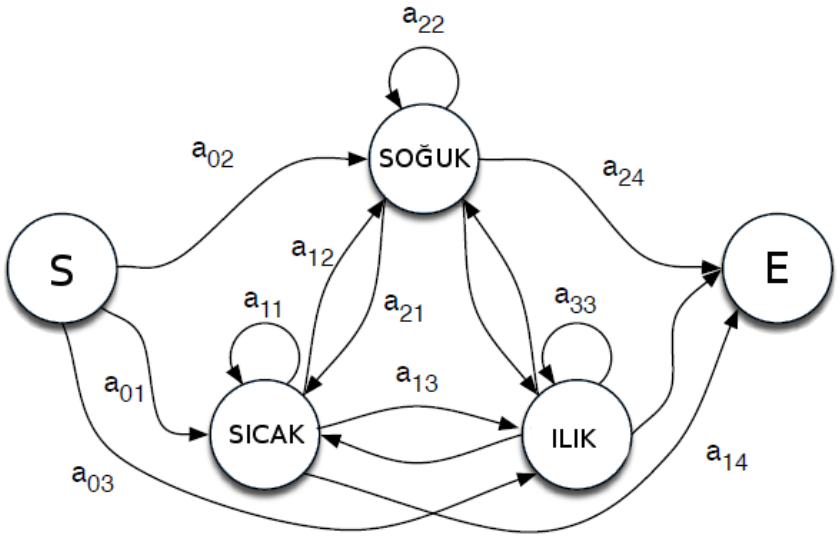
2.1 Markov Zinciri

SMM'leri tam olarak tanımlayabilmek için öncelikle Markov zincirini tanımlamak gerekmektedir. Bir Markov zinciri Çizelge 2.1'deki bileşenler yardımıyla tanımlanabilir. Şekil 2.1'de hava durumu dizilerine bir olasılık atamak için kullanılabilecek örnek bir Markov zinciri gösterilmiştir. Durum alfabesi SICAK, SOĞUK ve ILIK durumlarından oluşmaktadır. Bu modeli kullanarak herhangi bir hava durumu dizisine bir olasılık atanabilir. Şekil 2.1'deki düğümler durumları, ayrıtlar durum geçişlerini, ayrıtların üzerindeki terimler ise geçiş olasılıklarını göstermektedir.

Bir Markov zinciri yardımıyla durum dizilerine olasılık atarken önemli bir varsayım yapılır. Birinci derece Markov zincirindeki bir durumun olasılığı sadece bir önceki duruma bağlıdır. Bu varsayım eşitlik (2.1) ile ifade edilebilir. Ayrıca eşitlik (2.2)'de belirtildiği gibi her bir durumdan ayrılan ayrıtların olasılıkları toplamı 1'e eşit olmalıdır.

Çizelge 2.1: Markov zinciri bileşenleri.

$Q = q_1 q_2 ... q_N$	N adet durum
$A = a_{01} a_{02} ... a_{n1} ... a_{nn}$	A geçiş olasılık matrisi olmak üzere her a_{ij}, i durumunda j durumuna geçişi temsil etmektedir ve $\sum_{j=1}^{N} a_{ij} = 1 \; \forall$i
q_0, q_F	herhangi bir gözleme dayanmayan özel başlangıç ve bitiş durumları

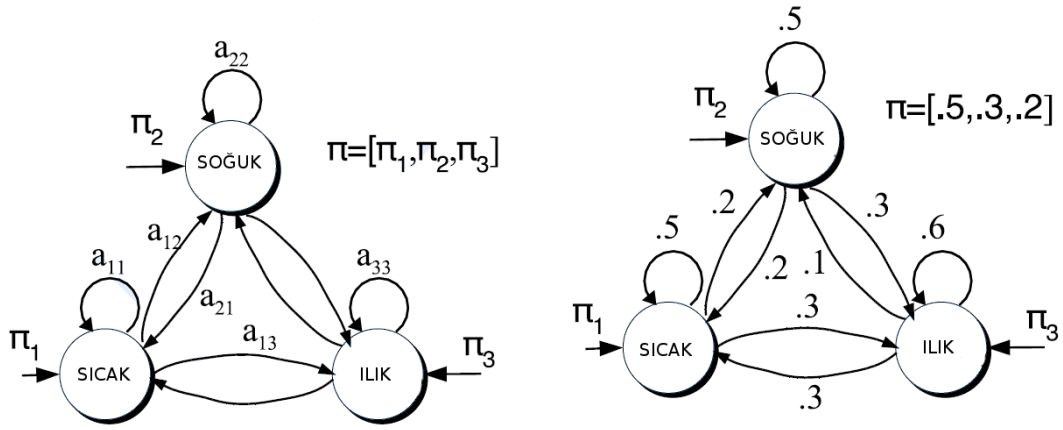


Şekil 2.1: Günlük hava durumu dizilimi için örnek SMM.

Uygun bir gösterim ile Markov zincirindeki başlangıç ve bitiş durumlarına gerek kalmayabilir. Bu gösterim Şekil 2.2'de verilmiştir. Bu gösterime göre ayrıca başlangıç ve bitiş durumları belirlemek yerine başlangıç durumları üzerinden bir dağılım belirlenir.

$$P(q_i|q_1 ... q_{i-1}) = P(q_i|q_{i-1}) \tag{2.1}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i \tag{2.2}$$



Şekil 2.2: Markov zincirinde başlangıç durumları için olasılık dağılımı.

2.2 Saklı Markov Modeli

Markov zinciri gözlemleyebildiğimiz olaylar dizisine bir olasılık atamak için kullanışlıdır. Ancak ilgilenilen bir problemde durumlar veya olaylar direkt olarak gözlemlenemeyebilir. Örneğin konuşma tanıma probleminde spektral vektörler gözlemlenir ve bu gözlem vektörünün hangi kelimeye veya ses birimine ait olduğu çıkarılmaya çalışılır. Saklı Markov Modeli bu amaç için uygun bir modeldir.

Bir SMM Çizelge 2.2'de verilen bileşenlerden oluşur. Eşitlik (2.1) ile verilen Markov varsayımı SMM'ler için de geçerlidir. Ayrıca gözlem o_i sadece üretildiği durum olan q_i durumuna bağlıdır. Bu özellik eşitlik (2.3) ile verilmiştir.

SMM'yi daha iyi anlamak için basit bir örnek düşünülebilir: Elimizde tam sayılardan oluşan bir dizi olsun ve bu dizideki her bir eleman bir kişinin ilgili günde yediği dondurma sayısı olsun. Bu diziden yola çıkarak bu günler için karşı düşen hava durumu dizisini bulmaya çalışalım. Hava durumu alfabesi ise sadece "SICAK" ve "SOĞUK" durumlarından oluşsun [15].

$$P(o_i|q_1...q_i,...,q_T,o_1,...,o_i,...,o_T) = P(o_i|q_i) \tag{2.3}$$

İlgilenilen örnekte bir günde yenen dondurma sayılarına karşı düşen gözlemler dikkate alınmaktadır. Bu yüzden ayrık gözlem olasılık dağılımı kullanılmıştır.

Çizelge 2.2: SMM bileşenleri.

$Q = q_1 q_2 ... q_N$	N adet durum
$A = a_{11} a_{12} ... a_{n1} ... a_{nn}$	A geçiş olasılık matrisi olmak üzere her a_{ij}, i durumunda j durumuna geçişi temsil etmektedir ve $\sum_{j=1}^{N} a_{ij} = 1 \ \forall$i
$O = o_1 o_2 ... o_T$	T adet gözlemden oluşan dizi
$B = b_i(o_t)$	gözlem olabilirlikleri: herbiri i durumundan o_t gözleminin üretilme olasılığını ifade eder
q_0, q_F	herhangi bir gözleme dayanmayan özel başlangıç ve bitiş durumları

Şeki 2.3'te belirtilen problem için örnek bir SMM gösterilmiştir. Bu modelde sıcak ve soğuk havaya karşı düşen durumlar sırasıyla H ve C'dir. Bu şekilde başlangıç durumu S harfiyle gösterilmiştir. Bitiş durumuna geçiş olasılıkları H ve C durumlarına dağıtılmış böylece bu durumlardan birinin bitiş durumu olması sağlanmıştır. Gözlemler 1,2,3 alfabesinden çekilen sayılardır ve bir günde yenen dondurma sayısıdır. Böyle bir SMM tam bağlı SMM olarak adlandırılır. Çünkü herhangi iki durum arasındaki geçiş olasılığı sıfır değildir. Ancak bazı SMM'lerde bazı durumlar arası geçiş olmayabilir. Bunlara örnek olarak Bakis SMM'leri verilebilir. Bakis SMM'leri durumların sadece kendine dönmesine ve bir sonraki duruma geçemsine izin veren SMM'lerdir. Şekil 2.4'te böyle bir SMM gösterilmiştir. Bu yapıdaki SMM'ler konuşma tanımada kullanılmaktadır.

Bu bölümün devamında SMM'lerin üç temel problemi üzerinde durulacaktır. Bu problemler [16] çalışmasında konuşma tanıma için belirlenmiş ve çözüm olarak sunulan algoritmalar incelenmiştir. Problemler ve tanımları aşağıda sıralanmıştır.

- Problem 1: Olabilirlik Hesabı

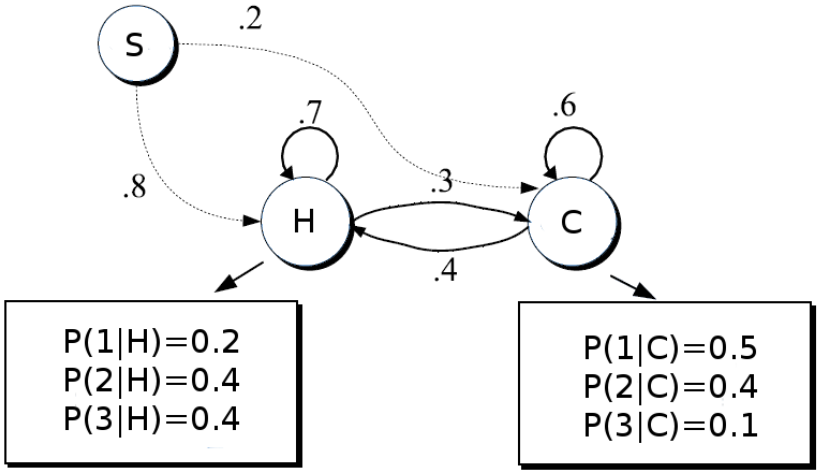
  Tanım: Bir SMM $\lambda = (A, B)$ şeklinde tanımlansın. Bu SMM için O gözlem dizisinin olabilirliği $P(O|\lambda)$ nedir?

- Problem 2: Kod Çözme

  Tanım: Bir SMM $\lambda = (A, B)$ şeklinde tanımlansın. Bu SMM için gözlem dizisi O verildiğinde olası en iyi saklı durum dizisi Q nedir?

- Problem 3: Öğrenme

  Tanım: Bir SMM'e ait gözlem dizisi O ve bu SMM'e ait saklı durum alfabesi verilsin. SMM parametreleri A ve B nedir?

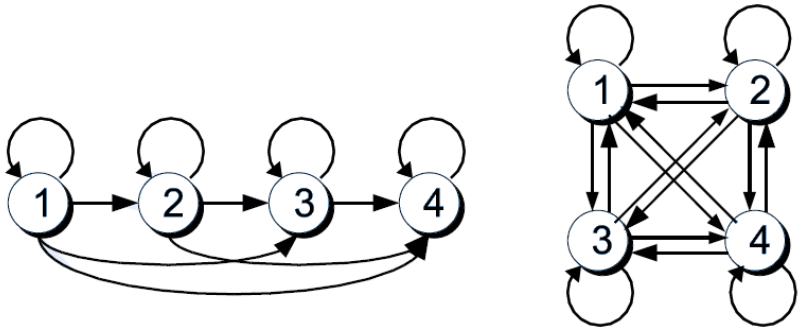


Şekil 2.3: Günlük yenilen dondurma sayısı (gözlemler) ile hava durumunu (saklı durumlar) ilişkilendiren SMM.

2.2.1 Olabilirlik hesabı

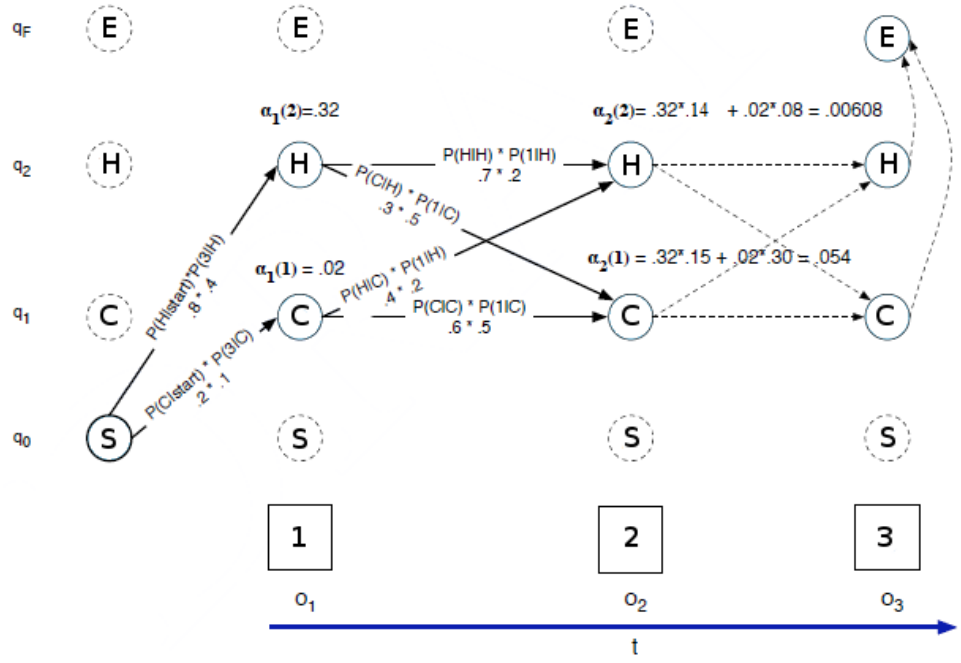
Bu bölümde bir SMM için verilen gözlem dizisinin olabilirliğini hesaplayan algoritma üzerinde durulacaktır. Olabilirlik hesabına örnek olarak Şekil 2.3'teki SMM için 3 1 3 gözlem dizisinin gözlenme olasılığını bulunmak istensin.

Markov zinciri için gözlemler ile saklı olaylar aynı şeye tekabül ettiğinden 3 1 3 dizisinin olabilirliği basitçe 3 1 3 diye etiketlenmiş durum dizisi takip edilerek ve ayrıtlardaki olasılıklar çarpılarak hesaplanır. Ancak bir SMM için problem daha zordur. Çünkü 3 1 3 dizisinin olabilirliği saklı olaylar dizisi bilinmeden hesaplanacaktır.



Şekil 2.4: Bakis SMM (solda) ve tam bağlı SMM (sağda).

Bu problemin çözümü için ileri-yön algoritması kullanılmaktadır. İleri-yön algoritması muhtemel tüm saklı olay dizilimleri için, bir gözlem dizisinin toplam üretilme olasılığını hesaplar. İleri yön algoritmasının kafes üzerindeki akışı ve her bir düğüme ait olasılıkların hesabı Şekil 2.5'te verilmiştir. Her bir $\alpha_t(j)$ kafes üzerindeki her bir yolun o düğümden geçme olasılığının toplamıdır. Bu anlatılanlar ışığında ileri-yön algoritması aşağıdaki gibi üç adımda tanımlanabilir.



Şekil 2.5: İleri-yön algoritmasının kafes üzerindeki akışı.

- İlklendirme

$$\alpha_1(j) = a_{0j} b_j(o_1) \quad 1 \leq j \leq N \tag{2.4}$$

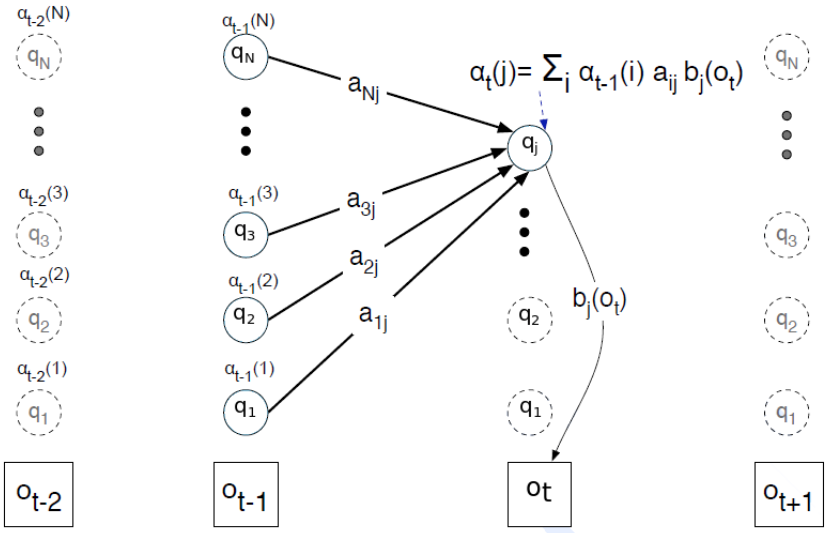
- Yineleme

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, \quad 1 \leq t \leq T \tag{2.5}$$

- Sonlandırma

$$P(O|\lambda) = \alpha_T(q_F) = \sum_{i=1}^{N} \alpha_T(i) a_{iF} \tag{2.6}$$

Şekil 2.6'da bir düğüme ait $\alpha_t(j)$ değerinin hesabı daha açık bir şekilde gösterilmiştir. Buna göre bir düğüme ait $\alpha_t(j)$ değeri şu şekilde hesaplanmaktadır: Kafeste bulunan $\alpha_{t-1}(j)$ değerleri geçiş olasılıkları a_{ij} ve gözlem olasılıklarının $b_j(o_t)$ çarpımı ile ağırlıklandırılarak toplanır. SMM'lerin çoğu uygulamasında geçiş olasılıklarının çoğu sıfır olduğundan, bir önceki durumların hepsi o anki durumun olabilirliğine katkı yapmamaktadır.

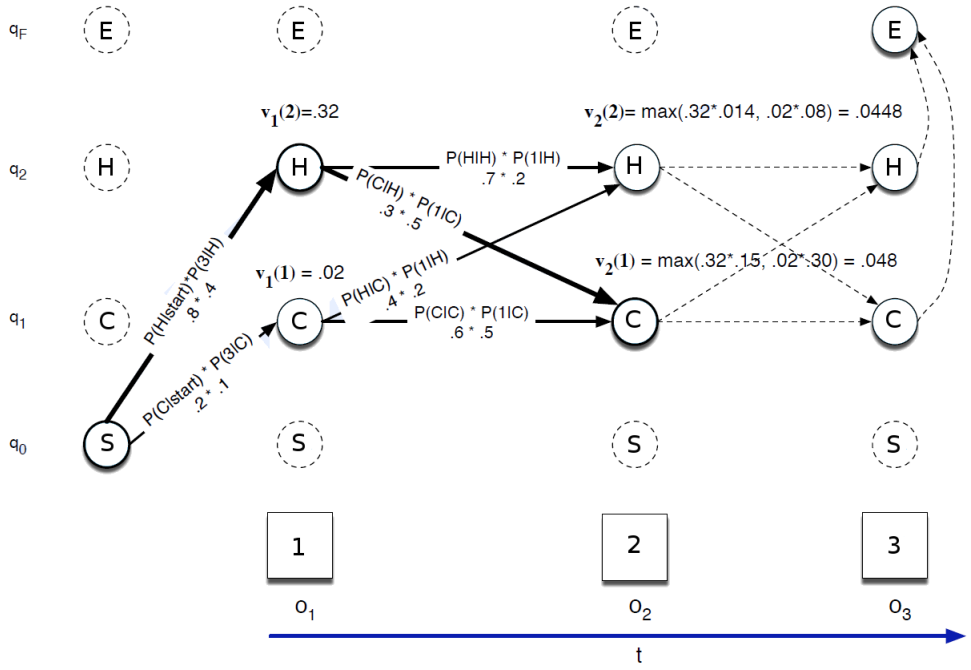


Şekil 2.6: İleri-yön algoritmasında kafesteki herhangi bir düğümün olasılığı.

2.2.2 Kod çözme

Saklı değişkenler içeren herhangi bir model göz önüne alındığında, bir gözlem dizisinin hangi saklı durum dizisi tarafından üretildiğini bulma işlemine kod çözme denir. Bir önceki bölümde verilen örnek düşünülecek olursa, kod çözücünün görevi 3 1 3 gözlem dizisi verildiğinde bu diziyi üretebilecek en iyi durum dizisini bulmaktır.

SMM'ler için kullanılan en yaygın kod çözme algoritması Viterbi algoritmasıdır. Şekil 2.7'de Viterbi algortitmasının akışı örnek bir kafes üzerinde gösterilmiştir. Gözlemler sırasıyla işlenirken kafesin düğümlerine ait olasılıklar da algoritma tarafından hesaplanmaktadır. Her bir $v_t(j)$, t adet gözlemden sonra en olası durum yolunu takip ederek j. durumda olma olasılığıdır. Her bir $v_t(j)$ ileri-yön algoritmasında olduğu gibi özyinelemeli olarak hesaplanır. Bu değerlerin hesabında ileri-yön algoritmasına göre tek bir farklılık vardır: t anında $t-1$ anına ait tüm durumların olabilirlikleri aynı şekilde hesaplanır ancak bu olabilirlikler toplanmaz, bunun yerine maksimum olabilirlik alınır.



Şekil 2.7: Viterbi algoritmasının kafes üzerindeki akışı.

Ayrıca Viterbi algoritmasında her bir düğüm için bir geri işaretçi tutulur. Çünkü Viterbi algoritması ileri-yön algoritması gibi bir olabilirlik değil en olası durum dizisi döndürmelidir. Bunun için her bir düğüme gelen en yüksek olabilirlikli yolun hangi düğümden geldiğinin kaydı tutulur. Kafesteki bütün düğümler için $v_t(j)$ değerleri hesaplandığında ise kafesin sonundan başına doğru geri işaretçiler takip edilerek en olası durum dizisi bulunmuş olur. Geri işaretçiler yardımıyla en olası durum dizisinin bulunması Şekil 2.8'de gösterilmiştir. Viterbi algoritması aşağıdaki gibi üç adımda tanımlanabilir:

- İlklendirme

$$v_1(j) = a_{0j} b_j(o_1) \quad 1 \leq j \leq N \tag{2.7}$$

$$b_{t_1}(j) = 0 \tag{2.8}$$

- Yineleme

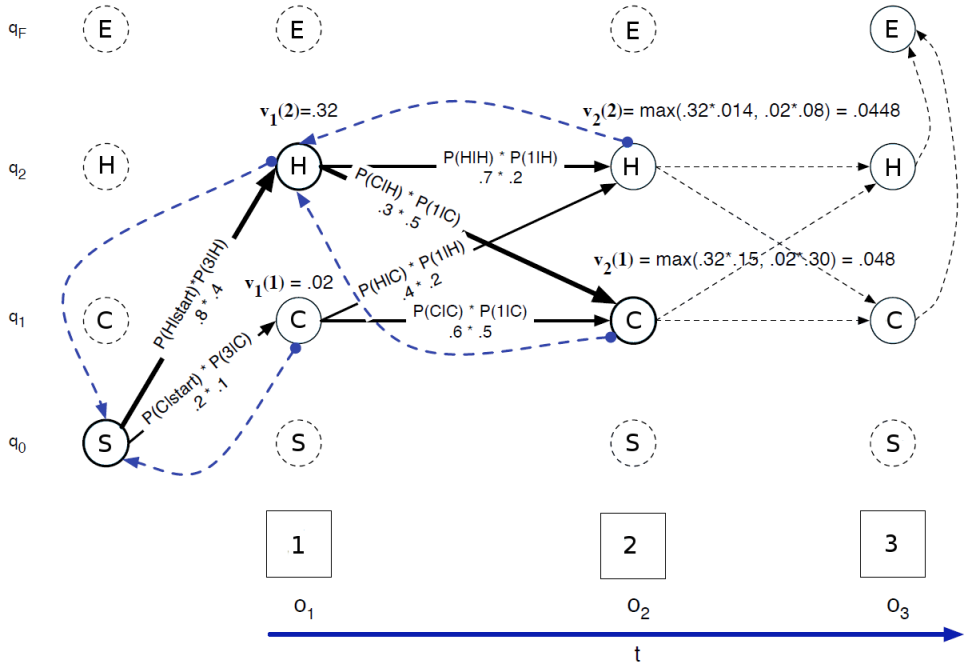
$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, \quad 1 \leq t \leq T \tag{2.9}$$

$$b_{t_t}(j) = \operatorname*{argmax}_{i=1}^{N} v_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, \quad 1 \leq t \leq T \tag{2.10}$$

- Sonlandırma

$$\text{En iyi skor:} \qquad P_* = v_t(q_F) = \max_{i=1}^{N} v_T(i)a_{iF} \qquad \textbf{(2.11)}$$

$$\text{En iyi durum dizisinin başlangıcı:} \quad q_{T^*} = b_{(}t_T)(q_F) = \operatorname*{argmax}_{i=1}^{N} v_T(i)a_{iF} \quad \textbf{(2.12)}$$



Şekil 2.8: Geri işaretçiler yardmıyla en olası durum dizisinin belirlenmesi.

2.2.3 Öğrenme

Bu bölümde SMM parametrelerinin öğrenilmesi problemi ele alınacaktır. Öğrenilecek parametreler A ve B matrisleridir. Bu problemi çözecek algoritmanın girdileri bir gözlem dizisi O ve saklı durum alfabesi Q'dur. Önceki bölümlerde verilen dondurma örneği düşünüldüğünde $O = 1,3,2,...$ gibi bir gözlem dizisi ve saklı durumlar olan H ve C algoritmanın girdileri olacaktır.

Bu problem için önerilen standart algoritma Baum-Welch algoritmasıdır. Bu algoritma sayesinde hem geçiş olasılıklarını içeren A matrisi hem de gözlem olasılıklarını içeren B matrisi öğrenilmektedir. Baum-Welch algoritmasının anlaşılabilmesi için öncelikle geri-yön algoritması açıklanmalıdır. Geri yön algoritması ile Şekil 2.9'dan da görülebileceği gibi kafes üzerindeki her bir düğüm için $t+1$ zamanından T'ye kadar olan gözlem dizisi verildiğinde t zamanında i. durumda olma olasılığı hesaplanır.

Geri-yön algoritması ileri-yön algoritmasına benzer şekilde aşağıdaki gibi üç adımda tanımlanabilir:

- İlklendirme

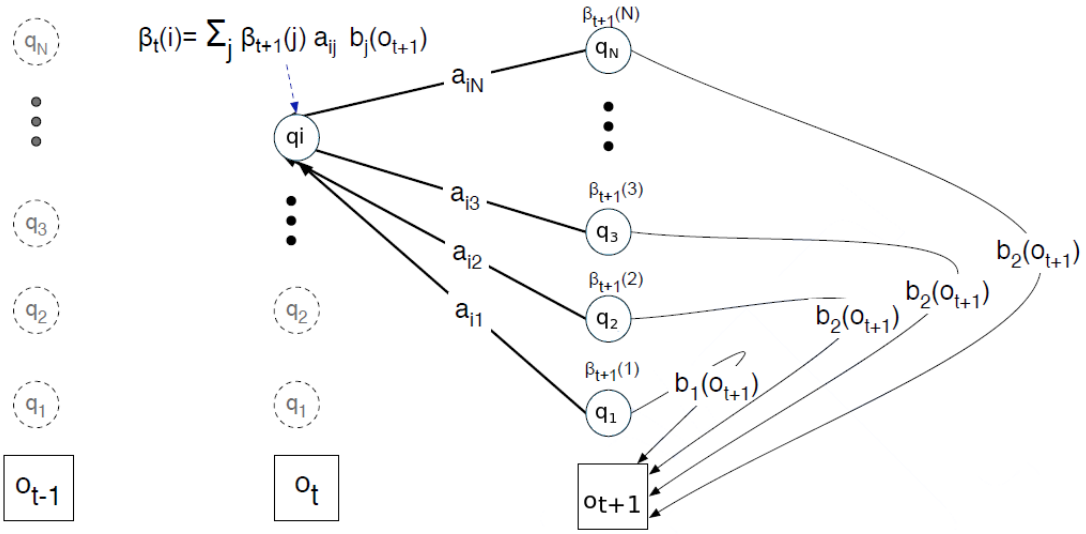
$$\beta_T(i) = a_{iF}, \quad 1 \leq i \leq N \tag{2.13}$$

- Yineleme

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad 1 \leq i \leq N, \quad 1 \leq t \leq T \tag{2.14}$$

- Sonlandırma

$$P(O|\lambda) = \alpha_T(q_F) = \beta_1(0) = \sum_{j=1}^{N} a_{0j} b_j(o_1) \beta_1(j) \tag{2.15}$$

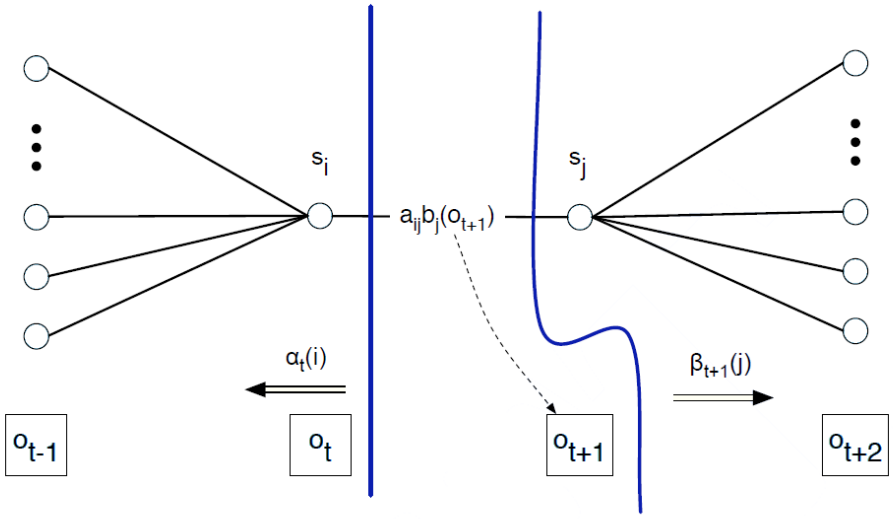


Şekil 2.9: Geri yön algoritmasında kafesteki herhangi bir düğümün olaısılığı.

Bu aşamada ileri ve geri yön algoritmalarının geçiş olasılıkları ve gözlem olasılıklarını öğrenmede nasıl kullanılacağı anlaşılabilir. Bir geçiş olasılığı $\hat{a}_{ij}$ eşitlik (2.16)'da belirtildiği gibi hesaplanabilir.

$$\hat{a}_{ij} = \frac{i \text{ durumundan } j \text{ durumuna geçiş sayısı}}{i \text{ durumundan herhangi bir duruma geçiş sayısı}} \tag{2.16}$$

Eşitlik (2.16)'nın payını hesaplamak için ξ_t olasılığını tanımlamak gerekir. Bu olasılık t anında i durumunda ve $t+1$ anında j durumunda bulunma olasılığıdır ve eşitlik (2.17)'de verilen formül ile hesaplanır. Ayrıca payın hesabı için Şekil 2.10 incelenebilir. Eşitlik (2.17)'daki payda eşitlik (2.18)'den de görülebileceği gibi bütün gözlemler için ileri-yön algoritması veya geri-yön algoritması kullanarak hesaplanan büyüklüktür.



Şekil 2.10: t anında i durumunda ve $t+1$ anında j durumunda bulunma olasılığını hesaplarken kullanılan olasılıklar.

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(N)} \tag{2.17}$$

$$P(O|\lambda) = \alpha_T(N) = \beta_T(1) = \sum_{j=1}^{N} \alpha_t(j)\beta_t(j) \tag{2.18}$$

Böylece i durumundan j durumuna geçiş sayısı için beklenen değer bütün ξ'lerin t üzerinden toplamı ile bulunabilir. Eşitlik (2.16)'nın paydasını hesaplamak için ise t üzerinden alınan toplama ek olarak bir de j durumları üzerinden toplam alınır. Böylece i durumundan çıkmış geçişlerin sayısı için beklenen değer bulunur ve $\hat{a}_{ij}$ değeri eşitlik (2.19)'daki gibi hesaplanır.

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{N} \xi_t(i,j)} \tag{2.19}$$

Geçiş olasılıklarının yanı sıra gözlem olasılıklarına yakınsamak için de bir formüle ihtiyaç vardır. Temel olarak bulunmak istenen olasılık j durumunda bulunup v_k gözleminin üretilme olasılığıdır. Bu olasılık eşitlik (2.20)'deki gibi ifade edilebilir. Bu olasılığı hesaplayabilmek için öncelikle t anında j. durumunda bulunma olasılığı tanımlanmalıdır. Bu olasılık eşitlik (2.21)'deki gibi hesaplanabilir. Herhangi bir $\gamma_t(j)$ olasılığının hesabı için Şekil 2.11 incelenebilir.

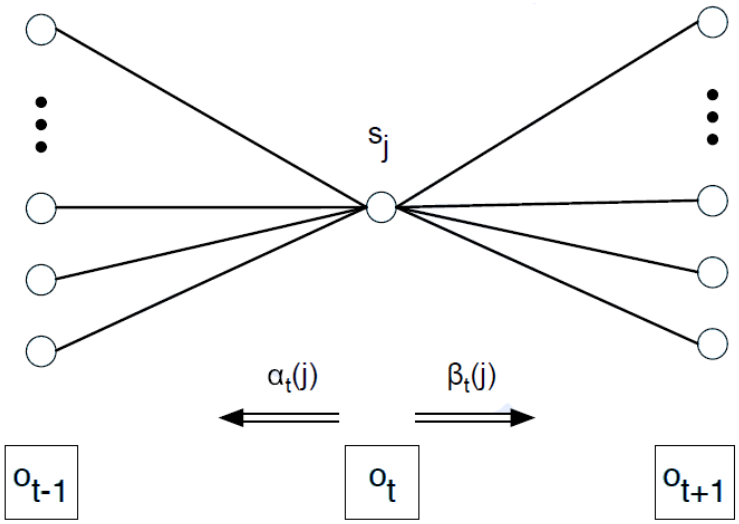
$$\hat{b}_j(v_k) = \frac{j \text{ durumunda bulunup } v_k \text{ gözleminin üretilme sayısı}}{j \text{ durumunda bulunma sayısı}} \tag{2.20}$$

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)} \tag{2.21}$$

Bu noktadan sonra $b_j(v_k)$ olasılıkları hesaplanabilir. Pay için bütün t anlarında v_k sembolünün gözlenmesi durumuna karşılık gelen $\gamma_t(j)$ olasılıkları toplanacaktır. Payda için ise $\gamma_t(j)$ olasılıkları bütün t anları için toplanacaktır. Böylece sonuç j durumunda bulunulan anların kaçında v_k sembolünün gözlemlendiğini belirten oran olacaktır. $\hat{b}_j(v_k)$ hesabı eşitlik (2.22)'de verilmiştir.

Şu ana kadar anlatılanlar ışığında Baum-Welch algoritmasının adımları belirlenmiş oldu. Bu algoritma aslına Beklenti Maksimizasyon algoritmasının SMM yapısına uyarlanmış halidir. Bilindiği gibi Beklenti Maksimizasyon algoritmasının temel adımları beklenti bulma (E-adımı) ve maksimizasyon (M-adımı) adımlarıdır. Algoritmanın her bir yinelemesi bu iki adımdan oluşur. E-adımında gözlenen verilerin parametrelerine ait kestirimler kullanılarak bilinmeyen veri ile ilgili en iyi sonsal olasılıklar tahmin edilir. M-adımında ise tahmin edilen sonsal olasılıklar yardımıyla, maksimum olabilirlik hesaplanarak parametrelerin yeni kestirimleri elde edilir. Aslında $\xi_t(i,j)$ ve $\gamma_t(j)$ hesabı E-adımına, $b_j(v_k)$ ve $\hat{a}_{ij}$ hesabı ise M-adımına karşılık düşmektedir. Baum-Welch algoritması Şekil 2.12'de verilmiştir.

$$\hat{b}_j(v_k) = \frac{\sum_{t=1:O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{2.22}$$



Şekil 2.11: t anında j durumunda bulunma olasılığını hesaplarken kullanılan olasılıklar.

A ve B'yi ilklendir.

E-adımı:

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)} \quad \forall\, t \text{ ve } j$$

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij} b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(N)} \quad \forall\, t,\ i,\ \text{ve}\ \ j$$

M-adımı:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{N} \xi_t(i,j)}$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1:O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

A ve B'yi döndür.

Şekil 2.12: Baum-Welch algortiması.

Gelinen noktada, SMM'nin üç temel problemine ilişkin kullanılan çözümler irdelenmiş oldu. Ancak SMM yapısının konuşma tanımada kullanımına bu bölümde yer verilmedi. Çalışmanın bir sonraki bölümünde SMM'lerin konuşma tanımada nasıl kullanıldığı örnek bir uygulama üzerinden anlatılacaktır.

2.3 Özet

Bu bölümde Markov zinciri ve SMM yapısı anlatılmıştır. Markov zincirinde gözlemlenen olaylar durumlara karşı düşerken, SMM'de gözlemler saklı durumlar tarafından üretilmektedir.

SMM'nin yapısı gereği ortaya çıkan 3 temel problem bu bölümde ele alınmıştır. Bu problemlere çözüm olarak kullanılan algoritmalar incelenmiş basit bir örnek üzerindeki kullanımları aktarılmıştır.

3. SMM TABANLI KONUŞMA TANIMA

Bu bölümde SMM tabanlı bir konuşma tanıma sistemi hakkında ayrıntılı bilgi verilecektir. Yoğun olarak SMM'nin konuşma tanımaya uygulanması üzerinde durulacak olsa da otomatik konuşma tanıma sistemlerinin diğer bileşenlerinden de bahsedilecektir. Sırasıyla şu başlıklar üzerinde durulacaktır: Konuşma tanıma sistemi mimarisi, SMM'nin konuşma tanımaya uygulanması, öznitelik çıkarımı, akustik olabilirlik hesabı, sözlük ve dil modeli, kod çözme işlemi, akustik model eğitimi ve değerlendirme metriği.

3.1 Konuşma Tanıma Sistemi Mimarisi

Bir ses işaretini 10ms'lik parçalara ayırdığımızı ve her bir parçayı o parçanın frekans veya enerji değerleri ile temsil ettiğimizi düşünelim. Sonuç olarak elimizde bir gözlem dizisi oluşur. Bu dizideki elemanların indisleri de art arda gelen zaman aralıklarına karşı düşmektedir. Konuşma tanıma sistemlerinin amacı, bahsedilen bu akustik gözlem dizisine karşı düşen kelime dizisini bulmaktır. Gözlem dizisi eşitlik (3.1) ile, karşı düşen kelime dizisi ise eşitlik (3.2) ile gösterilecek olursa konuşma tanıma sistemi olasılıksal olarak eşitlik (3.3) ile verilen işlemi gerçeklemektedir.

$$O = o_1, o_2, o_3, ..., o_t \tag{3.1}$$

$$W = w_1, w_2, w_3, ..., w_n \tag{3.2}$$

$$\hat{W} = \underset{W \in \mathscr{L}}{\operatorname{argmax}} P(W|O) \tag{3.3}$$

Bayes kuralını kullanarak $P(W|O)$ olasılığı, eşitlik (3.4)'te olduğu gibi yazılabilir. Böylece hesabı daha kolay olasılıklar ile çalışma imkanı elde edilir. Eşitlik (3.4)'ün paydasındaki olasılık ihmal edilebilir. Çünkü olası her bir cümle için $P(O)$ aynıdır ve burada amaç maksimizasyon olduğundan eşitlik (3.5)'teki gibi $P(O)$ ihmal edilebilir.

$$\hat{W} = \underset{W \in \mathscr{L}}{\operatorname{argmax}} \frac{P(O|W)P(W)}{P(O)} \tag{3.4}$$

Özetlemek gerekirse herhangi bir cümle için O gözlem dizisi verildiğinde en olası kelime dizisi W, eşitlik (3.5)'teki iki olasılığın çarpımını en büyük yapan kelime dizisidir. Bir konuşma tanıma motorunun da iki temel bileşeni aslında bu iki olasılığın hesabına karşı düşmektedir. Bunlardan $P(W)$ önsel olasılığı N-gram dil modelleri yardımıyla hesaplanırken, $P(O|W)$ gözlem olabilirliği akustik model yardımıyla hesaplanır.

$$\hat{W} = \underset{W \in \mathscr{L}}{\operatorname{argmax}} \frac{P(O|W)P(W)}{P(O)} \propto \underset{W \in \mathscr{L}}{\operatorname{argmax}} P(O|W)P(W) \tag{3.5}$$

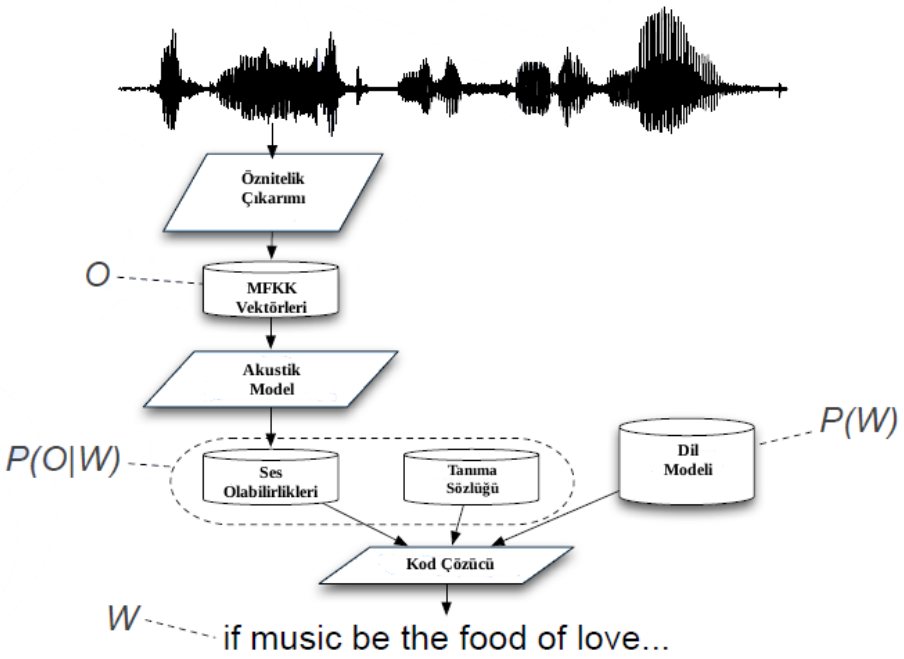
Şekil 3.1 SMM tabanlı konuşma tanıma işleminin tek bir konuşma için nasıl yapıldığını ve konuşma tanıma sisteminin bileşenlerini göstermektedir. Şekil konuşma tanımayı üç aşamaya bölmüştür. Bunlardan ilki olan öznitelik çıkarma işlemi sırasında işaret 10ms'lik parçalara bölünür ve her bir parça spektral özniteliklere dönüştürülür. Böylece her bir zaman penceresi, frekans ve enerji bilgisi taşıyan 39 boyutlu vektörler ile ifade edilmiş olur.

İkinci aşama olan ses tanıma işlemi ile gözlemlenen vektörlere ait akustik olabilirlik hesaplanır. Örneğin ses veya ses altı bir birime karşı düşen herhangi bir SMM durumu q için Gauss Karışım Modelleri sınıflandırıcı olarak kullanılır. Böylece $p(o|q)$ hesaplanır. Bu aşamanın çıktısı her bir gözlem vektörü için ses veya ses altı parçaların olabilirliklerine karşı düşen vektörlerdir.

Üçüncü aşama olan kod çözme işlemi sırasında ise, akustik modelin çıktıları olan ses olabilirlikleri, kelimelerin telaffuzlarını SMM durumlarının karşı düştüğü ses parçalarının birleşimi cinsinden ifade eden sözlük, ve dil modeli (genellikle N-gram modelleri) girdi olarak kullanılır. Bu aşamanın çıktısı ise en olası kelime dizisidir.

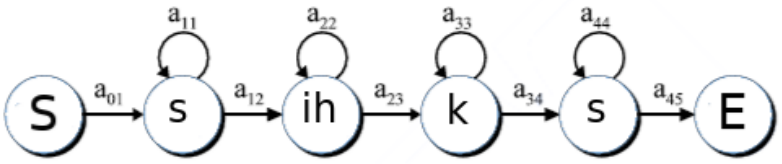
3.2 SMM'nin Konuşma Tanımaya Uygulanması

Konuşma tanıma işlemi için SMM durumları kelimelere, seslere veya ses altı birimlere karşı düşebilir. Gözlemler ise daha önce bahsedildiği gibi ilgili zaman aralığındaki frekans ve enerji bilgisini taşıyan vektörlerdir.



Şekil 3.1: Konuşma tanıma aşamaları.

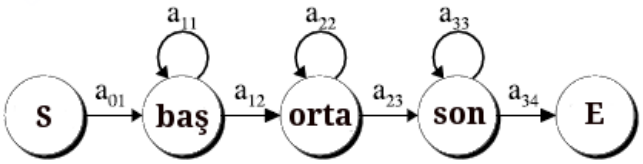
Arka arkaya gelen rakam dizilerini tanıma gibi küçük konuşma tanıma uygulamalarında, her bir saklı durumu bir sese karşı düşen SMM'ler kullanılabilir. Fakat geniş dağarcıklı konuşma tanıma uygulamalarında SMM'lerin saklı durumları ses altı birimlere karşı düşmektedir. Örneğin Şekil 3.2'de her bir saklı durumu bir sese karşı düşen bir SMM verilmiştir. Bu SMM aslında "six" kelimesini ifade etmektedir. Durumlar için kendine dönme veya bir sonraki duruma geçme ihtimali vardır. Bu yapı konuşma sinyalinin doğasından kaynaklanmaktadır ve Bakis SMM'leri ile modellenebilir.



Şekil 3.2: "six" kelimesini modelleyen SMM.

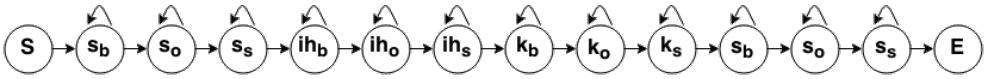
Her bir sesin süresi, sesin yapısına ve konuşmacının konuşma hızına göre değişmektedir. Örneğin Switchboard derleminde 1.3 saniye boyunca süren sesler bulunmaktadır [15]. Her bir gözlem vektörünün 10 ms'yi temsil ettiği düşünülürse aslında bu arka arkaya gelen yaklaşık 130 gözlem vektörünün tek bir sese ait olması

anlamına gelir. Bu bilgi SMM durumlarının kendini tekrar edebilmesinin gerekliliğini açıklar.



Şekil 3.3: Bir sesi modelleyen SMM.

Bir ses boyunca sese ait spektral karakteristik ve enerji miktarı ciddi değişimlere uğrayabilir. Bunun sonucu olarak özellikle geniş dağarcıklı konuşma tanıma uygulamalarında SMM durumları ses altı birimlere karşı düşer. Böyle bir kullanım için Şekil 3.3 incelenebilir. Bu modelde bir ses 5 durumlu bir SMM ile modellenmiştir. Ancak daha önceki örneklerde olduğu gibi başlangıç ve bitiş durumları herhangi bir gözlem üretmemektedir. Şekil 3.4'te olduğu gibi ses yapısını daha ayrıntılı modelleyen bu yapılar uç uca eklenerek kelimelere karşı düşen SMM modelleri elde edilebilir.



Şekil 3.4: Ses modelleyen SMM'lerin uç uca eklenmesi ile oluşmuş kelime modelleyen SMM.

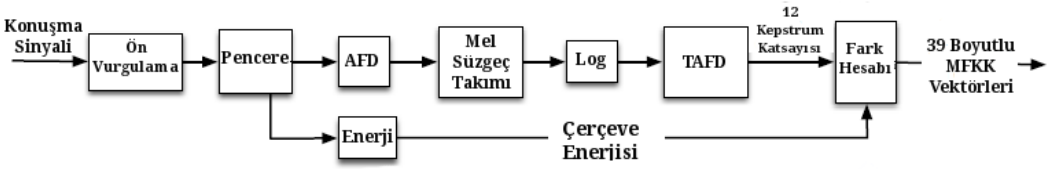
3.3 Öznitelik Vektörlerinin Çıkarımı

İdealde öznitelik vektörlerinin benzer sesleri birbirinden ayırt edebilecek bilgi taşıması, çok fazla eğitim verisine ihtiyaç duymadan akustik modellerin eğitilmesine imkan vermesi ve konuşmacı ve konuşulan ortamdan bağımsız olarak aynı konuşma sinyali için aynı sonuçları verebilmesi beklenmektedir.

Literatürde yukarıda sayılan beklentileri karşılamak üzere birçok öznitelik çıkarım prosedürü önerilmiştir. Bunlardan en çok kullanılanı Mel Frekans Kepstrum Katsayıları(MFKK)'dır. MFKK çıkarımı ile ilgili literatürde birçok çalışma bulunmaktadır. Bu bölümde sadece temel adımlardan bahsedilecektir. Bu adımlar Şekil 3.5'te verilmiştir.

3.3.1 Ön vurgulama

Ünlü harflere ait seslerin frekans spektrumu incelendiğinde, düşük frekanslarda yüksek frekanslara göre daha fazla enerji olduğu görülmüştür [15]. Yüksek frekanslardaki



Şekil 3.5: MFKK çıkarım prosedürü.

enerjiyi arttırmak, seslerin ayrışmasına katkıda bulunmakta ve akustik modeli iyileştirmektedir. Bu işlem birinci dereceden yüksek geçiren bir filtre ile yapılabilir.

3.3.2 Pencereleme

Konuşma işareti durağan bir işaret değildir. Bu yüzden seslerin iyi ayrıştırılması adına, işaretin çok küçük zaman aralıklarında sabit istatistiklere sahip olduğu kabul edilir ve oluşan bu çerçevelerden öznitelik çıkarılır. Literatürde genellikle 25 ms'lik pencereler 10 ms kaydırılarak çerçeveler çıkarılır. En yaygın pencere ise Hamming penceresidir. Bu pencere sayesinde, dikdörtgen bir pencerenin kullanımı durumunda ortaya çıkabilecek süreksizlikler giderilmiş olur.

3.3.3 Ayrık Fourier dönüşümü

Bir sonraki adımda çerçevelerin frekans domenindeki gösterimleri bulunur. Bunun için AFD kullanılır. AFD'yi verimli bir şekilde hesaplamak için çoğu zaman Hızlı Fourier Dönüşümü kullanılır. Çerçevelerin frekans spektrumu, hangi frekans bandında ne kadar enerji biriktiğini hesaplamak için gereklidir.

3.3.4 Mel ölçekli süzgeç takımı ve logaritma alma

Yapılan araştırmalarda, insan kulağının bütün frekans bantlarına karşı aynı duyarlılıkta davranmadığı görülmüştür [15]. Bu durumu, öznitelik vektörü çıkarırken modellemek otomatik konuşma tanıma başarımını arttırmaktadır. Bu modelleme işlemi için Mel ölçeği kullanılır. Hertz ölçeğinden Mel ölçeğine geçilirken 1 kHz'in altı için doğrusal, 1 kHz üstü için logaritmik aralıklar ile karşı düşürme işlemi yapılır. Bu yeni ölçeğe geçmek için işaret bir süzgeç takımından geçirilir. Bu süzgeç takımı 1 kHz'e kadar doğrusal aralıklarla, 1 kHz'den sonra ise logaritmik aralıklar ile yerleştirilen süzgeçlerden oluşmaktadır. Bu şekilde her bir süzgeçten kapsadığı frekans bandına ait enerji elde edilir. Daha sonra bu değerlerin doğal logaritması alınır. Doğal logaritma

alma işleminin temel amacı öznitelik vektörlerinin sesin şiddetinin artması ya da azalması gibi durumlardan olabildiğince az etkilenmesini sağlamaktır.

3.3.5 Kepstrum ve ters ayrık Fourier dönüşümü

Kepstrum aslında bir işaretin AFD'sinin logaritması alındıktan sonra TAFD'sinin alınması işlemidir. Ön vurgulama ve Mel ölçeği gösterimi adımlarını bir kenara bırakırsak aslında kepstrum hesabı için öncelikle bir çerçevenin AFD'si bulunur. Daha sonra frekans spektrumundaki herbir genlik değerinin logaritması alınır. Elde edilen bu yeni işaretin TAFD'si alınır ve elde edilen yeni sinyalin ilk 12 örneği kepstrum katsayıları olarak kullanılır. Böylece aslında her bir çerçeve için 12 adet katsayı bulunmuş olur.

3.3.6 Fark vektörleri ve enerji değerleri

Her bir çerçeve için bulunan 12 katsayıya 13. katsayı olarak çerçeveye ait enerji eklenir. Ayrıca kepstral katsayıların değişimini daha iyi modellemek adına elde edilen bu 13 katsayıya delta ve çift delta değerleri eklenir. Delta değeri eşitlik (3.6)'daki gibi hesaplanabilir. Çift delta değerleri ise kepstral katsayılardaki değişim yerine deltalardaki değişimi ölçer ve delta katsayılarının hesaplandığı şekilde hesaplanabilir.

$$d(t) = \frac{c(t+1) - c(t-1)}{2} \tag{3.6}$$

3.3.7 MFKK çıkarım özeti

Her bir çerçeve için çıkarılan 12 kepstral katsayıya enerji, delta ve çift delta değerleri eklendiğinde 39 boyutlu vektörler elde edilir. Aşağıda elde edilen vektörün bileşenleri verilmiştir.

12 kepstral katsayı

12 delta katsayısı

12 çift delta katsayısı

1 enerji katsayısı

1 delta enerji katsayısı

1 çift delta enerji katsayısı

---

39 boyutlu MFKK vektörü

3.4 Akustik Olabilirlik Hesabı

Bir önceki kısımda bir konuşma işaretinin ötüşen çerçeveleri için MFKK vektörlerinin nasıl çıkarıldığı anlatıldı. Bu kısımda ise herhangi bir SMM durumu için öznitelik vektörlerinin olabilirliklerinin nasıl hesaplandığı anlatılacaktır.

3.4.1 Gauss karışım modelleri

Bu kısımda GKM'lerin akustik olabilirlik hesabında kullanımları anlatılacaktır. Öncelikle bir SMM durumunun tek bir Gauss ile modellendiği varsayılsın. SMM-GKM tabanlı konuşma tanıma sistemlerinde genellikle 39 boyutlu MFKK öznitelikleri kullanılmaktadır. Bu yüzden çok boyutlu Gauss dağılımları kullanılmalıdır. Çok boyutlu bir Gauss dağılımı kullanılarak bir gözlem vektörünün olabilirliği eşitlik (3.7)'deki gibi hesaplanabilir. Bu eşitlikte D gözlem vektörünün boyutu, μ ortalama vektörü, Σ ise kovaryans matrisidir.

$$f(x|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} exp\big(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\big) \tag{3.7}$$

Bu durumda verilen μ_j ve Σ_j istatistiklerine sahip bir SMM durumunun bir o_t gözlem vektörü için verdiği akustik olabilirlik eşitlik (3.8) ile ifade edilebilir. Konuşma tanıma sistemlerinde çoğunlukla kovaryans matrisi Σ yerine bu matrisin köşegenindeki elemanları içeren vektör kullanılır. Bu durumda öznitelik vektörlerindeki her bir boyutun birbirinden bağımsız olduğu varsayılmış olur.

$$b_j(o_t) = \frac{1}{(2\pi)^{\frac{D}{2}}} exp\big(-\frac{1}{2}(o_t-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)\big) \tag{3.8}$$

Kovaryans matrisini bütün olarak kullanmanın iki kötü etkisi vardır. Birincisi tam kovaryans matrisinde D^2 parametre varken köşegen kovaryans matrisi sadece D adet parametre içermektedir ve bu durum hız anlamında konuşma tanıma sistemlerini

kötü etkilemektedir. İkincisi bu kadar ayrıntılı bir modeli eğitebilmek için köşegen kovaryans matrisi kullanan modellere göre çok daha fazla eğitim verisi gerekmektedir.

Konuşma tanımada kullanılan öznitelik vektörleri bazı boyutlarda normal dağılıma sahipken bazı boyutlarda farklı dağılım gösterebilirler. Bu yüzden olabilirlik hesabı için çok boyutlu bir Gauss kullanmak yerine birden fazla çok boyutlu Gauss'un ağırlıklı karışımı kullanılır. Bu modele Gauss Karışım Modeli (GKM) denir. Eşitlik (3.9) GKM için olabilirlik fonksiyonunu göstermektedir. Eşitlik (3.10)'da ise bu modelin verilen bir SMM durumu ve gözlem vektörü için kullanımı gösterilmiştir.

$$f(x|\mu,\Sigma) = \sum_{k=1}^{M} c_k \frac{1}{\sqrt{2\pi|\Sigma_k|}} exp\big((x-\mu_k)^T \Sigma^{-1}(x-\mu_k)\big) \tag{3.9}$$

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} \frac{1}{\sqrt{2\pi|\Sigma_{jm}|}} exp\big((o_t-\mu_{jm})^T \Sigma_{jm}^{-1}(o_t-\mu_{jm})\big) \tag{3.10}$$

3.4.2 Akustik modeli eğitimi ve GKM

GKM bir SMM durumuna ait dağılımı modellemek için kullanışlı yapı sunmaktadır. Ancak eğitimde bu modele ait parametreleri öğrenmek gerekmektedir. Baum-Welch algoritması yardımıyla bir gözlemin tam olarak hangi SMM durumunu ile ilişkili olduğu bilinmeden de o SMM durumuna ait GKM parametreleri hesaplanabilir. Çünkü ikinci bölümden hatırlanabileceği gibi Baum-Welch algoritması E-adımı sırasında herhangi bir t zamanında j durumunda bulunma olabilirliğini sağlamaktadır. Bu olabilirlikler GKM için uygun hale getirilirse Baum-Welch algoritması ile GKM içeren SMM'ler eğitilebilir. Bunun için öncelikle $\xi_{tm}(j)$ tanımlansın. Bu olasılık, j durumunu modelleyen GKM'nin m. karışımının o_t gözlemini üretmiş olma olasılığıdır ve eşitlik (3.11)'deki gibi tanımlanır. Bu olasılık kullanılarak ilgili karışım için ortalama, karışım ağırlığı ve varyans sırasıyla eşitlik (3.12),(3.13),(3.14)'teki gibi hesaplanabilir. Bu şekilde Baum-Welch aklgoritması SMM ve GKM'nin birlikte kullanıldığı yapıyı eğitmek için kullanılabilir şekle gelmiş olur.

$$\xi_{tm}(j) = \frac{\sum_{i=1}^{N} \alpha_{t-1}(j) a_{ij} c_{jm} b_{jm}(o_t) \beta_t(j)}{\alpha_T(N)} \tag{3.11}$$

$$\hat{\mu}_{im} = \frac{\sum_{t=1}^{T} \xi_{tm}(i) o_t}{\sum_{t=1}^{T} \sum_{m=1}^{M} \xi_{tm}(i)} \tag{3.12}$$

$$\hat{c}_{im} = \frac{\sum_{t=1}^{T} \xi_{tm}(i)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \xi_{tm}(i)} \tag{3.13}$$

$$\hat{\Sigma}_{im} = \frac{\sum_{t=1}^{T} \xi_{tm}(i)(o_t - \mu_{im})(o_t - \mu_{im}^T)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \xi_{tm}(i)} \tag{3.14}$$

3.5 Dil Modeli ve Sözlük

İstatistiksel dil modellemenin temel amacı konuşma tanıma gibi dil teknolojileri için doğal bir dilin dağılımına yakınsamaktır [17]. İstatistiksel dil modelleri temel olarak eğitim için kullanılan bir metinden gerekli istatistikleri çıkarır. Dolayısıyla doğru bir istatistik kestirimi için geniş miktarda eğitim verisi gerekmektedir.

Eşitlik (3.4)'teki $P(W)$ çarpanı konuşma tanımanın dil modelleme kısmına karşı düşmektedir. Bu olasılık dildeki bir kelime dizisinin olasılığıdır. $P(W)$ eşitlik (3.15)'teki gibi ifade edilebilir.

$$P(W) = P(w_1, w_2, ..., w_n); \tag{3.15}$$

Zincir kuralı kullanılarak eşitlik (3.15) aşağıdaki gibi ifade edilebilir:

$$P(W) = \prod_{i=1}^{N} P(w_i | w_1, ..., w_{i-1}) \tag{3.16}$$

Eşitlik (3.16)'daki $P(w_i|w_1,...,w_{i-1})$ olasılığı, $w_1,...,w_{i-1}$ kelimeleri söylendikten sonra w_i kelimesinin söylenme olasılığıdır. $w_1,...,w_{i-1}$ kelime dizisine geçmiş denmektedir.

Bazı durumlarda $P(w_i|w_1,...,w_{i-1})$ olasılığına yakınsamak imkansız hale gelmektedir. Çünkü bazı geçmişler eğitim verisinde sadece bir kere geçebilmektedir veya çok az tekrar etmektedir. Bu probleme çözüm olarak kelime geçmişi sınırlandırılmıştır. Eğer

bir kelimenin geçmişi $N-1$ kelimeyle sınırlandırılıyorsa N-gram dil modelleri elde edilmiş olur ve $P(W)$ bu durumda eşitlik (3.17)'deki gibi hesaplanır:

$$P(W) = \prod_{i=1}^{N} P(w_i | w_{i-N+1}, ..., w_{i-1}) \quad \textbf{(3.17)}$$

Konuşma tanıma işlemi sırasında kullanılan sözlük aslında bir kelime dizisidir. Bu kelime dizisinin içinde ayrıca her bir kelimeye ait telaffuzun karşı düşen ses dizisi bulunmaktadır. Konuşma tanıma uygulamaları için kullanıma açık bazı sözlükler mevcuttur. Bu sözlüklerdeki çoğu kelimeye sadece bir telaffuz karşı düşmektedir. Ancak bazı kelimeler için birden fazla telaffuz da bulunabilmektedir.

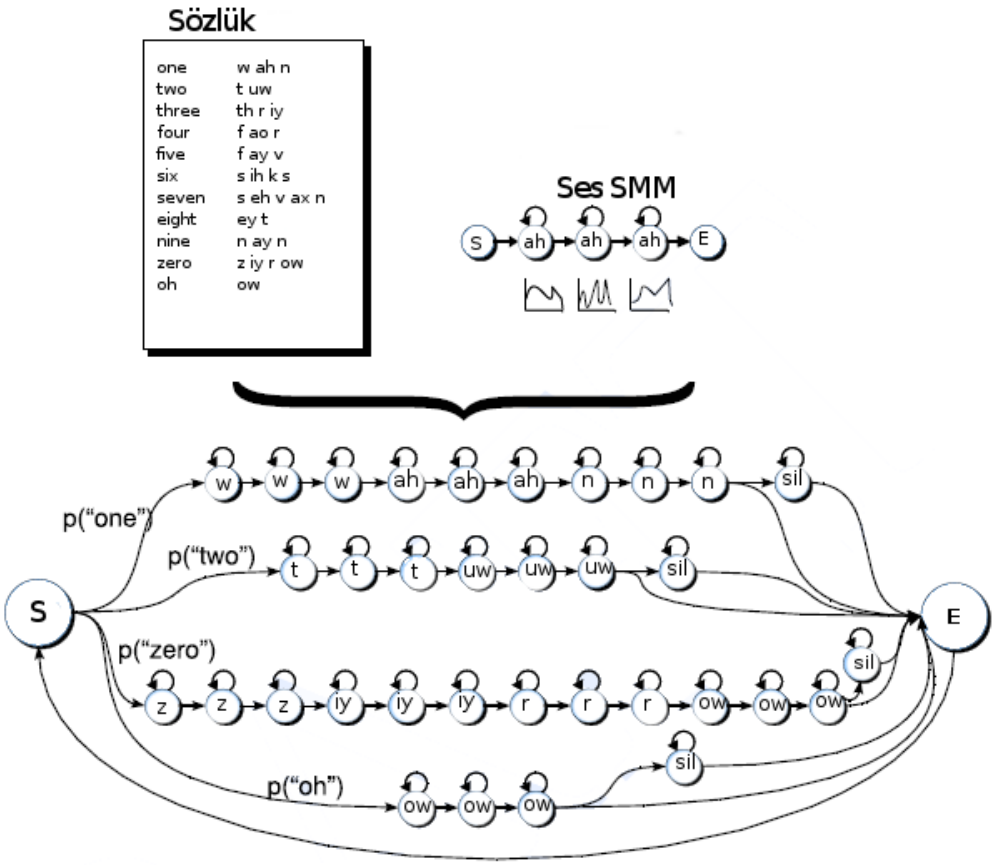
3.6 Kod Çözme

SMM yapısında kod çözme işleminin nasıl yapılacağı ikinci bölümde basit bir örnek üzerinden anlatılmıştı. Fakat konuşma tanıma sistemleri kod çözme işlemi sırasında akustik model skoru, dil modeli skoru ve okunuş sözlüğü gibi bileşenlerden gelen bilgileri kullanmaktadır. Bu bölümde kod çözme işlemi bir rakam tanıma uygulaması üzerinden anlatılacaktır.

Rakam tanıma uygulamaları küçük konuşma tanıma uygulamalarıdır ve dağarcık genişliği 11'dir. Bu 11 kelimenin 10'u rakamlardır diğer kelime ise "oh"'dur. Rakam tanıma işlemi sırasında kullanılacak gerekli SMM'leri oluşturmak için okunuş sözlüğünden kelimeye ait ses dizisi çekilir ve her bir ses için üç durumdan oluşan SMM'ler uç uca eklenir. Bu yapıda durumların kendine ve bir sonraki duruma olmak üzere iki geçiş olasılığı vardır. Ayrıca her kelimenin sonuna bir sessizlik durumu eklenir. Sessizlik durumu opsiyoneldir ve kelimeler arasındaki olası duraklamaları modellemek için kullanılır. Bu aşamada Baum-Welch algoritması yardımıyla SMM durum geçişlerinin ve gözlem dağılımlarını modelleyen GKM istatistiklerinin öğrenilmiş olduğu varsayılmaktadır.

Şekil 3.6 kod çözme sırasında kullanılabilecek örnek SMM yapısını göstermektedir. Bu yapıda başlangıç ve bitiş durumları sırasıyla 'S' ve 'E' harfleriyle gösterilmiştir. Bu durumlar herhangi bir gözlem üretmemektedir. Ancak her bir kelime bittikten sonra bitiş durumuna ve bitiş durumundan da başlangıç durumuna olan olası geçişler rastgele uzunluktaki bir rakam dizisinin tanınmasına olanak vermektedir.

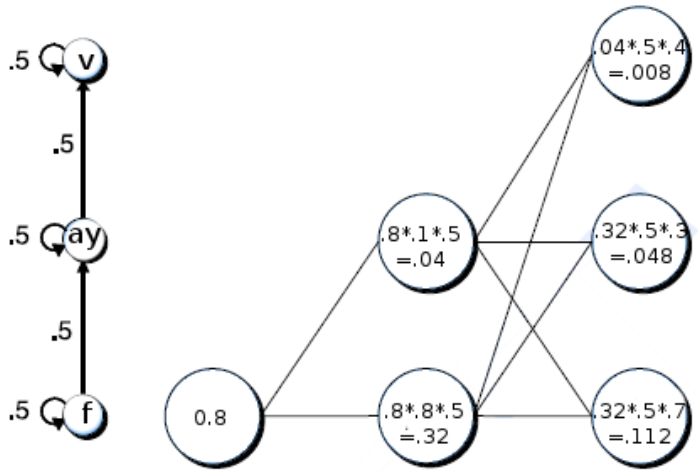
Rakam tanıma uygulamalarında çoğunlukla kelime olasılıkları kullanılmaz. Çünkü bu uygulamalarda genellikle tüm kelimeler eşit olasılığa sahiptir. Ancak Şekil 3.6'da yine de kelime olasılıkları grafa dahil edilmiştir.



Şekil 3.6: Rakam tanıma uygulaması için SMM örneği.

İkinci bölümden de hatırlanabileceği gibi bir SMM için verilen gözlem dizisine karşı düşen en olası durum dizisi Viterbi algoritması yardımıyla bulunmaktadır. Algoritma koşarken kafesteki her bir düğüme gelen yol olabilirliklerinden en yükseği tutulur ve o düğüme atanır. Ayrıca her bir düğüme gelen en yüksek olabilirlikli yolun hangi düğümden geldiği bilgisi de tutulur. Böylece kafes geri doğru tarandığında gözlem dizisine karşı düşen en olası durum dizisi bulunmuş olur.

Başlangıç için tanıma sözlüğünün sadece "five" kelimesinden oluştuğu varsayılsın. Şekil 3.7'de Viterbi algoritması ilk üç zaman aralığı için koşturulmuştur. Kafesin düğümleri için elde edilen olabilirlikler Çizelge 3.1'deki gözlem olabilirlikleri yardımıyla hesaplanmıştır. SMM durumları için kendine dönme ve bir sonraki duruma geçiş olasılıkları eşittir. Ayrıca Çizelge 3.1'de 5 zaman aralığı için düğümlerin değerleri verilmiştir. Çizelge 3.1'deki gözlem olabilirlikleri rastgele seçilmiştir.



Şekil 3.7: İlk 3 zaman aralığı için Viterbinin kafes üzerindeki akışı.

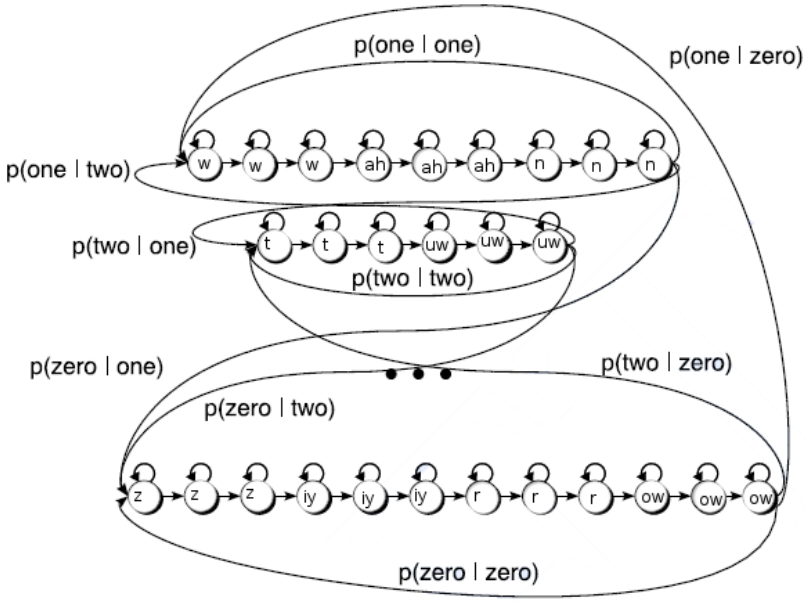
Çizelge 3.1: 5 zaman aralığı için gözlemler ve olabilirlikler.

V	0	0	0.008	0.0072	0.00672
AY	0	0.04	0.048	0.0448	0.0269
F	0.8	0.32	0.112	0.0224	0.00448
Zaman	1	2	3	4	5
B	f 0.8 ay 0.1 v 0.6	f 0.8 ay 0.1 v 0.6	f 0.7 ay 0.3 v 0.4	f 0.4 ay 0.8 v 0.3	f 0.4 ay 0.8 v 0.3

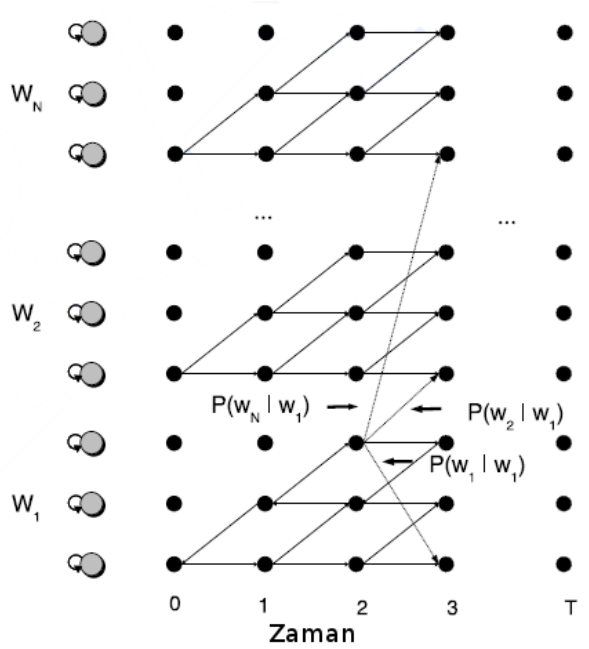
Viterbi algoritması ile bir kelime dizisini tanımak için, kelime içi durum geçişlerinde kullanılan olasılıklara ek olarak kelimeler arası geçiş olasılıklarının da hesaba katılması gerekir. Şekil 3.6'daki SMM yapısında her bir kelimenin birbirinden bağımsız olduğu varsayılmıştır dolayısıyla 1-gram dil modeli kullanılmıştır.

Kelimelerin birbirinden sonra gelme olasılıklarının da hesaba katıldığı bir SMM yapsı Şekil 3.8'de verilmiştir. Bu yapıda her bir kelimenin olasılığı kendinden bir önceki kelimeye bağlıdır. Dolayısıyla 2-gram dil modeli kod çözme işleminde kullanılmıştır. Bu SMM yapısının kod çözme işlemi sırasında kullanımı sonucu ortaya çıkan kafes Şekil 3.9 ile verilmiştir. Bu kafesteki düğümlere ait olabilirlikler hesaplandıktan sonra geri işaretçiler yardımıyla en olası durum dizisi dolayısıyla kelime dizisi bulunabilir.

Konuşma tanıma sistemlerinde hız önemli bir parametredir ve konuşma tanıma işleminin hızlandırılması adına literatürde birçok çalışma yapılmaktadır. Bu amaçla bir konuşma tanıma işlemi sırasında olası bütün yollar için hesap yapılmaz. Bunun yerine belirli bir olabilirliğin altında kalan yollar budanır.



Şekil 3.8: 2-gram olasılıklarının dahil edildiği SMM yapısı.



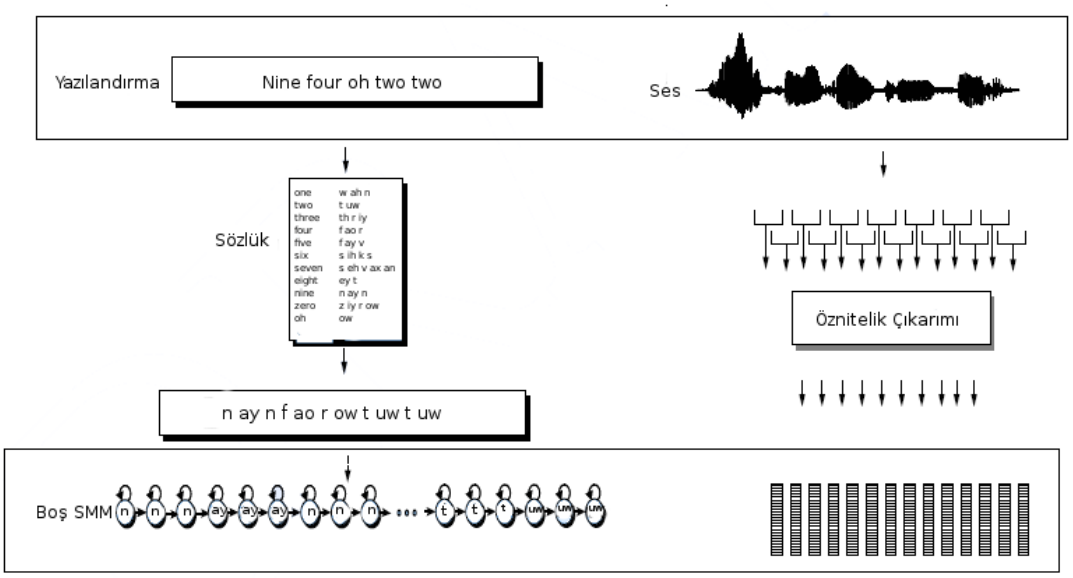
Şekil 3.9: 2-gram olasılıklarının kafes yapısında kullanımı.

Konuşma tanıma sistemlerinde budama işlemi için aktif liste adı verilen bir liste kullanılır. Aktif liste her bir zaman aralığı için tutulur. Bu listede o zaman aralığı için aktif olan durumlar vardır ve bir sonraki zaman aralığına sadece bu durumlardan çıkan yollar ile devam edilir. Aktif listeye giren durumları belirlemek için en yüksek olabilirlikli yol baz alınır ve bir limit değeri belirlenir. Daha sonra aktif listeye girecek durumlarda en iyi durumdan en fazla bu limit kadar kötü olması şartı aranır.

3.7 Akustik Model Eğitimi

Akustik model eğitimi için öncelikle eğitimde kullanılacak her bir cümleye ait ses dosyası ve karşı düşen yazılandırma mevcut olmalıdır. Ayrıca eğitim setine ait okunuş sözlüğü ve model seti hazırlanmalıdır. Yazılandırmalar ve okunuş sözlüğü yardımıyla bütün bir cümle için SMM yapısı oluşturulabilir. Bu işlemler için Şekil 3.10 incelenebilir.

Bu yapı için Baum-Welch algoritması koşturulmadan önce geçiş olasılıklarının ve gözlem dağılımlarına ait ilk değerler atanmalıdır. Bunun için genellikle şöyle bir yol izlenir: Durum geçişlerinde kendine dönme ve bir sonraki duruma geçme olasılıkları 0.5 olarak atanır. Bütün Gauss dağılımlarının ortalama ve varyans değerlerine ise tüm eğitim verisinin ortalama ve varyansı atanır. Daha sonra Baum-Welch algoritması koşturulur ve her bir yinelemede yeni SMM parametrelerine yakınsanmış olur.



Şekil 3.10: Baum-Welch algoritması girdileri.

Baum-Welch algoritması koşarken $\xi_t(i)$ olasılıkları yardımıyla t anında i durumundan geçen bütün yollar dikkate alınmış olur. Bu yaklaşım eğitim süresini uzatabilir. Bunun yerine daha verimli bir yaklaşım olan Viterbi eğitimi yapılır. Bu şekilde sadece t anında i durumundan geçen en yüksek olasılıklı yol dikkate alınır. Eğitim verisi üzerinde Viterbi algoritması koşturulmasına zorlanmış hizalama da denmektedir. Zorlanmış hizalama sırasında kod çözme işleminin aksine bir gözlem dizisine hangi kelime

dizisinin atanacağı bilinmektedir. Bu şekilde algoritma uygun geçiş olasılıkları ile belirli kelimelerden geçmeye zorlanmış olur.

Viterbi eğitimi sırasında herbir gözlem sadece bir duruma atanır. Bu durumda eşitlik (3.18) ve (3.19)'daki gibi her bir durum için kendisine atanan gözlemler ile ortalama ve varyans hesabı yapılarak yeni GKM parametrelerine yakınsanır.

$$\hat{\mu}_i = \frac{1}{T}\sum_{t=1}^{T} o_t \quad o_t\text{: } i \text{ durumuna atanmış gözlem} \tag{3.18}$$

$$\hat{\sigma}_i = \frac{1}{T}\sum_{t=1}^{T} (o_t - \mu_i)^2 \quad o_t\text{: } i \text{ durumuna atanmış gözlem} \tag{3.19}$$

3.8 Değerlendirme Metriği

Konuşma tanıma için standart metrik kelime hata oranıdır. Kelime hata oranı bir konuşma tanıma sisteminin döndürdüğü bir kelime dizisinin doğru kelime dizisinden yani referanstan ne kadar farklı olduğunu ölçer. Bunun için öncelikle iki kelime dizisi arasındaki Levensthein uzaklığı hesaplanır. Bu işlemin sonucunda konuşma tanıma yoluyla elde edilen kelime dizisinden doğru kelime dizisini üretebilmek için gerekli minimum sayıda değiştirme, ekleme ve silme sayısı elde edilmiş olur. Daha sonra kelime hata oranı aşağıdaki gibi hesaplanır:

$$\text{Kelime Hata Oranı} = 100 \times \frac{\text{Ekleme} + \text{Değiştirme} + \text{Silme}}{\text{Referanstaki toplam kelime sayısı}} \tag{3.20}$$

3.9 Özet

Bu bölümde bir konuşma tanıma sisteminin mimarisi ana hatlarıyla anlatılmıştır. Konuşma tanımada en çok kullanılan özniteliklerden bir olan MFKK çıkarım prosedürü, SMM gözlem dağılımlarını modellemek ve akustik olabilirlikleri elde etmek için GKM kullanımı, dil modeli ve sözlüğün konuşma tanıma işlemindeki yeri ayrıntılı olarak anlatılmıştır.

Yukarıda anlatılan adımlardan sonra temel adımlar olan kod çözme ve eğitim adımları anlatılmıştır ve son olarak da başarım değerlendirme metriği hakkında bilgi verilmiştir.

4. ÖRNEKLEM TABANLI KONUŞMA TANIMA

Bu bölümde örneklem tabanlı konuşma tanıma hakkında bilgi verilecektir. Öncelikle literatürde çokça kullanılan ve Destek Vektör Makineleri, Yapay Sinir Ağları ve Saklı Markov Modelleri gibi örnekleri olan global modelleme teknikleri ve örneklem tabanlı modelleme tekniklerinin farkı üzerinde durulacaktır. Daha sonra örneklem tabanlı teknikleri konuşma tanımaya uygularken takip edilen adımlar açıklanacaktır. Bölümün devamında ise seyrek gösterimin örneklem tabanlı konuşma tanıma uygulamalarında kullanımı incelenecektir. Son olarak ise bu çalışmada gerçeklenen örneklem tabanlı yöntem anlatılacaktır.

4.1 Global Modelleme ve Örneklem Tabanlı Modelleme

Sınıflandırma ve tanıma gibi problemlerin çözümü, gerçek dünyada karşılaşılan verinin ve içerisindeki belirsizliğin kurallı bir şekilde modellenmesini gerektirmektedir. Verinin içerisindeki belirsizlik, verinin üretiminde birçok kaynağın rol oynaması ve bu kaynaklara ait etkilerin doğrudan ölçülememesinden kaynaklanmaktadır. Örneğin konuşma işareti göz önüne alındığında, belirsizliğin nedeni ses üretim yolunun her insan için farklı olması veya işaretin yapısını bozan gürültüler olabilir. Modellemenin temel amacı ise eğitim verisinden genel bir kanı elde etmek ve elde edilen bu kanıyı görülmemiş bir veri üzerinde tanıma, karar verme veya sınıflandırma gibi işlemler sırasında kullanmaktır.

Modelleme yöntemleri iki farklı kategoride toplanabilir. Bu kategorilerden ilki test verisi görülmeden önce eğitim verisinin tamamını kullanarak bir model oluşturan yöntemleri kapsar. İkinci kategorideki yöntemler ise her bir test örneği için eğitim verisindeki bazı örnekleri akıllı bir şekilde seçerek yerel modeller oluşturan yöntemlerdir. İlk kategorideki yöntemler global veri modelleme yöntemleri, ikinci kategorideki yöntemler ise örneklem tabanlı yöntemler olarak adlandırılmaktadır.

Otomatik konuşma tanıma alanında otuz yılı aşkın bir süredir global veri modellemesinin bir örneği olan SMM'ler kullanılmaktadır. SMM'ler akustik

işaretlerin zamanla değişen yapısını modellemek için kullanılırken, GKM'ler de SMM gözlem dağılımlarını modellemek için kullanılmaktadır. Bu yapı Baum-Welch algoritması ile birlikte otomatik konuşma tanıma için ölçeklenebilir ve güvenilir bir modelleme yöntemi sunmaktadır.

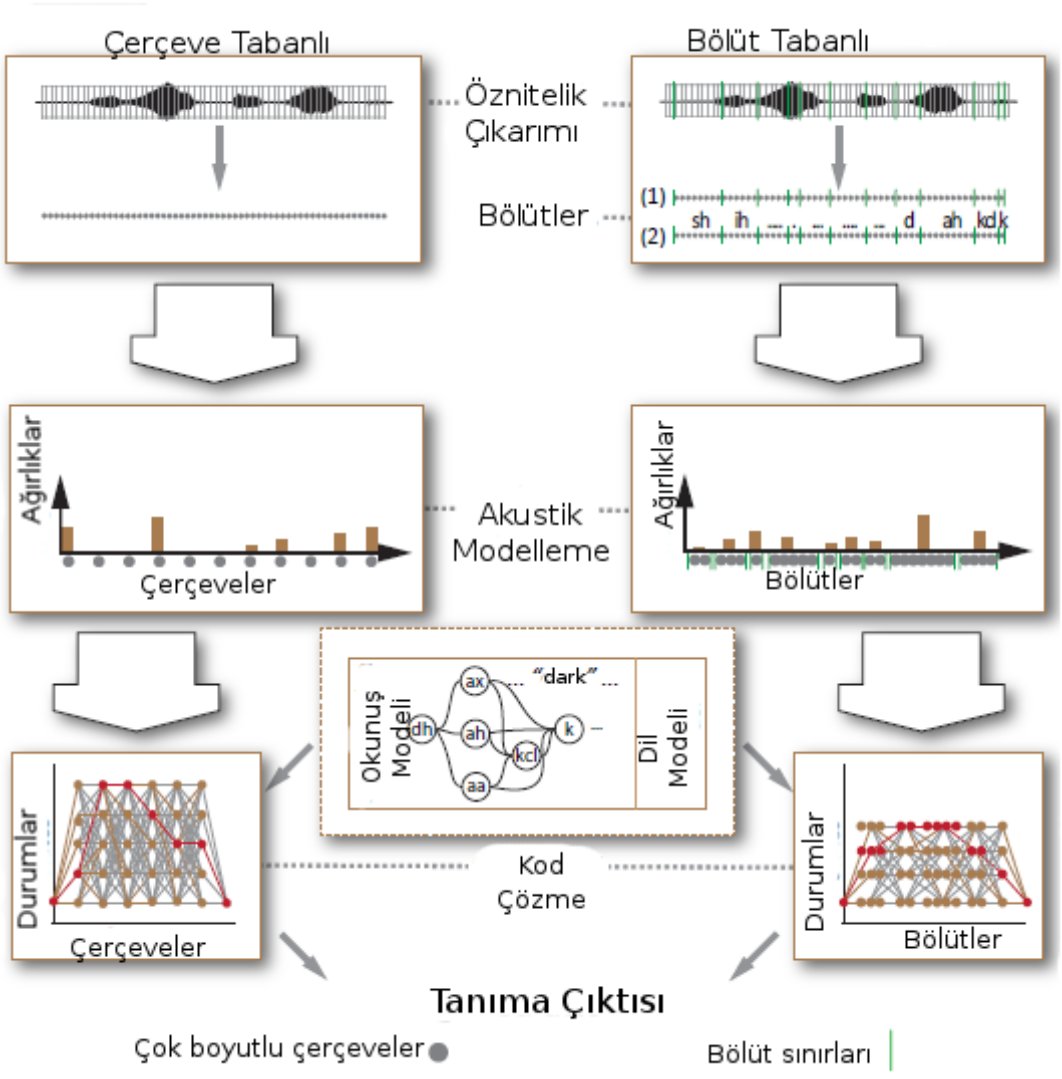
Global modelleme yöntemleri ile, yeterli eğitim verisi bulunduğunda başarılı modeller eğitilebilmektedir. Ancak sınırlı eğitim verisi ile çalışıldığında bu modeller gözlenen verileri yeterince ayrıntılı bir şekilde betimleyememektedir. Ayrıca veriyi yeterince ayrıntılı bir şekilde modellemek, modeldeki parametre sayısını arttırmakta ve model eğitim işlemini zorlu hale getirmektedir. Örneklem tabanlı yöntemler her bir test verisi için bilgi taşıyan örneklemleri seçerek bir model oluşturduğundan bu verimsizliği aşmak için uygun yöntemlerdir.

4.2 Örneklem Tabanlı Yöntemler

Örneklem tabanlı yöntemler kullandıkları öznitelik vektörlerine göre çerçeve tabanlı ve bölüt tabanlı çalışan yöntemler olmak üzere iki kategoriye ayrılır. Şekil 4.1'den görülebileceği gibi bu iki yaklaşım arasındaki temel fark çerçeve tabanlı yöntemlerde sabit uzunluklu öznitelik vektörleri kullanılırken, bölüt tabanlı yöntemlerde ise bir ses birimine ait değişken uzunluklu bölütlerin kullanılmasıdır.

Diğer örüntü tanıma yöntemlerinde olduğu gibi örneklem tabanlı yöntemler kullanılırken de ilk olarak öznitelik çıkarımı yapılmaktadır. Elde edilen öznitelikler sabit uzunluklu vektörler(çerçeve) veya değişken uzunluklu vektör dizileri(şablon) olabilir. Örneklem tabanlı yöntemlerin kullanımından önce, elde edilen öznitelikler Temel Bileşen Analizi [18] ve Doğrusal Ayrım Analizi [19] gibi çokça kullanılan öznitelik dönüşümlerinden geçirilebilir. Bu dönüşüm yöntemleri yardımıyla hem daha az boyutlu öznitelikler üzerinde çalışma imkanı elde edilir hem de veri içerisinde bulunan gereksiz bilgiler ayıklanmış olur. Bu dönüşüm teknikleri hem global modelleme hem de örneklem tabanlı modelleme yöntemleri ile birlikte kullanılabilir.

Öznitelikler belirlendikten sonra örneklem tabanlı konuşma tanıma temel olarak üç adımdan oluşmaktadır. Bunlar örneklem seçimi, örnek modellemesi ve çerçeve veya bölüt tabanlı kod çözme adımlarıdır.



Şekil 4.1: Çerçeve ve bölüt tabanlı konuşma tanıma [21].

- Örneklem Seçimi: Test sırasında karşılaşılan örnekleri en iyi ifade edebilecek örneklem kümesi eğitim verileri arasından seçilir. Örneklem kümesi rastgele oluşturulabileceği gibi akıllı yöntemler kullanılarak da oluşturulabilir.

- Örnek Modellemesi: Test sırasında karşılaşılan örnekleri en iyi modelleyebilecek örneklemler, belirli bir kritere göre birinci adımda belirlenen kümenin elemanları arasından seçilir. Bu adımdan sonra, ilgili örneklemler ve her bir örneklemin test örneğini modellemek için yaptığı katkıyı ifade eden ağırlıklar elde edilir. Test örneğini modelleyecek uygun örneklemleri bulmak için K komşu algoritması veya seyrek gösterim yöntemleri kullanılabilir. Örneğin K komşu algoritması eğitim örnekleri arsından test örneğine en yakın olanları seçer ve bu örnekleri kullanarak modelleme yapar. Seyrek gösterim yöntemleri ise sözlükteki atomların bu test örneğini oluşturmadaki katkısını hesaplar.

- Çerçeve veya Bölüt Tabanlı Kod çözme: Bir gözlem için, ikinci adımdan elde edilen örnek modeli kullanılarak ses altı birimlere ait akustik olabilirlikler bulunur. Bu olabilirliklere ek olarak dil modelinden elde edilen olabilirlikler de hesaba katılır ve Viterbi algoritması yardımıyla kod çözme işlemi yapılır.

Bu çalışmada çerçeve tabanlı deneyler yapılmıştır. Dolayısıyla sözlükte bulunan örneklemler sabit uzunluğu olan vektörlerdir. Bölümün geri kalanında da bu yöntemlerin uygulamalarından bahsedilecektir.

4.3 Seyrek Gösterimin Konuşma Tanımada Kullanımı

Seyrek gösterim yöntemleri daha çok sıkıştırmalı algılama [20, 21] alanındaki kullanımları ile bilinmektedir. Bununla birlikte son yıllarda örüntü tanıma problemlerinde de bu yöntemler çokça kullanılmaktadır. Seyrek gösterim algoritmaları bir test örneğini sözlükteki atomların doğrusal birleşimi cinsinden modellemektedir.

Son yıllarda seyrek gösterim uygulamalarının işaret örnekleri içeren sözlükler ile birlikte kullanıldığı çalışmalar yapılmıştır. Bu çalışmalarda sözlükteki örneklemler çerçevelerin kendileri [22,23], belirli bir sayıda çerçevenin uç uca eklenmesi ile oluşan çerçeve dizileri [14, 23, 24] veya değişken uzunluklu şablonlardan yeniden örnekleme yardımıyla elde edilen sabit uzunluklu çerçeve dizileri [25] olabilir.

Bölüm 4.2'deki adımlara sadık kalarak seyrek gösterime ait akış aşağıda verilmiştir:

4.3.1 Örneklem seçimi

Bir test örneği x'i seyrek gösterim ile modelleyebilmek için öncelikle sütunları ilgilenilen işaret ailesinden alınan örnekler olan A matrisi öğrenilmelidir. A matrisi $D \times N$ boyutlarında bir matristir ve D örneklemlerin uzunlukarı, N ise sözlükteki örneklem sayısıdır.

Seyrek gösterim yöntemleri her bir test örneği x için bir α vektörü bulmaktadır. Bu vektör eşitlik (4.1)'den görülebileceği gibi sözlükteki her bir örneklemin ilgili test örneğinin oluşmasına yaptığı katkıyı belirlemektedir. Sözlüğün çok geniş olması durumunda problemin çözümü olanaksız hale gelmektedir. Bu amaçla A matrisi bütün eğitim setinin bir alt kümesi şeklinde seçilmektedir. Bu seçim işlemi [14, 23] çalışmalarındaki gibi rastgele olabileceği gibi çalışma anında K komşu algoritması yardımıyla bir önseçim işlemi ile de yapılabilir [22].

$$x = A\alpha \tag{4.1}$$

İdealde optimum bir α vektörü seyrek olmalıdır. Çünkü x'e sadece aynı sınıfta bulunduğu sözlük elemanlarından katkı gelmelidir [20, 26, 27]. Örneklem tabanlı konuşma için α vektörü L_1 kısıtı altında Öklit mesafesini minimum yapacak şekilde hesaplanabilir [28]. Seyrek gösterim hesabı için başka metrikler de kullanılmaktadır. Örneğin [14] çalışmasında Kullback-Leibler metriği kullanılmıştır. [25] çalışmasında ise L_1 ve L_2 kısıtlarının aynı anda kullanımı ile grup seyrekliği zorlanmıştır.

4.3.2 Örnek modellemesi

Bu adım için genelde şu yaklaşım kullanılmaktadır: Sözlükteki herbir örneklemi bir sınıf ile ilişkilendiren bir etiket matrisi G kullanılır. Sonsal olabilirlikler ise $\hat{P}(q|x) \propto g_q\alpha$ yaklaşımıyla hesaplanır. Sonsal olasılıklar [29] çalışmasında ses sınıflarına karşı düşerken, [14] çalışmasında SMM durumlarına karşı düşmektedir. Böylece her bir örneklem için bir ağırlık elde edilirken aynı zamanda bu örneklemin saklı değişkenler olan seslerle veya ses altı birimlerle ilişkisi de modellenmiş olur. Bu ilişki yardımıyla kod çözme işlemi klasik Viterbi kod çözücüsü ile gerçekleştirilebilir.

4.3.3 Kod çözme

Konuşma tanıma işlemi için $\hat{P}(q|x)$ olabilirlikleri çerçeve tabanlı gözlem olasılıklarına çevrilir. Herhangi bir x gözlemi için tüm $\hat{P}(q|x)$ olabilirlikleri toplamları 1 olacak şekilde normalize edilir. Elde edilen yeni değerler artık $p(x|q)$ olasılıklarıdır. Bu adımdan sonra Viterbi algoritması yardımıyla, bir gözlem dizisi için maksimum olabilirliğe sahip ses dizisi bulunur. Eğer ses sınıfları için önsel bir dağılım modeli varsa, bu modelden gelen olasılıklar gözlem olasılıklarına dahil edilebilir [30].

4.4 Seyrek Gösterim ve Gürbüz Konuşma Tanıma

Bu bölümde arka plan gürültüsü içeren konuşmaları tanımak için seyrek gösterimin nasıl kullanıldığı anlatılacaktır. Öncelikle, literatürde kullanılan yöntemlerden bahsedilecektir. Daha sonra bu yöntemlerden biri olan ve bu çalışmada da gerçeklenmiş olan "Seyrek Sınıflandırma" yöntemi anlatılacaktır.

4.4.1 Seyrek gösterimin gürbüz konuşma tanıma uygulamaları

Daha önceki bölümlerde anlatıldığı gibi seyrek gösterim yöntemleri test örneklerini, sözlükte bulunan örneklemlerin doğrusal birleşimi cinsinden modellemektedir. Seyrek gösterimin bu özelliği gürbüz konuşma tanıma için de verimli yöntemler geliştirilmesine olanak sağlamıştır.

Gürbüz konuşma tanıma için kullanılan seyrek gösterim yöntemlerinden biri [23] çalışmasında anlatılmıştır. Bu yöntem gürültü altında da bazı özniteliklerin bozulmadığını ve güvenilir kaldığını varsaymaktadır. Güvenilir olmayan öznitelikler için bir kestirim yapılmakta ve elde edilen yeni özniteliklerden bir SMM-GKM sistemi eğitilmektedir. Böylece aslında bir öznitelik iyileştirme işlemi yapılmaktadır.

Diğer bir yöntem ise kaynak ayrıştırma tabanlı bir yöntemdir. Buna göre her bir test örneği temiz konuşma ve gürültü sözlüklerindeki örneklemlerin doğrusal birleşimi olarak modellenir. Bunun için Mel-Genlik domeninde çalışılır ve gürültünün toplamsal olduğu varsayılır. Bu adımdan sonra sadece temiz konuşma örneklemleri ve bu örneklemlere karşı düşen katsayıları kullanarak temiz konuşma kestirimi yapılır [14]. Gürültülü konuşma örneklemleri ve karşı düşen ağırlıklar kod çözme sırasında kullanılmaz.

4.4.2 Seyrek sınıflandırma

Seyrek gösterim uygulamalarında bir sinyal birden çok kaynağın karışımı olarak modellenebilmektedir. Bunun için her bir kaynağı ifade eden sözlükler kullanılmaktadır. Daha sonra bu sözlüklerdeki örneklemlerin olası en seyrek birleşimi Negatif Olmayan Matris Ayrıştırması (NOMA) veya Sıkıştırılmış Algılama (SA) teknikleri yardımıyla bulunmaktadır. Tek bir sözlüğün aktif örneklemleri ve karşı düşen ağırlık vektörleri kullanılarak istenen kaynağın sinyaldeki bileşenine yakınsanmaktadır.

Seyrek gösterimin diğer uygulama alanlarından biri daha önce de bahsedildiği gibi örüntü tanıma uygulamalarıdır. Bunun için sözlükteki örneklemlerin sınıf etiketleri ile ilişkilendirilmesi gerekmektedir. Çünkü sonrasında bu örneklemlerin ağırlıkları, gözlemlerin oluşmasında ilgili sınıfın ne kadar payı olduğunu gösterecektir.

Seyrek sınıflandırma algoritması da kaynak ayrıştırma ve sınıflandırma uygulamalarında kullanılan yöntemleri birleştirerek gürbüz konuşma tanıma yapmayı amaçlayan bir yöntemdir [14]. Burada kaynaklar dolayısıyla da sözlükler temiz konuşma ve gürültü bileşenleriyken sınıflar ise ses veya ses altı birimleri olmaktadır.

Seyrek sınıflandırma işlemi yardımıyla aslında melez bir konuşma tanıma yöntemi elde edilmektedir. Buna göre sınıf olarak kabul edilen ses altı birimler arasındaki geçiş SMM durum geçiş olasılıkları ile modellenirken, seslere ait olabilirlikler GKM yerine sözlükteki temiz konuşma örneklemlerinin ilgili test örneğinin temiz konuşma kısmına yaptığı katkıyı belirten ağırlıklar veya aktivasyonlar ile modellenmektedir. Bu tarz melez yaklaşımlar yapay sinir ağları yardımıyla ses olabilirliklerinin hesaplandığı çalışmalarda da kullanılmıştır [31].

4.4.2.1 Gürültülü konuşmanın seyrek gösterimi

Konuşma sinyallerinin en temel ifade biçimi frekans-zaman uzayındaki enerji dağılımını gösteren genlik spektrogramlardır. Genlik spektrogramı denmesinin nedeni enerjilerin karekökü kullanıldığı içindir. Bir genlik spektrogramı aslında $B \times T$ boyutlarında bir matristir. Burada B enerji dağılımının hesaplandığı frekans bandı sayısıdır. T ise band enerjilerinin kaç çerçeve için hesaplandığını gösterir. Gösterimi kolaylaştırmak için bu matrisler uzunluğu $E = B \times T$ olan vektörlere dönüştürülür. Bu işlem matrisin sütunlarının sırayla uç uca eklenmesi şeklinde yapılır.

Herhangi bir konuşma spektrogramı s'in bir sözlükte bulunan temiz konuşma örneklemlerinin doğrusal ve negatif olmayan birleşimi şeklinde modellenebileceği varsayılsın. Burada sözlükteki örneklemlerde eğitim setinden aynı yolla elde edilmiş genlik spektrogramlarıdır ve yukarıda belirtildiği gibi birer vektör haline getirilmişlerdir.

$$s \approx \sum_{j=1}^{J} a_j^s x_j^s = A^s x^s \quad x^s > 0 \text{ olmak üzere} \tag{4.2}$$

$$A^s = [a_1^s a_2^s ... a_J^s] \tag{4.3}$$

Eşitlik (4.2)'deki A matrisi ve x vektörünün üst indisi olan s harfi temiz konuşma örneklemlerini ifade etmek için kullanılmıştır. Burada A^s matrisi eşitlik (4.3)'deki gibi spektrogramların vektör hallerinin sütun olarak barındıran matristir ve $j = 1, 2, ...J$ örneklem indisleridir. x vektörü ise her bir örnekleme ait ağırlıkları barındırmaktadır. Bu ağırlıklara aktivasyon da denmektedir.

Önceki çalışmalar [32] x^s vektörünün oldukça seyrek olduğunu göstermiştir. Herhangi bir konuşma spektrogramı yalnızca birkaç örneklemin katkısıyla yeterli kesinlikte temsil edilebilmektedir. Ayrıca ses analizinde genlik spektrogramı kullanan yöntemler için aktivasyonların negatif olmaması da önemli bir noktadır [33].

Temiz konuşmada olduğu gibi gürültü için de $B \times T$ boyutularında spektrogramlar kullanılabilir. Bu spektrogramlar konuşma spektrogramları gibi vektör haline getirilebilir. Bu durumda elde edilen n vektörü $k = 1, 2, ..., K$ olmak üzere a_k^n gürültü örneklemlerinin doğrusal birleşimi cinsinden ifade edilir.

Gürültülü konuşmaya ait bir Y spektrogramı y vektörü haline getirildikten sonra eşitlik (4.4), (4.5), (4.6) ve (4.7)'daki gibi gürültü ve konuşma örneklemlerinin doğrusal birleşimi cinsinden ifade edilebilir. x_n gürültü örneklemlerine ait aktivasyon vektörü iken A_n de gürültü örneklemlerini içeren matristir. A matrisi ise $E = B \times T$ ve $L = J + K$ olmak üzere $E \times L$ boyutlarında bir matristir ve konuşma ve gürültü örneklemlerinin tamamını içerir. A matrisine sözlük de denmektedir. x ise A matrisindeki bütün örneklemlere ait aktivasyonları barındırır ve seyrek gösterim olarak adlandırılır.

$$y \approx s+n \tag{4.4}$$

$$y \approx \sum_{j=1}^{J} a_j^s x_j^s + \sum_{k=1}^{K} a_k^n x_k^n \tag{4.5}$$

$$= \begin{bmatrix} A_s A_n \end{bmatrix} \begin{bmatrix} x_s \\ x_n \end{bmatrix} \tag{4.6}$$

$$= Ax \tag{4.7}$$

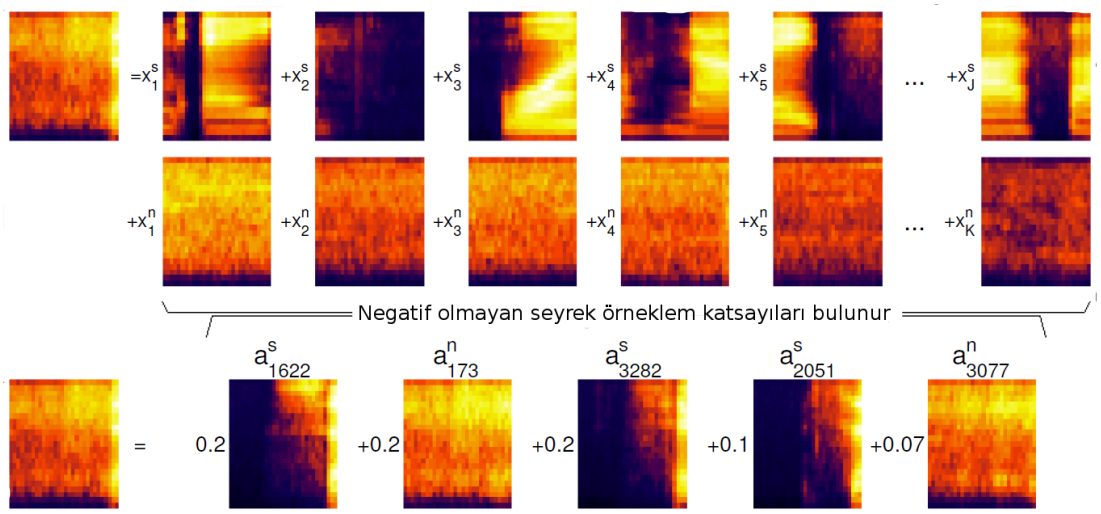
4.4.2.2 Aktivasyonların bulunması

Örneklemlerin doğrusal birleşimi (4.8)'deki maliyet fonksiyonu minimize edilerek bulunur. Bu fonksiyondaki ilk terim model ile gürültülü gözlem arasındaki uzaklığı ölçer. İkinci terim ise aktivasyon vektörünün içindeki sıfırdan farklı elemanları L_p norm kullanarak cezalandırmaktadır. Böylece seyrek gösterim zorlanmaktadır. Bu çalışmada L_1 norm kullanılmıştır. Böyle bir maliyet fonksiyonunu minimize etmek için [34] çalışmasında yinelemeli bir güncelleme kuralı önerilmiştir ve bu kural (4.9) ile verilmektedir. Eşitlik (4.8)'deki uzaklık terimi için Kullback-Liebler(KL) uzaklık metriği kullanılmıştır. (4.9) ile verilen güncelleme kuralında $.*$ ve $./$ işaretleri eleman eleman çarpma ve bölme işlemlerine karşı düşmektedir. 1 ise bütün elemanları 1'lerden oluşan E uzunluğunda bir vektördür. λ vektörü ise x'in içindeki sıfır olmayan elemanların ne kadar cezalandırılacağını belirler.

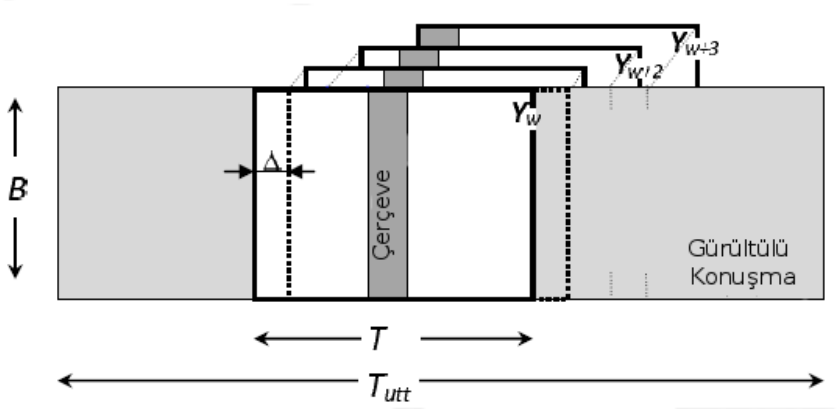
$$d(y,Ax) + ||\lambda .* x||_p \quad x > 0 \text{ olmak üzere} \tag{4.8}$$

$$x \leftarrow x .* \left(A^T * \left(y./(Ax) \right) \right) ./ (A^T 1 + \lambda) \tag{4.9}$$

Bu kural yardımıyla test sırasında karşılaşılan bir pencerenin sözlükteki örneklemlerin doğrusal birleşimi cinsinden ifade edilebilmesi için gerekli aktivasyonlar bulunur. Örnek bir pencerenin bu şekilde gösterimi Şekil 4.2'de verilmiştir.



Şekil 4.2: Bir test örneğinin sözlükteki örneklemlerin doğrusal birleşimi ile gösterimi [14].

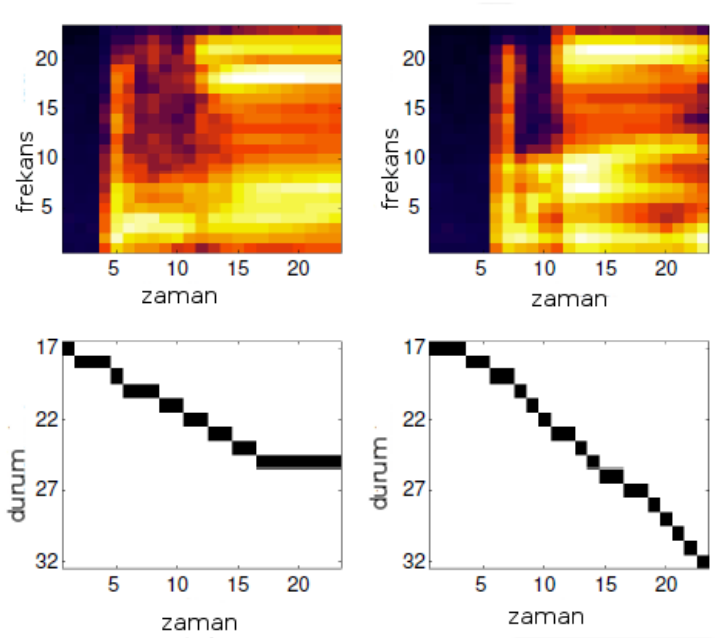


Şekil 4.3: Kayan pencere yaklaşımı [14].

4.4.2.3 Kayan pencere yaklaşımı

Konuşma sinyalinin seyrek gösterimi için bölüm 4.4.2.1'de anlatılan model kısa konuşma işaretleri için uygundur. Ancak keyfi uzunluktaki bir konuşma işareti için [24] çalışmasındaki yaklaşım kullanılabilir. Bu yaklaşıma göre her bir konuşma örneklem boyutuyla aynı boyutta örtüşen pencerelere bölünür. Daha sonra herbir pencere için seyrek gösterim bulunur. Son olarak da elde edilen pencereler örtüşen kısımların ortalaması alınarak birleştirilir.

Şekil 4.3'te görüldüğü gibi herhangi bir konuşma işareti $B \times T_{utt}$ boyutlarında bir Y_{utt} spektrogramı şeklinde ifade edilsin. $B \times T$ boyutlarında bir pencere Y_{utt} bandı üzerinde Δ öteleme miktarı ile kaydırılır ve $Y_1, Y_2, ..., Y_W$ pencereleri elde edilir. W bir konuşma spektrogramından çıkan toplam pencere sayısıdır.



Şekil 4.4: İki farklı örneklem için etiket matrisleri [14].

Konuşma spektrogramından çıkarılan her bir pencere bölüm 4.4.2.1'deki gibi vektör haline getirilir. Elde edilen $y_1, y_2, ..., y_w$ vektörleri Ψ gözlem matrisinin vektörleridir. Ψ matris $E \times W$ boyutlarında bir matristir. Bu matris kullanılarak eşitlik (4.10)'daki gibi bütün bir konuşma spektrogramı için seyrek gösterim bulunabilir. $X = [x_1, x_2, ..., x_W]$ olmak üzere X matrisinin her bir sütunu karşı düşen pencere için aktivasyonları içerir. Böylece bu matris bütün bir konuşma için aktivasyon vektörlerini içermiş olur.

$$\Psi \approx AX \tag{4.10}$$

4.4.2.4 Seyrek sınıflandırma

Seyrek sınıflandırma yöntemi melez bir yöntemdir ve [24] çalışmasında önerilmiştir. Bu yöntemde SMM yapısı sabit tutulurken, ses olabilirliklerine GKM ile yakınsamak yerine örneklem aktivasyonları kullanılır.

Sözlüğü oluşturmak için kullanılan temiz konuşma örneklemlerine ait her bir çerçevenin SMM durumları ile etiketlendiğini varsayılsın. Bu durumda her bir örneklemin etiket matrisi $\mathscr{L}_j$ oluşturulabilir. Bu matris ikili bir matristir ve $Q \times T$ boyutlarındadır. Q toplam SMM durumu sayısıdır. $\mathscr{L}_j$ matrisi bir örneklemin zaman ve SMM durumu ilişkisini verir. Örnek etiket matrisleri için Şekil 4.4 incelenebilir.

Herhangi bir w penceresi için hesaplanan aktivasyonları $x^s_{w,j}$ ile gösterecek olursak w penceresi için durum olabilirlik matrisi L_w aşağıdaki gibi hesaplanır:

$$L_w = \sum_{j=1}^{J} \mathscr{L}_j x^s_{w,j} \tag{4.11}$$

L_w matrisinin sütunları $t = 1,2,...T$ olmak üzere $l_{w,t}$ şeklinde ifade edilir. Daha sonra durum olabilirlikleri bu vektörlerin konuşmadaki sırası göz önünde bulundurularak birleştirilmesiyle elde edilir. Örtüşen pencerelere ait olabilirlikler toplanır. Durum olabilirlik hesabı şu şekilde ifade edilebilir:

$$l^{utt}_{\tau} = \sum_{t=max(1,\tau-T_{utt}+T)}^{min(T,\tau)} l_{\tau-t+1,t} \tag{4.12}$$

Durum olabilirlikleri bu şekilde belirlendikten sonra SMM durum yapısı ve geçiş olasılıkları muhafaza edilerek Viterbi algoritmasıyla kod çözme işlemi yapılır.

4.5 Özet

Sınıflandırma ve tanıma gibi problemler çözülürken literatürde kullanılan yöntemler genel olarak global veri modellemesi ve örneklem tabanlı modelleme başlıkları altında toplanabilir. Global veri modellemesinin konuşma tanıma alanında en çok kullanılan örneği SMM-GKM sistemleridir. Örneklem tabanlı yöntemler ise son yıllarda konuşma tanıma alanında üzerinde çalışılan popüler araştırma konularından biridir.

Örneklem tabanlı yöntemler temel olarak çerçeve ve bölüt tabanlı yöntemler başlıkları altında incelenebilir. Çerçeve tabanlı yöntemler sabit uzunluklu vektörleri öznitelik olarak kullanırken bölüt tabanlı yöntemler değişken uzunlukta öznitelikler kullanabilmektedir.

Bu bölümde bu çalışmada da gerçeklenmiş olan seyrek sınıflandırma yöntemi ayrıntılarıyla açıklanmıştır. Bu yöntem çerçeve tabanlı bir yöntemdir ve iki önemli yaklaşımı barındırmaktadır. Birincisi NOMA yöntemini kullanarak konuşma ve gürültü sözlükleri yardımıyla bir kaynak ayrıştırması yapmaktadır. İkincisi ise sözlükteki her bir örneklemin hangi ses altı birimi tarafından üretildiğini kullanarak test sırasında karşılaşılan örneklerin oluşmasında her bir sınıfın katkısını hesaplamaktadır.

Böylece SMM-GKM sistemlerinde kullanılan klasik Viterbi kod çözücüsünün bu yapıyla birlikte kullanımına imkan vermektedir.

5. DENEYLER

Bu bölümde yapılan deneyler ve sonuçları anlatılacaktır. Öncelikle eğitim ve test setlerinin nasıl elde edildiği ve hangi işlemlerden geçirildiği anlatılacaktır. Daha sonra bu setler kullanılarak HTK(Hidden Markov Model Toolkit) [35] ile yapılan eğitim ve test işlemleri, sonrasında ise seyrek sınıflandırma yöntemi için yapılan çalışmalar anlatılacaktır.

5.1 Veri Seti

Bu çalışmada TIDIGITS [36] veri seti baz alınarak eğitim ve test setleri hazırlanmıştır. TIDIGITS veri setinde yetişkin erkek ve bayanlara ait konuşmalar bulunmaktadır. Verinin konuşmacı gruplarına göre dağılımı Çizelge 5.1'de verilmiştir. Konuşmlalar sadece 10 adet İngilizce rakam ve sıfırın bir diğer versiyonu olan "OH" kelimesini içermektedir.

Çizelge 5.1: TIDIGITS veri setindeki konuşmacı dağılımı.

	Erkek	**Bayan**
Eğitim	55	57
Test	56	57

Bütün kümeden üzerinde çlışılacak eğitim ve test alt kümelerini belirlemek için AURORA-2 [5] veri seti oluşturulurken izlenen adımlar uygulanmıştır. Eğitim için yetişkin bayan ve yetişkin erkek setlerinden 8440 dosya rastgele seçilmiştir. Bu iki setten de 55'er konuşmacı eğitim setine dahil edilmiştir. Bu dosyalar yine rastgele bir şekilde 20 alt kümeye ayrılmıştır. Çünkü çok durumlu eğitimde 4 farklı gürültü tipi ve 5 farklı işaret gürültü oranına sahip alt kümeler kullanılacaktır.

Bu adımdan sonra elde edilen 20 alt kümeye istenen işaret gürültü oranı ve ilgili gürültü için FaNT [37] aracı yardımıyla gürültü eklenmiştir. Böylece çok durumlu eğitim için gerekli eğitim verisi elde edilmiştir. Bu verinin gürültü eklenmemiş hali de ayrıca ileride anlatılacak deneylerde kullanmak üzere muhafaza edilmiştir.

Test seti için ise TIDIGITS test kümesindeki yetişkin bayan ve yetişkin erkek alt kümelerinden rastgele 4004 adet dosya seçilmiştir. Elde edilen test kümesinde bütün erkek ve bayan yetişkin konuşmacılara ait dosyalar bulunmaktadır. Daha sonra her bir gürültü için bir adet olmak üzere bu küme 4 alt kümeye bölünmüştür. Elde edilen 4 alt kümenin herbirine 20dB, 15dB, 10dB, 5dB, 0dB, -5dB olmak üzere 6 farklı işaret gürültü oranında gürültüler eklenmiştir. Böylece 28 farklı işaret gürültü oranı ve gürültü tipi kombinasyonu için 28028 adet dosya içeren bir test seti elde edilmiştir.

Hem test hem de eğitim sırasında temiz konuşma sinyallerine eklenen gürültüler NOIZEUS [38] veri setinden elde edilmiştir. Gürültü ekleme işleminden önce işaretlerin örnekleme hızı 8 kHz'e düşürülmüştür.

5.2 HTK ile Akustik Model Eğitimi ve Testler

Bu bölümde HTK ile eğitilen akustik modeller ve yapılan testler anlatılacaktır. İki farklı akustik model eğitilmiştir. Bunlardan ilki oluşturulan çok durumlu eğitim kümesinden eğitilen modeldir. İkincisi ise çok durumlu eğitim setindeki dosyaların temiz hallerini içeren eğitim setinden eğitilen modeldir.

Modellerin eğitiminde [5] çalışmasındaki SMM-GKM tabanlı model eğitilirken izlenen adımlar izlenmiştir. Eğitilen modelin temel özellikleri aşağıda verilmiştir:

- 16 SMM durumuna sahip kelime modelleri
- 3 SMM durumuna sahip bir sessizlik modeli
- 1 SMM durumuna sahip bir kelimeler arası duraklama modeli
- Kelime modelleri için durum başına 3 Gauss
- Sessizlik ve duraklama modelleri için durum başına 6 Gauss
- Diagonal kovaryans matrisi kullanan Gauss dağılımları
- 39 boyutlu MFKK vektörleri

Temiz konuşmalardan eğitilen modele ait sonuçlar Çizelge 5.2'de, çok durumlu eğitim kümesinden eğitilen modele ait sonuçlar ise Çizelge 5.3'te verilmiştir. Beklendiği gibi

temiz konuşma kümelerindeki testler hariç çok durumlu eğitim kümesinden eğitilen model genel olarak daha iyi sonuçlar vermiştir.

Çizelge 5.2: Çok durumlu eğitim kümesinden eğitilmiş model sonuçları.

İGO/dB	**Metro**	**Konuşma**	**Araba**	**Sergi Salonu**	**Ortalama**
clean	98.75	98.91	98.81	98.30	98.69
20	98.44	98.24	98.32	97.86	98.21
15	98.13	97.81	97.86	97.14	97.73
10	97.03	96.93	96.69	95.66	96.57
5	93.67	93.28	92.34	91.60	92.72
0	84.65	79.65	77.76	82.72	81.19
-5	59.47	48.51	42.26	59.41	52.41
0-20 arası ortalama	**94.38**	**93.18**	**92.59**	**92.99**	**93.28**

Çizelge 5.3: Temiz konuşmalar içeren eğitim kümesinden eğitilmiş model sonuçları.

İGO/dB	**Metro**	**Konuşma**	**Araba**	**Sergi Salonu**	**Ortalama**
clean	99.36	99.36	99.20	99.15	99.27
20	97.03	97.17	96.60	96.31	96.77
15	94.03	93.02	93.96	92.91	93.48
10	81.31	74.89	79.32	79.96	78.87
5	48.77	43.72	40.83	44.88	44.55
0	20.77	20.54	15.75	17.75	18.70
-5	10.18	6.38	7.71	6.45	7.68
0-20 arası ortalama	**68.38**	**65.86**	**65.29**	**66.36**	**66.47**

5.3 Seyrek Sınıflandırma Yöntemi Deneyleri

Seyrek sınıflandırma yöntemi ile konuşma tanıma yapmak için SMM-GKM sistemlerindeki gibi parametrik bir model gerekmez. Ancak eğitim kümesinden örneklemler seçilmeli ve SMM durumları ile etiketlenmelidir. Literatürde konuşma tanıma sırasında kullanılacak sözlüğü öğrenme tabanlı yöntemler [39] ile belirleme yoluna gidilmiştir. Ancak bu çalışmada örneklemlerin eğitim setinden rastgele seçilmesi yaklaşımı benimsenmiştir.

Öncelikle benzer bir veri kümesi [5] için belirlenmiş olan ve [14] çalışmasındaki sözlük elimizdeki test kümesi için kullanılmıştır. Bu durumda elde edilen sonuçlar Çizelge 5.4'te verilmiştir. Elde edilen sonuçlar SMM-GKM sistemine göre daha kötüdür. Bunun temel nedeni gürültü sözlüğü ile test verisine eklenen gürültüler arasındaki uyumsuzluktur. Çünkü daha düşük İGO değerlerindeki test

kümeleri karşılaştırıldığında başarımlar arasındaki fark açılmaktadır. Ayrıca gürültü uyuşmazlığı kadar önemli bir neden olmasa da temiz konuşmalara ait örneklem çeçrçevelerinin SMM durumları ile etiketlenmesi sırasında iyi bir akustik model kullanılırsa seslerin daha iyi tanınacağı düşünülmektedir.

Çizelge 5.4: Hazır sözlük ile yapılan seyrek sınıflandırma tabanlı konuşma tanıma sonuçları.

İGO/dB	**Metro**	**Konuşma**	**Araba**	**Sergi Salonu**	**Ortalama**
clean	91.8	92.01	90.9	91.69	91.60
20	86.94	89.51	87.75	88.45	88.16
15	82.75	86.50	86.00	86.19	85.36
10	75.96	79.73	79.72	80.37	78.94
5	64.40	63.65	66.13	69.01	65.79
0	45.14	37.19	41.87	49.91	43.57
-5	21.13	10.36	14.85	22.62	17.24
0-20 arası ortalama	**71.04**	**71.35**	**72.29**	**74.78**	**72.37**

Bu nedenle öncelikle eğitim setinde kullanılan gürültü işaretlerinden rastgele seçilen spektrogramlar yardımıyla bir gürültü sözlüğü oluşturulmuştur. Konuşma sözlüğü ise sabit bırakılmıştır. Yeni oluşturulan bu sözlük ile yapılan deneylerin sonuçları Çizelge 5.5'te verilmiştir. Çizelge 5.4 ve Çizelge 5.5 karşılaştırıldığında gürültü uyuşmazlığının sonuçlar üzerinde çok büyük bir etkisi olduğu görülmektedir. Gürültülü ve bazı temiz konuşma kümelerinde başarım artmıştır. Gürültülü kümeler için bu durum beklenen bir sonuçtur. Ancak temiz konuşmaya ait kümelerdeki başarım artışı, bir önceki deneyde temiz konuşmalar için uygun atomların seçiminde de bir miktar gürültü örnekleminin yer aldığını, dolayısıyla tanıma başarımını düşürdüğünü göstermektedir.

Çizelge 5.5: Sözlükteki gürültü örneklemlerinin güncellendiği durumdaki seyrek sınıflandırma tabanlı konuşma tanıma sonuçları.

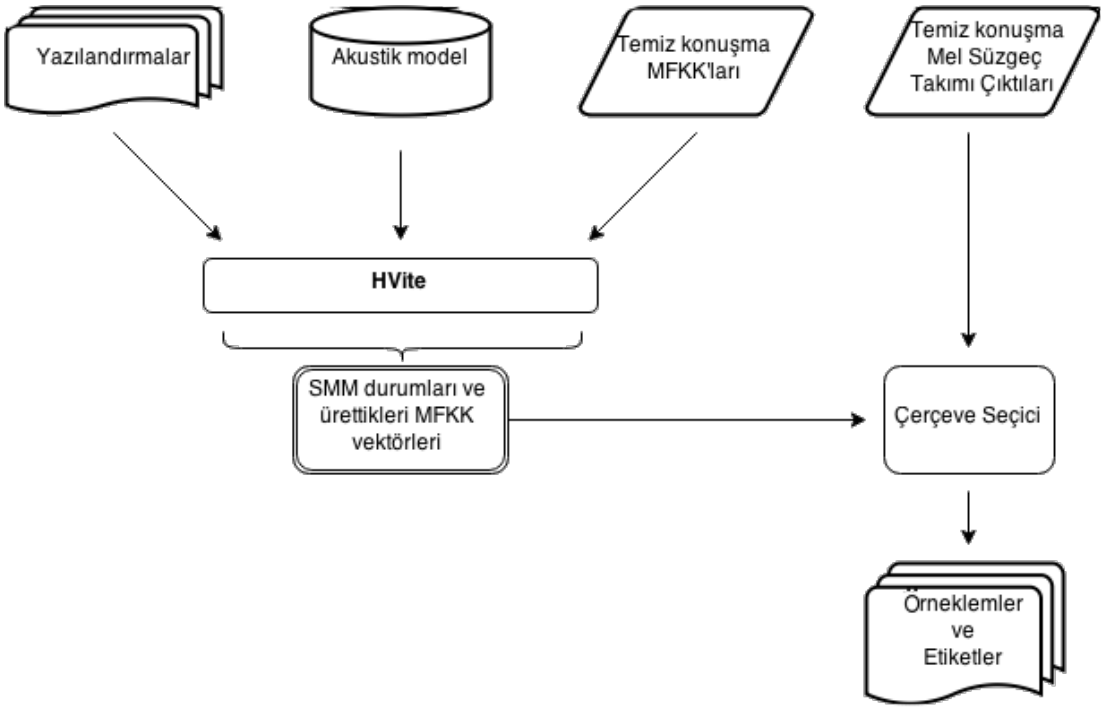
İGO/dB	**Metro**	**Konuşma**	**Araba**	**Sergi Salonu**	**Ortalama**
clean	91.93	91.76	90.90	91.82	91.60
20	88.38	89.57	88.02	88.61	88.65
15	85.54	87.99	86.40	86.34	86.57
10	81.16	84.19	82.33	81.97	82.41
5	75.11	77.23	75.80	74.86	75.75
0	64.34	61.95	62.66	64.00	63.24
-5	46.06	35.68	45.18	47.42	43.59
0-20 arası ortalama	**78.90**	**80.19**	**79.04**	**79.16**	**79.32**

```
"/home/fatih/thesis/data/clean-train-mfcc-8khz/snr_10db_subway/pk-woman-5b.rec"
0 100000 SIL[2] -48.436069 SIL
100000 500000 SIL[3] -210.992783
500000 600000 SIL[4] -55.182404
600000 700000 SIL[2] -56.503124 SIL
700000 1200000 SIL[3] -267.437897
1200000 1700000 SIL[4] -268.673370
1700000 2100000 FIVE[2] -240.203094 FIVE
2100000 2200000 FIVE[3] -62.622402
2200000 2400000 FIVE[4] -116.842125
2400000 2500000 FIVE[5] -55.865757
2500000 2600000 FIVE[6] -54.384102
2600000 2800000 FIVE[7] -109.025902
2800000 3200000 FIVE[8] -190.865982
3200000 4300000 FIVE[9] -434.426025
4300000 4600000 FIVE[10] -109.735695
4600000 5400000 FIVE[11] -377.123199
5400000 5900000 FIVE[12] -318.987610
5900000 6100000 FIVE[13] -140.777512
6100000 6300000 FIVE[14] -147.618561
6300000 6500000 FIVE[15] -174.874069
6500000 6600000 FIVE[16] -78.388786
6600000 8300000 FIVE[17] -1185.492798
8300000 8500000 SIL[2] -125.059502 SIL
8500000 8600000 SIL[3] -55.008450
8600000 8700000 SIL[4] -54.569492
8700000 8800000 SIL[2] -52.887501 SIL
8800000 9900000 SIL[3] -505.739685
9900000 10000000 SIL[4] -41.770767
```

Şekil 5.1: Örnek bir dosya için hizalama çıktısı.

Bu adımdan sonra sözlüğün konuşma örneklemlerini de güncellemek gerekmektedir. Ancak bunun için öncelikle tüm eğitim kümesinin zorlanmış hizalama işlemiyle SMM durumları ile etiketlenmesi gerekmektedir. Bunun için HTK çatısı altındaki HVite aracı kullanılmıştır. Bir zorlanmış hizalama çıktısı Şekil 5.1'de gösterilmiştir. Bu çıktıdaki her satır ilgili konuşmada bir SMM durumuna karşı düşen hizalamaya ait bilgileri içerir. Satırların ilk iki elemanı, satırın sonunda belirtilen SMM durumuna hizalanan çerçevelerin başlangıç ve bitiş zamanlarını belirtmektedir. Çerçeve genişliği sabit olduğundan herbir duruma kaç çerçevenin atandığı görülebilir: HTK zaman birimi olarak 100 nano saniye kullanmaktadır. Her bir çerçevenin 10 ms'ye karşı düştüğü göz önüne alınırsa, Şekil 5.1'de verilen çıktı daha iyi okunabilir. Bu çıktıya göre örneğin "FIVE" kelimesini modelleyen SMM'nin ilk durumuna 4 çerçeve atanmıştır.

Bu işlem sonucunda hizalanan çerçeveler 39 boyutlu MFKK vektörleridir. Ancak seyrek sınıflandırma yönteminde Mel Süzgeç takımı çıktısı kullanılmaktadır. Bu yüzden ilgili zaman aralığındaki Mel-Süzgeç takımı çıktısı hizalamadan elde edilen durumlara atanır. Bu işlem sonucunda örneklemler ve bu örneklemlere ait etiket matrisleri elde edilir. Bu prosedür için Şekil 5.2 incelenebilir.



Şekil 5.2: Hizalama işlemi ve sözlük çıkarımı.

Çizelge 5.6: Tamamı yeniden oluşturulmuş sözlükle yapılan seyrek sınıflandırma tabanlı konuşma tanıma sonuçları.

İGO/dB	**Metro**	**Konuşma**	**Araba**	**Sergi Salonu**	**Ortalama**
clean	96.33	95.99	95.50	96.13	95.99
20	94.74	95.17	94.36	94.93	94.80
15	93.21	94.29	93.60	93.58	93.67
10	90.28	91.88	91.64	89.93	90.93
5	85.38	85.87	88.36	84.77	86.09
0	76.70	74.16	81.26	76.90	77.25
-5	62.32	50.33	64.20	62.43	59.82
0-20 arası ortalama	**88.06**	**88.27**	**89.84**	**88.02**	**88.55**

Sonuç olarak elde edilen sözlük ile yapılan deneylerin sonuçları Çizelge 5.6 ile verilmiştir. Seyrek sınıflandırma yöntemi ile en iyi başarımlar bu testte elde edilmiştir. Test sonuçlarında en dikkat çekici noktalardan biri -5 dB İGO için bütün alt kümelerde seyrek sınıflandırmanın SMM-GKM sistemine göre daha iyi sonuç vermesidir.

5.4 Özet

Bu bölümde öncelikle, TIDIGITS veri kümesine gürültü eklenerek oluşturulan veri kümesi hakkında bilgi verilmiştir. Daha sonra HTK aracı kullanılarak temiz ve gürültülü konuşmalardan eğitilen akustik modeller ile ilgili bilgi verilmiş ve test sonuçları verilmiştir. Ayrıca Seyrek Sınıflandırma yöntemi kullanılarak gerçeklenen

konuşma tanıma sisteminin aynı test kümesi üzerindeki sonuçları da sunulmuştur. Seyrek sınıflandırma yöntemi için, örneklemlerin bulunduğu sözlüğün temiz konuşma kısmı zorlanmış hizalama yardımıyla elde edilmiştir.

6. SONUÇ ve ÖNERİLER

Bu çalışmada örneklem tabanlı bir konuşma tanıma sistemi gerçeklenmiştir. Bu yöntem gürültülü bir konuşmayı sözlükteki örneklemlerin seyrek doğrusal birleşimi şeklinde modellemektedir. Bu örneklemler konuşma işaretinden elde edilen çerçevelerin birleştirilmesiyle elde edilir. Genelde kullanılan örneklemlerin uzunluğu 50-300 ms arasında değişmektedir. Bu çalışmada 200 ms'lik örneklemler kullanılmıştır.

Seyrek sınıflandırma yöntemi yardımıyla melez bir konuşma tanıma sistemi gerçeklenmiştir. Bu sistemin melez olmasının nedeni durum geçişlerinde SMM yapısındaki geçiş olasılıklarını kullanırken akustik olabilirlikleri örneklem aktivasyonları yardımıyla modellemesidir. Sözlükteki örneklemlerin SMM durumları ile etiketlenmesi ile tanıma işlemi klasik Viterbi kod çözücüsü yardımıyla yapılabilmektedir.

Bunun yanında HTK kullanarak klasik SMM-GKM tabanlı iki akustik model eğitilmiştir. Bunlardan ilki çok durumlu eğitim kümesinden eğitilirken, ikincisi temiz konuşmalardaan eğitilmiştir. İlk model ikinci modele göre çok daha iyi başarım vermiştir. İkinci model sadece temiz konuşmaları tanırken daha iyi başarım vermektedir.

Temiz konuşmalardan eğitilen akustik model seyrek sınıflandırma yönteminde kullanılacak sözlüğü oluşturmak için zorlanmış hizalama işlemi sırasında kullanılmıştır. Son durumda elde edilen sözlük ile yapılan deneylerde seyrek sınıflandırma yöntemi gayet iyi sonuçlar vermiştir. Özellikle düşük İGO içeren test kümelerinde seyrek sınıflandırma yöntemi iyi sonuçlar vermektedir. Örneğin -5 dB İGO içeren kümelerde HTK ile eğitilen modellerden daha iyi sonuçlar elde edilmiştir.

Genel olarak seyrek sınıflandırma yönteminin yüksek İGO değerlerinde klasik SMM-GKM sistemleri kadar iyi çalışmadığı görülmüştür. Bunun için yapılabilecek iyileştirmeler üzerinde durulmuş ve önerilerde bulunulmuştur. Örneğin seyrek

sınıflandırma yönteminde konuşma işaretindeki değişimleri modellemek adına delta vektörleri gibi yaklaşımlar kullanılmamıştır. Klasik SMM-GKM sistemlerinde başarım arttıran bu yöntem yapılan örneklem tabanlı çalışmalarda da başarım arttırmıştır [40]. Ayrıca grup seyrekliğinin zorlanması [41] literatürde yapılan çalışmalarda başarım arttırmıştır. Ek olarak örneklemlerin büyüklüğü ve minimizasyon kuralı ile ilgili çalışmalar başarım arttırabilir. Genel olarak yapılan çalışmalarda 20 ardışık çerçevenin birleştirilmesiyle elde edilen örneklemler iyi sonuçlar vermiştir [14]. Bu çalışmada spektrogramlardaki enerjiler logaritmik değildir. Ancak bilindiği gibi logaritma alma işlemi özniteliklerin ses şiddeti gibi etkenlerden etkilenme miktarını azaltmaktadır. Dolayısıyla logaritmik ölçekte çalışabilecek bir güncelleme kuralı da başarım arttırabilir.

Bu çalışmanın sonucunda seyrek sınıflandırma yöntemi yardımıyla bir konuşma tanıma sistemi gerçeklenmiştir. Bu yöntem parametrik bir yöntem olmadığından bazı parametrelerin veri yetersizliğinden dolayı iyi öğrenilememesi gibi bir problem oluşturmamaktadır. Dahası çok esnek bir yapı sunmakta olan bu yöntem yardımıyla temiz konuşmalar kestirilip bu durum için yeni SMM-GKM sistemleri eğitilebilir. Çünkü yöntem içinde bir kaynak ayrıştırma işlemi de yapılmaktadır. Ayrıca tanıma işleminin Viterbi kod çözücüsü yardımıyla yapılması da esneklik sağlayan diğer bir noktadır. Bu çalışma ışığında örneklem tabanlı yöntemlerin gürbüz konuşma tanıma için efektif bir yol olduğu ancak üzerinde çalışılması gereken birçok noktayı da barındırdığı görülmüştür.

KAYNAKLAR

[1] **Davis, K.**, **Biddulph, R. ve Balashek, S.** (1952). Automatic recognition of spoken digits, *The Journal of the Acoustical Society of America*, ***24***(6), 637–642.

[2] **Levinson, S.E.**, **Rabiner, L.R. ve Sondhi, M.M.** (1983). An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition, *Bell System Technical Journal, The*, ***62***(4), 1035–1074.

[3] **Hinton, G.**, **Deng, L.**, **Yu, D.**, **Dahl, G.E.**, **Mohamed, A.r.**, **Jaitly, N.**, **Senior, A.**, **Vanhoucke, V.**, **Nguyen, P. ve Sainath, T.N.** (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, *Signal Processing Magazine, IEEE*, ***29***(6), 82–97.

[4] **Gemmeke, J.F.** (2011). *Noise Robust ASR: Missing data techniques and beyond,* Doktora Tezi, Radboud Universiteit, Nijmegen, Gelderland.

[5] **Pearce, D.**, **günter Hirsch, H. ve Gmbh, E.E.D.** (2000). The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions, *in ISCA ITRW ASR2000*, s.29–32.

[6] **Moreno, P.J.**, **Raj, B. ve Stern, R.M.** (1996). A vector Taylor series approach for environment-independent speech recognition, *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, cilt**2**, IEEE, s.733–736.

[7] **Gales, M. ve Young, S.** (1996). Robust continuous speech recognition using parallel model combination, *Speech and Audio Processing, IEEE Transactions on*, ***4***(5), 352–359.

[8] **Wright, J.**, **Yang, A.Y.**, **Ganesh, A.**, **Sastry, S.S. ve Ma, Y.** (2009). Robust face recognition via sparse representation, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, ***31***(2), 210–227.

[9] **Zhang, H.**, **Berg, A.C.**, **Maire, M. ve Malik, J.** (2006). SVM-KNN: Discriminative nearest neighbor classification for visual category recognition, *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, cilt**2**, IEEE, s.2126–2136.

[10] **Brunelli, R. ve Poggio, T.** (1993). Face recognition: features versus templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ***15***(10), 1042–1052.

[11] **Wold, E.**, **Blum, T.**, **Keislar, D. ve Wheaten, J.** (1996). Content-based classification, search, and retrieval of audio, *MultiMedia, IEEE*, ***3***(3), 27–36.

[12] **Li, S.Z.** (2000). Content-based audio classification and retrieval using the nearest feature line method, *Speech and Audio Processing, IEEE Transactions on*, ***8***(5), 619–625.

[13] **Plumbley, M.**, **Blumensath, T.**, **Daudet, L.**, **Gribonval, R. ve Davies, M.** (2010). Sparse Representations in Audio and Music: From Coding to Source Separation, *Proceedings of the IEEE*, ***98***(6), 995–1005.

[14] **Gemmeke, J.**, **Virtanen, T. ve Hurmalainen, A.** (2011). Exemplar-Based Sparse Representations for Noise Robust Automatic Speech Recognition, *Audio, Speech, and Language Processing, IEEE Transactions on*, ***19***(7), 2067–2080.

[15] **Jurafsky, D. ve Martin, J.H.** (2009). *Speech and Language Processing (2Nd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[16] **Rabiner, L.** (1989). A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, ***77***(2), 257–286.

[17] **Rosenfeld, R.** (2000). Two decades of statistical language modeling: where do we go from here?, *Proceedings of the IEEE*, ***88***(8), 1270–1278.

[18] **Jackson, J.E.** (2005). *A user's guide to principal components*, cilt587, John Wiley & Sons.

[19] **Mika, S.**, **Ratsch, G.**, **Weston, J.**, **Scholkopf, B. ve Muller, K.** (1999). Fisher discriminant analysis with kernels, *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, s.41–48.

[20] **Candès, E.J. ve Wakin, M.B.** (2008). An introduction to compressive sampling, *Signal Processing Magazine, IEEE*, ***25***(2), 21–30.

[21] **Sainath, T.**, **Ramabhadran, B.**, **Nahamoo, D.**, **Kanevsky, D.**, **Van Compernolle, D.**, **Demuynck, K.**, **Gemmeke, J.**, **Bellegarda, J. ve Sundaram, S.** (2012). Exemplar-Based Processing for Speech Recognition: An Overview, *Signal Processing Magazine, IEEE*, ***29***(6), 98–113.

[22] **Sainath, T.**, **Ramabhadran, B.**, **Picheny, M.**, **Nahamoo, D. ve Kanevsky, D.** (2011). Exemplar-Based Sparse Representation Features: From TIMIT to LVCSR, *Audio, Speech, and Language Processing, IEEE Transactions on*, ***19***(8), 2598–2613.

[23] **Gemmeke, J.F.**, **Cranen, B. ve Remes, U.** (2011). Sparse imputation for large vocabulary noise robust ASR, *Computer Speech & Language*, ***25***(2), 462–479.

[24] **Gemmeke, J.**, **ten Bosch, L.**, **Boves, L. ve Cranen, B.** (2009). Using sparse representations for exemplar based continuous digit recognition, *Proc. EUSIPCO*, Citeseer, s.24–28.

[25] **Sainath, T.**, **Nahamoo, D.**, **Kanevsky, D.**, **Ramabhadran, B. ve Shah, P.** (2011). A convex hull approach to sparse representations for exemplar-based speech recognition, *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, s.59–64.

[26] **Mallat, S.** (2008). *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*, Academic Press, 3. sürüm.

[27] **Elad, M.** (2010). *Sparse and redundant representations: from theory to applications in signal and image processing*, Springer.

[28] **Tibshirani, R.** (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

[29] **Sainath, T.**, **Nahamoo, D.**, **Ramabhadran, B.**, **Kanevsky, D.**, **Goel, V. ve Shah, P.** (2011). Exemplar-based Sparse Representation phone identification features, *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, s.4492–4495.

[30] **Deselaers, T.**, **Heigold, G. ve Ney, H.**, (2007), Speech Recognition With State-based Nearest Neighbour Classifiers.

[31] **Bourlard, H.**, **Hermansky, H. ve Morgan, N.** (1996). Towards increasing speech recognition error rates, *Speech communication*, ***18***(3), 205–231.

[32] **Gemmeke, J.F.**, **Van Hamme, H.**, **Cranen, B. ve Boves, L.** (2010). Compressive sensing for missing data imputation in noise robust speech recognition, *Selected Topics in Signal Processing, IEEE Journal of*, ***4***(2), 272–287.

[33] **Virtanen, T.** (2007). Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria, *Audio, Speech, and Language Processing, IEEE Transactions on*, ***15***(3), 1066–1074.

[34] **Lee, D.D. ve Seung, H.S.** (2001). Algorithms for non-negative matrix factorization, *Advances in neural information processing systems*, s.556–562.

[35] **Young, S. ve Young, S.** (1994). The HTK Hidden Markov Model Toolkit: Design and Philosophy, *Entropic Cambridge Research Laboratory, Ltd*, ***2***, 2–44.

[36] **Leonard, R.** (1984). A database for speaker-independent digit recognition, *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84.*, cilt**9**, s.328–331.

[37] **Hirsch, H.G.** (2005). F a NT-Filtering and Noise Adding Tool, Alındığı Tarih: 15.11.2014, Adres: http://aurora.hsnr.de/download.html.

[38] **Hu, Y. ve Loizou, P.** (2006). Subjective Comparison of Speech Enhancement Algorithms, *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, cilt 1, s.I–I.

[39] **Aharon, M.**, **Elad, M. ve Bruckstein, A.** (2006). K -SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, *Signal Processing, IEEE Transactions on*, ***54***(11), 4311–4322.

[40] **Van Segbroeck, M.** *ve diğerleri* (2009). Unsupervised learning of time-frequency patches as a noise-robust representation of speech, *Speech Communication*, ***51***(11), 1124–1138.

[41] **Majumdar, A. ve Ward, R.K.** (2009). Classification via group sparsity promoting regularization, *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, IEEE, s.861–864.

ÖZGEÇMİŞ

Ad Soyad	: Fatih AKTÜRK
Doğum Yeri ve Tarihi	: Kırklareli, 1990
Adres	: İstanbul, Maslak, İTÜ Ayazağa Kampüsü
E-Posta	: akturkf@itu.edu.tr
Lisans	: İTÜ Telekomünikasyon Mühendisliği
Y. Lisans	: İTÜ Telekomünikasyon Mühendisliği