T.C.
ADNAN MENDERES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI
2016-YL-054

SÖZCÜK VE HECE TABANLI KONUŞMA TANIMA SİSTEMLERİNİN KARŞILAŞTIRILMASI

Özlem YAKAR

Tez Danışmanı:
Yrd. Doç. Dr. Rıfat AŞLIYAN

AYDIN

T.C.
ADNAN MENDERES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE
AYDIN

Matematik Anabilim Dalı Yüksek Lisans Programı öğrencisi Özlem YAKAR tarafından hazırlanan "Sözcük ve Hece Tabanlı Konuşma Tanıma Sistemlerinin Karşılaştırılması" başlıklı tez, 23/09/2016 tarihinde yapılan savunma sonucunda aşağıda isimleri bulunan jüri üyelerince kabul edilmiştir.

	Ünvanı, Adı Soyadı	Kurumu	İmzası
Başkan:	Yrd. Doç. Dr. Rıfat AŞLIYAN	Adnan Menderes Üni.	……
Üye:	Yrd. Doç. Dr. Korhan GÜNEL	Adnan Menderes Üni.	……
Üye:	Yrd. Doç. Dr. Refet POLAT	Yaşar Üniversitesi	……

Jüri üyeleri tarafından kabul edilen bu Yüksek Lisans tezi, Enstitü Yönetim Kurulunun …… Sayılı kararıyla .../.../2016 tarihinde onaylanmıştır.

Prof. Dr. Aydın ÜNAY

Enstitü Müdürü

T.C.
ADNAN MENDERES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bu tezde sunulan tüm bilgi ve sonuçların, bilimsel yöntemlerle yürütülen gerçek deney ve gözlemler çerçevesinde tarafımdan elde edildiğini, çalışmada bana ait olmayan tüm veri, düşünce, sonuç ve bilgilere bilimsel etik kuralların gereği olarak eksiksiz şekilde uygun atıf yaptığımı ve kaynak göstererek belirttiğimi beyan ederim.

…/…/2016

Özlem YAKAR

ÖZET

SÖZCÜK VE HECE TABANLI KONUŞMA TANIMA SİSTEMLERİNİN KARŞILAŞTIRILMASI

Özlem YAKAR

Yüksek Lisans Tezi, Matematik Anabilim Dalı
Tez Danışmanı: Yrd. Doç. Dr. Rıfat AŞLIYAN
2016, 47 sayfa

Bu tezde hece ve sözcük tabanlı Türkçe konuşma tanıma sistemleri geliştirilerek karşılaştırılmıştır. Yapılan bu uygulamalar, orta ölçekli, ayrık ve kişiye bağımlı sistemlerdir. Bu sistemlerde, Dinamik Zaman Bükmesi (DZB), Destek Vektör Makinesi (DVM), Çok Katmanlı Algılayıcı (ÇKA) ve Saklı Markov Modeli (SMM) metotları kullanılarak eğitim ve test işlemleri yapılmıştır. SMM, ÇKA ve DVM metotlarıyla her hece ve sözcük için hece ve sözcük modelleri oluşturulmuştur. Bu modellere göre tanıma işlemi gerçekleştirilmiştir. Sistemler genel olarak önişleme, öznitelik çıkarma, hece ve sözcük eğitim ve tanıma safhalarından oluşmaktadır. Hece tabanlı sistemlerde artişleme işleminde uygulanmıştır. Önişleme safhasında ses sinyalleri düzleştirilir ve pencereleme işlemi yapılır. Sonra, sözcük ve hece sınırları belirlenir. Öznitelik çıkarma aşamasında, her bir sözcük ve hece için MFCC öznitelik vektörleri oluşturulur. Vektör olarak temsil edilen bu hece ve sözcükler SMM, ÇKA ve DVM metotlarıyla eğitildikten sonra tanıma işlemi yapılır. Hece tabanlı sistemlerde, artişleme yapılarak sistemlerin başarısı önemli ölçüde artırılmıştır. 200 Türkçe sözcükle yapılan test işleminde, hece tabanlı sistemlerdeki en iyi doğru tanıma oranları DZB için %94,2; ÇKA için %88; SMM için %82,6; DVM için ise %90,8 olmuştur. Sözcük tabanlı sistemlerde ise DZB için %96; ÇKA için %82,6; SMM için %89,4; DVM için ise %90,7 oranında doğru tanıma gerçekleştirildi.

Anahtar Kelimeler: Konuşma Tanıma, Türkçe Konuşma Tanıma, Sözcük ve Hece Tabanlı Konuşma Tanıma, SMM, ÇKA, DZB, DVM

ABSTRACT

A COMPARISON OF WORD AND SYLLABLE-BASED SPEECH RECOGNITION SYSTEMS

Özlem YAKAR

M. Sc. Thesis, Department of Mathematics
Supervisor: Assist. Prof. Dr. Rıfat AŞLIYAN
2016, 47 pages

In this thesis, word and syllable-based Turkish speech recognition systems developed and compared. The developed systems are discrete, middle-sized and user-dependent. In these systems, Dynamic Time Warping (DTW), MultiLayer Perceptron (MLP), Support Vector Machine (SVM) and Hidden Markov Model (HMM) methods are used in training and testing operations. Using HMM, SVM and MLP methods, word and syllable models are generated for every word and syllable. The recognition operation is applied with these models. The developed systems consist of preprocessing, feature extraction, word and syllable training, recognition and postprocessing operations. Postprocessing has been implemented for syllable-based systems because these approach can be applied for subwords as syllables or letters. In preprocessing, speech signals are flattened, and windowing has been made before the boundaries of the word and syllables are detected. In feature extraction phase, the vectors of Mel Frequency Cepstral Coefficient (MFCC) features which represent the utterances of word and syllables are constructed for each word and syllable. After these feature vectors are trained by HMM, MLP and SVM, the recognition operation of each word in test set is made to measure the systems success. In syllable-based systems, postprocessing is highly effective to increase the accuracy of the systems. In testing operation made with 200 Turkish words, the best accuracy rate of syllable-based systems are 94.2%, 88%, 82.6% and 90.8% for DTW, MLP, HMM and SVM respectively. But, in word-based systems, the accuracy rates for DTW, MLP, HMM and SVM are measured as 96%, 82.6%, 89.4% and 90.7% respectively.

Key Words: Speech Recognition, Turkish Speech Recognition, Word and Syllable-Based Speech Recognition, HMM, MLP, SVM, DTW

ÖNSÖZ

Hayatım boyunca beni en iyi şekilde yetiştirmiş, her şeyin en güzeline layık, tüm eğitim hayatım boyunca benden maddi ve manevi desteklerini esirgemeyen ve her zaman yanımda olan değerli ailem, İlyas YAKAR, Selma YAKAR ve Çilem YAKAR... Hepinize çok teşekkür ederim.

Yardımlarını unutamayacağım ve bende her zaman yerleri özel olan sevgili teyzem Sevinç TOPRAK ve rahmetli dayım Aydoğan KUTLU'ya, tezimin hazırlanması sırasında beni cesaretlendiren ve manevi destek sağlayan değerli dostlarım sevgili Simge ER ve Taylan ŞEN'e, çalışmam boyunca benden yardımlarını esirgemeyen değerli arkadaşlarım Keriman İŞÇİMEN ESER ve Gökşin GÜRBÜZ'e, tezim boyunca beni yürekten destekleyen ve her daim yanımda olan değerli yol arkadaşım Lütfü YAPICI'ya çok teşekkür ederim.

Tez çalışmam süresince her türlü yardımı ve fedakârlığı sağlayan, bilgi, tecrübe ve güler yüzü ile çalışmama ışık tutan, kullandığı her kelimenin hayatıma kattığı önemini asla unutmayacağım saygıdeğer hocam Yrd. Doç. Dr. Rıfat AŞLIYAN'a teşekkürlerimi bir borç bilirim. İyi ki varsınız…

Özlem YAKAR

İÇİNDEKİLER

KISALTMALAR DİZİNİ

SMM : Saklı Markov Modeli (Hidden Markov Model)

YSA : Yapay Sinir Ağları (Artificial Neural Network)

DZB : Dinamik Zaman Bükmesi (Dynamic Time Warping)

DVM : Destek Vektör Makinesi (Support Vector Machine)

FFT : Hızlı Fourier Dönüşümü (Fast Fourier Transform)

MFCC : Mel-Frekansı Kepstrum Katsayıları (Mel Frequency Cepstral Coefficients)

KTS : Konuşma Tanıma Sistemleri

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

EKLER DİZİNİ

1. GİRİŞ

Konuşma tanıma (Speech Recognition), sözcük ses sinyallerinin işlenerek metne dönüştürülmesi işlemidir (Vicens, 1969; Rabiner ve Juang, 1986; Rabiner ve Juang, 1993; Jelinek, 1998; Mengüşoğlu, 1999; Koç, 2002; McCowan vd., 2004; Yalçın, 2008; Aşlıyan, 2008). Konuşulan sözcük veya ifadeleri tanımlayarak, elektronik makineler tarafından okunabilecek bir formata dönüştüren konuşma tanıma işlemi, fare veya klavye gibi araçları kullanmadan, cihazların sesli komutlarla yönetilmesini sağlar.

Konuşma tanıma metotları, bünyesinde bulunan sözcük sayısına göre, az sayıda sözcüğe sahip olanlar küçük ölçekli ($0<x<200$, x:sözcük sayısı), orta sayıda sözcük içerenler orta ölçekli ($200\leq x<1000$) ve çok miktarda sözcük içerenler ise büyük ölçekli ($x>1000$) konuşma tanıma sistemleri olarak adlandırılırlar (Türk ve Arslan, 2004; Yalçın, 2008). Ayrıca, konuşma tanıma sistemleri konuşmacı sayısına bağlı olarak ikiye ayrılır. Kişiye bağımlı sistemlerde, sistem tek bir kişinin sesiyle eğitilir (Tunalı, 2005). Bu sistemlerin başarı oranı çoğunlukla yüksektir. Kişiden bağımsız (kişiye bağlı olmayan) sistemlerde ise çok sayıda kişinin sesini tanıma olanağı vardır (Rabiner vd., 1979). Bu sistemleri herkes çalıştırabileceği için, sınırsız sayıda insanın sesini sisteme tanıtmak gerekir. Dolayısıyla bu durum, konuşma tanımanın başarısını olumsuz etkiler. Konuşma tanıma alanında yapılan çalışmalar son yıllarda hız kazanmıştır. Konuşma tanıma çalışmalarının çoğunluğunda ses birimlerinden olan fonem tabanlı, hece tabanlı ve sözcük tabanlı birimler kullanılmıştır.

Konuşma tanımayı zorlaştıran bazı etmenlere bakıldığında; konuşmacının kişiden kişiye farklılaşması, kullanılacak kelimelerin çokluğu, konuşmaya dış çevredeki seslerin (gürültü, müzik vs.) karışması ya da iki konuşmanın birbiriyle örtüşmemesi, günlük konuşma ve aksan nedeniyle kelimelerin okunuşlarının çeşitlenmesi (örneğin "geleceğim" sözcüğünün "geleceğim, gelecem, gelicem, gelcem" şekillerinde okunması.) gibi etkenlerin olduğu görülür. Konuşma Tanıma Sistemleri (KTS), bu zorlukları en aza indirgeyerek tanımayı kolaylaştırmaya yardımcı olmaktadır. Günümüzde KTS, birçok farklı alanda kullanılmaktadır (Yılmaz, 1999; Burcu, 2007; Asefisaray, 2012). Genel olarak bir KTS'nin görevi, bir insana ait ses sinyalini alıp, bu sinyal üzerinde bir takım işlemler yapmak ve yapılan bu işlemlerin sonunda ise, ses sinyalinin hangi kelimelere karşılık geldiği

bulunup yazıya dökmektir (Aksoylar vd., 2009). Dolayısıyla KTS, girdi olarak bir konuşma sinyalini alıp, çıktı olarak metin üretmektedir (Nadas vd., 1988).

İnsan davranışlarını taklit edebilen makineleri icat etmek, geçmişten günümüze her zaman bütün uzmanların hayali olmuştur. İnsanın konuştuğu gibi konuşabilen robotları geliştirmek ise bu hayaller arasında en dikkat çekici olanıdır.

Konuşma tanıma teknolojileri, her geçen gün önemini korumaya devam etmektedir. Bu sayede hayatı hem kolaylaştırdığını, hem de zamandan tasarruf etmenin günümüzde oldukça önemli olduğunu bu teknolojiyle de görmekteyiz.

İlk konuşma tanıma sistemleri sadece rakamları anlayabilmekteydi. Bu alandaki bilim adamları bu rakamlardan oluşan sayılara odaklanmışlardı. 1952 yılında Bell Laboratuvarlarında geliştirilen "Audrey" adındaki konuşma tanıma sistemi, bir kez söylenen rakamları tanıyabiliyordu. 1962 yılındaki dünya fuarında IBM şirketinin "Shoebox" adlı konuşma tanıma makinesi, İngilizcedeki 16 sözcüğü tanıyabilmekteydi. 1950 ve 1960'larda ABD, İngiltere, Japonya ve Sovyetler Birliğindeki laboratuvarlarda sesli ve sessiz harfleri tanıyan donanım tabanlı konuşma tanıma sistemleri geliştirilmiştir.

ABD Savunma Bakanlığı'nın DARPA Konuşma Anlama Araştırma Programı (SUR) 1971'den 1976'ya kadar devam etmiş ve konuşma tanıma alanında büyük ilerlemeler sağlanmıştır. 1970'lerde Carnegie Mellon Üniversitesi'nin geliştirdiği "Harpy" adındaki konuşma anlama sistemi 1011 İngilizce sözcüğü anlayabiliyordu. Bu yıllarda konuşma tanıma alanında "Threshold Technology" ismiyle ilk ticari şirket kuruldu ve farklı insanların seslerini anlayabilen ilk ticari konuşma tanıma sistemi (VIP-100) piyasaya sürüldü.

1980'lerde konuşma tanıma sistemleri, bu zamana kadar birkaç yüz kelime tanırken, artık birkaç bin kelime tanıyabilecek kadar geliştirildi. Bu atılıma en büyük katkıyı sağlayan, yeni bir istatistiksel metot olan Saklı Markov Modelinin bulunmasıyla olmuştur. Konuşma tanımadaki tanınan sözcük sayısının artması, ticari uygulamaların (özellikle tıp, bankacılık ve oyuncak sektöründe) daha da artmasına sebep olmuştur.

1990'larda ise bilgisayar işlemcilerinin çok hızlı işlem yapmasıyla birlikte sıradan insanların bile konuşma tanıma yazılımlarını kullanmalarına olanak sağlandı. Dragon şirketi bu zamanlarda çok uygun fiyata ilk tüketici konuşma tanıma

sistemini (Dragon Dictate) piyasaya sürdü. Birkaç yıl sonra bu ürün iyileştirilerek "Dragon Naturally Speaking" adını aldı. Bu ürün, dakikada yüz kelimeyi tanıyabilecek şekilde tasarlanmış, sürekli konuşma tanıma sistemiydi. BellSouth şirketi ilk olarak "VAL" adında telefonla konuşarak uygulanan interaktif konuşma tanıma sistemi geliştirdi. Bu sistem, telefonla konuşarak istenilen menülere ulaşarak bilgi edinmeyi sağlıyordu.

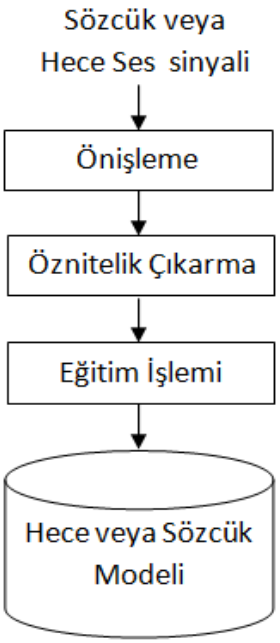
2000'li yıllara kadar konuşma tanıma sistemlerinin doğruluk tanıma oranı yaklaşık %80'e ulaşmıştı. 2000'li yıllarda ise hem bu oran daha da yükselmiş, hem de tanınan sözcük sayısı çok daha fazla artmıştır. İnternet ve mobil telefonlardaki çok büyük gelişmeler, konuşma tanıma sistemlerinin bu alanlara kaymasına yol açmıştır. Konuşma tanıma sistemlerin gelişmesinde Google'ın büyük katkıları olmuştur. İnternet ve cep telefonlarında sesli arama uygulamaları geliştirilmiş ve çok yaygınlaşmıştır. Bilgisayarlar, cep telefonları ve diğer akıllı cihazlar artık sesli komutlarla kontrol edilmeye başlanmıştır.

Günümüze kadar "konuşma tanıma ve anlamlandırma" ile ilgili yapılan çalışmalara bakıldığında çoğunlukla İngilizce dilinin dikkate alındığı görülmüştür. En çok geçerli dil olan İngilizce'nin seçilmesi, Türkçe ve benzeri farklı yapılara sahip dillerin, konuşma tanıma teknolojisi içindeki kullanımını büyük oranda olumsuz etkilemekteydi. Fakat son yıllarda, konuşma tanıma alanında çalışanlar, Türkçe'nin de önemini fark ettiğinden, Türkçe konuşma tanıma çalışmalarına da (Baygün, 2006; Aksoylar vd., 2009) ağırlık vermişlerdir.

Bu tezde, sözcük ve hece tabanlı sistemler karşılaştırılmıştır. Hece tabanlı konuşma tanıma sistemlerinde en küçük birim olarak Türkçe heceler seçilmiştir. Türkçe sondan eklemeli bir dil olduğundan, bir sözcüğe birden fazla ek getirilerek onlarca yeni sözcük türetilebilmektedir. Dolayısıyla her sözcüğü modellemek zor olacağından, konuşma tanıma metotlarından orta ölçekli konuşma tanıma sistemi kullanılarak, sözcükler hecelere ayrılmış ve dört farklı metot kullanılarak uygulamalar gerçekleştirilmiştir. Uygulamada kullanılan metotlar, Saklı Markov Modeli, Çok Katmanlı Algılayıcı, Dinamik Zaman Bükmesi ve Destek Vektör Makinesi yöntemleridir (Rabiner ve Juang, 1986; Rabiner, 1989; Charniak, 1993; Fraser ve Dimitriadis, 1994). Elde edilen veriler ile ele alınan konuşma tanıma uygulaması, her bir metot için uygulanmış ve elde edilen sonuçlar karşılaştırılmıştır.

Şekil 1.1'de görüldüğü gibi her bir sözcüğe ve heceye ait ses sinyallerini eğiterek sözcük ve hece ses modelleri oluşturulur. Sözcük ve hece model oluşturma işlemi önişleme, öznitelik çıkarma ve eğitim işlemi olmak üzere üç aşamadan meydana gelmektedir. Önişlemede ses sinyalleri ilk olarak düzleştirilir ve pencereleme işlemi yapılır. Sonra, vektör şeklindeki ses sinyalleri, 10 ms'lik örtüşme ile 20 ms'lik bloklara ayrılır. Öznitelik çıkarmada ise konuşma tanımada en etkili öznitelik metotlarından MFCC (Mel Frequency Cepstral Coefficients) kullanılarak, her blok 10 tane MFCC özniteliklerine dönüştürülür.

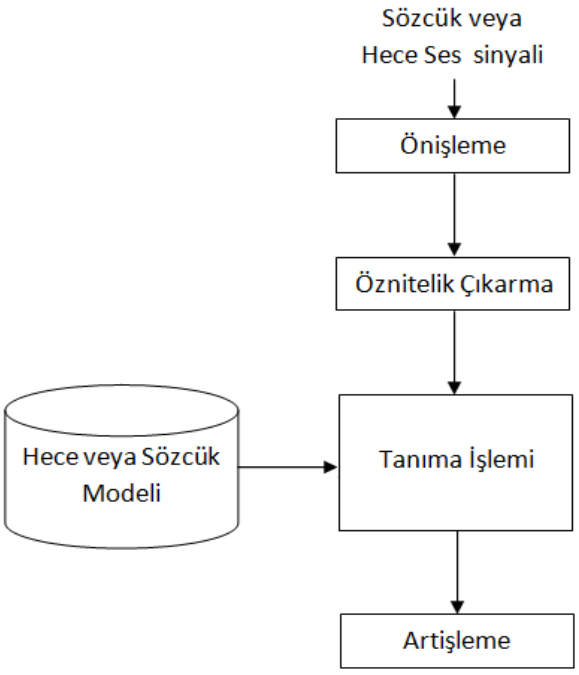
Dinamik Zaman Bükmesi (DZB, Dynamic Time Warping) metodunda, eğitim aşaması olmadığından sözcük ve hece modeli oluşturulmaz. DZB ile test veri setindeki her farklı boyuttaki sözcük ve hece, eğitim setindeki sözcük ve hece ile benzerlikleri hesaplanarak sözcük ve hece tespit edilir. Eğitim aşamasında, tanıma sürecinde kullanmak üzere SMM, DVM ve ÇKA metotları ile her sözcük ve hece için modeller oluşturulur.



Şekil 1.1. Sözcük ve hece ses sinyallerinin eğitilerek model oluşturulması

Şekil 1.2'de konuşma tanıma sisteminin genel yapısı görülmektedir. Önişleme ve öznitelik çıkarma işlemi, Şekil 1.1'deki veri modeli oluşturmadaki önişleme ve öznitelik çıkarma işlemiyle aynıdır. Tanıma işleminde ise, test veri setindeki

sözcük ve hece ses sinyallerinin MFCC öznitelik matrisi ile yukarda belirtilen metotlara ait sözcük ve hece modelleri ile benzerlikleri ölçülür ve en çok benzeyen sözcük ve hece sistem tarafından tanınmış olacaktır. Hece tabanlı sistemlerde artişlemede yapılmaktadır. Artişleme, hece tabanlı sistemlerde tanıma başarısını oldukça artırmaktadır. Artişlemede, sözcük ses sinyallerindeki her heceye en çok benzeyen beş hece bulunduktan sonra en çok benzeyen heceler eklenerek olası sözcükler bulunur. Eğer, bulunan sözcük Türkçe ise, tanınan sözcük olarak kabul edilir.



Şekil 1.2. Konuşma tanıma sisteminin genel yapısı

Tezin ikinci bölümünde, kullanılan sınıflandırma metotları ile MFCC öznitelik çıkarma metodu, Artişleme algoritması, Saklı Markov Modeli ve algoritmaları, Yapay Sinir Ağları, Dinamik zaman Bükmesi ve Destek Vektör Makinesi metotları ayrıntılı bir şekilde açıklanmaktadır. Konuşma tanıma sistemlerinden elde edilen bulgular ve sonuçlar üçüncü bölümde karşılaştırılarak verilmiştir. Son bölümde ise geliştirilen sistemlerden genel olarak ortaya çıkan verileri değerlendirerek kullanılan metotlara göre karşılaştırma yapılmıştır.

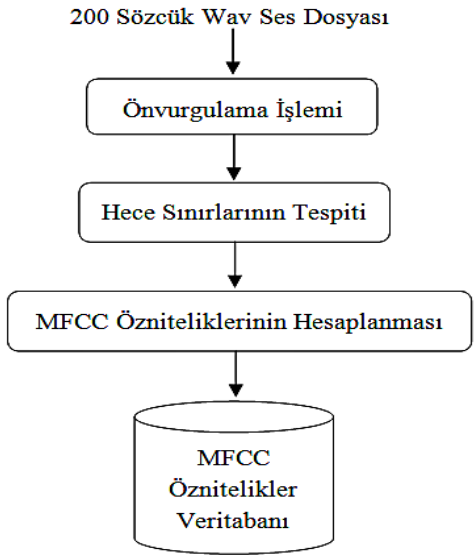
2. MATERYAL VE YÖNTEM

Bu tezde öznitelik çıkarımında MFCC (Mel Frequency Cepstral Coefficients) kullanılmıştır. MFCC özniteliklerinin nasıl elde edildiği aşağıda özet olarak ifade edilmektedir. Türkçe konuşma tanıma uygulamalarının gerçekleştirilmesinde Saklı Markov Modeli, Yapay Sinir Ağlarından Çok Katmanlı Algılayıcı (ÇKA), Destek Vektör Makinesi ve Dinamik Zaman Bükmesi metotlarıyla eğitim ve test işlemleri yapılmıştır.

2.1. MFCC Öznitelik Çıkarma Metodu

Öznitelik çıkarma, işlenmemiş bir verinin sınıflar arasındaki farklı özelliklerini belirleyip, sınıf içindeki farklılığı azaltacak nitelikteki bilgilere ulaşma işlemidir (Çapar vd., 2002). Öznitelik çıkarmada, örüntü tanıma, istatistiksel işaret işleme ve sınıflandırma yapmak amacıyla, alınan ölçümlerin bazı dönüşümlerinde daha özlü, daha az gürültülü ve daha az sayıda ayırt edici değerlere dönüştürülmesi işlemi yapılır.

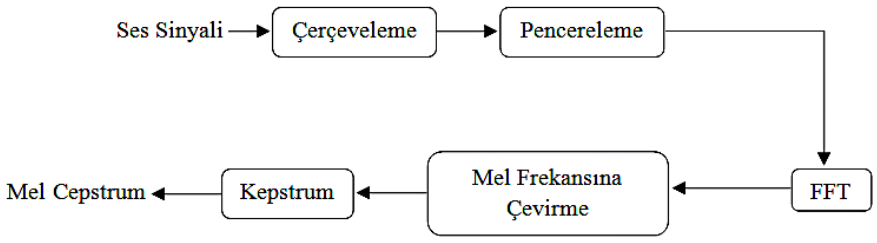
Konuşma tanıma sistemleri, Öznitelik Vektörlerini çıkarma ve Modelleme olarak iki kısma ayrılır (Karasartova, 2011). Konuşma tanıma işleminde, hangi özniteliklerin hesaplanacağı, yani öznitelik vektörlerin ne olacağı çok kritik bir faktördür. MFCC, konuşma tanıma sistemlerinin başarısını etkileyen en önemli metotlardan biridir. Öznitelik çıkarma safhasında sırasıyla şu işlemler yapılır: Önvurgulama metoduyla filtreleme yapılır. Sonrasında, 20 ms zaman aralığında ve 10 ms örtüşme ile çerçeveleme işlemi, çerçevelere de Hamming penceresiyle pencereleme yapılır. Karşılıklı ilinti işlemiyle özilinti vektörü hesaplanır (Rabiner ve Juang, 1993; Proakis ve Manolakis, 1996). Levinson metoduyla, Kepstrum öznitelik değerleri bulunur (Rabiner ve Juang, 1993). Her bir çerçeveye 10 boyutlu MFCC değeri denk gelecek şekilde, sözcükler ya da heceler, 10xN boyutlu (N:çerçeve sayısı) öznitelik matrisinden oluşur. Böylece, eğitim setindeki sözcük ses veri setinden MFCC sözcük ve hece öznitelik matris veri seti oluşturulur (Aşlıyan, 2008).



Şekil 2.1. Öznitelik veri setinin oluşturulması

Öznitelik çıkarma metotlarından birisi olan MFCC Katsayıları, algı temelli sesi temsil eden katsayılardır ve yaygın otomatik konuşma (konuşmacı) tanımada kullanılan bir özelliktir (Moralı ve Aygün, 2007). Genellikle Fourier Dönüşümü veya Ayrık Kosinüs Dönüşümünden elde edilir. Öznitelik çıkarma yöntemlerinde, en sık kullanılan metotlara örnek olarak MFCC verilebilir.

MFCC özniteliklerin elde edilmesi, Fourier Dönüşümü (Fourier Transform) temelli bir işlemdir. Seslerin insan kulağında nasıl işlemden geçip algılanıyorsa MFCC değerleri de buna benzetilerek tespit edilmektedir. Aynı sözcüklerin seslendirilirken oluşan değişimlerden çok fazla etkilenmemesi önemli avantajlarındandır. Aşağıdaki şekilde, MFCC katsayılarının elde edilmesi algoritmasının blok diyagramı gösterilmektedir (Karasartova, 2011).



Şekil 2.2. MFCC katsayılarının elde edilmesi

2.2. Artişleme Algoritması

K: Veri setindeki sözcüklerin en çok kaç heceden oluştuğunu belirtir.

$S_t(k)$: Sözcükteki *t*'inci hecenin *k*'ıncı sırada benzer olan heceyi ifade eder.

1. k = 1, 2, ... , ϕ ve k: Kaçıncı sırada benzediğinin gösterir.

Uygulamalarımızda $\phi = 10$ olarak alınmıştır. $S_1(k)$, $S_2(k)$, ... , $S_K(k)$ gibi hecelerin birleştirilmesiyle sözcükler elde edilir. 10^K ($K \epsilon \mathbb{Z}^+$) tane sözcük oluşturulur.

2. Elde edilen sözcüğe seviye atanır. Birinci adımda, sözcüklerin hecelerinin sırası toplanır ve bu toplam değer o sözcük için seviye olarak atanır.

3. Elde edilen sözcükler seviyelerine göre küçükten büyüğe göre sıralama işlemi yapılır.

4. Sıralanmış sözcüklerin en başından başlayarak teker teker sözcüklerin Türkçe olup olmadığına bakılır. Eğer Türkçe ise tanınan sözcük bu sözcük olacaktır. Eğer Türkçe değilse bir sonraki sözcük kontrol edilir. Eğer hiçbir sözcük Türkçe olarak bulunamadıysa tanıma işlemi başarısız olur (Aşlıyan ve Günel, 2009).

Artişlemenin nasıl çalıştığını ifade eden artişleme algoritması Bölüm 2.2'de verilmiştir. Fakat, kolay anlaşılması için bu algoritmanın çalışma mantığını bir örnekle açıklayabiliriz. Çizelge 2.1'de "kalemlik" sözcüğünün her hecesinin en çok benzeyen sıralı beş hecesi görülmektedir. Bu çizelgede en iyi tanınan "ka", "lim" ve "lik" heceleridir. Bu heceler yan yana eklendiğinde "kalimlik" sözcüğü oluşmaktadır. Fakat, "kalimlik" sözcüğü Türkçe bir sözcük olmadığından tanınan sözcük olarak alınmaz. Benzer şekilde "kat", "lim", "lik" heceleri birleştirilerek "katlimlik" sözcüğü elde edilir. Bu sözcükte Türkçe değildir. Bu şekilde devam edildiğinde sıra "ka", "lem" ve "lik" hecelerine gelecektir. Bu heceler birleştirildiğinde "kalemlik" sözcüğünü buluruz. Bu sözcük Türkçe olduğundan sistem tarafından tanınan sözcük "kalemlik" olacaktır.

Çizelge 2.1. "Kalemlik" sözcüğünün tanınan ilk beş hecesi

1. Hece	2. Hece	3. Hece
ka	lim	lik
kat	lem	lak
kit	lam	ler
ka	lim	lik
ke	lem	lık

SMM için seçilen parametreler: Durum sayısı: 40, Mixture= 1, Maksimum iterasyon sayısı: 300.

MLP için seçilen parametreler: Öğrenme oranı: 0,02, Moment katsayısı: 0,9, Maksimum devir sayısı (epoch): 100000, Hedef: 6e-3, Gizli katman sayısı: 2, Gizli katmanlardaki nöron sayıları sırasıyla: 30 ve 10, Aktivasyon fonksiyonu: logsig.

DVM için seçilen parametreler: Alfa toleransı: 1e-2, kkt toleransı: 5e-2, Kernel: Radyal Tabanlı Fonksiyon (Radial basis function).

2.3. Saklı Markov Modeli

Saklı Markov Modeli (SMM), ayrık veriler üzerinde çalışan, durum (state) ve durumlar arasında geçiş olasılıklarına göre tasarlanan ve zamana bağlı olarak oluşturulan istatistiksel bir modelleme metodudur (Baum ve Eagon, 1967; Rabiner, 1989; Özcan, 2015; Yakar ve Aşlıyan, 2016). Konuşma kavramı genel olarak, gürültüde gözlenen bir Markov zinciridir. SMM üzerindeki ilk çalışmalar, 1940'larda başlamış ve o yıllarda teori halinde olan SMM, tam olarak geliştirilemediği için uygulaması yapılamamıştır. L.P. Neuwirth tarafından icat edilen adı "*Saklı Markov Modeli*" olan Saklı Markov modelinin teorisi, 1960'lı yıllarda sinyal işleme amacıyla kullanılmış ve daha sonra bu modelin teorisine 1970'li yıllarda Baum (Baum vd., 1970; Baum, 1972), Eagon (Baum ve Eagon, 1967), Petrie (Baum ve Petrie, 1966) tarafından üzerinde çalışarak geliştirilmiştir.

Saklı Markov modelleri büyük ölçüde başarılı bir konuşma tanıma (Rabiner, 1989) uygulaması ile aynı zamanda el yazısı tanıma, jest tanıma, müzik ve biyoinformatik alanında da uygulanarak, kendi alanında popülerlik kazanmıştır (Sewell, 2008).

Saklı Markov Modeli, sinyal işlemede sınıflandırıcı (classifier) olarak kullanılan, özellikle el yazısı tanıma (Çapar vd., 2002), ses tanıma (Aydın, 2005; Baygün, 2006), vücut hareketlerini tanıma, müzik notasyonlarının takibinde, biyoinformatik (biyolojik bilgi), gen tahmini, kripto-analiz gibi, pek çok bilim ve mühendislik alanlarında kullanılır. SMM, istatistiksel bir yöntem olup, sınıflandırmada ve eldeki verilere göre tahmin yapmada en çok kullanılan modellerden birisidir.

Diğer yandan SMM'yi kullanan diğer uygulamalar (Rabiner ve Juang 1986; Poritz, 1988; Rabiner,1989), **Doğal Dil İşleme** (*Natural Language Processing - NLP*) alanında, Konuşma Etiketleme (*Part-of-speech tagging*), Kelime Bölümleme (*Word segmentation*), Bilgi Çıkarma (*Information extraction*), Optik Karakter Tanıma (*Optical Character Recognition - OCR*); **Konuşma Tanıma** (*Speech Recognition*) alanında, Akustik Modelleme (*Modeling acoustics*); **Bilgisayar Görme** (*Computer Vision*) alanında, Hareket Tanıma (*Gesture Recognition*); **Biyoloji alanında,** Gen Bulma (*Gene finding*), Protein Yapı Tahmini (*Protein structure prediction*); Ekonomi, İklim Bilimi (*Climatology*), İletişim (*Communications*) ve Robotik alanlarında da kullanımı mevcuttur.

SMM'nin girdisi, vektör olarak temsil edilen zamana bağlı ayrık verilerden oluşan bir dizidir. SMM, her birinin olasılık dağılımlarıyla bağlı olan sonlu durumlardan (state) oluşmaktadır**.** Durumlar arasındaki geçişler, geçiş olasılık (transition probability) değerleriyle hesaplanır. Durum içindeki bir gözlem veya sonuç, ona bağlı olan olasılık dağılımlarından elde edilmektedir. Durumlar, dışarıdaki gözlemcilere görünür değildir. Bu sebepten **Saklı** (Hidden) sözcüğü Saklı Markov Modeli metodunda bulunmaktadır (Yakar ve Aşlıyan, 2016).

SMM metodunu tanımlamak için aşağıdaki değişkenlere ihtiyaç vardır:

N: Modeldeki durum sayısı.

M: Alfabedeki gözlem sembollerinin sayısı. Eğer gözlemler sürekli ise *M* sonsuz olacaktır.

A: Denklem 2.1'de görüldüğü gibi geçiş olasılıkları.

$$A = \{a_{ij} \mid a_{ij} = p(q_{t+1} = j \mid q_t = i),\ 1 \leq i,j \leq N\} \quad (2.1)$$

Burada, *i*, *j*: durumları, a_{ij}: *i*. durumdan *j*. duruma geçiş olasılıklarını, *t*: durumların oluş zamanını, q_t: şimdiki durumu, q_{t+1}: bir sonraki durumu, *p*: durumların olasılık dağılımlarını (bir sonraki durumun şimdiki duruma oranını) temsil etmektedir. Geçiş olasılıkları, Denklem 2.2 ve 2.3'teki normal olasılıksal kısıtları sağlar.

$$a_{ij} \geq 0 \quad (2.2)$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad (2.3)$$

Denklem 2.4'te, durumların olasılık dağılımları ifade edilir.

$$B = \{b_j(k) \mid b_j(k) = p(o_t = v_k \mid q_t = j,\ 1 \leq k \leq M, 1 \leq j \leq N\} \quad (2.4)$$

Burada, B: geçiş olasılığını, $b_j(k)$: *k*. gözlemdeki *j*. durum geçiş olasılığını, v_k: alfabedeki *k*. gözlem sembolünü ve o_t: şimdiki parametre vektörünü ifade etmektedir. Denklem 2.5 ve 2.6'daki olasılıksal kısıtlar sağlanmalıdır.

$$b_j(k) \geq 0 \quad (2.5)$$

$$\sum_{k=1}^{M} b_j(k) = 1 \quad (2.6)$$

Eğer gözlemler sürekli ise ayrık olasılık yerine olasılık yoğunluk işlevini kullanmak zorunda olacağız. Bu durumda olasılık yoğunluk işlevinin parametrelerini belirlememiz gerekir. Genelde, Denklem 2.7'de görüldüğü üzere, olasılık yoğunluğu *M* Gaus dağılımlarının, Ω ağırlıklarının toplamına yaklaştırılır. C_{jm}, aşağıdaki olasılıksak kısıtları sağlamak zorundadır.

$$b_j(o_t) = \sum_{m=1}^{M} c_{jm} \Omega(\mu_{\mathrm{jm}}, \sum \mathrm{jm}, o_t) \quad (2.7)$$

c_{jm}: Ağırlık katsayıları

μ_{jm}: Ortalama vektörleri

$\sum_{jm}$: Ortak değişinti matrisleri

$$c_{jm} \geq 0,\ 1 \leq m \leq M,\ 1 \leq j \leq N \tag{2.8}$$

$$\sum_{m=1}^{M} c_{jm} = 1 \tag{2.9}$$

Aşağıdaki denklemlerde başlangıç durum dağılımları verilmiştir.

$$\pi = \{\pi_i \mid \pi_i = p(q_1 = i)\} \tag{2.10}$$

Kompakt notasyon kullanmak istersek, Denklem 2.11 ve 2.12'de görüldüğü gibi sürekli yoğunluklar kullanılarak, olasılık dağılımlı SMM'yi ifade edebiliriz.

$$\lambda = (A, B, \pi) \tag{2.11}$$

$$\lambda = (A, c_{jm}, \mu_{jm}, \sum jm\,, \pi) \tag{2.12}$$

SMM, durum ve geçiş olasılıklarının bütünü şeklinde düşünülebilir. SMM'nin yapısı karmaşık bir model olmasının yanında, aşağıda Şekil 2.3'te, basit bir SMM gösterilmiş ve olasılık parametreleri verilmiştir.

R: Başlangıç durum modeli

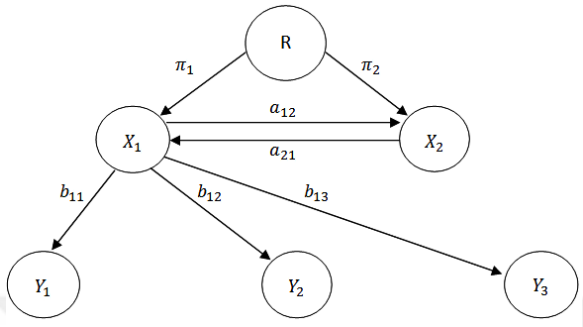
X=(X_i): Durumlar

Y=(Y_i): Olası gözlemler

$\pi = (\pi_i)$: Başlangıç durumlar

A=(a_{ij}): Geçiş olasılıkları durumu

B=(b_{ij}): Çıkış olasılıkları



Şekil 2.3. Saklı Markov Model örneği

Şekil 2.3'te, R durumundan X_1'e ve X_2'ye geçişlerdeki başlangıç olasılıkları sırasıyla π_1 ve π_2 olarak verilmiştir (Özcan, 2015). Yukardaki bu model için 2 tane başlangıç olasılığı bulunmaktadır. Bu iki olasılığın toplamı 1 olacaktır. Genellersek, *N* başlangıç olasılığı toplamı,

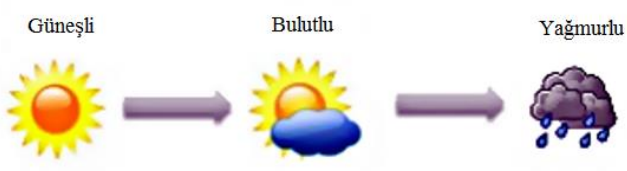
$$\pi_1 + \pi_2 + \cdots + \pi_N = 1 \qquad (2.13)$$

olur (Özcan, 2015). Ayrıca Y_1, Y_2, Y_3 olası gözlemler kümesini ve X_1, X_2 ise durumlar kümesini oluşturur. İlâveten, a_{ij}: state geçiş matrisini, b_{ij}: çıkış olasılık matrisini gösterir.

Tüm zaman serisi uygulamalarında Saklı Markov Modelleri görülebilmektedir. Örneğin; mevsimin yaz olması durumunda gün sıcaklığının önemi, depremlerin gece saatlerinde olma ihtimali, kişilerin kansere yakalanma riski gibi.

Saklı Markov Model'ine bir örnek olarak (Rabiner, 1989), "Hava durumunun tahmin edilmesinde SMM'yi nasıl kullanırız?" sorusuna cevap olarak, gözlemlenebilir hava durumu tahmin diyagramını aşağıdaki gibi oluşturalım:

Güneşli, bulutlu ve yağmurlu olmak üzere üç hava durumunu ele alalım. Her bir durum, bir önceki duruma bağlıdır. Burada, sırasıyla hava durumlarına bakacak olursak tek bir gün alınır.

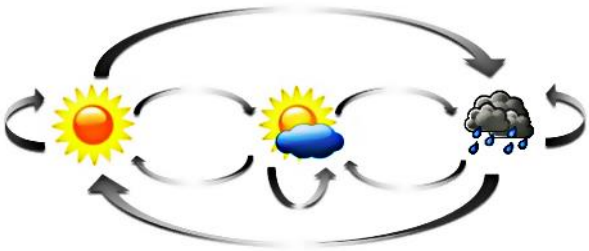


Şekil 2.4. Hava durumu tahmini

Şekil 2.4'te görüldüğü gibi, hava durumu sırasıyla güneşli, bulutlu ve yağmurlu şeklinde görünmektedir. Hava durumu her zaman bu şekilde sırasıyla olmayabilir. Yani, Şekil 2.5'te gösterildiği gibi hava durumu sırası, güneşli bir günün ardından yine güneşli bir gün olacak şekilde de ilerleyebilir. Bu döngü, genellikle kendi içinde geçiş durumu olarak adlandırılır ve bu da olası bir başka durumdur.

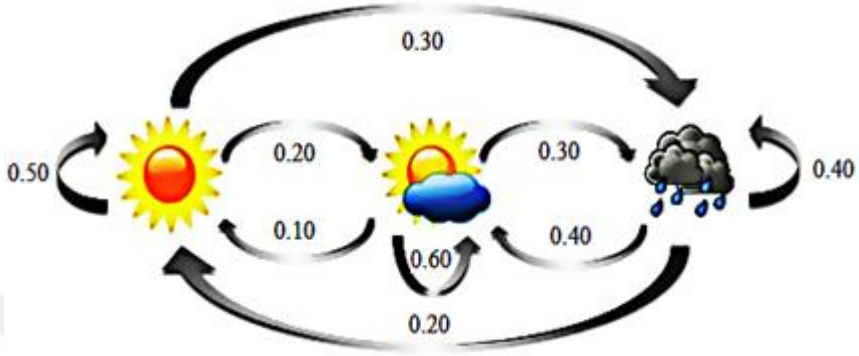
Şekil 2.5. Bir hava olayının kendi içindeki geçiş durumu

Şekil 2.6'da üç hava durumunun olası tüm durumlarını gözlemleyebiliriz.



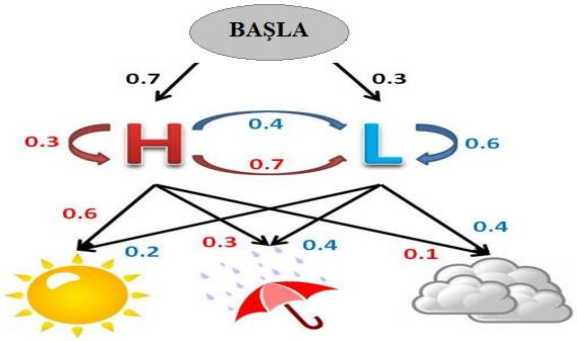
Şekil 2.6. Hava olaylarının olası tüm durumları

Şekil 2.7'de görüldüğü üzere gözlemlere dayalı çeşitli hava olayları için olasılık değerlerini de bulabiliriz. Örneğin, güneşli bir günün ardından güneşli bir gün olma olasılığı **0.50** ihtimalidir.



Şekil 2.7. Tüm hava olaylarının kendi içindeki olasılık değerleri

Ayrıca, aşağıdaki şekilde gösterildiği gibi tüm durum olasılıklarının toplamı da 1'e eşittir.



Şekil 2.8. Hava olaylarının tüm durum olasılıkları

Bulunan değerler bir matris yardımıyla Çizelge 2.2'de gösterilmiştir. Bu matris, aynı zamanda bir geçiş matrisi olarak da adlandırılır.

Çizelge 2.2. Hava olaylarının olasılıksal geçiş matrisi

		Hava Durumu (Bugün)		
		Güneşli	Bulutlu	Yağmurlu
Hava Durumu (Dün)	Güneşli	0.50	0.20	0.30
	Bulutlu	0.10	0.60	0.30
	Yağmurlu	0.20	0.40	0.40

2.3.1. Saklı Markov Modeli Algoritmaları

Saklı Markov Model'indeki durum dizisini bulabilmek için, üç önemli algoritma kullanılır.

2.3.1.1. İleri Yön Algoritması (Forward Algorithm)

Modelde bulunan durumların (state) sırasını bulmada kullanılan bir algoritmadır. Ortaya çıkabilecek tüm durumlar hesaplanır ve bu durumların baştan sona doğru olan olasılıkları toplanır.

2.3.1.2. Viterbi Algoritması (Viterbi Algorithm)

Viterbi algoritmasında, tüm durum sıralarının ses vektörleriyle denk olacak en uygun durum dizileri seçilir. Algoritmadaki amaç, olasılığı maksimize edecek durum dizisini bulabilmektir. Böylece daha doğru sonuçlar elde edilir.

2.3.1.3. Baum-Welch Algoritması (Forward-Backward Algorithm)

Gözlem dizisinde, gözlemlenme olasılığının maksimum olması için oluşturulmuş bir algoritmadır. Gözlem dizisini baştan sona ve tekrar sondan başa doğru geçerek gözlem olasılıklarını hesaplar. Böylece sonuçların doğruluk değerleri daha kesin bulunmuş olur.

2.4. Yapay Sinir Ağları

Yapay Sinir Ağları (YSA), bilgisayar bilimlerinde önemli yere sahip olan Yapay Zeka'nın alt dallarından biridir (Öztemel, 2012). Yapay sinir ağlarının nörobiyoloji alanında insanların beyne ilgi duyması ve elde edilen bu verileri bilgisayar bilimine uygulamak istemesiyle ortaya çıkmıştır.

YSA, insan beyninden yola çıkarak, beynin çalışma ve düşünebilme yeteneğinden esinlenerek oluşturulan bir bilgi işlem mekanizmasıdır. Biyolojik sistemlerdeki sinir ağlarına benzer şekilde öğrenme algoritmasına sahip olan, insan beyninden esinlenerek oluşturulmuş, nöronların birbirine sinaptik bağlantılarla bağlanmaları sonucu oluşturulan bir düşünme yapısıdır. Bu şekilde oluşturulmuş ağ ile öğrenme ve bilgi işleme olayı gerçekleşir. Bu nedenle, yapay sinir ağlarını, programlanması çok zor veya mümkün olmayan olaylar için geliştirilen, var olan bilgileri işleme ve

tahminleme yaparak herhangi bir yardım almadan otomatik olarak gerçekleştiren bilgisayar sistemleri olarak da söylemek mümkündür (Öztemel, 2012).

Yapay sinir ağlarında, özellikle en güçlü teknikler arasında sayılabilen sınıflandırma, örüntü tanıma, sinyal filtreleme, veri sıkıştırma ve optimizasyon çalışmaları, günlük hayatta pek çok alanda başarılı şekilde kullanıldığı görülmektedir. YSA'ların pek çok sektörde kullanım alanları mevcuttur. Uygulama alanları genellikle tahmin, sınıflandırma, veri yorumlama ve filtreleme işlemlerinde olup, kullanım alanlarına bakıldığında ise finansal, ekonomi, haberleşme, mühendislik, tıp bilimi, uygulamalı alanlardan kontrol ve sistem tanımlama, görüntü ve ses tanıma, tahmin ve kestirim, arıza analizi, uzay-uçak-askeri teknoloji, robotik, otomotiv, veri madenciliği, trafik ve üretim yönetimi olarak karşımıza çıkar (Pirim, 2006).

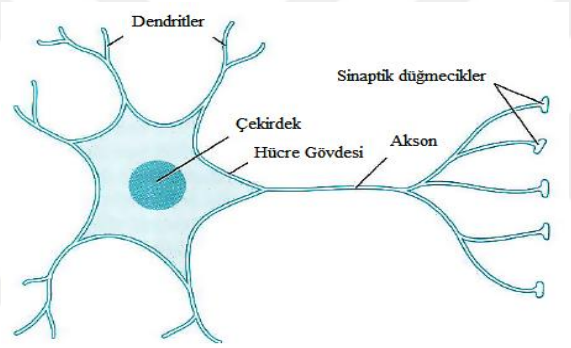
Yapay sinir ağları biyolojik sinir ağlarına benzer bir modellenmeyi içerir. Dolayısıyla yapay sinir ağlarının çalışma prensibini anlayabilmek için, öncelikle biyolojik sinir sistemi ve biyolojik sinir hücresinin yapısını incelemek gerekir. Biyolojik sinir sisteminin en küçük birimi olan sinir hücrelerinin (nöron), yapay sinir ağlarının da en küçük birimi olduğu söylenebilir.

2.4.1. Biyolojik Sinir Hücresi (Nöron)

Sinir hücresi (nöron), biyolojide sinir sisteminin temel yapıtaşı olarak adlandırılır. Nöronların görevi, sistemler arasında bilgi transferini gerçekleştirmektir. Bir bakıma, nöronları uyartının iletimini sağlayan kablo ağı gibi düşünülebilir. Bir insanın sinir sisteminde bulunan sinir hücreleri sayısı yaklaşık yüz milyardır. Bu sinir hücrelerinin bağlanmasını sağlayan taşıyıcıların sayısı ise yaklaşık 60 trilyon civarındadır. Duyu organlarıyla alınan sinyaller sinirler aracılığıyla sinir sistemine ulaştırılır. Sinir hücresinde, impuls oluşturan en küçük uyartı şiddetine eşik değeri denir. Bu işlemlerin geçerliliği, belli bir eşik değerinin aşılması ile mümkün olur.

Sinir hücresinin yapısında, çekirdek, gövde, büyük miktarda sinir uçları (dendritler), akson ve sinir uçları arasındaki uzantılar (sinapslar) bulunur. Hücre gövdesinde, çekirdek ve üyeler (organeller) bulunur. Çekirdek, hücrenin yaşamsal faaliyetlerinden sorumlu mekanizmadır. Sinir hücrelerinde sinirsel iletim, dendritten aksona doğru gerçekleştirilir. Dendritler, gelen sinyalleri bir araya toplayıp çekirdeğe gönderir ve tekrar gelen sinyaller bir araya toplanıp aksona

iletilir. Aksonda toplanmış olan bu sinyaller, işlenip bir sonraki sinir hücresine (sinaps) iletilir. Sinaps, bir sinyali başka sinir hücrelerine ileterek sinir hücresinde iletişimi sağlamış olurlar. Sinapsların bir görevi de, kendisine gelen sinyalleri işledikten sonra belli bir eşik değere göre ayarlayarak iletim mekanizması olmasıdır. Gelen sinyalleri belli bir aralıkta indirgeyerek diğer sinir hücrelerine iletirler. Buradan yola çıkarak, sinapslarda "öğrenme" işleminin gerçekleştiği fikri ele alındığından, bu teori günümüzde yapay sinir ağı dünyası için önemli bir gelişme halinde kullanılmaya başlanmıştır.



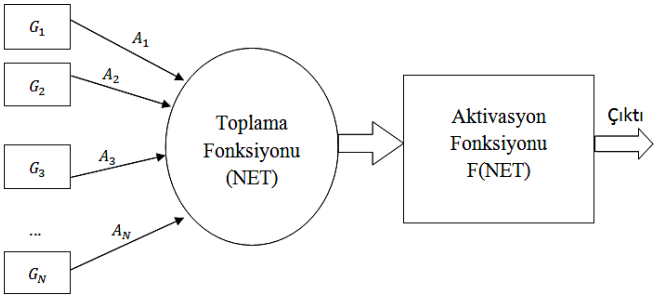
Şekil 2.9. Bir sinir hücresinin (nöron) yapısı

2.4.2. Yapay Sinir Hücresi

Yapay sinir ağı, biyolojik sinir sisteminde gerçek sinir hücrelerinin çalışma şekline benzetilerek, yapay sinir hücrelerinin (perceptron) birleşiminden oluşan yapının tamamı olarak adlandırılır (Lippmann, 1987; Efe ve Kaynak, 2000; Elmas, 2003). Diğer bir deyişle, insan beyninin bilgi işleme ve kullanma donanımından yola çıkarak geliştirdiği bir bilgi işlem teknolojisi olduğu da söylenebilir.

YSA'ları oluşturan temel elemanlar yapay sinir (yapay nöronlar) hücreleridir. YSA'lar bu yapay sinir hücrelerinin bir araya getirilmesiyle oluşmaktadır.

YSA oldukça basit bir yapıya sahiptir. YSA hücresinin yapısında ilk olarak, dışarıdan veri alınmasını sağlayan girişler (G_1, G_2, ..., G_N) bulunur. Sonrasında sırasıyla, katsayıların sürekli değişmesiyle öğrenmenin gerçekleştiği ağırlıklar (A_1, A_2, ..., A_N), verilerle ağırlıkların çarpılıp toplanmasıyla net girdiyi hesaplayan toplama fonksiyonu (NET), net çıktıyı veren aktivasyon fonksiyonu (F(NET)) ve başka hücreye aktarılacak veya ağın sonucunu veren çıkışlar vardır.



Şekil 2.10. Yapay sinir hücresinin yapısı

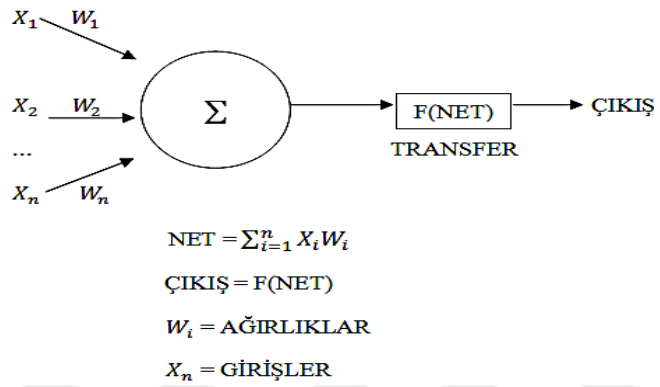
Çizelge 2.3. Biyolojik ve Yapay sinir hücrelerinin özellikleri

BİYOLOJİK VE YAPAY SİNİR SİSTEMLERİNİN KARŞILAŞTIRILMASI	
Biyolojik Sinir Sistemi	**YSA Sistemi**
Nöron	İşlem Elemanı
Dendrit	Girdiler
Hücre Gövdesi	Transfer Fonksiyonu
Akson	Yapay Nöron Çıkışı
Çekirdek	Aktivasyon Fonksiyonu
Sinapslar	Ağırlıklar

2.4.3 YSA Genel Yapısı ve Öğrenme

Yapay sinir ağları (YSA), yapay sinir hücreleri tarafından kendi içinde sınıflandırma yaparak öğrenmeyi sağlar. Öğrenmenin ilk adımında, önceki hücrelerin çıktıları girdi olarak kabul edilir ve her bir girdi ağırlık ile eşleştirilerek hücreye bağlanmış olur. Girdiler ve ağırlıklar çarpılıp toplanarak net girdi hesaplandıktan sonra aktivasyon fonksiyonundan geçirilerek o hücrenin net çıktısı bulunmuş olur. Bütün hücreler için aynı prosedür kullanılır. YSA, farklı girdi örnekleri verildiğinde ağırlıkları sürekli değiştirmek suretiyle örneklerin çıktılarıyla ağın çıktılarını eşit ya da eşite yakın yapmaya çalışır (Fausett, 1994).

McCulloch ve Pitts (1943) ilk temel hesap yapan nöron modelini tasarlamayı başarmış ve Şekil 2.11'de olduğu gibi "entegre et ve ateşle (computational model of neuron)" modelini önermiştir.

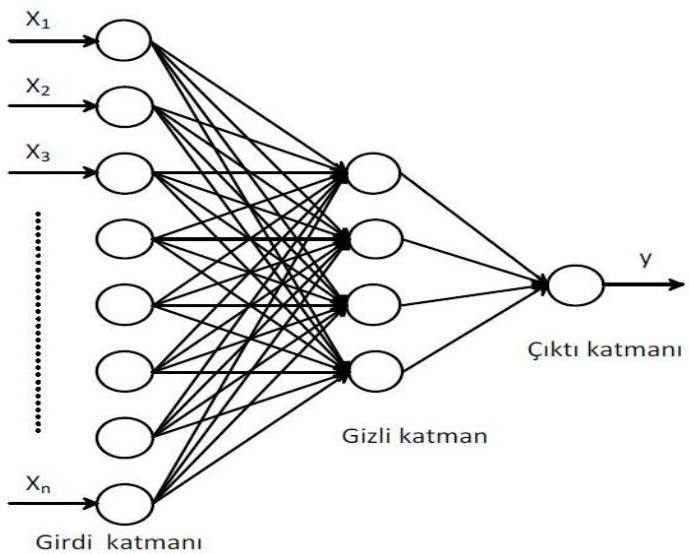


Şekil 2.11. Bir yapay sinir hücresinin matematiksel model örneği

Basit bir sinir hücresinin işleyişi şu şekilde ilerler:

- Giriş sinyallerinin ağırlıkları toplamı alınır.
- Bir eşik değeri seçilir.
- Girdilerin toplamı eşik değeri ile karşılaştırılır. Bu değer eşik değerinden küçükse -1, büyükse +1 olur.

Örnek olarak, bir yapay sinir ağının modeli aşağıdaki gibidir:



Şekil 2.12. Bir yapay sinir ağı modeli

Her bir yapay sinir hücresinin birleşimi ile yapay sinir ağları meydana gelmektedir. YSA'lar, sistemli şekilde birbirine bağlı ve paralel şekilde çalışan sistemlere sahip yapay hücrelerden oluşmuştur. YSA'daki hücrelerin birbirine bağlandığı kısımlara **proses elemanları** denir ve her bağlantının bir ağırlığı olduğu kabul edilir. Dolayısıyla, proses elemanlarının ağırlıklandırılmış şekilde birbirine bağlanmaları sonucu oluşturdukları ağa da **yapay sinir ağı** denir.

YSA, basit bir biyolojik sinir sisteminin çalışma şeklinin birebir aynısıdır. Ağdaki sinir hücreleri nöronları içerir ve bu nöronlar çeşitli şekillerde birbirleriyle bir araya gelerek ortaya çıkacak mevcut ağı oluştururlar. Nöronlar rastgele bir araya gelmezler. Oluşturulan ağlarda hafızaya alma, öğrenme ve verileri ilişkilendirme gibi çeşitli özelliklere sahip olur.

Şekil 2.12'de görüldüğü gibi, giriş (girdi) ile gizli katman arasındaki, gizli katmanlar arasındaki ve gizli katman ve çıkış (çıktı) katman arasındaki nöronlar birbirine bağlantılıdır. Bir yapay sinir hücresinin çıktısında temel olarak, hücrenin iç aktivasyonu ya da ham çıktısı, girdilerin ağırlıklandırılmış toplamıdır. Ancak genelde son değeri belirlemek için bir eşik fonksiyonu da kullanılır. Çıktı 1 ise, sinir hücresi aktif hale gelir. Çıktı 0 ise, sinir hücresi harekete geçmez.

YSA birden çok veri işleme katmanından oluşmaktadır. Buradaki ağlar proses elemanları (proses üyeleri, yapay sinir hücreleri) ile birbirine bağlıdır ve örneklerden yardım alarak öğrenme işlemini gerçekleştirirler. Her üyenin sahip olduğu bağlantının bir ağırlık değeri mevcut olup, yapay sinir ağının her ağırlık değerinde var olan bilgi saklanır ve örümcek ağına benzer bir görünümle sisteme yayılmışlardır.

YSA, yapay sinir hücrelerinden oluşur. Bir yapay sinir ağının en önemli görevi, kendisine verilen bir girdi setine karşılık çıktı setini oluşturmasıdır. Yani her resmi bir vektör (sayı) haline dönüştürür. Yapay sinir ağlarının bilgisayar bilimine önemli katkıları olmuştur. Günümüzdeki bilgisayar yazılım teknolojisi ile çözümü olmayan birçok problemin yapay sinir ağları yöntemi ile çözülebildiği görülür. Özellikle, en güçlü problem çözme tekniği olarak görülen YSA, bilgisayarların da öğrenebildiğini, durumlar hakkında bilgilerin olmadığı fakat örneklerin olduğu durumlarda, bir karar verme ve hesaplama tekniği olarak görülebilir.

Bir YSA'nın karakteristik özelliklerinde aşağıdaki özellikler mevcuttur (Öztemel, 2012).

- Makine öğrenmesini gerçekleştirirler.
- Çalışma stili bilinen programların, programlama tekniğiyle aynı değildir.
- Bilginin saklanmasında kullanılır.
- Örnekleri kullanarak ve işleyerek öğrenme tekniğini kullanır.
- Güvenle çalıştırılabilmesi için eğitilmeleri ve sonuçların test edilmesi gerekir.
- Bilinmeyen örnekler için bilgi üretebilirler.
- Sınıflandırmada kullanılırlar.
- Örüntü tanıma yapabilirler.
- Bazı bilgiler eksik olsa dahi çalıştırılmaları mümkündür.
- Hata toleransı vardır.
- Sadece nümerik bilgiler ile çalışabilmektedirler.

2.5. Dinamik Zaman Bükmesi

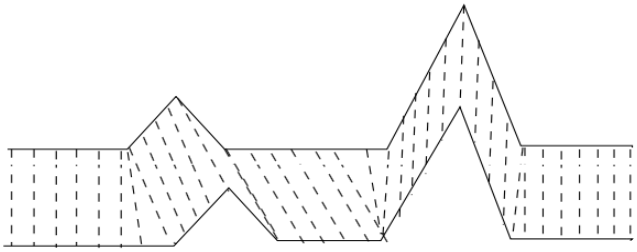
Ses tanıma (speech recognition), konuşma tanıma teknolojisi olarak da adlandırılan, insan-makine iletişiminde tuşları aradan kaldırarak, insan sesini algılayıp ses sinyallerine göre işlenerek bir konuşmacının söylediklerinin ne olduğunun tespit edilmesi işlemidir.

Konuşma tanımada, ilk olarak sözcükler ses sinyalleri olarak kaydedilir. Sonrasında ses sinyalleri işlenir, öznitelikler çıkarılır ve kullanılan metotlara göre ya özniteliklerin şablonu oluşturulur ya da metotlara göre modeller oluşturulur ve en son aşamada sözcüklerin tanınması işlemi yapılır.

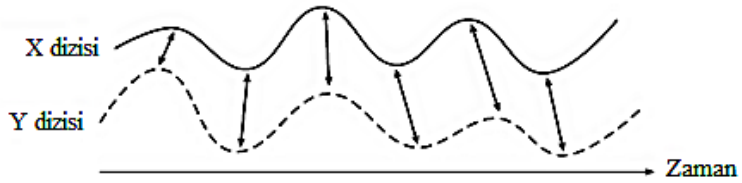
Konuşma tanımada ilk ve en çok kullanılan yöntemlerden birisi de Dinamik Zaman Bükmesi (DZB) yöntemidir. DZB, zaman serileri analizinde, farklı

uzunluktaki dizilerin arasındaki benzerliği ölçmeye yarayan bir metottur (Aşlıyan vd., 2008).

Genel olarak DZB, zaman serilerini sınıflandırma işleminde kullanılır. Bu metot, konuşma tanımadaki sözcüklerin ses sinyallerini temsil eden farklı uzunluktaki öznitelik vektörlerinin benzerliklerini hesaplamada kullanılabilir. Ayrıca, doğrusal bir diziye dönüştürülen herhangi bir veri, veri grafikleri, zamansal ses dizileri ve videolar da DZB ile analiz edilebilir. İyi bilinen bir uygulama da, otomatik konuşma tanımaya baktığımızda, farklı konuşma hızları ile başa çıkmak için bu yöntem tercih edilir.



Şekil 2.13. Dinamik Zaman Bükmesi



Şekil 2.14. İki dizinin DZB ile zamana bağlı hizalanması

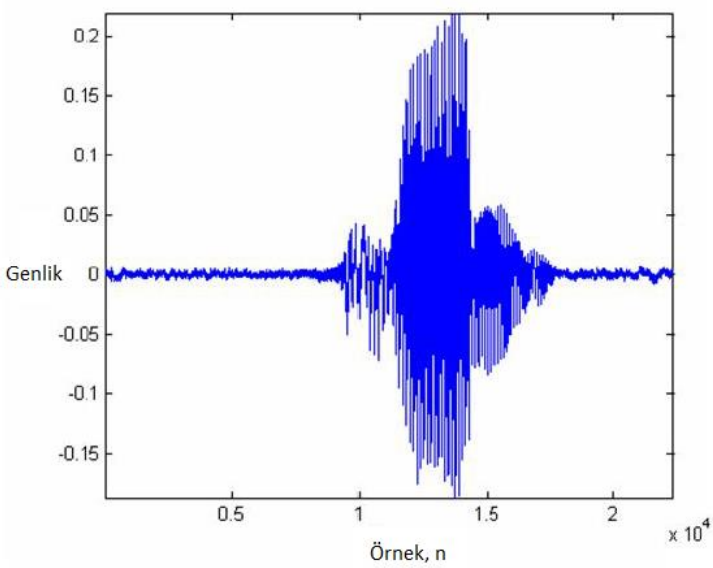
DZB, belirli kısıtlamalar altında (zamana bağlı) verilen iki dizi arasında, optimal uyum bulmaya yarayan iyi bir tekniktir. Şekil 2.14'te, iki dizinin DZB ile zamana bağlı hizalaması gösterilmiştir (Müller, 2007). Diziler, doğrusal olmayan bir şekilde birbirleriyle eşleştirilmiş ve hizalanmış olan noktalar oklarla gösterilmiştir.

DZB'nin bir diğer özelliği de, otomatik konuşma tanımadan farklı olarak, konuşma kalıplarını karşılaştırmak için kullanılır. Veri Madenciliği ve bilgi alma gibi alanlarda, zamana bağlı veri ve farklı hızlarla otomatik olarak başa

çıkmak için zaman deformasyonunu dengelemede başarılı şekilde uygulanmıştır (Müller, 2007).

Konuşma tanımadaki en büyük problemlerden birisi, bir sözcüğün farklı kişiler tarafından farklı şekilde söylenmesi olduğu kadar, bir kişinin aynı sözcüğü farklı zamanlarda aynı şekilde söyleyememesi durumudur. Bir kişi söylediği bir sözcüğü tekrar söylediğinde bile bu iki sözcüğün çok benzemediği görülür. Farklı zamanlarda söylenen sözcüklerin ses sinyallerinden oluşan öznitelik vektör boyutlarının kimisi uzun kimisi kısa olmaktadır. DZB metoduyla, karşılaştırılan sözcük veya fonem ses sinyallerinin aynı uzunlukta olması için zaman boyutunda kısaltma ve uzatma yapılır.

DZB yöntemi, genel olarak sözcük, hece ya da fonem tanıma işleminde de kullanılır. Şekil 2.15'te "fen" sözcüğünün zamana bağlı olarak genliğinden oluşan grafiği görülmektedir. Bu sözcüğün ses sinyallerinin başladığı ve bittiği zamanlar yaklaşık olarak tespit edilebilir. Ses sinyallerinin başladığı ve bittiği zaman aralığı, sözcüğün seslendirilme süresini belirlemektedir. "fen" sözcüğü her söylendiğinde büyük olasılıkla farklı seslendirilme süreleri bulunacaktır.



Şekil 2.15. "Fen" sözcüğünün genliğine göre dijital gösterimi

DZB yöntemi, dinamik programlama mantığını kullanarak vektörlerin benzerliklerini hesaplamaktadır. Yalıtık ses tanıma sisteminin en önemli özelliklerinden birisi, zamanda birebir eşleştirme yerine daha esnek yapıda bir eşleştirmeyi mümkün kılmasıdır. DZB'nin MATLAB kodları aşağıda verilmiştir.

```
function sonuc = DZB(k,t)
n = length(k+1);
m = length(t)+1 ;
dzb = zeros(n,m);
for x= 2:n
dzb (x,1) = inf;
end
for x = 2:m
dzb (1,x) = inf;
end
dzb (1,1) = 0;
for x = 2:n
for y = 2:m
uzaklik = abs(k(x-1)-t(y-1));
dzb (x,y) = uzaklik + min([dzb (x-1,y) dzb (x,y-1) dzb(x-1,y-1)]);
end
end
sonuc = dzb (n,m);
```

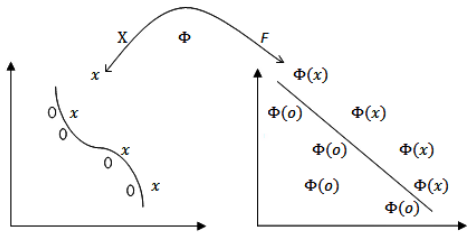
2.6. Destek Vektör Makinesi

Sınıflandırma işleminin veri madenciliğinde çok önemli bir yeri vardır. Veri madenciliği ile çok büyük miktardaki verinin işlenerek istenen veriye hızlı bir şekilde ulaşılabilmesi sağlanmaktadır. Sınıflandırma işlemiyle benzer veriler tespit

edilmektedir ve bu veriler daha sonra kullanılmak üzere saklanmaktadır. Sonrasında aranan veriyi daha hızlı bir şekilde bulabiliriz.

Veri madenciliği, bilinmeyen bilgiyi ortaya çıkarmak için çalışır. Veriden bilgiye giden süreç olarak tanımlayabiliriz. Örneğin; bir veri nasıl olmalı sorusunu sorduğumuzda; şirketin ihtiyacına göre tasarlanması, verinin temiz ve kaliteli olması ve tarihsel derinliğe sahip olması gerektiği yanıtları verilebilir.

Veri madenciliğinde kullanılan sınıflandırma yöntemlerinden birisi de Destek Vektör Makineleri (DVM) dir. DVM, veri setlerindeki örneklerin benzerlik durumlarına göre ayrılabilmelerini sağlayan sınıflandırma problemlerinde kullanılan yapay zeka metotlarındandır (Osuna vd., 1997; Aşlıyan ve Günel, 2009). Özellikle sınıflandırma ve regresyon gibi farklı alanlarda kullanılabilen öğretmenli öğrenme (supervised learning) yöntemi olarak da adlandırılır.



Şekil 2.16. DVM'de sınıflandırma

DVM, sınıflandırma konusunda kullanılan etkili ve basit yöntemlerden birisidir. İki boyutlu bir düzlemde iki farklı sınıfı temsil eden verilerin sınıflandırılması, sınıflar arasında bir doğru çizmek suretiyle yapılabilir. Lineer (doğrusal) olarak ayıramadığımız veri setlerini, daha büyük boyutlara dönüştürerek hiperdüzlem yardımıyla sınıflara bölebiliriz. DVM, iki sınıfın en iyi bir şekilde ayrılabilmesini sağlamak için sınıflar arasındaki en uygun sınırı belirlemeye çalışır.

DVM'ler parmak izi, yüz, el yazısı, retina tanıma vb. gibi birçok biyoinformaktik ve zaman serisi tahmininde yaygın olarak kullanılmaktadır. Girdisi ve çıktısı belli olan eğitim veri setlerindeki örnekler eğitilerek yani sınıflardaki örnekler arasındaki sınır belirlenerek modeller oluşturulur. Böylece, test setindeki örnekler için karar verme süreci başlamış olur. DVM modeli hangi sınıfa ait olduğu bilinmeyen bir örneği alarak sınıfının ne olacağına karar verir.

Eğitim setindeki girdi örnekleri lineer olarak ayrılabilirler ya da lineer olarak ayrılamayabilirler. Lineer olarak ayrılabilme durumunda sınıfları ayırabilen sonsuz doğrudan sınıflar arasındaki mesafesi maksimum yapacak en uygun doğru tespit edilir. Lineer olarak ayrılamama durumunda ise örnek verilerin boyutları daha büyük boyuta taşıyarak optimum sınırı hedefleyen hiperdüzlem bulunmaya çalışılır.

DVM'ler, genel olarak veri setlerini iki sınıfa bölmek için herhangi bir boyutta optimum hiperdüzlemi tespit eder. Bunun için eğitim setindeki örnek verilerin farklı sınıflarda bağımsız ve aynı sınıflarda benzer olmaları varsayılır (Yakut vd., 2014). Genellikle sınıflandırma problemlerinde kullanılır. Sınıflandırma problemlerinde, doğrusal olmayan durumların çözümlenebilmesi için çekirdek fonksiyonları kullanılır. DVM'de sıklıkla kullanılan çekirdek fonksiyonları şunlardır:

Doğrusal Fonksiyon: $\text{Ç}(x_i, x_j) = x_i^T x_j$

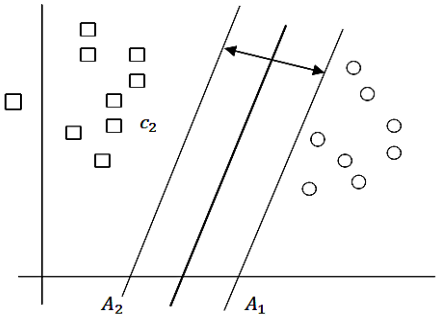
Sigmoid Fonksiyonu: $\text{Ç}(x_i, x_j) = \tanh(x_i x_j - \delta)$

Radyal Temelli Fonksiyon: $\text{Ç}(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right), \gamma > 0$

DVM, lineer olarak ayrılabilen ve lineer olarak ayrılamayan veri kümeleri olarak sınıflandırılabilir.

2.6.1. Lineer Olarak Ayrılabilme Durumu

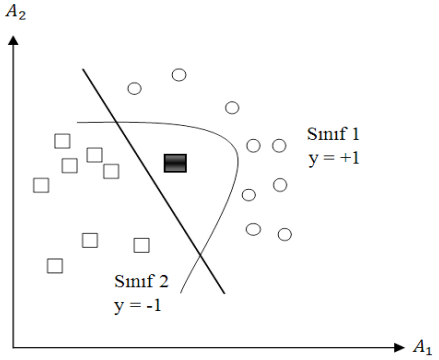
Eğer veri setindeki örnekler lineer olarak ayrılabiliyorsa, doğrudan bir düzlemle iki sınıf ayrılabiliyor demektir (Yakut vd., 2014). Bu düzleme ***ayırıcı düzlem*** denir. Destek Vektör Makinelerinde, ayrı sınıfta bulunan bu iki düzlemin, örnek gruba eşit mesafede olması amaçlanmaktadır.



Şekil 2.17. Optimal ayırıcı düzlem

2.6.2. Lineer Olarak Ayrılamama Durumu

Eğitim setindeki örnekleri her zaman için iki gruba, lineer olarak bir düzlemle ayıramayabiliriz. Şekil 2.18'de birbirinden lineer olarak ayrılamayan örnekler gösterilmiştir (Yakut vd., 2014).



Şekil 2.18. Doğrusal olarak ayrılamayan örnekler

İki sınıfı doğrusal olarak ayıramamak, eğitim örneklerinin belirtilen uzayda doğrusal olarak ayrılmaması anlamına gelmektedir. Bu durumda, doğrusal olmayan uzaydan doğrusal olarak kolayca sınıflandırma yapabileceği uzaya dönüşüm yapar. Yani, çekirdek fonksiyonları kullanılarak bulunan değerlerin, tercih edilen çekirdek fonksiyonundaki değeri yerine koymak suretiyle nitelik uzayında değeri tespit edilir. Eğitim setindeki örneklere göre fonksiyon oluşturarak değerleri bulunur ve kalıp değerler bulunduğundan diğer örneklerin değerlerini hesaplamak oldukça kolaylaştırmaktadır (Yakut vd., 2014).

3. BULGULAR VE TARTIŞMA

3.1. Sözcük Ses Veri Seti

Türkçe sözcük ses veri seti, 200 farklı Türkçe sözcüğün Windows ortamında MATLAB ile her bir sözcüğün 35 defa kaydedilmesiyle oluşturuldu. Toplamda 7000 sözcük ses dosyasından oluşan veri seti elde edilmiştir. Ek 1'de verilen bu sözcükler, bir konuşmacıyla 3 saniye içinde seslendirilmiştir. 11025 Hz frekansında örneklenerek 16 bitlik Darbe Kod Kiplenimi (PCM) ile nicemlenmiştir. Bu ses dosyaları 354 MB yer kaplamaktadır. Her sözcüğün 25 tane ses örneği eğitim aşamasında, geri kalana 10 tanesi de test için kullanılmıştır.

3.2. Sistemlerin Değerlendirilmesi ve Karşılaştırılması

Doğruluk (Accuracy), konuşma tanıma sistemlerinin değerlendirilmesinde en çok kullanılan değerlendirme ölçüsüdür.

Doğruluk (Accuracy): Doğru sınıflandırılan örneklerin sayısının toplam örnek sayısına oranı olarak ifade edilir.

Çizelge 3.1, "Pozitif" ve "Negatif" sınıfların örneklerinin test sonucunda doğru ve yanlış sınıflandırma sayılarını belirtmektedir. D, sistemin doğru sınıflandırdığını; Y, ise yanlış sınıflandırdığını göstermektedir. DP: Pozitif sınıf içindeki örneklerden sistemin pozitif olarak belirlediği örnek sayısı. YP: Negatif sınıf içindeki örneklerden sistemin pozitif olarak belirlediği örnek sayısı. YN: Negatif sınıf içindeki örneklerden sistemin negatif olarak belirlediği örnek sayısı. DN: Pozitif sınıf içindeki örneklerden sistemin negatif olarak belirlediği örnek sayısı. Sistemin değerlendirilmesini sağlayan Doğruluk değerlerine Denklem 3.1'deki eşitlilikten ulaşılabilir:

Çizelge 3.1. Hata matrisi

	Sınıf: "Pozitif"	Sınıf: "Negatif"
Test: "Pozitif"	DP	YP
Test: "Negatif"	YN	DN

$$\text{Doğruluk} = (\text{DP} + \text{DN})/(\text{DP} + \text{DN} + \text{YP} + \text{YN}) \quad (3.1)$$

3.3. Sistemin Eğitilmesi ve Test Edilmesi

Geliştirilen ses tanıma sistemleri, Intel şirketinin 3,4 GHz işlemcili (i7-2600), 16 GB ana belleğe sahip ve Windows işletim sistemli bilgisayar üzerinde uygulanmıştır. MATLAB ile kodlanarak önişleme, MFCC öznitelikleri çıkarılması, eğitim, test ve artişleme (postprocessing) işlemleri yapılmıştır.

Vektör şeklindeki hece ve sözcük ses sinyalleri 10 milisaniye örtüşme yapılarak 20 milisaniyelik bloklara ayrıldıktan sonra her bir blok, 10 tane MFCC değere sahip olan $x(10, n)$ öznitelik matrisine dönüştürülür. Buradaki n değişkeni hece için 30, sözcük için 120 seçilmiştir. Eğer hece öznitelik matrisinin sütun sayısı 30'dan az ise sütunları 30 olacak şekilde öznitelik matrisine eşit aralıklarla aynı sütunlar eklenir. Eğer hece öznitelik matrisi 30'dan fazlaysa yine sütun sayısı 30 olacak şekilde eşit aralıklarla sütunlar silinir. Benzer işlem sözcükleri içinde öznitelik sütun 120 olacak şekilde yapılır.

Çizelge 3.2'de hece tabanlı sistemlerin doğruluk yüzdeleri gösterilmiştir. Hece tabanlı sistemlerde kullanılan metotlar, Dinamik Zaman Bükmesi (DZB), Saklı Markov Modeli (SMM), Destek Vektör Makinesi (DVM) ve Yapay Sinir Ağlarında kullanılan yöntemlerden biri olan (ÇKA) Çok Katmanlı Algılayıcı'dır. Çizelge, Artişleme durumuna göre ikiye ayrılır. İlk olarak, Artişleme yapılmadan uygulanan metot sonuçları incelendiğinde, en yüksek değere sahip olan metodun %89,6 doğruluk oranı ile DZB olduğu görülmektedir. Doğruluk oranına göre sırasıyla, en yüksek değerden en düşük değere göre sıralamak istenildiğinde, %82,8 ile DVM, %79,0 ile ÇKA ve %65,4 ile SMM olduğu görülür. Tüm metotlara Artişleme yapıldıktan sonraki değerlere bakıldığında ise, en yüksek değere sahip olan metodun %94,2 doğruluk oranı olarak DZB olduğu görülmektedir. Sırasıyla, en yüksek değerden en düşük değere göre sıralayacak olursak, %90,8 ile DVM, %88,0 ile ÇKA ve son olarak %82,6 ile SMM olduğu görülür.

Çizelge 3.2. Hece tabanlı sistemlerin doğruluk yüzdeleri

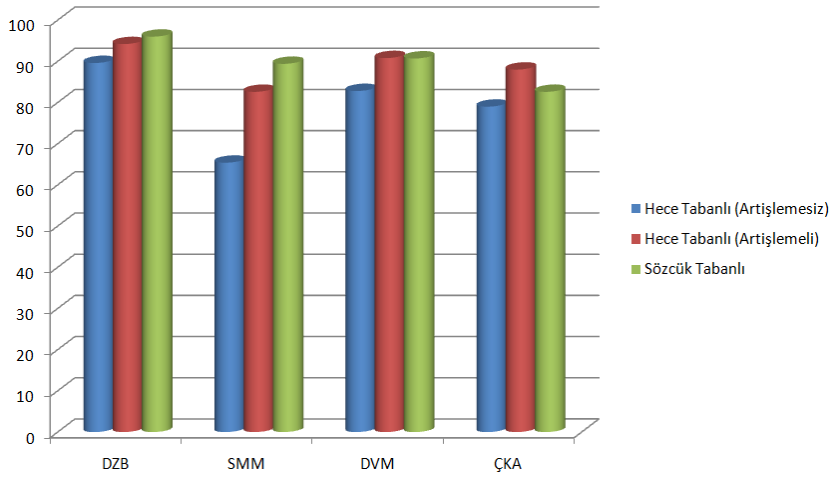
Metotlar	Doğruluk (%) Artişleme (Hayır)	Doğruluk (%) Artişleme (Evet)
DZB	89,6	94,2
SMM	65,4	82,6
DVM	82,8	90,8
ÇKA	79,0	88,0

Çizelge 3.3'e bakıldığında, sözcük tabanlı sistemlerin doğruluk yüzdeleri gösterilmiştir. Kullanılan metotlar Çizelge 3.2'deki gibi aynı olup, sadece doğruluk değerlerine göre sıralanmıştır. Doğruluk değerlerine göre en yüksek değere sahip olan metodun %96,0 ile DZB olduğu görülmektedir. Doğruluk değeri en yüksek olandan en düşük olana göre sıralandığında ise, sırasıyla, %90,7 ile DVM, %89,4 ile SMM ve %82,6 ile ÇKA olduğu görülür.

Çizelge 3.3. Sözcük tabanlı sistemlerin doğruluk yüzdeleri

Metotlar	Doğruluk (%)
DZB	96,0
SMM	89,4
DVM	90,7
ÇKA	82,6

Şekil 3.1’de görüldüğü gibi hece tabanlı ve sözcük tabanlı sistemlerin doğruluk oranına göre karşılaştırılması yapılmıştır. Buna göre, sözcük tabanlı sistemlerde DZB ve SMM metotları hece tabanlı sistemlere göre daha başarılı olurken, DVM ve ÇKA metotlarında ise artişlemeli hece tabanlı sistemlerin çok daha başarılı olduğu tespit edilmiştir. Artişlemeli hece tabanlı sistemlerin artişlemesize göre çok başarılı olduğu görülmektedir. DZB ve DVM metotların başarıları bir birine oldukça yakındır. Buna karşın, SMM ve ÇKA metotlarının başarı oranları arasında çok belirgin bir fark olduğu gözlemlenmiştir.



Şekil 3.1. Konuşma tanıma sistemlerinin genel karşılaştırılması

Çizelge 3.4'te hece tabanlı sistemlerdeki bir hecenin eğitim ve test zamanları görülmektedir. Bu çizelgeye göre eğitim ve test zamanlarına göre sırasıyla 4,7 ve 0,7 saniye olmak üzere en kısa zamana sahip metot SMM metodu olmuştur. DVM ve ÇKA metotlarının eğitim süreleri SMM metoduna göre oldukça yüksek olduğu görülmektedir. Buna karşın test süreleri, SMM'den yüksek olamamakla birlikte çok yüksek değildir. DZB metodunda eğitim işlemi olmadığından eğitim süresi verilmemiştir. Test süresi ise 3,6 saniye olarak ölçülmüştür.

Çizelge 3.4. Hece tabanlı sistemlerin hece eğitim ve test süreleri

Metotlar	Eğitim Zamanı (Saniye)	Test zamanı (Saniye)
SMM	4,7	0,7
DVM	111.3	1,3
ÇKA	109,8	1,1
DZB	-	3,6

Çizelge 3.5'te görüldüğü üzere sözcük tabanlı sistemlerin ortalama eğitim ve test sürelerinde en az süreye sahip metot hece tabanlıda olduğu gibi SMM metodudur. Sonrasında sırasıyla en az süreye sahip metotlar ÇKA, DVM ve DZB olmuştur. DZB metodunun test işleminin oldukça uzun sürdüğü tespit edilmiştir.

Çizelge 3.5. Sözcük tabanlı sistemlerin sözcük eğitim ve test süreleri

Metotlar	Eğitim Zamanı (Saniye)	Test zamanı (Saniye)
SMM	23,9	1,4
DVM	165,3	1,7
ÇKA	132,7	1,5
DZB	-	213,7

4. SONUÇLAR

Bu tezde, DZB, SMM, ÇKA ve DVM metotlarıyla, orta ölçekli, ayrık ve kişiye bağımlı Türkçe konuşma tanıma uygulamaları yapılmıştır. Hece ve sözcük tabanlı olarak gerçekleştirilen bu sitemlerin başarıları karşılaştırılmıştır. Konuşma tanıma sistemlerinde çok başarılı özniteliklerden MFCC kullanılarak eğitim ve test işlemleri yapılmıştır. 200 Türkçe sözcük içeren ses veri setimizdeki her hece ve sözcük SMM, ÇKA ve DVM metotlarıyla eğitilerek modeller oluşturulmuştur. Hece tabanlı sistemimizde artişleme aşaması da yapılmıştır. Bu işlem sonucunda başarı yaklaşık %14 artmıştır. Artişleme yapılan hece tabanlı sistemlerdeki en başarılı metotlar sırasıyla DZB (%94,2), DVM (%90,8), ÇKA (%88) ve SMM (%82,6) olarak tespit edilmiştir. Sözcük tabanlı sistemlerdeki en başarılı metotlar ise sırasıyla DZB (%96), DVM (%90,7), SMM (%89,4) ve ÇKA (%82,6) olmuştur. Görüldüğü üzere DZB ve SMM metotlarında sözcük tabanlı sistemler daha başarılı iken DVM ve ÇKA metotlarında hece tabanlı sistemler daha başarılı bulunmuştur.

Daha sonraki çalışmalarda farklı öznitelik çıkarma metotları kullanılarak hece ve sözcük tabanlı sistemler geliştirilecektir. Aynı zamanda farklı sınıflandırma metotlarıyla uygulamalar geliştirilecektir. Hece ve sözcük tabanlı hibrit sistemler konuşma tanıma sistemlerinin başarısını artırabilir.

KAYNAKLAR

Aksoylar, C., Mutluergil, S. O. ve Erdoğan, H., 2009. Bir Türkçe Konuşma Tanıma Sisteminin Anatomisi. **IEEE 17th Signal Processing and Communications Applications Conference**, pp. 512-515. Antalya.

Asefisaray, B. 2012. Konuşma Tanıma Sistemleri. **TMMOB EMO Ankara Şubesi Haber Bülteni**, 4: 18.

Aşlıyan, R. 2008. Design And Implementation of Turkish Speech Recognition Engine. Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi (Basılmış), İzmir.

Aşlıyan, R. ve Günel, K., 2009. Destek Vektör Makinesi Yöntemiyle Türkçe Konuşma Tanıma Sistemi Gerçekleştirilimi. **Akademik Bilişim'09 - XI. Akademik Bilişim Konferansı Bildirileri**. Şanlı Urfa.

Aşlıyan, R., Günel, K. ve Yakhno, T., 2008. Dinamik Zaman Bükmesi Yöntemiyle Hece Tabanlı Konuşma Tanıma Sistemi. **Akademik Bilişim'08 - X. Akademik Bilişim Konferansı Bildirileri**, pp. 19-23. Çanakkale.

Aydın, Ö. 2005. Yapay Sinir Ağlarını Kullanarak Bir Ses Tanıma Sistemi Geliştirilmesi. Trakya Üniversitesi Fen Bilimleri Enstitüsü, 82, Edirne.

Baum, L. E. ve Eagon, J. A. 1967. An Inequality with Applications to Statistical Estimation for Probabilistic Functions of a Markov Process and to a Model for Ecology. **Bulletin of the American Mathematical Society**, 73(3): 360-363.

Baygün, M. K. 2006. Türkçe Komutları Tanıyan Ses Tanıma Sistemi Geliştirilmesi. Yüksek Lisans Tezi. Pamukkale Üniversitesi, Denizli.

Burcu, C. 2007. Bir Hece-Tabanlı Türkçe Sesli İfade Tanıma Sisteminin Tasarımı ve Gerçekleştirimi. Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara.

Çapar, A., Taşdemir, K., Kılıç, Ö. ve Gökmen, M., 2002. Türkçe Elyazısı Tanıma Sistemlerinde Öznitelik Çıkarma Ve Sınıflandırma Yöntemlerinin Karşılaştırılması. **Internatonal Union of Radio Science,** pp. 1-4, İstanbul.

Dreyfus-Graf, J. 1950. Sonograph and Sound Mechanics, **The Journal of the Acoustical Society of America** [Electronic Journal], 22(6): 731, Erişim [http://dx.doi.org/10.1121/1.1906680]

Efe, M. Ö. ve Kaynak, O. 2000. Artificial Neural Networks. **Boğaziçi University Publishing**, İstanbul.

Elmas, Ç. 2003. Artificial Neural Networks. Seçkin Yayıncılık, Ankara.

Jelinek, F. 1997. Statistical Methods for Speech Recognition. MIT Press, Cambridge.

Karadaş, İ. 2013. Konuşma Tanıma Teknolojisini Kullanılarak Okul Öncesi Bilişsel Kazanımlarının Öğretimi İçin Yazılım Geliştirme. Gazi Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.

Karasartova, S. 2011. Metinden Bağımsız Konuşmacı Tanıma Sistemlerinin İncelenmesi Ve Gerçekleştirilmesi. Ankara Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi (Basılmış), Ankara.

Koç, A. 2002. Acoustic Feature Analysis For Robust Speech Recognition. M. S. Thesis. Boğaziçi University, İstanbul.

Lippmann, R. 1987. An Introduction to Computing with Neural Nets. **IEEE Trans. ASSP Magazine**, 4(2): 4-22.

McCowan, I. A., Moore, D., Dines, J., Gatica-Perez, D., Flynn, M., Wellner, P. ve Bourlard, H. 2004. On the Use of Information Retrieval Measure for Speech Recognition Evaluation. **IDIAP Research Institute**, Martigny, Switzerland.

Mengüşoğlu, E. 1999. Bir Türkçe Sesli İfade Tanıma Sisteminin Kural Tabanlı Tasarımı ve Gerçekleştirimi. Yüksek Lisans Tezi. Hacettepe Üniversitesi, Ankara.

Moralı, İ. A. ve Aygün, F. F., 2007. Çok Katmanlı Algılayıcı Ve Geriye Yayılım Algoritması İle Konuşmacı Ayırt Etme. **Akademik Bilişim'07 - IX. Akademik Bilişim Konferansı Bildirileri**, pp. 57-62, Kütahya.

Müller, M. 2007. Dynamic Time Warping: Information Retrieval for Music and Motion, 69-84, Springer Berlin Heidelberg, Germany.

Nadas, A., Nahammoo, D. ve Picheny, M. A. 1988. On a Model Robust Training Method for Speech Recognition. **IEEE Transactions on Accoustic, Speech, and Signal Processing**, 36(9): 1432-1436.

Olson, H. F. ve Belar, H. 1956. Phonetic Typewriter, **The Journal of the Acoustical Society of America** [Electronic Journal], 28(6): 1072-1081, Erişim [http://dx.doi.org/10.1121/1.1908561]

Osuna, E., Freud, R. ve Girosi, F. 1997. Training support vector machines: An applications to face detection. **Proc. IEEE**, pp. 130-136, San Juan.

Özcan, G. 2015. Saklı Markov Modelleri Ve Uygulamaları. **Akademik Bilişim'15 - XV. Akademik Bilişim Konferansı Bildirileri**, pp. 1-10, Eskişehir.

Öztemel, E. 2012. Yapay Sinir Ağları. **Papatya Yayıncılık**, 29, İstanbul.

Öztemel, E. 2012. Yapay Sinir Ağları. Yapay Sinir Ağları (Dr. Rıfat Çölkesen, Necdet Avcı ve Ziya Çölkesen), **Papatya Yayıncılık**, 41, İstanbul.

Öztemel, E. 2012. Yapay Sinir Ağlarının Genel Özellikleri. Yapay Sinir Ağları (Dr. Rıfat Çölkesen, Necdet Avcı ve Ziya Çölkesen), **Papatya Yayıncılık**, 31-33, İstanbul.

Pirim, H. 2006. Yapay Zeka. **Journal of Yaşar University**, 1(1): 81-93.

Poritz, A. 1988. Hidden Markov Models: a guided tour. **In International Conference on Acoustics, Speech, and Signal Processing**, 1: 7-13.

Proakis, J. G. ve Manolakis, D. G. 1996. Digital Signal Processing: Principles and Application. 480, Prentice-Hall, Upper Saddle River, NJ.

Rabiner, L. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. **Proc. IEEE**, 77(2): 257-286.

Rabiner, L. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. **Proceedings of the IEEE**, 77(2): 257–286.

Rabiner, L. R. ve Juang, B. H. 1986. An introduction to Hidden Markov Models**. IEEE Acoustic, Speech, and Signal Processing Magazine**, 3(1): 4-16.

Rabiner, L. ve Juang, B. H. 1993. Fundamentals of Speech Recognition. New York: Prentice-Hall, 277, Englewood Cliffs, NJ.

Rabiner, L., Levinson, S., Rosenborg, A. ve Wilpon, J. 1979. Speaker Independent Recognition of Isolated Words Using Clustering Techniques. **IEEE Trans. ASSP**, 27(4): 336-349.

Sewell, M. 2008. Hidden Markov Models. University College London Department of Computer Science, England.

Tunalı, V. 2005. A Speaker Dependent, Large Vocabulary, Isolated Word Speech Recognition System for Turkish. M. S. Thesis, Marmara University, İstanbul.

Vicens, P. 1969. Aspects of Speech Recognition by Computer. Ph. D. Thesis. Computer Science, Stanford University, California.

Yakar, Ö. ve Aşlıyan, R. 2016. Saklı Markov Modeli Kullanarak Türkçe Konuşma Tanıma. **Akademik Bilişim'16 - XVIII. Akademik Bilişim Konferansı Bildirileri**, pp. 1-7, Aydın.

Yakut, E., Elmas, B. ve Yavuz, S. 2014. Yapay Sinir Ağları ve Destek Vektör Makineleri Yöntemleriyle Borsa Endeksi Tahmini. **Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, 19(1): 139-157.

Yalçın, N. 2008. Konuşma Tanıma Teorisi ve Teknikleri. Kastamonu Eğitim Dergisi, 16(1): 249-266.

Yılmaz, C. 1999. A Large Vocabulary Speech Recognition System for Turkish. M. S. Thesis, Bilkent University, Ankara.

EKLER

EK 1. Konuşma Tanımada Kullanılan Sözcükler

abajur, abaküs, aceleci, acemice, acımasız, acil, aç, açıklamak, adaletsizlik, adapazarı, ağaçlandırmak, ahşap, ak, akarsu, akça, akıcılık, akıllanmak, akreditasyon, aks, aktar, akvaryumculuk, alçı, alt, anıtlaştırmak, aydın, baba, badanacılık, bağlam, bahçıvan, bahçıvanlık, bakla, bal, baltık, bar, bardak, basamaklı, başarısızlık, başbakanlık, belirlemek, beneklenmedi, benzerlik, benzeşim, bereket, biçimsel, bilet, bilimsel, biyosfer, boncuk, borç, bordo, boyutlandırmak, burun, bülten, can, caz, cesaretli, cevizli, cezalı, coğrafya, cumhuriyet, çabalamak, çabuklaştırmak, çağla, çakır, çal, çalgı, çalım, çalışkan, çalışmak, çam, çay, çekim, çeşitlilik, çiçekçilik, çimenlik, çobanpüskülü, dalgınlaşma, damga, danışmak, dansimetre, dargın, dayanışma, defter, deha, delgi, demokrasi, deneme, denetleyici, denizaltı, denizyıldızı, dert, dev, divan, doğru, doksan, doktor, durak, ehliyet, eldiven, elektrik, elektrikçi, elektroteknik, endüstrileşme, evcilleştirme, faks, fark, farklılaştırma, faydalı, felaketzede, felek, ferman, feza, fındık, fikir, fiyasko, fotokopi, gazetecilik, gecekondu, habersiz, halıcı, hareketli, ihtiyarlamak, inandırma, iştahsızlık, iyotlu, izcilik, kabataş, kafkasyalı, kahramanlık, kalorimetre, kamburlaştırmak, kan, kap, kapitülasyon, karikatürcü, kılavuz, kitabevi, kundura, lösemi, macun, maç, maden, mafya, maharet, makas, malümat, mart, mat, mert, misafirlik, mükemmeliyet, mütemadiyen, naftalin, nakliyat, nefeslenmek, neşelendirmek, neşeli, nicelemek, nitelendirmek, not, nur, of, ok, oksijen, okuryazarlık, organizasyon, ormanlık, ot, pim, plak, plan, prens, programcılık, radyoelektrik, radyoloji, renk, resimlendirme, rey, ring, risk, robotlaştırmak, sabunlaştırmak, samimiyetlik, sevindirmek, simülasyon, siyasetname, sosyoloji, şekerleme, tank, tarz, taş, tatbikat, termodinamik, uygarlaştırmak, ücretlendirme, yabancılık, yaz, ziyaret, ziyaretçi, zor.

EK 2. ÇKA ve DVM MATLAB Programları

ÇKA Eğitim Programı

```
% Her sınıf eğitilip bir model oluşturulur.
clear all, clc
load TestSet.mat % TestSet.P, TestSet.T ,TestSet.Tsayi
load TrainSet.mat % TrainSet.P, TrainSet.T, TrainSet.Tsayi
sozcuksayisi=200; % Model oluşturulacak sözcük sayısı
sozcukorneksayisi=25; % Eğitim içindeki sözcük örnek sayısı
toplamornek=size(TrainSet.P,2);
oznitelikvektorboyutu=size(TrainSet.P,1);
Diger_KacOrnekAlinacak=5; % 1 için 25 örnek -1 için alınacak örnek sayısı
Egit1_bas=1; %her sınıf için 25 artırılacak
for sinif=1:sozcuksayisi
fprintf('Toplam Sozcuk: %d   Kaçıncı Sözcük: %d\n',sozcuksayisi,sinif);
% Sınıf 1 için
clear P T1 T2 T sozcuk
sozcuk=TrainSet.T{Egit1_bas};
Egit1_son=Egit1_bas+sozcukorneksayisi-1;
P=TrainSet.P(:,Egit1_bas:Egit1_son);
T1(1:sozcukorneksayisi)=1;
T2(1:((sozcuksayisi-1)*Diger_KacOrnekAlinacak))=0;
T=[T2 T1];
% sınıf -1 için
for i=1:sozcukorneksayisi:toplamornek
```

```
if i~=Egit1_bas
P=[TrainSet.P(:,i:i+Diger_KacOrnekAlinacak-1) P];
end
end
% ÇKA Eğitimi
clear X Y net
X=P;
Y=T;
R=oznitelikvektorboyutu; % Girdi boyutu
inputrange=ones(R,2);
inputrange(:,1)=-10;
inputrange(:,2)=1;
net=newff(inputrange,[30,10,1], { 'logsig' , 'logsig' , 'logsig'}, 'traingd' ) ;
% Başlangıç değerlerin atanması
net = init(net);
% ÇKA eğitiliyor.
net.trainParam.show=2000;
% Öğrenme oranı
net.trainParam.lr=0.02;
% Momentum katsayısı
net.trainParam.mc=0.9;
net.trainParam.epochs=2000000;
% Hedef
net.trainParam.goal=6e-3;
[net,tr]=train(net,X,Y);
MLP_Model.sozcuk{sinif}=sozcuk;
```

```
MLP_Model.model{sinif}=net;
% MLP Eğitim son
Egit1_bas=Egit1_bas+sozcukorneksayisi;
end
save 'MLPModel.mat' MLP_Model
```

ÇKA Test Programı

```
% Test Setindeki her sözcük MLPModel e göre test edilir.
clear all, clc
load TestSet.mat % TestSet.P, TestSet.T ,TestSet.Tsayi
load MLPModel.mat % MLP_Model.sozcuk{sinif}=sozcuk; MLP_Model.model{sinif}
modelsayisi=length(MLP_Model.model);
sozcukorneksayisi=10; % Test içindeki sözcük örnek sayısı
toplamornek=size(TestSet.P,2);
Dogrusayisi=0;
for i=1:toplamornek
disp(i)
clear P sonuc bulunansozcuk
P=TestSet.P(:,i);
for m=1:modelsayisi
clear net
net=MLP_Model.model{m};
Deger=sim(net,P);
sonuc(m)=Deger;
end %m=1:modelsayisi
```

```
[a sinif]=max(sonuc);
bulunansozcuk=MLP_Model.sozcuk{sinif};
if strcmp(TestSet.T{i},bulunansozcuk)
Dogrusayisi=Dogrusayisi+1;
end
fprintf('%s %s\n',TestSet.T{i},bulunansozcuk);
end
DogrulukOrani=100*(Dogrusayisi/toplamornek);
fprintf('Dogruluk Oranı:%4.2f',DogrulukOrani);
save 'DogrulukOrani.mat' DogrulukOrani
```

DVM Eğitim Programı

```
% Her sınıf eğitilip bir model oluşturulur.
clear all, clc
load TestSet.mat % TestSet.P, TestSet.T ,TestSet.Tsayi
load TrainSet.mat % TrainSet.P, TrainSet.T, TrainSet.Tsayi
sozcuksayisi=200; % Model oluşturulacak sözcük sayısı
sozcukorneksayisi=25; % Eğitim içindeki sözcük örnek sayısı
toplamornek=size(TrainSet.P,2);
Diger_KacOrnekAlinacak=5; % 1 için 25 ornek -1 için alınacak örnek sayısı
Egit1_bas=1;
for sinif=1:sozcuksayisi
% Sınıf 1için
clear P T sozcuk
sozcuk=TrainSet.T{Egit1_bas};
```

```
Egit1_son=Egit1_bas+sozcukorneksayisi-1;
P=TrainSet.P(:,Egit1_bas:Egit1_son);
T(1:sozcukorneksayisi)=1;
T(26:sozcukorneksayisi+((sozcuksayisi-1)*Diger_KacOrnekAlinacak))=-1;
% Sınıf -1 için
for i=1:sozcukorneksayisi:toplamornek
if i~=Egit1_bas
P=[P TrainSet.P(:,i:i+Diger_KacOrnekAlinacak-1)];
end
end
% SVM Eğitimi
clear X Y
X=P';
Y=T';
net = svm(size(X, 2), 'rbf', 80, 300);
net.qpsize = 18;
alpha0 = net.alpha;
net.c = [5 100];
net=svmcv(net,X,Y,[0.4 0.8],1.1);
net=svmtrain(net, X, Y,alpha0,2);
% Test işlemi
SVM_Model.sozcuk{sinif}=sozcuk;
SVM_Model.model{sinif}=net;
% SVM Eğitim son
Egit1_bas=Egit1_bas+sozcukorneksayisi;
end
```

```
save 'SVMModel.mat' SVM_Model
```

DVM Test Programı

```
% Test setindeki her sözcük SVMModel e göre test edilir.
clear all, clc
load TestSet.mat % TestSet.P, TestSet.T ,TestSet.Tsayi
load SVMModel.mat % SVM_Model.sozcuk{sinif}=sozcuk; SVM_Model.model{sinif}
modelsayisi=length(SVM_Model.model);
sozcukorneksayisi=10; % Test içindeki sözcük örnek sayısı
toplamornek=size(TestSet.P,2);
Dogrusayisi=0;
for i=1:toplamornek
disp(i)
clear P sonuc bulunansozcuk
P=TestSet.P(:,i)';
for m=1:modelsayisi
clear net
net=SVM_Model.model{m};
[Yout, Y1out] = svmfwd(net,P);
sonuc(m)=Y1out;
end %m=1:modelsayisi
[a sinif]=max(sonuc);
bulunansozcuk=SVM_Model.sozcuk{sinif};
if strcmp(TestSet.T{i},bulunansozcuk)
Dogrusayisi=Dogrusayisi+1;
```

```
end
fprintf('%s %s\n',TestSet.T{i},bulunansozcuk);
end
DogrulukOrani=100*(Dogrusayisi/toplamornek);
fprintf('Dogruluk Oranı:%4.2f',DogrulukOrani);
save 'DogrulukOrani.mat' DogrulukOrani
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı:	Özlem YAKAR
Doğum Yeri Ve Tarihi:	Şişli / 21.05.1989

EĞİTİM DURUMU

Lisans:	Adnan Menderes Üni. / Fen-Edebiyat Fakültesi / Matematik Bölümü
Yüksek Lisans:	Adnan Menderes Üni. / Fen Bilimleri Enstitüsü / Uygulamalı Matematik ABD
Yabancı Dil:	İngilizce, Almanca

BİLİMSEL FAALİYETLERİ

Bildiri:	Yakar, Ö. ve Aşlıyan, R., Saklı Markov Modeli Kullanarak Türkçe Konuşma Tanıma, Akademik Bilişim 2016, Aydın.
Seminer:	Yakar, Ö., Destek Vektör Makinesi, Adnan Menderes Üniversitesi / Fen Bilimleri Enstitüsü.
Seminer:	Yakar, Ö., Goldie Halkası, Adnan Menderes Üniversitesi / Fen Edebiyat Fakültesi.

İLETİŞİM

E-Posta Adresi:	ozlemyakar.34@gmail.com

Tarih:	.../.../2016