BIL735 Konuşma Tanıma - Ödev Raporu

Emre Kağan Akkaya - N14128491
Hacettepe Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Bölümü
emrekaganakkaya@gmail.com

1. Giriş

Bu rapor BIL735 Konuşma Tanıma dersinde verilen ödevlere ilişkin bir özet niteliğindedir. Her bir ödeve karşılık aşağıdaki başlıklarda ilişkili konuya kısaca değinilmiştir. Bununla birlikte ödevlere ait kaynak koda ve ödev çıktılarına buradan ulaşılabilir.

2. Veri yakalama ve otomatik son bulma

Ses sinyallerinin yakalanması ve kaydedilmesi için, ses dalgalarının mikrofona vurmasıyla oluşan analog ses sinyallerinin belirli aralıklar örneklenmesi ve dijitalize edilmesi gerekmektedir.

Örnekleme; ses sinyalinden belirli aralıklarla örnek çıkarılıp kesikli zaman sinyaline çevrilmesi anlamına gelmektedir ve tek düze alınan bu örneklerin orjinal sesi tekrar oluşturabilecek kadar sık olması gerekliliği vardır. Burada örnekleme için kullanılacak frekans 44100 olarak alınmıştır, insan kulağı ise 16000Hz'e kadar frekansları işitebilir. Alınan düşük frekans değeri veri kaybına ve bozulmaya neden olurken, gereğinden yüksek frekans değeriyse veri kalitesine etki etmez yani alınması kullanışsız/anlamsızdır. İdeal sıklık için, *Nyquist teoriminde* belirtildiği gibi *F* frekansını isabetli bir şekilde temsil edebilmek için en az *2F* sıklığında örnek alınmalıdır.

Örnekleme öncesindeyse elektriksel sinyalin çevre seslerden bir nebze olsun yalıtılabilmesi için, örnekleme frekansının yarısındaki tüm frekanslar filtrelenebilir. (Örneğin 44.1Khz kullanılıyorsa, 22050Hz üzeri frekanslar yalıtılmalıdır). Bu işleme *anti-aliasing* denilmektedir ve ses sinyallerinin insan kulağı tarafından duyulamayacak yükseklikteki çevre seslerden kaynaklı frekanslardan ayrılmasını sağlayarak, düşük frekanstaki seslerin bozulmasını (*distortion*) önler.

Tek-düze (*uniform*) yakalanan veride tablo index değerleri gerçek değerleriyle doğru orantılı olarak bulunur ve bu da doğrudan kullanılabilir. Bu çevrime, PCM kodlaması (*Linear PCM*) adı verilir. Ses yakalama iki ayrı modda gerçekleştirilmektedir. Bunlar; *blocking* adı verilen uygulamanın ses cihazından veri talep ettiği yada *callback* adı verilen uygulamanın ses cihazını izlediği ve yeterince veri olduğunda çağrıldığı modlardır. Ödev kapsamında *callback* modu, kullanıcının klavyede bir tuşa basmasıyla başlamakta ve otomatik son bulma ile sona ermektedir.

Otomatik son bulma ise, sesin var olduğu bölgenin sesin olmadığı bölgede ayrıştırılması işlemidir. Buna göre isabetli son bulma sesin nerede bittiğinin hatta hangi sesten oluştuğunun belirlenmesinde son derece önemlidir. Bu amaçla öne sürülen; ses seviyesi belirli bir eşik değerin altına belirli bir süre düştüğünde sonlanan *threshold-based endpointing*, belirli aralıklarla enerji hesabı yapılıp sürekli değişen bir arkaplan sinyal seviyesine göre düştüğünde sonlanan *adaptive energy-based endpointing* gibi birçok yaklaşım mevcuttur. Burada önemli olan, entropi yada enerji hesabı gibi dinamik yöntemlerle varlığını sürdüren ve seviyesi değişebilen arkaplan seslerinin ayırt edilip kayıt edilmek istenen ses sinyalinin gerçekten hangi anda bittiğinin isabetli ve doğru bir şekilde belirlenebilmesidir.

Ödev gerçekleştiriminde sesi yakalayıp kaydetmek için *pyAudio* adlı kütüphaneden faydalanılmış ve Python 2.7 ile yapılan gerçekleştirimde 16-bit PCM ve 44100Hz örneklem oranı kullanılmış, son bulma yöntemi olarak ise adaptif enerji hesabı dikkate alınmıştır. Ders slaytlarında da kaba kodu bulunan bu gerçekleştirimde kaydın başlamasıyla birlikte (genellikle ilk 5 ya da 10 çerçeve uzunluğu kadar) bir miktar ses verisinin ortalama enerjisi hesaplanır ve arkaplan seviyesi olarak değeri tutulur. Daha sonra üzerinde

işlem yapılan her çerçevede, enerji hesabı yapılıp arkaplan seviyesiyle karşılaştırılır. Güncel enerji seviyesi arkaplan seviyesinden düşükse, arkaplan seviyesi doğrudan bu değere kurulur. Tam tersi, güncel enerji seviyesi daha yüksekse, arkaplan seviyesi (*adjustment* parametresi oranınca) güncel enerji seviyesine bir miktar yaklaştırılır (adaptif ismi buradan geliyor). Unutma çarpanı (forget factor) ile düzeltilen enerji seviyesinden arkaplan seviyesi çıkartıldığında ön-tanımlı bir eşik değer geçiliyorsa, konuşma belirlenmiş olur. Aksi takdirde konuşmanın belirlenemediği n çerçeve boyunca uygulama çalışmaya devam eder ve en sonunda sonlandırılır.

3. Özniteliklerin çıkartılması ve MFCC

Öznitelik çıkarma konuşma tanımanın temel konularından biridir öyle ki sesi iyi tanımlayan özniteliklerin yakalanabilmesi, içeriğinin daha iyi tanımlanabilmesi dolayısıyla (daha sonraki adımlar da) üretilen/eğitilen fonemlerin de daha iyi gösterimi anlamına gelmektedir. Burada yapılan işlem kısaca, yalın sinyallerin konuşmacı özelliklerini yansıtan ancak diğer fazlalıklardan arındırılmış öznitelik vektörlerine çevrilmesi işlemi olarak özetlenebilir.

MFCC (Mel Frequency Cepstral Coefficients) konuşma tanımada sıklıkla kullanılan bir grup özniteliktir ve genel olarak özniteliklerin çıkarılması süreci insan kulağını taklit edecek adımları içerecek şekilde insan kulağının duyarlı olduğu ses frekanslarıyla ilgilenir.. Zamana bağlı olarak sinyaldeki çeşitli frekansların enerjilerinin gösterildiği grafik, *spectrogram* olarak adlandırılır. Bir sesin farklı örneklemlerinin birbirine benzer *spectrogramları* oluşturması yani aynı karakteristiğe sahip olması beklenir.

Bu tanımlar doğrultusunda, özniteliklerin çıkarılması için izlenen adımlar şu şekilde özetlenebilir; sinyalin spectrogramının çıkarılması ve ön-vurgu (pre-emphasis) denilen yöntem ile yüksek frekansın vurgulanması (sesli bölümlerin düşük frekanslarda yüksek frekanslardan daha çok enerjisi vardır), bu sayede daha fazla bilgi korunması amaçlanır.

Bir sonraki adım olarak genellikle birbiriyle (20ms boyunda) üst üste binmiş şekilde 30 ms'lik her birinde N örnek noktasının olduğu çerçevelerin oluşturulması sağlanır, burada çerçeve boyutu çok küçükse güvenilir bir spectral tahmin yürütülemeyecek kadar az örneğe sahip olunur, çok büyükse de sinyal tek bir çerçevede çok fazla değişerek en baştaki varsayım olan sinyalin yeterince küçük çerçevede durağan olduğu ilkesi çiğnenmiş olur.

Üçüncü adım olaraksa pencereleme (*windowing*) denilen yöntem ile DFT öncesi sinyalin smooth edilmesi sağlanır. DFT hesaplamasındaki varsayım girdi sinyalin sürekli bir şekilde tekrar ettiğidir, pencereleme yöntemiyle bu tekrarı bozan (*discontinuity*) kısımlardan kurtulunur. Bu sayede sinyalin iki ucu da sıfıra doğru yakınsarken, istenmeyen çıktılar engellenmiş olur. Pencereleme için konuşma tanımada genellikle *Hamming window* yöntemi kullanılmakla birlikte Hanning, Blackman, Gauss gibi farklı fonksiyonlar da vardır.

Daha sonra, N örnekten oluşan her bir çerçevenin işlenebilmesi adına zaman alanından (time domain) frekans alanına çevrimi için FFT (Fast Fourier Transform) / DFT (Discrete Fourier Transform) kullanılır.

İnsan kulağının tüm frekanslara hassas olmadığı, yüksek frekanslarda daha az duyduğu (>1Khz) bilinmektedir. Mel bir perde (pitch) birimi olarak tanımlanır, Mel-frekans ölçeği ise yaklaşık olarak 1KHz'e kadar lineer artmakta ve daha yüksek frekanslarda logaritmik olarak ilerleyerek aslında bir bakıma insan kulağının davranışına karşılık gelmektedir. İnsan kulağının sadece belli frekanslara odaklanmasını sağlayan filtreler ise *band-pass filter* olarak adlandırılır. Bu filtreler tek-düze olmayan şekilde dağılmıştır, örneğin düşük frekanslı bölgelerde daha fazla filtre, yüksek frekanslı bölgelerde daha az filtre mevcuttur.

Buna göre Mel-ölceğine göre spectruma uygulanan filtre bankaları sonucu elde edilen çıktı üzerinden de Log enerjisi hesaplanır. Buradaki logaritma hesabı sayesinde yine insan kulağının davranışına benzer olarak girdideki küçük değişimlerde frekans hesaplamaları daha az etkilenir.

Son adım olarak da ters-DFT alınarak log Mel spectrum değeri tekrar zaman alanına çevrilir, elde edilen sonuç *Mel Frequency Cepstrum Coefficient* olarak adlandırılmakta ve söz konusu katsayılar

akustik vektörler olarak tanımlanmaktadır. Sonuç olarak söz konusu hesaplamalarla, ses sinyali girdisi bir dizi akustik öznitelik vektörüne çevrilmiş olur.

4. DTW (Dinamik Zaman Saptırması) - İzole kelime tanıma

Dinamik zaman saptırması, aynı sınıfa ait farklı zaman-ölçekli varyasyonların örüntü eşlemesi için kullanılan bir yöntemdir. Yani daha önceki ödevde elde edilen öznitelik vektörleri göz önünde bulundurulduğunda, aynı sese/kelimeye karşılık gelen iki sıra vektörün hizalanması için zaman ekseninin saptırılması işlemi olarak nitelendirilebilir. Bu sayede, örneğin, öznitelik vektörlerinin aynı kelimeye işaret edip etmediği belirlenebilir. Matematiksel olarak yapılan işlem basitçe, iki vektör dizisinin bir gridin üst ve sol olmak üzere iki kenarına yerleştirilmesi (*trellis*) ve her bir grid hücresinde uzaklık hesabı yapılarak değerlerin karşılaştırılması işlemidir. Uzaklık hesabı için genellikle Öklid kullanılmakla birlikte Manhattan, Minkowski gibi diğer fonksiyonlar da kullanılmaktadır. Bu iki vektörün eşlenmesi yada hizalanması denilen aslında aralarında toplam uzaklığın minimize edildiği (dinamik programlama) grid hücrelerinden geçerek oluşturulan yoldur.

Örneğin ödevdekine benzer şekilde, izole kelime tanıması (isolated word recognition) için kullanılacak bir DTW gerçekleştiriminde belirli kelimeler için daha önceden verilen şablonlar (template - örnek kayıt) kullanılarak basit bir tanıma sistemi geliştirilebilir. Bu sayede girdi verisinin tüm şablonlara olan uzaklıkları hesaplanarak, en kısa olanıyla eşleştirilerek tanıma sağlanmaktadır.

Tahmin edileceği gibi bir kelime için ne kadar çok şablon kullanılırsa (hatta farklı konuşmacılar tarafından oluşturulmuş şablonlar daha iyidir çünkü aynı sesin farklı özelliklerini içerebilirler) tanıma işlemi o kadar başarılıdır ancak tanıma işlemi için geçen hesaplama süresi de doğrusal bir şekilde artar. Hesaplama süresinin azaltılması amacıyla budama (pruning) denilen bir yöntem kullanılabilir. Bu yönteme göre, girdi ile en iyi eşleşen şablonun birbirinden çok da farklı olmadığı ve daha önce bahsedilen trelliste yol bulma işleminde en iyi yolun köşegene yakın olacağı varsayılır. Bu bağlamda köşegenden sadece belirli uzaklıkta arama yapılarak, diğer hücreler yok sayılır. Buna *fixed-width pruning* adı verilirken, yüksek maliyete sahip kısmi yolların zamanla en iyi yola yakınsayamacağı varsayımını güden budama yöntemineyse *pruning by liminting the path cost* adı verilebilir.

5. Fonemler - modelin eğitilmesi

Konuşma tanımanın eğitilen model ile nasıl gerçekleştirildiğini incelemeden önce ilişkili bazı terimlerin tanımlanması yardımcı olacaktır. Fonem; temel ses birimi olarak tanımlanabilir. Örneğin “cat” kelimesi “K AE T” fonemlerinden oluşmaktadır. Akustik model; girdi olarak ses kaydı alıp, çıktı olarak fonemler üzerinden dağılımını gösteren bir modeldir. Sözlük (*dictionary* veya *lexicon*) ise sözcüklere karşılık fonemlerinin listelendiği tanımları içerir. Konuşma tanıma sistemlerinin bir kelimenin okunuşuna dair bilgiyi elde edebilmesinde kullanılan araçtır, bir bakıma kelimeden okunuşa yada foneme bir çevrim sunar. Bu kapsamda akustik model ile elde edilen fonemler sözlükteki tanımlar yardımıyla kelimelere çevrilir. Dil modeli ise girdi olarak bir cümle alıp, onun var olma olasılığını gözler. Öyle ki “the cat in the hat” cümlesi “cat the in hat the” cümlesinden çok daha olasıdır ve iyi bir dil modeli ilk cümleye ikincisine kıyasla çok daha yüksek bir olasılık değeri atayacaktır.

Bu tanımlar doğrultusunda akustik modelin işleyişinde öncelikle çerçevelere bölünen ses kaydının sinyal işleme kullanılarak öz niteliklerinin çıkarılması (40 mel-sıklığı kepstral katsayıları gibi) daha sonra da olaslılıksal (örneğin HMM veya nöral ağ gibi) bir modele sunularak her fonemin olasılıklarının bulunması rol oynar. Burada modelin eğitilmesi burada saymadıklarımla birlikte kullanılan verinin çokluğuna ve kalitesine son derece bağımlıdır. Örneğin sadece genç erkek bir konuşmacının ses kayıtlarıyla eğitilen bir model bir kadın veya yaşlı bir şahsın sesinin tanımlama da kötü başarım gösterecektir. Benzer şekilde kaliteli bir mikrofon ve gürültüsüz kayıttan elde edilen model gürültülü ses sinyalinde aksi duruma kıyasla başarısız olacaktır. Tüm bunlara bağlı olarak akustik modelin eğitilmesinin uzun ve zahmetli bir iş olduğu söylenebilir. Yüksek başarımlı bir sunucuda günlerce ya da haftalarca

eğitilen bir modelin gösterdiği başarım ev bilgisayarların da az miktardaki veriyle birkaç saatliğine yada günlüğüne eğitilmiş modelinkinden çok daha yüksektir.

DTW ile elde edilen model izole kelimeyi içeren ses kaydını girdi olarak alan ve sözlüğünde tuttuğu şablon kayıtları bu girdi ile karşılaştırarak en olası (yani girdi ile şablon kelime arasındaki fark en küçük olan / min.cost) kayıt seçilerek izole kelime tanınmış olur. Bu yaklaşımı sürekli konuşma tanımaya uyarlamak istediğimizde ortaya çıkan sorun; her kelime için modelin eğitilmesi gerekliliğidir. Gerçek bir ses tanıma sisteminde bunu eğitmek kelimelerin sayısının çok fazla olmasından dolayı neredeyse imkansızken, sisteme yeni katılan her kelime için de model yeniden eğitilmelidir. Dolayısıyla konuşma tanımada kelime altı bazı birimlere ihtiyaç doğar. Bu birimlere fonem adı verilmektedir ve her dilde sadece belirli sayıda fonem vardır. Örneğin İngilizce'de yaklaşık 40 fonem mevcuttur ve bu fonemlerin bir araya gelmesiyle dildeki herhangi bir ses oluşturulabilir. Dolayısıyla her bir kelime ya da kelime grubu için bir dil akustik model eğitmek yerine eldeki fonemler üzerinden model eğitilebilir. Bu bağlamda örneğin İngilizce için 40 adet HMM eğitildiği takdirde herhangi bir sesin tanınmasının önü açılmış olur. Fonemler üzerinden eğitilen HMM'ler bir araya getirilerek kelime düzeyine çıkarılmış olur yani tanıma yine kelime üzerinden yapılmaktadır diyebiliriz. Burada eğitilen akustik model konuşmayı tanımada kullanılabilse de tek başına yeterli değildir. Çünkü HMM'lere göre bir kelimeden sonra herhangi bir kelimenin gelmesi de gayet olasıdır. Burada gerçek bir dili yansıtabilmek adına, hangi kelimelerin nasıl gruplandığı hangi kelimeden sonra nelerin olası gelebileceği gibi kısıtlar eğitilen dil modeli sayesinde sağlanır.

Dil modeli konuşma tanımanın başarılı bir şekilde çalışıp çalışamadığının belirleyicisi rolündedir. Genellikle N-gram modelinin kullanıldığı yaklaşımlarda (genellikle tri-gram), eğitim verisinde her bir 3 kelime sayılarak her bir üç kelimenin olasılığı, o üç kelimelik dizilimin bulunma olasılığı bölü eğitim verisindeki dizilimlerin sayısı kadardır. Dil modeli bol miktarda veri ile eğitilerek kelimelerin var olma olasılıkları öğrenilmiş olur ve akustik modelin arama uzayını daraltmak amacıyla, yani daha iyi tanıma için kullanılmaktadır.