

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 ödevde 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 (Endpointing)

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 orijinal sesi tekrar oluşturabilecek kadar sık olması gerekliliği vardır. Burada örnekleme için kullanılacak frekans 44100 olarak alınmıştır, insan kulağı ise 16000Hz'e kadar frekansları işitebilir. Burada düşük frekans değeri veri kaybına ve bozulmaya neden olurken, gereğinden yüksek frekans değeriye veri kalitesine etki etmez yani alınması kullanışsızdır. İdeal sıklık için, Nyquist teorisinde belirtildiği gibi F frekansını isabetli bir şekilde temsil edebilmek için en az $2F$ sıklığında örnek alınmalıdır.

Örnekleme öncesindeyse elektriksel sinyalin çevre seslerden bir adım 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 (distort) önler.

Tek-düze (uniform) yakalanan veride tablo index değerleri gerçek değerleriyle doğru orantılı olarak bulunur ve bu da doğrudan kullanılabilir. Bu çevrime, PCM kodlaması (yada 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 sestten oluştuğunun belirlenmesinde son derece önemlidir. Bu amaçla öne sürülen; ses seviyesi belirli bir eşik değerin altına belirli bir süre düştüğünde sonlanan *threshold-based endpointing*, sinyal seviyesi belirli aralıklarla enerji hesabı yapıp 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 16-bit PCM ve 44100Hz örnekleme oranı kullanılmış, son bulma yöntemi olarak ise enerji hesabı dikkate alınmıştır.

3. Özniteliklerin Çıkartılması ve MFCC

Öznitelik çıkarma konuşma tanımının temel konularından biridir öyle ki sesi iyi tanımlayan özniteliklerin yakalanabilmesi içeriğinin daha iyi tanımlanabilmesi dolayısıyla üretilen 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 tanımda sıklıkla kullanılan bir grup özniteliktir. İnsan kulağının duyamayacağı 1Khz üzerindeki sesleri algılayamaz yani sadece insanın algılayabileceği aralıkla ilgilenir. Zamana bağlı olarak sinyaldeki çeşitli frekansların enerjilerinin gösterildiği grafik, spectrogram olarak adlandırılır. Bir sesin farklı örneklemelerinin 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 ucuda sıfıra doğru yakınsarken, istenmeyen çıktılar engellenmiş olur. Pencereleme için konuşma tanımda genellikle Hamming window 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-ölçeğine 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 (coefficients) akustik vektörler olarak tanımlanmaktadır. Sonuç olarak söz konusu hesaplamalarla, girdiler bir dizi akustik vektöre çevrilmiş olur.

4. DTW (Dinamik Zaman Saptırması)

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 ve her bir grid hücresinde uzaklık hesabı yapılarak değerlerin

karşılaştırılması işlemidir. 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 tanınması (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 yöntemde göre, girdi ile en iyi eşleşen şablonun birbirinden çok da farklı olmadığı ve daha önce bahsedilen gridde yol bulma işleminde en iy yolun gridin köşegenine 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 limiting the path cost* adı verilebilir.