

KONUŞMA TANIMA İÇİN HETEROSKEDASTİK AYIRTAÇ ANALİZİNİN DÜZENLİLEŞTİRİLMESİ REGULARIZING HETEROSCHEDASTIC DISCRIMINANT ANALYSIS FOR SPEECH RECOGNITION

Hakan Erdoğan

Faculty of Engineering and Natural Sciences
Sabanci University
Orhanli Tuzla 34956 Istanbul Turkey
haerdogan@sabanciuniv.edu

Özetçe

Doğrusal Ayırtaç Analizi (LDA) ve bunu takiben ortak değişintiyi köşegenleştiren bir enbüyük-olabilirlik dönüşümü matrisi (maximum likelihood linear transform - MLLT) kullanımının birçok durumda konuşma tanıma performansını iyileştirdiği gösterilmiştir. Biz bu makalede LDA ve heteroskedastik LDA dönüşümlerinin düzenlenileştirilmeleri için iki değişik yol öneriyoruz: (1) Dönüşüm matrisi için istatistiksel önsel olasılık kullanımı (2) Dönüşüm matrisi üzerinde (blok yapısı gibi) yapısal sınırlamalar getirilmesi. Yapısal sınırlamamız ise bir blok yapılı dönüşüm matrisini zorlamaktır. Önerdiğimiz ikinci metot, statik, birinci ve ikinci fark olarak daha farklı operatörlerin kullanımını sadece veriden yola çıkarak önermekte ve başarımları böylece arttırmaktadır. Yeni algoritmaları iki değişik standart İngilizce veri bankası üzerinde denedik: TIMIT ve AURORA2. Her seferinde MFCC özniteliklerinden daha iyi başarımlar elde edebildik. AURORA2 testlerinde ise bazı gürültü seviyelerinde LDA+MLLT metoduna göre iyileştirme sağladık.

ABSTRACT

Linear Discriminant Analysis (LDA) followed by a diagonalizing maximum likelihood linear transform (MLLT) applied to spliced static MFCC features yields important performance gains as compared to MFCC+dynamic features in most speech recognition tasks. It is reasonable to regularize LDA transform computation for stability. In this paper, we regularize LDA and heteroschedastic LDA transforms using two methods: (1) Statistical priors for estimating the transform, (2) Structural constraints on the transform. Our structural constraint imposes a block structured LDA transform where each block acts on the same cepstral parameters across frames. The second approach suggests using new coefficients for static, first difference and second difference operators as compared to the standard ones. We test the new algorithms on two different tasks, TIMIT and AURORA2. We obtain consistent improvement over standard MFCC features. We also improve upon LDA+MLLT features for certain noise levels in AURORA2 tests.

1. GİRİŞ

Örüntü tanıma sistemlerinin ana parçalarından birisi de öznitelik çıkartıcıdır. Zaman ekseninde ses sinyali çok fazla değişkenlik gösterdiğinden sinyalin kendi değerlerini öznitelik olarak kullanmak mümkün değildir. Bunun yerine, sinyal birtakım karmaşık veri azaltıcı ve değişkenlikten fazla etkilenmeyecek işlemlerden geçirilir. Ses sinyali zamanda 20-30 ms uzunluğunda ve birbiriyle örtüşen çerçevelere ayrılır ve her bir çerçeveden statik öznitelikler çıkarılır. Konuşma tanımadaki ayrıca dinamik özniteliklerin kullanımının da faydalı olduğu gözlemlenmiştir [1]. Δ ve $\Delta\Delta$ öznitelikleri diye adlandırılan bu öznitelikler statik özniteliklerin birinci ve ikinci türevi yakınsanarak elde edilirler. Türevi

yakınsayan farklar bulunurken belli bir komşuluktaki komşu çerçeveler kullanılır.

Komşu çerçeveleri bir araya getirip büyük bir vektör oluşturulduktan sonra bu büyük vektöre doğrusal ayırtaç analizi (linear discriminant analysis - LDA) uygulanmasıyla yüksek boyutlu daha düşük boyuta düşüren bir doğrusal dönüşüm elde edilir. Bu dönüşüm bir komşuluktaki dinamik değişim bilgisini otomatik olarak çıkartma görevi de üstlenir. LDA algoritmasında ölçüt varsayılan ses sınıflarını en fazla ayırtıcı özellik taşıyan boyutların seçilmesidir. LDA ile elde edilen öznitelikler aslında statik, Δ ve $\Delta\Delta$ özniteliklerinin rakipleridir ve istatistiksel olarak elde edilmişlerdir. LDA varsayılan her sınıfın sınıf-İçi kovaryans matrisinin aynı olduğunu farzeder. Heteroskedastik LDA (HLDA) [2] ise değişik sınıf-İçi ortak-değişintilere izin veren algoritmanın adıdır. LDA dönüşümünün işe yaraması için çoğu zaman onu takiben sınıf-İçi ortak-değişintileri daha köşegen yapacak bir doğrusal dönüşüm matrisi bulunur ve bu matris de LDA matrisinden sonra uygulanır. Bu dönüşüme de enbüyük-olabilirlik doğrusal dönüşümü (maximum likelihood linear transform - MLLT) denir [3]. HLDA dönüşümünde ise MLLT adımına gerek yoktur, çünkü HLDA metodu aynı anda hem boyut düşürmeye hem de sınıf-İçi ortak-değişintileri (covariance) köşegenleştirmeye (diagonalize) çalışır [2].

Bu bildiriye, LDA ve HLDA için düzenleme metotlarını sunmaktayız. İki değişik metot geliştirdik. Birincisinde, Bayeşçi bir bakış açısıyla problemi değerlendirerek dönüşüm matrislerini kestirmeye çalışıyoruz. İkinci metot ise LDA dönüşüm matrisinin yapısını belli blok bir yapı ile sınırlama temeline dayanıyor. Bölüm 2'de, Bayeşçi çözümümüzü sunuyoruz. Blok-yapılı LDA metodumuzu ise bölüm 3'de anlatıyoruz. 4. bölümde TIMIT ve AURORA2 veritabanlarındaki deneysel sonuçlarımızdan bahsetmekteyiz. Son olarak, sonuçlarımızı bölüm 5 altında dillendiriyoruz.

2. DÜZENLİLEŞTİRİLMİŞ HLDA

Doğrusal ayırtaç analizi Fisher oranını enbüyükleyerek elde edilir [4]. Sınıfların Gauss dağılımında olduğu varsayılırsa, aynı çözüme enbüyük-olabilirlik formülü ile de varılabilir [2]. Bu düzenlemede özniteliklere uygulanacak kare bir A matrisi bulunmaya çalışılır. Bu matrisle dönüşümden sonra bütün ayırtıcı bilginin ilk p boyutta kalması amaçlanır. Buna erişmek için, dönüştürmeden sonraki vektörde yer alan son $n - p$ boyutun bütün sınıflarda aynı ortalama vektörü ve ortak-değişinti matrisine sahip olması zorlanır [2]. Bu

durumda eğer bütün sınıflar aynı sınıf-içi ortak-değişimtiye sahip varsayılırsa LDA çözümü değişik bir yoldan elde edilir. Her bir sınıfın kendi ayrı sınıf-içi ortak-değişimtisine sahip olmasına izin verilince de, HLDA dönüşümü elde edilir [2]. HLDA algoritmasının türetilmesini bir gözden geçirelim.

Diyelim ki $x_i \in \mathbb{R}^n : i = 1 \dots N$ orjinal uzaydaki öznitelik vektörleri olsun. Ayrıca, her bir x_i 'in hangi sınıfa ait olduğunu gösteren $c_i = j \in 1, \dots, J$ değişkeni olsun. $y = A_p x$ şeklinde bir dönüşüm bulmak istiyoruz. Burada $A_p : \mathbb{R}^n \rightarrow \mathbb{R}^p$ ve $p < n$ olmalı. y 'leri öyle seçmek istiyoruz ki x 'lerdeki sınıf ayırıcı bilginin çoğu kendisinde kalsın. Enbüyük-olabilirlik denklemleri için $n - p$ satırlı A_{n-p} matrisini A_p matrisine ekleyerek

$$A = \begin{bmatrix} A_p \\ A_{n-p} \end{bmatrix},$$

dönüşümünü elde edilir. Dönüşmüş sınıf öznitelikleri Gauss dağılımı gösterebilir ve köşegen ortak-değişimtiye sahip olsun varsayılır. Ayrıca son $n - p$ boyuttaki ortalama ve ortak-değişimti değerleri de bütün sınıflar için aynı olmalıdır. Bu varsayımlar ve model kısıtları altında eğitim verisini enbüyükleyen bir A matrisi bulunması amaçlanır.

Eğitim verisinin olabilirliği A 'nın bir fonksiyonu olarak şöyle yazılabilir [2, 5]:

$$L(A) = \sum_{j=1}^J \frac{N_j}{2N} \log \frac{|A|^2}{|\text{diag}(A_p W_j A_p)| |\text{diag}(A_{n-p} T A_{n-p})|}.$$

Burada

$$W_j = \frac{1}{N_j} \sum_{c_i=j} (x_i - \mu_j)(x_i - \mu_j)^T$$

kestirilen sınıf-içi ortak-değişimtileri ve

$$T = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T$$

ise eğitim verisinin toplam ortak-değişimtisini gösterir. μ_j sınıf ortalamaları ve μ ise genel ortalamadır.

Olabilirlik fonksiyonunun düz metotlarla enbüyüklenmesi mümkün değildir ve tekrarlamalı (iterative) çözümlere gerek duyulur. Olabilirlik dışbükey (convex) olmadığından ve birçok çözüm olabildiğinden tekrarlamalı çözümleri geliştirmek de zordur. Kumar çözüm için bir en-keskin iniş (steepest descent) algoritması kullanılmıştır [2]. Gales ise daha hızlı çalışan ve matris satırlarını güncelleyen bir algoritma önerir [5]. Olabilirlik fonksiyonunu A 'nın satırları cinsinden tekrar yazarak o çözüme ulaşabiliriz.

$$L(A) = \log(a_r^T c_r) - \frac{1}{2} \sum_{r=p+1}^n \log a_r^T T a_r \quad (1)$$

$$- \frac{1}{2} \sum_{j=1}^J \frac{N_j}{N} \sum_{r=1}^p \log a_r^T W_j a_r$$

denkleminde $a_r^T A$ 'nın r 'nci satırını gösterir ¹ ve c_r^T de a_r^T 'nin kofaktörüdür. Denklemdaki ilk terim herhangi bir r satırını kullanarak yazılabilir.

Bu bölümde amacımız A 'nın kestirimi için Bayeşçi bir denklem yazmaktır. Bu amaçla A için bir önsel olasılık dağılımı varsaymak zorundayız. Kolaylık olması için A 'nın dağılımı olarak köşegen

ortak-değişimtili bir Gauss dağılımı öngörüyoruz. Bu durumda sonsal (aposteriori) olabilirlik şöyle yazılabilir:

$$\Phi(A) = -L(A) + 1/2 \sum_{r=1}^n (a_r - \bar{a}_r)^T P_r (a_r - \bar{a}_r).$$

Burada \bar{a}_r ve P_r sırasıyla a_r vektörünün önsel ortalamasını ve önsel ortak-değişimti matrisinin tersini (ya da kesinlik matrisi) gösterir. Yukarıdaki hedef fonksiyonun enküçültülmesi (minimizasyon) gereklidir.

Hedef fonksiyonun a_r 'ye göre gradyanı kolayca aşağıdaki gibi hesaplanır:

$$\nabla_{a_r} \Phi = P_r (a_r - \bar{a}_r) - \frac{c_r}{a_r^T c_r} \quad (2)$$

$$+ \begin{cases} \sum_{j=1}^J \frac{N_j}{N} \frac{W_j a_r}{a_r^T W_j a_r} & r \leq p \\ \frac{T a_r}{a_r^T T a_r} & r > p \end{cases}.$$

Enküçükleme problemini çözmek için $\nabla_{a_r} \Phi = 0$ denklemi her a_r için çözümlenmelidir. Tekrarlamalı yöntemler kullanılmadan bunu başarmak olası değildir. Biz de Gales [5] benzeri bir teknik ile $a_r^T c_r$, $a_r^T W_j a_r$ ve $a_r^T T a_r$ terimlerinin döngüden döngüye fazla değişmediğini varsayarak önceki değerlerini denkleme yerleştirip a_r 'yi $\nabla_{a_r} \Phi = 0$ denkleminde çekebiliriz. Bu metot aşağıdaki basit algoritmayı doğurur.

Başlat: $A = \bar{A}$

Yakınsamadıkça:

Her bir $r = 1, \dots, n$ için

$$\text{Hesapla } G_r = \begin{cases} \sum_{j=1}^J \frac{N_j}{N} \frac{W_j}{a_r^T W_j a_r} & r \leq p \\ \frac{T}{a_r^T T a_r} & r > p \end{cases}$$

$$\text{Hesapla: } \alpha_r = (a_r^T c_r)^{-1} = |A|^{-1}$$

$$\text{Güncelle: } a_r = (G_r + P_r)^{-1} (\alpha_r c_r + P_r \bar{a}_r)$$

döngü sonu

döngü sonu

Dikkat edilirse bu algoritmanın [5]'den biraz farklı olduğu görülür. Fakat, bu algoritma önsel bilgi olmadığında ($P_r = 0$) [5]'deki ile aynı doğrultuda bir a_r oluşturur. Bu da iyidir çünkü A 'nın satırlarını katlamak enbüyük olabilirlik hedef fonksiyonunu değiştirmez [2].

Sınıf-içi ortak-değişimtiler, W_j , eşit kabul edildiklerinde ve ortak olan W da bunların ağırlıklı ortalamaları olarak hesap edildiğinde, yukarıdaki algorithmandan bilinen LDA çözümü elde edilir [2]. Tabii diğer durumlarda ve önsel bilgi olduğunda yukarıdaki algoritma enbüyük sonsal olabilirlik (maximum a posteriori probability - MAP) hedef fonksiyonunu eniyiler.

Genelde, önsel ortalama olarak $p \times n$ 'lik bir matrisimiz elimizde olur. Bu durumda $r > p$ olan satırlar için $P_r = 0$ kullanılarak p 'den yüksek satırlar için önsel bilgi kullanılmamış olur. $r \leq p$ satırları için, $P_r = \beta I$ şeklinde basit ve sabit bir kesinlik matrisi kullanıyoruz. Diğer kesinlik matrisleri kullanmak da mümkündür.

Önsel ortalama dönüşümü olarak birçok olası aday vardır. Biz Statik+ Δ + $\Delta\Delta$ (S+D+DD) dönüşümünü bu çalışmada önsel ortalama olarak kullanıyoruz. Onüç boyutlu bir statik öznitelik vektörü alndığı ve statik vektörlerin yedisi ardi ardına sıralandığında ve HTK 3.2 yazılımında DELTAWINDOW=2 ve ACCWINDOW=1

¹Bütün vektörler sütun vektörüdür

değerleri kullanıldığında aşağıdaki önsel ortalama matrisi elde edilir.

$$\bar{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & -1 & 0 & 1 & 2 & 0 \\ 2 & 1 & -2 & -2 & -2 & 1 & 2 \end{bmatrix} \cdot \otimes I_{13 \times 13}$$

Burada \otimes sembolü Kronecker çarpımını gösterir. Bu dönüşümün iyi sonuçlar verdiğini biliyoruz, o halde bunun önsel ortalama olarak kullanımı makuldür. Aslında diğer çözümler de olabilir. Örneğin, daha küçük bir komşuluk kullanarak bulunmuş bir LDA dönüşümü gerekli sıfırlar eklenerek önsel dağılımda kullanılabilir. Burada amaç şu olacaktır: Küçük komşulukta hesaplanan LDA daha güvenilir olacak ve yüksek komşuluklara çıkıldığında hesaplanacak LDA matrisinin numerik olarak daha düzenli olmasını sağlayacaktır. Bilindiği gibi yüksek boyutlara çıkıldıkça ortak-değişinti matrisleri numerik olarak daha az güvenilir olacaklardır.

Denediğimiz diğer bir düzenleme yöntemi de blok-yapılı LDA dönüşümleri bulunmasıdır. Aşağıda bu metodu detaylandırıyoruz.

3. BLOK YAPILI LDA

A_p matrisinin yapısını belli bir şekilde olmaya zorlamak da bir nevi düzenleme yoludur. Örneğin, A_p matrisinde bazı katsayıları sıfır olmaya zorlamak veya bazı katsayıların diğerleri ile aynı olmasını zorlamak matrisin kestirimini düzenleyecektir. Bu tür kısıtlardan biri de A_p matrisinin blok yapılı olmasını zorlamaktır.

Bu bildiriye biz basit bir blok-yapılı LDA dönüşümü teklif ediyoruz. Orijinal boyutları gruplara ayırıp her gruba ayrı boyut düşürücü dönüşüm uygulanmaktadır. Bu aslında grup dışında kalan boyutlara denk gelen diğer sütunlardaki A_p katsayılarının sıfıra zorlanması anlamına geliyor. Bu yapı geçen bölümde bahsedilen S+D+DD matrisine benzer bir yapı ortaya çıkarmaktadır. Fakat aradaki fark, bu metotla elde edilen katsayıların otomatik olarak bulunması ve farklı katsayılar yol açmasıdır.

Böyle bir dönüşümün bulunması aslında oldukça kolaydır. Boyut gruplarına karar verip her grubun kaç alt boyuta indirgeneceği belirlenir. Sonrasında, W and T matrislerinin sadece o boyutları kapsayan satır ve sütunları alınır. Bundan sonrası aynen LDA veya HLDA matrisi bulunmasına benzer işlemler gerektirir ve daha düşük boyutlu LDA ve HLDA matrisleri bulunur. En son, A_p matrisini oluşturmak için her grup için bulunan farklı dönüşümler alt alta eklenir.

S+D+DD matrisinin yapısı göz önünde bulundurulursa, boyut gruplarını aynı statik parametrenin değişik çerçevelerdeki karşılıklarının bir grup oluşturmaları mantıklı bir seçimdir. Böylece, 13 statik parametre ve 7 tane komşu çerçeve alındığında, her birinde 7 tane boyut bulunan 13 tane grup elde ederiz. Her 7 boyutluk grubu 3 boyuta indirgersek sonuçta 39×91 'lik bir A_p matrisi elde ederiz. Bu şekilde elde edilen matris oldukça sıfırlıdır (sparse). Bu metot açıkça statik, birinci fark ve ikinci fark operatörlerinin yerine geçen katsayılar üretmenin değişik ve veri tabanlı bir yoludur. Biz yaptığımız çalışmalarda gördük ki elde ettiğimiz katsayılar gerçekten de statik, birinci ve ikinci fark katsayılarına oldukça yakın ve benzer çıkmaktadır. Deneyler bölümünde bu sonuçlardan bahsedeceğiz.

4. DENEYSEL SONUÇLAR

Geliştirdiğimiz metotları TIMIT and AURORA2 adlı iki değişik veritabanında uyguladık.

Features	Accuracy	Correct Detection
MFCC	62.92%	80.34%
LDA+MLLT	70.59%	82.34%
HLDA	68.57%	81.49%
HLDA-MAP+MLLT	69.25%	82.07%
Block LDA	68.05%	80.46%

Table 1. 39 boyutlu öznelikler ile TIMIT veritabanı üzerinde fonem tanıma başarımları.

TIMIT veritabanı üzerinde fonem tanıma deneyleri yaptık [6]. TIMIT'te bulunan 64 detaylı fonemi [6]'de yapıldığı gibi 48 taneeye düşürdük. Böylece 48 tane tekli-ses modeli oluşturuldu. Performans değerlendirirken de 48 tane fonem 39 taneeye tipik olarak yapıldığı gibi indirildi [6].

Her biri otuz dokuz boyutlu değişik özneliklerden bağlı durumlu üçlü-ses modelleri oluşturduk. MFCC öznelikleri standart 12 kepra + enerji ve Δ ve $\Delta\Delta$ dinamik özneliklerinden oluşuyor. LDA tipi öznelikler ise 7 komşu çerçeveden alınan 91 boyutlu büyük öznelik vektörünün LDA dönüşümü uygulanarak 39 boyuta indirgenmesi ile oluşturuldu. LDA sonrasında MLLT de uygulandı. HLDA-MAP+MLLT metodunda, S+D+DD dönüşümü önsel ortalama olarak kullanıldı ve $P_r = 1000I$, $r \leq p$ için kullanıldı. Blok-yapılı LDA için ise yukarıda açıklandığı gibi S+D+DD matrisine benzer yapıda bloklar kullanıldı. Sınıf tanımı olarak her bir HMM durumu kullanıldı. Sınıf-İçi ortak-değişinti matrisleri W_j ve toplam kovaryans matrisi T ise Viterbi algoritması ile hizalanmış (aligned) eğitim verisinden hesaplandı.

Sonuçlar Tablo 1'de gösterilmektedir. En iyi sonuç LDA+MLLT öznelikleri ile elde edildi. HLDA metodu aslında LDA'den teorik olarak daha iyi sonuç vermelidir. Ama bu sadece kovaryanslar tamamen bilindiğinde geçerlidir. Fakat, biz kovaryansları sınırlı sayıda veriden elde ettiğimiz için HLDA çok da güvenilir olmayan W_j kestirimleri kullanmak zorunda kalmakta ve daha kötü başarımlar göstermektedir. Aslında Monte-Carlo simülasyonları ile denediğimizde ve kovaryanslar gerçek olarak bilindiğinde HLDA'nin LDA+MLLT'den daha iyi sonuç verdiğini gözlemledik. Bayesçi ve blok-yapılı yaklaşımlar da LDA+MLLT başarımlarını yakalayamamaktadır, fakat MFCC temel çizgisini geçmektedirler. Bu sonuçları eğitim ve test verilerinde gürültü ve kanal uyumsuzluğu olmamasına bağlıyoruz ve bu test için düzenleme yapmanın en azından bu boyutlar için belki de gerekli olmadığını düşünüyoruz.

AURORA2, gürültüye dayanıklı tanıma algoritmalarının denemesi için oluşturulmuş, ETSI tarafından dağıtılan standart bir veritabanıdır. Bu veritabanındaki veriler üzerlerine değişik miktarda ve tipte gürültü eklenmiş rakam dizilerinin seslendirilmelerinden oluşur. Bu veritabanında TIDIGITS veritabanından alınan gürültüsüz seslere ayrıca yapay olarak gürültü eklenmiştir. ETSI, MFCC özneliklerine göre daha iyi başarımlar gösteren ve adı "ileri seviye dağıtık konuşma tanıma önışlemesi" olarak anılan bir öznelik çıkartma algoritması geliştirmiştir (ES 202 050). Bu metot gürültü altında MFCC'ye göre çok iyi başarımlar sağlamaktadır. Biz bu yeni önışleme ile çıkarılan özneliklere AFE öznelikleri dedik. AURORA2 veritabanında yaptığımız deneyler sadece temiz eğitim verisi kullanılarak yapılmıştır.

Tablo 2 değişik öznelikler ile farklı gürültü seviyelerindeki başarımlarını göstermektedir. MFCC ve AFE öznelikleri önceden bahsedilen öznelikler. LDA tipi özneliklerimizi ise AFE öznelikleri temel olarak hesapladık. Yani statik öznelik olarak AFE kullandık. Her HMM durumu gene bir sınıf olarak kullanıldı. Bir kez daha, LDA+MLLT öznelikleri temel özneliklere nazaran

Oznitelikler/SNR(dB)	Temiz	20	15	10	5	0	-5
MFCC	99.0	94.1	85.0	65.5	38.6	17.1	8.5
AFE-MFCC	99.1	98.0	96.5	92.4	82.3	58.2	27.2
LDA+MLLT	99.3	98.3	97.1	93.4	83.1	58.7	27.2
HLDA	98.4	96.6	94.4	85.4	60.4	27.8	12.1
HLDA-MAP+MLLT	99.3	98.2	96.8	92.8	81.4	54.9	22.3
Block LDA	99.2	97.9	96.6	92.8	83.0	60.3	28.8

Table 2. AURORA2 veritabanında temiz veri ile eğitilen konuşma tanıma sistemlerinin doğruluk oranları. Sonuçlar bütün testler ve bütün gürültü tiplerinin ortalamasıdır. Her SNR verisi için referans sözcük sayısı 32883'dür.

başarımı arttırmıştır. HLDA LDA+MLLT'ye göre çok daha kötü başarımlar göstermiştir. Bunun sebebini gene sınıf-ıçi kovaryans kestirimlerinin doğru olmamasına bağlıyoruz. Düşük sinyal-gürültü oranı (SNR) koşullarında blok-yapılı LDA diğer metodlardan daha iyi başarımlar sağlamıştır. Bu sonuç, blok LDA metodunun eğitim-test verilerindeki uyumsuzluklara daha az duyarlı olduğunu göstermiştir.

Son olarak, Figür 1'de, blok LDA metodu için, her statik öznelik için 7'şer tane birincil katsayıları birbiri üstüne çizilmiş olarak görebilirsiniz. Bu katsayılar her çerçevede karşılık gelen statik parametre ile çarpılacaklardır. Bu çizimden görebildiğimize göre, birincil LDA matris satırı her bir statik öznelik için ortalama alıcı gibi davranmaktadır. Bu tam olarak statik bilgiye karşılık gelir ama statik bilginin sadece ortadaki çerçeveden değil yanlardakilerden de pay alınarak ağırlıklı ortalama şeklinde hesaplanmasını önermektedir. Figür 2 yedi tane ikincil katsayıyı göstermektedir. 11 tane parametre için bu satırlar sanki birinci fark gibi davranmakta diğer 2 tanesi için ise ikinci fark olarak davranıyor gibi görünmektedir.

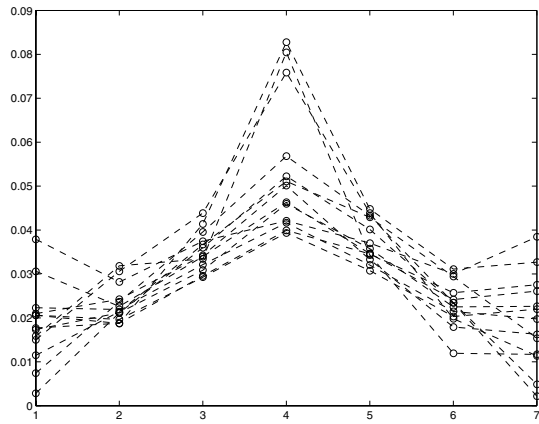


Fig. 1. 13 statik parametre için birincil "blok-yapılı LDA" katsayıları

5. SONUÇ

LDA+MLLT metodu çalıştığımız iki veritabanında da oldukça iyi çalışmaktadır. Blok-yapılı LDA metodu ile AURORA2 veritabanında çok yüksek gürültülü ortamlarda sınırlı miktarda iyileştirme elde ettik. HLDA ve HLDA-MAP metodları için daha güvenli sınıf-ıçi kovaryans matrisleri elde etmek gerektiğini düşünüyoruz. Ayrıca HLDA-MAP metodu için önsel parametrelerin belirlenmesi zor görünmekte ve bunları önceden belirlemek

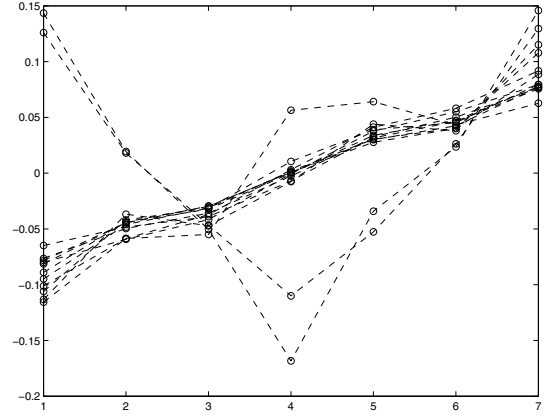


Fig. 2. 13 statik parametre için ikincil "blok-yapılı LDA" katsayıları

için değişik teknikler düşünülmesi gerekmektedir. Önsel parametreler veritabanına bağlı olarak değişebilir de. Önerilen metodların daha etkili ve başarılı olmaları için düzenliliştirme yolları hakkında daha fazla çalışmalar yapılması gerekmektedir.

6. REFERENCES

- [1] S. Furui, "Speaker independent isolated word recognition using dynamic features of speech spectrum," *IEEE Tr. Acoust. Sp. Sig. Proc.*, vol. 34, no. 1, pp. 52-59, 1986.
- [2] N. KuMar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Communication*, vol. 26, pp. 283-97, 1998.
- [3] R. A. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, volume 2, pp. 661-4, 1998.
- [4] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*, John Wiley & Sons, New York, 1973.
- [5] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Tr. Speech and Audio Proc.*, vol. 7, no. 3, pp. 272-81, May 1999.
- [6] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Tr. Acoust. Sp. Sig. Proc.*, vol. 37, no. 11, pp. 1641-1648, November 1989.