Başarım, Çeşitleme ve Hamming Uzaklığı Kullanılarak Hata Düzeltten Çıktı Kodu Budaması

ECOC Pruning using Accuracy, Diversity and Hamming Distance Information

Cemre Zor, Terry Windeatt, Josef Kittler
Centre for Vision, Speech and Signal Processing (CVSSP),
University of Surrey
Guildford, United Kingdom
{c.zor, t.windeatt, j.kittler}@surrey.ac.uk

Özetçe—Literatürdeki sınıflandırıcı topluluğu budama algoritmalarının, asıl olarak oy çokluğu ya da ağırlıklı oy çokluğu toplulukları için tanımlanmış olmaları; bu algoritmaların Hata Düzeltten Çıktı Kodlaması (HDÇK) topluluklarına uygulamalarını başarısız kılmasıdır. Bu bildiride, yeni bir başarıım metriğinin, Hamming uzaklığı ve çeşitleme bilgileri ile beraber kullanılması temeline dayalı yeni bir HDÇK budama algoritması sunulmuştur. Yapılan deneylerde, yeni sunulan algoritmanın literatürdeki en gelişmiş tekniklere göre çok daha başarılı sonuçlar verdiği tespit edilmiştir.

Anahtar Kelimeler — sınıflandırıcı topluluğu; budama; HDÇK.

Abstract— Existing ensemble pruning algorithms in the literature have mainly been defined for unweighted or weighted voting ensembles, whose extensions to the Error Correcting Output Coding (ECOC) framework is not successful. This paper presents a novel pruning algorithm to be used in the pruning of ECOC, via using a new accuracy measure together with diversity and Hamming distance information. The results show that the novel method outperforms those existing in the state-of-the-art.

Keywords — ensemble; pruning; ECOC.

I. GİRİŞ

Sınıflandırıcı topluluğu budamasının temel amacı, topluluktaki sınıflandırıcı sayısını azaltırken başarıım düzeyini korumak ya da arttırmak, ve dolayısıyla hesaplama karmaşıklığını ve depolama gereksinimlerini azaltmaktır. Budama yaparken, hesaplama karmaşıklığı yüksek olan tümünü kapsayan arama yöntemi yerine, fırsatçı arama ve sıralama tabanlı algoritmalar çoğunlukla tercih edilmektedir. Literatürdeki bu tarz algoritmaların taksonomileri ve detaylı analizleri için [1] ve [3]'e başvurulabilir.

Hata Düzeltten Çıktı Kodlaması (HDÇK) çok sınıflı sınıflandırma problemlerinde kullanılan güçlü bir sınıflandırıcı birleştirme yöntemidir [2]. HDÇK'nın temel prensibi, eldeki çok sınıflı problemi çözmek için, kendisini çok sayıda ve daha basit iki sınıflı problemlere ayırmak üzerine kuruludur.

Literatürdeki sınıflandırıcı topluluğu budama algoritmaları, temel olarak oy çokluğu ya da ağırlıklı oy çokluğu topluluklarına uygulanmak üzere tanımlandığından, bu algoritmaların HDÇK budaması için kullanımlarında problemler ortaya çıkmaktadır: Budama sırasında, iki sınıflı taban sınıflandırıcılarının başarımları ile çok sınıflı ana sınıflandırma probleminin başarıımı arasındaki ilişki göz önünde bulundurulmamaktadır. Bu eksikliği tamamlamak ve budama başarımlarını daha da arttırmak üzere, yeni bir başarıım metriğinin Hamming uzaklığı ve çeşitleme bilgileri ile kullanılması üzerine kurulu yeni bir HDÇK budama algoritması bu bildiride sunulmuştur.

Bölüm II literatürdeki en gelişmiş budama tekniklerinden bahsederken, Bölüm III HDÇK hakkında bilgi vermektedir. Bölüm IV'te HDÇK budaması ve önerilen yeni algoritma sunulmuş, Bölüm V'te ise deney sonuçları verilip, bu sonuçlar tartışılarak analiz edilmiştir.

II. BUDAMA HAKKINDA LİTERATÜR TARAMASI

Literatürde genel olarak oy çokluğu ya da ağırlıklı oy çokluğu toplulukları için tanımlanmış olan budama algoritmaları arasından en gelişmiş tekniklere sahip bir kaçının özeti aşağıda verilmiştir.

Sıralı yığılma esasına dayalı budama yöntemleri, seçilen bir ölçüye göre bütün sınıflandırıcıları sıralayıp, istenilen miktardakileri dizinin başından itibaren seçme prensibi ile çalışır.

Hata Azaltma Budaması (HAB) [11], ilk durumunu bir doğrulama kümesi üzerinde en düşük hata oranına sahip olan sınıflandırıcıyı seçerek gerçekleştiren bir sıralı yığılma budama yöntemidir. Sonraki her durumda, eldeki alt-sınıflandırıcı topluluğuna eklenildiğinde en fazla başarıım oranını getiren sınıflandırıcı seçilir.

Destekleme-bazlı Budama (DAB) [3] da sınıflandırıcıların destekleme algoritmasındaki [12] başarımlarını temel alan bir sıralı yığılma budama yöntemidir. Destekleme algoritmasının her bir durumunda, en düşük ağırlıklı eğitim hatasına sahip sınıflandırıcı seçilir.

Yukarıda bahsedilen sıralı yığılma budamalarının yanısıra, sınıflandırıcı budaması probleminin bir eniyileme çerçevesi içerisinde çözülmesi [6]'da önerilmiştir. Bu yöntemde (OPTB), sınıflandırıcı topluluk başarısının taban sınıflandırıcılarının bireysel sınıflandırma güçlerine ve birbirlerini tümleme kapasitelerine bağlı olduğu buluşsal kuralına dayanarak, başarımlar ve çeşitlemeyi enbüyütmeyi amaçlayan bir eniyileme problemi formüle edilmiştir.

Genel olarak, taban sınıflandırıcılarının başarımları arttıkça, birbirlerinden farkları azalmakta dolayısıyla da çeşitleme miktarı düşmektedir. OPTB'de, en iyi başarımlar-çeşitleme ödünleşimini hesaplayabilmek için, seçilen bir doğrulama kümesi S 'deki verileri kullanarak bir P matrisi oluşturulur. i taban sınıflandırıcısı S_m veri noktasında başarılı ise, $P_{mi} = 0$; değilse $P_{mi} = 1$ olarak atanır. P matrisini kullanarak oluşturulan $G = P'P$ matrisinin herhangi bir köşegen ögesi G_{ii} , i sınıflandırıcısı tarafından yapılan toplam hata sayısını; köşegen dışı herhangi bir G_{ij} ögesi ise i ve j sınıflandırıcıları tarafından yapılan toplam hata sayısını gösterir. Böylece, G matrisinin tüm ögelerini $[0,1]$ aralığına standartlaştırarak elde edilen $\tilde{G}$ matrisi üzerinde hesaplanan $\sum_i \tilde{G}_{ii}$ değeri, sınıflandırıcı topluluğunun gücünü taban sınıflandırıcılarının doğruluğu cinsinden, $\sum_{i,j,i \neq j} \tilde{G}_{ij}$ değeri ise çeşitlemeleri cinsinden belirtmekte kullanılabilir. Böylelikle, hem başarımlar hem çeşitleme bilgilerini içeren $\tilde{G}$ değeri, sınıflandırıcı topluluğunun genel başarımlarını ölçmek için şu şekilde kullanılabilir:

$$\begin{aligned} \min_x \quad & x' \tilde{G} x \\ \text{s.t.} \quad & \sum_t x_t = p \\ & x_t \in \{0,1\} \end{aligned}$$

Bu formülde x , budama sonucunda t sınıflandırıcısı seçilmişse $x_t = 1$ ögesine, seçilmişse de $x_t = 0$ ögesine sahip bir vektörken, p de budanacak sınıflandırıcı topluluğunun alması istenen ebatın değeridir. Bu problem NP-zor olduğu için, çözüme yarı-belirli programlama (SDP) ile gidilmesi önerilmiştir.

III. HATA DÜZELTEN ÇIKTI KODLAMASI (HDÇK)

HDÇK yöntemi, çok sınıflı sınıflandırma problemini eğitim verilerinin etiketlerini değiştirmek suretiyle çok sayıda iki sınıflı problemlere ayrıştırma esasına dayalıdır. Temel amaç, bu basit iki sınıflı problemlerin karar sınırlarını birleştirerek, çok daha karmaşık olan çok sınıflı sınıflandırma problemini çözebilmektir. Eğitim verilerinin etiketlerinin nasıl değiştirileceği, ve dolayısıyla iki sınıf sınıflandırıcılarının nasıl eğitileceği, değişik yöntemler kullanılarak hazırlanabilen kod matrisinde belirlenir.

Verilen bir HDÇK matrisi C 'deki herhangi bir eleman, $C_{ij} \in \{-1,1\}$, i taban sınıflandırıcısı eğitilirken j sınıfına ait verilerin alacağı yeni etiket değerini belirtir. Test aşamasında gelen yeni veri bütün sınıflandırıcılardan geçirilir ve elde edilen karar etiketleri bir vektörde toplanır. Daha sonra, bu vektör ile her sınıfa ait kod kelimeleri (C_j satır dizilimine j

sınıfına ait kod kelimesi denir) karşılaştırılıp, karar etiketleri vektörüne en yakın olan kod kelimesi belirlenir. Bu kod kelimesine sahip sınıfın etiketi de test verisine atanır. Ölçülen uzaklık, Hamming veya Öklid uzaklığı gibi uzaklık ölçüleri ile ölçülebilir.

İsminden de anlaşılacağı üzere HDÇK, yeni veri testi sırasında en yakın kod kelimesi seçilirken, taban sınıflandırıcılarının hatalarını bir miktar düzeltebilir. Tam olarak, sınıf çiftleri arasındaki en küçük Hamming uzaklığına d dersek, test çıktısında $\lfloor (d-1)/2 \rfloor$ taneye kadar bit hatası düzeltebilir.

Kod matrislerinin belirlenmesi için veriye bağımlı ya da veriden bağımsız çok sayıda yöntem bulunmaktadır. Fakat, teorik ve deneysel yöntemlerle ispatlandığı üzere [5], rastgele üretilmiş elemanlardan oluşan ve çok sayıda kolona sahip HDÇK matrisleri, güçlü taban sınıflandırıcıları ile kullanıldığında ideale yakın başarımlar elde edilmektedir. Bu nedenle, çalışmamızda bu tarz HDÇK matrisleri tercih edilmiştir.

IV. HDÇK BUDAMASI

Bölüm II'de tarif edilen algoritmalar, deneylerimizde HDÇK çerçevesinde kullanılmışlardır. Bu yöntemler ile, HDÇK matrisleri ile belirlenmiş iki sınıflı ayrıştırmalarda eğitilen iki sınıf sınıflandırıcıları budanmıştır. Fakat, bu budamalar sırasında kullanılan iki sınıf taban sınıflandırıcılarının doğruluk/başarımlar bilgisi ile çok sınıflı sınıflandırıcı topluluğunun doğruluğu/başarımlar arasında bir bağ kurulmuyor olması, genel performansın düşük olmasına sebebiyet vermiştir. Unutmayalım ki, bu algoritmaların asıl olarak tanımlandıkları oylama topluluklarında, taban sınıflandırıcılarının çözdükleri problemler ile sınıflandırıcı topluluğunun çözdüğü problem aynı olduğundan, böyle bir bağ kurulmasına ihtiyaç duyulmamıştır.

Bu bağ kurmak ve budama performansını arttırmak amacıyla, [9]'da tanıttığımız yeni başarımlar metriği HdPerf'in, Hamming uzaklığı ve çeşitleme bilgileri ile beraber kullanılması temeline dayalı yeni bir HDÇK budama algoritmasını (BCH) bu bildiride sunmaktayız.

A. HDÇK Başarımlar Metriği (HdPerf)

HdPerf, k sınıflı sınıflandırıcı topluluğuna ait bir taban sınıflandırıcısının başarımlarını, her sınıf için olmak üzere toplam k kere ölçüp, bu ölçümlerin ortalamasını almak esasına dayanan bir HDÇK başarımlar metriğidir.

N adet taban sınıflandırıcısından oluşan ve k adet sınıfa ait bir HDÇK matrisi ve μ adet eğitim verisine sahip bir seçim kümesi S 'ye sahip olduğumuzu düşünelim. Taban sınıflandırıcısı i 'yi eğitmek için kullanılan j sınıfına ait S_m verisinin HDÇK çerçevesinde alacağı yeni sınıf etiketi

$$\theta_{m,i} = \Psi_i(j) = \Psi(\emptyset(S_m))$$

olarak gösterilebilir. Burada, $1 \leq m \leq \mu$, $1 \leq i \leq N$ 'dir, $\emptyset$ çok sınıflı problem için hedef etiket fonksiyonudur, ve de Ψ_i HDÇK matrisinin i . taban sınıflandırıcısı için tanımlanan ikili ayrıştırmadır.

Bir taban sınıflandırıcısı i 'ye ait HdPerf'i hesaplamak için, seçim kümesi S 'deki her veri S_m , k kere yeniden-etiketlendirilip, hedef-eşleştirilir. Bu k kere yenilenen işlemin her aşaması l 'de, yeniden-etiketlendirme fonksiyonu r şöyle tanımlanırken

$$r_l(S_m) = +1, \quad \text{eğer } (\emptyset(S_m) = 1) \\ r_l(S_m) = -1, \quad \text{aksi takdirde}$$

hedef-eşleme fonksiyonu f ise

$$f_{i,l}(S_m) = \Psi_i(l)d_i(S_m)$$

olarak formüle edilmiştir. Bu formülde $d_i(S_m)$, taban sınıflandırıcısı i 'nin veri S_m için karar etiketini belirtir.

Her durum l içerisinde r fonksiyonu, ilgili sınıf olan l 'ye ait verilere 1, diğerlerine -1 değerini atayarak 1-ekarşı-herşey yeniden-etiketlendirmesi yapmaktadır. Diğer taraftan, eşleştirme fonksiyonu f , HDÇK matrisinden edindiği bilgi doğrultusunda taban sınıflandırıcısı i 'nin kararlarını yeni değerlere eşleştirir.

Sınıf l ile aynı ayrışmada bulunan ve dolayısıyla aynı HDÇK etiketine ($\theta_{m,i} = \Psi_i(l)$) sahip her veri S_m 'in etiketi, i tarafından doğru sınıflandırılmışsa $f_{i,l}(S_m) = +1$ değerine, yanlış sınıflandırılmışsa $f_{i,l}(S_m) = -1$ değerine eşleştirilir. Karşıt şekilde, sınıf l ile farklı ayrışmada (farklı HDÇK etiketlenmesinin sonucu olarak) bulunan veriler, i tarafından doğru sınıflandırılmışsa $f_{i,l}(S_m) = -1$ değerine, yanlış sınıflandırılmışsa $f_{i,l}(S_m) = +1$ değerine eşleştirilir. Dikkat etmek gerekir ki f fonksiyonu i 'ye bağlıyken r fonksiyonu değildir.

S üzerinde ve l durumunda, i için ölçülen HdPerf _{i,l} (i 'nin S üzerinde sınıf- l -yekarşı-herşey analizi olarak düşünülebilir) şu şekilde verilmiştir:

$$\text{HdPerf}_{i,l} = \sum_{m=1}^{\mu} f_{i,l}(S_m) r_l(S_m).$$

Son olarak ise HdPerf _{i} , HdPerf _{i,l} 'in tüm durumlar için ($l=1\dots k$) ölçülüp, ortalaması alınması ile hesaplanır.

$$\text{HdPerf}_i = \frac{1}{k} \sum_{l=1}^k \sum_{m=1}^{\mu} f_{i,l}(S_m) r_l(S_m).$$

B. HdPerf'in Yorumlanması

HdPerf, verilen bir taban sınıflandırıcısı i 'nin, çok sınıflı problemdeki her sınıf için ortalama 1-ekarşı-herşey başarımını ölçmeyi hedefler. Bu ölçüm, i 'nin kararları ve HDÇK matris bilgisi kullanılarak yapılmaktadır. Aşağıda, i taban sınıflandırıcısı için l durumunda HdPerf _{i,l} metriğinin özet yorumu verilmiştir:

1) Verilerden l sınıfına ait olanlar, doğru sınıflandırılmış iseler +1 katsayısı ile ödüllendirilip ($f_{i,l}(S_m)r_l(S_m) = 1$), değilse -1 ile cezalandırılmaktadırlar ($f_{i,l}(S_m)r_l(S_m) = -1$).

2) Verilerden l sınıfı ile farklı HDÇK ayrışımında bulunanlar, doğru sınıflandırılmış iseler +1 katsayısı ile ödüllendirilip, değilse -1 ile cezalandırılmaktadırlar.

3) Verilerden l sınıfı ile aynı HDÇK ayrışımında bulunanlar (ama l sınıfına ait olmayanlar), doğru sınıflandırılmış iseler -1 katsayısı ile cezalandırılıp, değilse +1 ile ödüllendirilmektedirler.

3. maddeye ithafen belirtilmelidir ki, taban sınıflandırıcısının l durumunda gücü kendisinin l sınıfını diğer sınıflardan ayırabilme becerisine dayandığı için, l sınıfına ait olmayan fakat l sınıfı ile aynı HDÇK ayrışımında bulunan veriler, doğru sınıflandırıldıkları için cezalandırılırlar.

Böylelikle i için, verilen bir sınıf l -yekarşı-herşey analizinde,

$$\text{HdPerf}_{i,l} = (no(GP_l) + no(GN_l)) - (no(YP_l) + no(YN_l))$$

olduğu görülebilir. Burada, μ eğitim verisi sayısı; $no(GP_l)$, $no(GN_l)$, $no(YP_l)$ ve $no(YN_l)$ ise sırasıyla gerçek pozitif, gerçek negatif, yanlış pozitif ve yanlış negatif veri sayılarıdır. Böylelikle, HdPerf _{i,l} metriğinin aslında i taban sınıflandırıcısı için l sınıfı üzerinde ölçülen bir başarımlı göstergesi olduğu görülebilir.

C. Başarım, Çeşitleme ve Hamming Uzaklığı Kullanılarak HDÇK Budaması (BCH)

BCH'deki temel amaç, genel topluluk sınıflandırıcılarının performanslarını belirlemede önemli rol tutan başarımlı ve çeşitleme bilgilerini, HDÇK çerçevesine özel hale getirip, ayrıca HDÇK için önemli bir performans unsuru olan Hamming uzaklığı bilgisinden de yararlanarak budama yapmaktır. Bu sebeple, HDÇK için özel olarak formüle edilmiş başarımlı metriği HdPerf, Q çeşitleme metriği ve HDÇK satır Hamming uzaklığı (HUU) ölçüsüyle beraber, II'de açıklaması verilen OPTB optimizasyon çerçevesini güncellemek için kullanılmıştır.

Q çeşitleme metriğinin [10] teorik sınırları ile sınıflandırıcı topluluğu başarımı arasındaki ilişkilerin ispatlanmış olması ve bu metriğin yapay sinir ağları oylama toplulukları gibi sınıflandırıcı topluluklarının başarımı ile ilintisinden dolayı, çalışmamızda çeşitleme metriği olarak kullanılmasına karar verilmiştir. D_i ve D_i 'nin iki taban sınıflandırıcısı olduğunu düşünelim. Genellik kaybı olmadan, D_i 'nin z verisini doğru sınıflandırmasını $D_{iz} = 1$ ile, ve yanlış sınıflandırmasını $D_{iz} = 0$ ile belirtirsek, D_i ve D_i arasındaki Q çeşitleme metriği aşağıdaki gibi tanımlanmaktadır:

$$Q_{ij} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}.$$

Burada N^{ab} , $D_{iz} = a$ ve $D_{iz} = b$ eşitliklerini sağlayan toplam veri sayısını belirtir. [8]'de gösterildiği gibi Q 'nun alt ve üst limitleri sırası ile -1 ve 1'dir. Aynı verileri doğru sınıflandırmaya yatkın olan sınıflandırıcılar pozitif Q 'ya ve dolayısıyla düşük çeşitlemeye, farklı veriler üzerinde hatalar yapan sınıflandırıcılar da negatif Q 'ya yani yüksek çeşitliliğe sahiptir. Bu sebeple, budama sırasında Q 'nun enküçüklenmesi hedeflenmelidir.

HDÇK satır Hamming uzaklığı (HUU), HDÇK'nın hata düzeltme kapasitesini budama sırasında olabildiğince muhafaza etmek amacıyla bu makalede tanımlanmaktadır. HUU, satırların tüm ikili katışimleri ele alındığındaki toplam bit benzerliği sayısı olarak şu şekilde ifade edilmektedir:

$$HUU_i = \sum_{r=1}^N |I(C(r, i) = C(s, i))|$$

Bu formülde $I(\text{doğru}) = 1$, $I(\text{yanlış}) = 0$ olarak ifade edilmektedir. HUU_i ölçüsü azaldıkça hata düzeltme kapasitesi artacağından, budama sırasında bu ölçünün de enküçülenmesi hedeflenmelidir.

BÇH'yi OPTB optimizasyon çerçevesinde formüle etmek için, bu çerçevede tanımlanmış olan G matrisi; $HDPerf$, Q ve HUU bilgilerini içermek üzere oluşturulan 3 yeni matrisin toplamı olmak üzere değiştirilir:

$$G = w_A A + w_D D + w_E E.$$

N tane sınıflandırıcının olduğu bir durum için, A matrisi $N \times N$ 'lik bir kare köşegen matris olup köşegen girdilerinin değerleri $A_{ii} = 1 - HDPerf_i$ 'dir. D ise, köşegen girdileri 0'a eşit, diğer girdileri ise $D_{ii} = Q_{ii}$ olan simetrik bir matristir. Son olarak, bir başka köşegen matrisi olan E 'nin köşegen girdileri $E_{ii} = HUU_i$ olarak hesaplanmıştır ($1 \leq i, j \leq N$). Bu durumda G ; başarımlar, çeşitleme ve Hamming uzaklığı bilgilerini içerecek şekilde güncellenmiş olmuştur. Bütün girdileri $[0,1]$ aralığına standartlaştırılan G matrisi, $\tilde{G}$ ismi ile anılmakta; ve OPTB eniyileme çerçevesinde tanımlanan budama problemi bu yeni $\tilde{G}$ 'yi kullanarak çözülmektedir. Bu eniyileme sırasında; A , D ve E matrislerine $[0,1]$ aralığında katsayılar verilerek ızgara araması da yapılmaktadır.

V. DENEYLER

Deneyler 2 adet yapay veri kümesi ve 7 adet UCI Makine Öğrenmesi Veri Arşivinden (UCI MLR [7]) elde edilmiş veri kümelerinde denenmiştir. Budama algoritmaları, taban sınıflandırıcıları CART ağaçları ve 16 düğümlü & 15 turlu yapay sinir ağları (NN) olan 50 sınıflandırıcı topluluklarda, %30,...,%70 budama oranları ile çalıştırılmıştır. Deneyler 10'ar kere, değişik eğitim-test verileri ayrımları ve farklı HDÇK matrisleri kullanılarak tekrar edilmiştir. Sonuçların tutarlı olmasından dolayı sadece %70 budama oranı ile edilen sonuçlar Şekil 1'de sunulmuştur.

		RAST	OPTB	HAB	DAB	BÇH
CART	artMulti1	0.711	0.6967	0.7222	0.7184	0.7295
	artMulti2	0.739	0.7397	0.7768	0.7581	0.77
	derma	0.929	0.8813	0.9573	0.9296	0.9427
	glass	0.622	0.5854	0.6641	0.6483	0.6813
	vehicle	0.71	0.672	0.7254	0.7223	0.7254
	yeast	0.499	0.435	0.5328	0.5235	0.5318
	optdigits	0.87	0.8858	0.9047	0.8939	0.9087
	segment	0.876	0.8395	0.9129	0.9024	0.916
	sat	0.861	0.8463	0.8786	0.8739	0.8802
NN	artMulti1	0.75	0.7467	0.7543	0.7549	0.7549
	artMulti2	0.783	0.7812	0.8055	0.8014	0.8068
	derma	0.946	0.9187	0.9513	0.8884	0.9566
	glass	0.612	0.5651	0.6533	0.6337	0.6483
	vehicle	0.788	0.725	0.8035	0.7889	0.8081
	yeast	0.549	0.5142	0.5735	0.5646	0.5769
	optdigits	0.896	0.8694	0.9461	0.935	0.935
	segment	0.881	0.8693	0.9082	0.899	0.9096
	sat	0.876	0.8644	0.8897	0.884	0.8879

Fig. 1. 50 taban sınıflandırıcı için budama algoritma sonuçları, %70 budama oranı ile, NN ve CART kullanarak.

Şekil 1'de, OPTB, HAB, DAB ve BÇH algoritmalarının yanı sıra, rastgele seçilerek budanmış matrislerin başarımlar oranları da sunulmuştur (RAST). RAST sütununda, her deney için 20 rastgele seçimle elde edilen budanmış matrislerin ortalama başarımları verilmiştir.

Her veri kümesi ve sınıflandırıcı çeşidi için en iyi başarımlar oranına sahip olan metod kalın yazı stili ile belirtilmiştir. Bu sonuçlara göre BÇH diğer metotlara karşı genel bir performans üstünlüğüne sahip olmuş, HAB ise onu takip eden metod olarak karşımıza çıkmıştır. BÇH'nin RAST, OPTB ve DAB'a karşı olan mutlak üstünlüğünün de altı çizilmelidir.

Gelecekte, $\tilde{G}$ matrisini oluşturan elemanların katsayılarının budama üzerindeki etkileri incelenecek; ve buna göre de başarımlar, çeşitleme ve Hamming uzaklığının BÇH metoduna bireysel olarak katkılarının önemleri karşılaştırılacaktır.

KAYNAKÇA

- [1] Tsoumakas G., Partalas I., and Vlahavas I., *A taxonomy and short review of ensemble selection*, In ECAI 08, SUEMA Workshop, 2008.
- [2] Dietterich, T.G., Bakiri, G., Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research. (JAIR)* 2, 263-286 (1995).
- [3] O. Martinez-Munoz, G., Suarez, A., Using boosting to prune bagging ensembles. *Pattern Recognition Letters* 28, 156-165 (2007).
- [4] Martinez-Munoz, G., Hernandez-Lobato, D., Suarez, A., An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Trans. Pattern Anal. Mach. Intell.* 31(2), 245-259 (2009).
- [5] James, G., Hastie, T., *The error coding method and picts* (1998)
- [6] Zhang, Y., Burer, S., Street, W.N.: Ensemble pruning via semi-definite programming: *JMLR* 7, 1315-1338 (2006).
- [7] Asuncion, A., Newman, D., UCI machine learning repository (2007), <http://www.ics.uci.edu/mllearn/MLRepository.html>
- [8] Kuncheva, L I., and Whitaker, C. J., Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181-207 (2003).
- [9] Zor, C., Windeatt, T., and Kittler, J., *ECOC Matrix Pruning Using Accuracy Information*. Multiple Classifier Systems (MCS) Workshop, 2013, accepted.
- [10] Yule G. U., On the association of attributes in statistics. *Phil. Trans*, 194:257-314 (1900).
- [11] Margineantu, D.D., Dietterich, T.G., *Pruning adaptive boosting*. ICML. pp.211-218 (1997).
- [12] Freund, Y.; Schapire, R. E., A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of Computer and System Sciences In Computational Learning Theory*, 55(1), 119-139 (1997).