

Sıradüzensel Kümeleme İçin Yeni bir Uzaklık Ölçütü

A New Distance Measure for Hierarchical Clustering

Hasan Serhan Yavuz, Hakan Çevikalp

Elektrik ve Elektronik Mühendisliği Bölümü

Eskişehir Osmangazi Üniversitesi

hsyavuz@ogu.edu.tr; hakan.cevikalp@gmail.com

Özetçe

Destek vektör makine (DVM) sınıflandırıcısı esasen ikili sınıf problemleri için tasarlanmış olup çoklu sınıf problemlerinde nasıl kullanılması gerektiği hala tartışmaya açık bir araştırma konusudur. Bire-bir veya bire-tüm gibi klasik yaklaşımlar çoklu sınıf problemini kısmen çözmesine rağmen bu yaklaşımlar eğitim kümesi sınıf sayısının çok büyük olduğu durumlarda çekiciliğini kaybeder. Son zamanlarda, sınıf sayısının çok olduğu durumlarda klasik yöntemlere göre daha etkili olan sıradüzen tabanlı sınıflandırma yaklaşımları kullanılmaktadır. DVM tabanlı sıradüzensel sınıflandırıcılar veri örneklerinin bir kümeleme algoritması ile bölütlenmesini içerir ve sınıflandırma başarısı, yüksek derecede üretilen kümelerle bağlıdır. Kümeleme algoritması olarak k -ortalama (k -means), çekirdek k -ortalama (kernel k -means), yuvarsal kabuklar (spherical shells), dengeli alt-küme kümeleme (balanced subset clustering) gibi yöntemler kullanılabilir fakat bu yöntemlerin veri örneklerini bölütlemeye kullandıkları uzaklık ölçütleri DVM sınıflandırıcısıyla uyumlu değildir.

Bu çalışmada, DVM tabanlı sıradüzensel sınıflandırıcılardaki verilerin kümelenebilirliği sırasında kullanılabilen yeni bir uzaklık ölçütü tanıtılmıştır. Önerdiğimiz uzaklık ölçütü, diğer kümeleme yöntemlerinden farklı olarak, DVM sınıflandırıcı prensipleriyle uyumludur. Yapılan deneyler, önerilen uzaklık ölçütünü kullanan kümelendirici algoritmasının DVM tabanlı sıradüzensel sınıflandırıcılarında, tanıma oranlarında küçük bir kayba rağmen, sistemin hesap yükü verimliliğini önemli ölçüde geliştirdiğini göstermektedir.

Abstract

Support Vector Machine (SVM) classifier formulation is originally designed for binary classification, and the extension of it to the multi-class case is still an open research problem. Classical approaches such as one-against-one or one-against-all have been used to address the multi-class problem, but these approaches become less appealing when the number of classes in the training set is too large. Recent approaches use hierarchical based classification for the multi-class problems since they scale well with the number of classes. SVM based hierarchical classifiers involve the partition of data samples through a clustering algorithm, and classification performance of the overall system heavily depends on the generated clusters. The clustering methods such as k -means, kernel k -means, spherical shells and balanced subset clustering have been used for this goal, but their distance measures, which are used for partitioning the data samples, are not compatible with the SVM classification goal.

This paper introduces a new distance measure for partition of data samples for SVM based hierarchical

classification. Unlike other clustering methods used for this goal, our proposed method is suitable when SVMs are used as the base classifier. As demonstrated in the experiments, integrating the proposed clustering scheme into the hierarchical SVM classifiers significantly improves the computational efficiency with a small decrease in the recognition accuracy.

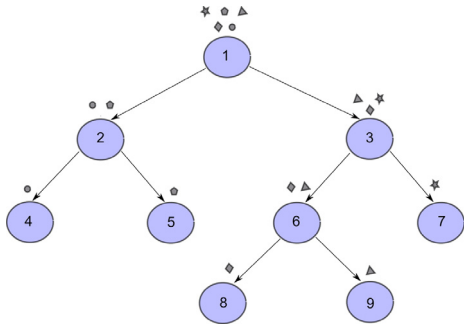
1. Giriş

Destek vektör makine (DVM) sınıflandırıcısı eşzamanlı olarak ampirik sınıflandırıcı hatasını enküçükleyip, sınıflar arasındaki geometrik boşluğu (marjin) enbüyükleyen başarılı bir yöntemdir [1]. Yöntem basitçe, iki sınıfa ait örnekler arasındaki en büyük boşluğu veren ayırıcı bir üst-düzlem (hyper-plane) bulur. DVM formülasyonu esasen ikili sınıflandırma için geliştirilmiş olup, çoklu sınıf problemlerinde oldukça kompleks optimizasyon gerektirmesi yüzünden formülasyon bu yönde genişletilmemiştir. Ancak, birçok örüntü tanıma uygulaması ikiden fazla sınıf içerir. Çoklu sınıf DVM problemleri, birçok ikili sınıflandırıcı oluşturularak çözülür. Bu amaca ulaşmak için birkaç yöntem mevcuttur. Bu yöntemler arasında en eski ve en yaygın olanları bire-tüm (one-against-all, OAA) ve bire-bir (one-against-one, OAO) yaklaşımlarıdır [2]. Birinci yöntem, C sınıf problemi için, sırayla eğitim setindeki bir sınıfı seçer ve bu sınıfı geriye kalan $C-1$ sınıftan ayıran üst-düzlem parametrelerini bulur. Bu işlem eğitim setindeki tüm sınıflar için (C kere) tekrarlanır. Her aşamadaki sınıflandırıcı eğitimi için tüm eğitim verisi kullanılır. Test verisi sınıf ataması da bu sınıflandırıcıların en yüksek çıkış değerine göre yapılır. İkinci yöntemse C sınıf arasında oluşturulması mümkün tüm $C(C-1)/2$ adet ikili sınıflandırıcıyı eğitir. Topluluk kararı tipik olarak en yüksek kazanır (max wins) algoritmasına göre belirlenir: Her OAO sınıflandırıcısı tercih ettiği sınıfa bir oy verir ve nihai karar en çok oyu alan sınıf olarak belirlenir. OAO yöntemi, OAA yönteminden daha çok sınıflandırıcı oluşturmasına rağmen daha az sayıda örnek üzerinde eğitim yaptığı için genellikle eğitim aşamasında OAA'dan daha hızlıdır [2]. Ancak, OAO sınıflandırıcılarının sayısı toplam sınıf sayısı ile süper doğrusal (quadratic) biçimde artar. Bu yüzden sınıf sayısının çok büyük olduğu durumlarda işlem yükü çok ağırdır.

Son zamanlarda DVM kullanan sıradüzensel sınıflandırıcıların fazla sınıflı problemlerde kullanımı dikkat çekmeye başlamıştır. [3-5]. Sıradüzensel sınıflandırıcılar öncelikle karmaşık olan bir problemi daha basit alt problemlere ayırır ve bu alt problemleri çözer. Sonra bu alt problemlerin sonuçları ana problemi çözmek için izlemsel olarak birleştirilir. Karara yönlendirilmiş çevrimsiz çizelgeler

(Decision Directed Acyclic Graphs (DDAGs)) [3] ve ikili sıradüzensel karar ağaçları (Binary Hierarchical Decision Trees (BHDTs)) [4,5] bu gruba düşen yaygın yöntemlerdendir. OAO yaklaşımında olduğu gibi, DDAG yöntemi $C(C-1)/2$ adet ikili sınıf eğitimi gerektirir ancak test aşamasında yönlendirilmiş bir çevrimsiz çizelge (Directed Acyclic Graph (DAG)) kullanılır. Bu yüzden, OAO yönteminden farklı olarak test örneği ataması sırasında $C(C-1)/2$ sınıflandırıcı hesaplaması yerine sadece $C-1$ karar düğümü hesabı gereklidir; bu durum test aşamasını oldukça hızlandırır. BHDT yöntemi ise veriyi sıradüzensel olarak iki altkümeye bölen bir karar ağacı içerir. Bu işlem her alt küme sadece bir sınıfı temsil edene kadar sürdürülür. Daha sonra, bölütlenen altkümeleri ayırmak için ikili karar ağacının her iç düğümünde DVM sınıflandırıcısı kullanılır. Şekil 1'de 5 sınıfı ayıran tipik bir BHDT verilmiştir. Veri bölüntülemesi genellikle bir kümeleme algoritması tarafından gerçekleştirilir ve her bir iç düğümdeki DVM sınıflandırıcısının başarısı üretilen bölüntülere bağlıdır. Literatürde kullanılan k -ortalama, kernel k -ortalama, yuvarsal kabuklar ve dengeli altküme kümeleme gibi farklı kümeleme algoritmaları, bölüntüleme için farklı uzaklık ölçütleri kullanmaktadır. [4,5]. İyi dengelenmiş bir karar ağacı, bir test örneğinin etiketlenmesi esnasında, tepe düğümünden terminal karar düğümüne giden bir yol için yaklaşık $\log_2 C$ sınıflandırıcı hesaplaması gerektirir. Bu yüzden bu yapı, hesaplama süresi bakımından DDAGlerden çok daha verimlidir.

Bu çalışmada DVM tabanlı BHDTler üzerinde yoğunlaşıp, sınıfları kümelemek için yeni bir kümeleme algoritması önerdik. Önerilen kümeleme yönteminde DVM sınıflandırıcı temellerine uyumlu olan yeni bir uzaklık metriği kullandık. Bildirinin 2. bölümünde önerilen kümeleme yöntemi tanıtılmıştır. Veri kümeleri ve deneysel yordam 3. bölümde tanımlanmıştır. Bölüm 4 de ise görüş ve sonuçlar verilerek çalışma tamamlanmıştır.



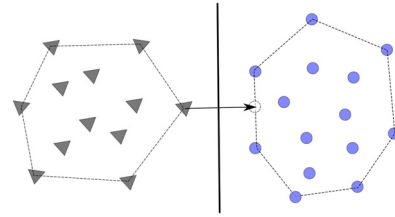
Şekil 1: 5 sınıflık tanıma problemini çözen ikili sıradüzensel karar ağacı. Karar ağacı 5'i terminal 4'ü iç düğüm olmak üzere toplam 9 düğüm içermektedir. Her bir iç düğümdeki (1,2,3,6 nolu düğümler) kümeleme yöntemiyle iki gruba bölünen sınıfları ayırmak için DVM kullanılmıştır.

2. Yöntem

2.1. BHDT Tasarımı İle İlgili Hususlar

Veri bölüntülenmesinde BHDTler kümeleme algoritmaları kullanır, bu yüzden sınıflandırıcı verimliliği üretilen kümelere bağlıdır. Daha ayrıntılı ifade etmek gerekirse, karar ağacının her bir iç düğümünde iyi dengelenmiş ayrıştırılabilir kümeler oluşturulması tüm sınıflandırıcı verimliliğini arttıracaktır.

Ayrıca uygulanan algoritmanın kullanılan sınıflandırıcı biçimiyle de uyumlu olması gerekmektedir.



Şekil 2: İki sınıfa ait dışbükey zarf ve bu sınıfları ayıran üst düzlem. Dışbükey zarfların üzerindeki en yakın noktalar DVM sınıflandırıcısının sınıfları ayırmak için kullandığı üst-düzlemi belirler.

BHDTlerde kullanılacak iyi bir kümeleme algoritması tasarlamak için, öncelikle sınıflandırıcı olarak kullanılan DVM yöntemi mercek altına alınmalıdır. DVM sınıflandırıcısı, iki sınıftaki birbirine en yakın örneklerle üstdüzlem arasındaki uzaklığı enbüyükleyen bir ayırıcı üstdüzlem belirlemeye çalışır. Bunu başarmak için yöntem, her bir sınıfı bir dışbükey zarf (convex hull) olarak modeller [6]. Bir dışbükey zarf, orjinal kümedeki noktaların dışbükey katışımı (convex combination) biçiminde yazılabilen tüm noktalardan oluşur. Noktaların dışbükey katışımı, katsayıları eksi olmayan ve toplamları 1'e eşit olan doğrusal bir katışımıdır. İki sınıfa ait dışbükey zarflar Şekil 2'de gösterilmiştir. Bu yaklaşımı izleyerek DVM, dışbükey zarflar üzerindeki en yakın noktaları bulur [6]. Bulunan noktalar bir bölüt çizgisiyle bağlanır. Bölüt çizgisini iki eşit parçaya bölecek biçimde geçen ve bölüt çizgisine dik olan düzlem ayırıcı üst-düzlem olarak seçilir. Geometrik bakış açısından bakıldığında, sınıfların doğrusal olarak ayrılabilirdiği durumlarda, dışbükey zarflardaki birbirine en yakın iki nokta ayırıcı üst düzlemi belirler ve DVM marjinin tanımı dışbükey zarflar arasındaki en küçük uzaklığı ifade eder. Bu yüzden, k -ortalama kümeleme temeline dayanan kümeleme algoritmaları, doğrudan marjin enbüyüklemesini hedeflemediği için, DVM tabanlı BHDTlerde iyi bir seçim değildir. Diğer yandan, Shi ve Malik [7]'in düzelenmiş kesi (the normalized Cuts - NCuts) kümeleme algoritmasının amacı DVM temellerine uyumludur. Bu algortmada veri örnekleri sonsuz boyutlu bir öznelik uzayına haritalanır ve haritalanmış verideki boşluklar üzerinden bir üstdüzlem geçirilerek veri bölütlenir [8]. Daha sonra üstdüzlemin aynı tarafında kalan noktalar aynı kümeyle ait olacak biçimde etiketlenirler. Bununla birlikte NCuts yöntemi eğitimsiz bir yöntemdir. Bu yüzden algoritma sonucunda, düğümlerde aynı sınıfa ait olan verilerin aynı gruplarda yer alması, sınıf içi verilerin birbirlerine çok yakın ve diğer sınıflara ait örneklerden çok uzak olmadığı durumlarda garantilenemez. Bu çalışmada NCuts yöntemi geliştirilerek enbüyük marjinle iyi dengelenmiş iki küme üretmesi sağlanmıştır. Bir sonraki bölümde, yapılan değişiklikler anlatılmıştır.

2.2. Geniş Marjin Tabanlı Kümeleme

Eğitim kümesinin herbiri N_c örnek içeren toplam C adet sınıftan $\{\omega_1, \dots, \omega_C\}$ oluştuğunu varsayalım. Sıradüzensel ağacın her düğümünde n sınıf verildiğinde, önerilen kümeleme yöntemi n vertisli $\{v_1, \dots, v_n\}$ (her bir sınıf için bir tane) bir ağırlıklı çizge ve bu vertisleri içeren ayrıklar kümesi oluşturur. Vertis v_i ve v_j arasındaki her ayrık, vertisleri temsil eden sınıflar arasındaki benzerliğe göre eksi olmayan

bir ağırlık $w_{ij} = w_{ji} \geq 0$ taşır. Vertisler arasındaki sıfır ($w_{ij} = 0$) ağırlık vertislerin birbirine bağlı olmadığını gösterir. Tam bağlı bir çizge tüm ayrıtlar bağlı olduğunda elde edilir. Ayrıtları ağırlandırmada genellikle ısı kerneli (heat kernel) $w_{ij} = \exp(-d(\omega_i, \omega_j)^2/t)$ kullanılır. Burada t çekirdeğin genişliğini, $d(\omega_i, \omega_j)$ ise tercih edilen bir uzaklık fonksiyonu ile hesaplanan ω_i ve ω_j sınıfları arasındaki uzaklığı temsil eder. Buraya kadar anlatılan algoritma NCuts kümeleme yöntemine bir fark haricinde denktir. Bu fark algoritmada tek başına örneklerin değil, sınıfların ele alınmasıdır. Bu çalışmada DVM ile aynı amacı güttüğü için, sınıflar arasındaki uzaklık ölçüsü olarak sınıfların dışbükey zarfları arasındaki en küçük uzaklığı kullanmak önerilmiştir. ω_i ve ω_j sınıflarının dışbükey zarfları arasındaki en kısa uzaklık bu dışbükey zarfları içindeki en yakın iki nokta ile belirlenir. Bu iki noktayı bulma problemi, aşağıdaki eniyileme problemi biçiminde yazılabilir.

$$\min_{u,v} \frac{1}{2} \|X_i u - X_j v\|^2 \quad (1)$$

$$\text{s.t. } \sum_{m=1}^{N_j} u_m = 1, \sum_{m=1}^{N_j} v_m = 1, u \geq 0, v \geq 0.$$

Burada X_i , sütunları ω_i sınıfına ait olan örnek vektörlerinin oluşturduğu matristir ve $u \geq 0$ u vektörün tüm elemanlarının sıfırdan büyük veya eşit olduğunu ifade eder. Eşitlik (1)'in eniyi çözümü u^* ve v^* vektörleri olduğu kabul edilirse, sınıfların dışbükey zarfları arasındaki en küçük uzaklık $d(\omega_i, \omega_j) = \|X_i u^* - X_j v^*\|$ ile bulunur. Burada karşılaşılabilecek en önemli sorun sınıfların dışbükey zarflarının iç içe geçmesi yani sınıfların doğrusal olarak ayrılabilmesi olabilir. Bunun gibi bir durumda sınıflar arası uzaklık sıfır olur ve bu uzaklık sınıflar arasındaki gerçek benzerlikleri yansıtmayabilir. Böyle durumlarda, iç içe geçen bölgeleri ayırmak için veri, çekirdek hilesini (kernel trick) kullanan yöntemlerde olduğu gibi daha yüksek boyutlu bir uzaya haritalanabilir. Eşitlik (1)'den görüleceği üzere, eşitlikteki amaç fonksiyonu, çekirdek hilesi kullanmaya olanak sağlayan iç çarpımlar cinsinden yazılabilir. Sonuçta veri, uygun bir çekirdek fonksiyonu kullanarak DVM sınıflandırıcısında olduğu gibi dışbükey zarfların kesişmediği ve sınıflar arası uzaklıkların hesap edilebileceği daha yüksek boyutlu bir uzaya haritalanabilir.

Çizgenin ağırlıklandırılmış bitişiklik matrisi $W = (w_{ij})_{i,j=1,\dots,n}$ matrisidir. İkili kümeleme durumunda her ω_i sınıfına $y_i \in \{-1, +1\}$ etiketini atamak, çizgeyi -1 isimli vertislerden oluşan A kümesi ve $+1$ isimli vertislerden oluşan B kümesine bölüntülemektir. Yöntemin maliyet fonksiyonu aşağıdaki gibi tanımlanır:

$$NCut(A, B) = \left(\frac{1}{Vol(A)} + \frac{1}{Vol(B)} \right) \sum_{i \in A, j \in B} w_{ij} \quad (2)$$

Burada Vol bir kümedeki ağırlıkların toplamı ve $\sum_{i \in A, j \in B} w_{ij}$ ise

A ve B kümelerini ayırık kümeler yapmak için ortadan kaldırılması gereken ayrıtların toplam ağırlığıdır. Bu maliyet fonksiyonu iyi dengelenmeyen kesitleri cezalandırırken A ve B kümelerinin yaklaşık olarak aynı sayıda eleman içermelerini garanti eder [7]. Bu da herbir iç düğümde iyi dengelenmiş ve

birbirinden uzak, kolay ayrılabilen sınıf kümeleri oluşturur. Böylece DVM sınıflandırıcısının işi çok daha kolay hale gelir. Fakat, eşitlik (2)'de verilen problemin çözümü imkansızdır [7]. Bu nedenle ana problem, gevşetme yöntemine başvurulup, NCuts ağırlıklı çizgenin Laplace matrisinin enküçültülmesi problemine dönüştürülür. Laplace matrisi, $L = D - W$ biçiminde gösterildiğinde (D matrisi, girişleri W matrisinin sütun/satır toplamlarından oluşan köşegen bir matris) algoritma, aşağıdaki genel özdeğer problem çözümünden oluşur.

$$La = \lambda Da \quad (3)$$

Bu durumda, eşitlik (3)'deki ikinci en küçük özdeğerine karşılık gelen a^* özvektörünün bileşenleri veriyi iki kümeye bölmede kullanılır. Dikkat edilirse, ilk baştaki kümeleme işlemi özvektörün elemanlarını kümeleme işlemine dönüştürülmüştür. Seçilen eşik Δ ise (Δ genellikle sıfır olarak seçilir) kümeleme aşağıdaki gibi yapılır:

$$\begin{cases} y_i = -1, & \text{if } a_i^* > \Delta \\ y_i = +1, & \text{if } a_i^* \leq \Delta \end{cases} \quad (4)$$

BHDTlerde, yukarıdaki kümeleme algoritmasının kullanımı, etkili ve güvenilir bir sınıflandırma için karar ağacının her düğümünde iyi dengelenmiş ayrılabilir bölütler oluşturur. Sonuç olarak, ilk kez karşılaşılan bir örneği etiketlemek için tüketilecek ortalama test zamanı $O(\log_2 C)$ olacaktır. Bu da işlem karmaşıklığı en az $O(C)$ olan klasik yöntemlere göre oldukça hızlıdır.

Bu çalışmada anlattığımız bölütleme yönteminde sadece bir parametrenin (çekirdek genişliği t) ayarlanması gerekmektedir. Bu parametrenin seçimi oldukça önemlidir. İyi seçilmiş t değerleri dengelenmiş bölütler oluşturmaya rağmen bu parametreye çok küçük bir değer atanması durumunda, NCuts algoritması diğer sınıflardan oldukça uzakta olan izole bir sınıfı diğerlerinden ayırma amacını güdecektir. Bu da sistemin tanıma hızını azaltacaktır. Yaptığımız deneylerde en iyi t değerleri sınıflar arasındaki tüm uzaklıkların ortalamasına yakın çıkmıştır.

BHDTlerde kullanılan diğer kümeleme yöntemleriyle karşılaştırıldığında, önerdiğimiz yöntem bir çok avantajı beraberinde getirmektedir. Birincisi, kullandığımız uzaklık ölçütü DVM sınıflandırıcısıyla aynı amacı güttüğünden DVM sınıflandırıcısının iç düğümlerdeki grupları ayırma işlemi oldukça kolaylaşmaktadır. İkinci avantaj NCuts yönteminin otomatik olarak iyi dengelenmiş bölütler oluşturmaya göre ki bu durum sistemin hızlı çalışması için son derece önemlidir. Son olarak, kullandığımız bölütleme yöntemi çekirdek kernel fonksiyonlarının kullanımına izin verdiğinden, diğer bölütleme yöntemlerinin aksine doğrusal kernel dışında başka bir kernel fonksiyonu kullanan DVM sınıflandırıcısıyla da uyumlu olacaktır.

3. Deneyler

Önerilen yöntemin verimliliğini belirlemek için yöntemi Coil100 veritabanı üzerinde test ettik. Bu veritabanı 100 farklı nesnenin 5'er derece döndürülerek çekilmiş 72 adet 128×128 'lik görüntüsünden oluşur. Nesnelerin görünümüne göre bu veritabanından, doğrusal çekirdekle birlikte doğrusal olmayan çekirdekleri de test edebilmek için, doğrusal olarak en iyi ayrıştırılabilen 40 nesne seçilmiş, seçilen görüntüler Şekil 3'de gösterilmiştir.



Şekil 3: Coil100 veri tabanından seçilen sınıflar.

Tablo 1: Coil100 Veritabanı Sınıflandırma Sonuçları

Yöntem	Sınıflandırma Oranı (%) ve Standart sapma değeri (σ)			Test Süresi (saniye)		
	Doğrusal	Polinom	Gaussian	Doğrusal	Polinom	Gaussian
Önerilen Yöntem	94.62, $\sigma = 0.89$	95.82, $\sigma = 0.66$	97.06, $\sigma = 0.77$	2.23	6.15	5.95
BHTD ([4])	95.50, $\sigma = 1.25$	96.23, $\sigma = 0.75$	97.18, $\sigma = 0.71$	2.23	6.44	7.18
DVM, OAO	97.64 , $\sigma = 0.62$	95.52, $\sigma = 0.63$	98.13 , $\sigma = 0.58$	202.60	235.70	493.15
DVM, OAA	95.10, $\sigma = 0.61$	97.70 , $\sigma = 0.55$	97.39, $\sigma = 0.50$	14.55	46.89	67.43

Görüntüler gri seviyeliye dönüştürüldükten sonra temel bileşen analizi kullanılarak boyutları 100'e indirgenmiştir. Doğrusal çekirdek, 2. dereceden polinom çekirdeği ve Gauss çekirdeği kullanan DVM sınıflandırıcıları için test sonuçları çıkarılmıştır. Önerilen yöntem, OAO ve OAA DVM yaklaşımları ve k -ortalama tabanlı kümeleme yöntemi kullanan BHDTlerle [4] sınıflandırma doğruluğu ve test süreleri cinsinden karşılaştırılmıştır. Her bir sınıftan rastgele 36 örnek eğitim için seçilmiş, geriye kalan 36 örnek ise testte kullanılmıştır. Deney 5 kez tekrarlanarak deneysel parametreler ayarlanmış, 10 kez tekrarlanarak da test sonuçları elde edilmiştir. Sonuçlar, Tablo 1'de verilmiştir.

Tabloda verilen test süresi test örneklerinin tümünü sınıflandırmada harcanan ortalama süredir. OAO ve OAA yaklaşımı kullanan çok sınıflı DVMler için bir örneği sınıflandırmak için tüketilen zaman sabit olup, BHDTler için değişir. BHDTlerde en kısa hesaplama süresi aranan sınıf ilk düğümde bulunduğu; en uzun hesaplama süresi ise aranan sınıf $C-1$ karar fonksiyonu sonrasında bulunduğu gerçekleşir. Tablodan görüldüğü üzere, test süresi cinsinden en iyi sonucu önerdiğimiz yöntem vermiştir. Ancak önerdiğimiz yöntemin sınıflandırma başarısı OAO ve OAA kullanan çok sınıflı DVM sınıflandırıcısından genellikle daha azdır. Önerilen kümeleme yöntemi kullanan BHDTler, Vural ve diğerlerinin [4] önerdiği k -ortalama kümeleme kullanan yöntemle çok yakın ama daha kötü sonuçlar vermiştir. Bilindiği üzere, k -ortalama yaklaşımında, sınıf saçılımlarının küresel olduğu kabul edilir. Bu varsayımlar, bu deneyde sağlandığı için k -ortalama iyi sonuçlar vermiştir ancak gerçekte birçok sınıflandırıcı probleminde bu varsayımlar sağlanmaz. Ayrıca, k -ortalama kümeleme çekirdek fonksiyonları kullanılmasına izin vermez. Bu yüzden, önerdiğimiz yöntemin sınıf saçılımlarıyla ilgili herhangi bir kısıtı olmadığı ve çekirdek fonksiyonu kullanımına izin verdiği için [4]'de verilen k -ortalama kümelemeden daha uygun olduğunu düşünmekteyiz.

4. Sonuçlar

Bu çalışmada, DVM tabanlı sıradüzensel sınıflandırıcılarda veri örneklerini kümelemede kullanılmak üzere yeni bir

bölütleme algoritması tanıtılmıştır. Bu amaçla kullanılan diğer kümeleme algoritmalarının aksine, önerdiğimiz kümeleme algoritması, temel sınıflandırıcı olarak kullanılan DVMler ile uyumludur. Önerilen kümeleme algoritmasının BHDTlerde kullanımı karar ağacının her bir düğümünde iyi dengelenmiş bölüntüler üretmektedir. Sonuçta, bilinmeyen bir örneği etiketlemede gereken ortalama test zamanı $O(\log_2 C)$ olup, çoklu sınıf DVMlere kıyasla önemli bir iyileştirme getirmektedir.

5. Kaynakça

- [1] C. J. C. Burges "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge discovery*, pp. 121-167, 1998.
- [2] C. Hsu and C. Lin, "A comparison of methods for multi-class support vector machines", *IEEE Trans. on Neural Networks*, Vol. 13, pp. 415-425, 2002.
- [3] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification", Conf. on Neural Information Processing Systems (NIPS), 2000.
- [4] V. Vural and J. G. Dy, "A hierarchical method for multi-class support vector machines", International Conference on Machine Learning (ICML), 2004.
- [5] D. Casasent and Y. C. Wang, "A hierarchical classifier using new support vector machines for automatic target recognition", International Joint Conference on Neural Networks (IJCNN), 2005.
- [6] K. P. Bennett and E. J. Bredensteiner, "Duality and geometry in SVM classifiers", International Conference on Machine Learning (ICML), 2000.
- [7] J. Shi and J. Malik, "Normalized cuts and image segmentation", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, pp. 888-905, 2000.
- [8] A. Rahimi and B. Recht, "Clustering with normalized cuts is clustering with a hyperplane", *Statistical Learning in Computer Vision*, 2004.