DEÜ MÜHENDİSLİK FAKÜLTESİ
MÜHENDİSLİK BİLİMLERİ DERGİSİ
Cilt: 16 Sayı: 48 sh. 76-85 Eylül 2014

ÖRNEK TABANLI SINIFLANDIRICILDA KÜMELEME YÖNTEMİYLE PERFORMANS ARTIRIMI

(BOOSTING THE PERFORMANCE OF INSTANCE BASED CLASSIFIERS BY USING CLUSTERING)

Faruk BULUT[1], M. Fatih AMASYALI[2]

ÖZET/*ABSTRACT*

Örnek tabanlı sınıflandırıcılar basitliği, uygulanabilirliği ve şeffaflığından ötürü yaygın bir kullanıma sahiptir. *k* en yakın komşuluk sınıflandırıcısı (k-EKS) bu alanda en çok tercih edilen algoritmalardan biridir. k-EKS'de performans, k parametresi ile doğrudan ilişkilidir. En uygun *k* parametresi, kullanıcı tarafından genellikle deneme-yanılma yöntemiyle seçilir. Bununla birlikte, bir veri setinde çapraz geçerleme işlemi süresince her bir test örneği için aynı k parametresinin kullanılması genel sınıflandırma başarısını olumsuz etkilemektedir. Her bir test örneği için en uygun k değerinin seçilmesi daha başarılı sonuçlar elde edilmesini sağlayabilmektedir. Çalışmamızda her bir test örneği için en uygun k parametresini kümeleme yöntemiyle bulan ve bu sayede genel sınıflandırma başarısını artıran bir yöntem üzerinde çalışılmış ve başarılı sonuçlar elde edilmiştir.

Instance based classifiers have a world-wide usage due to their simplicity, applicability, and clearness. k Nearest Neighbors (k-NN) classifier is one of the most preferred algorithm in this area. The performance of k-NN is directly related with the k parameter. The best k parameter is generally chosen by the user and the optimal k value is found by experiments. Additionally, the chosen constant k value is used during the whole cross validation process. The fixed k value used for each test sample can decrease the overall prediction performance. The optimal k value for each test sample should vary from others' in order to have better performance. In this study, a dynamic k value selection method for each test sample is proposed. This improved method employs a simple clustering procedure in classification. In the experiments, more accurate results are found.

ANAHTAR KELİMELER/*KEYWORDS*

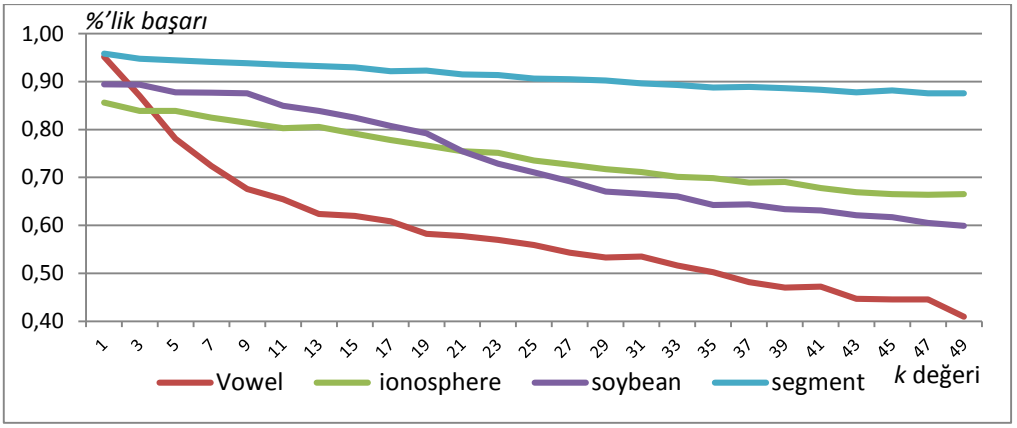
k En yakın komşuluk algoritması, Sınıflandırma, Kümeleme, Dinamik parametre seçimi
k Nearest neighbors algorithm, Classification, Clustering, Dynamic parameter selection

[1] YTÜ., Elektrik-Elektronik Fak., Bilgisayar Müh. Böl., Davutpaşa, İSTANBUL, f0110303@std.yildiz.edu.tr
[2] YTÜ, Elektrik-Elektronik Fak., Bilgisayar Müh. Böl., Davutpaşa, İSTANBUL, mfatih@ce.yildiz.edu.tr

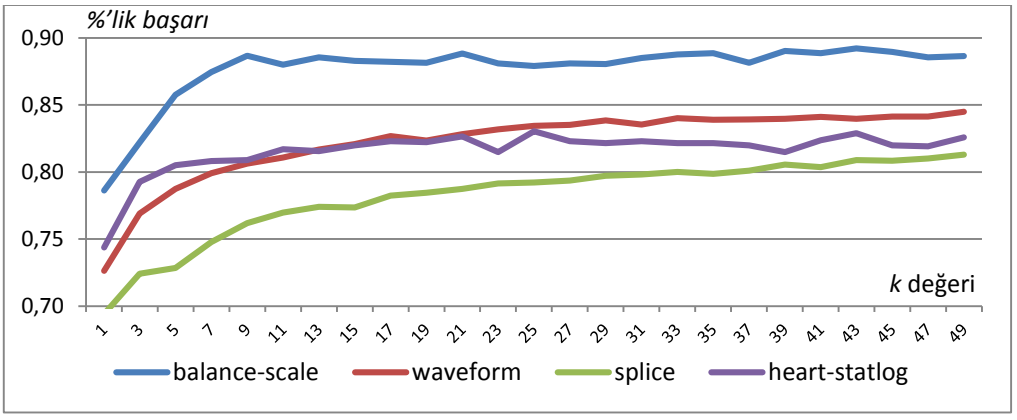
1. GİRİŞ

Yaygın bir kullanım alanına sahip k en yakın komşuluk algoritması (k-EKS), kullanıcı tarafından belirlenen sabit bir *k* değeri ile kullanılmaktadır. Literatürde en uygun *k* değerinin deneme yanılma yöntemi ile bulunduğu belirtilmektedir (Myatt, 2007a). Sabit *k* değerinin genel sınıflandırma başarısı üzerindeki olumsuz etkileri bu çalışmada incelenmiş ve bir takım çözümler önerilmiştir. Bu amaçla *k* parametresinin sınıflandırma başarısına olan etkisi değişik açılardan kanıtlanmaya çalışılmıştır.

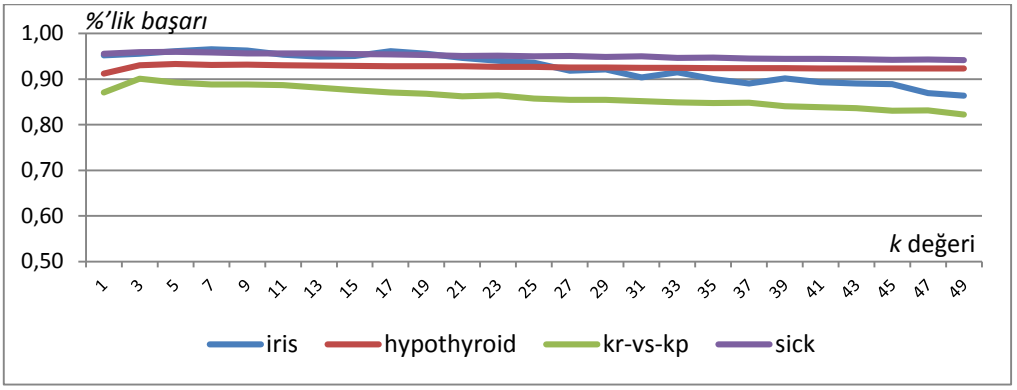
Şekil 1, Şekil 2, Şekil 3 ve Şekil 4'de görüldüğü üzere farklı *k* değerlerinin sınıflandırma başarısına olan değişik etkileri incelenmiştir. Bazı UCI veri setleri üzerinde k-EKS sınıflandırıcısı, 1'den 50'ye kadar *k* parametreleriyle sırayla denenmiş ve doğruluk (*accuracy*) oranları çapraz geçerleme işlemiyle elde edilmiştir (Bache ve Lichman, 2013). Görüldüğü üzere *k* parametresinin artırılmasıyla sınıflandırma başarısı Şekil 1'deki UCI veri setlerinde (*vovel, ionosphere, soyabean* ve *segment*) artmış; Şekil 2'deki veri setlerinde (*balance-scale, waveform, splice* ve *heart-statlog*) azalmış; Şekil 3'deki veri setlerinde ise (*iris, hypothroid, kr-vs-kp* ve *sick*) belirgin bir değişime uğramamıştır. Şekil 4'te ise daha farklı bir durum vardır. *colic, credit-a* ve *sonar* gibi veri setlerinde ise artan *k* parametresi ile performansta yer yer artmalar ve azalmalar meydana gelmektedir. Tüm bu durumlar birçok veri seti üzerinde k-EKS algoritması ile elde edilecek sınıflandırma başarısının *k* parametresi ile dorudan ilişkili olduğunu göstermektedir.



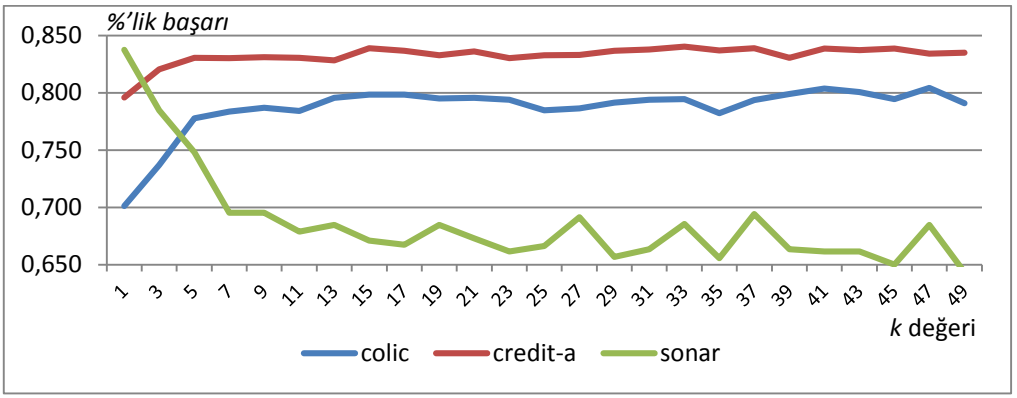
Şekil 1. k-EKS'de artan k parametresi ile performansın azalışı



Şekil 2. k-EKS'de artan k parametresi ile performansın artması



Şekil 3. k-EKS'de k parametresinin performansa etki etmemesi



Şekil 4. k-EKS'de k parametresi ile performansın değişmesi

Bir veri setinde her bir test örneğini doğru sınıflandırmak amacıyla faklı bir *k* değerine gereksinim duyulduğu Çizelge 1'de görülmektedir. *audiology* veri setinin test kısmından seçilen bazı örneklerin geçerleme işleminde, sınıflandırıcının farklı *k* değerleriyle bu test örneğini doğru sınıflandırıp sınıflandırmadığı gözlemlenmeye çalışılmıştır. 1 değeri sınıflandırıcının ilgili örneğin sınıf etiketini doğru tahmin ettiğini; 0 ise yanlış tahmin ettiğini göstermektedir. Bu çizelgeye göre *k* değeri 1 alındığında 8 örnekten sadece 3 tanesi doğru bilinebilirken; *k* değeri 4 alındığında 6 tanesi doru bilinebilmektedir.

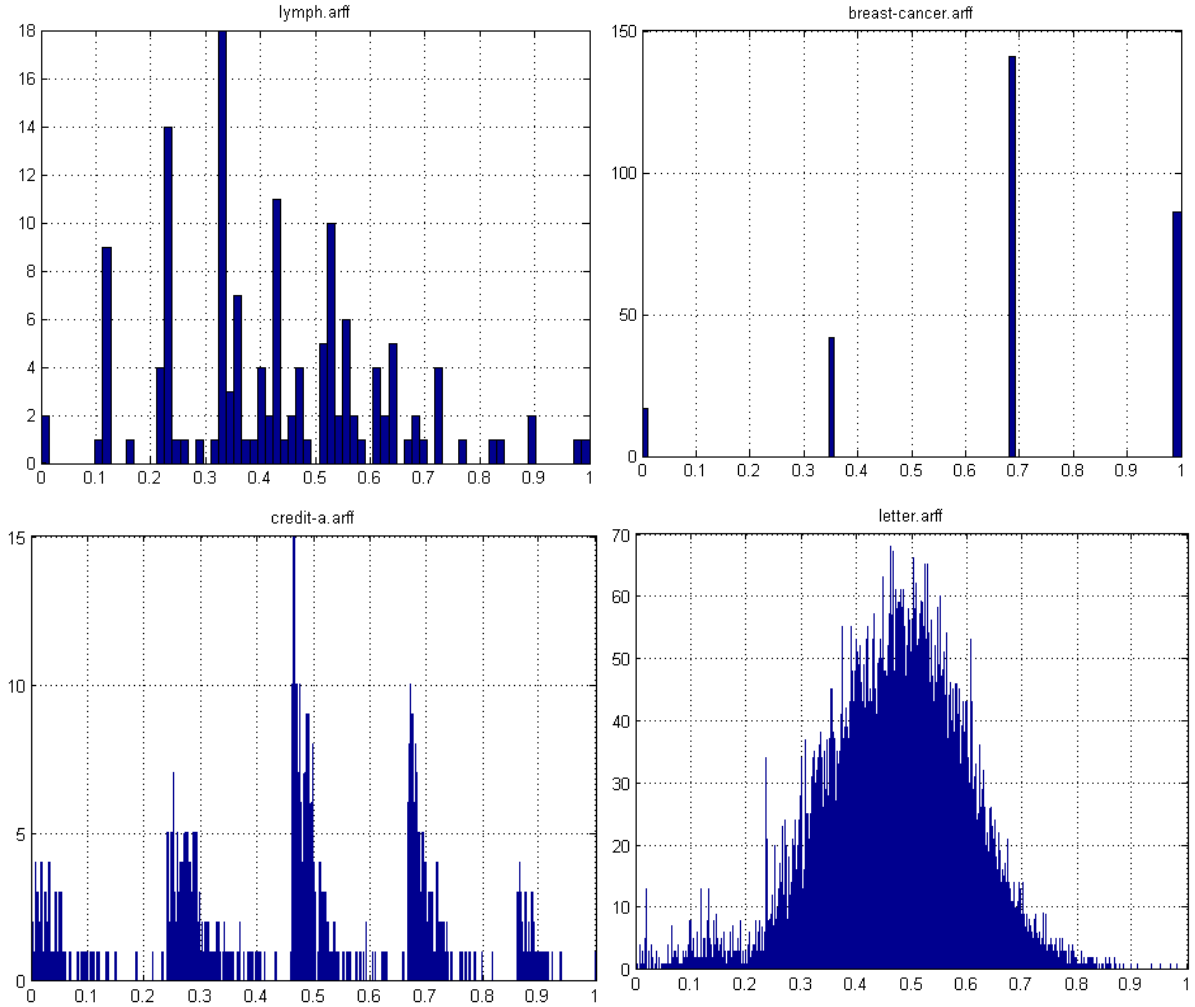
Çizelge 1. k değerinin başarıya etkisi

	k-EKS için *k* parametresi									
	1	2	3	4	5	6	7	8	9	10
1.örnek	0	0	0	0	0	0	0	1	1	1
2.örnek	1	1	1	1	1	1	1	1	1	1
3.örnek	0	0	0	0	0	0	0	0	0	0
4.örnek	0	0	1	1	1	1	1	1	1	1
5.örnek	0	0	0	1	1	1	0	0	0	1
6.örnek	1	1	1	1	1	1	1	1	0	0
7.örnek	1	1	1	1	1	0	1	1	1	1
8.örnek	0	0	0	1	0	0	0	0	0	0
Doğru tahmin sayısı	3	3	4	6	5	4	4	5	4	5

Çizelge 1'de görüldüğü üzere bazı test örnekleri için *k* değeri artırıldığında, azaltıldığında ya da belirli aralıklarda alındığında doğru tahmin yapılmaktadır. Bu durum bize genel

sınıflandırma başarısının artırmak amacıyla her test örneği için sabit bir *k* değeri yerine uygun bir *k* değerinin seçilmesi gerektiğini göstermektedir.

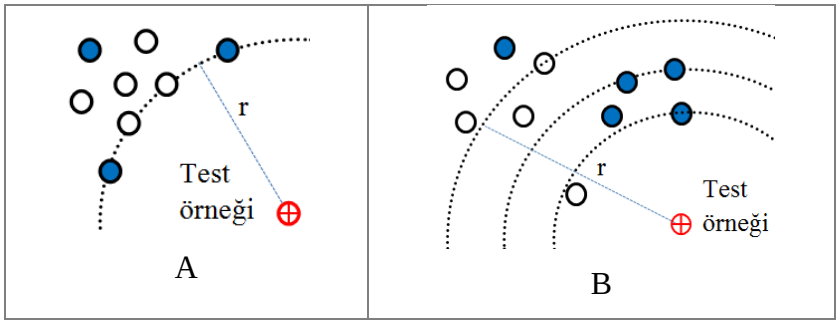
Şekil 5'de tüm verileri normalize edilmiş UCI veri setlerinden sırayla *lymph*, *breast-cancer*, *credit-a* ve *letter*'ın orijin noktalarına göre tüm noktaların uzaklıklarının histogramları gözükmektedir. Bu şekiller veri setlerinin orijin noktalarına göre kendi uzaylarındaki yayılımları hakkında bir fikir vermektedir. Orijindeki bir test noktasının k-EKS tekniği ile sınıflandırılmasında histogram grafiğine bakacak olursak en uygun *k* değeri *lymph* veri setinde 2; *breast-cancer*'da 15 alınması gerektiği anlaşılır. Çünkü orijindeki noktaya aynı uzaklıkta olan ve aynı yörünge üzerinde duran sırasıyla 2 ve 15 tane noktalar kümesi vardır. Bu durum bize herhangi bir test örneğinin etrafındaki noktaların uzaklıklarına bağlı olarak k-EKS için uygun bir *k* parametresinin bulunabileceğini göstermektedir. Diğer bir taraftan *letter* ve *credit-a* veri setinde ise en uygun *k* değeri için 1'den başlayarak deneme yanılma yöntemi uygulanabilir. Çünkü bu veri setlerinde orijin noktasına olan örneklerin uzaklıkları farklılık göstermektedir.



Şekil 5. Bazı veri setlerinin histogram bilgileri

Sınıfı belirlenmek istenen bir test noktası etrafında hemen hemen aynı uzaklıkta birden fazla nokta bulunabilir. Bu durumda k-EKS sınıflandırıcısı için seçilen k parametresinin 1 alınması durumunda rastgele seçimden ötürü sınıflandırma işlemi güvenilirliğini yitirmektedir. Şekil 6A'da görüldüğü üzere rastgele yapılan bu işlemde sınıf etiketi her denemede başka çıkabilmektedir. Bir test örneğine aynı uzaklıkta birden fazla örneğin bulunma ihtimali oldukça

azdır. Fakat benzer uzaklıkta birden fazla örnek bulunabilmektedir. Şekil 6B'deki senaryoda *k* değerinin 1 veya daha fazla alınması durumunda sonuç farklı çıkacaktır.



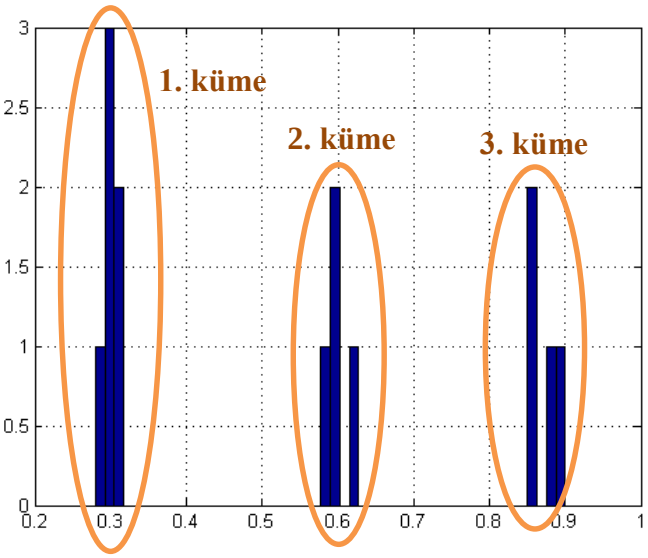
Şekil 6. Sınıflandırma örnekleri

Çalışmamız beş bölümden oluşmaktadır. İlk bölümde, yukarıda bahsedildiği gibi örnek tabanlı sınıflandırıcılarda her bir test girdisi için neden farklı parametrelere ihtiyaç duyulduğu açıklanmaya çalışılmıştır. 2. bölümde k-EKS'de her bir test örneği için en uygun k parametresini seçen bir sınıflandırıcı algoritmasına yer verilmiştir. 3. bölümünde bellek tabanlı sınıflandırıcılarda kullanılan arama yöntemlerine ve zaman karmaşıklıklarına; 4. bölümünde elde edilen deneysel sonuçlara ve son bölümde ise değerlendirmelere yer verilmiştir.

2. EN YAKIN KÜME İLE SINIFLANDIRMA

k-EKS algoritmasında her bir test örneği için en uygun k parametresinin seçilmesi gerektiğini düşüncesi bir önceki bölümde anlatılmıştır. Bu amaçla çalışmamızda k-EKS algoritmasına benzeyen fakat daha yüksek doruluk oranına sahip olabileceği düşünülen bir sınıflandırıcıya yer verildi. Bir en yakın küme sınıflandırıcısı (1EKS) diye isimlendirilen bu öğrenici, test noktasına en yakın ilk kümedeki örneklerin tamamını hesaplamaya katmaktadır. 1EKS tekniğinin işlem basamakları şu şekildedir:

1. *N*, veri setinde örnek sayısı olmak üzere, veri setine ait tüm değerleri normalize et,
2. Test noktasına en yakın *M* adet örneği al ve *l* adet kümeye böl,
3. Test noktasına en yakında noktanın ait olduğu kümenin tüm elemanlarını k-EKS yöntemiyle sınıflandırma işlemine al.



Şekil 7. Kümeleyerek sınıflandırma (1EKS) örneği

Şekil 7'teki örnek senaryoda test noktana değişik uzaklıkta olan noktaların histogram grafiği görülmektedir. Bu noktalar kümeleme yöntemiyle (*clustering*) 3 adet kümeye bölünmüştür. 1EKS sınıflandırıcısı için en yakında bulunan küme içerisindeki 6 adet örnek hesaplamaya katılacaktır (k-EKS için *k*=6 alınacaktır).

Test noktasına en yakın M adet örneğin l adet kümeye bölünmesi ve sadece en yakındaki ilk kümenin işleme dâhil edilmesi dinamik bir yapıyı oluşturmaktadır. l adet kümenin (*k*-means'deki *k* ifadesinin k-EKS'deki *k* ile karıştırılmaması için l ifadesi tercih edilmiştir) her birinde yaklaşık olarak M/l adet eleman olduğu düşünülebilir. Normalde k-EKS sınıflandırıcısı için kullanıcı tarafından seçilen *k* parametresi ile bizim yöntemimizdeki M/l kombinasyonuna denk olması şu formül ile sağlanabilir:

$$k\text{NN}'deki\ k \cong \frac{M}{l} \quad (1)$$

Bu sayede M/l ikilisiyle her bir test örneğinin sınıflandırılması için uygun bir *k* değeri hesaplanmış olmaktadır.

Üzerinde çalışılan 1EKS tekniğinde *k-EKS* ve *k*-means yöntemlerinin toplam zaman karmaşıklıkları olduğu için normal *k-EKS* sınıflandırıcısına göre bir miktar daha maliyetlidir (Myatt, 2007b).

3. EN YAKIN ÖRNEKLERİ ARAMA YÖNTEMLERİ

k en yakın komşuluk algoritması (k-EKS) bellek tabanlı bir sınıflandırıcıdır ve sınıflandırma işleminde her bir test örneği için eğitim setinde ayrı ayrı arama yapılmaktadır. Bu durum hesaplama zamanını artırmaktadır. Uygulamamızda hesaplama zamanını düşürmek için iki tip arama algoritması kullanılmıştır. Literatürde bir veri setinde boyut sayısı 10'dan büyük ise tam kapsamlı arama (*Exhaustive Search*); 10'dan küçük ise kB-Ağaç veri yapısı ile yapılan arama yöntemini tavsiye edilmektedir (MATLAB R2014a Tutorial, 2014). Kullanılan arama yöntemleri algoritmanın başarısını etkilememektedir, sadece hesaplama süresi üzerinde etki yapmaktadır.

3.1. Tam Kapsamlı Arama

Tam kapsamlı arama (*Exhaustive Search*) ile hiçbir veri yapısı ve algoritma kullanılmadan Öklid uzaklığına göre test noktasına en yakın örnek sıralı bir şekilde aranır. Bu arama yönteminin zaman karmaşıklığı oldukça yüksektir. D veri setinin boyut sayısı, N de eleman sayısı olmak üzere bu algoritmada Big-O notasyonuna göre istenilen bir elemana en yakın M adet noktaya ulaşmanın zaman karmaşıklığı $O(D * M * N)$'dir (Weiss, 2013).

3.2. kB-Ağaç İle Arama

Jon Bentley tarafından 1975 yılında geliştirilen *k Boyutlu Ağaç (*kB-Ağaç) veri yapısı, *İkili Uzay Bölütleme* yöntemlerinden biridir ve ikili arama ağacı olan *İkili Arama Ağacı* veri yapısının çok boyutlu türüdür. Uzayda bulunan noktalar her bir düzlemde bir doğruyla iki ayrı bölüme (*partition*) ayrılır. Bu işlem özyinelemeli olarak her bir bölümde belirlenen sayıda noktalar kalana dek devam eder.

Big-O notasyonuna göre kD ağacının kurulumu ile ilgili zaman karmaşıklığı $O(D * N * \log N)$'dir. Kurulumdan sonra sadece bir örneğini aramanın maliyeti $O(D * \log N)$; her hangi bir test noktasına en yakın M adet noktanın bulunma maliyeti ise $O(D * M * \log N)$'dir. Bu durum tam kapsamlı aramada $O(D * M * N)$ şeklindedir. D ve M değerlerinin sabit olduğu

düşünülürse karmaşıklıklar $O(N)$ ve $O(\log N)$ şeklinde olacaktır. Görüldüğü üzere tam kapsamlı aramaya göre kB-Ağaç ile arama daha avantajlıdır.

4. PRATİK UYGULAMA VE DENEYSEL SONUÇLAR

Verileri normalize edilmiş, kayıp değerleri yer değiştirilmiş, nominal değerleri ikili sayısal değerlere dönüştürülmüş 36 adet UCI veri seti, MATLAB ortamında 5x2 çapraz geçerleme ile üzerinde çalışılan sınıflandırıcılarla test edilmiştir. Çalışmamızın doğruluğunu sınamak için 5x2 kat çapraz geçerleme (*5x2 Fold Cross Validation*) yöntemi tercih edilmiştir. Bu işlemde 2 kat çapraz geçerleme işlemi 5 defa tekrarlanarak yapılır. 2 kat çapraz geçerleme işleminde eğitim seti rastgele 2 parçaya ayrılır. İlk parça test seti, ikici parça eğitim setidir. Daha sonra test setindeki kayıtların yüzde kaçının doğru bilindiği hesaplanır. İlk işlem tamamlandıktan sonra bu sefer birinci parça test seti iken eğitim setine; ikinci parça eğitim seti iken test setine dönüştürülerek yüzdelik başarı tekrar hesaplanır. Bu işlemler 5 defa tekrarlanarak 10 tane yüzdelik başarı sonucu elde edilir (Alpaydın, 2010). Çapraz geçerleme işleminin sonucu ise elde edilen 10 tane sonucun aritmetik ortalaması alınarak bulunur. Bulunan sonuç sınıflandırma işleminin çalışılan eğitim seti üzerindeki başarısını gösterir. Veri setlerinde arama yöntemi olarak kD ağaç veri yapısı ve tam kapsamlı arama yöntemlerinden uygun olanı kullanılmıştır.

Çizelge 2'te yaptığımız çalışmanın sonuçları verilmektedir. Her bir veri setine ait örnek (*instance*) sayısı, özellik sayısı (*attribute*) ve sınıf etiketi (*class label*) sayısı sütunlar halinde sırasıyla verilmektedir. Bu bilgiler veri setlerinin yoğunluğu ve istatistiksel yapısı hakkında bir ön fikir vermektedir. Sonraki iki sütunda ise veri setlerinin *k-EKS* sınıflandırıcısı ile verdiği en yüksek doğruluk oranını hangi *k* parametresi ile verdiği gösterilmektedir. Yaptığımız uygulamalarda 36 adet veri setinde k-EKS algoritmasının 1 ile 100 arasındaki tüm *k* parametreleri tek tek denenmiş ve en yüksek doğruluk oranının hangi *k* değerinde bulunduğu tespit edilmiştir. Şekil 8'de de görüldüğü üzere 14 adet veri setinde en iyi sınıflandırma *k*=1 alındığında gerçekleşmiştir.

Çizelge 2'nin 1NN sütununda *k*=1 alındığında k-EKS'in başarısı listelenmiştir. Sıradaki sütunda ise *k*=5 alınarak elde edilen k-EKS sınıflandırıcısının; sonraki sütunda *en yakındaki küme* (1EKS) sınıflandırıcısı ile elde edilen doğruluk oranları görülmektedir. 1EKS sınıflandırıcısı ile *k-EKS* sınıflandırıcısını karşılaştırabilmek için yapılan uygulamalarda k-EKS için *k* parametresi 5 alınmıştır ve tüm veri setleri için doğruluk sonuçları hesaplanmıştır. 1EKS'de her bir test noktasına en yakın (*M* değeri) 20 eleman alınarak *k*-means kümeleme yöntemiyle (iterasyon sayısı=100 ve *k*-means için *k* parametresi=4 alındı) 4 adet kümeye (*l* değeri) bölünmüştür. Bu durumda 4 adet kümenin her birinde yaklaşık olarak 5'er adet örnek bulunduğu düşünülebilir. Bu durum k-EKS'de *k*=5 anlamına geldiği düşünülebilir.

İki farklı sınıflandırma mekanizması tarafından elde edilen sonuçların istatistiksel anlamlılığını tespit etmek için T-Test yöntemi tercih edilmiştir (Demsar, 2006; Berg, 2008). İki sınıflandırma mekanizmasının karşılaştırılmasında T-Test yöntemi üç farklı sonuç vermektedir: *win* (birinci sınıflandırıcı daha başarılı), *loss* (başarısız) ve *tie* (eşit) şeklindedir. k-EKS (*k*=5) ile 1EKS $(M/l = 20/4)$ sınıflandırıcıları ile elde edilen doğruluk oranlarının yüzdelik artış-azalış oranları ve bu iki sınıflandırıcının T-Test sonuçları 1EKS/k-EKS *karşılaştırma* isimli sütununda görülmektedir. Bu karşılaştırmada %15'e kadar sınıflandırma başarısında artış olduğunu gözlemlenmiştir. Her iki sınıflandırıcının karşılaştırması sonucunda 8 adet veri kümesinde başarı (*win*) sağlanmış, 20'sinde değişme olmamış (*tie*) ve 8 tanesinde başarısız (*loss*) olunmuştur. Ayrıca 1EKS-1NN sütununda da görüldüğü üzere 1EKS yöntemi, 1NN ile karşılaştırıldığında daha başarılı sonuçlar elde edilmiştir: 12 adet *win*, 20 adet *tie* ve sadece 4 adet *loss*. Görüldüğü üzere 1EKS algoritması ile bazı veri setlerde daha başarılı sonuçlar elde edilmiştir.

1EKS için M/l ikililerinden k-EKS'deki k=5 parametresine denk gelen 10/2, 15/3, 30/6, 50/10 ve 100/20 kombinasyonları ayrı ayrı denenmiştir. Denemelerde 20/4 ile elde edilen benzer doğruluk oranlarına ve hemen hemen aynı sayıda *win*, *tie* ve *loss* sonuçlarına ulaşılmıştır. Bu sonuçlar üç farklı çıkarım yapılmasını sağlamıştır:

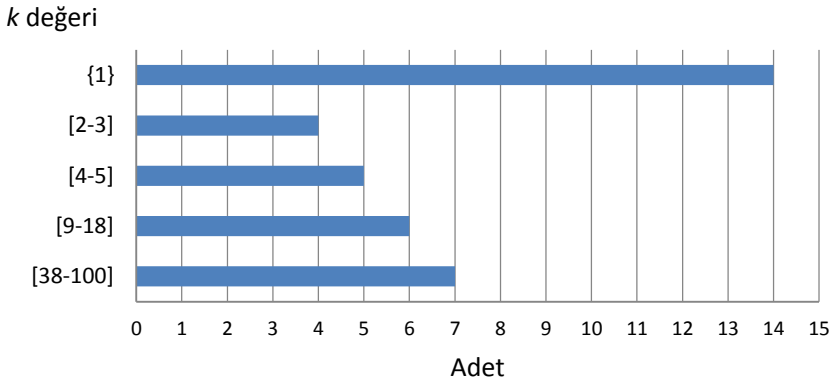
1. Benzer sonuçları veren düşük değerlikli M/l ikilisinin tercih edilmesi hesaplama süresinin kısalmasını sağlar.
2. k-EKS'deki k'ya eşit olması için genel olarak M, k'nın 3 katı alınabilir ve l değeri de 3'e sabitlenebilir ($k \cong M/l$ olduğunu hatırlayınız).
3. Bu sayede M/l ikilisi tek bir parametre gibi düşünülebilir.

Çizelge 2: Algoritmaların 36 veri kümesi üzerinde karşılaştırılması

				En iyi *k-EKS* sonucu					1EKS-kNN karşılaştırma		1EKS-1NN karşılaştırma	
Veri seti	Örnek sayısı	Özellik sayısı	Sınıf sayısı	*k*	Doğruluk oranı	1NN	*k-EKS* *k*=5	1EKS *M*/*l*=20/4	%'lik artış	T-test sonucu	%'lik artış	T-test sonucu
abalone	4153	19	10	55	0.2682	0.2023	0.2301	0.2249	-2.26	loss	11.15	win
anneal	890	4	62	1	0.9769	0.9769	0.9584	0.9715	1.36	tie	-0.55	tie
audiology	169	5	69	4	0.7053	0.6757	0.6852	0.6556	-4.32	loss	-2.98	loss
autos	202	5	71	1	0.6505	0.6505	0.5723	0.6010	5.02	win	-7.61	loss
balance-scale	625	3	4	100	0.8931	0.7894	0.8576	0.8547	-0.34	tie	8.27	win
breast-cancer	286	2	38	9	0.7308	0.6643	0.7098	0.7084	-0.20	tie	6.64	win
breast-w	699	2	9	5	0.9671	0.9548	0.9671	0.9557	-1.18	tie	0.09	tie
col10	2019	10	7	1	0.7241	0.7249	0.7072	0.7168	1.36	tie	-1.12	tie
colic	368	2	60	74	0.8196	0.6957	0.7777	0.7380	-5.10	loss	6.09	win
credit-a	690	2	42	38	0.8452	0.7901	0.8304	0.8043	-3.14	loss	1.80	tie
credit-g	1000	2	59	14	0.7248	0.6824	0.7176	0.7116	-0.84	tie	4.28	win
d159	7182	2	32	1	0.9453	0.9451	0.9404	0.9490	0.92	tie	0.42	tie
diabetes	768	2	8	15	0.7466	0.6943	0.7286	0.7143	-1.97	tie	2.88	tie
glass	205	5	9	1	0.6780	0.6713	0.6410	0.6644	3.65	win	-1.03	tie
heart-statlog	270	2	13	62	0.8356	0.7467	0.8052	0.7785	-3.31	loss	4.26	win
hepatitis	155	2	19	5	0.8361	0.7948	0.8361	0.8039	-3.86	loss	1.14	tie
hypothyroid	3770	3	31	5	0.9329	0.9125	0.9329	0.9289	-0.44	tie	1.79	tie
ionosphere	351	2	33	1	0.8558	0.8598	0.8387	0.8701	3.74	win	1.19	tie
iris	150	3	4	10	0.9693	0.9467	0.9613	0.9520	-0.97	tie	0.56	tie
kr-vs-kp	3196	2	39	3	0.9008	0.8891	0.8923	0.9260	3.78	win	4.15	win
labor	57	2	26	1	0.8807	0.8732	0.8421	0.8316	-1.25	tie	-4.76	loss
letter	20000	26	16	1	0.9441	0.9438	0.9343	0.9416	0.78	tie	-0.23	tie
lymph	142	2	37	12	0.8338	0.7592	0.7915	0.7859	-0.71	tie	3.52	win
mushroom	8124	2	112	1	1.0000	1.0000	0.9999	0.9998	-0.01	tie	-0.02	tie
prim.-tumor	302	11	23	18	0.4755	0.3874	0.4430	0.4291	-3.14	loss	10.77	win
ringnorm	7400	2	20	2	0.7915	0.7257	0.6623	0.7354	11.03	win	1.33	win
segment	2310	7	18	1	0.9583	0.9580	0.9443	0.9539	1.01	tie	-0.43	tie
sick	3772	2	31	5	0.9598	0.9569	0.9598	0.9562	-0.38	tie	-0.07	tie
sonar	208	2	60	1	0.8375	0.8375	0.7481	0.8433	12.72	win	0.69	tie
soybean	675	18	83	1	0.8942	0.8916	0.8776	0.8806	0.34	tie	-1.23	tie
splice	3190	3	287	99	0.8413	0.7357	0.7285	0.7628	4.71	win	3.68	win
vehicle	846	4	18	3	0.6839	0.6723	0.6825	0.6752	-1.07	tie	0.43	tie
vote	435	2	16	3	0.9297	0.9228	0.9297	0.9264	-0.35	tie	0.39	tie
vowel	990	11	11	1	0.9517	0.9473	0.7806	0.8994	15.22	win	-5.05	loss
waveform	5000	3	40	75	0.8492	0.7278	0.7875	0.7476	-5.06	loss	2.72	win
zoo	84	4	16	1	0.9976	0.9987	0.9929	0.9762	-1.68	tie	-2.25	tie

Ayrıca tüm veri setleri için k-EKS'de k değeri, sırasıyla 1'den 100'e kadar alınarak sınıflandırma başarıları elde edilmiştir. Şekil 8'de de görüldüğü üzere k-EKS sınıflandırıcı için

genelde küçük değerlikli *k* parametresi ile en yüksek başarı elde edilmiştir. 14 adet veri setinde en yüksek başarı, *k*=1 alındığında hesaplanmıştır. Sadece 7 adet veri kümesinde 38 ve üzeri *k* değeri ile en başarılı sonuçlar elde edilmiştir. k-EKS için en uygun parametrenin küçük değerler olması gerektiği bazı çalışmalarda ispatlanmıştır (Özger vd., 2013).



Şekil 8. *k-EKS* sınıflandırıcısında en iyi sonucu veren *k* değerlerinin adedi

1NN'in en iyi sınıflandırıcı olduğu bazı veri kümeleri üzerinde 1EKS yönteminin daha yüksek başarı elde etmesi mümkün değildir diye düşünülmüştü. Fakat 1EKS yönteminde M/l ikilisi 20/4 alındığında *ionosphere, d159* ve *sonar* gibi bazı veri setlerinde 1NN'e göre daha da yüksek başarılar elde edilmiştir. Bu durum, 1NN'den farklı olarak 1EKS yönteminde her bir test örneği için esnek ve dinamik bir parametre oluşturma mekanizması ile ilgilidir. Diğer veri setlerindeki (*autos, glass, vowel, anneal, col10, labor, letter, mushroom, segment, soybean, zoo*) T-Test analizinde *tie* sonucu alınmıştır. Bu durum bazı veri setler için 1EKS'nin, en başarılı sonucu veren küçük *k* parametreli k-EKS'den bile daha başarılı olabileceğini göstermektedir.

5. SONUÇ

k-EKS sınıflandırıcısında her bir test noktası için sabit bir *k* parametresinin çapraz geçerleme işlemi boyunca kullanıldığı ve bu durumun genel sınıflandırma başarısını düşürdüğü çalışmamızda anlatılmıştır. Çalışmamızda genel sınıflandırma başarısını artırmak amacıyla her bir örnek için en uygun ve farklı *k* değerlerinin kullanılması üzerine bir yöntem önerilmiştir. Kümele yönteminden yararlanılarak yapılan sınıflandırma işleminde daha yüksek doğruluk oranları edilmiştir. Yapılan istatistiksel analizlerde üzerinde çalışılan karma sınıflandırma sisteminin yalın k-EKS sınıflandırıcısına göre daha başarılı olduğu tespit edildi. Ayrıca üzerinde çalışılan yöntem, k-EKS'de olduğu gibi kullanıcı tarafından atanan tek bir parametre ile çalışmaktadır. Diğer bir taraftan üzerinde çalıştığımız yöntemde zaman karmaşıklığı bir miktar armış oldu.

KAYNAKLAR

Alpaydın E. (2010): "Yapay Öğrenme kitabı", Boğaziçi Üniversitesi Yayınevi, ISBN: 978-6-054-23849-1, İstanbul, s.1-35.

Bache K., Lichman M. (2013): "UCI Machine Learning Repository Official", http://archive.ics.uci.edu/ml, Irvine, University of California, ABD.

Berg M., Eindhoven T. U. (2008): "Computational Geometry: Algorithms and Applications", ISBN: 978-3-540-77973-5, Springer Publishing, s.99-105.

Demsar J. (2006): "Statistical Comparisons of Classifiers over Multiple Data Sets", Journal of Machine Learning Research 7, s.1-30.

MATLAB R2014a Tutorial (2014): "KD Tree Searcher class Tutorial", www.mathworks.com/help/stats/

Myatt G. J. (2007a): "Making Sence of Data: A Practical Guide to Exploratory Data Analysis and Data Mining", Wiley, s.176-181.

Myatt G. J. (2007b): "Making Sence of Data: A Practical Guide to Exploratory Data Analysis and Data Mining", Wiley, s.120-129.

Özger Z. B., Amasyalı M. F. (2013): "Meta Öğrenme ile KNN Parametre Seçimi KNN Parameter Selection Via Meta Learning", IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SİU2013), ISBN: 978-1-4673-5562-9, Girne, KKTC, s.1-4.

Weiss M. A. (2013): "Data Structures and Algorithm Analysis in C++", Pearson, ABD, s.83-85, 614-618 ve 629.