Academic Platform Journal of Engineering and Science

journal homepage: http://apjes.com/

Elipsoit Destek Vektör Öbekleme Algoritmasının Biyomedikal Veri Setleri Üzerinde Karşılaştırmalı Performans Analizi

[1]Furkan Burak Bağcı, [*2]Ömer Karal

Ankara Yıldırım Beyazıt Üniversitesi, Mühendislik Fakültesi, Ankara, D1000000832@ybu.edu.tr,

Ankara Yıldırım Beyazıt Üniversitesi, Mühendislik Fakültesi, Ankara, karal@ybu.edu.tr,

Araştırma Makalesi Geliş Tarihi: 16.05.2018 Kabul Tarihi: 06.10.2018

Öz

Hastalıklı kişilerde hastalığın teşhisinin önceden yapılması, tanısının konulması ve gerekli önlemlerin alınmasına yardımcı olmalarından dolayı öbekleme algoritmalarının performansı biyomedikal araştırmalarda çok önemlidir. Ancak, çoğu öbekleme algoritması benzerlik metriği olarak Öklid uzaklığını kullanır. Öklid uzaklığı verilerin varyanslarını eşit kabul eder. Gürültülü veya aykırı değerlerin veriye bulaşması durumunda, geleneksel Öklid uzaklığı kullanan öbekleme yöntemlerinin performansı oldukça düşmektedir. Bu çalışma, yukarıda bahsedilen olumsuzlukları gidermek için kernel tabanlı öbekleme yöntemlerinden biri olan Elipsoit Destek Vektör Öbekleme (EDVÖ) algoritmasını önerir. EDVÖ algoritmasında, önceden öbek sayısının belirtilmesine gerek yoktur. Ayrıca, EDVÖ algoritması, mahalanobis benzerlik ölçüsünü kullanarak verilerin dağılımına uygun kümelenme şekilleri üretebilir. Önerilen EDVÖ algoritması hem gerçek biyomedikal verilere hem de sentetik verilere uygulanmış ve daha sonra geleneksel kümeleme yöntemleri ile karşılaştırılmıştır. EDVÖ algoritmasının doğruluk, özgüllük ve duyarlılık açısından iyi bir performans gösterdiği gözlemlenmiştir.

Anahtar kelimeler: Biyomedikal veriler, öbekleme, elipsoit destek vektör öbekleme

Comparative Performance Analysis of Ellipsoidal Support Vector Clustering on Biomedical Data Sets

[1] Furkan Burak Bağcı, *[2] Ömer Karal

[1] D1000000832@ybu.edu.tr

[2] karal@ybu.edu.tr

Abstract

The performance of clustering algorithms is very important in biomedical research because they help in the pre-diagnosis of diseases, recognize diseases and take necessary precautions in diseased people. However, most clustering algorithms use the Euclidean distance as a similarity metric. Euclidean distance assumes the variances of the data samples are equal. The performance of traditional clustering methods that use Euclidean distance is quite low if the data contains noise or outlier samples. This study proposes the Ellipsoidal Support Vector Clustering algorithm, which is one of the kernel-based clustering methods, in order to eliminate the above mentioned problems. In the ESVC algorithm, there is no need to specify the cluster number in advance. Moreover, the ESVC algorithm is capable of generating clustering shapes that are appropriate to the distribution of data using the mahalanobis similarity metric. The proposed ESVC algorithm was applied to both real biomedical data and synthetic data and then compared to conventional clustering methods. It has been observed that ESVC algorithm performs well in terms of accuracy, specificity and sensitivity.

Keywords: Biomedical data, clustering, ellipsoidal support vector clustering

* Sorumlu Yazar: Omer KARAL Ankara Yıldırım Beyazıt Üniversitesi, Mühendislik Fakültesi, Ankara, D1000000832@ybu.edu.tr

Doi: 10.21541/apjes.424247

1. GİRİŞ

Denetimsiz öğrenme yöntemlerinden biri olan öbekleme algoritmalarının amacı, sınırlı bir veri kümesini, kümeler arası benzerliği en aza indirecek şekilde sonlu sayıda öbeklere ayırmaktır [1]. Bu ayırma işleminin sonucunda, aynı öbek içinde yer alan veriler birbirine büyük oranda benzer özellikler gösterirken, farklı öbeklerde yer alan veriler de söz konusu benzerlik oldukça azalmaktadır [2]. Sınıflandırma problemlerinden farklı olarak, öbekleme problemlerinde verilere ait etiket bilgileri önceden bilinmemektedir [3].

Öbekleme algoritmaları, öbekleme sonucu elde edilen verileri kullanarak ilgili alanda yeni kararların verilebilmesini sağlar. Örneğin, hastalıklı kişilerde, hastalık teşhisinin önceden yapılması, tanısının konulması ve gerekli önlemlerin alınmasında öbekleme yöntemleri ilk başvurulan yöntemler arasındadır. Özellikle, k-merkez (k-means) öbekleme [4], hiyerarşik öbekleme [5], bulanık C-merkez (Fuzzy C-Means - FCM) öbekleme [6] gibi algoritmalar biyomedikal araştırmalarda sıklıkla kullanılmaktadır.

Son zamanlarda, farklı öbekleme algoritmalarının aynı veriler üzerinde uygulanması ve sonuçlarının (performanslarının) karşılaştırılmasına ilişkin çalışmaların sayısında önemli bir artış görülmektedir. Velmurugan [7], k-merkez ve FCM öbekleme algoritmalarını haberleşme veri setine uygulamış ve performans tabanlı analizini gerçekleştirmiştir. Velmurugan k-merkez öbekleme algoritmasına ait hesaplama süresinin FCM öbekleme algoritmasına ait hesaplama süresinden daha kısa sürdüğünü ancak gürültüye karşı daha hassas olduğunu vurgulamıştır. Erken [8], k-merkez, spektral ve Girvan-Newman öbekleme algoritmalarını göğüs kanseri veri setine uygulayarak artı ve eksi yönlerini incelemiştir. Elde edilen sonuçlardan, k-merkez öbekleme algoritmasının en hızlı sonucu veren algoritma olduğu, spektral öbekleme algoritmasında bazı sapmaların ortaya çıktığı ve son olarak Girvan-Newman öbekleme algoritmasında da çok fazla süreye ihtiyaç duyulduğu saptanmıştır. Sharma ve ark. [9], hiyerarşik maksimum olasılık tabanlı öbekleme algoritması ile k-merkez öbekleme algoritmasını genom verilerine uygulayarak öbekleme performanslarını karşılaştırmışlardır. Buna göre, özellikle üst üste çakışan öbeklerde hiyerarşik maksimum olasılık tabanlı öbekleme algoritmasının daha iyi sonuç verdiğini ve böylece veri sayısının, verinin boyutundan daha az olduğu durumlarda bile kullanışlı olduğunu göstermiştir.

Yukarıda bahsi geçen öbekleme algoritmaları, veri örnekleri arasında benzerlik metriği olarak, Öklid uzaklığını kullanırlar. Öklid uzaklığı verilerin varyanslarını eşit kabul eder. Ancak, çoğu gerçek dünya verisi farklı varyansta veriler içerebilir.

Son yıllarda, verinin dağılımı hakkında ön bilgi gerektirmeksizin çalışabilmesi ve kullanılan kernel (çekirdek) fonksiyonundaki parametre sayesinde veriye uygun öbek sınırlarını oluşturabilmesi gibi özelliklerinden dolayı, Destek Vektör Öbekleme (DVÖ) yöntemi oldukça dikkat çekmiştir [10-15].

DVÖ algoritması ile gerçekleştirilen öbekleme işlemlerinde, k-merkez ve FCM öbekleme algoritmalarında olduğu gibi, veri örnekleri arasında, benzerlik metriği olarak Öklid uzaklığı kullanılır. Öklid uzaklığında, veri örnekleri arasındaki farkın karesi minimize edilir. Eğer gürültülü veya aykırı örnekler veri setinde bulunursa, Öklid metriğini kullanan öbekleme algoritmalarında hatalı öbeklerin elde edilmesi kaçınılmaz olur. Yukarıda belirtilen problemin üstesinden gelmek için, Wang ve ark. [16], Öklid uzaklığı yerine Mahalanobis uzaklığı kullanan kernel tabanlı yeni bir öbekleme algoritmasını önermiş ve Elipsoit Destek Vektör Öbekleme (EDVÖ) algoritması olarak adlandırmıştır.

Literatürde, herhangi bir hastalığa ait verilerin analizinde, EDVÖ yönteminin uygulandığı bilimsel bir çalışmaya rastlanmamıştır. Ancak, yakın geçmişte, fonksiyonel MRI verisinde aktif bölgelerin tespit edilmesi için EDVÖ algoritması önerilmiştir [16]. Elde edilen sonuçlardan, EDVÖ algoritmasının k-merkez ve FCM gibi geleneksel öbekleme algoritmalarından daha iyi bir performans gösterdiği gözlemlenmiştir.

Bu çalışmada, EDVÖ yöntemi ilk kez sentetik yüzük (Ring) veri seti ile University of California Irvine (UCI) veri deposundan [17] alınan Hepatit ve Parkinson hastalıklarına ait gerçek veri setlerine uygulanmıştır. Ayrıca, söz konusu veri setleri üzerinde, literatürde yaygın olarak kullanılan k-merkez, hiyerarşik ve FCM gibi geleneksel öbekleme algoritmaları ile EDVÖ algoritmasının karşılaştırmalı performans analizleri de gerçekleştirilmiştir.

Makalenin geri kalan kısmı; 2. bölümde DVÖ algoritmasının özeti ve EDVÖ algoritmasının teorik detayından, 3. bölümde önerilen EDVÖ yöntemi ile FCM, k-merkez, hiyerarşik gibi literatürde yaygın olarak kullanılan öbekleme yöntemlerinin performans analizinden ve son bölümde sonuç ve gelecekte yapılması düşünülen çalışmalara ait bilgilerden oluşur.

2. MATERYAL VE YÖNTEM

Bu bölümde, çalışmamızda kullanılan EDVÖ algoritması teorik ayrıntıları ile açıklanmıştır. EDVÖ, DVÖ algoritması temel alınarak oluşturulan bir algoritmadır. Bundan dolayı, ilk olarak, DVÖ algoritmasının teorik altyapısı verilmiş ardından EDVÖ algoritmasının teorisi DVÖ teorisine dayanarak detaylı bir şekilde açıklanmıştır.

2.1. Destek Vektör Öbekleme

Destek vektör öbekleme yöntemi iki ana aşamadan oluşur. İlk aşamada, giriş uzayında verilen veriler doğrusal olmayan bir kernel fonksiyonu ile yüksek boyutlu bir uzaya taşınır. Gauss fonksiyonları en sık seçilen kernel fonksiyonlarından biridir. Bu yüksek boyutlu uzayda, tüm veri noktalarını

kapsayacak şekilde en küçük yarıçaplı küre tanımlanmaya çalışılır.
İkinci aşamada, yüksek boyutlu uzayda elde edilen kürenin giriş uzayına iz düşümü alınır. Giriş uzayındaki bu iz düşüm birbiri ile kesişmeyen çeşitli şekiller (öbekler) oluşturur. Şekillerin sınırları öbek sınırları olarak kabul edilir.

DVÖ algoritmasının matematiksel formülasyonu aşağıdaki gibi tanımlanır. Giriş uzayındaki N tane d boyutlu veri $\{x_1, x_2, \dots, x_N\}, x_i \in \mathbb{R}^d$ doğrusal olmayan bir dönüşüm fonksiyonu Φ ile özellik uzayına taşınsın. Özellik uzayında, verilerin tamamını içine alacak şekilde yarıçapı en küçük olan bir hiper küre belirlenmeye çalışılır. Ayrıca, az bir hatayla oluşturulacak bu kürenin hemen dışında kalan verilerin de kümeye dâhil edilebilmesi için ξ parametresi tanımlanır. Böylece daha esnek küme sınırları oluşur. Bu hiper küre denklem 1'de ifade edilmiştir.

$$\begin{gathered} \min \left\{R^2 + C\sum_{j=1}^{N}\xi_j\right\} \\ subject\ to \\ \|\Phi(x_i) - a\|^2 \le R^2 + \xi_j \\ R > 0, \xi_j \ge 0, j = 1, \dots, N \end{gathered} \tag{1}$$

Bu denklemde R hiper kürenin yarıçapı, a hiper kürenin merkezi, C ise regülarizasyon sabitidir. Denklem 1'de verilen optimizasyon problemini çözmek için Lagrange çarpanları yöntemi kullanılır (denklem 2).

$$\begin{gathered} L = R^2 - \sum_j (R^2 + \xi_j - \left\|\Phi(x_j) - a\right\|^2)\beta_j - \\ \sum_j \xi_j\mu_j + C\sum_j \xi_j \\ \beta_j \ge 0, \mu_j \ge 0, i = 1, \dots, N \end{gathered} \tag{2}$$

β_j ve μ_j Lagrange katsayılarıdır.

Denklem 2'nin R, a ve ξ parametrelerine göre türevi alınıp sıfıra eşitlenirse, denklem 3, 4 ve 5'te gösterilen eşitlikler elde edilir.

$$\sum_j \beta_j = 1 \tag{3}$$

$$a = \sum_j \beta_j \Phi(x_j) \tag{4}$$

$$\beta_j = C - \mu_j \tag{5}$$

Karush-Kuhn-Tucker (KKT) şartlarına göre optimal çözümde Lagrange çarpanları ile kısıtlar arasındaki çarpımın sıfır olması gerekir (denklem 6 ve 7).

$$\xi_j\mu_j = 0 \tag{6}$$

$$(R^2 + \xi_j - \left\|\Phi(x_j) - a\right\|^2)\beta_j = 0 \tag{7}$$

Denklem 7'den $\xi_j > 0$ ve $\beta_j > 0$ şartını sağlayan veri örneklerinin, hiper kürenin dışında kaldığı anlaşılmaktadır. Denklem 6'dan $\xi_j > 0$ şartına göre $\mu_j = 0$ olmalıdır. Bu durum, denklem 5'den görüleceği üzere $\beta_j = C$'dir. Yani, $\xi_j > 0$ ve $\beta_j = C$ koşulunu sağlayan veri örnekleri hiper kürenin dışında kalan aykırı örneklerdir.

$\xi_j = 0$ ve $0 < \beta_j < C$ şartlarını sağlayan veri örnekleri ise hiper kürenin yüzeyindedir. Bu veri örneklerine destek vektörler (support vectors) denir ve giriş uzayındaki öbek sınırları, bu destek vektörler ile belirlenir.

Eşitlik 3, 4 ve 5 denklem 2'de yerine konup gerekli sadeleştirmeler yapılırsa (giriş değişkenleri yok edilirse) Lagrange çarpanları cinsinden ikinci dereceden bir optimizasyon denklemi (8) elde edilir.

$$\begin{gathered} W = \sum_j \Phi(x_j)^2\beta_j - \sum_{i,j}\beta_i\beta_j K(x_i, x_j) \\ subject\ to \\ 0 \le \beta_i \le C, \quad i = 1, \dots, N \end{gathered} \tag{8}$$

Burada,

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{9}$$

kernel olarak adlandırılır.

Literatürde genellikle Gauss kernel fonksiyonu kullanılır (10).

$$\begin{gathered} K(x_i, x_j) = e^{-q\|x_i - x_j\|^2} \\ q = \frac{1}{2\sigma^2} \end{gathered} \tag{10}$$

Denklem 8'in çözümünden Lagrange çarpanlarının optimal değeri bulunur. İkinci aşama öbek etiketleme adımıdır. DVÖ algoritmasında kullanılan geometrik yaklaşıma göre farklı iki öbekte yer alan iki örnek veriyi giriş uzayında birleştiren bir hat, özellik uzayında hiperkürenin dışında olmalıdır. Bu adımın uygulanması için, rastgele seçilen iki noktayı birleştirecek hattı oluşturan noktalar belirlenir. Noktalar arası mesafe fonksiyonunu bulmak için, denklem 11'de gösterilen, özellik uzayındaki bir verinin kürenin merkezine olan uzaklığını belirten denklem kullanılır.

$$R^2 = \left\|\Phi(x_j) - a\right\|^2 \tag{11}$$

Denklem 11'de a parametresi yerine, denklem 4'de gösterilen a parametresinin eşiti yazılırsa DVÖ için eğitilmiş kernel destek fonksiyonu aşağıdaki gibi (denklem 12) elde edilir.

$$\begin{gathered} f(x) = K(x, x) - 2\sum_j \beta_j K(x_j, x) + \\ \sum_{i,j}\beta_i\beta_j K(x_i, x_j) \end{gathered} \tag{12}$$

Denklem 10'da gösterilen Gauss kernel fonksiyonunun q parametresi, veri uzayından özellik uzayına atanan verilerin ölçeğini belirlemektedir. Bu parametrenin seçilmesi için [10]'da, denklem 13'de gösterilen formüldeki değerin ilk değer olarak seçilmesi ve bu değerin artırılarak devam edilmesi önerilmiştir. İlk q değerinde çok sayıda öbek oluşacakken, q değeri arttırıldığında öbek sayısı giderek azalacaktır.

$$q = \frac{1}{\max i,j \left\| x_i - x_j \right\|^2} \quad (13)$$

2.2. Elipsoit Destek Vektör Öbekleme

DVÖ algoritmasına benzer olarak EDVÖ algoritmasında da veriler önce yüksek boyutlu bir uzaya taşınır, taşınan veri örnekleri arasında benzerlikler hesaplanarak öbek sınırları oluşturulur. Diğer bir ifadeyle, optimizasyon ve öbek etiketlerinin belirlenmesi gibi adımlar EDVÖ algoritmasında da izlenmektedir. Bundan dolayı, DVÖ algoritmasına benzer bir yaklaşım EDVÖ algoritmasında da söz konusudur. Özellik uzayındaki verileri ifade etmek için ortalama vektörü (μ^{Φ}) ve kovaryans matrisi (Σ^{Φ}) kullanılır. Bu durum denklem 14'de ifade edilmiştir.

$$x_i \mapsto \Phi(x_i)\left(\mu^{\Phi}, \Sigma^{\Phi}\right), i = 1, \dots, N, \quad (14)$$

EDVÖ'de DVÖ'den farklı olarak özellik uzayına atanmış veriler hiperküre yerine denklem 15'de belirtilen hiperelipsoitin içine alınmasıdır. Bunun anlamı, özellik uzayına atanmış veri örneklerinin, birbirine olan benzerliklerini bulurken Öklid benzerlik metriği yerine mahalanobis benzerlik metriği kullanılacak olmasıdır. Mahalanobis benzerlik metriği, veri örneklerinin arasındaki benzerliği eliptik bir düzlemde bulur.

$$(\Phi(x) - a)^T (\Sigma^{\Phi})^{-1} (\Phi(x) - a) = r^2 \quad (15)$$

Burada, a elipsoitin merkezini, $(\Sigma^{\Phi})^{-1}$ kovaryans matrisinin tersini ifade eder.
EDVÖ yönteminin optimizasyon problemi denklem 16'da gösterildiği şekilde tanımlanır [16].

$$\begin{gathered} \min \{r + C \sum_{i=1}^{N} \xi_i\} \\ subject\ to \\ \sqrt{(\Phi(x_i) - a)^T \Sigma^{\Phi^{-1}} (\Phi(x_i) - a)} \leq r + \xi_i \\ r > 0, \xi_i \geq 0, i = 1, \dots, N \end{gathered} \quad (16)$$

Denklem 16'da ifade edilen optimizasyon probleminde kernel fonksiyonunun kullanabilmesi için verileri giriş uzayından yüksek boyutlu uzaya taşıyan doğrusal olmayan fonksiyonların iç çarpım olarak ifade edilmesi gerekmektedir. Konunun geri kalan kısmı denklem 9'da verilen bu iç çarpımı gerçekleştirmek için yapılan teorik işlemleri açıklamaktadır.

Teorem 1
Φ(x), özellik uzayına atanmış örnek olmak üzere μ^{Φ} ve Σ^{Φ} denklem 17'deki şekilde ifade edilebilir.

$$\begin{gathered} \mu^{\Phi} = \frac{1}{N} \sum_{i=1}^{N} \Phi(x_i) \\ \Sigma^{\Phi} = \frac{1}{N} \sum_{i=1}^{N} (\Phi(x_i) - \mu^{\Phi})(\Phi(x_i) - \mu^{\Phi})^T \end{gathered} \quad (17)$$

Burada, optimal a parametresi, özellik uzayına atanmış verilerin lineer bileşeni şeklinde yazılabilir. Bu durum denklem 18'de ifade edilmiştir. Özellik uzayındaki verilerden oluşan matrix X^{Φ} ile gösterilmektedir.

$$a = \sum_{i=1}^{N} w_i \Phi(x_i) = X^{\Phi} w \quad (18)$$

Optimizasyon probleminde kernel fonksiyonunu kullanabilmek için merkezileştirilmiş kernel matrisi, denklem 19'daki gibi tanımlanır.

$$K_C = K - EK - KE + EKE \quad (19)$$

Burada, E tüm değerleri 1/N olan [NxN] boyutlu matristir. K kernel matrisidir.

Teorem 2
K_C'nin özdeğer analizi $K_C = A^T \Omega A$ olarak kabul edildiğinde kovaryans matrisi Σ^{Φ} denklem 20'deki gibi yazılabilir.

$$\Sigma^{\Phi} = (\Omega^{-\frac{1}{2}} A X^{\Phi^T})^T \left(\frac{1}{N}\right) (\Omega^{-\frac{1}{2}} A X^{\Phi^T}) \quad (20)$$

Teorem 1 ve Teorem 2 nin ispatları [16]'da açıklanmıştır. Teorem 2'den kovaryans matrisinin sözde-ters formu Σ^{Φ^+}, denklem 21'deki gibi hesaplanır.

$$\Sigma^{\Phi^+} = N X^{\Phi} A^T \Omega^{-2} A X^{\Phi^T} \quad (21)$$

Denklem 18 ve denklem 21'i denklem 16'nın içinde kullandığımızda, denklem 22 ile gösterilen kısıtlı optimizasyon problemi oluşturulur.

$$\begin{gathered} \min \ \ r + C \sum_{i=1}^{N} \xi_i \\ s.t. \quad \left\| \sqrt{N} \Omega^{-1} A (K^i - Kw) \right\| \leq r + \xi_i, \\ r > 0, \quad \xi_i \geq 0, \quad i = 1, \dots, N \end{gathered} \quad (22)$$

Denklem 22'nin çözümünden eğitilmiş kernel destek fonksiyonu, aşağıdaki gibi (denklem 23) elde edilir.

$$f(x) = \left\| \sqrt{N} \Omega^{-1} A (K^x - Kw) \right\| \quad (23)$$

Yukarıda da bahsedildiği gibi, optimizasyon problemi, özellik uzayına atanmış verilerin iç çarpımları kullanılarak ifade edilmiştir.

Optimizasyon denklemindeki doğrusal olmayan kısıtlamalar ikinci dereceden koni (İDK) kısıtlamaları olarak adlandırılır.

Doğrusal bir denklemi doğrusal ve doğrusal olmayan kısıtlarla birlikte minimize etmeye ikinci dereceden koni programlama (İDKP) denir.

İDKP iç nokta yöntemi (interior-point method) kullanan Sedumi [18] optimizasyon algoritması ile MATLAB ortamında etkili bir şekilde çözülebilmektedir. Ara-yüz programı olarak CVX [19] ile birlikte Sedumi optimizasyon algoritması, sayısal işlemlerde önemli ölçüde kolaylık sağlamaktadır.

Bu çalışmada, Sedumi optimizasyon programı MATLAB ortamında CVX arayüz programı ile EDVÖ algoritmasının optimizasyon probleminin çözülmesi için kullanılmıştır. Ayrıca, k-merkez, hiyerarşik ve FCM öbekleme algoritmalarının çözümü için MATLAB ortamında bulunan hazır fonksiyonlardan yararlanılmıştır.

3. BULGULAR

Bu bölümde, giriş kısmında bahsedilen k-merkez, hiyerarşik ve FCM gibi geleneksel öbekleme algoritmaları ile önerilen EDVÖ algoritması hem sentetik yüzük veri setine hem de UCI veri seti deposundan [17] alınan gerçek hepatit ve parkinson biyomedikal veri setlerine uygulanmıştır. Ayrıca, elde edilen sonuçlardan konfüzyon matrisi çıkarılmış, öbek hata oranı, özgüllük, hassasiyet değerlerine göre söz konusu algoritmalar karşılaştırılmıştır.
EDVÖ ve diğer öbekleme algoritmalarının (k-merkez, hiyerarşik, FCM) performansları değerlendirilirken konfüzyon matrisi aşağıdaki formatta oluşturulmuştur.

Konfüzyon Matrisi	Pozitif işaretlenmiş	Negatif işaretlenmiş
Pozitif	TP	FN
Negatif	FP	TN

Hepatit veri seti için pozitif sınıf kişinin hepatit hastalığı taşıdığını (ölü), negatif sınıf ise kişinin sağlıklı olduğunu (hayatta), Parkinson veri seti için pozitif sınıf kişinin Parkinson hastalığı taşıdığını negatif sınıf ise kişinin sağlıklı olduğunu ifade etmektedir. TP hasta veri örneğinin hasta olarak işaretlendiği, FN hasta veri örneğinin sağlıklı olarak işaretlendiği, TN sağlıklı veri örneğinin sağlıklı olarak işaretlendiği, FP sağlıklı veri örneğinin hasta olarak işaretlendiği veri örneklerinin sayısını belirtmektedir.

Öbekleme algoritmalarının karşılaştırılmasında kullanılan hesaplamalar denklem 24'de belirtilmiştir.

Hassasiyet parametresi, hastalara ait veri örneklerinin ne kadarının hasta olarak işaretlendiğinin, özgüllük parametresi ise sağlıklı insanlara ait veri örneklerinin ne kadarının sağlıklı olarak işaretlendiğinin oranını gösterir. Hata oranı parametresi, toplamda hatalı işaretlenen veri örneklerinin tüm veri örneklerine göre yüzdesini ifade eder.

$$\text{Özgüllük} = \frac{TN}{TN + FP}$$

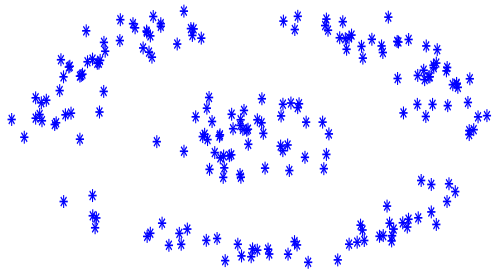
$$Hassasiyet = \frac{TP}{TP + FN} \quad (24)$$

$$Hata\ Oranı = \frac{TP + TN}{TP + FP + TN + FN} * 100$$

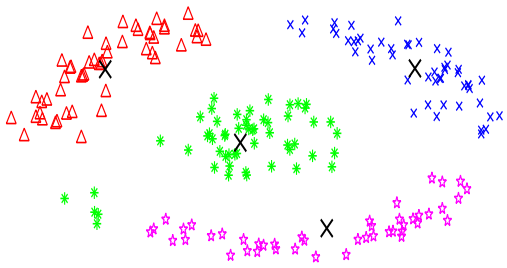
3.1. Sentetik Veri Seti

Yüzük veri seti, 2 boyutlu 200 örnek içerir. Öbekleme işlemini görselleştirmek için uygun olan bu veri seti, Şekil 1'deki gibi bir gösterime sahiptir. Hiyerarşik, k-merkez, FCM ve EDVÖ öbekleme algoritmaları ayrı ayrı yüzük veri setine uygulanmıştır. Buna göre en iyi sonuç, veri setinin 4 öbeğe ayrılması ile elde edilmiş ve görsel sonuçlar sırasıyla Şekil 2, 3, 4 ve 5'de gösterilmiştir. Ayrıntılı analiz sonuçları ise Tablo 1'de verilmiştir. Yüzük veri setini öbeklere ayırmak için kullanılan k-merkez, FCM ve hiyerarşik öbekleme algoritmalarını çalıştırmadan önce öbek sayısını belirten parametrenin (k) değeri belirtilmelidir. Örneğimizde, bu parametre 2, 3 ve 4 olacak şekilde seçilmiş ve her bir durum için öbekleme hata oranları Tablo 1'de sunulmuştur. Ancak, kernel tabanlı EDVÖ algoritması daha önce de bahsedildiği gibi önceden öbek sayısını belirten k parametre değerine ihtiyaç duymadığından, kernel parametresinin değeriyle veri setine uygun sayıda öbek oluşturmuştur. Kernel olarak Gauss kernel kullanılmış olup, k-merkez, hiyerarşik ve FCM için belirlenen öbek sayılarını sağlayacak şekilde varyans değeri 2 ile 3 arasındaki değerlerden seçilmiştir.

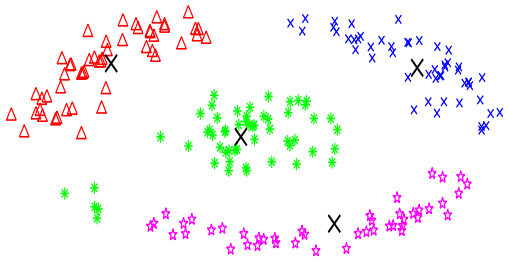
Tablo 1'den görüleceği üzere, k=4 alındığında, FCM ve k-merkez yöntemleri ile %2.5 hata elde edilmiştir. Şekil 2 ve Şekil 3'den görüleceği üzere, k-merkez ve FCM algoritmaları diğer bir öbeğe ait olan verileri merkezdeki öbeğe atamıştır. Bu durum verinin düzgün bir şekilde öbeklenemediği anlamına gelir. Diğer taraftan, hiyerarşik öbekleme algoritmasında, k=4 alındığında yüzük veri setine ait örnekleri hatasız bir şekilde öbekleyebildiği Şekil 4'den görülmektedir. Tablo 1'den ve Şekil 5'den görülebileceği üzere önerilen EDVÖ algoritmasında varyans değeri σ=2.4 alındığında % 0.5 hata ile yüzük veri setini 4 öbeğe ayırabilmiştir. Ancak, EDVÖ algoritması ile verinin dağılımına uygun olarak oluşturulan öbeklerin sınırları diğer algoritmalara göre daha belirgindir. Bu durum Şekil 5'de açıkça görülmektedir.



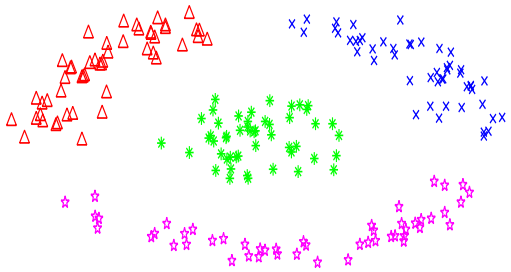
Şekil 1.Yüzük Veri Seti



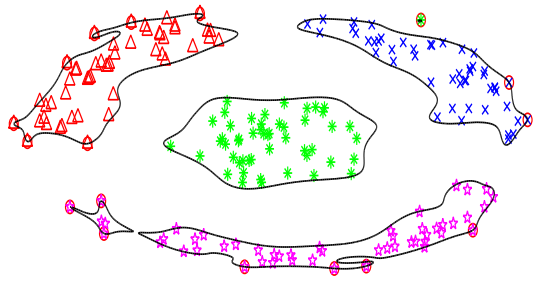
Şekil 2. k-merkez algoritması ile öbekleme (k=4)



Şekil 3. FCM algoritması ile öbekleme (k=4)



Şekil 4. Hiyerarşik algoritma ile öbekleme (k=4)



Şekil 5. EDVÖ algoritması ile öbekleme (σ=2.4)

Tablo 1. Sentetik yüzük veri setine ait öbekleme sonucu elde edilen hata oranları

Öbek Sayısı / Hata Oranı (%)	**Öbekleme Yöntemi**			
Cinsiyet	k-merkez	FCM	Hiyerarşik	*EDVÖ
K = 2 $*\sigma = 2.9$	50	50	50	50
K = 3 $*\sigma = 2.76$	25	25	25	25.5
K = 4 $*\sigma = 2.4$	2.5	2.5	0	0.5

3.2. Gerçek Biyomedikal Veri Seti

Gerçek dünya veri setleri çok boyutlu verilerden oluşmaktadır. Her bir boyutta gösterilen ölçümler farklı büyüklükteki değerler ile dolayısıyla farklı aralıklarda ifade edilmektedir. Örneğin, biyomedikal veri setlerinde, yaş özelliği 1-100 arasında değişebilen bir değere karşılık gelirken, kanda bulunan bazı maddelerin ölçüm değerleri 0-5 arası değerler ile ifade edilmektedir. Farklı aralıklarda olan verilerin sisteme doğrudan verilmesi optimizasyon aşamasında hataya neden olabilir. Bu sorundan kaçınmak için, veri öznitelik vektörleri üzerine ortalama, normalleştirme ve ölçekleme gibi işlemler uygulanarak veri örneklerinin benzer ölçekte (genellikle 0-1 aralığında) olması sağlanır. Ayrıca kategorik öznitelik vektörleri, her bir kategori için sadece o kategorik özelliğin 1, diğer kategorik özelliklerin 0'dan oluştuğu ayrı ayrı öznitelik vektörleri oluşturularak sisteme verilmelidir. Bu çalışmada, veri setlerinin hazırlanmasında söz konusu bu işlemlerin tamamı uygulanmıştır.

Hepatit veri seti, hepatit virüsüne sahip kişiler ile sağlıklı kişilerden oluşmaktadır. Dolayısıyla, veri seti sadece 2 öbek içerir. Bu durum ikili (binary) elemanlara sahip bir öznitelik vektörü ile ifade edilmektedir. Bundan dolayı, k-merkez ve FCM algoritmalarında önceden sisteme verilmesi gereken küme sayısı 2 olarak belirlenmiştir. EDVÖ algoritmasında ise, sisteme önceden herhangi bir küme sayısı girilmediğinden, öbek sayısı iki olacak şekilde kernel parametresi belirlenmiş ve optimizasyon algoritması çalıştırılmıştır. Ayrıca, öbekleme algoritmaları, sınıflandırma algoritmalarının aksine, denetimsiz yaklaşımlar olduğundan bu öznitelik vektörü başta sisteme verilmemektedir.

Hepatit veri seti, 155 adet 19 boyutlu örnek içerir. Burada, 155 rakamı 155 farklı kişiden alınan verilerin kullanıldığını, 19 rakamı ise her bir kişiden alınan verinin 19 farklı ölçüme sahip olduğunu belirtir. Veri seti içerisinde ikili değere sahip 13 öznitelik değeri bulunurken, değer aralığı 6 ile 8 arasında olan 6 ayrık (discrete) sayısal değerli öznitelik vektörü bulunur. Hepatit veri setinde bulunan her bir özniteliğe ait bilgiler Tablo 2'de ayrıntıları ile gösterilmektedir.

Tablo 2. Hepatit veri seti öznitelikleri tablosu

Öznitelik İsmi	Öznitelik Değeri
Yaş	10, 20, 30, 40, 50, 60, 70, 80
Cinsiyet	Erkek, Kadın
Steroid	Hayır, Evet
Antivirals	Hayır, Evet
Fatigue	Hayır, Evet
Malaise	Hayır, Evet
Anorexia	Hayır, Evet
Liver big	Hayır, Evet
Liver firm	Hayır, Evet
Spleen Palpable	Hayır, Evet
Spiders	Hayır, Evet
Ascites	Hayır, Evet
Varices	Hayır, Evet
Bilirubin	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
Alk Phosphate	33, 80, 120, 160, 200, 250
Sgot	13, 100, 200, 300, 400, 500
Albumin	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
Protime	10, 20, 30, 40, 50, 60, 70, 80, 90
Histology	Hayır,Evet

Önerilen EDVÖ algoritması ile literatürde yaygın olarak kullanılan k-merkez, hiyerarşik ve FCM algoritmalarının hepatit veri seti kullanılarak yapılan öbekleme sonuçlarına ait konfüzyon matrisleri sırasıyla Tablo 3, 4, 5 ve 6'da verilmiştir.

Tablo 3. Hepatit veri seti ile k-means kullanılarak yapılan öbekleme sonucu elde edilen konfüzyon matrisi

k-merkez	Pozitif işaretlenmiş	Negatif işaretlenmiş
Pozitif	5	14
Negatif	21	72

Tablo 4. Hepatit veri seti ile FCM kullanılarak yapılan öbekleme sonucu elde edilen konfüzyon matrisi

FCM	Pozitif işaretlenmiş	Negatif işaretlenmiş
Pozitif	6	13
Negatif	21	72

Tablo 5. Hepatit veri seti ile hiyerarşik algoritması kullanılarak yapılan öbekleme sonucu elde edilen konfüzyon matrisi

Hiyerarşik	Pozitif işaretlenmiş	Negatif işaretlenmiş
Pozitif	5	14
Negatif	21	72

Tablo 6. Hepatit veri seti ile EDVÖ kullanılarak yapılan öbekleme sonucu elde edilen konfüzyon matrisi

EDVÖ	Pozitif işaretlenmiş	Negatif işaretlenmiş
Pozitif	6	13
Negatif	9	84

Hepatit veri setinde uygulanan öbekleme algoritmalarından elde edilen konfüzyon matrislerine göre en iyi TN ve TP değerleri sırasıyla EDVÖ algoritması ile 84 ve 6 olarak elde edilmiştir (Tablo 6). Çalışmada kullanılan bir diğer veri seti ise parkinson hastalığına ait veri setidir. Oxford üniversitesinden Max Little tarafından, Kolorado eyâletinin başkenti Denver'de bulunan hastaların, konuşma sinyallerinin ulusal ses ve konuşma merkezinde kaydedilmesiyle oluşturulmuştur.

Tablo 7. Parkinson veri seti öznitelikleri tablosu

Öznitelik İsmi	Öznitelik Açıklaması
MDVP:Fo(Hz)	Ortalama Ses Frekansı
MDVP:Fhi(Hz)	En Yüksek Ses Frekansı
MDVP:Flo(Hz)	En Düşük Ses Frekansı
MDVP:Jitter(%), MDVP:Jitter(Abs), MDVP:RAP, MDVP:PPQ, Jitter:DDP	Frekans Değişimini Gösteren Bazı Ölçümler
MDVP:Shimmer, MDVP:Shimmer(dB), Shimmer:APQ3, Shimmer:APQ5, MDVP:APQ, Shimmer:DDA	Genlik Değişimini Gösteren Bazı Ölçümler
NHR,HNR	Ses tonuna ait bileşenler ile gürültü oranını belirten iki adet ölçüm
RPDE,D2	İki adet lineer olmayan karmaşıklık ölçüleri
DFA	Sinyal fraktal ölçeklendirme üssü
spread1, spread2, PPE	Temel frekans değişiminin üç doğrusal olmayan ölçümü

Söz konusu veri seti, Parkinson hastalığına sahip kişiler ile sağlıklı kişilerden alınan bir dizi biyomedikal ses ölçümü

içerir. Bundan dolayı, öbekleme sonucu ulaşılması gereken öbek sayısı 2'dir. Parkinson veri seti 197 adet 22 boyutlu örnek içerir. Her bir öznitelik belirli bir ses ölçüsüdür ve her örnek 197 adet ses kaydına karşılık gelir. Tablo 7'de Parkinson veri setine ait öznitelikler ayrıntıları ile verilmiştir. Önerilen EDVÖ algoritması ile k-merkez, hiyerarşik ve FCM algoritmalarının Parkinson veri seti kullanılarak yapılan öbekleme sonuçlarına ait konfüzyon matrisleri sırasıyla Tablo 8, 9, 10 ve 11'de verilmiştir.

Tablo 8. Parkinson veri seti ile k-merkez kullanılarak yapılan öbekleme sonucu elde edilen konfüzyon matrisi

k-merkez	Pozitif işaretlenmiş	Negatif işaretlenmiş
Pozitif	74	73
Negatif	0	48

Tablo 9. Parkinson veri seti ile FCM kullanılarak yapılan öbekleme sonucu elde edilen konfüzyon matrisi

FCM	Pozitif işaretlenmiş	Negatif işaretlenmiş
Pozitif	84	63
Negatif	1	47

Tablo 10. Parkinson veri seti ile hiyerarşik kullanılarak yapılan öbekleme sonucu elde edilen konfüzyon matrisi

Hiyerarşik	Pozitif işaretlenmiş	Negatif işaretlenmiş
Pozitif	103	44
Negatif	48	0

Tablo 11. Parkinson veri seti ile EDVÖ kullanılarak yapılan öbekleme sonucu elde edilen konfüzyon matrisi

EDVÖ	Pozitif işaretlenmiş	Negatif işaretlenmiş
Pozitif	144	3
Negatif	38	10

Parkinson veri setinde uygulanan öbekleme algoritmalarından elde edilen konfüzyon matrislerine göre en iyi TP değeri 144 ile EDVÖ algoritmasından elde edilirken en iyi TN değeri ise 48 ile k-merkez algoritmasından elde edilmiştir. Tablo 12 ve 13'de EDVÖ algoritmasının uygulanmasından elden edilen hassasiyet, özgüllük ve hata oranı ile literatürde kullanılan geleneksel diğer öbekleme yöntemlerinin (k-merkez, hiyerarşik ve FCM) uygulanmasıyla elde edilen sonuçlar karşılaştırmalı olarak gösterilmektedir.

Tablo 12. Hepatit veri seti ile yapılan deneyler sonucu performans karşılaştırması

Hepatit	k-merkez	FCM	Hiyerarşik	EDVÖ
Hassasiyet	0.26	0.32	0.26	**0.32**
Özgüllük	0.77	0.77	0.77	**0.90**
Hata Oranı(%)	31.25	30.35	31.25	**19.6**

Tablo 12'den görüleceği üzere, hepatit veri seti üzerinde en küçük hata oranı 19.6% en yüksek hassasiyet (0.32) ve özgüllük (0.9) değerleriyle EDVÖ algoritmasından elde edilmiştir.

Tablo 13. Parkinson veri seti ile yapılan deneyler sonucu performans karşılaştırması

Parkinson	k-merkez	FCM	Hiyerarşik	EDVÖ
Hassasiyet	0.5	0.57	0.7	**0.98**
Özgüllük	**1**	0.98	0	0.21
Hata Oranı(%)	37.43	32.82	47.18	**21.02**

Benzer şekilde Tablo 13'den görüleceği üzere, Parkinson veri setinde en küçük hata oranı 21.2% ve en yüksek hassasiyet 0.98 ile EDVÖ algoritmasından elde edilirken en yüksek özgüllük ise 1 ile k-merkez algoritmasından elde edilmiştir.

Bu sonuçlara göre önerilen EDVÖ algoritmasının gerçek biyomedikal veri setleri üzerindeki performansının diğer üç algoritmaya göre daha iyi olduğu anlaşılmaktadır.

4. DEĞERLENDİRME VE SONUÇ

Bu çalışmada, kernel tabanlı EDVÖ algoritmasının sentetik ve gerçek dünya veri setleri ile performans analizi yapılmıştır. EDVÖ, denetimsiz yaklaşım özelliğinden dolayı, sınıflandırma algoritmalarının aksine küme etiketlerinin bilinmesine gerek duymaması yönüyle oldukça dikkat çekmektedir. Ayrıca, kernel fonksiyonu parametresi sayesinde, algoritmanın çalıştırılabilmesi için önceden öbek sayısının belirtilmesine ihtiyaç duymaması ve farklı varyanslara sahip veri setlerine de başarı ile uygulanabilmesi için veriler arasında benzerlik metriği olarak mahalanobis uzaklığını kullanması gibi özellikleri ile son yıllarda diğer kümeleme algoritmalarından daha çok kullanım alanı bulmuştur. Hasta ve sağlıklı kişilerden alınarak oluşturulan gerçek hepatit ve Parkinson veri setleri kullanılarak gerçekleştirilen öbekleme işlemlerine yönelik, literatürde, k-merkez, FCM, hiyerarşik gibi çeşitli öbekleme algoritmaları önerilmesine karşın EDVÖ algoritması ile yapılmış bir çalışmaya rastlanmamıştır. Çalışmamızda, EDVÖ yöntemi, sentetik yüzük veri seti ile hepatit ve parkinson veri setleri üzerinde uygulanmış ve yukarıda bahsi geçen yöntemlerle öbekleme hata oranı cinsinden karşılaştırılmıştır.

Karşılaştırma sonuçlarına göre Yüzük veri setinde en iyi sonuçlar k=4 alındığında elde edilmiştir. Burada k-merkez ve FCM algoritmaları %2.5, hiyerarşik öbekleme algoritması %0 hata ile veriyi öbeklerken EDVÖ algoritması %0.5 hata ile veriyi öbeklemiştir. Hepatit veri seti üzerinde, önerilen EDVÖ algoritması ile %19.6 'lık en küçük hata oranı elde edilirken, benzer şekilde parkinson veri seti üzerinde de önerilen EDVÖ algoritması ile %21.02'lik en küçük hata oranı elde edilmiştir. Diğer bir ifadeyle, EDVÖ algoritması

ile hepatit veri seti %80.04 ve Parkinson veri seti %79.98 başarı oranı ile öbekleyebilmiştir. Ayrıca EDVÖ algoritması özgüllük açısından da diğer algoritmalardan daha başarılı sonuçlar vermiştir. Hassasiyet açısından EDVÖ algoritması Hepatit veri setinde en iyi değeri sağlarken Parkinson veri setinde en iyi değer k-merkez algoritması ile elde edilmiştir. Sonuç olarak önerilen EDVÖ algoritması, gelecek çalışmalar için diğer biyomedikal uygulamalarda da kullanılabilir.

KAYNAKÇA

[1].Fahad, Adil, et al. "A survey of clustering algorithms for big data: Taxonomy and empirical analysis", IEEE transactions on emerging topics in computing, 2(3):267-279, (2014).

[2].Starczewski, Artur. "A new validity index for crisp clusters", Pattern Analysis and Applications, 20(3):687-700, 2017.

[3].Xu, Rui, and Donald Wunsch, "Survey of clustering algorithms", IEEE Transactions on neural networks, 16(3):645-678, (2005).

[4].Khanmohammadi, Sina, Naiier Adibeig, and Samaneh Shanehbandy, "An improved overlapping k-means clustering method for medical applications", Expert Systems with Applications, 67:12-18, (2017).

[5].Rosati, Samanta, et al, "Muscle activation patterns during gait: A hierarchical clustering analysis", Biomedical Signal Processing and Control, 31:463-469, (2017).

[6].Kumar, Dhirendra, et al, "A modified intuitionistic fuzzy c-means clustering approach to segment human brain MRI image", Multimedia Tools and Applications, 1-25, (2018).

[7].Velmurugan, T, "Performance based analysis between k-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data", Applied Soft Computing, 19:134-146, (2014).

[8].Erken, Mücahit, "Comparing clustering algorithms on wisconsin data set", 24th. IEEE Signal Processing and Communication Application Conference (SIU), 2016.

[9].Sharma, Alok, et al, "Hierarchical maximum likelihood clustering approach", IEEE transactions on biomedical engineering, 64(1):112-122, (2017).

[10]. Ben-Hur, Asa, et al, "Support vector clustering", Journal of machine learning research, 125-137, (2001).

[11]. Dongiovanni, Danilo & Vaina, Lucia, "Select and Cluster: A Method for Finding Functional Networks of Clustered Voxels in fMRI", Computational Intelligence and Neuroscience, (7):1-19, (2016).

[12]. Yin, Zhong, and Jianhua Zhang, "Identification of temporal variations in mental workload using locally-linear-embedding-based EEG feature reduction and support-vector-machine-based clustering and classification techniques", Computer methods and programs in biomedicine, 115(3): 119-134, (2014).

[13]. Villazana, Sergio, César Seijas, and Antonino Caralli, "Lempel–Ziv complexity and Shannon entropy-based support vector clustering of ECG signals", 22(1):(2015).

[14]. Yin, Zhong, and Jianhua Zhang, "Identifying changes in human operator mental workload by locally linear embedding and support vector clustering approaches", IFAC Proceedings Volumes 46(13):353-358, 2013.

[15]. Lım, B., Lı, X., Zhang, J. B., & Shı, D, Feature Dıscretızatıon By Support Vector Clusterıng, "In Information Sciences", 797-803, (2007).

[16]. Wang, Defeng, et al. "Ellipsoidal support vector clustering for functional MRI analysis", Pattern Recognition, 40(10):2685-2695, (2007).

[17]. UCI Machine Learning Repository, URL: http://archive.ics.uci.edu/ml (Erişim Zamanı; Mart, 2018)

[18]. J. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, Optim. Methods Software", 11:625–653, (1999).

[19]. CVX, Matlab software for disciplined convex programming, URL: http://cvxr.com/cvx (Erişim Zamanı; Mart, 2018).

[20]. Kim, Dae-Won, et al. "A k-populations algorithm for clustering categorical data", Pattern Recognition 38(7): 1131-1134, (2005).