Politeknik Dergisi
Cilt: 8 Sayı: 2 s. 139-145, 2005

Journal of Polytechnic
Vol: 8 No: 2 pp. 139-145, 2005

DBSCAN, OPTICS ve K-Means Kümeleme Algoritmalarının Uygulamalı Karşılaştırılması

Turgay Tugay BİLGİN[*] , Yılmaz ÇAMURCU[**]
[*]Maltepe Üniversitesi Mühendislik-Mimarlık Fakültesi, Başıbüyük Kampüsü
Maltepe, İSTANBUL
[**]Marmara Üniversitesi Teknik Eğitim Fakültesi, Göztepe Kampüsü,
Kadıköy, İSTANBUL

ÖZET

Bu çalışmada, veri madenciliğinde güncel kümeleme algoritmalarından DBSCAN, OPTICS ile geçmişi daha eskilere dayanan K-means algoritması karşılaştırılmıştır. Karşılaştırma sentetik veritabanı üzerinde gösterdikleri küme bulma performansları değerlendirilerek yapılmıştır. Sonuçta, yakın zamanda literatüre giren DBSCAN ve OPTICS algoritmalarının K-means algoritmasından daha üstün küme oluşturma özelliklerine sahip olduğu tespit edilmiştir.

Anahtar Kelimeler : Veri madenciliği, kümeleme analizi, DBSCAN, OPTICS, K-means

Applied Comparison of DBSCAN, OPTICS and K-Means Clustering Algorithms

ABSTRACT

DBSCAN and OPTICS are two recent clustering algorithms on data mining. In this study, these two algorithms and K-means which is one of the oldest clustering algorithms are compared. Comparison is based on cluster discovery performance on synthetic database. Consequently, two recent clustering algorithms DBSCAN and OPTICS are performed superior accuracy and cluster discovery ability over K-means algorithm.

Keywords : Data mining, Clustering Analysis, DBSCAN, OPTICS,K-means

1. GİRİŞ

Veri madenciliği (VM) yapay zeka, bilgisayar bilimleri, makine öğrenimi, veritabanı yönetimi, veri gözlemleme, matematik algoritmalar ve istatistik gibi konuları içeren disiplinler arası bir alandır (1-3). Bu teknoloji karar verme, problem çözme, analiz, planlama, teşhis, bütünleştirme, koruma, öğrenme ve keşif için farklı yöntemler sağlar (1-8). VM, büyük veri tabanlarında örüntülerin, birlikteliklerin, anormalliklerin ve çeşitli yapıların yarı otomatik bir sistem ile keşfidir (1,3).

Kümeleme, verinin benzer nesnelerden oluşturulmuş gruplara bölünmesidir. Kümeleme işleminde küme içindeki elemanların benzerliği fazla, kümeler arası benzerlik ise az olmalıdır (1,9). Bir kümeleme yönteminin kalitesi bu prensibi sağlaması ile doğru orantılıdır. Kümeleme yöntemi seçimi kullanılacak veri türüne ve uygulamanın amacına göre farklılık gösterir.

Bu çalışmada yoğunluk tabanlı kümeleme algoritmalarından DBSCAN (Density Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points To Identify Clustering Structure) ve bölümlemeli kümeleme algoritması K-Means'in karşılaştırması yapılmıştır. Çalışmanın amacı, her algoritma için yazılım platformu hazırlayıp karşılaştırmaktır. Uygulaması yapılacak veri madenciliği algoritmalarının farklı veri dağılımlarında küme bulabilme kabiliyetlerini karşılaştırabilmek için sentetik veritabanı tercih edilmiştir. Uygulama geliştirme aşamasında MATLAB yazılımı kullanılmıştır.

DBSCAN algoritması, Ester, Kriegel, Sander ve Xu tarafından KDD'96 konferansında sunulmuştur (10). Bu algoritma, nesnelerin komşuları ile olan mesafelerini hesaplayarak belirli bir bölgede önceden belirlenmiş eşik değerden daha fazla nesne bulunan alanları gruplandırarak kümeleme işlemini gerçekleştirir. DBSCAN algoritması veri madenciliğine birçok yeni terim ve yaklaşım getirmiştir.

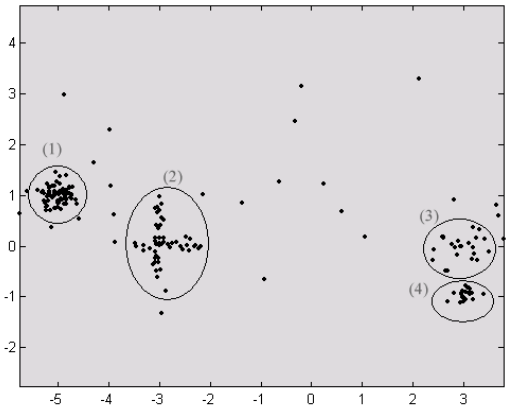
OPTICS algoritması, Ankerst, Breunig, Kriegel, ve Sander tarafından SIGMOD'99 konferansında sunulmuştur (11). DBSCAN algoritmasının geliştirilmiş hali olarak tanımlanabilir. DBSCAN algoritmasının zayıflığı olarak tanımlanabilen Eps ve MinPts değerlerine bağımlığı azaltmak için veri nesnelerini Eps değerine göre bir grafik üzerine yerleştirip MinPts değerine gerek kalmadan grafik üzerinden kümeleri bulmayı sağlar.

K-means bilimsel ve endüstriyel uygulamalarda yıllardır kullanılmaktadır (12-14). Algoritmanın adı, her biri Cj olmak üzere k adet kümenin cj ile ifade edilen ortalamalarının alınmasından gelir.

2. DENEYSEL ÇALIŞMALAR VE SONUÇLAR

2.1. Veri Setinin Tanıtımı

Çalışmada kullanılan sentetik veritabanı, düz metin dosyası formatındadır (15). Sentetik veritabanı gerçek veritabanlarında aynı anda rastlanması zor olan birkaç farklı tür kümelenme yapısını içermektedir. Sentetik veritabanının 2 boyutlu düzlemdeki görüntüsü Şekil 1'deki gibi soldan sağa toplam dört adet kümeden oluşmaktadır. Bu şekilden de görüleceği gibi, birincisi yoğun ve gürültü içeren küresel, ikincisi küresel olmayan ve gürültü içeren, üçüncüsü seyrek ve küresel, dördüncüsü yoğun ve gürültü içermeyen olmak üzere dört temel küme dağılımı vardır.



Şekil 1. Sentetik veritabanında kümelenme bölgeleri

2.2. K-means Algoritması

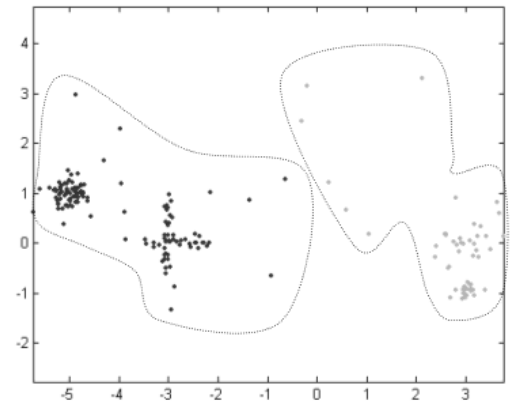
K-means algoritmasının detayları Alsabti, Ranka ve Singh makalesinde açıklanmıştır (16). K-means algoritmasında aranan küme sayısını ifade eden k önceden bilinen ve kümeleme işlemi bitene kadar değeri değişmeyen bir sabit olmalıdır (17). Kümeleme işlemi başlangıcında küme merkezlerini temsilen k adet rastgele nokta seçilir. Bu noktaların her biri prototip olarak adlandırılır. Kümeleme başlangıcında k adet prototip (w1,w2,w3 .. wk) ve her bir küme ya da örüntü (i1,i1,i3 .. in) olmak üzere

$$w_i = i_l,\ j \in \{1,...,k\}, l \in \{1,...,n\} \qquad (1)$$

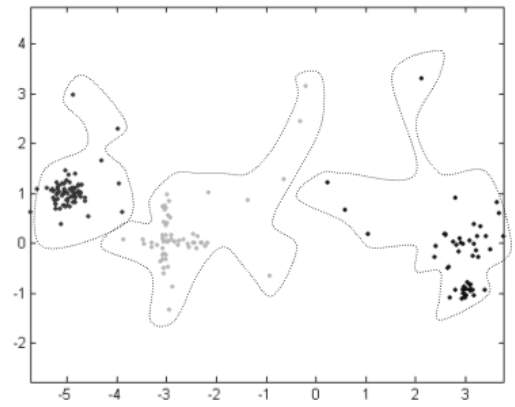
durumundadır. K-means algoritmasının matematiksel yorumlanışına ilişkin aşağıda verilen açıklamalarda, Cj ifadesi j. elemanı temsil etmek üzere, kümeleme işleminin kalitesi Denklem 2'deki hata fonksiyonu ile ifade edilir (9):

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} \left| i_l - w_j \right|^2 \qquad (2)$$

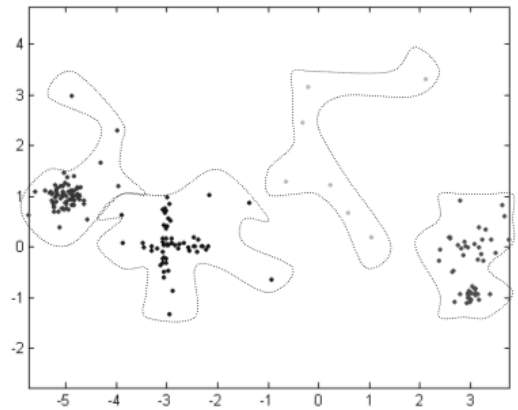
K-means algoritmasının en büyük problemi uygun *k* değerini tespit edememesidir. Bu yüzden, en uygun kümelenmeleri bulabilmek için farklı *k* değerleri ile birçok deneme yapmak gerekmektedir. Bu çalışmada kullanılan veri seti üzerinde farklı k değerleri ile yapılan denemeler Şekil 2 de görülmektedir.



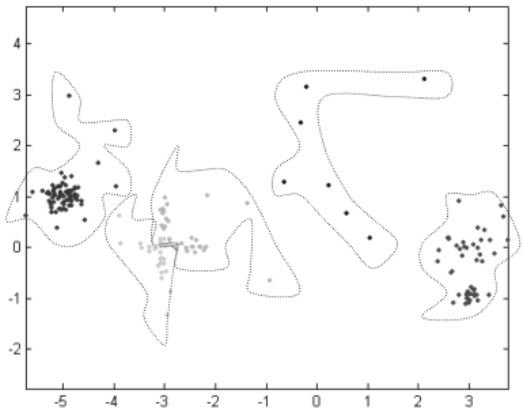
(a) k=2



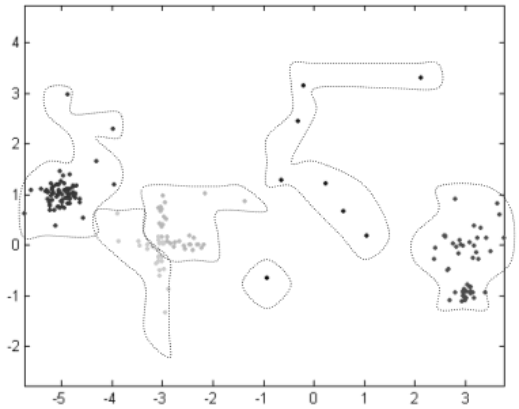
(b) k=3



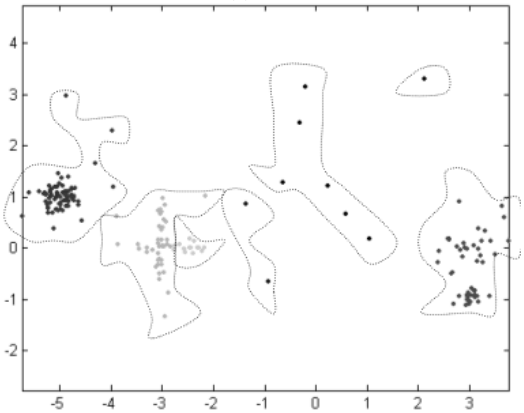
(c) k=4



(d) k=5



(e) k=6



(f) k=7

Şekil 2. K-means algoritmasında farklı k değerleri için elde edilen kümeler.

K-means algoritması *k* küme sayısını belirtmek üzere k=2,3,4,5,6,7 parametreleri ile 6 defa uygulanmıştır. Şekil 2.a'da görüldüğü gibi k=2 için 1 ve 2. kümeler ile 3 ve 4. kümeler birleşmiş durumda, idealden uzak bir kümelenme oluşturmuştur. Şekil 2.b'de k=3 için elde edilen sonuçta 1 ve 2. kümeler ayrılmış, fakat 3 ve 4. kümeler henüz birleşik durumdadır. Her iki şekilde de 2. ve 3. kümeler arasındaki gürültü noktalar da hatalı bir şekilde 2. ve 3. kümeler arasında paylaşılmıştır.

k=4 seçildiğinde(Şekil 2.c) 1. ve 2. kümeler doğru bir şekilde ayrılmışlar fakat 3 ve 4. kümeler ayrılmamış, 2 ile 3 arasındaki gürültü noktalar ile ayrı bir küme oluşmaktadır. Şekil 2.d'da görüldüğü gibi k=5 seçildiğinde 2 numaralı küme küresel olmadığı için parçalanmaya başlamıştır. K-means algoritmasının küresel olmayan kümelerdeki başarısızlığı k=6 seçildiğinde (Şekil 2.e) açıkça görülmektedir. Şekilde, 3. ve 4. kümenin ayrılması beklendiği halde küresel olmayan 2. kümenin kendi içinde parçalandığı görülüyor. Şekil 2.f'de görülen k=7 için elde edilen sonuçta da 3. ve 4. kümeler ayrılmamış, 2. ve 3. küme arasındaki gürültü noktalar daha fazla bölünerek idealden uzak kümelenmeler oluşmuştur. K-means algoritması hiçbir durumda Şekil 3'deki ideal kümelenmeyi elde edememiştir.

2.3. DBSCAN Algoritması

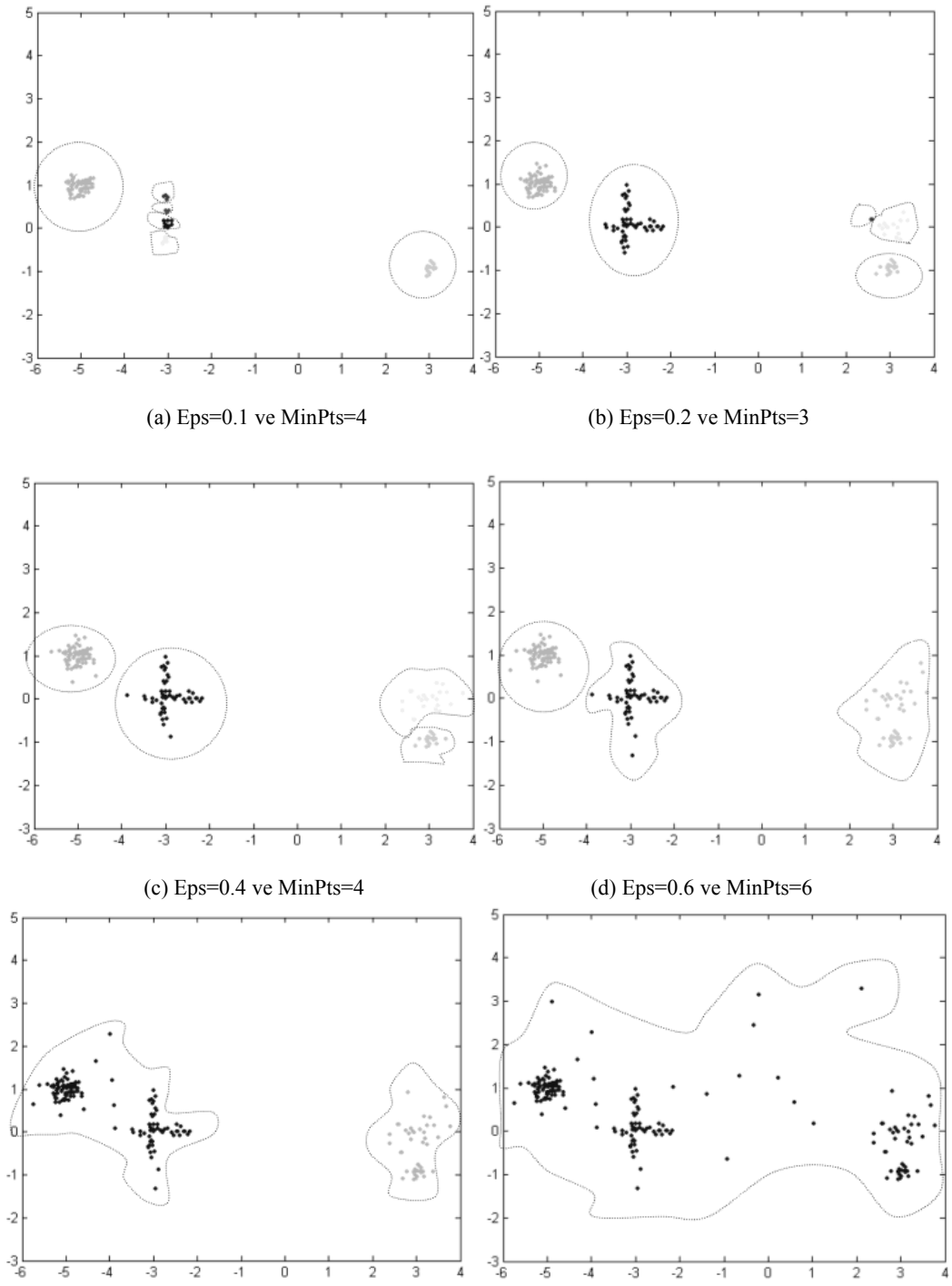
DBSCAN algoritması, veri noktalarının iki ya da çok boyutlu uzayda birbirleri ile olan komşuluklarını ortaya çıkarma temeline dayanır. Veritabanı, uzaysal bakış açısı ile ele aldığı için çoğunlukla uzaysal (spatial) verilerin analizinde kullanılmaktadır (10,18).

DBSCAN algoritması için çekirdek nesne, Eps, MinPts, doğrudan yoğunluk erişilebilir nokta, yoğunluk erişilebilir nokta, yoğunluk bağlı nokta terimleri temel kavramlardır. Algoritma, Eps ve MinPts değerlerini giriş parametresi olarak alır. Veritabanındaki herhangi bir nesneden başlayarak tüm nesneleri kontrol eder. Eğer kontrol edilen nesne daha önce bir kümeye dahil edilmiş ise işlem yapmadan diğer nesneye geçer. Eğer nesne daha önce kümelenmemiş ise, bir bölge sorgusu (Region Query) yaparak nesnenin Eps komşuluğundaki komşularını bulur. Komşu sayısı MinPts'den fazla ise, bu nesne ve komşularını yeni bir küme olarak adlandırır. Daha sonra, önceden kümelenmemiş her bir komşu için yeni bölge sorgusu yaparak yeni komşular bulur. Bölge sorgusu yapılan noktaların komşu sayıları MinPts'den fazla ise kümeye dahil eder.

Komşuluk bulma işlemi, DBSCAN algoritmasının en fazla işlem gücü gerektiren bölümüdür. Bu bölümde yapılacak performans iyileştirmeleri algoritmanın performansını önemli ölçüde arttırmaktadır (10). Komşuluk incelemesinde her noktayı incelemek yerine R*-tree (19) ya da uzaysal sorgulama (spatial query) (20) gibi çeşitli indeksleme algoritmaları ortaya atılmıştır. Bu algoritmalar ile DBSCAN algoritmasının **O(n*log n)** olan karmaşıklığını **O(log n)**'e düşürülerek önemli performans artışları sağlanabilmektedir .

DBSCAN algoritması Eps ve MinPts olmak üzere iki parametre aldığından her iki parametrenin de kümeleme sonucuna etkisini görebilmek amacıyla farklı parametreler ile 7 defa uygulanmıştır. DBSCAN algoritması K-means'den farklı olarak veritabanının her elemanını bir kümeye dahil etmez, istisna verileri süzme yeteneğine sahiptir. Sonuç grafiklerinde algoritmanın gürültü (istisna) olarak belirlediği değerler gösterilmemiştir.

Şekil 3.a'da görüldüğü gibi, Eps komşuluk mesafesine çok küçük değer verildiğinde yalnızca çok yoğun kümelenme alanları, diğer bir ifade ile küme çekirdekleri bulunmuştur. Eps değeri 0.2 olarak uygulandığında (Şekil 3.b) ideale çok yakın kümelenme oluşmasına rağmen 3. kümenin yakınında istenmeyen küçük bir küme daha oluşmuştur.



(a) Eps=0.1 ve MinPts=4 (b) Eps=0.2 ve MinPts=3

(c) Eps=0.4 ve MinPts=4 (d) Eps=0.6 ve MinPts=6

(e) Eps=0.8 ve MinPts=3 (f) Eps=3 ve MinPts=2

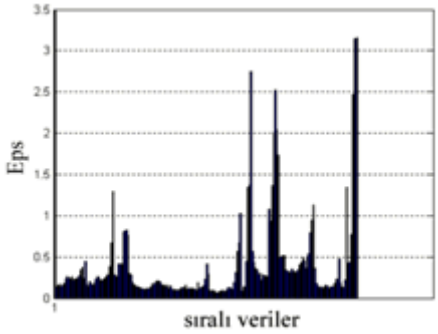
Şekil 3. DBSCAN algoritmasında farklı "k" değerleri için elde edilen kümeler.

Şekil 3.c'de, Eps=0.4 ve MinPts=4 parametreleri ile ideal sonuç elde edilmiştir. Şekilde gürültü noktalar küme alanlarından tamamen ayrılmış ve 1., 2., 3., ve 4. kümeler net olarak görülmektedir.

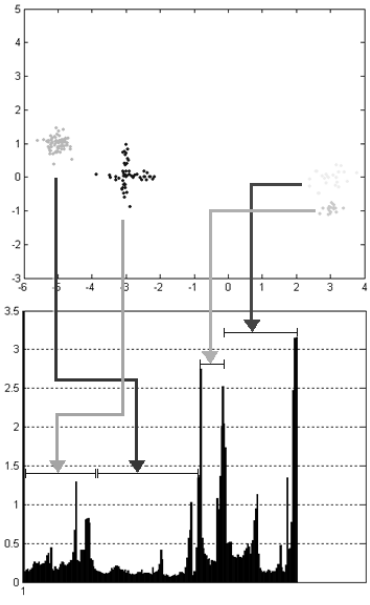
Şekil 3.d ve 3.e'de, Eps değeri artırıldığında önce 3. ve 4., daha sonra 1. ve 2. kümelerin birleştiği görülmektedir. Son olarak Şekil 3.f'de görüldüğü gibi Eps değeri çok büyük (Eps=3) seçildiğinde tüm noktalar bir tane büyük küme oluşturmuştur.

2.4. OPTICS Algoritması

DBSCAN algoritmasının Eps ve MinPts olmak üzere iki adet giriş parametresine bağlı olma dezavantajını gidermek üzere DBSCAN'ı bulan grup tarafından OPTICS algoritması geliştirilmiştir (10). OPTICS algoritması kendi başına bir kümeleme aracı değildir. Algoritma daha çok bir görselleştirme aracı olarak nitelendirilebilir. Veritabanını, değişken Eps değerlerinin dağılımına göre grafik üzerinde göstererek gözle ya da çeşitli ölçümler ile dolaylı yoldan kümeleri bulmaya olanak sağlar.



(a)



(b)

Şekil 4. (a) MinPts=4 için OPTICS sonucu, (b) OPTICS algoritması sonucunun DBSCAN ile karşılaştırılması

OPTICS giriş parametresi olarak kullanıcıdan yalnızca sabit bir MinPts değeri aldığı için DBSCAN kadar parametre seçimine bağlı değildir. Sabit bir MinPts değeri için her noktanın Eps değerlerini bulur ve bunları bir grafik üzerine yansıtarak kullanıcının istediği herhangi bir Eps değerine göre küme bulmasına olanak sağlar.

DBSCAN algoritmasının daha önce açıklanan tanımlarına ek olarak, OPTICS algoritması ile iki yeni tanım yapılmıştır (10):

- çekirdek uzaklık,
- erişilebilir uzaklık.

OPTICS algoritması DBSCAN'in parametre seçimine olan bağımlılığını azaltmak için geliştirilmiştir. OPTICS algoritması sadece MinPts parametresine ihtiyaç duymaktadır. Bu çalışmada kullanılan sentetik veritabanına OPTICS algoritması uygulanırken, DBSCAN için en iyi sonucu veren MinPts=4 değeri kullanılmıştır. Bunun sonucunda elde edilen erişilebilir uzaklık (RD) grafiği Şekil 4'de görülmektedir.

OPTICS algoritması uygulanarak elde edilen sonuç, DBSCAN ile elde edilen ideal sonuç ile Şekil 4'de görüldüğü gibi karşılaştırılarak hangi kümenin hangi vadiyi temsil ettiği tespit edilmiştir. Sonuçların benzer olduğu, şekilde görülmektedir.

OPTICS algoritması ile elde edilen grafik kullanılarak istenilen herhangi bir Eps değeri için kümelenme yapısı kolayca tespit edilebilmektedir. Bu avantajına rağmen çok büyük boyutlu verilerde bu tür bir grafiğin yorumlanması çok zorlaşmaktadır. Büyük boyutlu veritabanları için farklı görselleştirme araçları önerilmektedir. Bunların bir örneği Ankerst, Breunig, Kriegel ve Sander'in makalesinde bulunmaktadır (11).

3. TARTIŞMA VE DEĞERLENDİRME

Veri madenciliği alanında yeni bir algoritma keşfedildiğinde, keşfi duyurma amacı ile yazılan makaleler, yeni bulunan algoritma ile önceki algoritmaları performans ve kullanışlılık açısından karşılaştırmışlardır. Bu, oldukça sık görülen bir durumdur. Ester, Kriegel, Sander ve Xu, KDD'96 konferansında DBSCAN algoritmasını bilim dünyasına tanıttıkları makalelerinde bu algoritmayı CLARANS algoritması ile performans bakımından karşılaştırmışlar ve sonuçları bir tablo ile göstermişlerdir (18). Hinneburg ve Keim DENCLUE algoritmasını duyurdukları makalelerinde bu algoritmayı DBSCAN ile karşılaştırmışlardır (21). 1998 yılında Sheikholeslami, Chatterjee ve Zhang, WaveCluster adını verdikleri yeni bir kümeleme tekniği ile ilgili bir makaleyi VLDB (Very Large Data Bases) konferansında sunmuşlardır. Diğerlerine benzer şekilde, bu makalede de yeni bulunan WaveCluster algoritması BIRCH ve CLARANS ile karşılaştırılmıştır (22).

Literatürde az sayıda, bilgilendirme amacı ile yapılmış karşılaştırmalara da rastlanmıştır. Atina Üniversitesi Enformatik bölümünden Halkidi, Batistakis ve Vazirgiannis belli başlı tüm kümeleme algoritmalarının karşılaştırıldığı bir makale yayınlamışlardır (23). K-means, K-modes, PAM, CLARA, CLARANS, BIRCH, CURE, ROCK, DBSCAN, DENCLUE, WaveCluster ve STING algoritmalarını içeren bu makalede herhangi bir yazılım geliştirme veya ölçüm etkinliği bulunmayıp yal-

nızca algoritmaları bulan kişilerin ispatları doğrultusunda yazılı karşılaştırmalar yer almaktadır.

Bu çalışmada, yukarıda açıklananlardan farklı olarak yeni bir algoritma önerilmemiş olmasına rağmen, yalnızca bilgilendirme içerikli bir karşılaştırma ile yetinilmemiştir. Çalışmada belirlenen algoritmalar, her bir algoritma için yazılım platformu hazırlanarak karşılaştırılmıştır. Karşılaştırma, bilinenlerden yararlanılarak yalnızca yorum getirme şeklinde değil, incelenen her algoritmanın ürettiği sonuçlar gözlenerek yapılmıştır.

Böylece K-means, DBSCAN ve OPTICS algoritmalarının uygulanması ile elde edilen sonuçlar ışığında karşılaştırılmıştır ve şu değerlendirme yapılmıştır.

DBSCAN ve OPTICS algoritmaları küresel olmayan kümelenmeleri bulma konusunda K-means algoritmasından çok daha başarılıdırlar.

K-means algoritması sıradışı noktaları bulmakta diğer iki algoritmaya göre çok daha başarısız olmuştur. Birçok durumda sıra dışı noktaları yeni bir küme gibi görme eğiliminde olduğu görülmüştür. K-means algoritması diğer iki algoritmanın tersine bulunacak küme sayısını parametre olarak istediği için, elde edilen grafiklerden görüldüğü gibi küme sayısının doğru tahmin edilemediği durumlarda kötü sonuçlar üretmektedir.

DBSCAN ve OPTICS algoritmaları benzer sonuçlar üretmişlerdir. DBSCAN algoritmasında iyi sonuçlara ulaşabilmek için farklı parametreler ile algoritmayı tekrar uygulamak gerekmektedir. OPTICS algoritmasının diğer algoritmalara üstünlüğü, sadece bir defa uygulanmasıdır. İstenilen Eps değerine göre kümeler grafik üzerinde görülebilmektedir. OPTICS algoritması görselleştirme tabanlı olduğu için K-means ve DBSCAN algoritmalarının aksine, büyük boyutlu veritabanlarına uygulandığında elde edilen grafikleri yorumlamak çok zorlaşmaktadır.

4. KAYNAKLAR

1. Han, J., Kamber, M., "Data Mining Concepts and Techniques", Morgan Kaufmann Publishers Inc., 2001.
2. Shu-hsien Liao, "Knowledge management technologies and applications—literatur overview from 1995 to 2002", Expert Systems with Applications 25, 155–164, 2003.
3. R.Grossman., C. Kamath., V.Kumar, "Data mining for scientific and engineering Approach", Kluwer Academic Publishers, Inc. (2001)
4. M.S. Chen., J. Han., P.S. Yu., "Data Mining An Overview from a Database Perspective", IEEE Transactions on Knowledge and Data Engineering, cilt 8, 866-883, 1996.
5. Michalski, R.S., Step, R.E., "Learning from Observation Conceptual Clustering", Machine Learning An Artificial Intelligence Approach, Palo Alto, California USA, 331-363. 1983.
6. Fukunaga, K., "Introduction to Statistical Pattern Recognition", Academic press, Inc., Boston, USA, 2 edition, 1990.
7. M.S. Chen., J. Han., P.S. Yu., "Data Mining An Overview from a Database Perspective", IEEE Transactions on Knowledge and Data Engineering, V 8, 866-883, 1996.
8. Fayyad, U.M., Piatesky-Shapiro, G., Smyth, P., Uthurusamy, R., "Advances in data mining and Knowledge Discovery." AAAI Pres, USA, 1994.
9. Berkhin, Pavel., "Survey of Clustering Data Mining Techniques", Accrue Software Inc., San Jose, California, USA, 2002.
10. Ester, M., Kriegel, H. P., Sander, J., Xu, X., "A density based algorithm for discovering clusters in large spatial databases." Int. Conference of Knowledge Discovery and Data Mining (KDD'96), Portland, USA 226-231, 1996.
11. Ankerst, M., Breunig, M., Kriegel, H.P., Sander, J., "OPTICS: Ordering points to identify the clustering structure." ACM SIGMOD Int. Conf. Management of Data (SIGMOD'99), Philadelphia, Pennsylvania USA, 49-60, 1999.
12. Bradley, P., Fayyad, U., & Reina, C., "Scaling clustering algorithms to large databases", In proceedings of the 4th Int'l Conference on Knowledge Discovery and Data Mining, New York, NY, pp 9-15, 1998.
13. McQueen, J., "Some methods for classification and analysis of multivariate observation". L. Le Cam and J. Neyman (Eds.), 5th Berkeley Symp. Math. Stat. Prob., 1,pp 281-297, 1967.
14. Khaled A., Sanjay R., Vineet S., "An Efficient K-Means Clustering Algorithm", IPPS: 11th International Parallel Processing Symposium, 1998.
15. http://user.it.uu.se/~kostis/Teaching/DM/Assignments Erişim tarihi Şubat 2003.
16. Khaled A., Sanjay R., Vineet S., "An Efficient K-Means Clustering Algorithm", IPPS: 11th International Parallel Processing Symposium, 1998.
17. Kaufman, L., Rosseeauw, P.J., "Finding Groups in Data: An Introduction to Cluster Analysis." John Wiley and Sons Inc., New York, USA, 1990.
18. Martin, E., Kriegel, H.P., Sander, J., Wimmer, M., Xu, X., "Incremental Clustering for Mining in a Data Warehousing Environment." , Proceedings of the 24th VLDB Conference, New York, USA, 1998.
19. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", Proc. ACM SIGMOD Int. Conf. on Management of Data, Atlantic City, New Jersey, USA 322-331, 1990.
20. Brinkhoff, T., Kriegel, H.P., Schneider, R., Seeger, B., "Efficient Multi-Step Processing of Spatial Joins", Proc. ACM SIGMOD Int. Conf. on Management of Data, Minneapolis, USA, 197-208, 1994.
21. Hinneburg A., Keim D. A., "An Efficient Approach to Clustering in Large Multimedia Databases with Noise", Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD'98), New York, USA, 58-65, 1998.
22. Sheikholeslami, G., Chatterjee, S., Zhang, A., "WaveCluster: A Multi-Resolution Clustering Approach

for Very Large Spatial Databases." Proc. 24th Int. Conf. on Very Large Data Bases, New York, USA, 428-439, 1998.

23. Halkidi, M., Batistakis, Y., Vazirgiannis M., "On Clustering Validation Techniques", Journal of Intelligent Information Systems, 17:2/3, 107–145, Kluwer Academic Publishers, 2001.