YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

AYKIRI DEĞER TESPİTİNDE YOĞUNLUK TABANLI KÜMELEME YÖNTEMLERİ

Bilgisayar Mühendisi Mennan Tekbir

FBE Bilgisayar Mühendisliği Anabilim Dalında

Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Songül ALBAYRAK

İSTANBUL, 2009

İÇİNDEKİLER

KISALTMA LİSTESİ

ACFE	Association of Certified Fraud Examiners
AGNES	Agglomerative Nesting
ASPECT	Advanced Security for Personal Communications Technologies
CM	Continuity Map
COBWEB	Incremental system for hierarchical conceptual clustering
DBSCAN	Density based algorithm for discovering clusters in large spatial data sets with noise
DCBOR	A Density Clustering Based on Outlier Removal
DDSC	A Density Differentiated Spatial Clustering Technique
DenClue	DENsity-based CLUstEring
EM	Expectation-Maximization
FTC	Federal Trade Commission
IDBSCAN	An Improved Sampling-Based DBSCAN for Large Spatial Databases
KIDBSCAN	A New Efficient Data Clustering Algorithm
MALCOM	Maximum Likelihood Continuity Mapping
OLAP	Online Analytical Processing
R-P-DBSCAN	Recursive-Partitioned DBSCAN
SOM	Self organizing feature map
STING	Statistical information grid approach to spatial data mining
VDBSCAN	Varied Density Based Spatial Clustering of Applications with Noise

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

ÖNSÖZ

Bu çalışmamda, benden desteğini esirgemeyen, anlayış sahibi hocam Yard.Doç.Songül Albayrak' a teşekkürü bir borç bilirim.

ÖZET

Sahtecilik günümüzde, sonucunda yüksek miktarlarda maddi kayıpların yaşanabildiği ciddi bir sorun haline gelmiştir. İster sahteciliği önlemek için yapılan proaktif çalışmalar olsun ister sahteciliğin yakalanması için yapılan tespitler olsun daima veri madenciliği çevresinde odaklanılmaktadır.

Veri madenciliğinin çalışma alanına giren uç analizleri(outlier detection) ile veri içinde benzerlerinden farklı davranış sergileyen aykırı elemanlar bulunabilmektedir. Bunlar, genellikle hileli unsur olmaya aday olan elemanlardır. Uç analizleri için daha çok kümeleme yöntemleri kullanılmaktadır. Gürültü veya aykırı noktalara hassas olan kümeleme algoritmaları, hileli davranışların tespitinde etkin bir rol oynamaktadır.

Kümeleme, veri madenciliğinde verinin eğiticisiz olarak analizinde kullanılan yöntemlerden biridir. Özellikle, veri hakkında bir ön bilgi yoksa, kümeleme yöntemleriyle, birbirine benzer özellikleri taşıyan veriler gruplanır. Yoğunluk tabanlı yöntemlerden biri olan DBSCAN de kümeleme işlemini yoğunluk esasına göre yapar. Gürültülü veriye duyarlı olan bu yöntem, uç analizlerinde de kullanılabilmektedir.

DBSCAN yöntemi her ne kadar küçük hacimli veri kümelerinde verimli gözükse de, veri hacminin büyümesiyle verimliği azalır. Bu nedenden dolayı DBSCAN yöntemi, büyük hacimli veri kümeleri için uygun bir kümeleme yöntemi olarak değerlendirilmez.

Bu tez kapsamında, R-P-DBSCAN(Recursive-Partitioned DBSCAN) adlı yeni bir algoritma öne sürülmüştür. Bölümleme ve birleştirme temeline dayalı olan bu yöntem, kümeleme için DBSCAN algoritmasını kullanmaktadır. Büyük hacimli kümeler, daha küçük parçalara ayrılarak DBSCAN ile kümelenmektedir. Daha sonra kümelenen her parça birleştirilerek, bütün bir veri kümesi kümelenmiş olmaktadır. R-P-DBSCAN ile elde edilen her bir küme, klasik DBSCAN ile elde edilen küme ile aynı olmaktadır.

R-P-DBSCAN ile elde edilen sonuçlar göstermiştir ki, büyük veri kümelerinde DBSCAN algoritmasına göre %85' e kadar kümeleme süresinden kazanç elde edilmektedir.

Anahtar kelimeler: Veri madenciliği, kümeleme, yoğunluk tabanlı kümeleme, uç analizleri.

ABSTRACT

Fraud that causes high amounts of finance loss, has became one of the serious problems. Either proactive efforts that focuses on prevention of fraud or working on fraud detection always use data mining approaches.

Outlier detection, which is one of the data mining studies, detects objects that has different behavior in similar elements. These elements are usually nominated to be fraudulent elements. Clustering methods are mostly used for outlier detection. Clustering algorithms that are sensitive to noise or the inconsistent elements, are playing an active role in the detection of fraudulent behavior.

Clustering is one of the data mining methods that is used for the unsupervised analysis of the data. Especially, if the data has not enough information(foreknowledge), similar data is grouped by the help of the clustering methods. DBSCAN, which is the one of the density-based clustering methods, does the process of clustering, according to density of data.

Although DBSCAN method seems effective in the small data sets, its efficiency decreases with the growing of data volumes. Because of this reason, DBSCAN as a clustering method is not considered a suitable clustering method for large data sets.

In the scope of this thesis, R-P-DBSCAN (Recursive-Partitioned DBSCAN) algorithm is proposed. The new algorithm is based on partitioning & combining and DBSCAN algorithm is used for data clustering. Large-volume data sets are divided into smaller pieces and clustered by DBSCAN. Then, combining each clustered piece, until whole set of data is clustered. Each cluster obtained by R-P-DBSCAN, is the same as the clusters obtained with the classical DBSCAN.

The results obtained with R-P-DBSCAN have shown that, the proposed algorithm has better clustering performance (until 85%) according to classical DBSCAN algorithm.

Keywords: Data mining, clustering, density-based clustering, outlier detection.

1 GİRİŞ

Sahtecilik(Fraud), teknolojinin gelişmesi ile birlikte dünya çapında adından daha fazla söz ettirmeye başlamıştır. Adından bu kadar söz ettirir olmasının nedenleri arasında, yıllık milyarlarca dolar zarara yol açan kayıplar yer almaktadır. Bireyler veya kurumlar, daha önceleri sahtecilik konusuna bütçelerinden çok fazla yer ayırmazken, günümüzde bu iş için özel danışman firmalar ve profesyonelleşmiş kadrolar ile çalışılmaktadır. Yakın gelecekte sahtecilik konusunda yapılacak çalışmaların artacağı yönündeki düşünceler ise su götürmez birer gerçek haline gelmişlerdir.

Sahteciliğin önlenmesi çalışmaları, hayatın her alanında uygulama imkânı görmektedir. Bu çalışmaların başında, iletişim, sağlık, finans sektörleri, devlet uygulamaları ve kimlik hırsızlığı gibi suçların tespiti için yapılan araştırmalar gelmektedir. Bu ana başlıkların altına birçok alt başlık da eklenip liste arttırılabilir. Bu sektörlerin bilgi işlem, kalite veya bilgi güvenliği birimleri sahtecilik ile mücadele etmektedir. Sahtecilik ile mücadelede izlenen en öncelikli yol, sahteciliğin oluşmasını engellemektir. Alınan bütün önlemler bu eksende toplanmaktadır. Sahteciliğin yapıldıktan sonra yakalanması, kurumların ancak bir sonraki sahtecilik olayını, yakalanmasında etkili olmaktadır.

2007 yılı Federal Trade Commission(FTC)' nın ABD araştırmalarına göre 1.2 milyar dolar değerinde sahtecilik tespit edilmiştir (Federal Trade Commission, 2008). Bu miktar gün geçtikçe artış göstermektedir. Herhangi bir önlem alınmadığı taktirde, önümüzdeki yıllarda oluşabilecek sahtecilik olaylarının sayısı ve zararlarının miktarı düşündürücü olmaktadır.

Sahtecilik ile mücadelede dünya çapında faaliyet gösteren Association of Certified Fraud Examiners(ACFE) gibi örgütler bulunmaktadır. Bu örgütlerin buluştukları ortak nokta, insanların bilinçlendirilmesi konusundaki çalışmalarının artırılması yönündedir. Ancak bu sayede, en etkin sahtecilik çözümü elde edilmiş olacaktır.

Sahteciliğin tespitinde kullanılan en yaygın yöntem veri madenciliği ile araştırma yapmaktır. Veri madenciliği, büyük miktarda veri içinden gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanılarak aranmasını sağlamaktadır. Büyük veri tabanlarında tutulan bilginin çeşitli yöntemler ile incelenmesi sonucunda sahteciliğe ilişkin çok sayıda yaklaşım tespit edilebilmektedir.

Nix, D. A. ve Hogden, J. E.' nin yapmış oldukları bu çalışmada, sağlık sigortası ödeme poliçesi üzerinde sahtecilik araştırması uygulaması gerçekleştirilmiş, Maximum Likelihood Continuity Mapping(MALCOM) isminde yeni bir zaman serisi analiz tekniği kullanılmıştır.

Bu uygulamada eğitim veri kümeleri, karışık sırada ele alınmıştır. MALCOM, Continuity Map(CM) adı verilen sıralı nesil modeli üretmektedir. CM, eğitim örneklerinde verilen model sabitlerinde sıranın olabilirliğini maksimum yapmakta, eğitim kümesi içinde bulunmayan sırasal verilerin olasılığını hesaplayabilmektedir. Bu işlevsellik anormal arama ve sıra tahmininde kullanılmaktadır. Hastaların yapmış olduğu işlemleri belirten tıbbi tarihlerinin olasılığında kullanılan MALCOM, kategorik veri sıralarında ele alınmakta ve veri tabanı arama araçları için potansiyel yerleşime sahip olmaktadır. MALCOM doktorların(12 tane doktorun) hastaları üzerinde denenmiş ve iyi sonuçlar elde edilmiştir(Nix vd., 1997).

Chan, P., Fan, W., Prodromidis, A. L. ve Stolfo, S. J.' Nin yapmış oldukları buaraştırmada tümevarımsal öğrenme ve meta-öğrenme metotlarını birleştirerek dolandırıcılığı tespit eden bir sistem yapılmıştır. Tümevarımsal öğrenme algoritması dağıtık veri kümeleri içerisinde alışılmışın dışındaki verileri saptamak için kullanılmakta ve meta öğrenme metotları toplu bilgilerini daha yüksek sınıflandırma modellerinde veya meta-sınıflandırıcıda birleştirmektedir. Değiş tokuş veya verilerdeki sınıflandırıcı temsillerini destekleyerek, finans sektöründe dolandırıcılıktan korunma mekanizması gerçekleştirilmiştir(Chan vd., 1999).

Kredi kartı dolandırıcılığını öğrenme, tahmin etme teknikleri için önem arz eden bu çalışmada, veri madenciliği teknikleri ve yapay sinir ağları yöntemleri hata eşik değeri düşük olacak şekilde birleştirilmiştir. Yapay sinir ağları kullanımı dezavantajı olarak % 99,9 doğru sonuçlara ulaşmak mümkün olmamakla beraber, eğitim için kullanılan süre de bir hayli uzun olmuştur. Bu da tezdeki başarıyı etkilemiştir(Brause vd., 1999).

Bu çalışmada Bayes Ağları ve Yapay Sinir Ağları çeşitli alanlarda karşılaştırılmıştır. Bu karşılaştırma sonuçlarına göre Bayes Ağlarının daha hızlı ve daha doğru eğitildiği ama yeni örneklere uygulandığında yavaş olduğu belirlenmiştir. Yapay Sinir Ağlarının ise, Bayes Ağlarının tam aksine daha zor eğitildiği, ama yeni örneklere uygulandığında daha başarılı sonuçlar verdiği bulunmuştur. Bankacılıkta da bu sistem etkili bir şekilde kullanılmıştır(Maes vd., 2002).

Advanced Security for Personal Communications Technologies(ASPECT) araştırma grubu tarafından, yapay sinir ağları, tamamen denetimsiz öğrenme ile şimdiki ve geçmişe ait kullanıcı bilgilerini içeren kullanıcı profillerini eğitmişlerdir. Bu eğitimden sonra eğer şimdiki kullanıcı profili ile geçmiş kullanıcı profili arasında çok fazla fark varsa dolandırıcı olma ihtimali vardır sonucu çıkarılmıştır(Weatherfor, 2002).

Bu çalışmada, Adapte Dolandırıcılık Tespit Sistemi geliştirilmiştir. Dolandırıcılık tespitinde,

dolandırıcılık puanı sistemi kullanılmıştır. Geniş veritabanlarında çeşitli dolandırıcılık tiplerini bulmak için kurallar çıkarılmıştır. Her bir kullanıcıya has özellikte kural tabanını otomatik olarak oluşturan bu sistem, telsiz ve kablolu sistemleri kullanan 2 milyondan fazla müşteriler üzerinde uygulanmıştır(Cahill vd., 2002).

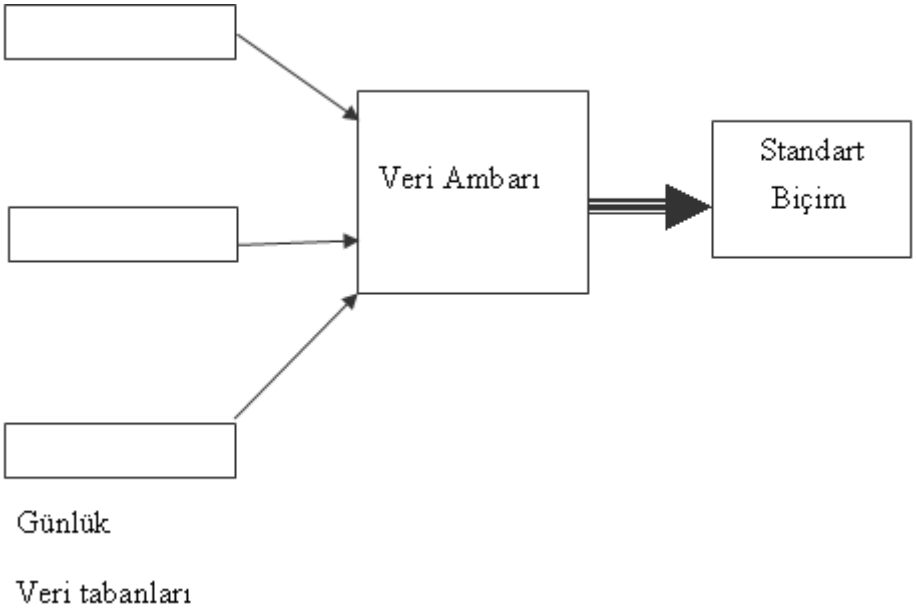
Hodge ve Austin' in 2004 yılında derledikleri "Uç Analiz Yöntemleri Üzerine Araştırma" isimli çalışmalarında, uç analizlerinin sahtecilik önlemede ne derece etkin olarak kullanılabileceğini belirtmişlerdir. Uç noktaların, insan hataları, enstrüman hataları, nüfustaki doğal kırılımlar, hileli davranışlar, sistemlerin davranış değişimleri veya hataları nedeniyle ortaya çıkabileceğini belirten yazarlar, önleme adına istatistiksel yöntemler, yapay sinir ağları, makine öğrenmesi ve hibrid sistemler gibi yöntemlerin kullanılabileceğini belirtmişlerdir(Hodge ve Austin, 2004).

"Identifying Density-Based Local Outliers" adlı çalışmada, uç nokta analizleri için yoğunluk tabanlı bir yaklaşımın nasıl olacağı belirtilmiştir. Bir noktanın, uç nokta olup olmadığını tespit etmek için kendisini çevreleyen komşu noktalardan ne kadar uzak olduğu incelenmiştir. Bu inceleme ile her noktaya LOF(local outlier factor, bölgesel uç faktör) adında bir derece verilmiştir. Çalışma sonucunda verilen deneysel sonuçlarda, yöntemin uygulanabilir bir yöntem olduğu belirtilmiştir(Breunig vd., 2000).

Sahtekârlık problemlerine yönelik birçok çözüm yöntemleri de belirtilmiştir. Bu konuda genel bir tartışma, "Sahtecilik Yakalama Teknikleri Üzerine Araştırma" adlı çalışmada sunulmuştur(Kou vd., 2004). Problemin istatistiksel boyutu Bolton ve Hand tarafından incelenmiştir(Bolton ve Hand, 2002). Kredi kartları ile ilgili çalışmalar gerek Maes vd. gerekse de Brause vd. tarafından yapılmıştır(Brause vd., 1999; Maes vd., 2002). Bu çalışmalar, yakalama sürelerinin yüksek olması ve var olan verinin ilgili algoritmanın kullanacağı hale dönüştürme ek yüküyle dikkat çekmektedir. Ayrıca, kullanılan yöntemler yapılarından dolayı, yeni hile tekniklerine karşı zayıf kalabilmektedirler. Bu çalışmalara ek olarak, problemi bir kümeleme problemi olarak değerlendiren(Weston vd., 2008) veya aynı problemi bir uç analizi olarak gören(Juszczak vd., 2008) başka yöntemler de bulunmaktadır.

2 VERİ MADENCİLİĞİ

Veri madenciliği, büyük miktarda veri inceleme amacı üzerine kurulmuş olduğu için veri tabanları ile yakından ilişkili olmaktadır. Önem arz eden verinin hızla ulaşılabilecek şekilde amaca uygun bir şekilde saklanması ve gerektiğinde bu veriye hızla ulaşılabilmesi gerekmektedir. Günümüzde yaygın olarak kullanılmaya başlanan veri ambarları günlük kullanılan veri tabanlarının birleştirilmiş hali olmakta ve işlemeye daha uygun bir özetini saklamayı amaçlamaktadır.



Şekil 2.1 Günlük veritabanlarından standart biçime dönüşün akışı

Şekil 2.1' de günlük veri tabanlarından istenen özet bilginin seçilip, gerekli ön işlemlerden geçtikten sonra veri ambarında saklanmasının ardından belirlenen amaç doğrultusunda ambardan alınarak veri madenciliği çalışması için standart bir forma çevrilmesine ilişkin akış gösterilmektedir(Alpaydın, 2000).

Veri ambarında veri oluşturulduktan sonra bu verinin elle veya gözle analizi yapılabilmektedir. Bunun için OLAP(*Online Analytical Processing*) programları kullanılmaktadır. Bu programlar verinin, her boyutu veride bir alana karşılık gelen çok boyutlu bir küp olarak kullanılıp, incelenmesini sağlamaktadır. Böylece boyut bazında gruplama, boyutlar arasındaki ilgileşimleri inceleme ve sonuçları grafik veya rapor olarak sunma olanağı sağlanmaktadır.

Veri madenciliğinde amaç, kullanıcının bilgi çıkarma sürecinde katkısının olabildiğince az tutulması, işin olabildiğince otomatik olarak yapılabilmesidir. Çünkü OLAP programlarını kullanırken bulunabilecek sonuçlar kullanıcının sormayı düşündüğü sorgularla sınırlı olmaktadır. Ama veri içinde çocuk bezi ile bira örneğindeki bağıntı gibi kullanıcının hiç aklına gelmeyecek bilgiler de olabilmektedir. Zaten veri madenciliğinde esas amaç bu tip bilgileri bulabilmektir.

2.1 Veri Madenciliğinde Kullanılan Yöntemler

Veri madenciliği çalışmalarında kullanılan yöntemler, istatistiksel yöntemler, bellek tabanlı yöntemler, yapay sinir ağları ve karar ağaçları olarak sınıflandırılabilirler.

2.1.1 İstatistiksel Yöntemler

Veri madenciliği çalışması esas olarak bir istatistik uygulamasıdır. Verilen bir örnek kümesine bir kestirici oturtmayı amaçlamaktadır. İstatistik literatüründe son elli yılda bu amaç için değişik teknikler önerilmiştir. Bu teknikler istatistik literatüründe çok boyutlu analiz (*multivariate analysis*) başlığı altında toplanmakta ve genelde verinin parametrik bir modelden (çoğunlukla çok boyutlu bir Gauss dağılımından) geldiğini varsaymaktadır. Bu varsayım altında sınıflandırma (*classification; discriminant analysis*), regresyon, öbekleme (*clustering*), boyut azaltma (*dimensionality reduction*), hipotez testi, varyans analizi, bağıntı (*association; dependency*) kurma için teknikler istatistikte uzun yıllardır kullanılmaktadır (Rencher, 1995).

2.1.2 Bellek Tabanlı Yöntemler

Bellek tabanlı veya örnek tabanlı bu yöntemler (*memory-based, instance-based methods; case-based reasoning*) istatistikte 1950'li yıllarda önerilmiş olmasına rağmen o yıllarda gerektirdiği hesaplama ve bellek yüzünden kullanılamamış ama günümüzde bilgisayarların ucuzlaması ve kapasitelerinin artmasıyla, özellikle de çok işlemcili sistemlerin yaygınlaşmasıyla, kullanılabilir olmuştur. Bu yönteme en iyi örnek en yakın k komşu algoritmasıdır (*k-nearest neighbor*) (Mitchell, 1997).

2.1.3 Yapay Sinir Ağları

1980'lerden sonra yaygınlaşan yapay sinir ağlarında (*artificial neural networks*) amaç fonksiyon, birbirine bağlı basit işlemci ünitelerinden oluşan bir ağ üzerine dağıtılmıştır (Bishop, 1996). Yapay sinir ağlarında kullanılan öğrenme algoritmaları veriden üniteler arasındaki bağlantı ağırlıklarını çıkarmaktadır. YSA istatistiksel yöntemler gibi veri hakkında parametrik bir model varsaymamakta yani uygulama alanı daha genişlemekte ve bellek tabanlı yöntemler kadar yüksek işlem ve bellek gerektirmemektedir.

2.1.4 Karar Ağaçları

İstatistiksel yöntemlerde veya yapay sinir ağlarında veriden bir fonksiyon öğrenildikten sonra bu fonksiyonun insanlar tarafından anlaşılabilecek bir kural olarak yorumlanması zor olmaktadır. Karar ağaçları ise veriden oluşturulduktan sonra yukarıdaki örnekte de olduğu gibi ağaç kökten yaprağa doğru inilerek kurallar (*IF-THEN rules*) yazılabilmektedir (Mitchell, 1997). Bu şekilde kural çıkarma (*rule extraction*), veri madenciliği çalışmasının sonucunun geçerlenmesini sağlamaktadır. Bu kurallar uygulama konusunda uzman bir kişiye gösterilerek sonucun anlamlı olup olmadığı denetlenebilmektedir. Sonradan başka bir teknik kullanılacak olsa bile karar ağacı ile önce bir kısa çalışma yapmak, önemli değişkenler ve yaklaşık kurallar konusunda bize bilgi vermekte ve tavsiye edilmektedir.

2.2 Veri Madenciliğinin Kullanıldığı Alanlar

Veri madenciliği, aşağıdaki listede belirtildiği gibi birçok alanda kullanılmaktadır (Han ve Kamber, 2006):

- Karakterizasyon ve ayrımcılık
- Bağıntı(korelasyon ve nedensellik)
- Sınıflandırma ve tahmin
- Küme analizi
- Uç analizi
- Eğilim ve evrim analizi

Uç analizi, sahtekârlık önlemede sıklıkla kullanıldığı için ilerleyen bölümlerde detaylandırılacaktır.

2.2.1 Uç Analizi(Outlier Detection)

Veri madenciliği uygulamaları, her zaman benzerlikler taşıyan veri gruplarıyla ilgilenmez. Kimi zaman, belirli bir kümeye ait olmayan veriler daha fazla önem arzetmektedir. Sahtekârlık analizi gibi uygulamalar, belirli bir gruba dahil olmayan veriler ile ilgilenmekte, onların anlizleriyle olası bir sahtekârlığı önlemeye çalışmaktadır. Uç analizleri, veri kümesi içinde, diğer verilerden aykırı davranış sergileyen -genellikle küçük hacimli- elemanların bulunması için veri madenciliği algoritmalarının yürütülmesini sağlamaktadır. Bu sayede grup içinde farklı davranış sergileyenler bulunabilmektedir.

Uç(outlier), Hawkins' in tanımına göre, dış bir mekanizma tarafından oluşturulduğundan şüphelenilen bir gözlemin, diğer gözlemlerden farklı olması durumudur(Hawkins, 1990). Bu tanımdan yola çıkarak uç noktanın şüphe uyandıran bir kaynak tarafından yapıldığı görülmektedir. Bu da sahtekârlık olması muhtemel bir verinin açığa çıkması anlamına gelmektedir.

Genellikle kümeleme yöntemleriyle yapılan gruplama sonucunda bulunan gürültülü elemanlar uç noktaların içine girmektedir. Bu nedenle gürültülü noktaları bulabilen bir kümeleme yönteminin kullanılması da uç analizinin yapılabilmesi için yeterli olmaktadır. Bölüm 4.1.1' de belirtilen yoğunluk tabanlı kümeleme yöntemlerinden DBSCAN, gürültü noktaların analizinde başarılı olduğundan, uç analizlerinde kullanılan bir yöntem olarak karşımıza çıkmaktadır.

Breunig ve arkadaşlarının yaptığı "Identifying Density-Based Local Outliers" adlı çalışmada, uç nokta analizleri için yoğunluk tabanlı bir yaklaşımın nasıl olacağı belirtilmiştir. Bir noktanın, uç nokta olup olmadığı için kendisini çevreleyen komşu noktalardan ne kadar uzak olduğu incelenmiştir. Bu inceleme ile her noktaya LOF(local outlier factor, bölgesel uç faktör) adında bir derece verilmiştir. Çalışma sonucunda verilen deneysel sonuçlarda, yöntemin uygulanabilir bir yöntem olduğu belirtilmiştir(Breunig vd., 2000).

3 SAHTEKÂRLIK

Association of Certified Fraud Examiners (ACFE) tarafından yapılan periyodik araştırmalara göre yıllık olarak ABD' de yapılan sahtekârlık zararı 640 milyar doları bulmaktadır. Bu moral bozucu gerçek, sahtekârlığın saptanmasındaki güçlükten kaynaklanmaktadır. Araştırmaların gösterdiği verilere göre, yapılan sahtekârlıkların % 44' ü personel tarafından yapılmaktadır. Fakat iç merkezli yapılan sahtekârlık kontrolleri sahtekârlıktan korunmada ölçülebilir etkiye sahip olabilmektedir. Araştırmalara göre, bazı iç merkezli sahtekârlık kontrol prosedürlerine daha az önem veren kuruluşların güvenlik açığı bulunmaktadır.

Toptan ticaret, inşaat ve imalat sektörleri dolandırıcılık için en hassas sektörler olmakla birlikte hükümet ve perakende kuruluşlarının da zayıflıkları bulunmaktadır. Genel olarak muhasebe departmanı bulunan kişiler tarafından üst yönetim tarafından toplam sahtekârlık yüzdesi % 30 - % 20 arasında iletilmiştir.

KPMG uluslar arası sektörel odaklı denetim, vergi ve danışmanlık firması dolandırıcı insan ve faaliyetlerinin profillerine ilişkin bir rapor yayımlamıştır. Bu raporda, sahtekârlık faaliyetlerinde bulunan insanların % 85' inin erkek olduğu ve bu kişilerin yaşlarının 36 - 55 yaş arasında olduğu belirtilmiştir.

Sahtekârlık üzerine yapılan ilk akademik araştırma 20. yüzyılda bir araştırmacı Edwin Sutherland tarafından yapılmıştır. Sutherland' e göre suç diğer insanların yardımı olmadan meydana gelmemektedir. Suç faaliyetleri içerisinde bulunanlar genellikle samimi kişisel gruplar oluşturmaktadırlar.

Başka bir araştırma da Donald R. Cressey tarafından Indiana Üniversitesi 1940' larda yapılmıştır. Araştırma Sahtecilik Üçgeni olarak adlandırılan bir model oluşturulmasına dayanmaktadır. Cressey' e göre, bir dolandırıcılık taahhüdünde bulunan 3 temel adım bulunmaktadır. Onun teorisi sahtekârlık algılama sistemi için bir temel olarak günümüzde kabul edilmektedir.

Dolandırıcılık denetiminin bir diğer öncüsü de Brigham Young Üniversitesi' nden Dr. Steve Albrecht' tir. Albrecht Keith Howe ve Marshall Romney isimli iki çalışma arkadaşı ile, 1980' lerde 212 dolandırıcılık hakkında analizler yapmışlar ve demografik kapsamlı anket ile sahtekârlık eylemlerinde bulunan insanlar hakkında çıkarımlarda bulunmuşlardır.

3.1 Sahtekârlık Tipleri

Aşağıda sahtekârlık tipleri açıklanmaktadır.

3.1.1 Personel Sahtekârlıkları

Personel sahtekârlıkları değişen iş uygulamaları nedeniyle artmaktadır. Bunun en yaygın görüldüğü yerler gider faturalarının büyük rakamlar olarak gösterilmesi ve rüşvet olmaktadır. Bir organizasyonda yapılan araştırmalara göre çalışanların % 5' in daima dürüst düşündüğü, çalışanların % 10' unun ise her zaman hileli bir şekilde düşündüğü ortaya çıkarılmıştır (Bozkurt, 2002).

3.1.2 Yönetim Sahtekârlıkları

Yönetim sahtekârlıkları genellikle bölüm yöneticileri ve diğer yöneticilerin kendi birimlerinin performans düzeylerine ilişkin yapmış oldukları sahtekârlıklardır. Bu tür sahtekârlıkların temel nedeni ikramiye, promosyonlar ve birimin başarılı performansına bağlı diğer alınacak olan teşvikler olmaktadır.

3.1.3 Yatırım Sahtekârlıkları

Bu yüzyılda artan teknoloji sayesinde, insanlar bilgilere çok çabuk şekilde ulaşabilmektedirler. Bu noktada meydana gelen sorun, insanların bilgilere çok kolay şekilde ulaşmaları sahtekârlıklara yol açabilmektedir. Yatırım sahtekârlıkları, ana amaç kendi değerlerini almak için bir proje üzerinde yatırım yapan insanlar, borsa veya şirket ile aldatmak üzerinedir. Bu araçlar zamanla değişmektedir ve en yaygın olanları tahviller, sıfır kupon tahvili, hisse senedi ve benzeri varlıklardır.

3.1.4 Satıcı Sahtekârlıkları

Satıcı sahtekârlığı, birisinin korunmayan emeğinizi çalmasının en kolay yollarından biridir. Satıcı sahtekarlığı döküm düşürme nedenleri arasında yer almaktadır. ABD' deki Florida Üniversitesi ve National Retail Security Üniversitesi tarafından yapılan araştırmalara göre şirketlerin envanterlerinin bu şekilde azaltılması ile satışlarında % 1,80 oranında düşüşler yaşanmaktadır.

3.1.5 Müşteri Sahtekârlıkları

Müşteriler genellikle aldatıcı mülk edinme konusunda sahtekârlıklarda bulunmaktadırlar. Bir ankete göre İngiliz Perakende Konsorsiyumu tarafından, müşteri sahtekârlıklarının en yaygın türünün, sırasıyla kredi kartı, çek ve bankamatik kartları ile yapılan sahtekârlıklar olduğu açıklanmıştır.

3.2 Sahtekârlık Önleme

Sahtekârlığı yakalama yaklaşımından daha önemli olan yaklaşım, proaktif olmaktır. Olası bir sahtekarlık eylemi daha olmadan yakalayıp, önlemini almak, bu eylemi bir problem olarak karşımıza çıkarmayacaktır. Finans kurumları, iletişim servis sağlayıcıları veya sigortacılık şirketleri yaptıkları yatırımları daha çok, sahtekârlığın önlenmesine ayırmaktadır.

Sahtekârlığı önlemede kullanılan yöntemler genel olarak sınıflama ve kümeleme olarak karşımıza çıkar. Sınıflama yöntemleri, daha önce sahtekârlık yapıldığı belirlenen bir veri kümesinin olması durumunda kullanılabildiği halde; kümeleme yöntemlerinde tam tersi bir durum söz konusudur. Veri kümesi hakkında herhangi bir bilgi yoksa, yani içinde hangi verilerin sahtekarlık yapan gruba ait olduğu bilgisi yoksa, daha etkin olarak yararlanılmaktadır.

Dikkat edilecek olursa sınıflandırma yöntemleri, var olan hileli davranışların önceden bilinmesini gerektirmektedir. Bu da sahtekârlık yöntemlerinin değişmesi halinde, yeni hileli davranışların bulunamayabileceğini akla getirmektedir. Kümeleme yöntemleri ise, veri hakkında herhangi bir bilgiye sahip olmadan çalışabildikleri için, sınıflandırma yöntemlerinde bulunan bu engeli içermemektedir. Teknolojinin her geçen gün ilerlemesi ve sahtekârlık davranışlarının gün geçtikçe kendilerini geliştirmeleri ve yenilemeleri göz önüne alındığında, kümeleme yöntemlerinin daha etkin olacağını söylemek yanlış olmayacaktır.

2.2.1 nolu bölümde belirtildiği gibi uç analizi, veri madenciliği ile sahtecilik ve benzeri aykırı veri analizinin önemli olduğu durumlarda kullanılan başarılı bir yöntemdir. Bu sayede benzer unsurlardan farklı davranış sergileyen elemanlar, bir diğer ifadeyle sahtekârlık yapması muhtemel unsurlar bulunabilmektedir. Kümeleme yöntemleri ile aktif olarak kullanılabilecek bu yaklaşım, daha etkin bir hile yakalama ve önleme disiplini sunacaktır.

Hodge ve Austin' in 2004 yılında derledikleri "Uç Analiz Yöntemleri Üzerine Araştırma" isimli çalışmalarında, uç analizlerinin sahtecilik önlemede ne derece etkin olarak kullanılabileceğini belirtmişlerdir. Uç noktaların, insan hataları, enstrüman hataları, nüfustaki doğal kırılımlar, hileli davranışlar, sistemlerin davranış değişimleri veya hataları nedeniyle ortaya çıkabileceğini belirten yazarlar, önleme adına istatistiksel yöntemler, yapay sinir ağları, makine öğrenmesi ve hibrid sistemler gibi yöntemlerin kullanılabileceğini belirtmişlerdir(Hodge ve Austin, 2004).

Sahtekârlık problemlerine yönelik birçok çözüm yöntemleri de belirtilmiştir. Bu konuda genel bir tartışma Kou ve arkadaşları tarafından düzenlenen "Sahtecilik Yakalama Teknikleri

Üzerine Araştırma" adlı çalışmada sunulmuştur(Kou vd., 2004). Problemin istatistiksel boyutu Bolton ve Hand tarafından incelenmiştir(Bolton ve Hand, 2002). Kredi kartları ile ilgili çalışmalar gerek Maes vd. gerekse de Brause vd. tarafından yapılmıştır (Brause vd., 1999; Maes vd., 2002). Bu çalışmalar, yakalama sürelerinin yüksek olması ve var olan verinin ilgili algoritmanın kullanacağı hale dönüştürülmesi ek yüküyle dikkat çekmektedir. Ayrıca, kullanılan yöntemlerin yapısından dolayı, yeni hile tekniklerine karşı zayıf kalabilmektedir. Bu çalışmalara ek olarak, problemi bir kümeleme problemi olarak değerlendiren(Weston vd., 2008) veya aynı problemi bir uç analizi olarak gören(Juszczak vd., 2008) başka yöntemler de bulunmaktadır. Dikkat edilecek olursa sahtekârlık probleminin çözümünde tek bir yaklaşım bulunmamakta olup farklı yöntemler ile soruna çözüm aranmıştır.

4 KÜMELEME YÖNTEMLERİ

Kümeleme(öbekleme, gruplama), veri madenciliğinde kullanılan disiplinlerden biridir. Veriyi eğiticisiz olarak(eğitici olmadan), birbirlerine benzeyen elemanlardan oluşan sınıflara (kümelere) ayırarak, heterojen bir veri grubundan, homojen alt veri grupları elde edilmesi işlemidir.

Bir diğer ifadeyle kümeleme, veritabanını bölümlendirir. Her bir küme, diğer kümelerden farklı özellikler içeren elemanlardan kuruludur(Michalski vd., 1983) (Zhong ve Ohsuga, 1994).

Kümeleme fonksiyonu genellikle bölümleme sorunlarını çözmekte kullanılır. Kümelemenin temel hedefleri arasında; geniş veri yığınları için tanımlayıcı veriler belirleyerek, işlenecek veri hacmini daraltmak, veri yığınlarındaki doğal kümeleri ortaya çıkararak aynı kümede olması gereken verileri belirlemek, belirlenmiş kümelerin dışında kalan istisna durumları tanımlamak sayılabilir. Başlangıç aşamasında verilerin hangi kümelere ayrılacağı veya kümelemenin hangi değişken özelliklerine göre yapılacağı bilinmemekte, konunun uzmanı olan bir kişi tarafından kümelerin neler olacağı tahmin edilmektedir.

Kümeleme algoritmaları; küme içinde benzerliğin maksimize (küme içi uzaklıkların minimize edilmesi) edilmesi, kümeler arası benzerliğin minimize (kümeler arası uzaklıkların maksimize edilmesi) edilmesi kavramına dayanır. Sonuçta elde edilen farklı kümelere ait elemanlar arasında benzerlik azdır.

Kümeleme fonksiyonu ile sınıflandırma fonksiyonu arasındaki en önemli fark, kümelemenin önceden tanımlanmış girdilere dayanmıyor olmasıdır. Sınıflandırma fonksiyonunda tanımlı girdiler ve bunların geçmişte aldıkları değerler temel modeli oluştururken, kümeleme fonksiyonunda önceden tanımlanmış girdiler ve örnekler yoktur. Veriler kendi içlerindeki benzerliklere göre gruplanırlar. Benzerliği tanımlayacak boyutlar ve özellikler modeli kuran tarafından öngörülür.

Kümeleme fonksiyonu bazı durumlar başka bir veri madenciliği fonksiyonun öncesinde kullanılabilir. Hangi promosyon kampanyasına müşteriler en iyi tepkiyi verirler diye değerlendirmek yerine öncelikli olarak müşterilerin belirli kümelere ayrılmasının ardından her küme için en iyi promosyon kampanyasının ne olacağı belirlenebilir. Müşterileri kümelemek için genellikle karlılık ve pazar potansiyeli boyutları kullanılır. Perakende sektöründe müşterilerin söz konusu firmadaki alım alışkanlıkları ve tüm mağazalardaki alım alışkanlıklarına göre kümelenmeleri ve en yüksek potansiyelli kümeye odaklanılması sıkça

rastlanan bir uygulamadır.

Benzer hücreleri tanımlamak, benzer davranışlar gösteren perakende müşterilerini tanımlamak, gen ve protein analizleri, ürün gruplaması, hastalık belirtileri, metin madenciliği, kümelemenin kullanıldığı başlıca uygulama alanlarıdır.

Bu yönteme örnek olarak yaş ve gelir düzeyleri belirtilmiş 40 kişiden oluşan bir grubu, grafik yardımı ile kümelerine ayırmak mümkündür. Yaş ve gelir düzeyi değerlerinin histograma yerleştirilmesi ve en yoğun durumların merkez olarak belirlenmesi en basit anlamda bir kümeleme işlemidir(Argüden ve Erşahin, 2008). Bu örnekte veri madenciliği yöntemleri kullanılmadan kümeler oluşturulmuştur. Ancak onlarca değişken olduğunda verileri kolayca kümelemek mümkün değildir, bu aşamada kümeleme fonksiyonuna özgü algoritmaları kullanmak gereklidir.

Kümeleme, bir veri setinde bulunan sınıfları belirlemede etkin rol oynayan bir yöntemdir. Uzaysal ve büyük ölçekli veritabanlarında aşağıda belirtilen hususların göz önünde tutulması gerekmektedir. (Ester vd., 1996)

- Çalıştırılacak algoritmalarda kullanılacak giriş parametrelerinin belirlenmesi, veritabanı hakkında çok az bilgiye sahip olunduğu için zordur.
- Veritabanı içinde öbekleşen kümeler, belirli bir geometrik şekilde olmayabilir.
- Veritabanı büyük olduğunda efektif bir yöntem olmalıdır.

Kümeleme yöntemleri, benzerlerinden farklı davranış sergileyen verileri bulmak için kullanılan uç analizleri içinde etkin bir şekilde kullanılabilmektedir. Veri kümesi kümelendikten sonra, ortaya çıkan uç noktalar(bunlar gürültü veya istisnalar olabilir) analiz edilerek istenen sonuçlara ulaşılabilmektedir. Sahtekârlık önlemede de kullanılan bu uç analizleri ile hileli olmaya aday öğeler bulunup, daha detaylı çalışmalar için incelemeye alınabilmektedirler.

Literatürde birçok kümleme yöntemi bulunmaktadır. Kimi zaman bir yöntem başka bir yöntemin içine girebildiği için bunları kategorize etmek oldukça zordur. Ama genel hatları ile sıralama yapmak istenirse, aşağıdaki gibi bölümlere ayrılabilirler(Han ve Kamber, 2006):

4.1 Bölümlemeli(Partitioning) Yöntemler

Dışarıdan alınan bir k sayısı adedince bölüm oluşturma esasına dayanmaktadır. Oluşan her bölüm bir kümeyi belirtir. Kümeleme işlemi, her nesnenin mutlaka bir bölüm içinde yer

almasına ve her bölümün en az bir nesneden oluşma esaslarına dayanır.

Verilen k sayısına göre başlangıç bölümü oluşturulur. Belirli bir iterasyon sayısına veya küme içi benzerliklerin aynı kalmasına kadar nesneler, yerdeğiştirme ile bölümler arası geçişlerde bulunabilir. Bölüm içi benzerlik ne kadar yüksek ve bölümler arası uzaklık ne kadar çok ise iyi bir kümeleme işleminden söz edilebilir.

Bu yöntem ile ilgili bir çok algoritma bulunmaktadır. Bunların en önemlileri k-means(Lloyd, 1957; MacQueen, 1967) ve k-medoids(Kaufman ve Rousseeuw, 1990) algoritmalarıdır.

4.2 Hiyerarşik(Hierarchical)Yöntemler

Hiyerarşik veya bir diğer ismiyle aşamalı yöntemler, veri kümesini hiyerarşik olarak ayrıştırır. Bu ayrıştırma birleştirmeli(agglomerative, bottom-up) veya bölmeli(divisive, top-down) olarak yapılabilir. Birleştirmeli yapıda, her nesne tek bir grup olarak ele alınarak bir diğer nesneyle olan yakınlığına göre gruplandırılır. Bu gruplandırma işlemi, tüm nesneler bir grup oluncaya kadar veya belirli bir şartın sağlanmasıyla sonlandırılır. Bölmeli yaklaşımda ise tüm veri kümesi bir grup olarak değerlendirilip en sonunda her nesne bir grup oluncaya kadar bölme işlemi devam ettirilir.

En bilinen hiyerarşik kümeleme yöntemleri AGNES(Kaufman ve Rousseeuw, 1990) ve DIANA(Kaufman ve Rousseeuw, 1990)' dır.

4.3 Izgara(Grid) Tabanlı Yöntemler

Bu yöntemde veriler, bir ızgara yapısı gibi bölümlere ayrılır. Kümeleme işlemleri her bir ızgara yapısı için ayrı ayrı yapılır. Bu yöntemle yapılan kümelemenin, hızlı işlem süresi ve veri boyutundan bağımsız kümeleme gibi faydaları bulunmaktadır. En bilinen ızgara tabanlı yöntemlere STING(Wang vd., 1997) örnek verilebilir.

4.4 Model Tabanlı Yöntemler

Model tabanlı yöntemlerde, veri kümesindeki her küme için bir model kurulup, veriye en uygun model seçilir. Model tabanlı bir algoritma, verileri uzaydaki dağılımlarına göre kümeleyebilir. Aynı zamanda bir takım istatistiksel yöntemlerle, gürültü ve uç noktaları dikkate alarak, küme sayısını otomatik bir şekilde bulabilir.

Model tabanlı kümeleme yöntemleri arasına EM(Dempster vd., 1977), COBWEB(Fisher, 1987) ve SOM(Kohonen, 1982a; 1989b) girmektedir.

4.5 Yoğunluk Tabanlı Yöntemler

Kümeleme, nesnelerin farklı gruplara sınıflandırılmasıdır. Her gruptaki veriler ortak bazı özellikleri paylaşabilirler. Çok sayıda kümeleme algoritması bulunmaktadır ve yoğunluk tabanlı yaklaşım onlardan biridir.

Yoğunluk tabanlı yaklaşımda, verilen bir veritabanında(veri noktalarından oluşan set) bulunan ve bir birbirine çok yakın noktaların belirli bir yoğunluğa bağlı olarak oluşturduğu kümeler bulunmaktadır. Bu şekildeki kümeler herhangi bir biçimi alabilirler. Veri setini oluşturan elemanlar küme içindeki yoğunluklarına göre kimi zaman içi boş bir halka kimi zaman da kıvrılmış bir yay şeklinde kümelenebilir. Bir veritabanındaki herhangi bir kümeye ait olmayan veri noktaları kümesinin gürültü olduğu düşünülür. Bunun yanı sıra, kümenin içine uzanan çekirdek nokta bir tanedir ve bir sınır nokta da kümenin sınırına uzanan bir noktadır.

DenClu(DENsity-based CLUstEring), küme içindeki her bir noktanın yoğunluğunu, o noktanın diğer tüm noktalara göre etki fonksiyonun(influence function) toplamı olarak değerlendirip buna göre kümeleme işlemini yapmaktadır(Hinneburg vd., 1998).

DCBOR(A Density Clustering Based on Outlier Removal), hem Uç noktaların giderilmesi hem de kümeleme işlemini birlikte yapar. Giriş parametresi olarak 0 ile 1 arasında değişen uç noktalar için eşik değeri girişi bekleyerek, farklı şekil, yoğunluk ve boyutlarda kümeler keşfeder(Fahim vd., 2008) .

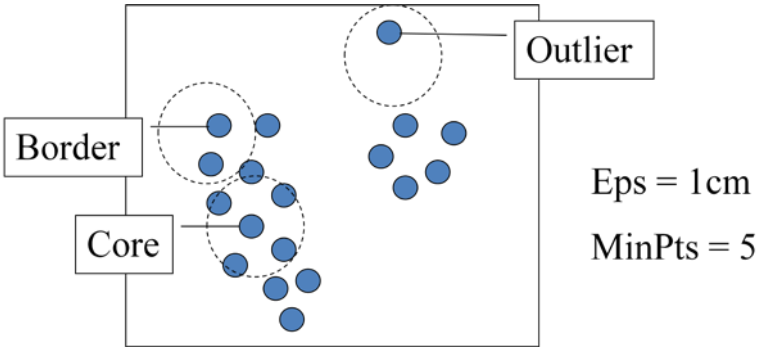
HIERDENC(Efficient Layered Density-based Clustering of Categorical Data), biyomedikal alanında kategorik verilerin kümelenmesi için öne sürülmüştür. Bu yöntem ile her bir eleman, yönteme has bir indeks yapısı ile indekslenir. Dışarıdan herhangi bir parametre almayan bu yöntemde, kümeye yeni gelen üyenin, hızlı bir şekilde, ait olduğu kümesi belirlenir. Bir küme içinde farklı yoğunluğa sahip diğer küme de bulunabilmektedir.(Andreopoulos, 2008)

4.5.1 DBSCAN

DBSCAN(Density based algorithm for discovering clusters in large spatial data sets with noise), içinde gürültünün de olduğu uzaysal veritabanlarında, noktaların yoğunluğuna bağlı olarak kümeleme yapmak için kullanılan bir yöntemdir. Yöntem dışarıdan sadece bir giriş parametresi alıp, rastgele şekillerde kümeleme yapabilen, uzaysal büyük ölçekli veritabanlarında etkin bir şekilde kullanılabilen bir yöntemdir(Ester vd., 1996).

Şekil 4.1 DBSCAN örnek veri kümeleri

Şekil 4.1' deki örnek veri setlerinden de görüleceği üzere noktalar, belirli bir yoğunluğa göre kümelenmiş durumdadır. DBSCAN yönteminin ana iskeletini, bir küme içinde bulunan noktaların yoğunluğunun diğer kümelerin oluşturduğu yoğunluktan farklı olması oluşturmaktadır. Gürültü verilerinin oluşturduğu alanı da bir küme olarak değerlendirirsek, o kümenin yoğunluğunun, diğer kümelere kıyasla çok düşük olduğunu fark edebiliriz. Gürültü verilerinin atılması bu yöntemle sağlanabildiği için DBSCAN veri ön-işlemede de kullanılabilecek bir yöntem haline gelmiştir.



Şekil 4.2 DBSCAN algoritmasının temel bileşenleri

DBSCAN yönteminin bam telini, bir nokta çevresinde olan noktaların, o noktayı orta nokta kabul eden bir r yarıçaplı çemberin içinde giren nokta sayısı, bir diğer ifadeyle yoğunluğu oluşturur. Çevresel noktaların ki bunlara komşu noktalar da denebilir, sayısının belirli bir eşik değerini geçmesi, bu noktanın bir küme içinde olduğunu gösterebilir. Çevresel noktaların, o noktaya olan uzaklıklarını bulmada kullanılacak mesafe fonksiyonuna göre kümenin şekli değişecektir. Örneğin Manhattan ile bulunacak mesafeler, ortaya dörtgenimsi bir şeklin çıkmasını sağlayacaktır. Öklid mesafesi ise daha eliptik şekilleri ortaya çıkaracaktır.

Algoritmada Epslion ve MinPts adlı iki global parametre kullanılmıştır(Şekil 4.2). İki nokta arasındaki mesafe Epsilon değerinden büyük ise ve bu noktaların sayısı MinPts sayısından daha fazla ise bu nokta asıl nokta olarak değerlendirilir. Bir asıl nokta bulunduktan sonra çevresindeki diğer noktalara da gidilerek onların da asıl olup olmadıkları test edilir. Bu

şekilde birbiri ardına bulunan asıl noktalar, belirli bir yoğunluk değerine ulaştığı için, bir küme olarak değerlendirilir(Ester vd., 1996).

Bir noktanın çevresel komşuluklarına göre incelendiği DBSCAN yöntemi, gürültülü noktaların analizinde başarılı bir grafik çizmektedir. Giriş parametrelerinin uygunluğuna göre oluşacak gürültülü noktalar, uç analizlerinde kullanılabilecek noktalar kümesini oluşturmaktadır. Bu noktadan yola çıkarak, DBSCAN yönteminin hileli davranış anlizlerinde kullanılabilecek bir yöntem olduğundan söz edilebilmektedir.

4.5.1.1 DBSCAN Yöntemi ile Yapılan Çalışmalar

IDBSCAN(An Improved Sampling-Based DBSCAN for Large Spatial Databases), var olan algoritmayı ölçeklenebilme açısından geliştirmeyi amaçlamıştır. Asıl nokta çevresindeki Epsilon komşuluğundaki noktalar içinden bir örneklem alıp bu noktaları İşaretlenmiş Sınır Nesne(MBO) olarak işaretler ve bu MBO' lar içinden en yakın nokta başlangıç olarak değerlendirilip DBSCAN algoritması işletilir. Bu yöntem, klasik DBSCAN algoritması ile aynı sonuçları üretmesine karşın daha verimlidir(Borah ve Bhattacharyya, 2004).

KIDBSCAN(A New Efficient Data Clustering Algorithm), k-Means ile IDBSCAN algoritmasının bileşiminden oluşur. K-Means, asıl noktaların bulunmasında kullanılır. Geri kalan işlemler IDBSCAN ile yürütülür(Tsai ve Liu, 2006).

VDBSCAN(Varied Density Based Spatial Clustering of Applications with Noise), birkaç aşamadan oluşan ve Epsilon değerlerinin her nokta için k-mesafe grafiğinin çıkartılmasıyla elde edilen bir yöntemdir. Kümelenen veri setleri, farklı Epsilon değerlerine göre şekillenir. Bu yöntem Epsilon değerinin iyi bir şekilde tahmin edilmesi amaçlanarak yapılmıştır(Liu vd., 2007).

DDSC(A Density Differentiated Spatial Clustering Technique) yönteminde bir set üzerinde farklı yoğunluklara sahip kümelerin bulunması amaçlanır. Klasik DBSCAN yöntemindeki parametrelere ek olarak bir parametre daha ister. Bu parametre değerlerini iyi bir şekilde ayarlamak, daha başarılı kümeleme için önemlidir. Birbirine komşu kümeler, aralarındaki yoğunluk miktarı fazla olduğunda ikiye bölünür(Borah ve Bhattacharyya, 2008).

4.5.1.2 DBSCAN Yönteminin Avantajları

DBSCAN yönteminin, diğer yoğunluk tabanlı yöntem ve diğer kümeleme yöntemlerine göre birtakım güçlü noktaları bulunmaktadır. Bunları aşağıdaki gibi sıralayabiliriz(Andreopoulos, 2008) :

- Düzgün olmayan şekillerde küme bulunabilmektedir.
- Gürültü veri setlerinde çalışabilmektedir.
- O($\log_n$) karmaşıklığa sahip olduğundan efektif bir yöntemdir.

4.5.1.3 DBSCAN Yönteminin Dezavantajları

Yoğunluk tabanlı yöntem olan DBSCAN' in bir takım zayıf noktaları olduğu gözlemlenmiştir(Andreopoulos, 2008).

- Bir set üzerinde farklı yoğunluklara sahip kümeler olabilir. Bu durumda Epsilon değeri bir küme için uygun olurken, diğer küme için uygun olmayabilir(Ankerst vd., 1999).
- Büyük ölçekli veri setleri üzerinde uzun sürebilmektedir(Guha vd., 2000).
- Giriş parametrelerini kestirmek zordur.
- Bir veri setine yeni bir eleman eklendiğinde, bu elemanın hangi kümeye girdiğini bulmak kolay olmayabilir.
- Bir küme içinde başka bir küme yer alıyorsa, onu keşfetmek mümkün olmayabilir(Grambeier vd., 2002).

4.5.1.4 DBSCAN Yöntemini Geliştirme Noktaları

Yapılan araştırmalar incelemeler sonucunda DBSCAN yönteminin geliştirmeye uygun görülen noktaları aşağıda belirtilmiştir:

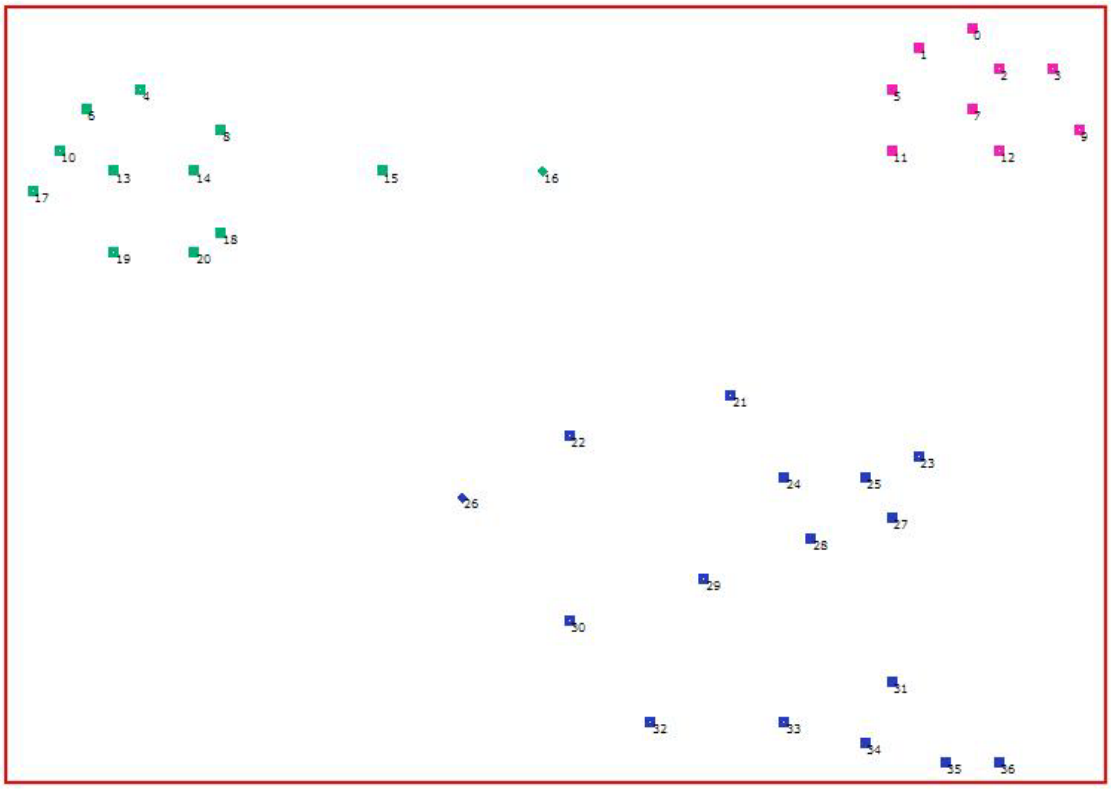
- DBSCAN yönteminin dezavantajlarından biri büyük ölçekli veritabanları üzerindeki kümeleme hızıdır. Bunu geliştirebilmek için bu algoritma paralel çalışabilecek hale getirilebilir. Bu sayede veri seti bölünerek, dağıtık sistemler üzerine gönderilebilir. Farklı sistemler aynı anda kendi sorumluluğundaki bölgeyi kümeler. En sonunda ise bu kendi içinde kümelenmiş veriler bir araya gelerek bütün için kümeleme yapılmış olur.
- Kümelenmiş bir veri seti içinde, başka bir küme de bulunabilir. Her ne kadar diğer büyük küme içinde değerlendirilse bile, bu gizlenmiş veri kümesini kümeleyebilmek, kimi durumlarda önemli olabilmektedir.
- Giriş parametrelerini kullanıcının seçimine bırakmadan hesaplayabilmek, yöntemi tamamen otomatik hale getirebilmek için gereklidir. Epsilon ve MinPts giriş parametrelerini bir takım ön işlemler ile kestirebilmek için bir takım yöntemler bulmak gereklidir.

5 ÖNERİLEN METOD: R-P-DBSCAN

DBSCAN, her ne kadar kümeleme yöntemleri içinde başarılı bir yer edinse de, büyük ölçekli veri setleri üzerinde kümeleme uzun sürebilmektedir(Guha vd., 2000). Var olan yöntemin, performans açısından verimli olmaması, yeni bir algoritma geliştirmeye yönlendirmiştir. İlerleyen bölümlerde, bu algoritma detaylandırılacaktır.

5.1 Önerilen Metodun Çıkış Noktası

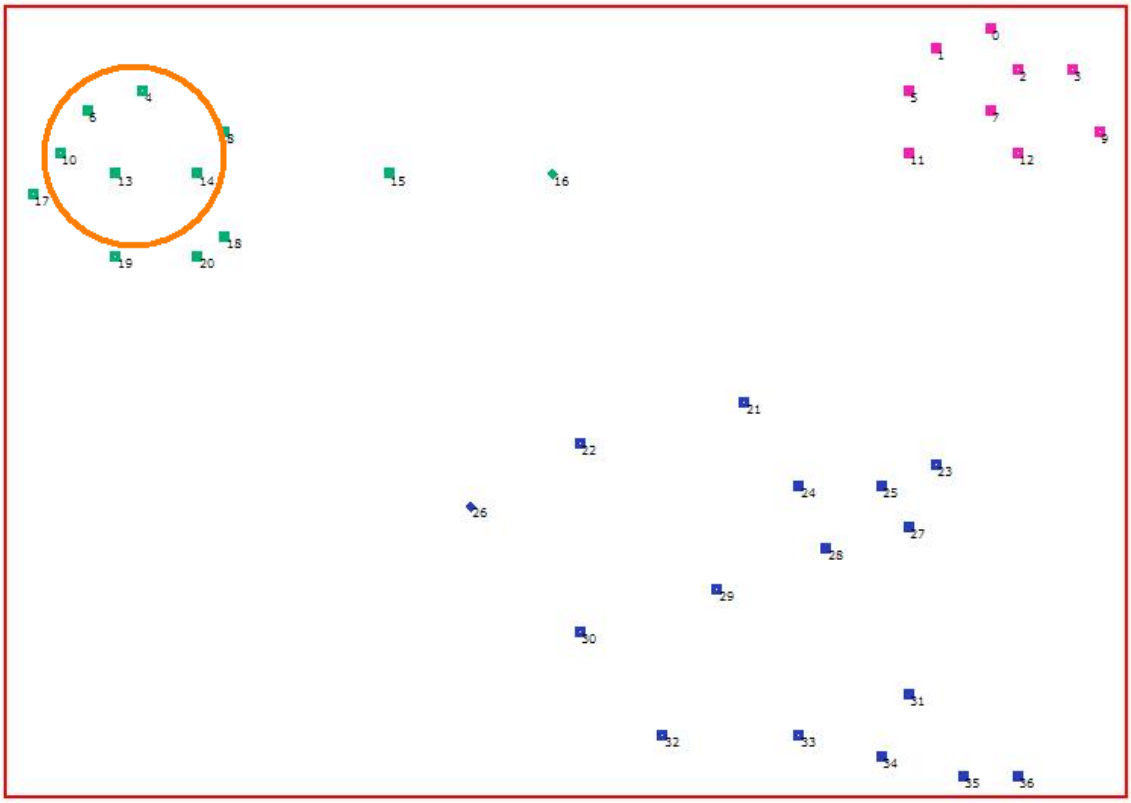
DBSCAN yöntemi incelediğinde, bir noktanın CORE nokta olup olmadığının testi için, kümedeki tüm elemanlar ile uzaklığına bakılmaktadır. Bu da kümedeki elamanların sayısıyla doğrusal olmayan bir hızla artmaktadır. DBSCAN, uzaklık testi yapacağı elemanları küçültebilirse, gereksiz hesaplamalardan kaçınılacağı gözlenmiştir. Bu nokta esas alınarak R-P-DBSCAN(Recursive-Partitioned DBSCAN) yöntemi bu tez çalışmasında önerilmiştir.



Şekil 5.1 Örnek veri seti

Şekil 5.1' deki veri seti incelendiğinde, 13 numaralı eleman üzerinde yapılacak uzaklık testinde, epsilon değeri çok büyük verilmediği sürece, 36 veya 12 numaralı elemanlara

bakmanın bir önemi olmayacaktır. Şekil 5.2' de 13 numaralı elemana epsilon mesafesinde bulunan elemanlar görülmektedir.



Şekil 5.2 Epsilonlu örnek veri seti

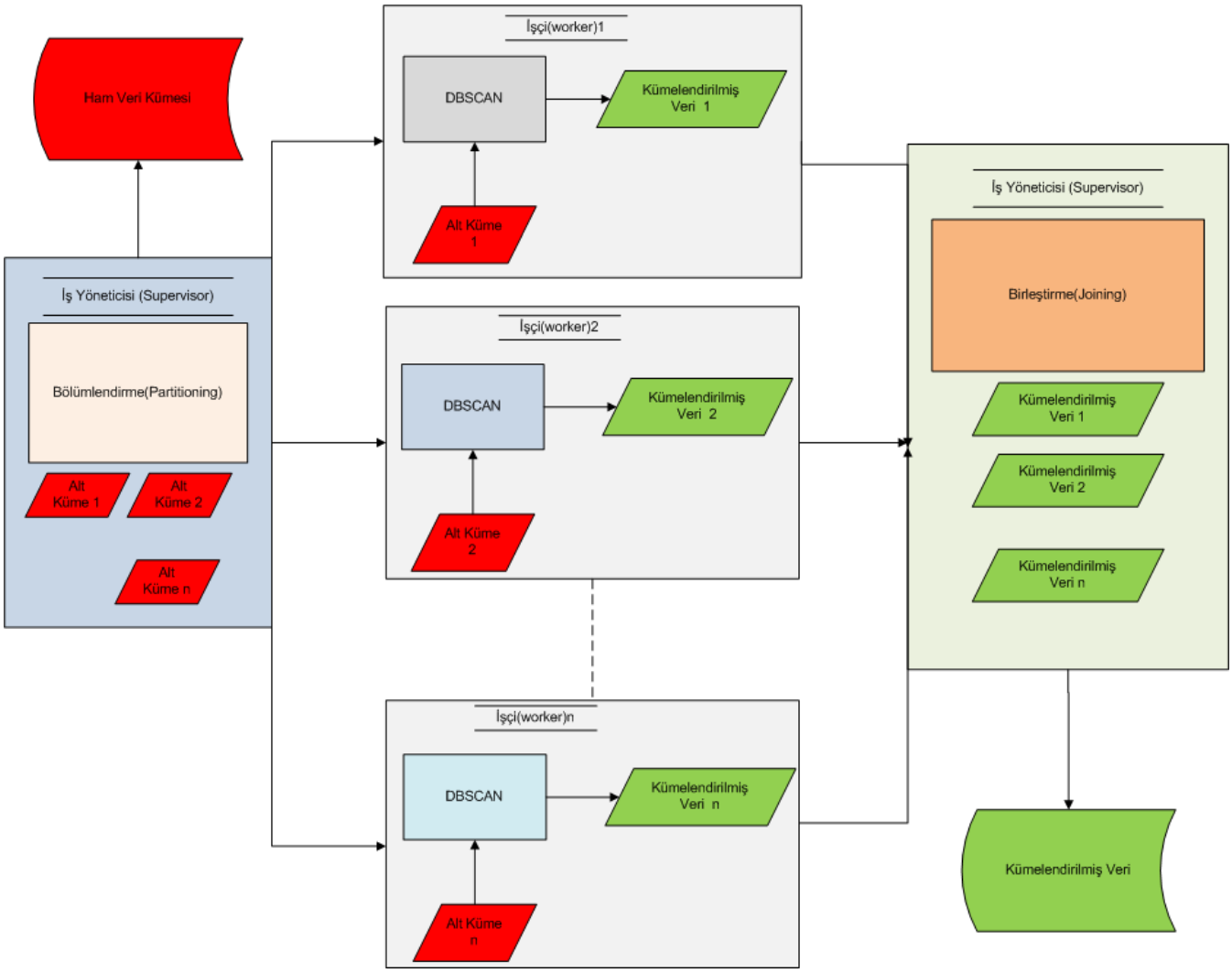
Şekil 5.2' den de görüleceği üzere, toplamda 37 elemanın oluşturduğu bir kümede, sadece 4(% 15 oranında) noktanın epsilon mesafesinde diğer 32(% 85 oranında) noktanın daha uzak mesafede olması ve gereksiz hesaplamalara dahil olması verimi düşürmektedir. Bir yöntemle, uzaklık testinde bakılacak noktaların sınırlandırılabilmesi, DBSCAN yönteminin verimini arttıracaktır. Öne sürülen R-P-DBSCAN yöntemiyle, bu problemin iyileştirilmesi amaçlanmaktadır.

5.2 Önerilen Metodun Ana Noktaları

R-P-DBSCAN, isminden de anlaşılacağı üzere, DBSCAN algoritmasının bölümlendirme(partitioning) yöntemiyle çalıştırılması esasına dayanmaktadır. Bir veri kümesi önce bölümlere ayrılıp, her bölüm kendi içinde DBSCAN ile kümelendirilmekte, daha sonra bu bölümlendirilen parçalar birleştirilerek ana set kümelendirilmiş olacaktır. Bölümlendirme ve birleştirme işlemleri özyinelemeli(recursive) olarak yapılarak daha etkin bir yöntem

izlenecektir. Bölümlendirilen alt kümeler, dağıtık bilgisayar sistemlerine gönderilip DBSCAN algoritması paralel bir şekilde çalıştırılarak, daha etkin bir yol izlenebilecektir.

R-P-DBSCAN algoritmasının ana akış diyagramı Şekil 5.3' de gösterilmiştir:



Şekil 5.3 R-P-DBSCAN akış diyagramı

Yeni yöntem, bir iş yöneticisi(supervisor) ile işçilerin(worker) koordineli bir şekilde çalışması esasına dayanmaktadır.

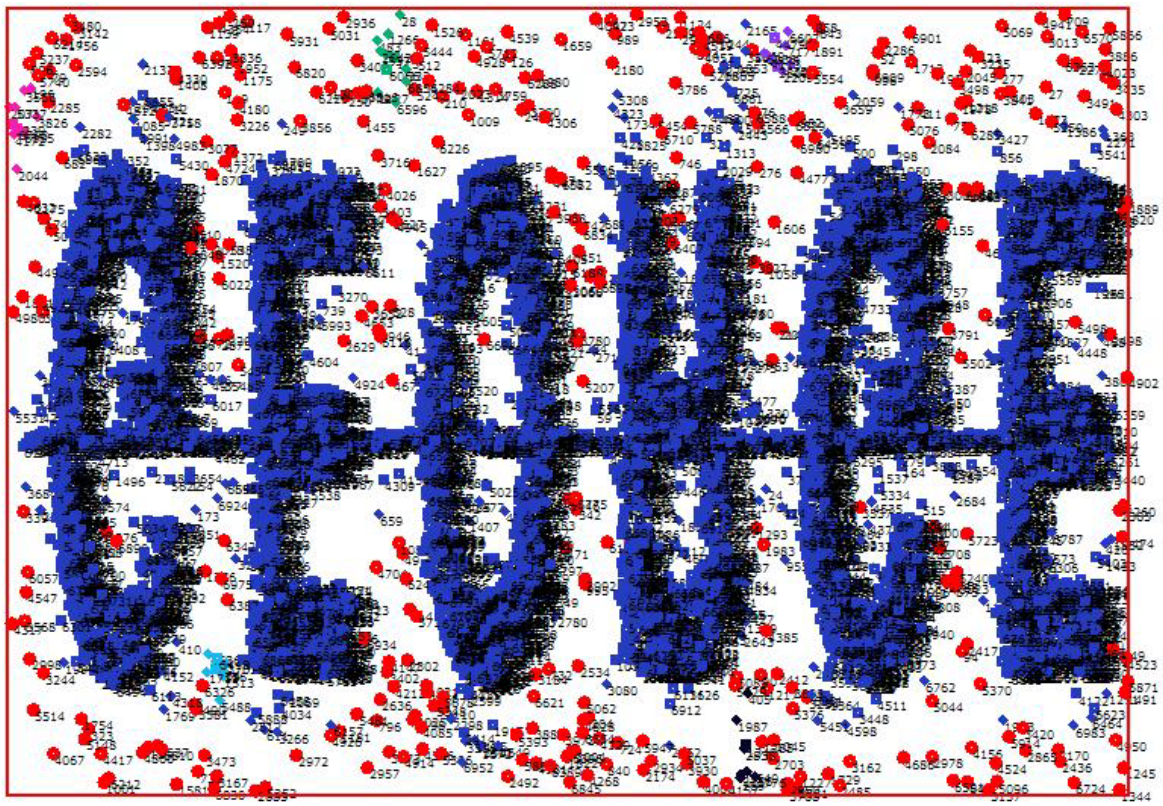
5.2.1 Bölümlendirme (Partitioning)

Veri setini oluşturan elemanlar, küme uzayında dağınık bir şekilde durmaktadır. Bölümlendirme, dışarıdan parametre olarak verilen ve ikinin üssü(2^n) olacak bir sayı ile yapılmaktadır. Bölüm sayısının fazla olması, paralel olarak işlenebilecek veri alt kümesi sayısını arttıracağından daha etkin bir kümelemeyi sağlayacak; buna karşın özyinelemeli olarak yapılan bölümlendirme ve birleştirme işlemlerini arttıracağından da sisteme ek bir yük

getirecektir.

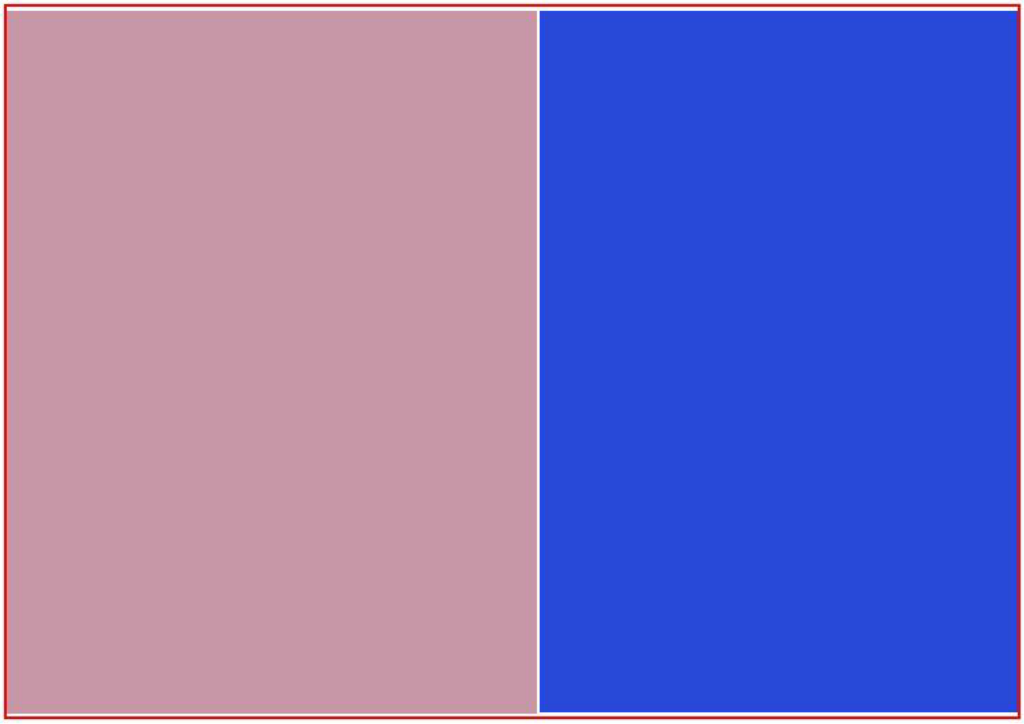
Bölümlendirme yapılırken, veri kümesinin tamamı alınarak, X ve Y eksenleri kullanılır. Hangi eksende bölümleme yapılacaksa, veri setindeki noktalar, o eksene göre sıralanır. Nokta sayısının yarısındaki(median) koordinat bilgisinde, o eksen, dik bir doğru ile iki parçaya bölünür. Bu işlem, istenilen bölüm sayısına ulaşıncaya kadar, özyinelemeli olarak, devam eder.

Örnek olarak Şekil 5.4' deki veri kümesini ele alıp bu küme üzerinde 4 bölüm oluşturmak isteyelim.



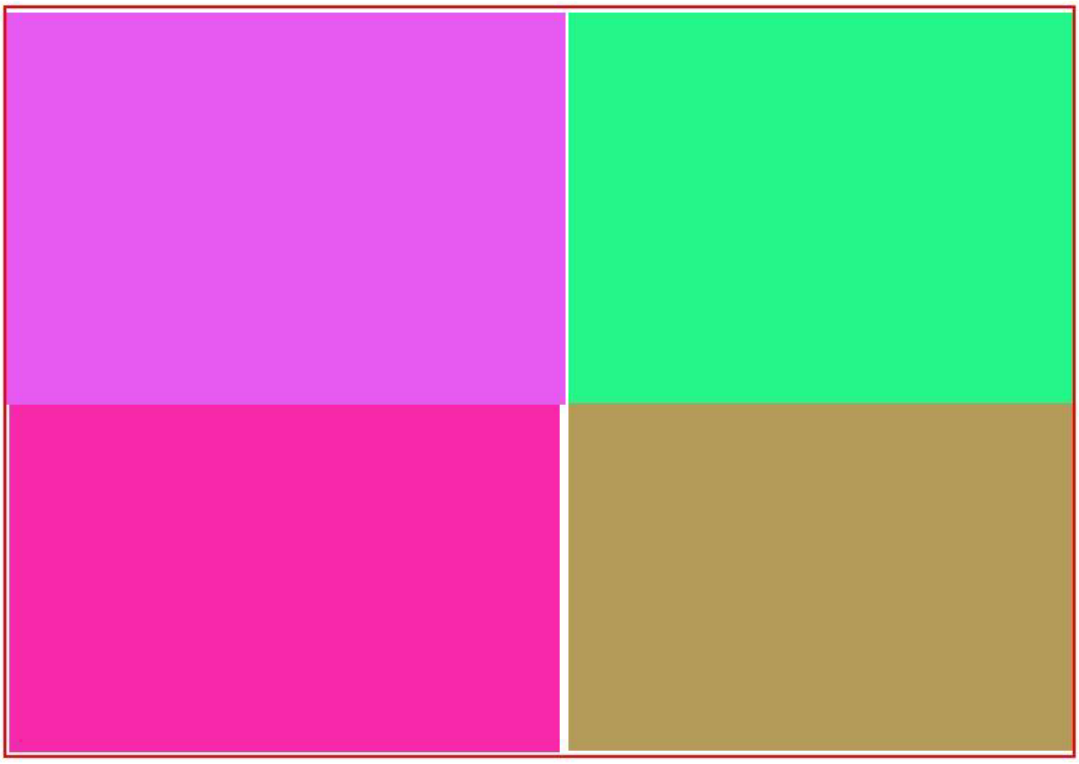
Şekil 5.4 George veri kümesi (DBSCAN, e=10, m=7)

Kümenin tamamı bir bölüm olarak değerlendirilirse, önce X eksenini dik kesen iki parça oluşturulacağı için, bölüm içindeki tüm noktalar X' e göre sıralanır. Şekil 5.4' deki veri kümesinde toplam 7000 nokta olduğundan, sıralanan noktalardan 3500 noktanın apsisi, X eksenine göre çizilecek dik doğrunun koordinatını belirler. Doğru çizildikten sonra artık bölümleme yapılmış, iki ayrı bölüm elde edilmiş olmaktadır(Şekil 5.5).



Şekil 5.5 George veri setinin 2 bölüme ayrılmış hali

İki bölüme ayrılan parçalardan her biri de kendi içinde Y eksenine göre iki ayrı bölüme ayrılır. Bu bölümde anlatılan bölümlendirme algoritması, özyinelemeli olarak işletilerek toplamda 4 ayrı bölüm edilmiş olur(Şekil 5.6).



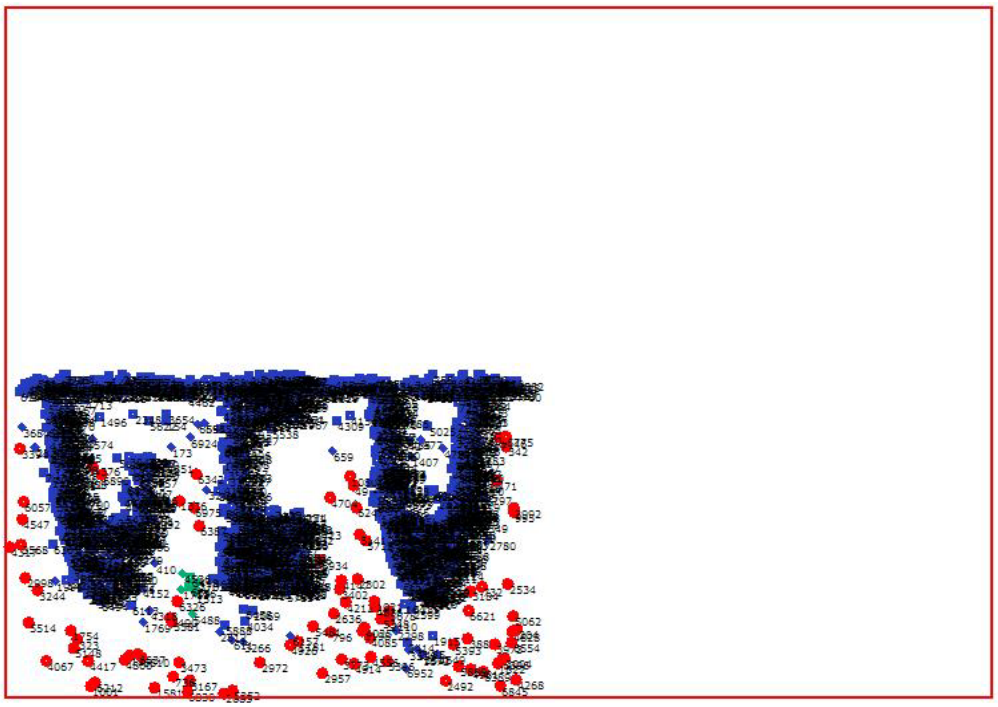
Şekil 5.6 George veri setinin 4 bölüme ayrılmış hali

Akış diyagramı Şekil 5.3’ de belirtilen algoritmanın bölümleme kısmı, aşağıda genel hatları belirtilen algoritma ile çalışır:

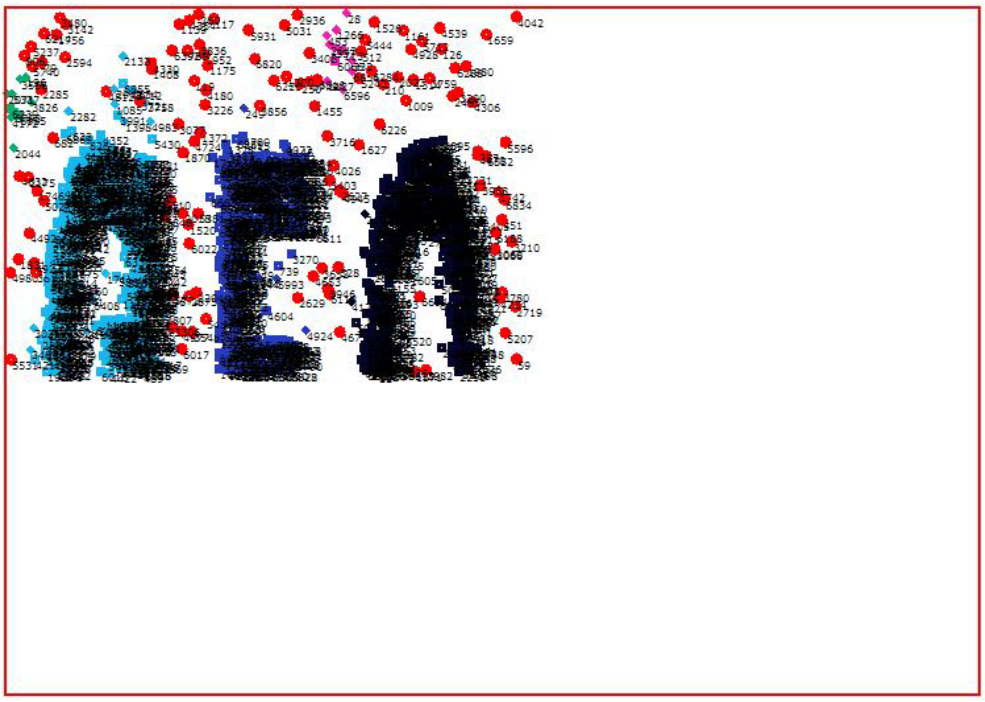
```
Partitionise(Partition _partition,int _partitionCount,Partition[] _partitionedList,
String _sortAxisName){
  if (_partitionCount < 2){
    _partitionedList[last] = _partition;
  }
  else{
    SortPoints(_partition, _sortAxisName);
    int MedianPointIndex = _partition.Points.Length / 2;

    DBSCANPoint[] FirstPoints = GetPoints(_partition.Points,0,MedianPointIndex-1);
    DBSCANPoint[] SecondPoints = GetPoints(_partition.Points, MedianPointIndex,
_partition.Points.Length);

    Partition FirstPartition = new Partition(FirstPoints);
    Partition SecondPartition = new Partition(SecondPoints);

    String SortAxisName = GetNextAxis(_sortAxisName);

    Partitionise(FirstPartition _partitionCount/2,_partitionedList,SortAxisName);
    Partitionise(SecondPartition,_partitionCount/2,_partitionedList,SortAxisName);
  }

}
```

5.2.2 DBSCAN

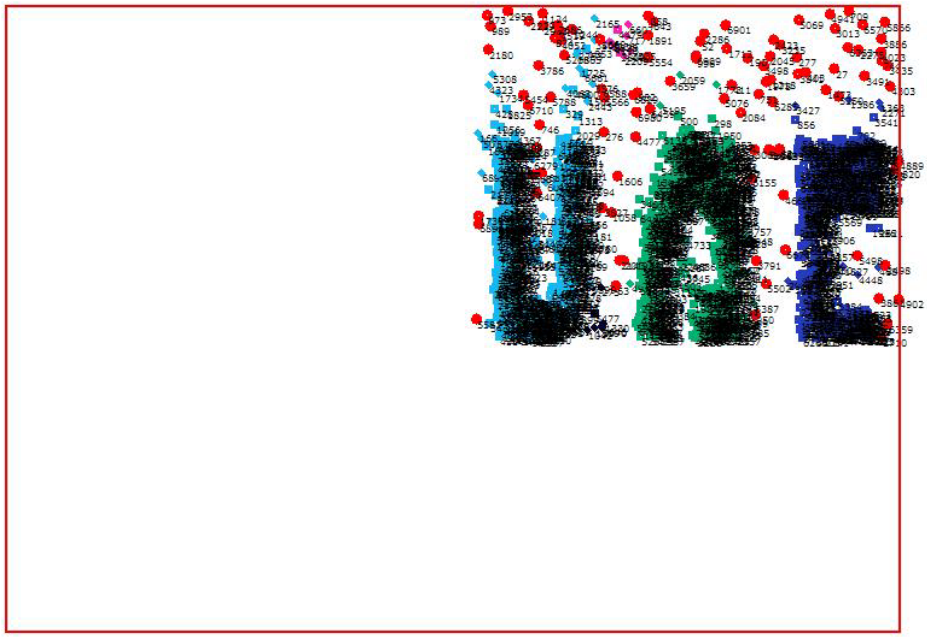
Kullanılan DBSCAN yöntemi üzerinde herhangi bir değişiklik yapılmamış olup orijinal hali kullanılmıştır. Bir bütün olarak tüm parçalar üzerinde DBSCAN algoritmasının çalıştırılması yerine, bölümlendirilmiş her bir alt veri kümesi üzerinde algoritma çalıştırılmıştır. Sonuç olarak elde 4 farklı bölüme ait noktaların oluşturduğu kümelenmiş veri seti oluşmuştur(Şekil 5.7, Şekil 5.8, Şekil 5.9 ve Şekil 5.10).



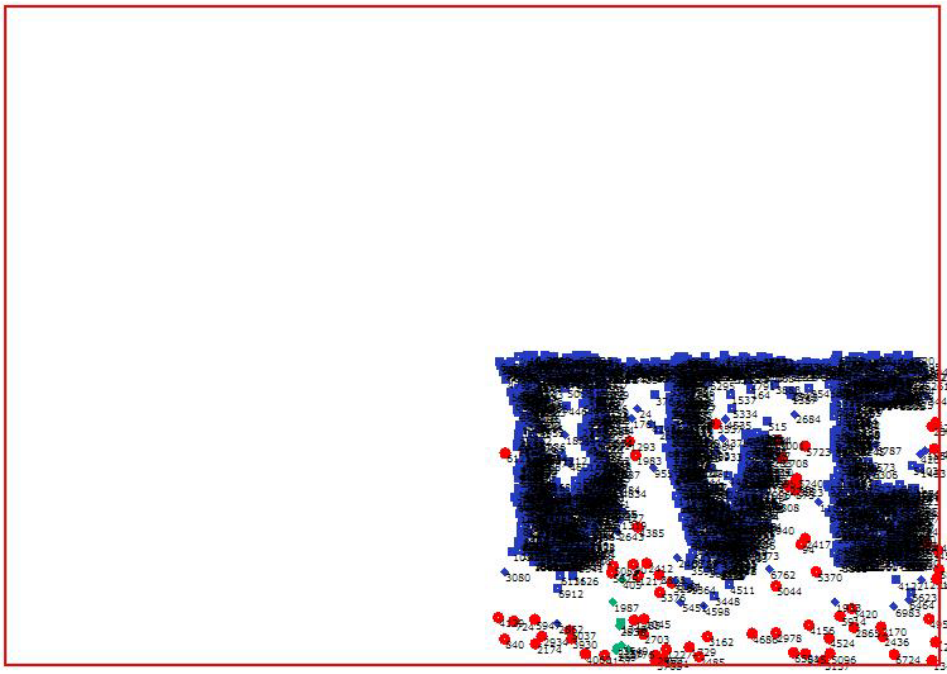
Şekil 5.7 George veri setinin 1. bölümüne ait DBSCAN



Şekil 5.8 George veri setinin 2. bölümüne ait DBSCAN



Şekil 5.9 George veri setinin 3. bölümüne ait DBSCAN



Şekil 5.10 George veri setinin 4. bölümüne ait DBSCAN

Bölümlenmiş parçalardaki DBSCAN algoritmasının yürütülmesi, dağıtık sistemler üzerinde olursa, geçecek toplam süre, en uzun süren DBSCAN işlemi kadar olacaktır. Bu noktadan yola çıkarak önerilen yöntemin dağıtık sistemler üzerinde paralel olarak çalışabildiğini söylebiliriz.

5.2.3 Birleştirme(Joining)

Birleştirme, bölümlenmiş ve kümelenmiş noktaların birleştirilerek tüm noktaları kapsayacak kümelemeyi bulma işlemidir. Buradaki asıl amaç, bölümlendirilen kısımlar birleştirilirken, birleşme noktalarındaki kümelerin tek küme haline gelmesini sağlamaktır. Bu sayede tüm veri kümesi, tek bir bütün halinde kümelenmiş olacaktır.

Birleştirme işlemi, bölümlendirme işleminde olduğu gibi özyinelemeli olarak devam eder. İki bölüm birleştirilerek tek bölüm haline getirilir. Parçaların, bölümlendirme sırasına göre birleştirme işlemi yapılır. Birleştirme işlemi, en sonunda tek bölüm kalınca bitirilir.

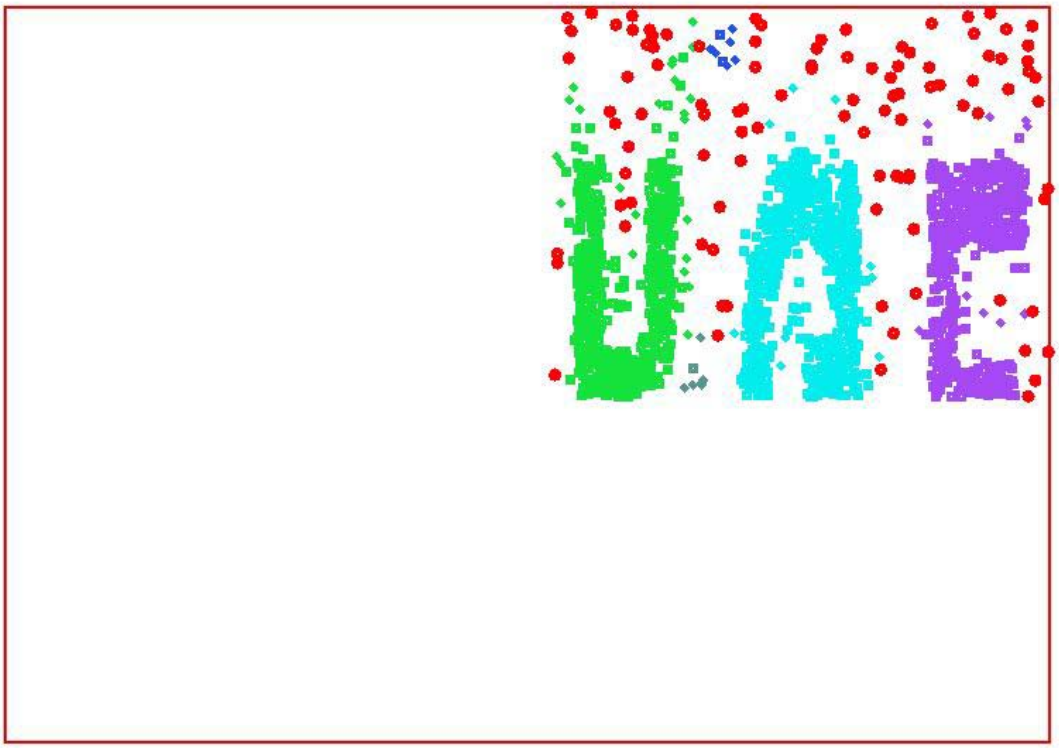
DBSCAN yöntemiyle her bölüm ayrı ayrı kümelendirildiğinden, bölüm birleşme noktalarında, noktaların özellikleri değişebilir. Bir bölüm içinde köşe(border) olarak kümelenen nokta, birleştirme işlemi sonunda asıl(core) nokta halini alabilir. Bu nedenle

birleştirme işlemi, bölümlendirmede yapıldığı kadar basit olmaz.

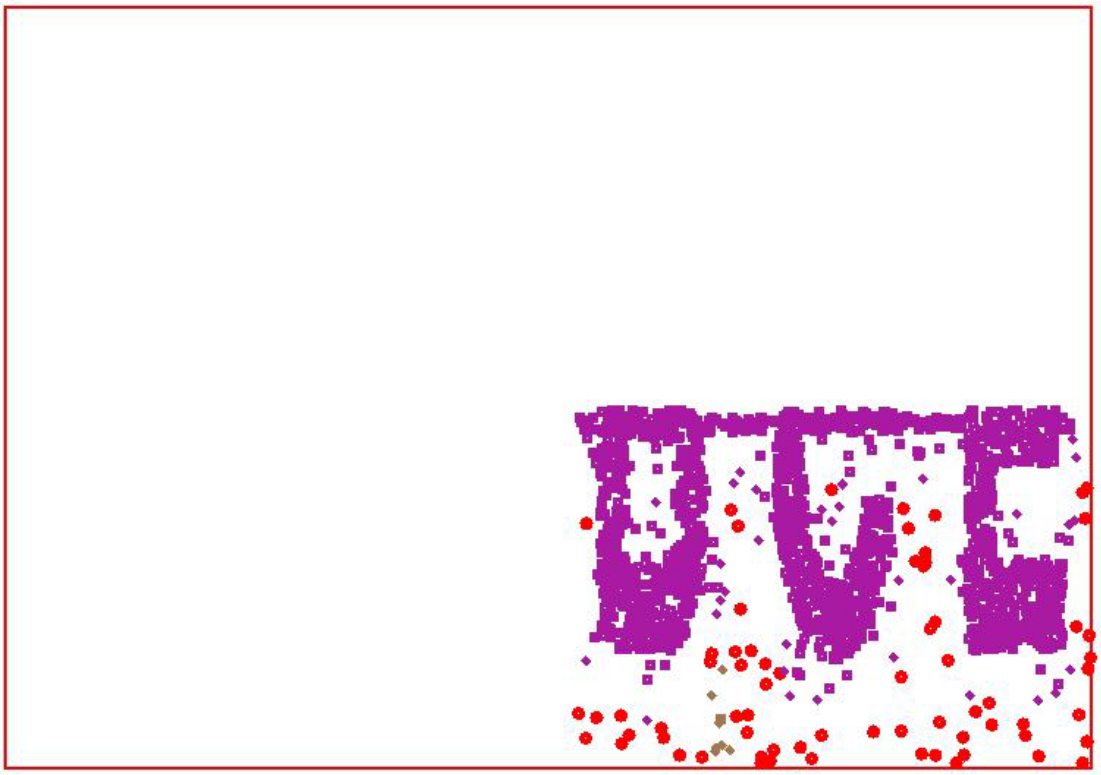
5.2.3.1 İki Bölümün Birleştirilmesi

İki bölüm birleştirirken, tüm veri kümesinin kümeleme yapısının değişmemesi için bir takım işlemler yapılır. Bu işlemlerin nasıl olduğunu daha iyi belirtebilmek için Şekil 5.4' de bulunan George veri kümesi incelendiğinde burada bulunan kümede toplam 7000 nokta bulunduğu görülmektedir. DBSCAN ile yapılan kümeleme sonucu toplamda 6 farklı kümenin oluştuğu ortaya çıkmıştır.

George veri kümesinin bölümlendirme işlemi sonucu, 4 farklı bölüme ayrılır(Şekil 5.6). Her bölüm kendi içinde DBSCAN(m=7, e=10) ile kümelenir. Oluşan 4 farklı kümelenmiş veri, ikişer ikişer alınarak birleştirme işlemine tabi tutulur. Şekil 5.11 ve Şekil 5.12' de ilk iki bölümün DBSCAN(e=10,m=7) ile kümelenmiş hali belirtilmiştir. İlk bölüm toplamda 6 farklı küme; ikinci bölüm ise toplamda 2 farklı küme içermektedir.



Şekil 5.11 George veri kümesi ilk bölümün kümelenmiş hali

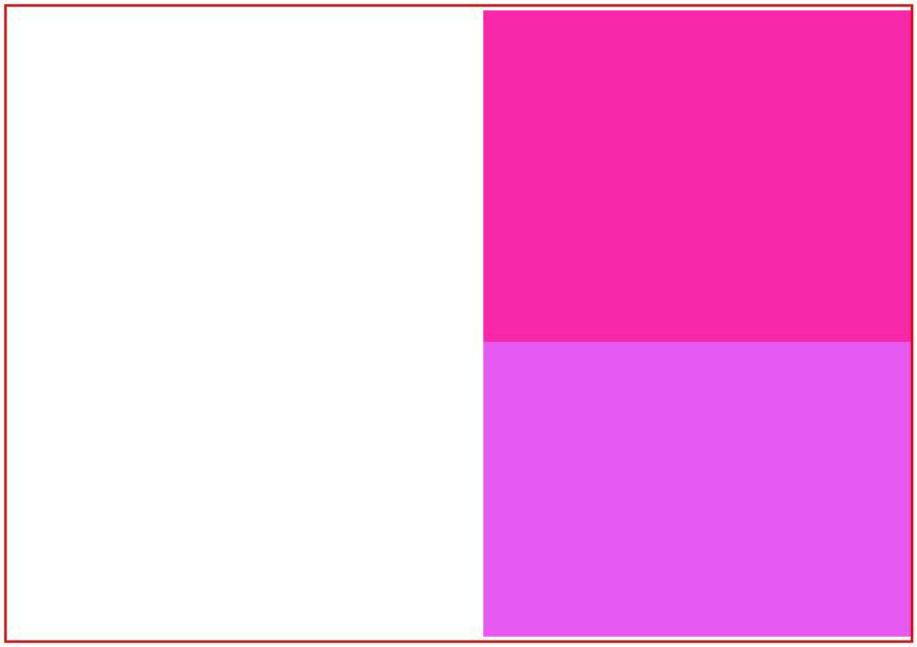


Şekil 5.12 George veri kümesi ikinci bölümün kümelenmiş hali

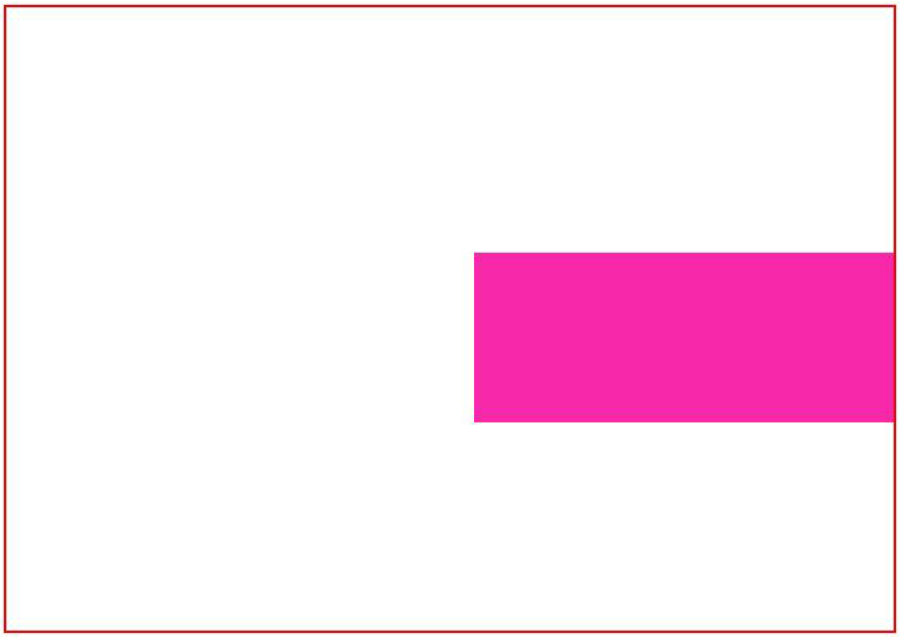
Dikkat edilirse, ilk bölümde aslında tek küme olarak kümelenmesi gereken 3 farklı kümenin oluştuğu görülecektir. Noktalar bütün olarak değerlendirildiğinde, 3 farklı küme, altındaki bölüm içindeki noktaların birleşmesiyle tek küme haline gelecektir. Her bölüm, sadece kendi elemanlarını görebileceği için, kümeleme işlemini de o elemanlar çerçevesinde yapacaktır. Buradan yola çıkarak yapılan bölümlendirmenin, gerçekte kümelenen bir takım noktaların özelliklerini değiştirdiği söylenebilir. Bu, kimi zaman gürültülü noktaların, gerçekte gürültü olmadıkları şeklinde de olabilmektedir. Bu noktaların kümelenebilmesi için yeni bir yöntem geliştirilmiştir. Bu yöntem ile, bölümlerin birleşme noktalarını kapsayan yeni bir bölüm oluşturulmuş, o bölüm de DBSCAN ile kümeleme işlemine tabi tutulmuştur. Bu sayede birleşme noktalarında oluşacak kayıplar önlenmiştir.

İki bölümün birleşme noktalarını kapsayan yeni bölümün sınırları, alt ve üst bölümlerin uç noktalarından 2 epsilon uzaklığında oluşturulmuştur(Şekil 5.13 ve Şekil 5.14). Bir noktanın DBSCAN ile analizinde epsilon mesafesindeki noktalara bakıldığı için, bu mesafe uygun görülmüştür. Teorik olarak birbirlerine epsilon mesafesinden daha uzak olan bölümlerdeki

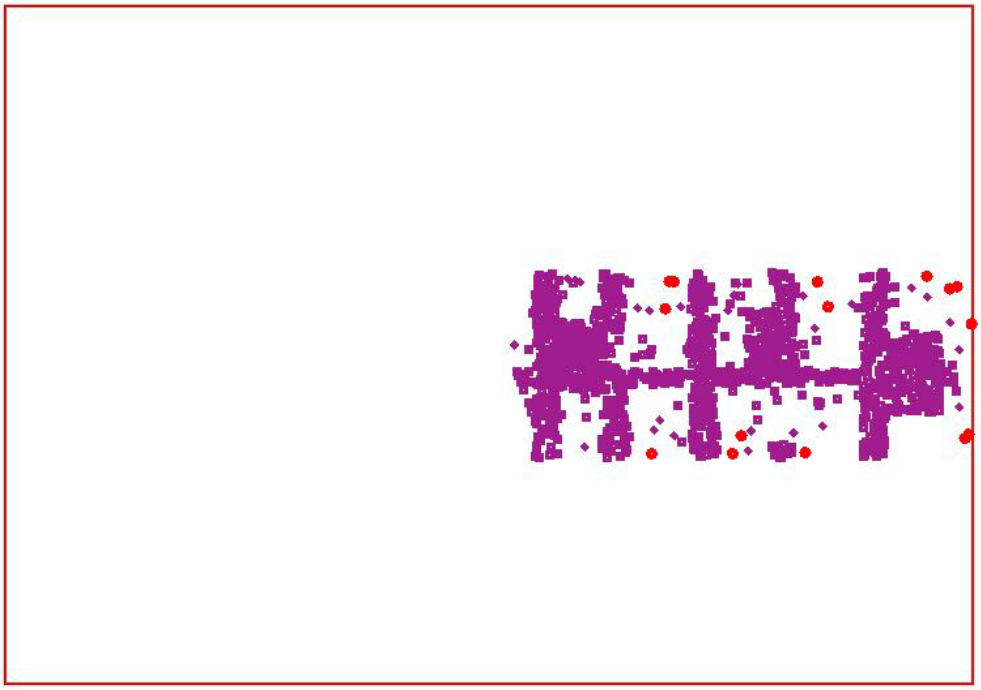
noktalar, birbirlerinin özelliklerini değiştiremeyeceğinden, bu ek işleme gerek kalmadan da doğrudan birleştirilebilir. Oluşan yeni bölümün sınırları ve DBSCAN ile kümelenmiş halleri Şekil 5.13 ve Şekil 5.14’ de görülmektedir.



Şekil 5.13 George kümesindeki ilk iki bölümün sınırları



Şekil 5.14 George kümesindeki oluşturulan yeni bölümün sınırları



Şekil 5.15 George veri kümesi yeni bölümün(birinci ve ikinci bölümler) kümelenmiş hali

Oluşan yeni bölüm, DBSCAN ile bölümlendirilmiştir(Şekil 5.15). Bu yeni bölüm, tek kümeden oluşmaktadır. Birleştirme işleminin yapılacağı bu noktada birinci(Şekil 5.11), yeni(Şekil 5.15) ve ikinci(Şekil 5.12) bölüm olmak üzere üç farklı bölüm bulunmaktadır. Oluşan yeni bölüm, hem birinci bölümden hem de ikinci bölümden noktalar taşımaktadır. Bu sayede iki bölümden biri referans alınarak sonraki bölüm ile birleştirilecektir. Referans(birinci) ve diğer(yeni) bölümde ortak olan noktalar incelenerek küme etiketleri değiştirilecektir. Örneğin 1000 numaralı nokta birinci bölümde A kümesinde ve yeni bölümde X kümesinde ise, 1000 numaralı noktanın küme etiketi A olarak kalacaktır. Bölüm içinde değişen bir küme etiketi, aynı bölümdeki aynı etikete sahip bütün noktaların küme etiketlerini değiştirecektir. Bu alt değişim işlemi, referans bölümdeki başka bir nokta ile çakışıyorsa, aynı değişim referans bölümünün küme etiketleri için de yapılacaktır. Bu işlem, ortak noktaların tamamı bitinceye kadar özyinelemeli olarak devam edecektir. İşlem sonunda referans ve yeni bölümün oluşturduğu bölüm elde edilecektir(Şekil 5.16).

Şekil 5.16 George veri kümesi birinci ve yeni bölümün kümelenmiş hali

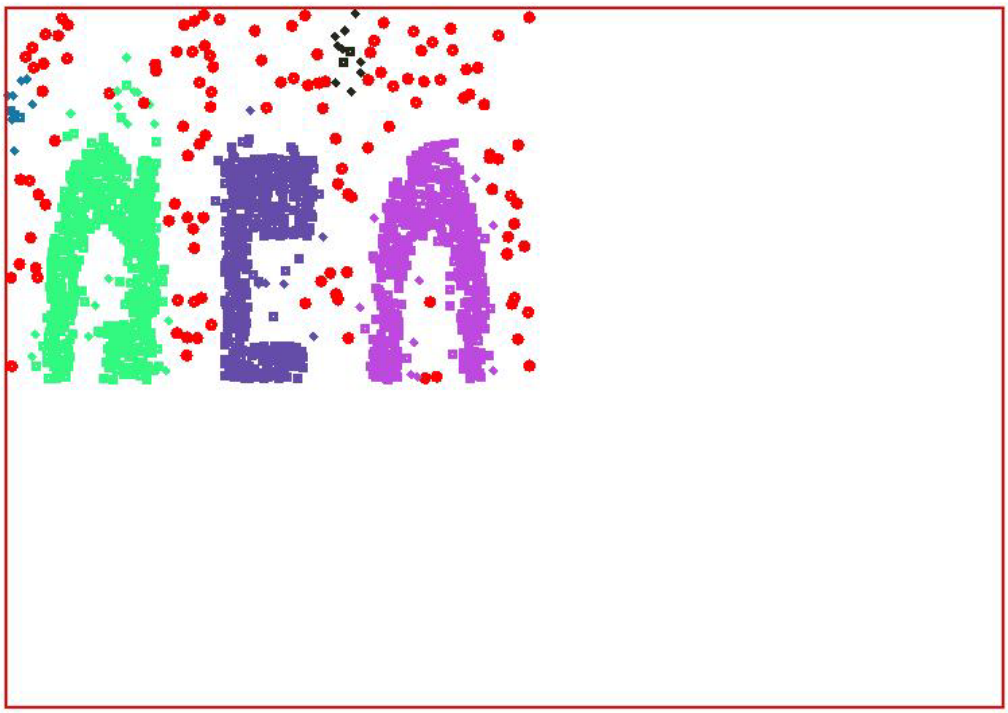
Oluşan bu bölüm, toplamda 2 farklı küme içermektedir. Birinci bölümde var olan 3 farklı küme, birleştirme işlemi sonucunda, teke indirilmiştir.

Yukarıda anlatılan yöntem, ikinci bölüme de uygulanmıştır. İşlem sonucunda, birinci ve ikinci

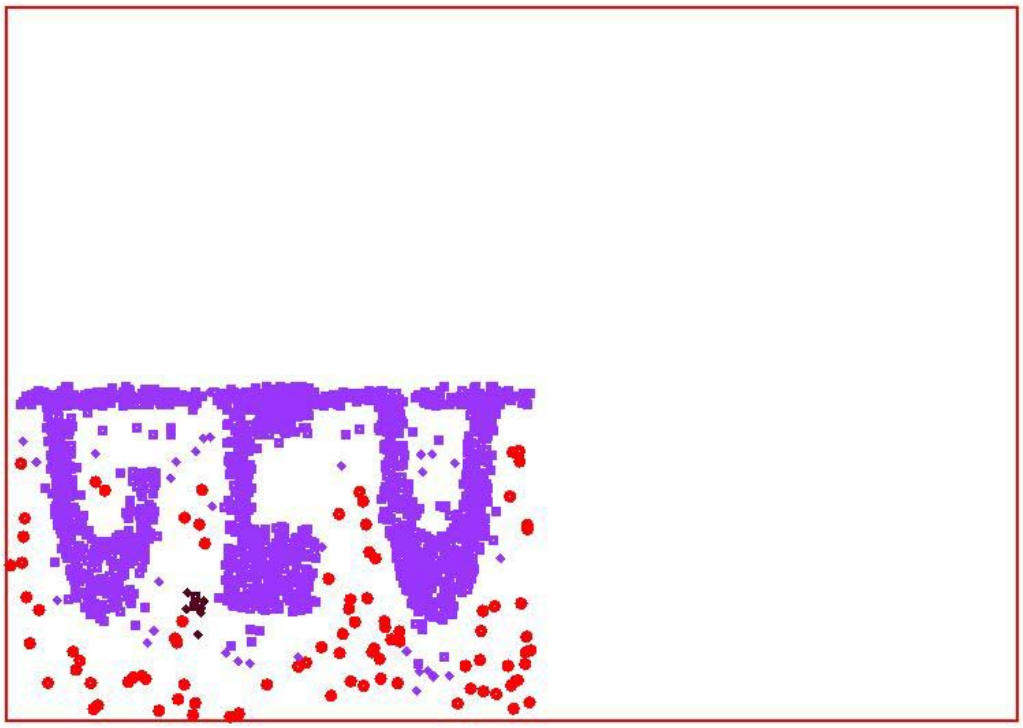
bölümler birleştirilerek tek bölüm haline getirilmiştir(Şekil 5.17).

Şekil 5.17 George veri kümesi birinci ve ikinci bölümün kümelenmiş hali

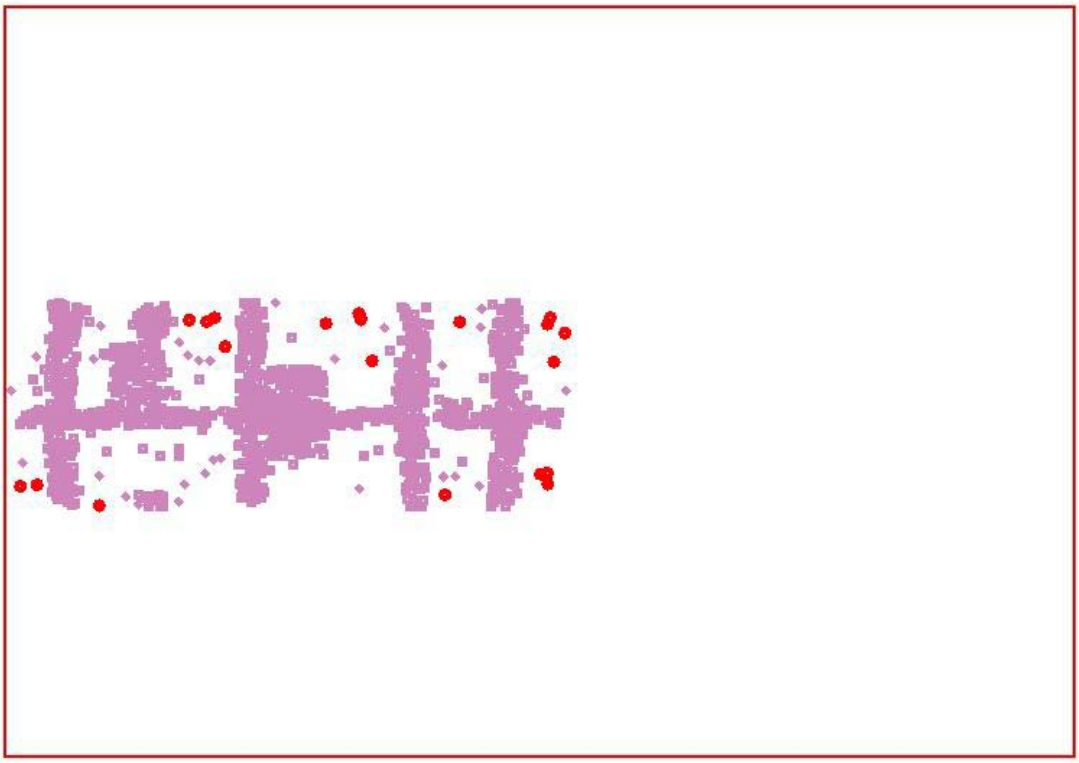
Birleştirme işlemi üçüncü ve dördüncü bölümler için de yapılmıştır(Şekil 5.18, Şekil 5.19, Şekil 5.20, Şekil 5.21 ve Şekil 5.22).



Şekil 5.18 George veri kümesi üçüncü bölümün kümelenmiş hali



Şekil 5.19 George veri kümesi dördüncü bölümün kümelenmiş hali

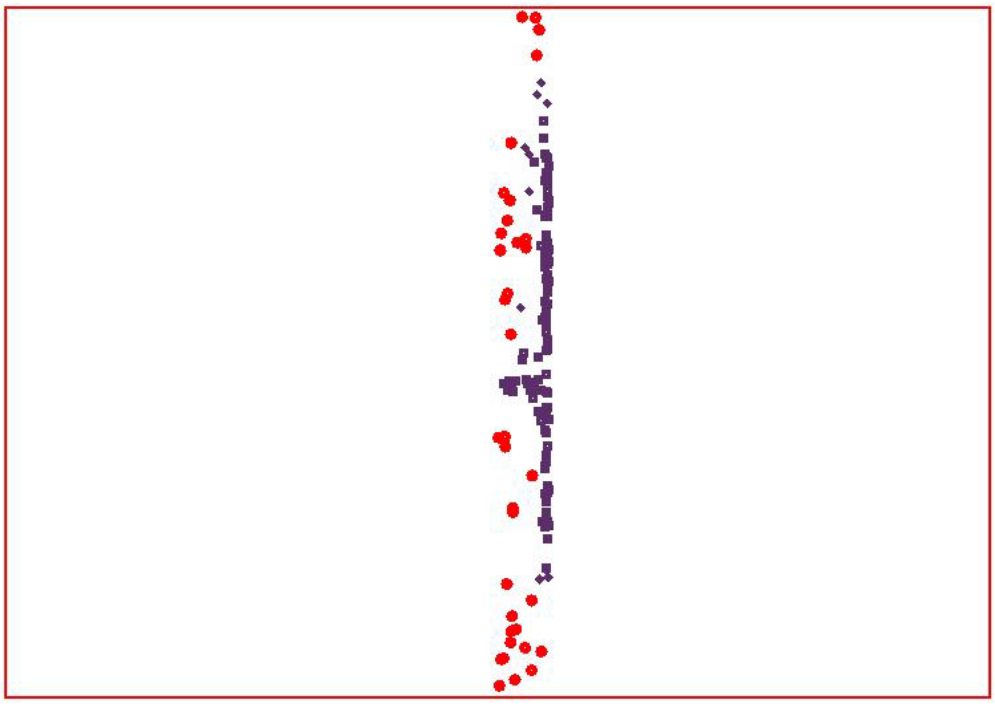


Şekil 5.20 George veri kümesi yeni bölümün(üç ve dördüncü bölümler) kümelenmiş hali

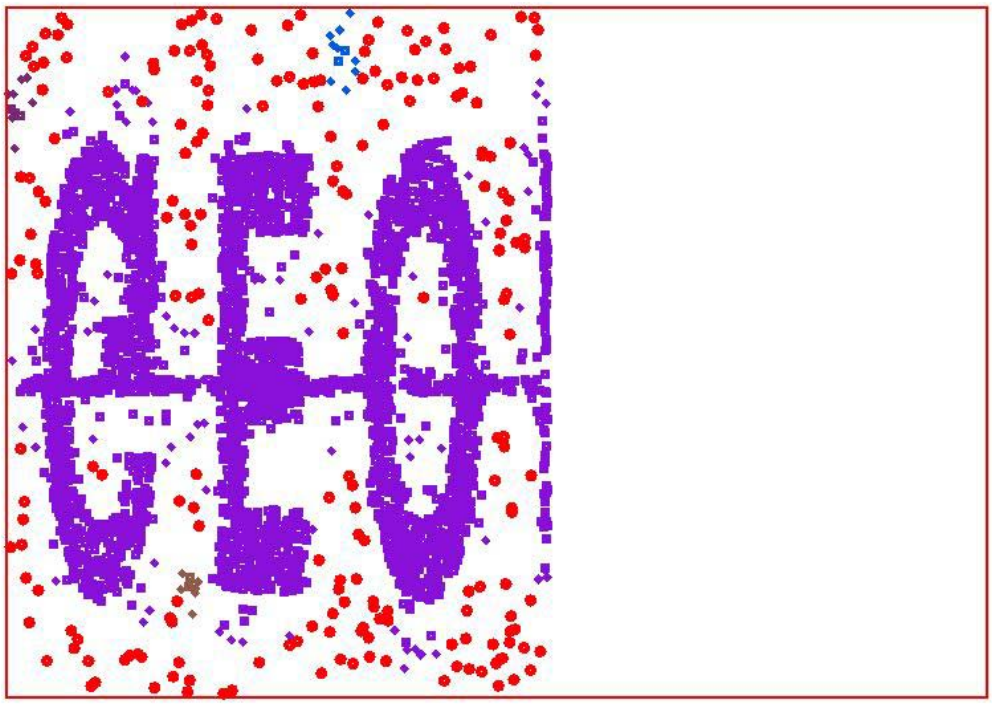
Şekil 5.21 George veri kümesi üçüncü ve yeni bölümün kümelenmiş hali

Şekil 5.22 George veri kümesi üçüncü ve dördüncü bölümün kümelenmiş hali

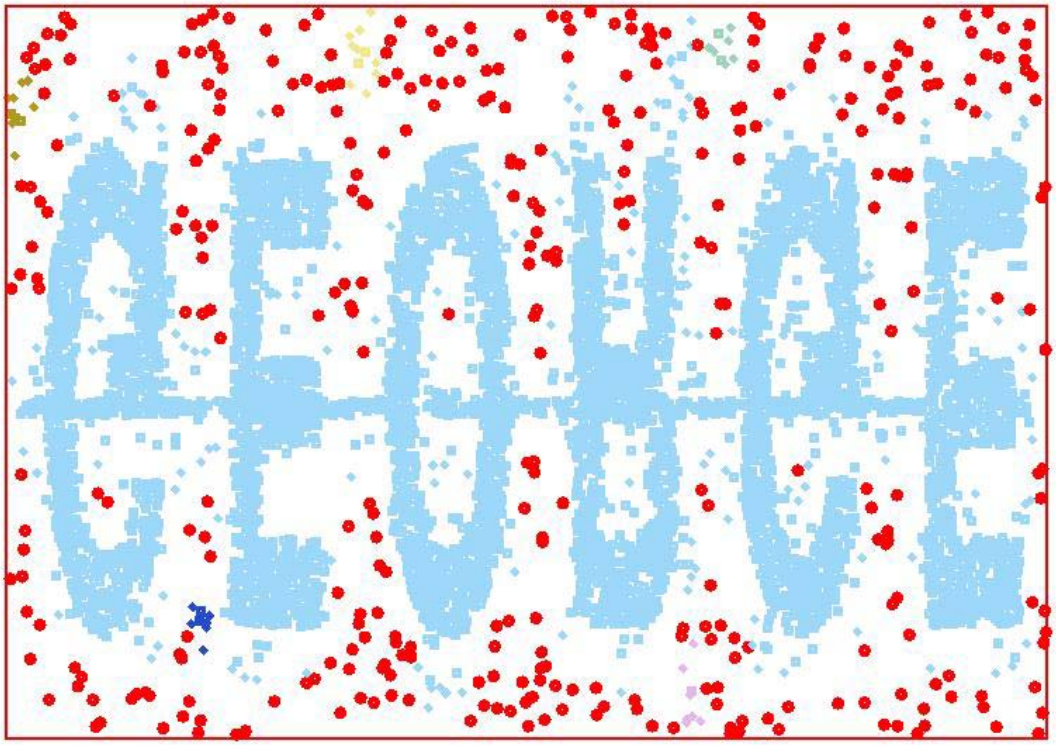
Son olarak oluşan iki büyük bölüm birleştirilerek tüm veri kümesi tamamlanmıştır(Şekil 5.23, Şekil 5.24 ve Şekil 5.25).



Şekil 5.23 George veri kümesi bir-iki ve üç-dört bölümlerin oluşturduğu yeni bölümün kümelenmiş hali



Şekil 5.24 George veri kümesi bir-iki ve yeni bölümlerin oluşturduğu bölümün kümelenmiş hali



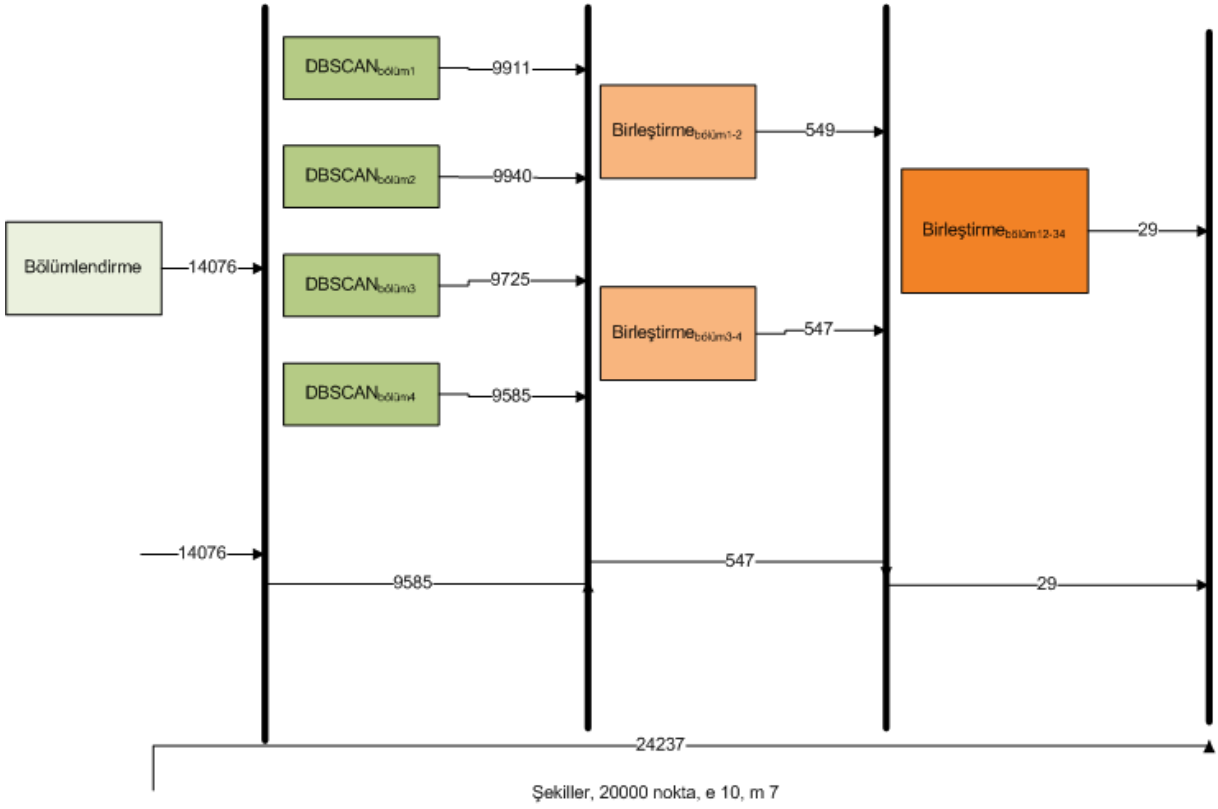
Şekil 5.25 George veri kümesi bir-iki ve üç-dört bölümlerin oluşturduğu ana bölümün kümenin kümelenmiş hali

Akış diyagramı, Şekil 5.3' de belirtilen algoritmanın birleştirme kısmı, aşağıda genel hatları belirtilen algoritma ile çalışır:

```
Partition Join(Partition[] _partitionList, double _epsilon, int _minimumNumberOfPoint){
  if (_partitionList.Length < 2){
    return _partitionList[0];
  }else if (_partitionList.Length == 2){
    Bound NewPartitionBound = GetInnerPartitionBound(_partitionList[0].PartitionBound,
_partitionList[1].PartitionBound, _epsilon);
    DBSCANPoint[] NewPoints = GetNewPoints[_partitionList[0].Points,
_partitionList[1].Points, NewPartitionBound];
    GenerateDBSCANClusters(NewPoints, _epsilon, _minimumNumberOfPoint);
    MergeReferenceAndOtherPoints(_partitionList[0].Points, NewPoints);
    MergeReferenceAndOtherPoints(NewPoints, _partitionList[1].Points);
    DBSCANPoint[] NewPartitionPoints = GetNewPartitionPoints(_partitionList[0].Points,
_partitionList[1].Points);
    return new Partition(NewPartitionPoints);
  }else{
    Partition[] NewPartitionListRight = GetPartitionList(_partitionList, 0,
_partitionList.Length / 2);
    Partition[] NewPartitionListLeft = GetPartitionList(_partitionList, _partitionList.Length
/ 2, _partitionList.Length);
    Partition JoinedPartitionRight = Join(NewPartitionListRight, _epsilon,
_minimumNumberOfPoint);
    Partition JoinedPartitionLeft = Join(NewPartitionListLeft, _epsilon,
_minimumNumberOfPoint);
    Partition[] NewPartitionList = new Partition[2];
    NewPartitionList[0] = JoinedPartitionRight;
    NewPartitionList[1] = JoinedPartitionLeft;
    return Join(NewPartitionList, _epsilon, _minimumNumberOfPoint);
    }
}
```

5.3 Önerilen Metodun Performans Analizi

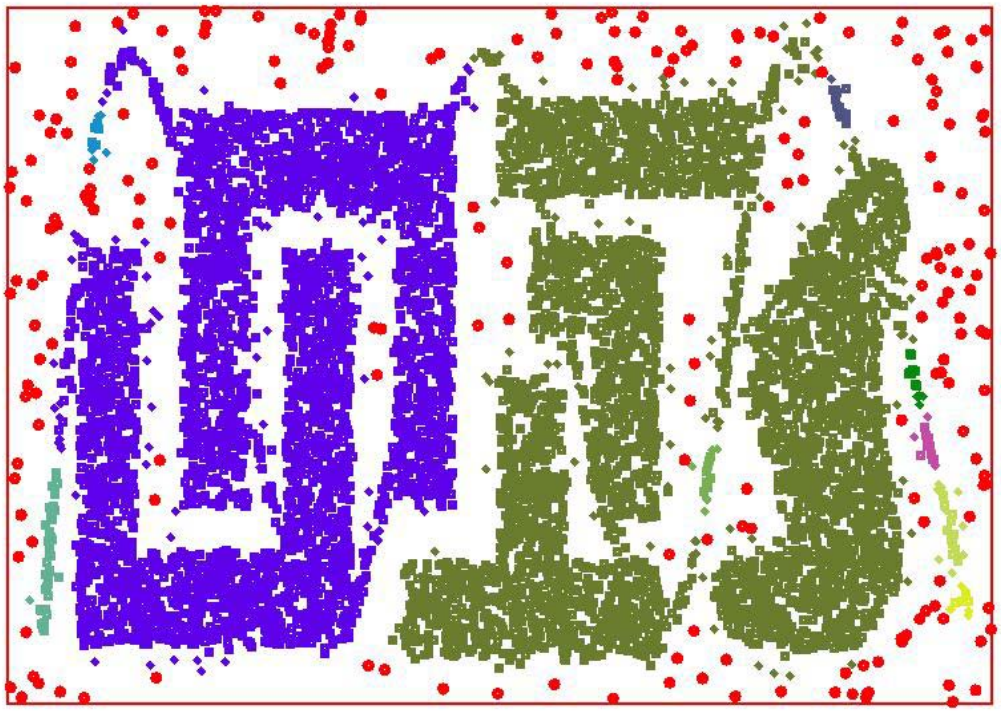
Öne sürülen algoritmanın çalışma süresi ile, orijinal algoritmanın çalışma sürelerinin analizi yapılmıştır. R-P-DBSCAN yöntemi, dağıtık sistemler üzerinde paralel proses olarak veya tekil proses olarak çalıştırılabilir. Paralel çalışma ile, DBSCAN algoritması ölçeklenebilir hale getirilir. Paralel çalışma ile bir iş, aynı anda birden fazla prosese gönderilir. İşleri dağıtan ve koordine eden iş yöneticisi de ayrı bir proses olarak düşünülebilir. Şekil 5.26' da, R-P-DBSCAN algoritmasının, 4 bölüm üzerinde ne kadar sürede çalıştığı belirtilmiştir. Dikkat edilecek olursa R-P-DBSCAN, paralel çalışmaya çok uygun bir yöntemdir. İş koordinatörü(master process) ile işçiler(slave processes) arasındaki iç mesajlaşmalar da(inter-communication) eklendiğinde, yöntemin net kazancı ortaya çıkacaktır. Paralel programlama, bu tezin kapsamına girmediğinden, detaylı analizleri yapılmamıştır. İlerleyen bölümlerde, tek proses olarak R-P-DBSCAN' nin performans analiz verileri sunulacaktır.



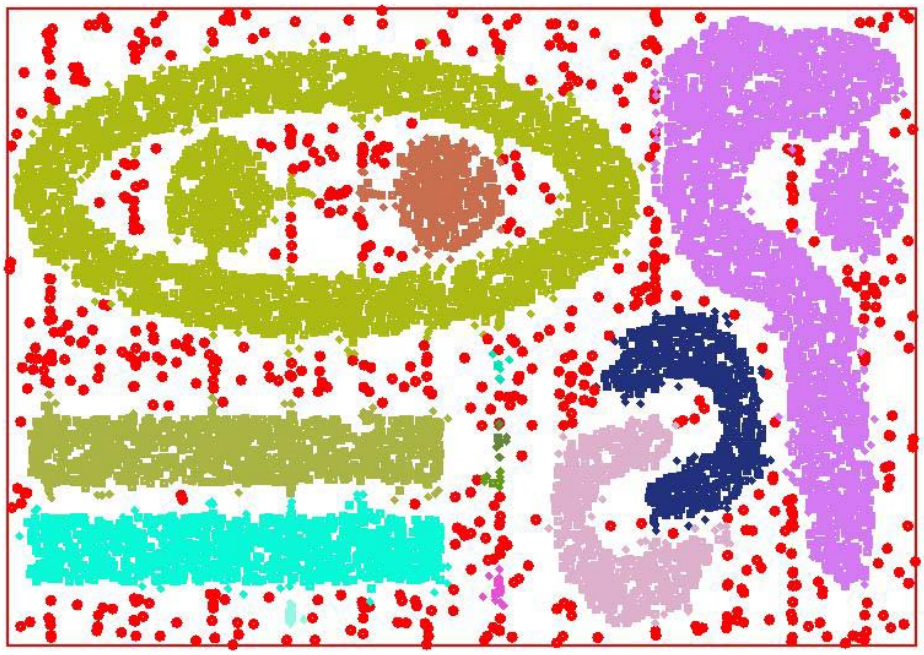
Şekil 5.26 32000 noktalı veri kümesinde 4 bölümlü R-P-DBSCAN performans analizi

Performans analizleri, bir başka yoğunluk tabanlı yöntem olan CHAMELON' un(Karypis vd., 1999) kullandığı veri kümeleri üzerinde yapılmıştır. Bu kümelerden üçü 8000, geri kalanı ise

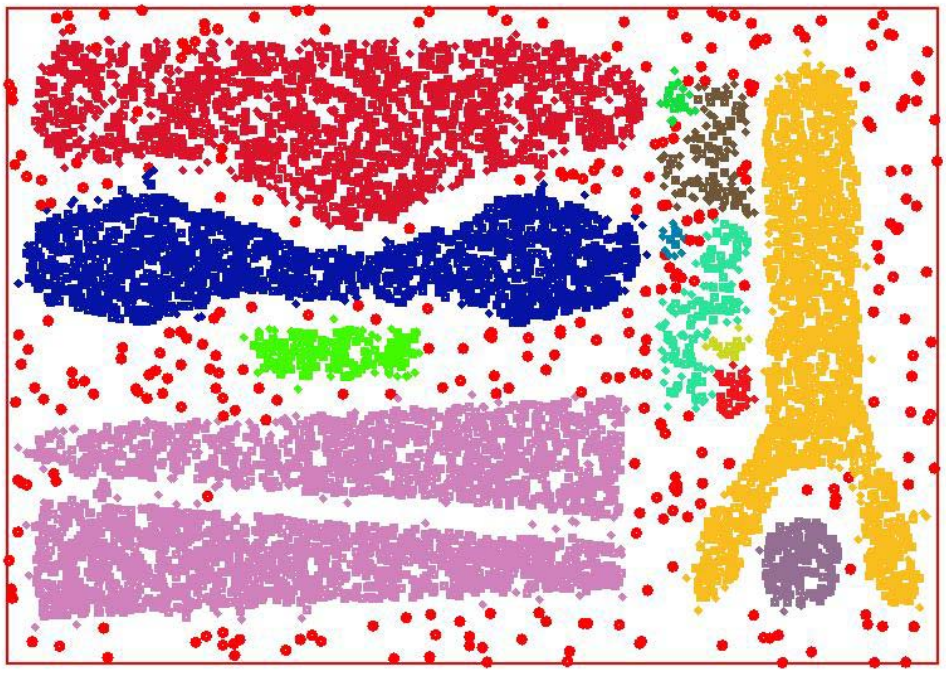
10000 noktadan oluşmaktadır. Bu veri setlerine, tez içinde daha anlamlı olması açısından isimler verilmiştir. Bu kümeler ve isimleri Şekil 5.27, Şekil 5.28, Şekil 5.29 ve Şekil 5.30’ da gösterilmiştir.



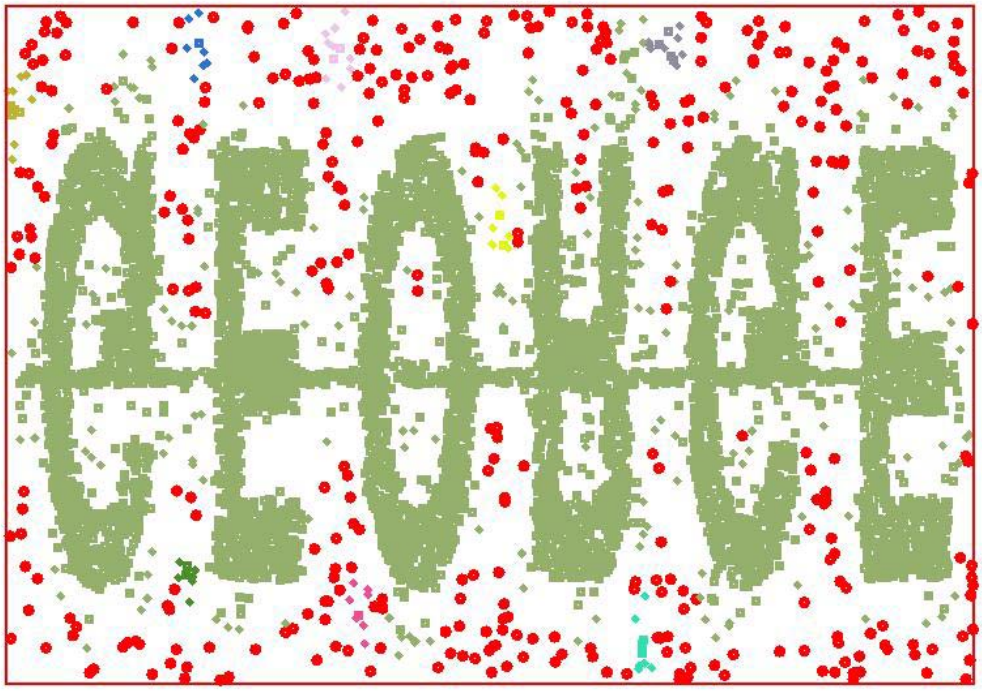
Şekil 5.27 Geçitler veri kümesi, 10000 nokta(e=10,m=7)



Şekil 5.28 Şekiller veri kümesi, 10000 nokta(e=10,m=7)

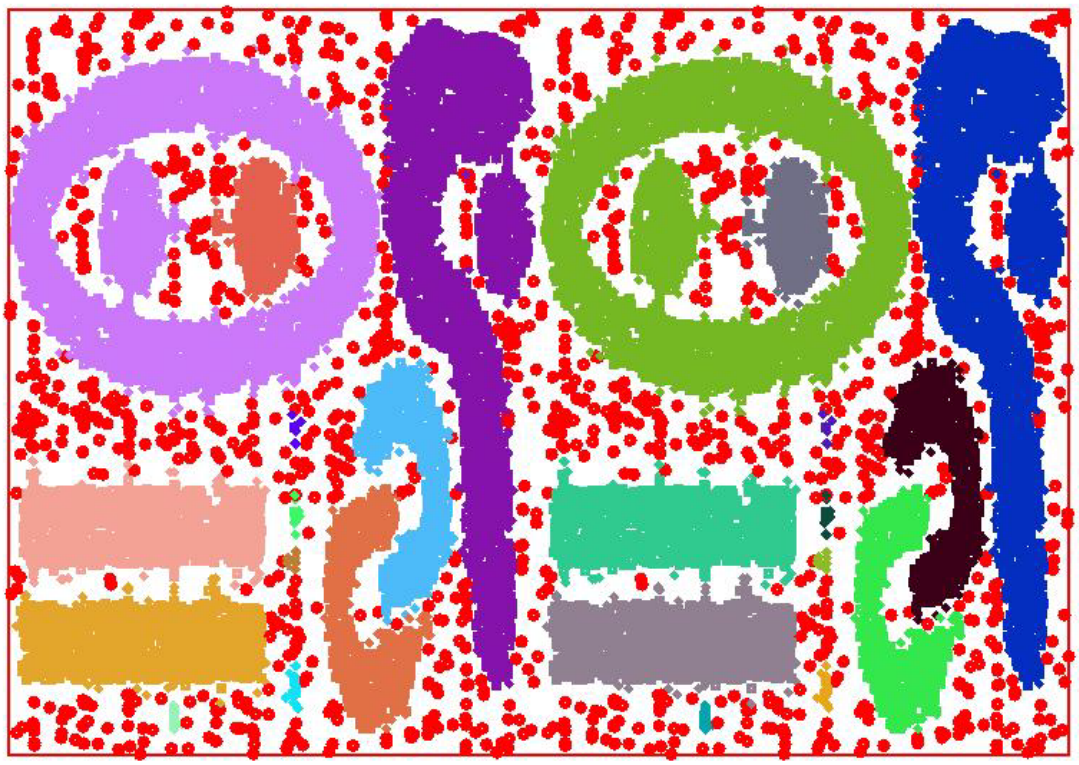


Şekil 5.29 Şeritler veri kümesi, 8000 nokta(e=10,m=7)

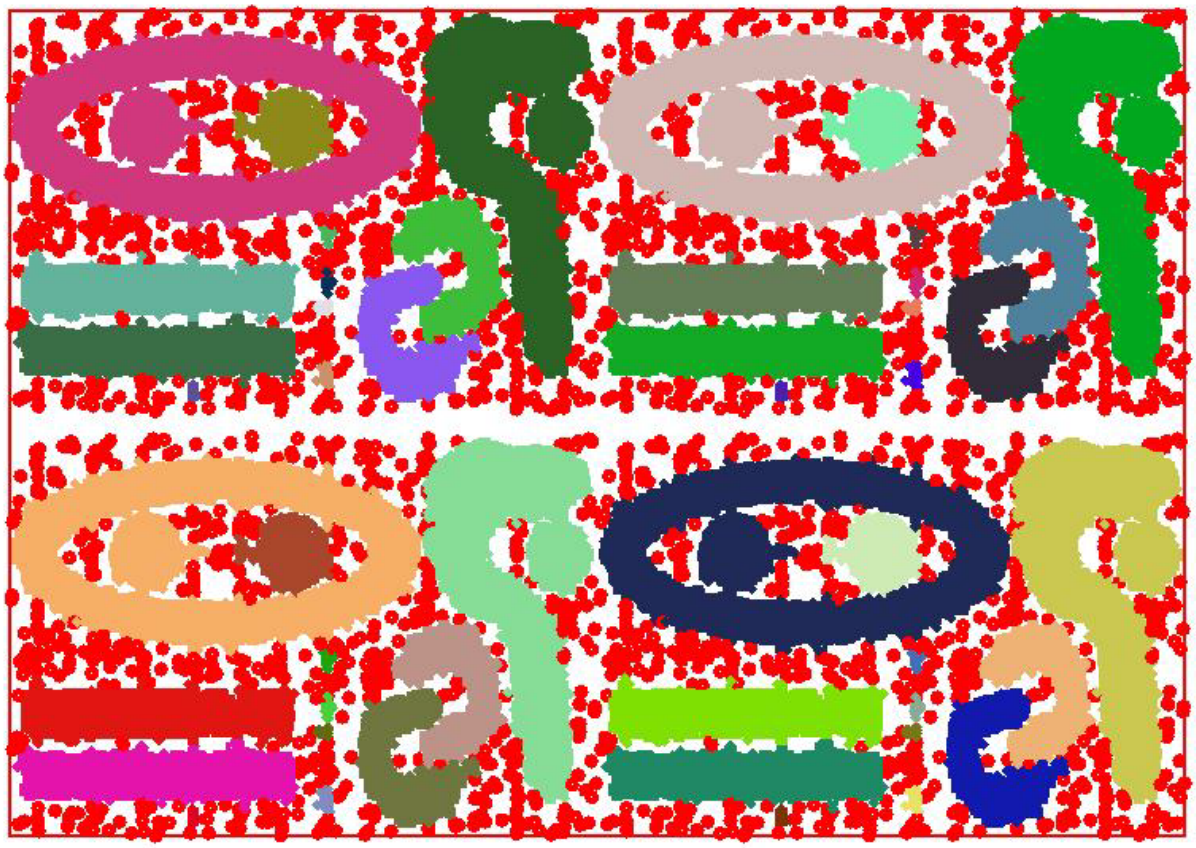


Şekil 5.30 George veri kümesi, 8000 nokta(e=10,m=7)

Performans analizlerinde kullanılmak üzere, nokta sayısı ile bölümlendirme sayısı arasındaki ilişkiyi bulabilmek adına, veri kümeleri zenginleştirilmiştir. Bu zenginleştirme işlemi, kümeleri yanyana ve üstüste kopyalamak suretiyle gerçekleştirilmiştir. Örnek olarak Şekiller veri kümesi 8000 noktadan oluşmaktadır. Şekil 5.31’ de bu kümenin 20000 noktalı, Şekil 5.32’ de ise aynı kümenin 40000 noktalı zenginleşmiş hali görülmektedir.

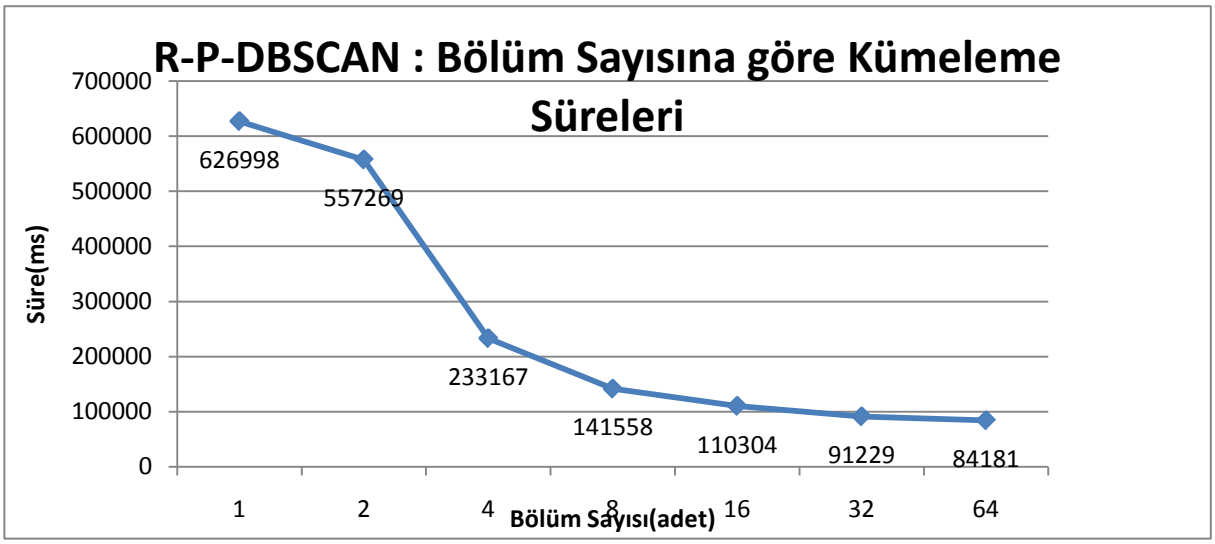


Şekil 5.31 Şekiller veri kümesi, 20000 nokta(e=10,m=7)

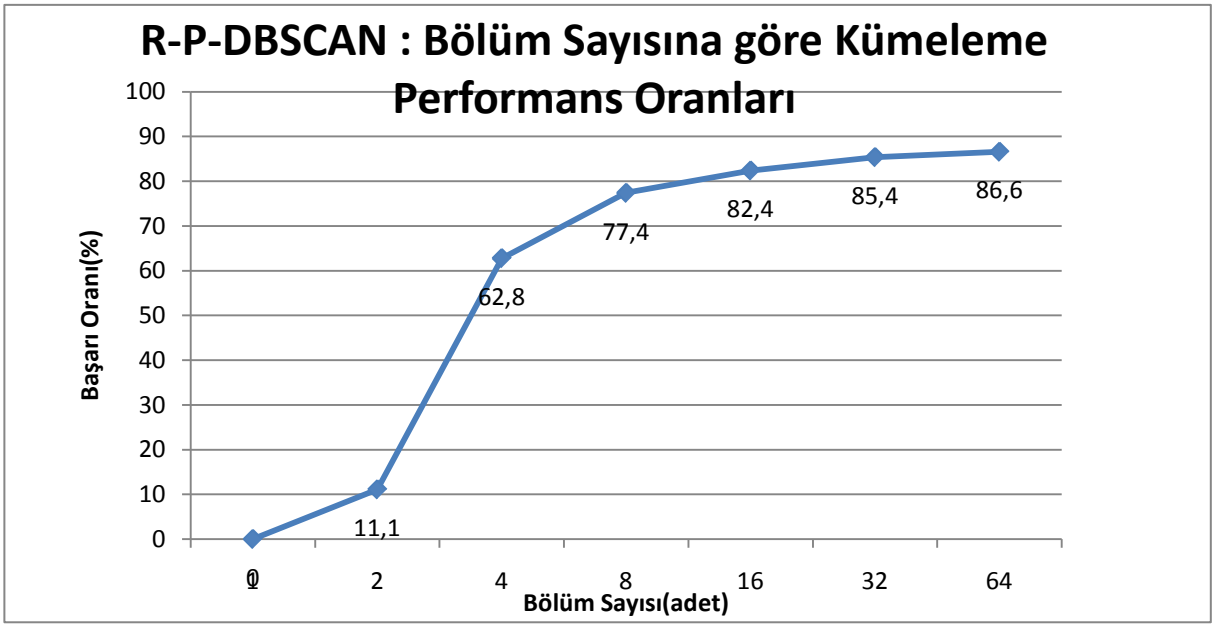


Şekil 5.32 Şekiller veri kümesi, 40000 nokta(e=10,m=7)

Şekil 5.33 ve Şekil 5.34’ de şekiller veri kümesinin 40000 noktalı haliyle yapılan performans grafiği verilmiştir. Grafiklerden de görüldüğü üzere, bölüm sayısı ile kümeleme süresi arasında üstel olarak ters orantı bulunmaktadır. 600 saniyelerde olan kümeleme süresi(1 bölüm, klasik DBSCAN algoritması), 100 saniyelerin altına inmiştir.(16 bölüm için % 85,4 başarı) Yine grafikten anlaşılacağı üzere, bölüm sayısının artması belirli bir değerden sonra(8), sonuca çok fazla etki etmemektedir. Farklı veri kümeleri ile yapılan testler sonucunda, ideal bölüm sayısının 8 veya 16 olduğu görülmüştür. Veri kümesi büyüdükçe, bölüm sayısının da arttırılması, performans için olumlu olacaktır.

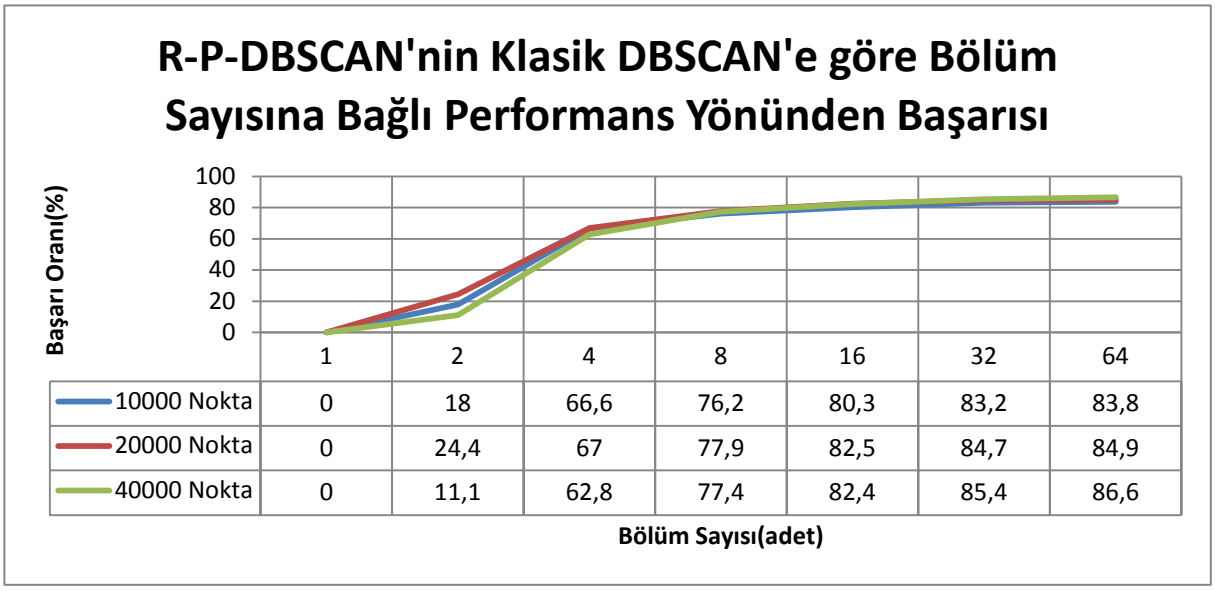


Şekil 5.33 R-P-DBSCAN: Bölüm Sayısına göre kümeleme süreleri(Şekiller,40000)



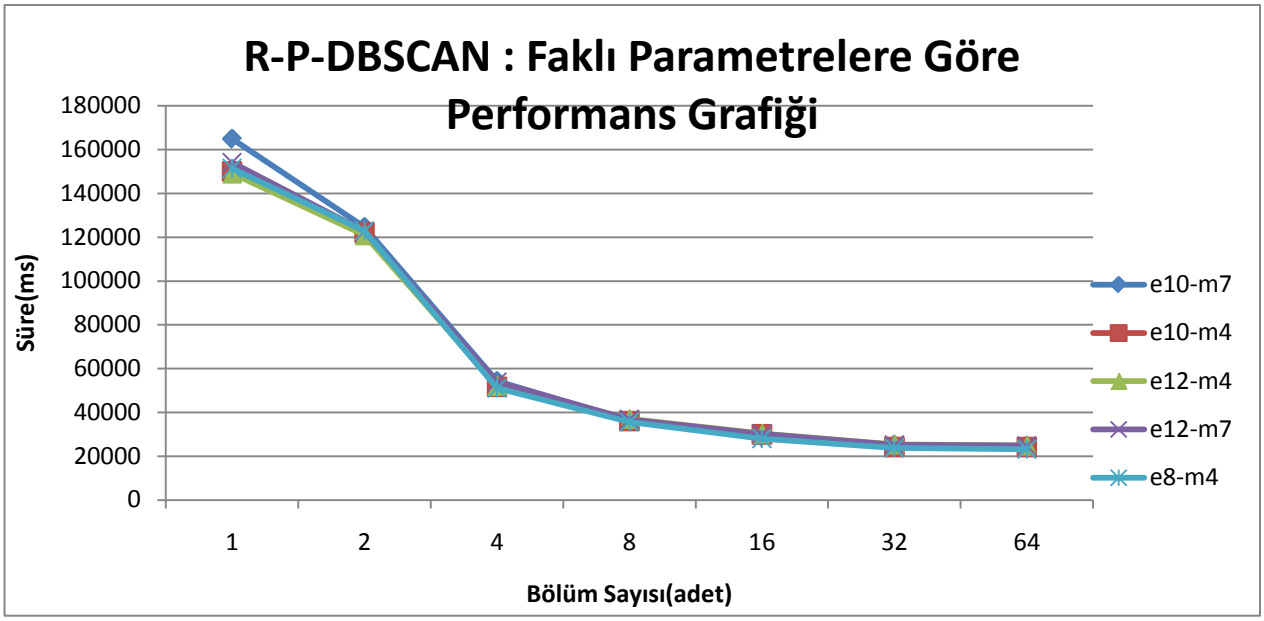
Şekil 5.34 R-P-DBSCAN: Bölüm sayısına göre kümeleme performans oranları (Şekiller,40000)

Şekil 5.35' de yer alan şekiller veri kümesinde, nokta sayısına bağlı performans grafiği bulunmaktadır. 10000, 20000 ve 40000 noktalı kümelerde 1, 2, 4, 8, 16, 32 ve 64 bölüm olacak şekilde R-P-DBSCAN çalıştırılmış, sonuçlar çıkarılmıştır. R-P-DBSCAN, her ne kadar büyük veri kümelerinde, küçük olanlara göre daha verimli(% 2 - 3) çalıştığı görülse de, veri kümesinin büyüklüğünden bağımsız bir başarı sağladığı görülmüştür.



Şekil 5.35 R-P-DBSCAN' nin klasik DBSCAN' e göre bölüm sayısına bağlı performans yönünden başarısı(Şekiller)

Şekil 5.36’ da ise, farklı DBSCAN parametrelerine göre(Çizelge 5.2) performans eğrileri bulunmaktadır. Parametrelerin farklı olmasının, kümeleme performansına pek de etki etmediği görülmüştür.

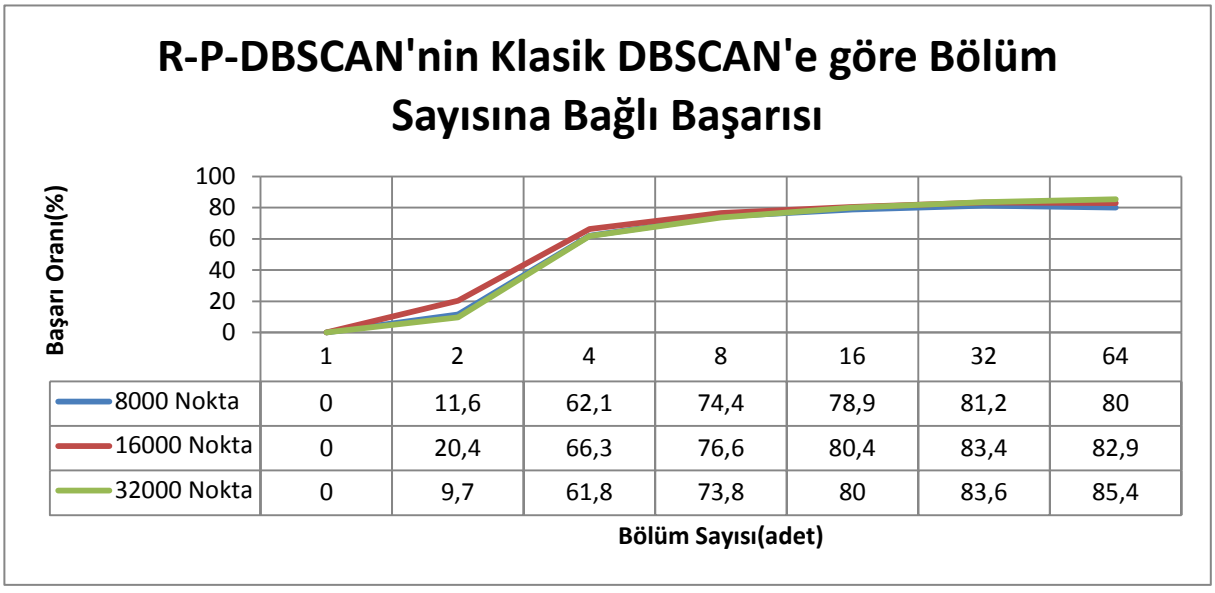


Şekil 5.36 R-P-DBSCAN: Faklı parametrelere göre performans grafiği(Şekiller,20000)

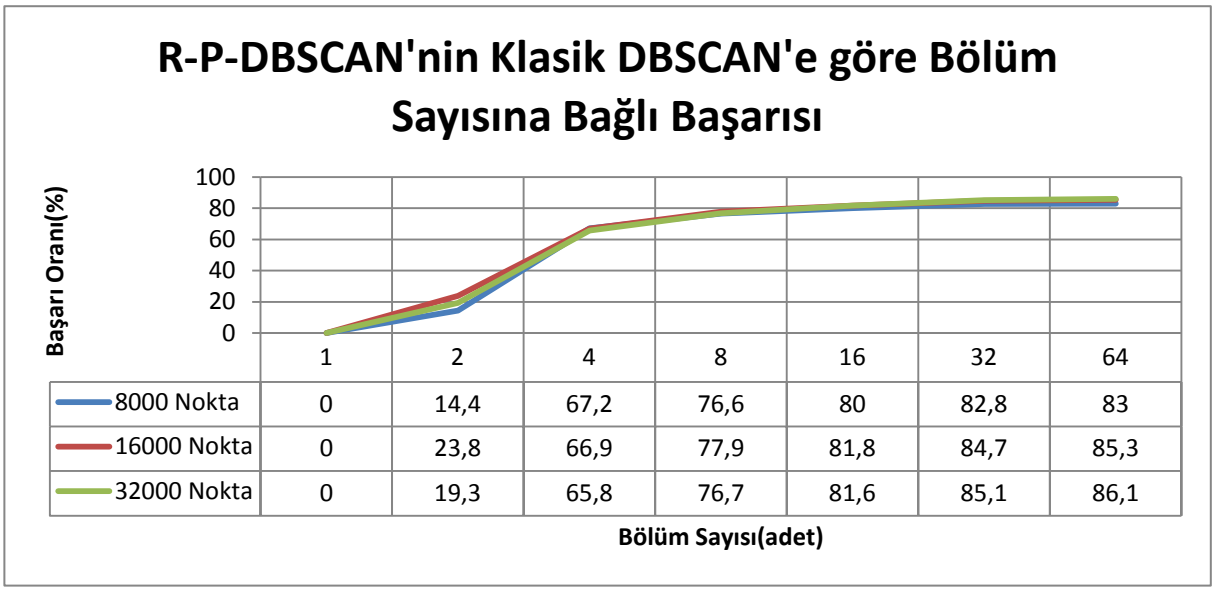
Çizelge 5.1 R-P-DBSCAN: Faklı parametrelere göre performans grafiği detayları (Şekiller,20000)

Epsilon	MinPts	Küme sayısı	Gürültülü Nokta Sayısı	Köşe Nokta Sayısı
10	7	25	1228	584
12	7	25	520	520
8	4	69	1112	516
10	4	67	748	376
12	4	47	410	354

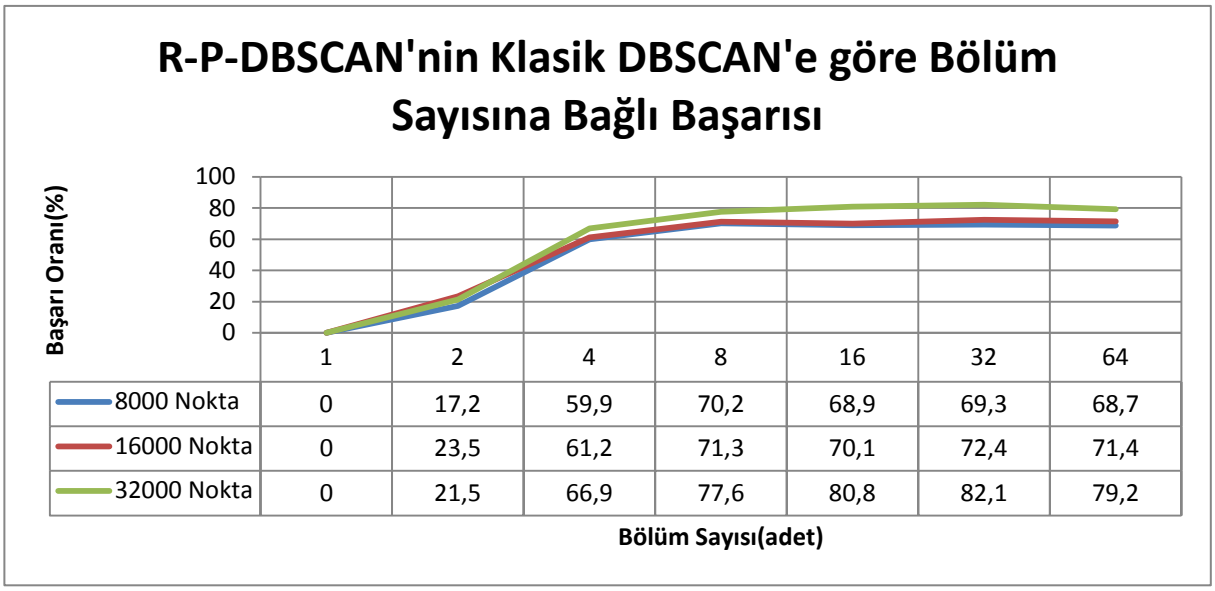
Diğer veri kümelerine yönelik grafikler, aşağıda belirtilmiştir(Şekil 5.37, Şekil 5.38 ve Şekil 5.39).



Şekil 5.37 R-P-DBSCAN' nin klasik DBSCAN' e göre bölüm sayısına bağlı performans yönünden başarısı(Geçitler)



Şekil 5.38 R-P-DBSCAN' nin klasik DBSCAN' e göre bölüm sayısına bağlı performans yönünden başarısı(Şeritler)



Şekil 5.39 R-P-DBSCAN' nin klasik DBSCAN' e göre bölüm sayısına bağlı performans yönünden başarısı(George)

Çizelge 5.2'de ise farklı veri kümelerinin farklı boyutlarıyla yapılan kümeleme işleminde R-P-DBSCAN'nin detaylı performans değerleri bulunmaktadır. Bu çizelgeden de anlaşılacağı üzere, epsilon ve MinPts parametrelerine bakmaksızın önerilen yöntem başarı göstermektedir.

Çizelge 5.2 R-P-DBSCAN' nin detaylı performans değerleri

Veri Kümesi	Nokta Sayısı	MinPts	Epsilon	Bölüm Sayısı	Bölümleme	Max (DBSCAN)	Toplam (DBSCAN)	Max (Birleştirme)	Toplam (Birleştirme)	Toplam

Şekiller	10000	4	10	8	4229	679	5310	175	1023	10278
				16	4555	172	2671	175	739	8249
				32	4452	46	1363	173	1280	7095
				64	4568	15	698	172	1687	6953
Şekiller	40000	7	10	8	64609	9652	76390	106	559	141558
				16	70833	2426	37787	160	1684	110304
				32	69376	621	19010	165	2843	91229
				64	69753	197	10352	156	4076	84181
George	32000	7	10	8	40903	6193	48850	94	413	90166
				16	44987	1554	24243	968	7862	77092
				32	49020	448	13747	1072	9020	71787
				64	57043	235	9674	1234	16971	83688

5.4 Diğer DBSCAN Türevleriyle Uyumu

DBSCAN yöntemi, özellikle son 5 yıl içinde çokça geliştirilmiştir. Her biri DBSCAN üzerinde bir iyileştirme olan türevler, R-P-DBSCAN ile de çalışabilmektedir. Yapılması gereken tek şey, yeni yöntemde, DBSCAN metodu yerine, türev yöntemi metodunun çağrılması işlemi olacaktır.

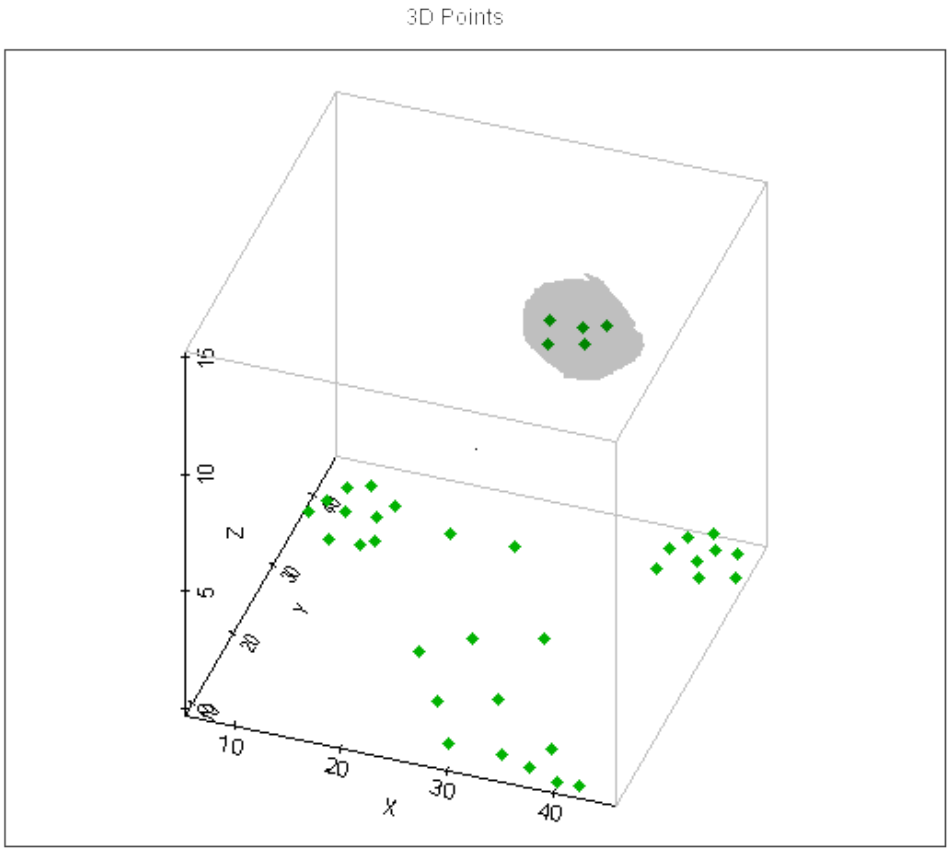
5.5 Çoklu Düzlemlerde(3 ve üzeri) R-P-DBSCAN

R-P-DBSCAN, sadece 2 boyutlu uzaylarda çalışabilen bir yöntem değildir. Arka planda DBSCAN algoritmasını kullandığı için, DBSCAN' nin çalışabildiği tüm çok boyutlu uzaylarda çalışabilmektedir. Boyut sayısının artması, bölümlemelerin de o boyutlar üzerinde yapılmasını gerektirir. Bu noktada R-P-DBSCAN, belirlenen düzlemlere göre bölümlendirme işlemini yapar. Birleştirme işlemi de aynı sıra ile yapılır.

3 boyutlu düzlemde bölümlendirme, X, Y, Z eksenlerine göre sırayla yapılır. Örneğin 3 boyutlu veri kümesi 8 bölüme ayrılmak istenirse, önce X eksenine dik iki bölüme; sonrasında her bir bölüm Y eksenine dik iki bölüme; en sonunda her bir bölüm(toplamda dört bölüm) ise Z eksenine dik iki bölüm olmak üzere toplamda 8 parçaya ayrılır. 8 bölüme ayırma işlemi 2 boyutlu uzayda yapılsaydı, Z ekseni olmadığından, X, Y, X eksenleri sırasında yapılacaktı.

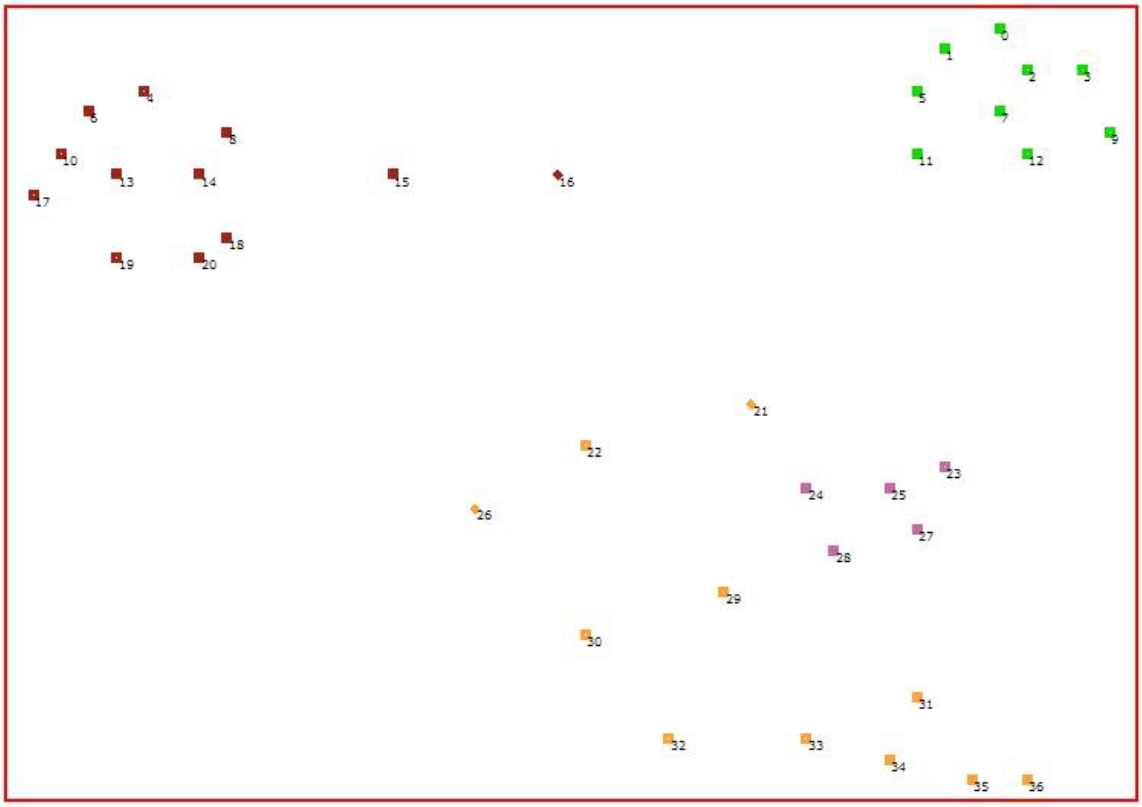
3 boyutlu veri kümesinde, belirtilen algoritmanın nasıl çalıştığını göstermek için 37 noktalı bir veri kümesi ele alınacaktır(Şekil 5.1). Bu küme, normalde 2 boyutlu bir küme iken Z koordinatları eklenerek 3 boyutlu hale getirlmiştir. Tüm noktaların Z koordinatları 0 olarak

işaretlenmiştir. 3 boyutlu düzlemde kümelemeyi gösterebilmek adına 5 noktanın(23, 24, 25, 27 ve 28) Z koordinatı 15 olarak ayarlanmıştır. Şekil 5.40' da, bu kümenin 3 boyutlu düzlemdeki gösterilişi bulunmaktadır. Gri çember içine alınan 5 nokta, Z eksenindeki değeri 15 olan noktaları göstermektedir.



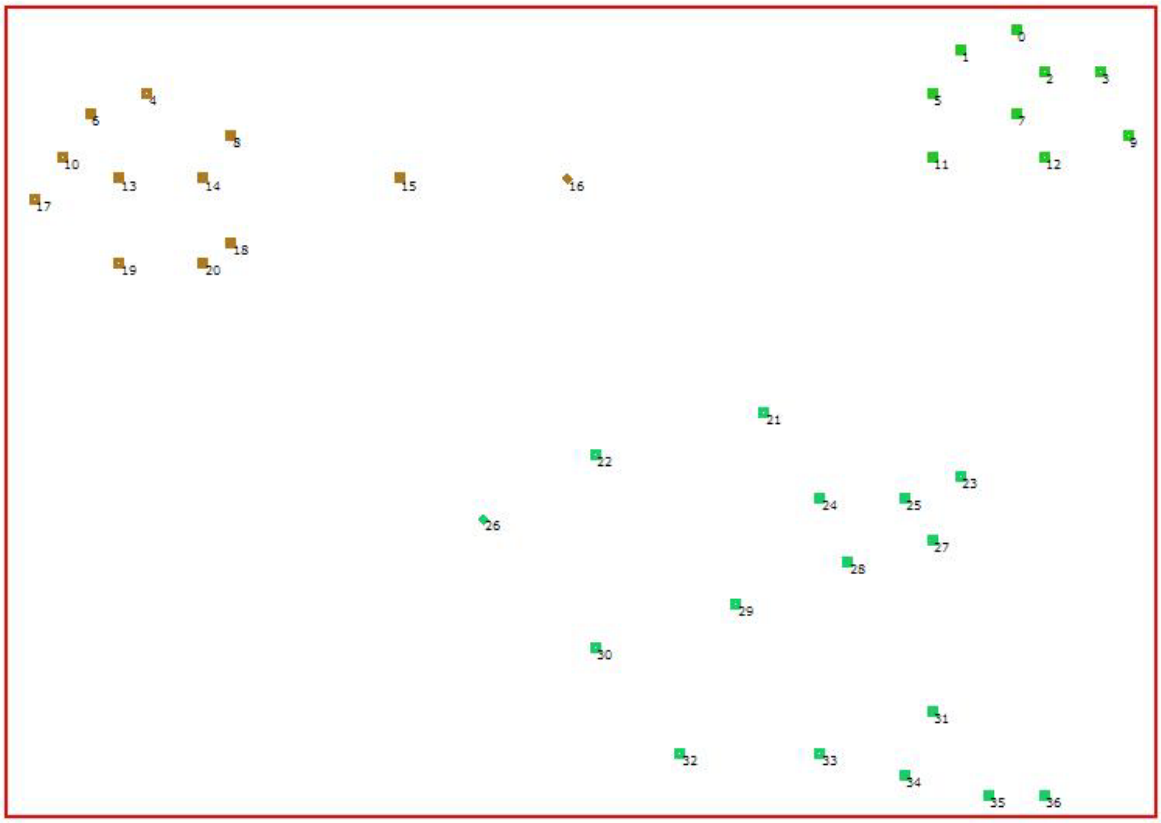
Şekil 5.40 Örnek veri seti(3 boyuta yükseltgenmiş)

R-P-DBSCAN bu veri kümesi üzerinde 8 bölüm olacak şekilde çalıştırıldığında, önce X eksenine dik iki bölüme; sonrasında her bir bölüm Y eksenine dik iki bölüme; en sonunda her bir bölüm(toplamda dört bölüm) ise Z eksenine dik iki bölüme ayrılıp, DBSCAN algoritmasıyla kümelenecektir. Birleştirme işlemi de aynı sırada yapıldığında, Şekil 5.41' deki kümelenmiş veri kümesine ulaşılacaktır.



Şekil 5.41 3 boyutlu örnek veri kümesi (DBSCAN, e=10, m=4)

3 boyutlu veri kümesinin, 2 boyutlu kümelenmiş hali(Şekil 5.42) incelendiğinde, Z koordinatı 15 olan noktaların farklı bir küme ile kümelendiği görülecektir.



Şekil 5.42 2 boyutlu örnek veri kümesi (DBSCAN, e=10, m=4)

R-P-DBSCAN yönteminde, noktaların 2 veya 3 boyutlu olmasının bir önemi yoktur. 2 boyutlu bir veri kümesi üçüncü defa bölündüğünde X eksenine göre tekrar bölümlenirken, 3 boyutlu bir veri kümesi için bölümleme Z eksenine göre yapılacaktır. Bu bağlamda, 2 veya 3 boyutun R-P-DBSCAN yöntemi için herhangi bir farkının olmadığını söyleyebiliriz.

5.6 R-P-DBSCAN Yönteminin Kusurları

Klasik DBSCAN' de var olan köşe noktalar problemi, R-P-DBSCAN' de de bulunmaktadır. Bir nokta, her iki kümenin de köşesi olabilecek durumdaysa, ilk kümelenen küme içine dahil edilmektedir. R-P-DBSCAN ve DBSCAN yöntemleriyle kümelenen aynı veri kümesi karşılaştırıldığında, köşe noktalardan dolayı, bazı noktaların farklı kümelendiği gözlemlenmiştir. Bu durum bir kusur olarak adlandırılamaz; çünkü aynı noktaların farklı sıralarda DBSCAN algoritmasıyla kümelenmesiyle de, aynı durum ortaya çıkabilmektedir.

6 SONUÇLAR ve ÖNERİLER

R-P-DBSCAN, klasik DBSCAN yöntemine getirdiği geliştirmeyle kümeleme performansında ciddi artışlar sağladığı görülmüştür. %85' e varan performans iyileştirmesi Şekil 5.34, Şekil 5.37, Şekil 5.38 ve Şekil 5.39' daki grafiklerde belirtilmiştir. 4 farklı veri kümesinin farklı hacimlerindeki örnekleriyle yapılan testler, bu iyileştirmeyi göstermektedir. DBSCAN algoritmasının büyük veri kümelerindeki, kümeleme performansı açısından başarısızlığı, öne sürülen algoritmayla iyileştirilmiştir. R-P-DBSCAN ile, veri kümesinin büyüklüğüne bağlı olarak, istenen sayıda bölüm yapılabilmesi de DBSCAN algoritmasını ölçeklenebilinir hale getirmiştir.

Sadece 2 boyutlu düzlemlerde değil, çoklu düzlemlerde de R-P-DBSCAN çalışabilmektedir. Şekil 5.41' da gösterildiği üzere, 3 boyutlu düzlemlerde de başarılı bir şekilde kümeleme işlemini yapmıştır.

DBSCAN algoritması, çıkışından bugüne kadar çok defa iyileştirilmiştir. R-P-DBSCAN, öne sürülen DBSCAN türevleriyle uyumlu bir şekilde çalışabilmektedir.

R-P-DBSCAN, Şekil 5.26' da görüldüğü üzere paralel işlemeye uygun bir algoritma olduğundan, gelecek çalışmalarda bölümleme ve birleştirme işlemlerini ayrı işlemciler üzerinde yapmak, performansını daha da iyi hale getirecektir. Dağıtık sistemlerde üzerinde çalışabilen R-P-DBSCAN, bu özelliğiyle de kümeleme yöntemleri içinde ayrı bir yer edinecektir.

KAYNAKLAR

Alpaydın, E.(2000), "ZEKİ VERİ MADENCİLİĞİ: Ham Veriden Altın Bilgiye Ulaşma Yöntemleri", Bilişim 2000 Eğitim Semineri, İstanbul.

Andreopoulos, B., An, A., Wang, X. ve Labudde, D.(2008), "Efficient Layered Density-based Clustering of Categorical Data."

Ankerst, M., Breunig, M., Kriegel, H. P. ve Sander, J.(1999), "OPTICS: Ordering Points to Identify the Clustering Structure". SIGMOD.

Argüden, Y. ve Erşahin B. (2008), Veri Madenciliği, ARGE Danışmanlık, İstanbul

Breunig, M., Kriegel, H. P., Raymond, T. N. ve Sander, J.(200), "LOF: Identifying density-based Local Outliers", ACM SIGMOD Konferansı, Dalles.

Bolton, R.J. ve Hand, D.J.(2002), "Statistical fraud detection: A review", Statistical

Science, 17(3):235–255.

Borah, B., ve Bhattacharyya, D.K.(2004), "An Improved Sampling-Based DBSCAN for Large Spatial Databases", Proceedings of International Conference on Intelligent Sensing and Information, 92-96.

Borah, B., ve Bhattacharyya, D.K.(2008), "DDSC: A Density Differentiated Spatial Clustering Technique", Journal of Computers, 3:72-79.

Bozkurt, N. (2002), "İşletme Çalışanları Tarafından Yapılan Hileleri Doğuran Nedenler", Yaklaşım Dergisi, Sayı: 92.

Brause, R., Langsdorf, T. ve Hepp M.(1999), "Neural Data Mining for Credit Card Fraud Detection".

Cahill, M., Lambert, D., Pingeiro, J. ve Sun, D.(2002), "Detecting Fraud In The Real World", The Handbook of Massive datasets, Kluwer Academic Publishers,911-930.

Chan, P., Fan, W., Prodromidis, A. L. ve Stolfo, S. J.(1999), "Distributed Data Mining in Credit Card Fraud Detection".

Dempster, A., Laird, A. ve Rubin, D.(1977), "Maximum likelihood fromincomplete data via the EM algorithm", J. Royal Statistical Society, 39:1–38.

Ester, M., Kriegel, H. P., Sander, J. ve Xu, X. (1996), “A Density Based Algorithm for Discovering Clusters in Large Spatial Data Sets with Noise”, 2nd International Conference on Knowledge Discovery and Data Mining, 226–231.

Fahim, A. M., Saake, G., Salem, A. M., Torkey, F. A. ve Ramadan, M. A.(2008), “DCBOR: A Density Clustering Based on Outlier Removal.”, Proceedings Of World Academy Of Science, Engineering And Technology, 35:171-176.

Fisher, D.(1987), “Improving inference through conceptual clustering”, Artificial Intelligence (AAAI’87), 461–465.

Grambeier, J. ve Rudolph, A., (2002), “Techniques of Cluster Algorithms in Data Mining”, Data Mining and Knowledge Discovery, 6: 303-360.

Guha, S., Rastogi, R. ve Shim, K.(2000), “ROCK: A Robust Clustering Algorithm for Categorical Attributes.”, Information Systems, 25(5): 345-366.

Han, J. ve Kamber, M.(2006), Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers.

Hawkins, D.(1990), Identification of Outliers, Chapman and Hall, London.

Hinneburg, A., Hinneburg, E. ve Keim, D. A.(1998), “An Efficient Approach to Clustering in Large Multimedia Databases with Noise,”, Knowledge Discovery and Data Mining, 58-65.

Hodge, V. J. ve Austin, J.(2004), “A Survey of Outlier Detection Methodologies”, Artificial Intelligence Review, 22:85-126, Hollanda.

Juszczak, P., Adams, N.M., Hand, D.J., Whitrow, C. ve Weston, D.J.(2008), “Off-the-peg or bespoke classifiers for fraud detection?”, Computational Statistics and Data Analysis.

Karypis, G., Han, E. H. ve Kumar, V.(1999), “CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling”, Computer, 32:68-75.

Kaufman, L. ve Rousseeuw, P. J. (1990), Finding Groups inData: An Introduction to Cluster Analysis, JohnWiley & Sons.

Kohonen, T.(1982), Self-organized formation of topologically correct feature maps, Biological Cybernetics, 43:59–69.

Kohonen, T.(1989), Self-Organization and Associative Memory, Springer-Verlag.

Kou, Y., Lu, C. T., Sirwongwattana, S. ve Huang, Y.P.(2004), “Survey of fraud detection techniques”, IEEE Int. Conference on Networking, Sensing and Control, 2:749–754.

Liu, P., Zhou, D. ve Wu, N.(2007), “VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise”, International Conference on Service Systems and Service Management (ICSSSM), 1-4.

Lloyd, S. P. (1982), “Least Squares Quantization in PCM”, IEEE Trans. Information Theory, 28:128–137, (orjinal versiyon: Teknik Rapor, Bell Labs, 1957)

MacQueen, J. (1967), “Some methods for classification and analysis ofmultivariate observations”, Proc. 5th Berkeley Symp. Math. Statist. Prob., 1:281–297.

Maes, S., Tuyls, K., Vanschoenwinkel, B. ve Manderick, B.(2002), “Credit Card Fraud Detection Using Bayesian and Neural Networks”, First International NAISO Congress on Neuro Fuzzy Technologies.

Mitchell, T. M. (1997), Machine Learning. McGraw-Hill Companies, ABD.

Michalski, R. S., Carbonell, J. G., ve Mitchell, T. M. (1983), Machine Learning: An Artificial Intelligence Approach, Morgan Kauffmann Inc, Kanada.

Ng, R. ve Han. J.(1994), “Efficient and effective clustering method for spatial data mining”, Very Large Data Bases (VLDB’94).

Nix, D. A. ve Hogden, J. E.(1998), “Maximum-Likelihood Continuity Mapping (MALCOM)”, Proceedings of the 1998 conference on Advances in Neural Information Processing Systems, 744–750.

Tsai, C. F. ve Liu, C. W.(2006), “KIDBSCAN: A New Efficient Data Clustering Algorithm”, Artificial Intelligence and Soft Computing – ICAISC, 702–711.

Zhong, N. ve Ohsuga, S. (1994), “Discovering Concept Clusters by Decomposing Databases”, Data & Knowledge Engineering, 12:223–244.

Wang, W., Yang, J. ve Muntz, R.(1997), “STING: A statistical information grid approach to spatial data mining”, Very Large Data Bases (VLDB’97), 186–195.

Weatherford, M.(2002), ”Mining for Fraud”, IEEE Intelligent Systems, (Tem/Agu):4–6.

Weston, D.J., Hand, D.J., Adams, N.M., Whitrow, C. ve Juszczak, P.(2008), “Plastic card fraud detection using peer group analysis”, Advances in Data Analysis and Classification, 2(1):45–62.

ÖZGEÇMİŞ

Doğum Tarihi	27.01.1982	
Doğum Yeri	Diyarbakır	
Lise	1994-1999	Diyarbakır Anadolu Teknik Lisesi
Lisans	2002-2006	Yıldız Teknik Üniversitesi Elktrik-Elektronik Fakültesi Bilgisayar Mühendisliği Bölümü
Yüksek Lisans	2006-2009	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı

Çalıştığı kurumlar

2005-2006	Turkcell İletişim Hizmetleri A.Ş.
2006-2007	Atos Origin Consultancy
2007-	i2i Technology and Consultancy