MİMAR SİNAN GÜZEL SANATLAR ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

MEKANSAL-ZAMANSAL VERİ MADENCİLİĞİNDE KÜMELEME ANALİZİ

YÜKSEK LİSANS TEZİ

Turgut ÖZALTINDİŞ

Anabilim Dalı: İstatistik

Programı: İstatistik

Tez Danışmanı: Dr. Öğr. Üyesi Elif Özge ÖZDAMAR

HAZİRAN 2018

MİMAR SİNAN GÜZEL SANATLAR ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

MEKANSAL-ZAMANSAL VERİ MADENCİLİĞİNDE KÜMELEME ANALİZİ

YÜKSEK LİSANS TEZİ

Turgut ÖZALTINDİŞ

Anabilim Dalı: İstatistik

Programı: İstatistik

Tez Danışmanı: Dr. Öğr. Üyesi Elif Özge ÖZDAMAR

HAZİRAN 2018

Turgut ÖZALTINDİŞ tarafından hazırlanan MEKANSAL-ZAMANSAL VERİ MADENCİLİĞİNDE KÜMELEME ANALİZİ adlı bu tezin yüksek lisans tezi olarak uygun olduğunu onaylarım.

Dr. Öğr.Üyesi Elif Özge ÖZDAMAR
Tez Yöneticisi

Bu çalışma, jürimiz tarafından İSTATİSTİK Anabilim Dalında yüksek lisans tezi olarak kabul edilmiştir.

Başkan : Dr. Öğr.Üyesi Elif Özge ÖZDAMAR

Üye : Doç.Dr. Semra Erpolat TAŞABAT

Üye : Doç.Dr. Kerem Yavuz ARSLANLI

Bu tez, Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

ÖNSÖZ

Araştırma boyunca değerli bilgilerini benimle paylaşan değerli danışmanım Dr. Öğr. Üyesi Elif Özge ÖZDAMAR başta olmak üzere, yüksek lisans öğrenimim boyunca desteklerini esirgemeyen ve kendimi geliştirmemi sağlayan bölümdeki bütün hocalarıma, her zaman yanımda olan çalışma arkadaşlarıma ve aldığım her kararda arkamda duran ve beni cesaretlendiren, eğitim ve öğretim hayatım boyunca maddi ve manevi olarak her türlü imkanlarıyla destek olan canım aileme teşekkürü bir borç bilirim.

Turgut ÖZALTINDİŞ

MEKANSAL-ZAMANSAL VERİ MADENCİLİĞİNDE KÜMELEME ANALİZİ

ÖZET

Günümüzde, teknolojik gelişmeler ile birlikte üretilen ve depolanan veri hiç olmadığı kadar büyümüş ve çeşitlenmiştir. Artık fotoğraf, video ya da uzayı gözlemleyerek elde etiğimiz büyük boyutlu sinyallerden veri madenciliği kullanılarak bilgi keşfi yapılmaktadır. Verinin büyümesi ve çeşitlemesinin sonucu olarak yapısı da farklılaşmakta ve veri madenciliğinde kullanılan tekniklerinin değişen boyut ve veri yapılarına göre uyarlanması gerekmektedir.

Bahsedilen bu farklı veri yapılarından biri mekansal veridir. Mekansal veri setleri, gözlemlere ait mekansal bilginin enlem ve boylam olarak veri setine dahil edilmesiyle oluşturulur. Mekansal (coğrafik) verilerin günümüz teknolojisinde sıklıkla kullanılmaya başlamasıyla, veri madenciliği bu alanda uygulanmış ve Mekansal Veri Madenciliği (Spatial Data Mining) kavramı ortaya çıkmıştır.

Mekansal bir veri setinin zaman değişkeni barındırması durumunda veri setinin yapısı değişmekte ve Mekansal Veri Madenciliğinde kullanılan tekniklerin bu yapıya uyarlanması gerekmektedir. Bu gereklilik ile birlikte, mekansal-zamansal veri üreten/depolayan kurum ve bilimsel araştırmalarının sayısının artması, yakın zamanda Mekansal-Zamansal Veri Madenciliği (Spatio-temporal Data Mining) kavramının ortaya çıkmasına neden olmuştur.

Bu çalışmada, Mekansal ve Mekansal-Zamansal Veri Madenciliğinde kullanılan kümeleme algoritmaları tanıtılmış ve 1970-2017 yılları arasında Türkiye'nin tüm illerini kapsayan ortalama sıcaklık ve yağış miktarları üzerinde ST-DBSCAN algoritmasını kullanılarak mekansal-zamansal kümelenme analizi yapılmıştır. Önümüzdeki yıllarda bu alanda literatür çalışmasının çoğalarak birçok algoritmanın mekansal-zamansal veri yapısına uyarlanması öngörülmektedir.

CLUSTERING ANALYSIS IN SPATIO-TEMPORAL DATA MINING

SUMMARY

Nowadays, with the technological developments, the data that is produced and stored has grown and diversified as never before. Now, data mining is used to discover information from large-sized signals obtained by observing photographs, video or space. As the result of the growth and diversification of the data, the structure of the data is different and the techniques used in the data mining have to be adapted according to the changing dimensions and data structures.

One of these different data structures is the spatial data. Spatial data sets are created by including the spatial information of the observations into the data set as latitude and longitude. As spatial (geographical) data is frequently used in today's technology, data mining has been applied to this area and the concept of Spatial Data Mining has emerged.

If a spatial data set contains time variable data, the structure of the data set changes and the techniques used in Spatial Data Mining need to be adapted to this structure. With this requirement, the increase in the number of institutions and scientific researches that produce/store spatial-temporal data has led to the emergence of Spatio-Temporal Data Mining in the near future.

In this study, clustering algorithm used in Spatial and Spatio-Temporal Data Mining were introduced and between the years 1970-2017 ST-DBSCAN spatio-temporal clustering algorithm was performed on the average temperature and precipitation covering all the provinces of Turkey. In the coming years, it is predicted that the literature study in this area will be increased and adapted to the spatio-temporal data structure of many algorithms.

İÇİNDEKİLER

Sayfa

ŞEKİL LİSTESİ

Sayfa

ÇİZELGE LİSTESİ

Sayfa

1. GİRİŞ

Son yıllarda tanınan ve gittikçe popüler bir hal alan veri madenciliği kısaca çeşitli veri depolama kaynaklarındaki (veritabanı, veri ambarı v.b.) çok büyük boyutlu veri setleri içindeki anlamlı ve istenilen soruya cevap sağlayan bilgileri keşfetme sürecidir. Bankacılık, bilim ve mühendislik, pazarlama, borsa vb. birçok alanda kullanılan veri madenciliği teknolojinin gelişmesiyle birlikte elde edilen GPS ve benzeri mekansal bilgilerin çoğalması ve depolanmasıyla birlikte Mekansal Veri Madenciliği (MVM) adında yeni bir dal ortaya çıkmıştır. MVM'nin önemi, akademik ve ürün geliştiriciler tarafından giderek daha benimsenmiş ve bu alanda yapılan çalışmalar artmıştır. Buna rağmen coğrafi verilerden bilgi keşfi halen genç bir araştırma disiplinidir ve analitik yöntemlerin çoğu mekansal veri için henüz uyarlanmamıştır. Mekansal Veri Madenciliğinin amacı, veritabanındaki mekansal etkileşimle birlikte gizli olan bilgiyi ortaya çıkarmaktır ve bu süreç; bir veritabanından otomatik veya yarı otomatik olarak ilgili ve anlaşılır bilgiler (kural, düzen, desen, ilişkiler vb.) elde etmeyi sağlayan geleneksel veri madenciliğinde olduğu gibidir.

Veri madenciliği bilgi keşfi sürecinde en çok kullanılan analizler sınıflandırma, kümeleme ve tahmin-öngörü şeklinde sıralanmaktadır. Bunlardan biri olan kümeleme birçok alanda yaygınca kullanılır. Kümeleme kısaca benzer özelliklere sahip olan gözlemlerin bir araya getirilmesidir. Mekansal kümeleme ise birbirine yakın olan ve benzer özellikler gösteren gözlemleri aynı kümeye atama sürecidir. Mekansal kümelemede ana amaç küme içi benzerliği maksimum seviyeye çıkarıp kümeler arası benzerliği ise minimum seviyeye indirmektir. Balıca mekansal kümeleme algoritmaları olarak DBSCAN, GDBSCAN, DBSC, VDBSCAN ve DBCLASD algoritmaları gösterilebilir.

Tezinde ana konusu olan MZVM'de veriler, mekansal veri kümesinin zamansal dilimler halinde saklanan verilere karşılık gelir. Mekansal-zamansal verilerden bilgi keşfi, veri madenciliğinin umut verici bir alt alanıdır, çünkü mekansal-zamansal veri yapısı günden güne çoğalmakta ve analiz edilmesi gerekmektedir. Mekansal-

zamansal veriler için bilgi keşfi süreci mekansal ve zamansal olmayan veri yapıları ile karşılaştırıldığında daha karmaşıktır. MZVM'de en çok kullanılan yötemler arasında Mekansal-zamansal kümeleme üst sıralardadır. Mekansal-zamansal kümeleme algoritmaları hava tahminleri, tıbbi görüntüleme ve coğrafi bilgi sistemleri gibi birçok alanda kullanılmaktadır. Birant ve Kut tarafından 2007 yılında literatüre kazandırılan ST-DBSCAN algoritması en popüler olan kümeleme algoritmalarından biridir. Bu tezin ana konusu ve uygulamada kullanılan ST-DBSCAN algoritmasıdır.

Uygulamada, Meteoroloji Genel Müdürlüğünden alınan 1970-2017 yılları arasında yıllık olarak kaydedilmiş ortalama sıcaklık ve toplam ortalama yağış miktarları ST-DBSCAN algoritması kullanılarak kümelenmiştir. Algoritmada kullanılan parametreler değiştirilerek parametrelerin algoritma sonucu üzerine etkisi incelenmiş ve herbir analiz sonucu görselleştirilerek yorumlanmıştır.

2. MEKANSAL VERİ ANALİZİ

Mekansal ya da diğer adıyla coğrafik veri, herhangi bir niteliğin yanında lokasyon bilgisiside içeren verilerdir. Mekansal veri ve mekansal veri analizi (MVA) kavramları, temeli 1950'li yıllara dayanan çoğunlukla Coğrafik Bilgi Sistemleri (Geographical Information Systems, GIS) ve Coğrafik Bilgi Bilimi (Geographical Information Science, GISc) literatüründe yaygın olarak kullanılan bir terimdir. MVA en basit tanımla, veri setindeki her bir gözlemin mekansal bilgilerini kullanan teknikler bütünü olarak açıklanabilir.

MVA üç temel ana birimden meydana gelmektedir. İlki harita modelleme ya da haritalamadır. Bir haritadaki belli bir uzaklığın içinde olan tüm alanların (istasyon, kuyu, yol) tanımlanma işlemidir. Herbir veri harita üzerinde temsil edilir. İkincisi, mekansal analiz modellemedir. Bu modeller mekansal nesneler arasındaki etkileşimin yapısına, mekansal ilişkinin yapısına ya da gözlemlerin coğrafik konumlarına dayanmaktadır. Üçüncü ve son olan ise istatistiksel tekniklerin uygulanmasıdır (Haining, 2003).

Mekansal veri analizini (MVA) mekansal olmayan geleneksel analizlerden ayıran en önemli ve mekansal analizin olmazsa olmaz unsuru veri setinde lokasyon bilgisi olmasıdır. Eğer mekansal ilişkileri göz ardı edip yalnızca değişkenler ile ilgilenilirse, gözlemler her ne kadar mekansal olarak tanımlanmış olsa bile MVA yapıldığından bahsedilemez. Böyle durumlarda değişkenler ne kadar önemli olursa olsun mekansallıkla beraber çalışılmadığında değer ve anlam kaybederler (Fischer & Wang, 2011). Mekansal olarak birbirine yakın olan gözlemlerin uzak olan gözlemlere göre birbirine daha çok benzemesi beklenir (Tobler, 1970). Bu durum bağımlılık yapısını ortaya çıkarmaktadır. Örneğin suç oranı yüksek bir şehrin yakınındaki illerde de suç oranı yüksek çıkabilir ya da gelir seviyesi düşük olan bir bölgenin yakınındaki bölgelerde de gelir oranı düşük olabilir. Bu mekansal kümelenmelerin varlığında, gözlemlerin bağımsızlığı varsayımı geçerli değildir (Anselin, 1992). Bu tür verilere klasik istatistik teorisinin uygulanması problemlere neden olmaktadır. Bu nedenle mekana bağlı veriler istatistiksel olarak analiz

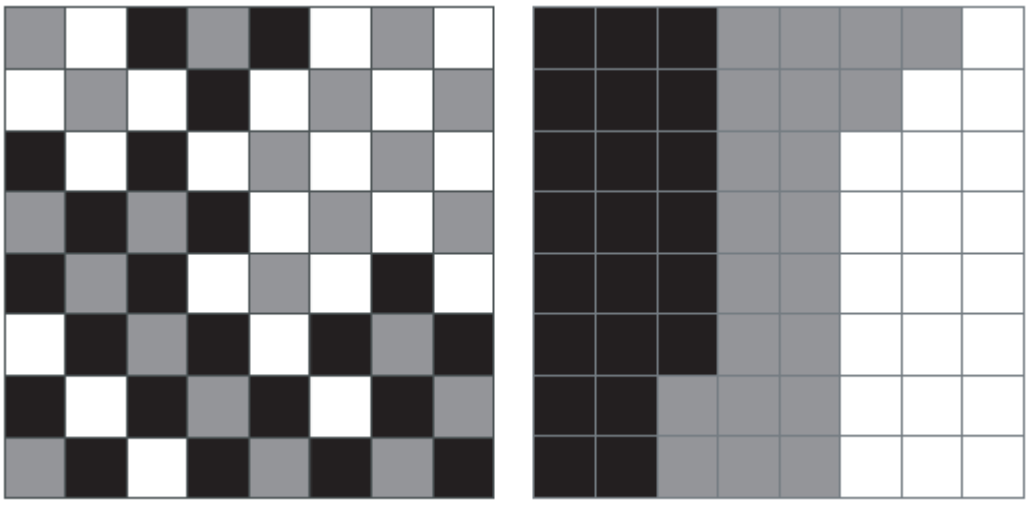
edilirken kendine özgü yöntem ve teknikler geliştirilmiştir. MVA ile klasik istatistiksel veri analiz arasındaki farklılık tamamen veri yapılarındaki farklılıktan meydana gelmektedir. Bu nedenle iki analiz türleri farklı tip sorulara cevap aramaktadır. MVA'ya örnek olarak aşağıdaki soru tipleri gösterilebilir.

- Geçen ay en çok hırsızlık nerede meydana gelmiştir?
- Havzalarda ne kadar ormanlık alan bulunmaktadır?
- Hangi illerde akciğer kanseri riski daha fazladır?

Klasik istatistiksel veri analizinde analiz edilmek istenen sorular ise şu şekildedir.

- Şehirdeki konut fiyatlarındaki değişimi etkileyen ana değişkenler nelerdir?
- Mekansal değişkenler konutların özelliklerinden daha anlamlı mıdır?
- Bu sonuçlar şehirler arasında nasıl karşılaştırılır?

MVA özellikle coğrafya ve bölgesel (mekansal) bilim dalları sayesinde son yıllarda teori, yöntemler ve uygulamalar açısından oldukça anlamlı bir büyüme sergilemiştir. Coğrafya haricinde biyoloji, meteoroloji, sağlık, ekoloji, tarım, mühendislik gibi diğer bilim dallarında da MVA yöntemleri kullanılmaktadır.



Şekil 2.2.1 (A) Zayıf Mekansal İlişki ve (B) Güçlü Mekansal İlişki.

2.1 Mekansal Analiz ve Mekansal Veri Analizi

Mekansal analiz (MA) terimi coğrafya alanında çok daha eskilere dayandığından dolayı daha çok Coğrafi Bilgi Sistemleri (CBS) alanında kullanılır. MA ile MVA'nın

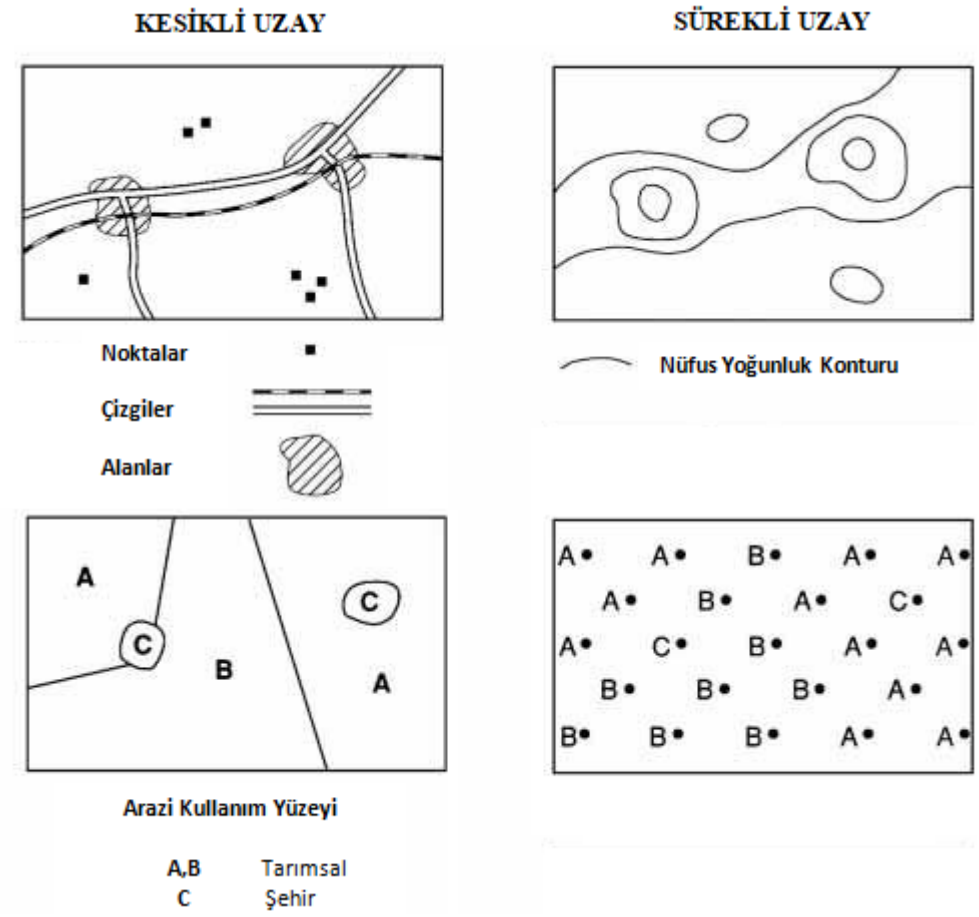
tanımı literatürde çoğunlukla benzer şekilde yapılmakta hatta çoğu kişi tarafından aynı terim olarak düşünülmektedir. MVA ve MA aynı şekilde tanımlanabilir. Fakat bazı durumlarda bu iki terim birbirinden ayrılmaktadır. MVA'ni tanımlamanın zorluğuda bu iki kavram arasındaki benzerlikten kaynaklanmaktadır. Bu iki terim arasındaki ayrım Bailey and Gatrell (1995) tarafından şu şekilde yapılmıştır; *MVA'nin amacı mekansal hadiseleri kolayca anlaşılabilir kılmak, hipotezleri test etmek ve konumlardaki bilinmeyen değerleri tahmin etmektir. Buna ek olarak MVA, konumlar arasındaki mekansal ilişkileri de tespit etmeye çalışmaktadır*. Fisher (2006) ise MVA'ni şu şekilde ifade etmektedir; *MVA, mekansal verilerin özelliklerini ortaya koyar, şekil yapılarını tanımlar, hipotez ve modellerle test ederek bunlar arasındaki ilişki yapısını çıkarır*.

2.2 Mekansal Veri ve Analiz Türleri

Coğrafi gerçeklik modelleme, dijital depolama mümkün olduğu sürece gerçek dünyanın karmaşıklığını yakalama sürecine denir. Nesneler ve kapsamlar coğrafi gerçekliği oluşturan birimlerin iki ana kavramıdır. Sıcaklık derecesi, kar derinliği ve denizden yükseklik seviyesi gibi değişkenler gerçeklik kavramına uygun örnekler olarak verilebilir. Bir ev (nokta), yol (çizgi) ve idari birim (alan) nesne kavramına uygun örneklerdir. Mekansal veri analizinde kullanılacak olan uygun istatistiksel tekniği belirlemenin ilk adımı mekansal verilerin sınıflandırılmasıdır (Fischer & Wang, 2011). Ölçme düzeyi, verilerin biçimsel özelliklerini ifade etmenin yanı sıra verilere hangi istatistiksel analizin uygulanacağını belirlemektedir. Bu nedenle değişkenlerin ölçme düzeylerinin belirlenmesi önemlidir. Mekansal verilerin yapısını tanımlamada, üzerinde değişkenlerin ölçüldüğü uzayların ve bu değişkenlerin kesikli veya sürekli yapıda olduğunun ayırt edilmesi önemlidir. Eğer uzay sürekli (alan) bir yapıda ise değişkenlerde sürekli yapıda olmalıdır. Çünkü kesikli değerler altında bu alanın sürekliliği korunamaz. Eğer uzay kesikliyse veya kesiklilerden sürekli bir alan yapılmışsa, değişken değerleri sürekli veya kesikli olabilir (Haining, 2003). Mekansal veriler nokta desenli, çizgisel, alansal ve kompleks olmak üzere dört yapıda karşımıza çıkmaktadır.

- **Noktasal:** Uzunluğu ve alanı olmayan, bir lokasyonu tanımlayan koordinat bilgilerine sahip verilerdir.

- **Çizgisel:** Uzunluğu olan fakat alanı olmayan, iki ya da daha fazla noktanın birleşmesiyle oluşan ve bazı değişkenleri tanımlamakta kullanılan veri yapılarıdır.
- **Alansal:** Alan verileri birçok koordinatın birleşmesiyle meydana gelmektedir. Basitçe tanımlamak gerekirse çizgiye benzer yapıdadır fakat ilk koordinat noktasıyla son koordinat noktası aynıdır.
- **Kompleks:** Kompleks nesneler 2 veya daha fazla yapıdan (nokta, çizgi, alan) meydana gelmektedir.



Şekil 2.2.2 Kesikli ve Sürekli Uzay Temsilleri (Haining, 2003).

Nesne uzayı nokta, çizgi ve alanlar tarafından temsil edilir. Örnek olarak şehir merkezleri bir alan olarak gösterilebilirken başka bir çalışmada ise nokta olarak alınabilir. Alanlar ise sonsuz sayıda lokasyon noktasının birleşiminden oluşabilmektedir. Fakat alan verilerini depolamak için bu yapının sonlu hale gelmesi gerekmektedir. Alanlar kontur çizgileri ile ve düzgün mekansal üniteler olan ızgara

yapıları ile de ifade edilebilir. Izgara boyutu, gösterimin mekansal çözünürlüğünü belirler (Haining, 2003).

Çizelge 2.2.1 Mekansal Veri Türleri (Haining, 2003)**.**

	Nokta	**Çizgi**	**Alan**	**Yüzey**
Sınıflayıcı	Soyulan / Soyulmayan konutlar	Onarılan/Onarılmayan yollar	Yaşam tarzlarına göre nüfus alanları	Arazi türleri
Sıralayıcı	Bir bölgedeki kentlerin yaşam kalitesine göre tercih sıraları	Yolların sınıflandırılması	Gelir sınıflarına göre nüfus alanları	Toprağın yapısı
Aralıklı	Şehirlerin gelişmişlik endeksi	Uzaklıklar (Greenwich meridyenine göre)	Alanlar İçin Gelişmişlik Endeksi	Yeryüzü sıcaklığı
Oranlı	Fabrikadan elde edilen üretim miktarı	Yük Tonajı	Bölgesel kişi başına gelir	Alana düşen yağış oranı

2.3 Mekansal Veri Matrisi

Mekansal veriler değişkenlerin mekansal nesne türlerine ve ölçüm düzeylerine göre sınıflandırılmaktadır (Fischer & Wang, 2011). Z_1, Z_2,…, Z_k; k sayıda rastgele değişken ve S ise bu değişkenlerin lokasyon bilgileri olmak üzere; mekansal veri matrisi şu şekilde ifade edilir (Haining, 2003):

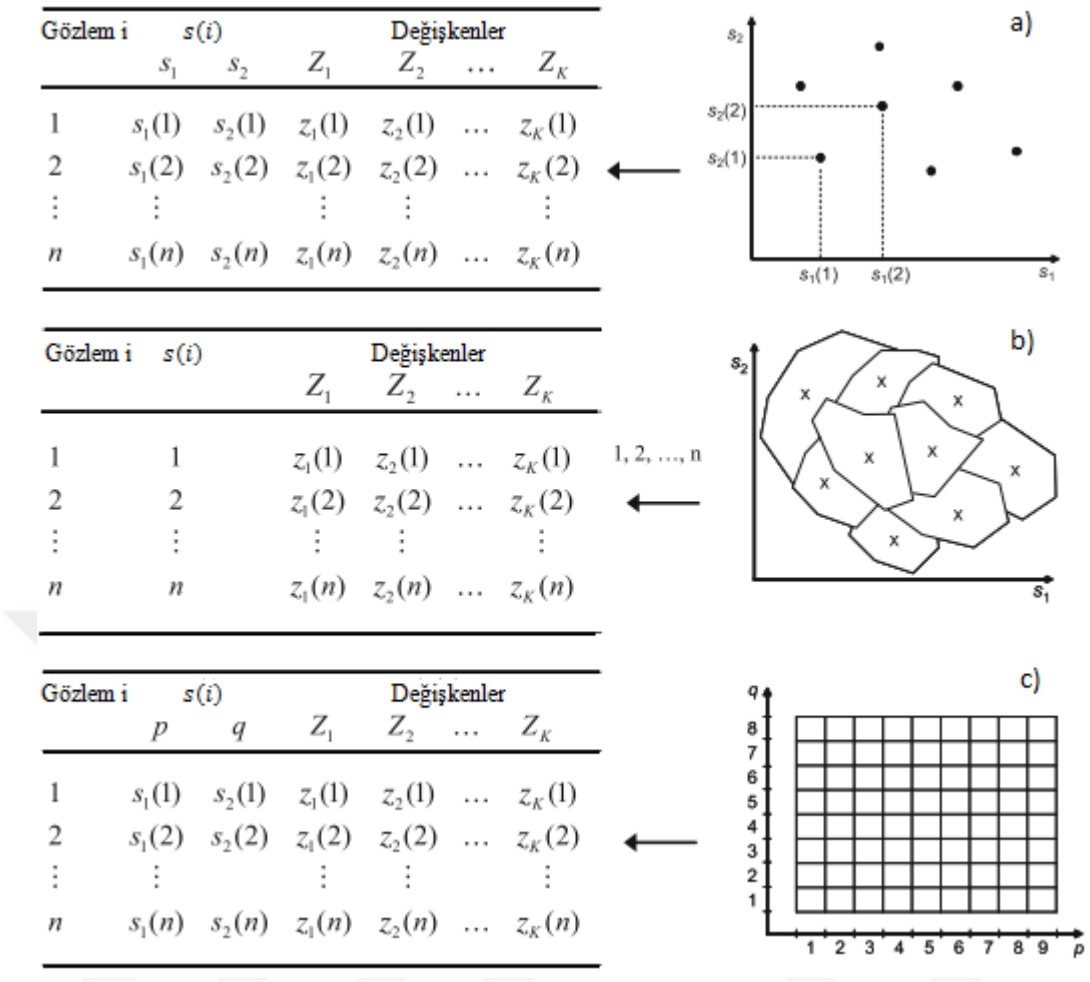
K Değişkenli Veri — Lokasyon

$$\begin{array}{cccccc} Z_1 & Z_2 & \dots & Z_K & S & \\ \begin{bmatrix} z_1(1) & z_2(1) & \dots & z_K(1) & s(1) \\ z_1(2) & z_2(2) & \dots & z_K(2) & s(2) \\ \vdots & \vdots & & \vdots & \vdots \\ z_1(n) & z_2(n) & \dots & z_K(n) & s(n) \end{bmatrix} & & & & & \begin{matrix} \text{Gözlem 1} \\ \text{Gözlem 2} \\ \vdots \\ \text{Gözlem 3} \end{matrix} \end{array}$$

Mekansal veri matrisi, daha kısa gösterimle eşittlik 1.1'de ifade edilen şekilde de kullanılmaktadır.

$$W_{i,j} = \{ z_1(i), z_2(i), \cdots, z_k(i)/s(i)\} \qquad i = 1,2, \dots, n \qquad (1.1)$$

Mekansal veri matrisinin kısa gösterimindeki z_k'lar değişkenlerin değerlerini, i indisi işlem sırasındaki olayı ve buna bağlı olarak da s(i) ise olaya ilişkin mekansal nesnenin konumunu ifade etmektedir. Mekansal veri setindeki nesnelerin yapıları ile mekansal veri matrisinin ilişkilendirilmesi detaylı olarak aşağıdaki Şekil 2.3'de gösterilmektedir.



Şekil 2.2.3 Mekansal Nesnelerin Konumlarının Atanması (Fischer & Wang, 2011).

Eğer iki boyutlu uzayda bulunan veriler noktasal nesnelerden oluşuyorsa, yukarıdaki Şekil 2.2.3 a)'da olduğu gibi i. gözlemin konumu ortogonal bir kartezyen koordinat çifti olarak verilir. Eğer eldeki veriler farklı ve düzensiz şekillerden oluşuyor ise Şekil 2.2.3 b)'de gösterildiği gibi o alanın merkezi varsayılan rasgele bir nokta seçilir ve $s(i) = \left(s_1(i), s_2(i)\right)^T \;\; i = 1,2, \dots, n$ bulmak için noktasal nesnelerdeki süreç aynı şekilde uygulanır. Eğer alanlar düzenli ve eşit şekillerden oluşan ve uzaktan algılanmış veriler ise Şekil 2.2.3 c)'de olduğu gibi gösterilebilirler (Fischer & Wang, 2011).

3. VERİ MADENCİLİĞİ

Günümüzde, teknolojik gelişmeler ile birlikte üretilen ve depolanan veri hiç olmadığı kadar büyümüş ve bu verilerden bilgi çıkarımı kolaylaşmıştır. Bu sayede çok büyük boyutlu veriler günlük olarak saklanabilmektedir. Her gün bireysel bilgisayarlarda bile çok büyük boyutlu (terabytes or petabytes) veri akışı olmaktadır. Bu gelişmeleri anlatmak adına günümüzde kullanılan "bilgi çağında yaşıyoruz" cümlesi çok popüler bir cümledir oysaki o cümle yerine "veri çağında yaşıyoruz" cümlesini kullanmak daha doğru olacaktır. Bu verilerden anlamlı bilgiler çıkarabilmek için geliştirilen yöntemlerin tamamına veri madenciliği denilebilir. 1900'lü yıllarda tanınan veri madenciliği günümüzde çok kullanılmakta ve popülerliği katlanarak artmaktadır. Veri madenciliğinin tanımı birçok şekilde yapılabilir ve bu durum literatürede yansımaktadır. Genel olarak tanımlamak gerekirse çok büyük boyutlu veri setleri içerisinden anlamlı ve kullanılabilir bilgilerin gün yüzüne çıkarılması olarak ifade edilebilir. Diğer bir deyişle veri madenciliğinin amacı verilerden elde edilmiş olan bilgilerin istatistiksel yöntemlerle analiz edilip ilgili alanlarda kullanılmasıdır (Akın, 2008). Veri madenciliğinin literatürde bulunan birkaç tanımı şöyledir:

Özkan (2008)'a göre Veri madenciliği "değeri olan" bilgiyi birçok veri arasından elde etme işi olarak tanımlamaktadır. Böylece eldeki verilere uygulanan belirli yöntemlerle mevcut veya geleceğe yönelik anlamlı bilgiler elde edilebilir. Elde edilen bu bilgiler karar verme aşamasında önemli rol oynamaktadır.

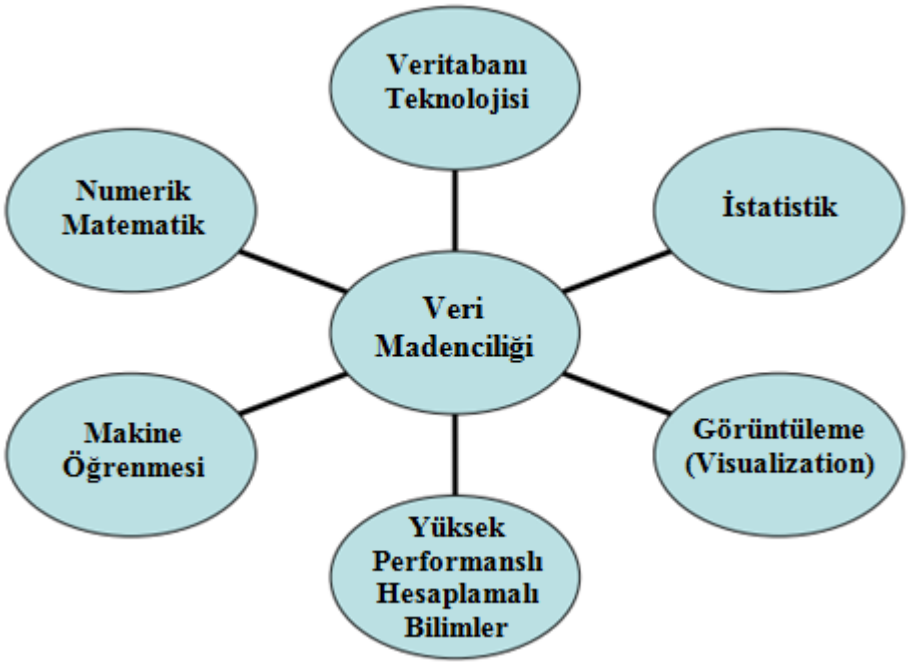
J. Han ve M. Kamber'e göre veri madenciliği "Her türlü veri depolama kaynaklarındaki (veritabanı, veri ambarı v.b.) çok büyük boyutlu veri içindeki anlamlı ve istenilen soruya cevap sağlayan bilgileri keşfetme sürecidir".

Dönmez (2008)'e göre "Veri madenciliğini istatistiksel bir yöntemler serisi olarak görmek mümkün olabilir. Ancak veri madenciliği, geleneksel istatistikten birkaç yönden farklılık gösterir. Veri madenciliğinde amaç, kolaylıkla mantıksal kurallara ya da görsel sunumlara çevrilebilecek nitel modellerin çıkarılmasıdır."

Sumathi ve Sivanandam (2006) veri madenciliği ve bilgi keşfi tanımlamalarını aşağıdaki gibi sıralamıştır;

- Veri madenciliği, Büyük boyutlu verilerin içindeki değerli bilgiyi ortaya çıkarmak için kullanılan etkili bir araştırma çeşididir.
- Veri madenciliği: Verilerdeki, kullanışlı ve anlamlı örüntüleri tanımlamak için kullanılan çok önemli bir süreçtir.
- Veri madenciliği: Değerli verilerdeki ilişkileri ortaya çıkarma araştırmasıdır.
- Veri madenciliği: Rekabet ortamı olan alanlarda kullanıldığında avantaj sağlayan bilginin keşfidir.
- Veri madenciliği: Veri setlerinden anlamlı model ve örüntüleri tespit eden tümevarımdır.
- Veri madenciliği: Büyük veri depolama alanlarından önceden bilinemeyen istenilen anlamlı bilginin gün yüzüne çıkarılması ve kritik kararlarda kullanma sürecidir.

Bu tanımları da göz önünde bulundurarak en genel anlamda Veri madenciliği, geniş veri yığınları içerisinde, yararlı olma potansiyeline sahip, aralarında beklenmedik (bilinmedik) ilişkilerin olduğu verilerin keşfedilerek, veri sahibi için hem anlaşılır hem de kullanılabilir bir biçime getirilmesine yönelik geliştirilmiş yöntemler topluluğudur. Veri Madenciliği, Veritabanı Sistemleri, İstatistik, Makine Öğrenmesi, İnsan Makine Etkileşimi, Veri Görselleştirme gibi farklı alanlardan faydalanan disiplinler arası bir alandır (Han ve Kamber, 2006). Veri madenciliğinin bazı disiplinlerle olan ilişkisi aşağıda Şekil 3.3.1'de gösterilmiştir.



Şekil 3.3.1 Veri Madenciliğini Disiplinlerarası Yapısı.

Veri madenciliği günümüzde belirtildiği gibi pek çok alanda uygulanabilmektedir. Aşağıda veri madenciliğinin kullanım alanları ve bunlara bazı örnekler verilmiştir:

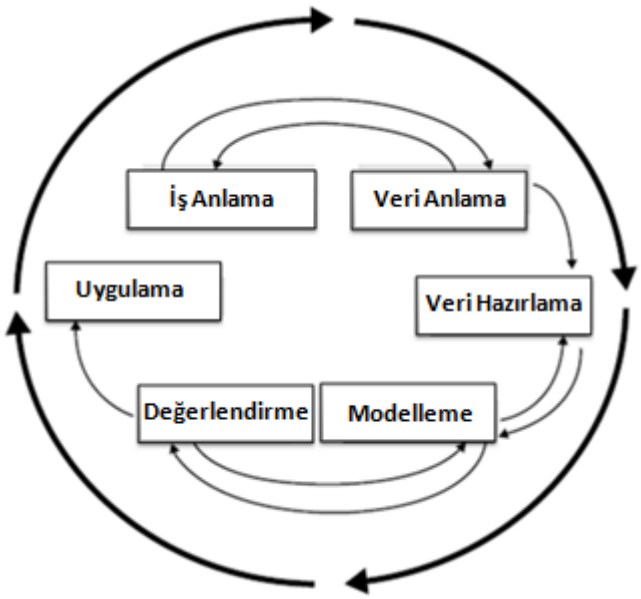
- ***Bankacılık:*** Kredi taleplerinin değerlendirilmesi, risk analizleri, dolandırıcılık tespiti, tarihsel pazar verileri kullanılarak belirli kuralların oluşturulması, kullanıcı gruplarının kredi kartı harcamalarını saptanması
- ***Bilim ve Mühendislik:*** Bilimsel ve teknik problemlerin çözümlenmesi
- ***Borsa:*** Hisse senedi fiyat tahmini, genel piyasa analizleri, alım-satım stratejilerinin optimizasyonu
- ***CRM/Müşteri Analitiği:*** Müşteri memnuniyetinin artırılması, yeni müşterilerin kazanılması, değerli müşterilerin elde tutulması, davranış analizi
- ***Elektronik Ticaret:*** Müşteri ilişkileri yönetimi, saldırıların çözümlenmesi
- ***Endüstri:*** Kalite kontrol, üretim süreci kontrol
- ***Pazarlama:*** Çapraz satış analizleri, müşteri değerlendirme, pazar sepeti analizi, müşterilerin öncelikli ihtiyaçlarının belirlenmesi, kampanyaların belirlenmesi, müşterilerin satın alma alışkanlıkların belirlenmesi, yeni müşterilerin kazanılması
- ***Sigortacılık:*** Müşteri kaybı sebeplerinin belirlenmesi, usulsüzlüklerin önlenmesi, risk analizi (riskli müşterilerin davranış kalıplarının tespit

edilmesi), sigorta poliçesi üzerinden ödenecek para analizi yapılması, yeni müşteri tahmini

- ***Telekomünikasyon:*** Müşteri bölünmeleri, hile tespiti, hatların yoğunluk tahminleri, kaybedilen müşterinin geri kazanılması, kalite ve iyileştirme analizleri, abonelik tespitleri
- ***Tıp:*** Tıbbi teşhis, uygun tedavi sürecinin belirlenmesi, hasta davranışlarının tahmin edilerek karakterize edilmesi, demografik ve tarihi veriler ışığında bölgelerin incelenerek potansiyel hastalık tehlikelerinin tahmin edilmesi, farklı hastalıklar üzerinde yapılan başarılı tıbbi terapilerin tanımlanması
- ***Güvenlik:*** İnternet sayfalarının anahtar kelimelerle taranarak lehte ve aleyhte propaganda yapan sayfaların belirlenmesi, haberleşme araçları takip edilerek terörist faaliyetlerin belirlenmesi

3.1 Veri Madenciliği Süreçleri

Veri madenciliği süreçlerinde en büyük zaman alan ve sonucu etkilemede en etkili olan bölüm veri hazırlama aşamasıdır. En önemli aşama olmasının nedeni ise eldeki veri ne kadar gerçeğe yakın (doğru), temizlenmiş ve güvenilir olursa sonuçların güvenilirliği ve doğruluğuda aynı oranda artacak olmasıdır. Veri hazırlama aşaması kendi içinde birçok kola ayrılmaktadır (verilerin temizlenmesi, verilerin birleştirilmesi, verilerin dönüştürülmesi, verilerin indirgenmesi).



Şekil 3.3.2 Veri Madenciliği Süreçleri.

3.1.1 Araştırılan konunun tanımlanması

Veri madenciliği sürecinin ilk aşaması olan bu evre, araştırılan problemi tanımlanma (Business Understanding) ya da öğrenilmesi istenen konuyu tanımlanma evresi olarak da adlandırılabilinir. Veri madenciliğinin önemli aşamalarından biridir. Anlama ve tanımlama aşaması olmasından dolayı uygulamanın hedefini, gereksinimlerini ve kısıtlamalarını tam olarak tanımlar ve bu doğrultuda uygulama öncesi plan program oluşturur. Amaç herkesce anlaşılabilir açık bir dille ifade edilmiş olmalıdır. Yapılan uygulama sonucunda doğru cevaplanmış birçok yanlış soru elde edilmek istenmiyorsa, uygulamanın cevabı istenen soru ile aynı doğrultuda olduğu kesin olarak bilinmelidir (Argüden ve Erşahin, 2008).

3.1.2 Verileri anlama aşaması (Data Understanding)

Bu aşama verileri elde etmekle birlikte başlamakta ve daha sonrasında verilerle aşina olma, veri kalitesini test etme, benzer verileri tespit etme gibi işlemlerle devam etmektedir. Genel anlamda veri keşfi aşamasıdır. Araştırılan konunun tanımlanması aşaması ile çok bağlantılı bir aşamadır. Verilere bakarak istenilen araştırma konusuna ek olarak yeni araştırma konuları oluşturulabilir. Benzer bir şekilde araştırılacak konuya göre veri hakkında bazı bilgiler elde edilebilir.

3.1.3 Veri hazırlığı aşaması (Data Preperation)

Önceden de belirtildiği üzere verinin hazırlanma aşaması veri madenciliğinin en önemli aşamasıdır. Veri kalitesi veri madenciliğinde anahtar bir konu olup kalite ne kadar yükselirse uygulama sonucunda elde edilen bilgilere güvende okadar yükselmektedir. Güvenilirliğin artması için veri hazırlama aşaması çok titiz ve ayrıntılı bir biçimde ugulanmalıdır. Aksi durumlarda çalışma sonucu hatalı olacak ve yanlış yorumlar elde edilecektir. Ayrıca bu yapılan yanlıştan dönmek ve bu aşamayı tekrardan uygulamak kullanıcıyı zamandan da çok büyük kayıplara uğratacaktır. Bu aşamada genel olarak veriler temizlenir ve düzenlenir. Verilerde kullanıcı tarafından yanlış olarak girilmiş veya kayıp gözlemler olabilir. Tüm bu durumların düzeltilmesi veri hazırlama aşamasının bir parçasıdır. Son zamanlarda verilerin boyutu çok büyük boyutludur ve günden güne gelişen teknoloji ve depolama alanları ile birlikte büyümeyede devam etmektedir. Bu büyüklükte veri setlerini hazırlamada dikkat edilmesi gereken maddeler aşağıda gösterilmektedir (Özmen, 2003).

- Gereksiz (elde edilmek istenen amaca hizmet etmeyen) olan fazla değişkenler varsa uygulamadan çıkarılmalı
- Hatalı giriş veya kayıp gözlem tespiti yapılmalı ve düzeltilmeli
- Kayıp gözlemlerin durumu incelenmeli hatalı sonuçlara neden olur mu öğrenilmeli
- Birbirini tekrar eden benzer veriler çalışmadan çıkarılmalı
- Eklenecek herbir değişken için çalışmaya katkıda bulunma durumu araştırılmalı istenilen seviyede katkıda bulunmayan değişkenler eklenmemeli

Veri hazırlığı aşamaları veri temizleme, veri birleştirme, veri dönüştürme ve veri indirgeme şeklinde sıralanabilir.

-Verilerin Temizlenmesi: Verinin kalitesini arttırma işlemlerinden oluşan bölümdür. Veri seti içerisindeki hatalı, tutarsız ve aykırı gözlemler tespit edilerek verisetinden çıkarılır. Genel olarak bu tarz hatalı ve eksik gözlemlerin çalışmadan çıkarılması tercih edilmektedir. Fakat bazı durumlarda ise kayıp gözlemleri tahmin etmekte kullanılan yöntemler (imputation) kullanılarak kayıp değerler doldurulabilir. Kayıp gözlem doldurmada kullanılan en bilindik ve basit yöntemlerin bazıları aşağıda belirtilmektedir (Han, Kamber, & Pei, 2012).

- Değişkendeki tüm gözlemlerin ortalaması kayıp gözlem yerine kullanılabilir.
- Kayıp gözlemler kullanıcı tarafından elle doldurulabilir. Büyük verilerde ve konu hakkında bilgi sahibi olmayan kullanıcılarda verimli değildir.
- Tüm örneklemler için değişkenin ortalama veya medyan değeri kullanılabilir.
- Elde olan veriler analiz edilerek en uygun değerler tespit edilip kullanılabilir. Burada sözü edilen değerlerin tespiti için regresyon veya karar ağacı gibi yöntemler kullanılmaktadır.

-Verilerin birleştirlmesi (aggregation): Veri madenciliğinde bazı çalışmalarda farklı veri depolarında bulunan veriler birleştirilmek istenmektedir. Birleştirme (aggregation) veride birden fazla veri setinini birleştirilerek tek veri seti haline gelmesidir. Birleştirme veriye daha geniş kapsamlı bakabilme imkanı sağlamaktadır. Birleştirme işlemi yapılırken bazı hatalar meydana gelebilmektedir. Örneğin, bir veri tabanında işleme alınan değişken "istasyon ID" şeklinde yapılmışken, bir diğerinde "istasyon numarası" şeklinde olabilir. Bu tür şema birleştirme hatalarından kaçınmak

için meta veriler kullanılır. Veri tabanları ve ambarları çoğunlukla meta veriye sahiptirler. Meta veri, veriye ilişkin veridir. Veri birleştirmede önemli olan diğer bir nokta ise ölçekleme kodlama ya da değişken birimlerindeki farklılıklardır. Örneğin, bir veride iki birim arası mesafe metre ile ifade edilmişken başka bir veride kilometre ile ifade edilmiş olabilir. Bu şekilde yapılan birleştirilmiş verilerde son derece yüksek seviyede hatalı sonuçlara neden olabilmektedir. Son olarak birleştirilecek olan değişkenler arası bağlantı olup olmadığı araştırılıp çalışmaya o şekilde dahil edilmelidir. Aynı şeyi ifade eden birden fazla değişkende hatalı sonuçlara neden olabilir (Akın, 2008).

-Verilerin Dönüştürülmesi: Veri madenciliği süreci daha verimli ve faydalı olsun diye ve bulunan modeller daha kolay anlaşılabilsin diye uygulanan bir süreçtir. Verinin analize kattığı anlamı korunarak şeklinin dönüştürülmesi işlemidir. Aykırı değere ve gürültüye sahip olan veri setlerinde ve değer aralığı çok geniş olan verilerde oluşabilecek potansiyel hataları azaltmak ve önlemek için kullanılır. Dönüştürme yapılmış veriler ile çalışmak analiz sonuçlarında daha doğru sonuçlar doğurmaktadır. Veri dönüştürme; düzleştirme, birleştirme, genelleştirme ve normalleştirme gibi birbirinden farklı anlamlara ve görevlere sahip olan yöntemler bulunmaktadır. Normalleştirme yöntemi en sık kullanılan işlemlerinden birisidir.

-Veri İndirgeme: Veri madenciliği, genelde büyük veri setleri ile uygulanmaktadır ve bu durum bazı sorunlar doğurmaktadır. Böyle büyük verilerle çalışırken karmaşık olan veri madenciliği yöntemlerini uygulamak büyük boyutta zaman problemlerine yol açmaktadır. Kullanıcının böyle durumlarda veri boyutunu küçültmesi gerekebilmektedir. Veri indirgeme, ana veri seti ile neredeyse aynı anlama sahip olan fakat aynı zamanda ana veriden olabildiğince düşük boyutlu veri oluşturma işlemidir. Veri indirgeme tekniklerinin bazıları aşağıda gösterilmektedir (Han ve Kamber, 2001) .

- Veri Küpü Birleştirme: Örnek ile anlatmak gerekirse bir veri setinde aylık olarak girilmiş bilgileri 6 aylık yada yıllık olarak değiştirerek boyutun küçültülmesi işlemidir.
- Temel bileşenler analizi: İlgisiz, az ilgili veya gereksiz olan değişkenlerin kaldırılmasını sağlar.
- Örnekleme yapılarak veri boyutu küçültülebilir.

- Karar ağaçları: Genelde sınıflama için kullanılmasına rağmen, değişkenler için kullanılmaktadır. Oluşturulan ağaç yapısında var olmayan değişkenler veri setinden çıkarılmış değişkenler olarak düşünülür.
- Regresyon değişken seçimi yöntemleride yaygın olmamakla birlikte boyut küçültmek için kullanılmaktadır.

3.1.4 Modelleme aşaması (Modeling)

Veri madenciliğinde modelleme aşamasıda en önemli aşamalardan biridir. Bunun nedeni eldeki verilerden güzel ve doğru bilgilerin alınabilmesi modelin doğru oluşturulmasına dayanmaktadır. Modelin doğru kurulması sonuçların kaliteli olmasını sağlamaktadır. Model yapısının doğru bir şekilde kurulamadığı durumlarda ise veri setindeki anlamlı bağlantılar doğru olarak tanımlanamaz ve verideki örüntü yapıları tespit edilemez. Bu nedenle de oluşturulan modelden doğru bilgiler çıkarma ihtimali azalır (Kökver, 2012).

Modelleme aşaması genel olarak en uygun modelin belirlenmesi ve uygulanması şeklindedir. Modeli belirlemede birçok yöntem bulunmaktadır. En uygun model, alternatif teknikler ile birçok model oluşturulduktan sonra deneme yanılma yöntemi ile bulunmaktadır. Modeller oluşturulduktan sonra modellerin araştırma konularına ne kadar anlamlı cevaplar sağladığı ve amaca uygunluk gibi kriterlerle sınınmalıdır. Modeller arasında bu kriterleri en iyi karşılayan model seçilir. Seçilen modelin başarısı ve anlamlılığı ne kadar tatmin edici olursa olsun, gerçek dünyayla yüzde yüz benzer model oluşturduğu garanti edilemez. Modelleme aşamasında istenilen sonuçlar elde edilemiyor ise veri hazırlama evresine geri dönülebilir (Larose, 2005).

3.1.5 Uygulama aşaması (Deployment)

Bu aşamaya kadar yapılanların tümünün kullanılıp sonuca bağlandığı veri madenciliği süreçlerinin sonudur. Bütün aşamalar sonucunda bu aşamada elde edilen sonuçlar ile raporlama ve araştırmanın incelenme işlemleri yer almaktadır. Sonuçlar elde edilir, değerlendirilir, uzman kişi tarafından raporlanır ve elde edilen sonuçlar gerçek hayata uyarlanır (Albayrak ve Yılmaz, 2009).

3.2 Mekansal Veri Madenciliği

Günümüz bilimsel ve teknolojik gelişmeler sonucunda, insanların gündelik olarak kullandığı teknolojik araç ve uygulamalarda, Global Konumlandırma Sistemi (GPS),

cep telefonu sinyalleri ve radyo frekansı tanımlama (RFID) gibi kayıt teknolojilerinin ürettiği coğrafik verinin analizi ile elde edilen sonuçlar kullanılır hale gelmiştir. Bunun bir doğal sonucu olarak Mekansal Veri Madenciliğinin (MVM) önemi, akademik ve ürün geliştiriciler tarafından giderek daha benimsenmiş ve bu alanda yapılan çalışmalar artmıştır. Buna rağmen coğrafi verilerden bilgi keşfi halen genç bir araştırma disiplinidir ve analitik yöntemlerin çoğu mekansal veri için henüz uyarlanmamıştır.

Mekansal verinin öneminin kavranmasıyla günümüzde kullanılan neredeyse tüm veritabanı sistemleri coğrafi verilerin depolanması ve işlenmesi için veri türlerini destekler hale getirilmiştir.

MVM'nin amacı, veritabanındaki mekansal etkileşimle birlikte gizli olan bilgiyi ortaya çıkarmaktır ve bu süreç; bir veritabanından otomatik veya yarı otomatik olarak ilgili ve anlaşılır bilgiler (kural, düzen, desen, ilişkiler vb.) elde etmeyi sağlayan geleneksel veri madenciliğinde olduğu gibidir (H. Cheng, 2008).

MVM, büyük ve karmaşık mekansal veriler üzerinde geleneksel mekânsal analiz (mekansal istatistik, analitik kartografi, keşifsel veri analizi gibi), istatistik ve veri madenciliği tekniklerinin (kümeleme, sınıflandırma, ilişkilendirme kural madenciliği, bilgi görselleştirme gibi) bir birleşimidir (Mennis & Guo, 2009) ve veri madenciliğinde olduğu gibi, bilgi keşfi için değişken seçimi, temizlik, ön işlem ve dönüşüm dahil olmak üzere birden çok adımı içeren yinelemeli bir süreçtir (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). MVM'nin teknolojik ve bilimsel gelişmeler doğrultusunda çalışma alanlarının giderek genişleyeceğini aşikardır, ancak günümüz literatüründe yaygın olarak kullanılan MVM metodları şu şekilde özetlenebilir;

- **Mekansal Sınıflama ve tahminleme:** Sınıflama, veri öğelerinin özelliklerine göre sınıf (kategorilere) gruplarına ayrılması işlemidir. Mekansal sınıflandırma yöntemleri, genel amaçlı sınıflandırma yöntemleri gibi yalnızca sınıflandırılacak nesnenin niteliklerini değil aynı zamanda komşu nesnelerin özelliklerini ve mekansal ilişkilerini de dikkate almak için geliştirilmiştir (Ester, Kriegel, & Sander, 1997). Mekansal sınıflandırma için görsel bir yaklaşım, geleneksel algoritma C4.5 ile türetilen karar ağacının sınıflandırma kurallarının mekansal yapılarını ortaya çıkarmak için harita görselleştirmesi ile birleştirilerek oluşturulmuştur (Andrienko & Andrienko, 1999).

Mekansal regresyon veya tahmin modelleri ise, mekansal otoregresif modeller (SAR) gibi belirli bir yerde bağımlı değişkeni tahmin etmede, yakındaki komşuların bağımsız ve/veya bağımlı değişkenini dikkate alan özel bir regresyon analizi grubu oluşturarak çalışılmaktadır.

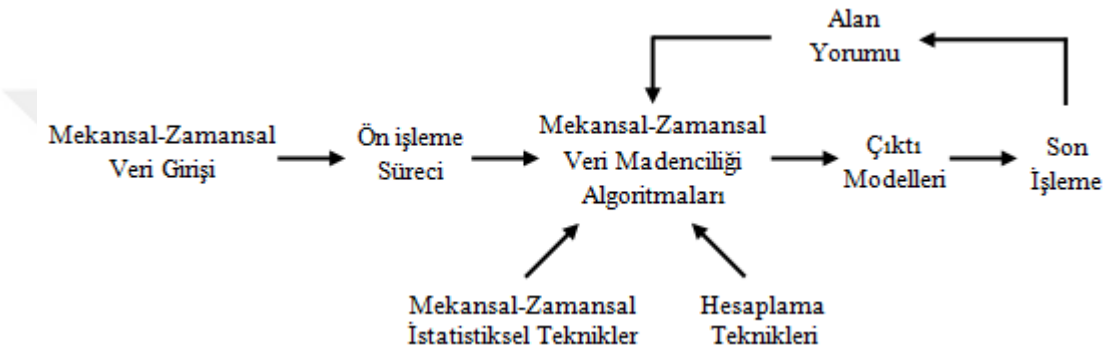
- **Mekansal Kümeleme:** Mekansal verilerin kümelenmesi, verilerin altta yatan yapısını tanımlamak için benzer mekansal ve mekansal olmayan özelliklerle birlikte gözlemleri bir araya getirmeyi amaçlamaktadır. Mekansal kümelenme, etkin (yoğun) noktaları keşfetmek için kullanılmaktadır. Buna örnek olarak bir şehirde suç faaliyetleri veya hastalıklar için etkin (yoğun) noktaların haritalanması verilebilir.
- **Mekansal İlişki Kuralları (Spatial Association Rule Mining):** Geleneksel ilişki kural madenciliğindenden farklı olarak, nesneler veya olaylar arasındaki mekansal ilişkilerin keşfedilmesi amaçlanmaktadır. "Amerikadaki en büyük şehirler kıyıya yakındır" cümlesi mekansal ilişki kuralına örnek olarak verilebilir.
- **Mekansal Aykırılık Tespiti (Spatial Anomaly Detection):** Genel popülasyondan istatistiksel olarak anlamlı farklılığa sahip olmamalarına rağmen, mekansal noktaların mekansal olmayan özellikleri arasında anlamlı farklılık olup olmamasının araştırılmasıdır.

3.3 Mekansal-Zamansal Veri Madenciliği

Mekansal bir veri matrisinin zaman bilgisini de içerdiği durumlarda, değişen veri yapısı sebebi ile mekansal veri madenciliğinde kullanılan tekniklerin bu yapıya uyarlanması gerekmektedir. Bu gereklilik ile birlikte, mekansal-zamansal veri üreten/depolayan kurum ve bilimsel araştırmaların sayısında artış gözlenmiş, ve yakın zamanda Mekansal-zamansal Veri Madenciliği (MZVM, Spatio-temporal Data Mining) kavramınının ortaya çıkmasına neden olmuştur.

MZVM, veri matrisindeki bilgiye ek olarak gözlemlere ait iki ya da daha fazla lokasyon ve gözlemlerin varoldukları ya da varolacakları zamanı da barındıran mekansal-zamansal veri matrisleri üzerinde uygulanmaktadır (Bittner, 2000). Bu tür veri yapısına örnek olarak meteorolojik istasyonlardan alınan günlük hava sıcaklığı ya da dünyadaki tüm başkentlerin aylık suç oranları verilebilir. Mekansal-zamansal veri matrisleri genellikle çok büyük boyutlardadır.

Şekil 3.3.3'de MZVM süreci gösterilmektedir. Bu süreçte ilk adım, veri setinin altında yatan mekansal zamansal dağılımı ortaya çıkarmak adına gürültü, hata, eksik verilerin saptandığı ve ayrıca tanımlayıcı mekan-zaman (space-time) analizlerinin yapıldığı ön işlem sürecidir. Ardından uygun mekansal-zamansal veri madenciliği algoritması ön hazırlığı yapılmış veriye uygulanır ve çıktı modelleri elde edilir. Mekansal-zamansal veri madenciliği algoritmaları genellikle istatistiksel temellere sahiptir ve ölçeklenebilir hesaplama tekniklerine entegre edilebilir. Son adım işlemleriyle iyileştirilen model çıktıları, gerekli görüldüğü takdirde sürece tekrar dahil edilir ve alanında uzman kullanıcı tarafından yorumlanır (Shekhar vd., 2015).



Şekil 3.3.3 Mekansal-Zamansal Veri Madenciliği Süreçleri (Shekhar vd., 2015).

Mekansal-zamansal veri madenciliği teknikleri karar mekanizması büyük, mekansal ve mekansal-zamansal veri setlerine dayanan kuruluşlar için oldukça önemlidir. Bu yapılardaki büyük verileri analiz eden kuruluşlara NASA, The National Geospatial-Intelligence Agency, The National Cancer Institute, The US Department of Transportation örnek olarak verilebilir. Buna ek olarak büyük mekansal-zamansal veri madenciliğinin kullanıldığı bazı alanlar ve toplumsal açıdan önemi aşağıdaki gibi sıralanabilir (Rao, Govardhan, & Rao, 2012).

- Meteoroloji: Sıcaklık yağış ve benzeri meteorolojik değişkenlerin gün ay yıl veya mevsimsel olarak analiz edilmesinde (tahmin, kümeleme vb.) kullanılır.
- Ekoloji ve çevresel düzenlemeler: Araştırmalardaki uzaktan algılama görüntülerini sınıflandırmak için kullanılır.
- Ulusal güvenlik: Suç analizlerinde suç oranı yüksek olan alanları harita üzerinden bulmaya ve bu sayede güvenlik görevlisi yoğunluklarını en etkin şekilde belirlemede kullanılmaktadır.

- Ulaşım: Bir yerden bir yere en hızlı şekilde ulaştırmayı sağlamak için kullanılır.
- Epidemioloji: Salgın hastalıkların olduğu bölgelerin risk ve salgın yoğunluklarının belirlenmesinde kullanılmaktadır. Doğru bitkiyi doğru yerde doğru zamanda yetiştirmek, varyasyonları değerlendirmek ve anlamak için kullanılır.
- Tarım: Verimlilik, toprak kalitesi ve sulama gibi değişkenlerin mevsimsel olarak analiz edilmesinde kullanılır.
- Biyoloji: Hayvan hareketleri, çiftleşme davranışları, türlerin yer değiştirmesi ve yok oluşlarını analiz eder.
- Jeoloji: Deprem ve volkanik hareketlilik tahminlerinde kullanılır.
- Sosyal Medya: Sosyal medya kullanıcıları bir içerik veya düşüncelerini paylaşırken bir zaman ve mekan bilgilerinide paylaşmaktadır. Bu bilgiler doğrultusunda kullanıcıların düşündükleri ve yaşadıkları MZVM ile analiz edilebilir.

MZVM üzerindeki literatür çalışmaları, istatistiksel temellere dayanan ve dayanmayan olmak üzere iki gruba ayrılmaktadır. İstatistiksel temellere dayanan literatürün başlıcaları, "Spatio-Temporal Clustering" isimli kitap Kisilevich tarafından yazılmıştır ve 2010 yılında literatüre kazandırılmıştır (Kisilevich, 2010). Aggarwal ve arkadaşları, "Outlier Analysis" isimli kitap içinde mekansal ve mekansal zamansal aykırı değer tespiti alanında bir bölüm hazırlamıştır (Aggarwal vd, 2013). Cheng ve arkadaşları ise mekansal-zamansal veri görselleştirmesinin yanı sıra, mekansal-zamansal otokorelasyon, mekansal-zamansal tahmin ve önkestirim, mekansal-zamansal kümeleme dahil olmak üzere birçok veri madenciliği alanı üzerinde birçok bilimsel çalışmalar yapmıştır (Cheng vd., 2014). İstatistiksel temellere dayanmayan çalışmalara örnek olarak ise Roddick ve arkadaşları mekansal ve mekansal-zamansal veri madenciliği için yaptıkları bibliyografik çalışma önemli bir konuma sahiptir (Roddick vd, 1996). Ayrıca Miller ve Han tarafından literatüre kazandırılmış "Geographic Data Mining and Knowledge Discovery" isimli kitap çalışması, en güncel istatistik temellere dayanmayan MZVM konularını içermektedir (Miller ve Han, 2009).

3.3.1 Mekansal-Zamansal Veri Türleri

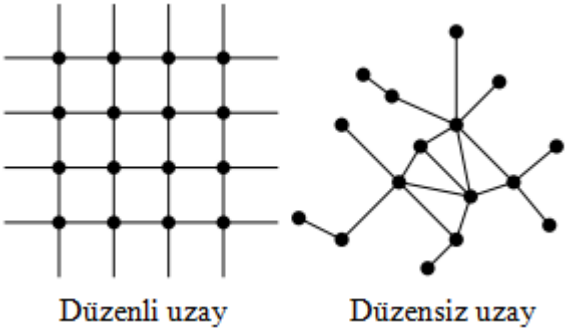
MZVM'de analiz edilen mekansal-zamansal veri seti farklı yapılarda olabilmektedir. Bu nedenle mekansal-zamansal bir veri setinin analizinden önce, veri seti yapısının iyice tanımlanarak uygun MZVM tekniklerini belirlemek oldukça önemlidir. Literatürde kabul görmüş mekansal-zamansal veri matrisi türleri; noktasal, hareketli, noktasal referans ve hücresel veri yapısı olmak üzere dört adettir (Atluri, Karpatne, & Kumar, 2017).

Nokta veriler (Event data): Bu yapıda bir mekansal-zamansal veri olayın ne zaman ve nerede meydana geldiğini gösteren bir nokta lokasyon ve zamandan meydana gelmektedir. Suçların gerçekleştiği yer ve zaman bilgileri bu veri yapılarına örnek olarak verilebilir.

Hareketli veri (Trajectory data): Hareketli veriler, haraket eden bir cisimden düzenli olarak alınan sinyaller ile izlediği güzergahı gösteren verilerdir. Başlangıç noktasından bitiş noktasına kadar olan taksi rotası, hayvanların göç sırasında izledikleri yol bu veri yapılarına örnek olarak verilebilir.

Nokta referans verileri (Point reference data): Nokta referans verileri, mekan ve zamanda bir dizi hareketli referans noktası üzerinde sıcaklık, bitki örtüsü veya popülasyon gibi sürekli ölçümlerinden oluşmaktadır. Bu veri tipine örnek olarak, hava sıcaklığı ve nem gibi parametrelerin havada uçan balonlar ile ölçülmesi ya da her lokasyon noktasında zamanla birlikte ölçümler yapan hareketli istasyonlar verilebilir. Benzer şekilde okyanus parametrelerini ölçen ve zamana göre lokasyonları değişen dubalardan elde edilen veriler de bu yapıyı barındırmaktadır.

Hücresel veri (Raster data): Hücresel veri yapıları, nokta referans verilerin aksine sabit bir noktadan (lokasyon) farklı zaman dilimlerinde sürekli veya kesikli ölçümlerden meydana gelmektedir. Oluşturulan noktaların ve zaman aralıklarının düzenli olması şart değildir. Mekansal-zamansal hücresel veriler günümüzde uzaktan algılama, iklim bilimi ve beyin görüntüleme gibi birçok alanda kullanılmaktadır.



Şekil 3.3.4 Hücresel Veriyi Temsil Eden Farklı Izgara Görünümleri.

Farklı mekansal-zamansal veri matrisleri üzerinde uygulanan MZVM teknikleri; mekansal zamansal tahminleme, mekansal-zamansal ilişki kural madenciliği, mekansal-zamansal sıralı desen madenciliği, mekansal-zamansal kümeleme ve sınıflama gibi teknikleri içermekle birlikte, farklı teknikler süregelen bilimsel araştırmalar ile uyarlanmaya çalışılmaktadır. Bunların haricinde ZMVM'nde görselleştirmenin önemi vurgulanmalıdır (T. Cheng, Haworth, Anbaroglu, Tanaksaranond, & Wang, 2014). MZVM teknikleri özetle aşağıdaki gibi gruplanabilir:

- **Mekansal-zamansal tahminleme:** Mekansal ve zamansal veriyi kullanarak bir tahmin edici model oluşturur ve herhangi bir zaman ve mekan için bir değişkeni tahmin eder. Çoğu mekansal-zamansal tahminleme yöntemi, mekansal ya da zaman serileri tahminlemesine dayanmaktadır (H. Cheng, 2008). Mekansal-zamansal tahminlerde verideki otokorelasyonlar, mekansal farklılıklar ve zamansal hareketlilikler kullanılmaktadır. Spatio-temporal Autoregressive Regression (STAR) modeli, farklı yerlerde değişkenler arasındaki zamansal ve mekansal bağımlılığı daha açık bir şekilde modelleyerek SAR'ın geliştirilmiş yapısıdır (Cressie, 2015). Spatiotemporal Kriging modeli, gözlemlerin bilindiği yerlere dayanarak gözlemlerin bilinmediği yerlerde öngörülerde bulunmak için bir tekniktir (Cressie, 2015).. Hierarchical Dynamic Spatiotemporal Models (DSMs) (Cressie and Wikle, 2015), Adından da anlaşılacağı gibi bir mekansal süreci dinamik olarak bir Bayes hiyerarşik çerçevesi ile modellemeyi amaçlamaktadır. Bu modellerin yanı sıra son zamanlarda yükselen bir şekilde kullanıcılar heterojen yapıda lineer olmayan ve çok ölçekli özelliklere sahip mekansal-zamansal veri setlerinde geleneksel yöntemler yerine makine öğrenmesi ve veri madenciliği teknikleri gibi teknikleri daha çok kullanmaktadır. Yapay sinir ağları ve karar

destek vektörleri zamansal-mekansal tahminlemede başarılı olan yeni yöntemlere örnek olarak gösterilebilir (T. Cheng vd., 2014).

- **Mekansal-zamansal ilişki kural madenciliği (spatio-temporal association rule) :** Genel olarak belirli parametreler arasında ilişki olduğunu ve birinde yaşanan değişimin diğerinde yaratacağı değişimin göstergesidir. Mekansal ve zamansal verilerde mekan ve zaman için bu birliktelik kurulur. Örneğin iki bölge toprak yapısındaki zamana bağlı olarak toprağın kimyasal parametreleri (tuzluluk (EC, pH, Na), iz element (Mn, Zn, Fe, Cu) ve ağır metal (Pb,As, Cd, Ni, Cr, Co) parametreleri gibi) ile bir ilişki kuralı oluşturulur. Benzer şekilde zamana göre iki bölge arasındaki meteorolojik verileri ele alarak iklim tipleri kuralları oluşturulabilir ve bu sayede tahminleme yapılabilir.
- **Mekansal-zamansal sıralı desen madenciliği:** Veri setinde belirli bir sıraya uyarak tekrar tekrar meydana gelen olayları ortaya çıkarmaktadır. Mekansal-zamansal sıralı desen madenciliği algoritmalarında kullanılan algoritmaların ana fikri belirli bir olay belirli birinci bölgede gerçekleşiyorsa, belirli bir zaman sonrasında da ikinci bölgede gerçekleşecek düşüncesidir. Bir nevi kural madenciliğide denilebilir. Örneklendirmek gerekirse meteorolojide belirlli parametreler ışığında A bölgesinde yaşanan yağıştan belirli bir süre sonra B bölgesindede yağış yaşanacaktır. Benzer bir şekilde C bölgesinde yaşanan trafik sıkışıklığından belirli bir zaman sonrasında D bölgesindede trafik sıkışacaktır.
- **Mekansal-zamansal Kümeleme:** Kümeleme, önceden bir bilgi bulunmadan birbirine benzer karakteristik özellikte olan gözlemleri birleştirmeyi içerir. Ana amacı birleştirilen kümeler içi benzerliği maksimum kümeler arası benzerliği ise minimum yapmaktır. Sınıflama, segmentasyon ve aykırı değer ayıklama gibi işlemlerde kullanılmaktadır. Mekansal-zamansal kümeleme algoritmaları, mekansal kümeleme algoritmalarının geliştirilmiş yapıları ile analiz edilmektedir. Mekansal ve zamansal bilgiler barındırdığı için bu tarz algoritmaları geliştirmek oldukça zordur. Örnekle anlatmak gerekirse hareketli nesneler analiz edilirken bir kümenin lokasyon bilgileri her harekette değişmektedir fakat küme aynıdır. Mekansal-zamansal kümeleme algoritmaları birçok alanda elde var olan bilgiyi anlamaya yardımcı olmakta ve kolaylaştırmaktadır. Örnek olarak bir hastalık salgını ile ilgili mekansal-

zamansal veriyi kümelemek ve epidemiyologların ve tıp uzmanlarının anlayabileceği şekilde görselleştirmek verilebilir.

- **Mekasal-zamansal sınıflama:** Bir eğitimli öğrenme (supervised) tekniğidir. Model oluşturma ve oluşturulan modeli kullanma olmak üzere iki adımdan meydana gelmektedir. Yaygın olarak karar ağaçları (Decision Tree), Sinir ağları (Neural Networks) ve genetik algoritmalar (Genetic algorithms) kullanılan sınıflama algoritmalarıdır. Mekansal-zamansal sınıflama yöntemlerinin normal sınıflama yöntemlerine göre farkı, örneğin sinir ağları algoritmasında girdi katmanı bir gözlem değerini alır ve bağlantı ağırlıklarını ve geriye doğru yayılım gösteren tekniklerde ise hatayı hesaplar. Mekansal-zamansal verilerde ise işleme alınan gözlem farklı zaman dilimlerindeki lokasyonları içerir. Burada her zaman dilimi bir girdi olarak alınır.

4. MEKANSAL KÜMELEME

Mekansal olarak birbirine yakın olan gözlemler uzak olan gözlemlere göre birbirine daha çok benzerlik göstermektedir (Tobler, 1970). Bu nedenle birbirine yakın konumlardaki gözlemlerin uzak olan gözlemlere göre daha fazla benzerlik göstermesi beklenen bir durumdur. Mekansal kümeleme, birbirine yakın olan ve benzer özellikler gösteren gözlemleri aynı kümeye atama sürecidir. Burada kümeler farklı örneklem lokasyonları arasındaki benzerlik derecelerine dayanmaktadır. Diğer bir deyişle bir gözlemi kendine en çok benzeyen fakat diğer kümelere benzerliği en az olan gözlemlerle aynı kümeye koyar ve bu sayede kümeler içi benzerliği maksimum seviyeye çıkarırken kümeler arası benzerliği ise minimum seviyeye indirir. Mekansal kümeleme, mekansal veri kümelerini araştırarak yoğunluk bulunan bölgelerin keşfedilmesinde yardımcı olur (Shekhar, Zhang, & Huang, 2010). Uygulama yapısına bağlı olarak mekanda, zamanda ya da mekan ve zamandaki gözlemlerin birbiriyle olan yakınlıklarını mekansal (enlem boylam) ve mekansal olmayan değişkenler ile incelemektedir. Mekansal kümelemede gruplamaya neden olan etken ve işlem öncesinde kaç grup olacağı bilinmediği için bir eğitimsiz öğrenme (unsupervised) tekniğidir. Bir mekansal kümeyi tanımlamak için ilk önce hangi tür veri yapısıyla çalışıldığı incelenmelidir. Mekansal veri yapıları nokta, çizgi, alansal (polygon) ve kompleks olmak üzere dört farklı yapıdadır ve bölüm 2.2'de bu veri yapıları tanımlanmıştır (Yavuzoğlu, 2009). Bunların arasında en sık kullanılanı noktasal veri yapısıdır. Bir mekansal nesne veya bir olayın konumu, koordinatları vasıtasıyla bir nokta olarak gösterilir. Noktasal veriye örnek olarak binaların veya suç olaylarının konumu gösterilebilir. Mekansal kümeleme analizi, coğrafi varyasyon kalıplarının sayısallaştırılmasında önemli bir rol oynamaktadır. Genellikle hastalık gözetimi, mekansal epidemiyoloji, popülasyon genetiği, peyzaj ekolojisi, suç analizi ve daha birçok alanda kullanılır (Yavuzoğlu, 2009). En bilinen mekansal kümeleme yöntemleri; Bölümlemeye dayalı, Hiyerarşik, Yoğunluğa dayalı, Izgara tabanlı, Bulanık kümeleme, Yapay sinir ağları, Özdüzenleyici haritalar ve genetik algoritmalar olarak bilinmektedir.

4.1 Bölümlemeye Dayanan Kümeleme Yöntemleri

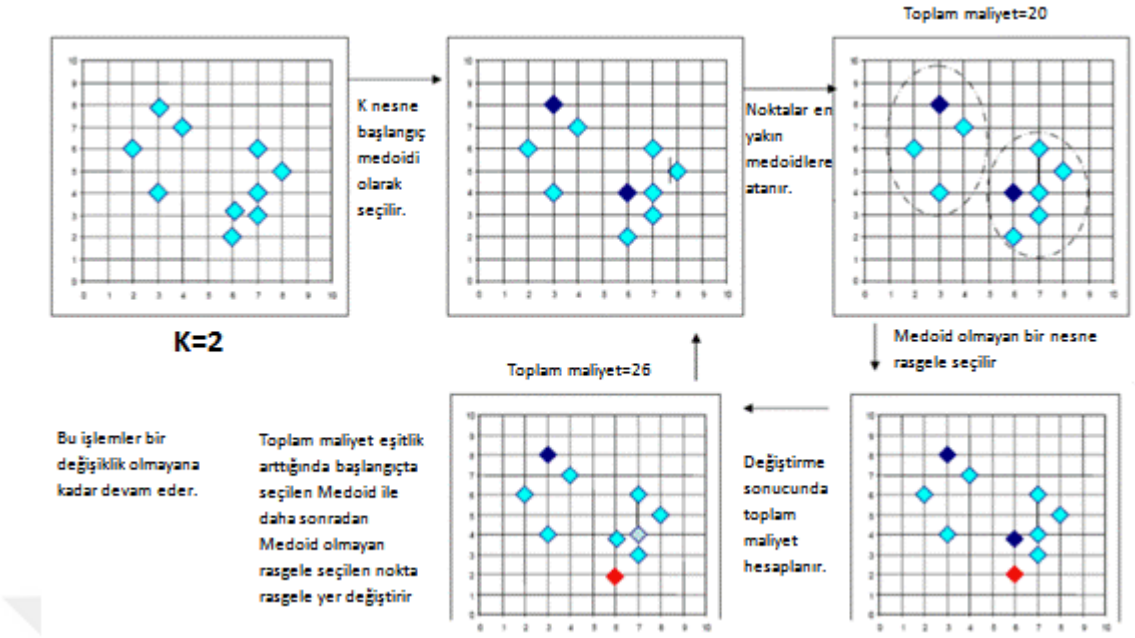
Bölümlemeye (partitioning) dayanan kümeleme yöntemleri, n gözlemli bir veri matrisini k ($k \leq n$) tane parçaya bölmeyi amaçlayan ve bunu yaparken aşağıda belirtilen varsayımların aynı anda geçerli olmasını gözeten kümeleme algoritmalarıdır.

- Her grup en az bir gözleme sahip olmalıdır
- Her gözlem bir gruba ait olmalıdır

Bölümleme algoritmalarında ilk adım veri setinin küme sayısına bölünerek başlanmasıdır ve algoritmalar iteratif bir şekilde çalışarak hareketli objelerin en doğru kümeye gelmesini hedefler. Objeler yakınlık ve uzaklık ölçülerine göre kümelere atanır. Bölümleme algoritmalarından en bilinenleri K-Means, K-Medoids (PAM), CLARANS ve EM algoritmalarıdır.

4.1.1 PAM

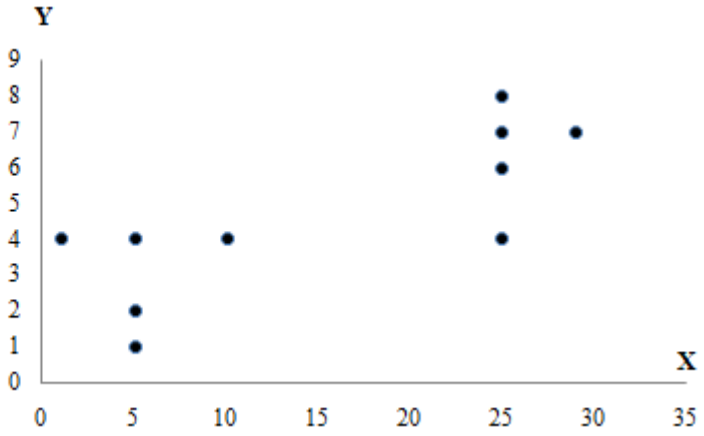
PAM (Partitioning Around Medoisds) algoritması ilk olarak Kaufman ve Rousseeuw tarafından 1990 yılında geliştirilmiştir (Kaufman & Rousseeuw, 1990). Literatürde K-medoids algoritması olarak da adlandırılmaktadır. PAM algoritması mekansal kümelemede güçlü bir algoritmadır ancak büyük veri setleri üzerinde uygulandığında yavaş çalışarak zaman kayıplarına neden olmaktadır (Yue, Mao, Li, & Zou, 2014). PAM, veri setindeki gözlemler arasında k adet temsilci gözlemi aramaya dayanan bir yöntemdir ve bu temsilcilerin veri setindeki mümkün olduğunda farklı yapıları temsil etmesi kümeleme analizinin başarısını etkiler. Belirlenen temsilci gözlemler, genellikle merkez tipler (centrotypes) olarak adlandırılır. PAM algoritmasında ise temsici gözlemler kümelerin medoidi olarak adlandırılır (Kaufman ve Rousseeuw, 1987). Bir dizi k temsilci gözlem bulduktan sonra k adet küme, veri setinin her bir nesnesini en yakın temsilci gözleme atayarak oluşturulur. PAM algoritmasının işleyişi Şekil 3.2'de gösterilmiştir.



Şekil 4.4.1 PAM Kümeleme Yöntemi.

PAM in işleyişini daha ayrıntılı olarak göstermek amacıyla, Şekil 4.2'de gösterilen örnek veri setini ele alalım. Bu veri seti X ve Y adında 2 değişken ve 10 gözlemden meydana gelmektedir.

Nesnelerin koordinatları		
Numara	X	Y
1	1	4
2	5	1
3	5	2
4	5	4
5	10	4
6	25	4
7	25	6
8	25	7
9	25	8
10	29	7



Şekil 4.4.2 10 Gözlemli 2 Değişkenli Veri Seti Örneği.

Bu veri setinin iki alt boyuta, yani iki kümeye bölünmesi gerektiği varsayılsın. PAM öncelikle iki temsilci gözlemini belirler ve ardından bu temsili gözlemlerin çevresinde kümeler oluşturulur. Örnek olarak 1. ve 5. gözlemler temsilci gözlem olarak kabul edildiğinde, PAM, Çizelge 4.1.1'de verilen tüm gözlemlerin bu 2 adet temsilci gözlemlere olan benzersizlik değerlerini hesaplar. Gözlemler bu benzersizlik değerlerinin en küçük olanına karşılık gelen temsilci gözlemin kümesine atanmaktadır.

Çizelge 4.1 Gözlemlerin Benzersizlik Değerleri, Birinci ve Beşinci Temsilci Gözlem.

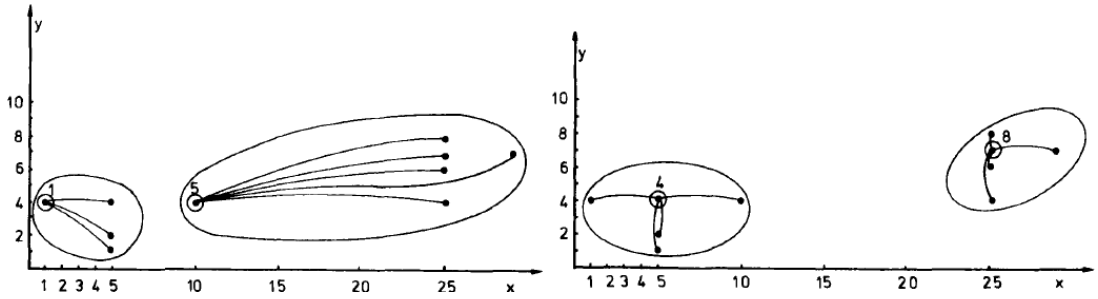
Gözlem	Gözlem 1	Gözlem 5	Minimum	En Yakın
1,00	0,00	9,00	0,00	1,00
2,00	5,00	5,83	5,00	1,00
3,00	4,47	5,39	4,47	1,00
4,00	4,00	5,00	4,00	1,00
5,00	9,00	0,00	0,00	5,00
6,00	24,00	15,00	15,00	5,00
7,00	24,08	15,13	15,13	5,00
8,00	24,19	15,30	15,30	5,00
9,00	24,33	15,52	15,52	5,00
			Ort:9,37	

Çizelgeden görüleceği gibi ortalama benzersizlik değeri 9,37 olarak hesaplanmıştır ve bu değer kümelerin sıkılık derecesini bir başka deyişle kümelemenin kalitesini belirtmektedir. Temsilci gözlemlerin 4 ve 8 numaralı gözlemler olarak seçilmesi durumunda ise benzersizlik değerleri Çizelge 4.1.2'deki gibi hesaplanacaktır.

Çizelge 4.2: Gözlemlerin Benzersizlik Değerleri, Dördüncü ve Sekizinci Temsilci Gözlem.

Gözlem	Gözlem 4	Gözlem 8	Minimum	En Yakın
1,00	4,00	24,19	4,00	4
2,00	3,00	20,88	3,00	4
3,00	2,00	20,62	2,00	4
4,00	0,00	20,22	0,00	4
5,00	5,00	15,30	5,00	4
6,00	20,00	3,00	3,00	8
7,00	20,10	1,00	1,00	8
8,00	20,22	0,00	0,00	8
9,00	20,40	1,00	1,00	8
			Ort:2,30	

Bu durumda temsilciler için ortalama benzersizlik değeri 2,30 olacaktır ve bu değer 1 ve 5 numaralı temsilci gözlemlerin 9,37 ortalama değerine göre daha küçüktür. Buradan da anlaşılacağı üzere kümelemeye başlamadan önce seçilen temsilci gözlemlerin kümeleme kalitesi üzerinde anlamlı bir etkisi bulunmaktadır. Farklı 2 grup temsilci kullanılarak PAM ile oluşturulan kümeler, temsilci seçiminin önemini vurgulamak adına Şekil 4.4.3'de gösterilmiştir.



Şekil 4.4.3 Temsilci Gözlemlerin Seçimine Göre Kümeleme Sonuçları.

Yukarıda bahsedildiği gibi amaç ortalama benzersizlik değerlerin minimum olmasıdır, bu amaç algoritmaya yansıtılırken, benzersizliklerin toplamının minimum olması kriteri gözetilir. Bu iki amaç matematiksel olarak aynı sonucu doğurmaktadır, ancak algoritmik açıdan benzersizliklerin toplamının minimum olması hesaplama kolaylığı sağlamaktadır.

PAM algoritması iki aşamadan oluşmaktadır. İlk aşamaya inşa (build) adı verilmektedir ve bu adımda k gözlem bulunana kadar temsilci nesnelerin art arda seçilimi ile bir öncü kümeleme yapılır. İlk gözlem, diğer tüm gözlemler arasındaki benzersizliklerin toplamının mümkün olduğunca küçük olmasını sağlayan gözlemdir. Bu gözlem, gözlemler kümesinin en merkezi konumunda bulunmaktadır. Daha sonrasında en iyi sonuca ulaşılana kadar her adımda başka bir gözlem seçilir. Bu gözlemin bulunması aşağıdaki adımlarla gerçekleşmektedir (Kaufman & Rousseeuw, 1990)

1. i ve j daha seçilmemiş birer gözlem olmak üzere
2. j için ona en benzer olan bir önceki seçilmiş gözlem ile olan benzersizliği D_j ile gözlem i ile olan benzersizliği d(j,i) arasındaki fark hesaplanır.
3. Eğer bu fark pozitif ise j gözlemi i gözleminin seçilmesine katkı sağlayacağından dolayı seçilmemiş i gözlemi için maliyet hesaplanır.

$$C_{j,i} = max\left(D_j - d(j,i), 0\right) \quad (4.1)$$

4. Seçilen gözlem i için toplam maliyet hesaplanır.

$$\sum_j C_{ji} \quad (4.2)$$

5. Toplam maliyeti maksimum yapan seçilmemiş i gözlemi seçilir.

$$\max_i \sum_j C_{ji} \quad (4.3)$$

Bu süreç k gözlem bulunana kadar devam etmektedir. Algoritmanın ikinci aşamasında seçilen temsilci gözlemler kümelemeyi geliştirmeyi amaçlamaktadır. Bu aşama seçilmiş gözlem olan i ve seçilmemiş gözlem olan h gözlem çifti kullanılınarak sürdürülür. Burada seçilmiş olan gözlem ile seçilmemiş olan gözlemin yer değiştirmesinin kümeleme değeri üzerine etkisi araştırılır. K adet temsilci gözlem tarafından belirtilen kümeleme değeri temsilci gözlemler ile diğer tüm gözlemler arasındaki benzersizlik toplamı olarak tanımlanmaktadır. Seçilen gözlem i ile seçilmeyen h arasındaki değişimin kümeleme üzerindeki etkisi aşağıdaki hesaplama adımlarıyla bulunmaktadır (Kaufman & Rousseeuw, 1990).

1. Seçilmemiş bir gözlem olan j'nin değişimdeki maliyete olan katkısı C_{jih} hesaplanır.

a) Eğer j'nin i ve h gözlemlerine olan uzaklığı diğer temsilci gözlemlerinkinden daha büyükse, C_{jih} sıfırdır.

b) Eğer j'nin i gözlemine olan uzaklığı diğer tüm temsili gözlemlerinkinden daha uzak değil ise ($d(j,i)=D_j$) iki durum ortaya çıkmaktadır.

(1) J gözlemi h gözlemine ikinci en yakın temsilci gözlemden daha yakındır.

$$d(j,h) < E_j \qquad (4.4)$$

Burada E_j, j gözlemi ile en yakın ikinci temsilci gözlem arasındaki benzersizliktir. Bu durumda j gözleminin i ve h gözlemleri arasındaki değişime olan katkısı aşağıdaki denklemle hesaplanmaktadır.

$$C_{jih} = d(j,h) - d(j,i) \qquad (4.5)$$

(2) J gözlemi, en azından ikinci en yakın temsilciye göre h gözleminden uzaktır.

$$d(j,h) \geq E_j \qquad (4.6)$$

Burada j'nin değişime olan katkısı aşağıdaki gibidir.

$$C_{jih} = E_j - D_j \qquad (4.7)$$

Dikkat edilmelidir ki b(1) maddesinde C_{jih} değeri j,h ve i gözlemlerinin konumuna bağlı olarak pozitif veya negatif olabilmektedir. Sadece eğer j gözlemi i gözlemine h gözleminden

daha yakın ise katkı pozitif olmaktadır ve bu durum değişimin j gözlemi açısından uygun olmadığını göstermektedir. Diğer bir taraftan b(2) maddesinde katkı her durumda pozitiftir. Bunun nedeni i gözlemi ile j gözlemi arasındaki uzaklığı ikinci en yakın temsilciden daha fazla olan h gözleminin yer değiştirmesinin bir avantajı olmayacaktır.

c) j gözleminin i gözlemine olan uzaklığı, en azından herhangi bir temsilci gözleminin uzaklığından daha büyüktür. Fakat j gözleminin h gözlemine diğer tüm temsilci gözlemlerden daha yakındır. Bu koşullarda j gözleminin değişime olan katkısı aşağıdaki denklemle belirlenir.

$$C_{jih} = d(j,h) - D_j \quad (4.8)$$

2. Tüm hesaplanan katkıları ekleyerek değişimin toplam değeri hesaplanır.

$$T_{ih} = \sum_j C_{jih} \quad (4.9)$$

Bundan sonraki adımlarda değişimin gerçekleştirilip gerçekleştirilmeyeceği kararına varılır.

3. T_{ih} değerinin minimum yapan (i,h) gözlem çifti seçilir.

4. Eğer minimum T_{ih} negatif bir değer ise, değişim gerçekleştirilir ve algoritma 1. adıma geri döner. Eğer T_{ih} değeri pozitif veya 0 ise kümeleme değeri değişim yapılarak daha çok küçültülemeyeceğinden dolayı algoritma durdurulur.

Potansiyel tüm takasların dikkate alınması nedeniyle, algoritmanın sonuçları girdi dosyasındaki gözlemlerin sırasına bağlı değildir (bazı gözlemler arasındaki uzaklıkların bağlı olmamak şartıyla). PAM algoritmasının genel işleyişinin özet olarak aşağıdaki adımlarla gösterilebilir (Silahtaroğlu, 2004).

1. K adet temsilci gözlem seçilir (rasgele veya yukarıdaki adımlarla).

2. İ seçilmiş h seçilmemiş gözlem iken, tüm i,h nesne çiftleri için T_{ih} değerleri hesaplanır.

3. T_{ih} değerinin minimum yapan (i,h) gözlem çifti seçilir. Eğer minimum T_{ih} negatif bir değer ise, değişim gerçekleştirilir ve 2. Adıma geri dönülür.

4. Değişiklik olmayana kadar Adım 3 ile Adım 6 arası işlemler tekrarlanır ve algoritma durdurulur.

4.1.2 CLARANS

Örnekleme tekniğini PAM ile birleştiren bir başka algoritma ise CLARANS algoritmasıdır. CLARANS (Clustering Large Aplications based on RANdomized Search) algoritması 1994 yılında VLDB'94 konferansında Raymond T. Ng ve Jiawei Han tarafından ilk kez sunulmuştur. Algoritmanın amacı, CLARA algoritmasını geliştirmek, güvenilirliğini ve ölçeklenebilirliliğini arttırmak ve veri seti içinde olabilecek mekansal yapıları tanımlamaktır (Akın, 2008). Sadece noktasal değil poligon veri tiplerini barındıran mekansal veri setlerinde de bu algoritmanın iyi çalıştığı ve ayrıca mekansal ve mekansal olmayan değişkenler arasındaki ilişkiyi tanımlaması algoritmanın özelliklerindendir (Ng & Han, 2002).

CLARA algoritması kümeleme yaparken araştırma süresince sabit bir küme kullanmaktadır. CLARA'nın aksine, CLARANS herhangi bir zamanda herhangi bir örneği sınırlamaz. CLARANS araştırmanın her aşamasında rasgele bir örneklem oluşturur. Kavramsal olarak kümeleme işlemi her bir düğümün potansiyel bir çözüm olduğu (k medoidler dizisi) $G_{n,k}$ grafiği aracılığıyla bir arama olarak görülebilir. Burada n gözlem sayısı iken k temsilci sayısını ifade etmektedir. Düğüm ise k adet temsilcilerin toplamı olarak tanımlanmakta ve her bir düğüm bir kümeye karşılık gelmektedir. $G_{n,k}$ grafiği temsilcilerden oluşmaktadır. Grafikteki bir düğüm $\{O_{m1}, O_{m2, \dots}, O_{mk}\}$ k adet gözlemler topluluğundan meydana gelmektedir. $O_{m1}, O_{m2, \dots}, O_{mk}$ seçilen temsilcilerdir. İki düğüm arasında sadece bir gözlem farkı var ise bu iki düğüm komşudur. Yani $S_1=\{O_{m1}, O_{m2, \dots}, O_{mk}\}$ ile $S_2=\{O_{m1}, O_{m2, \dots}, O_{mk}\}$ düğümlerinin komşu olabilmeleri için $|S_1 \cap S_2| = k - 1$ şartının sağlanması gerekmektedir. Burada her bir düğümün k(n-k) komşusu olduğu açıktır. Bir düğüm k adet temsilciler topluluğundan oluştuğu için her düğüm bir kümeleme anlamına gelmektedir. Bu nedenle her bir düğüme gözlemler ile bulunduğu kümenin temsilciler arasındaki toplam benzersizliği belirten bir maliyet atanabilir. CLARANS algoritmasının yapısı aşağıda adımlar halinde gösterilmektedir (Ng & Han, 2002).

1. Girdi parametresi olarak maksimum komşu sayısı ve bölgesel miktar girilir. Maksimum komşu sayısı incelenecek komşu sayısının üst limitini, bölgesel miktar ise elde edilecek bölgesel minimum nokta sayısının alt sınırını göstermektedir. i=1'den başlatılır ve minimum maliyeti büyük bir değer olarak belirlenir. Maksimum komşu sayısı parametresi ne kadar büyük olursa,

CLARANS algoritması o kadar PAM'a yaklaşır ve bölgesel minimum arama süresi uzar.

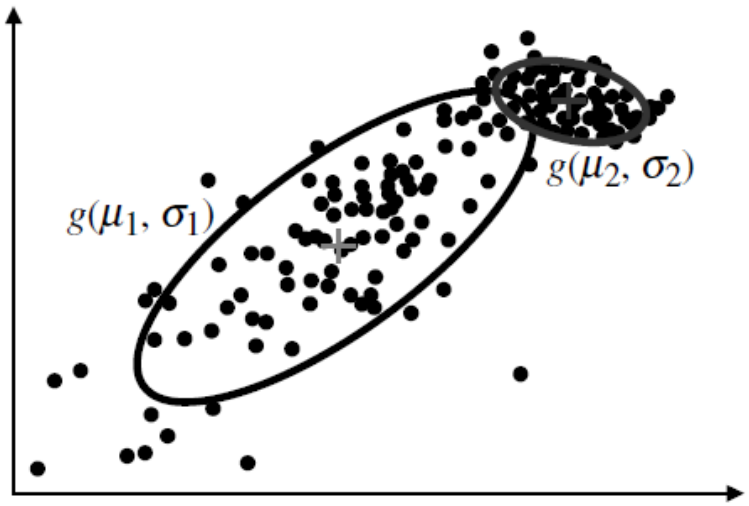
2. $G_{n,k}$ keyfi bir düğüm belirlenir.
3. j = 1 den başlatılır.
4. Mevcut düğümün rasgele bir komşu düğümü S seçilir ve maliyet hesaplanır.
5. Eğer S'nin maliyeti mevcut düğümün maliyetinden daha düşük ise S düğümü mevcut düğüm olarak atanır ve adım 3'e dönülür.
6. S maliyet değeri mevcut düğümden daha fazla ise j bir artırılır. Eğer j ≤ maksimum komşu sayısı ise adım 4'e dön.
7. Eğer j > maksimum komşu sayısı ise minimum maliyet ile mevcut olan maliyeti karşılaştır. Eğer eski maliyet minimum maliyetten daha az ise minimum maliyet mevcut maliyet olarak belirlenir.
8. i'yi bir artır. Eğer i > yerel miktar ise algoritma durdurulur ve en iyi düğüme ulaşılmış olur. Diğer durumlarda ise adım 2'ye geri dönülür.

PAM algoritması her adımda mevcut düğümün komşularını inceler. Mevcut düğüm daha sonra maliyeti en düşük olan komşusu ile değiştirilir ve araştırma minimum bulununcaya kadar devam eder. Örneğin büyük n ve k değerleri (n=1000, k=10) ile çalışıldığında bir düğüm için k(n-k) komşuyu incelemek çok fazla zaman kaybettirecektir. Bu durum PAM'ın büyük veri setleri için uygun olmadığını göstermektedir. Öte yandan, CLARA daha az komşu ile inceleme yapmaktadır. Daha açıklayıcı bir ifadeyle CLARA, tüm veri kümesinin bir örneği üzerinde çalıştığından, komşuları daha az incelemekte ve aramayı orijinal grafikten daha küçük olan alt grafiklere kısıtlamaktadır (Ng & Han, 2002). Yani sabit bir örneklem kullandığı için aranılan minimum nokta o örnek içinde olmayabilmekte ve bu büyük bir dezavantaj yaratmaktadır. CLARANS ise bir araştırmanın her adımında dinamik olarak komşuların rastgele bir örneğini çizer. Rastgele örneklenecek komşuların sayısı kullanıcı tarafından belirlenen bir parametre ile sınırlandırılmamıştır. Bu sayede CLARANS aramayı bölgeselleştirilmiş bir alanla sınırlandırmaz. Daha iyi bir komşu bulunursa (yani daha düşük hata veriyorsa) CLARANS komşunun düğümüne taşınır ve işlem tekrar başlar. Aksi halde mevcut kümeleme bölgesel bir minimum oluşturur ve CLARANS yeni bir bölgesel minimum aramak için yeni rastgele seçilmiş düğümlerle başlar. Kullanıcıların belirlediği sayıda bölgesel minimum bulunursa algoritma en iyi bölgesel minimum değerini vermiş olur. CLARANS'ın deneysel

olarak PAM'dan daha verimli olduğu belirtilmektedir. Buna ek olarak aynı sürede verilen CLARANS'ın CLARA'dan daha kaliteli kümelenme sonuçları üretmektedir (Harvey J. Miller & Jiawei Han, 2009).

4.1.3 Beklenti-Maksimizasyonu

Beklenti makzimizasyonu(Expectation Maksimation - EM), eksik veri seti olduğu durumlarda verinin dağılımının parametrelerinin tahmin edicilerini en çok olabilirlik yöntemiyle tahmin eden genel bir algoritmadır. Beklenti-makzimizasyonu algoritması, k-ortalama algoritmasının geliştirilmiş bir şekli olarak düşünülebilir fakat k-ortalama algoritmasından farklı olarak algoritma her gözlemi üyelik olasılığını gösteren bir ağırlığa göre bir kümeye atar. Başka bir deyişle kümeler arasında kesin bir sınır yoktur. Mekansal verilerile çalışıldığı durumlarda, komşu olmayan gözlemleri birbirine cezalandıran komşu-beklenti-makzimizasyonu (Neighborhood-Expectation-Maximization, NEM) yöntemi önerilmektedir. Algoritma ortalama ve kovaryansları içeren parametreleri tahmin ederek başlamaktadır. Ardından beklenti adımı (expectation) ve en büyükleme (maximization) adımları olmak üzere iki adımı bulunmaktadır. Beklenti adımında (expectation), önceki olasılıkları hesaplamak için var olan parametreler kullanılmaktadır. En büyükleme (maximization) adımında ise güncellenen olasılıklar vasıtasıyla log-olabilirlik fonksiyonu maksimum yapılır ve ortalama ve kovaryanslar güncellenir. Algoritma beklenti ve en büyükleme adımlarında yakınsama şartı sağlanana kadar döngüsel olarak devam ettirilir.

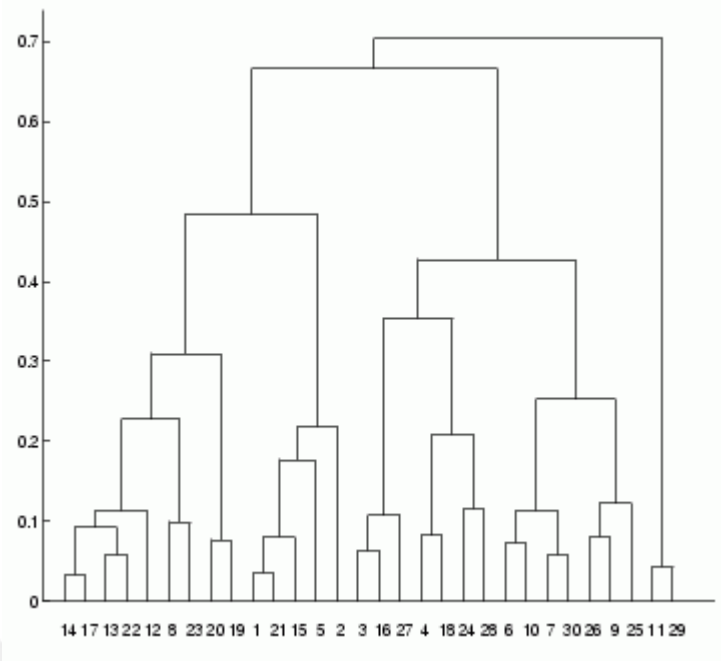


Şekil 4.4.4 EM Metodu Küme Gösterimi (İki Adet Gaussian Dağılımına Sahip Küme Bulunmaktadır).

EM algoritması, kümelemeye ilk olarak karışım modelinin parametrelerinin tahmin edilmesi ile başlar. Nesneler, parametreler tarafından üretilen karışım yoğunluğuna karşı iteratif olarak tekrar puanlandırır. Daha sonra puanlanan nesneler, parametre tahminlerini güncellemek için kullanılır.

4.2 Hiyerarşik Kümeleme Yöntemleri

Hiyerarşik kümeleme metotlarında nesneler hiyerarşik bir düzende gruplanmakta ve bir küme ağacı oluşturulmaktadır. Bu yöntemde, bir özellikten yola çıkarak aşamalı bir şekilde alt kümeler elde edilebilmektedir. Bu yöntemler verilerden özet bilgiler çıkarmada ve herkes tarafından anlaşılması daha kolay olan görselleştirme yöntemlerinde kullanılabilmektedir ve yaygın olarak kullanılan görselleştirme aracı dendrogramdır (Han vd., 2012). Hiyerarşik kümeleme yöntemleri, hiyerarşik ayrışmanın aşağıdan yukarıya (birleştirme) veya yukarıdan aşağıya (bölme) oluşup oluşmadığına bağlı olarak birleştirici (Agglomerative) veya bölücü (divisive) olarak sınıflandırılabilir. Hiyerarşik kümeleme algoritmalarında başlangıçta kullanıcı tarafından k (küme sayısı) parametresinin belirlenmesini istemesi gibi parametrelere ihtiyaç duyulmamaktadır. Fakat ne zaman birleştirme veya bölme işlemi sona erdirileceğini belirten bir sonlandırma koşulunun tanımlanması gerekmektedir. Hiyerarşik kümeleme yöntemi verideki nesneleri küme ağacına gruplayarak çalışır. Hiyerarşinin kökü kümelenecek olan veri gözlemlerinin tümünü temsil etmektedir. Ağacın her bir seviyesinde kümelere karşılık gelen düğümler oluşmaktadır. Hiyerarşinin her bir seviyesi bazı küme setlerine karşılık gelmektedir. Hiyerarşinin tabanı ağacın yapraklarından yani tekli noktalardan oluşmaktadır. Bu küme hiyerarşisine dendogram adı verilmektedir. Hiyerarşik kümeleme yöntemlerinin en temel avantajı, herhangi bir seviyede hiyerarşiye son vererek uygun kümeler elde edilebilmektedir (Pasin, 2015).



Şekil 4.4.5 Basit Bir Dendrogram Örneği.

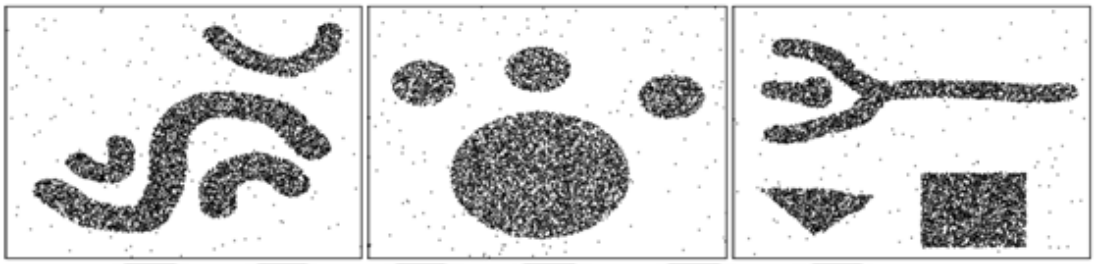
Hiyerarşik kümeleme uygularken karşılaşılacak bazı zorluklar vardır. Hiyerarşik kümeleme yöntemi basit olmasına rağmen genellikle birleştirme veya bölme noktaları seçimiyle ilgili zorluklarla karşılaşılabilir. Böyle bir karar önemlidir çünkü bir grup gözlem birleştirilmiş veya bölünmüşse bu durum sonraki aşama süreçlerinde yeni oluşturulan kümeleri de etkileyecektir. Daha önce yapılanları geri alınamaz ya da kümeler arasında gözlem değiştirme gerçekleştirilemez. Bu nedenle bir adımda iyi seçilmemiş olsa dahi birleştirme veya bölünmüş kararlar düşük kaliteli kümelenmelere neden olabilir. Dahası yöntem güzel ölçeklenemez çünkü gözlem ve küme sayısını doğru karar vermek için birleştirme veya bölme çok ayrıntılı incelenmelidir. Hiyerarşik kümelenmenin kalitesini yükseltmek için hiyerarşik kümelemeyi diğer kümeleme teknikleriyle birleştirmek bir yöntem olarak kullanılabilir.

4.3 Yoğunluğa Dayalı Kümeleme Yöntemleri

Kümeleme yöntemlerinin birçoğu gözlemler arasındaki uzaklığın hesaplanmasına dayanmaktadır. Ancak bu yöntemler sadece küresel şekilli kümeleri bulabilmekte ve keyfi şekilli kümelerin ortaya çıkarılmasında zorluk yaşanabilmektedir. Bu zorlukları ortadan kaldırmak ve keyfi şekilli kümeler elde etmek için yoğunluk tabanlı (density based) yöntemler kullanılabilir (Han vd., 2012). Yoğunluk tabanlı kümeleme yöntemlerinin öne çıkan özelliği, farklı ve düzensiz şekillerdeki kümeleri

belirleyebilmesidir. Bu algoritmalar kümeleri oluşturmak için, birbirine bağlantılı yoğun noktalar belirler. Yani bir küme, yoğunluğun yönlendirdiği herhangi bir yönde genişleyebilir. Böylece herhangi bir şekildeki küme oluşturulabilir. Yoğunluk değerlendirmesinin doğal bir etkisi olarak, yoğunluk tabanlı kümeleme yöntemleri gürültü (sıradışı veya anormal özelliklere sahip veri) ile baş edebilirler. Ayrıca, bu yaklaşımı benimseyen yöntemler genellikle düşük hesaplama karmaşıklığına sahiptirler. En belirgin dezavantaj ise, ortaya çıkan kümelerin nasıl değerlendirileceği ve yorumlanacağıdır (Atilgan, 2014).

Şekil 4.4.6, gürültülü verilerde yoğunluğa dayalı örnek kümeleme sonuçlarını göstermektedir. Kümeler yoğunluklarına göre ayrılmış ve kümeleme sonucunda düzensiz şekilli kümeler elde edilmiştir.



Şekil 4.4.6 Yoğunluk Tabanlı Kümeleme Örnekleri.

4.3.1 DBSCAN

Yoğunluk tabanlı kümelemenin temsilcisi DBSCAN (Density Based Spatial Clustering of Applications with Noise) algoritması kabul edilir (Han vd., 2012). DBSCAN algoritması, 1996 yılında Martin Ester ve arkadaşları tarafından önerilmiştir (Ester vd., 1996). Bu algoritma büyük veri setlerinde etkili olarak çalışabilmekte, aykırı değerler ile baş edebilmekte ve keyfi şekilli kümeleri saptayabilmektedir (Sander vd., 1998). DBSCAN, kümeleri oluştururken nesnelerin yoğunluklarını dikkate alarak yoğun ve seyrek bölgeleri belirleyerek kümeleme yapılmaktadır. Algoritma yeterince yüksek yoğunluklu bölgeleri kümelere dönüştürür ve böylelikle gürültülü veritabanlarında farklı ve düzensiz şekilli kümelerin keşfedilmesini sağlamanmış olunur.

DBSCAN algoritmasının avantajları;

- Rasgele şekil yapılarına sahip kümeleri bulabilmektedir.
- Gürültülü veri setlerinde kullanılabilir.
- İki adet kullanıcı tarafından belirlenen parametresi vardır.

- Küme sayıları hakkında ön bilgi gerektirmez.

dezavantajları ise;

- Yanlı veri seti üzerinde farklı yoğunluklar bulunduğu zamanlarda yararlı değildir.
- Güzel sonuç elde etmek için çok iterasyon sayısı gerektirir.
- Parametre seçimleri derin bir önbilgi gerektirir.

olarak sıralanabilir.

Yoğunluğa dayalı kümelenme algoritmaları, diğer kümeleme algoritmalarında olmayan parametreler kullanmaktadır. Bu parametreler aşağıda tanıtılmıştır (Ester vd., 1996).

Komşuluk (Neighborhood): p ve q gözlemi D veri setinin elemanı olan iki gözlem olsun. Bu iki gözlem arasındaki komşuluk uzaklık fonksiyonları (Manhattan, Euclidean) ile tanımlanmaktadır ve dist(p,q) ile gösterilir.

Eps Komşuluğu (Eps-neighborhood): D veri setinin elamanı olan p noktasının Eps komşuluğu $N_{Eps}(p)$ ile gösterilir ve $N_{Eps}(p) = \{ q \in D \mid d(p,q) \leq \varepsilon \}$ şeklinde tanımlanır. gözlemlerin komşularını belirlerken kullanılan yakınlık mesafesi olan ε yarıçapı içindeki gözlemlerin komşuluğuna denir.

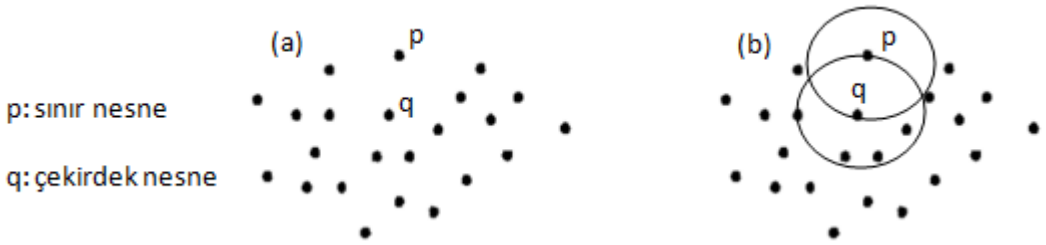
MinPts: Bir küme çevresinde bulunması gereken minimum nokta sayısını gösterir.

Çekirdek Gözlem (Core object): Bir gözlemin çekirdek gözlem olabilmesi için çevresinde Eps değeri koşuluna uyan Minpts sayısı kadar gözlem bulunması gerekmektedir.

Doğrudan yoğunluğa erişilebilirlik (Directly Density-Reachable): Aşağıdaki maddeler sağlandığı takdirde p gözlemi q gözlemi için doğrudan yoğunluk erişilebilirdir.

1. $p \in N_{Eps}(q)$
2. $\left|N_{Eps}(q)\right| \geq MinPts$

Doğrudan yoğunluğa erişebilirlik, çekirdek gözlemler için simetrik bir yapıdadır. Fakat bir çekirdek gözlem ve bir sınır gözlemi içeren durumlarda bu geçerli bir durum değildir. Şekil 4.4.7'de bu durum ifade edilmiştir. Şekilden görüleceği gibi, p gözlemi q gözlemi için doğrudan yoğunluk erişilebilir iken tam tersi için aynı durum geçerli değildir.



Şekil 4.4.7 Çekirdek Ve Sınır Nesneler.

Yoğunluğa erişebilirlik (Density reachable): Eps ve MinPts koşulları altında eğer $p_1,p_2,\ldots,p_n$ gözlemler zinciri varsa, $p_1 = q$ ve $p_n = p$ ise buradan yola çıkarak p_{i+1}, p_i'den doğrudan yoğunluğa erişebilirdir. Bu durumda p noktası q noktası üzerinden yoğunluğa erişebilir demektir. Yoğunluğa erişebilirlik, doğrudan yoğunluğa erişebilirliğin kanonik bir uzantısıdır. Bu ilişki geçişlidir fakat simetrik bir ilişki değildir. Şekil 3.8'de bu asimetrik yapı görülmektedir.

Yoğunluk bağlantılılık (Density-connected): Bir *o* gözlemi olsun ve p ve q gözlemleri Eps ve MinPts değerleri göz önüne alınarak *o* gözlemine yoğunluğa erişebilir durumda ise p gözlemi q gözlemi ile yoğunluk bağlatılıdır denir.

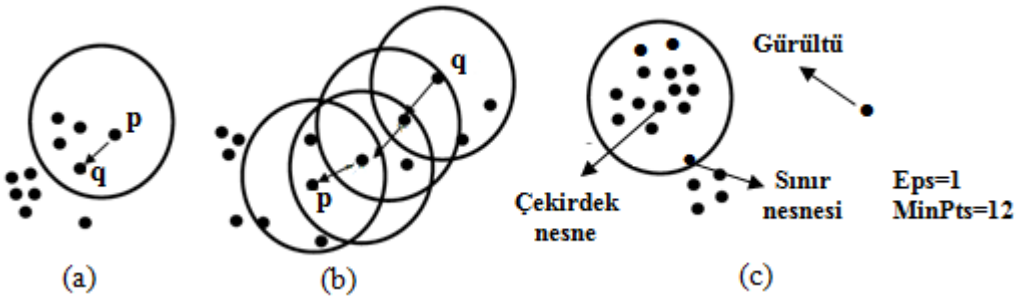
Küme: D gözlemlerden oluşan bir veritabanı olsun. Eps ve MinPts kuşullarını sağlayan bir K kümesi D'nin bir alt kümesidir.

(1) Her bir p ve q için, $p \in K$ ve q noktası p vasıtasıyla yoğunluğa erişebilirdir, $q \in K$

(2) Her bir $p,q \in K$ için p noktası q noktasıyla yoğunluk bağlantısallığına sahiptir.

Gürültü (Noise): $K_1,K_2,\ldots,K_n$, Eps ve MinPts koşullarını sağlamakta olan veri tabanındaki kümeler olsun. Veri tabanındaki herhangi bir K_i kümesine ait olmayan nokta ya da noktalara gürültü denir.

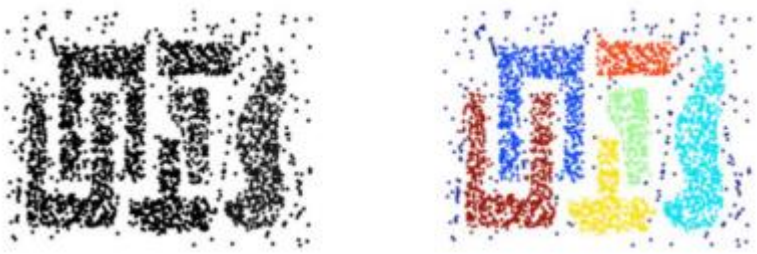
Sınır Gözlemi (Border Object): Eğer p gözlemi, bir çekirdek gözlem değilse ve başka bir çekirdek gözleminden yoğunluğa erişebilirlik var ise, bu gözleme sınır gözlemi denir.



Şekil 4.4.8 Temel Kavramlar Ve Terimler: (A) P Noktası Q Noktası Üzerinden Yoğunluğa Erişebilirdir, (B) P Ve Q Arasında O Noktası Aracılığı İle Yoğunluk Bağlantısallığı Bulunmaktadır, (C) Sınır, Çekirdek Nesneleri Ve Gürültü.

DBSCAN algoritması, aşağıda verilen adımlarla gerçekleştirilmektedir ve DBSCAN ile elde edilmiş örnek kümeler Şekil 4.4.9'da sunulmuştur.

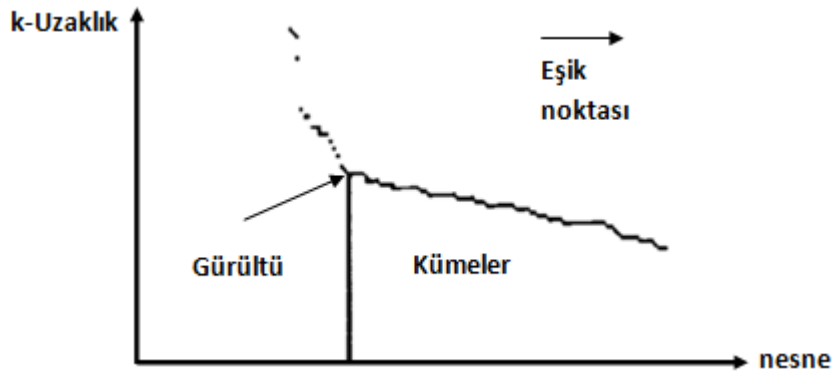
1. Eps ve MinPts değerleri belirlenir.
2. Rasgele ya da keyfi bir p noktası seçilir.
3. Önceden belirlenen Eps ve MinPts değerlerine göre doğrudan erişilebilir tüm nesneler alınır.
4. Eğer p bir çekirdek nesne (core object) ise küme oluşturulur.
5. Eğer p bir sınır nesnesi (border object) ise p noktasından farklı başka bir nesne seçilir.
6. Tüm nesneler işlemden geçirilene kadar bu süreç adımsal olarak devam eder.
7. Eğer hala bir kümeye atanmamış veri nesnesi bulunmakta ise bu nesneler gürültü olarak değerlendirilir.



Şekil 4.4.9 DBSCAN Algoritması Örnek Kümeleme Sonuçları.

DBSCAN algoritmasında amaç birbirine yakın sıkı kümeler elde etmektir ve bu sonucu elde etmek için doğru parametre seçimleri yapmak oldukça önemlidir. Bundan dolayı *MinPts* ve *Eps* parametrelerinin belirlenmesi detaylı bir şekilde ele alınmalıdır. Parametre seçimlerinde izlenecek yol adım adım şu şekilde ifade edilmektedir:

Öncelikle d, p nesnesi ile p nesnesinin k en yakın komşu uzaklığı olsun. Burada hemen hemen tüm p noktalarının d-komşu sayısı tam olarak k+1 tanedir. Sadece birden fazla nesnenin p ile uzaklığı d'ye eşit olduğu durumlarda d-komşu sayısı k+1'den fazla olabilmektedir ki bu oldukça ihtimal dışı bir durumdur. Bu nedenle k sayısını değiştirmek d uzaklığı üzerinde büyük bir değişikliğe neden olmayacaktır. Herhangi bir atanan k değeri için veri setindeki tüm nesnelerin k en yakın komşularına olan uzaklıkları hesaplanır ve küçükten büyüğe doğru sıralanır. Bu sıralanan uzaklıkların grafiği oluşturulur. Oluşturulan bu grafik ver setinin yoğunluğu hakkında bize bazı bilgiler sunmaktadır. Oluşturulan uzaklık grafiğinde ani değişimin bulunduğu nokta eşik değeri olarak alınır. Şekil 4.4.10 örnek bir uzaklık grafiğidir (Ester vd., 1996).



Şekil 4.4.10 Sıralanmış Uzaklık Grafiği.

Uzaklık grafiğinde, daha yüksek k uzaklık değerlerine sahip olan nesneler (eşik noktasının solunda bulunan nesneler) gürültü olarak düşünülmektedir. Diğer tüm nesneler ise bir kümeye atanmıştır. Genelde ani düşüşü tespit etmek çok zordur fakat grafiksel sunum ile bu durum kullanıcı için oldukça kolaylaşmaktadır. Sonuç olarak Eps değeri eşik noktasına karşılık gelen k-uzaklık değeri olarak seçilir. MinPts parametre değeri ise k değeri olarak seçilir (Ester vd., 1996).

4.3.2 GDBSCAN

GDBSCAN algoritması, (Generalized Density Based Spatial Clustering of Applications with Noise), DBSCAN algoritmasının iki farklı yolla genelleştirilmiş hali olarak 1988 yılında Sander ve arkadaşları tarafından ortaya atılmıştır (Sander, Ester, Kriegel, & Xu, 1998). GDBSCAN algoritmasının temeli DBSCAN algoritmasının iki önemli yolla genelleştirilmesidir. İlk olarak komşuluk tanımı simetrik ve refleksif olan ikili karşılaştırma belirtecine (predicate) dayandırıldığı

durumlarda Eps komşuluğu yerine başka bir komşuluk kavramının kullanılabilir olduğu düşüncesini ortaya atmıştır. Örnek olarak poligonlar kümelenirken, komşuluklar keşişim beliteci (predicate) ile tanımlanabilir. İkinci olarak bir nesnenin komşuluk sınırları içerisinde olan nesneleri saymak yerine örneğin, bir mahallenin ortalama geliri gibi mekansal olmayan nitelikleri göz önünde bulundurarak, bu mahallenin eleman sayısını tanımlamak için başka ölçüler kullanılabilir. Bu iki değiştirilmiş yolla oluşturulduğundan dolayı GDBSCAN algoritması mekansal ve mekansal olmayan niteliklere göre nesneleri kümeleyebilmektedir. GDBSCAN algoritmasının işleyiş süreci anlatılmadan önce tanımlanması gereken bazı kavramlar aşağıda belirtilmektedir (Sander vd., 1998).

Bir nesnenin komşuları: Herbir $p.q \in D$ için NPred, D veriseti içinde rekleksif ve simetrik ikili yüklem olsun. D veriseti elamanı olan *o* nesnesinin NPred-komşuluğu $NPred(o) = \{ o' \in D \mid NPred(o, o'\}$ olarak tanımlanmaktadır. DBSCAN algoritmasında komşuluk tanımı belirli nesnelerin uzaklığına dayandırılarak kısıtlanmaktadır. Fakat yüksek derecede farklı boyutlardaki poligonlarla çalışırken kesişme gibi komşuluk yüklemlerini kullanmak daha uygundur.

Bir nesnenin mekansal olmayan değişkenini hesaba katmanın bir diğer yolu ise bir nesnenin komşuluğunun kardinalitesini hesaplarken kullanılan ağırlıklardır. Bunun için ağırlıklandırılmış kardinalite fonksiyonu wCard oluşturulmuştur. Buna göre bir *o* nesnesinin ağırlığı $wCard$ ($\{o\}$) ile ifade edilir.

Bir nesneler kümesinin minimum ağırlığı (MinWeight): wCard, D veri setinin güç kaynağından negatif olmayan gerçel sayılara sahip bir fonksiyon,

$wCard: 2^D \rightarrow \Re^{\geq 0}$ ve MinCard pozitif gerçel sayı olsun. O zaman S nesneler kümesinin minimum ağırlığı $wCard(S) \geq MinCard$ olarak tanımlanır. Bu ifade DBSCAN algoritmasında olan $\left|N_{Eps}(o)\right| \geq MinPts$ ifadesinin genelleştirlmiş halidir.

Direk Yoğunluğa erişebilirlik (directly density-reachable): Bir p nesnesi NPred ve MinWeight parametrelerini dikkate alarak aşşağıdaki koşulları sağladığı sürece direk yoğunluğa erişebilirdir.

1. $p \in N_{NPred}(q)$
2. $MinWeight\left(N_{NPred}(q)\right)$ çekirdek nesne şartını sağlıyorsa

Direk yopunluğa erişebilirlik kavramı çekirdek nesneler için simetriktir. Fakat genelde bir çekirdek ve bir sınır nesnesi bulunmakta ise simetriklik kaybolmaktadır.

Yoğunluğa erişebilirlik (density-reachable): NPred ve MinWeight koşulları altında eğer $p_1,p_2,\ldots,p_n$ nesneler zinciri varsa, p_1 = q ve p_n = p ise buradan yola çıkarak (i = 1, …, n-1) p_{i+1}, p_i'den doğrudan yoğunluğa erişebilirdir. Bu durumda p noktası q noktası üzerinden yoğunluğa erişebilir demektir. Yoğunluğa erişebilirlik, doğrudan yoğunluğa erişebilirliğin kanonik bir uzantısıdır. Bu ilişki geçişlidir fakat simetrik bir ilişki değildir.

Yoğunluk-bağlantılı C nesneler topluluğunun iki sınır nesnesi muhtemelen birbirlerine yoğunluk bağlantılı olmayacaktır. Bunun nedeni ikisininde çekirdek nesneler koşullarının sağlayamayacak olmasıdır. Fakat Yoğunluk-bağlantılı nesneler topluluğunun oluşma koşulu bir çekirdek nesnesinin olması ve bu çekirdek nesnenin iki sınır nesnesi ile yoğunluk bağlantılı olmasıdır.

Yoğunluk bağlantılılık (Density-connected): Bir *o* nesnesi olsun ve p ve q nesneleri NPred ve MinWeight değerleri göz önüne alınarak *o* nesnesine yoğunluğa erişebilir durumda ise p nesnesi q nesnesi yoğunluk bağlatılıdır denir. Yoğunluk bağlanılabilirlik simetrik bir ilişkidir. Ayrıca yoğunluğa erişebilir nesneler için yoğunluğa bağlanılabilirlik ilişkisi de refleksifdir.

Yoğunluk bağlantılı set (Density-connected set): D verisetinin bir alt boyutu olan, NPred ve MinWeight parametrelerini göz önüne alan ve aşağıdaki koşulları sağlayan C setine yoğunluk bağlantılı set denir.

(1) Maksimallik: Her bir p ve q için, p∈C ve NPred, MinWeight parametrelerini dikkate alarak q noktası p noktasından yoğunluğa erişebilir ise , q∈C'dir.
(2) Bağlantı: Her bir p,q∈C için NPred ve MinWeight parametre şartları altında p noktası q noktasıyla yoğunluk bağlantısallığına sahiptir.

Kümeleme: NPred ve MinWeight parametrelerini göz önüne alarak D veri setinin bir kümelemesi olan CL, NPred ve MinWeight parametre koşullarına uyan yoğunluk bağlanabilir setlerin birleşimidir $CL = \{C_1, C_2, \ldots, C_k\}$. Eğer C yoğunluk bağlanabilir bir set ise $C \in CL$'dir.

Gürültü: $CL = \{C_1, C_2, \dots, C_k\}$ D veri setinde bulunan bir küme olsun. O zaman D verisetindeki gürültüler herhangi bir C_i yoğunluk bağlantılı sete ait olmayan nesnelere denir.

Teorem 1: p, D veri setine ait bir nesne ve $MinWeight\left(N_{NPred}(p)\right)$ çekirdek nesne şartını sağlıyor olsun. O zaman O seti $O = \{o \in D | \; o\; p'den\; yoğunluk\; ulaşılabilir\}$ yoğunluk bağlantılı bir settir.

Teorem 2: C yoğunluk bağlantılı bir set ve p'de C setinin $MinWeight\left(N_{NPred}(p)\right)$ çekirdek nesne şartını sağlıyor olan herhangi bir nesnesi olursa, C seti ile $O = \{o \in D | \; o\; p'den\; yoğunluk\; ulaşılabilir\}$ seti birbirine eşittir.

Yukarıdaki verilen tanımlar ve teoremler ışığı altında GDBSCAN algoritmasının işleyiş süreçleri basitçe aşağıda belirtilmiştir ve Şekil 4.4.11'de gösterilmiştir. GDBSCAN algoritması ilk olarak rastgele bir p gözlemi seçer NPred ve MinWeight parametre koşulları altında bu gözlemin tüm yoğunluk erişebilir olan nesnelerini bulur. Eğer p bir çekirdek nesne ise bu prosedür yoğunluk bağlantılı bir set oluşturmuş olur (tanım 1 ve 2). Eğer p çekirdek nesne değil ise hiçbir nesne p ile yoğunluk ulaşılabilir durumunda değildir ve p gürültü olarak atanır. Bu süreç tüm nesneler kümelenmemiş p nesnesi kalmayana dek adımsal olarak devam ettirilir.

```
GDBSCAN (SetOfObjects, NPred, MinCard, wCard
  // SetOfObjects is UNCLASSIFIED
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfObjects.size DO
    Object := SetOfObjects.get(i);
    IF Object.ClId=UNCLASSIFIED THEN
      IF ExpandCluster (SetOfObjects, Object, ClusterId, NPred, MinCard, wCard)
      THEN
        ClusterId := nextId (ClusterId)
      END IF
    END IF
  END FOR
END; // GDBSCAN
```

Şekil 4.4.11 GDBSCAN Algoritmasının Genel Yapısı.

GDBSCAN algoritmasının çalışma zamanı $O(n * bir\; komşu\; sorgu\; süresi)$ olarak hesaplanır. Herbir nesne için ClusterId gerekli olduğundan dolayı komşu sorgulama sayısı hiçbir şekilde azaltılamaz. Bu sebeple çalışma zamanı komşu sordulama performansına bağlıdır.

Çizelge 4.3 GDBSCAN Alogritmasının Çalışma Karmaşıklığı.

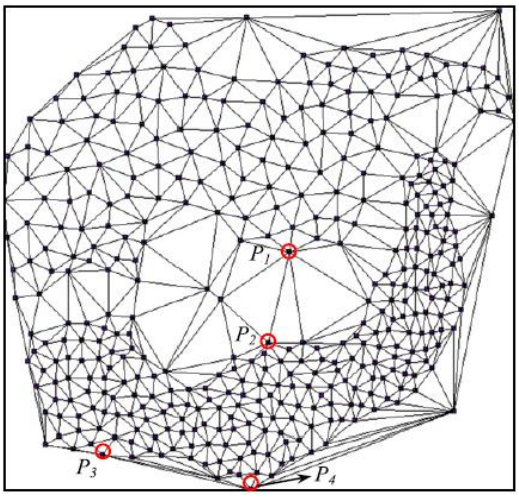
	Komşu sorgu süresi	GDBSCAN algoritması
İndeks Olmadan	$O(n)$	$O(n^2)$
Mekansal İndeksle	$O(logn)$	$O(n * logn)$
Direk Ulaşımla	$O(1)$	$O(n)$

4.3.3 DBSC

Geometrik özellikler ve değişkenler mekansal bir nesne için önemli iki karakteristik yapısıdır. Bahsedilen mekansal kümeleme algoritmalarında bu iki kavram çoğunlukla gözardı edilmiştir. DBSC (Density-Based Spatial Clustering) algoritmasında ise bu iki kavram kümeleme işlemine dahil edilmektedir. DBSC, mekansal ve değişkenlerdeki yakınlıkların ikisini birden dikkate alan bir algoritmadır. Kenar uzunluğu kısıtlamalarına sahip olan Delaunay üçgenlemesi kavramı, mekansal nesneler arasındaki mekansal yakınlık ilişkilerini modellemek için ilk olarak DBSC algoritmasında kullanılmıştır (Q. Liu, Deng, Shi, & Wang, 2012). DBSC algoritması mekansal yakınlık ilişkilerini belirlemede Delaunay üçgenlemesini kullanmaktadır. Bunun nedeni ise Delaunay üçgenlemesinin mekansal gözlemlerin doğal komşularını bulma konusunda kullanılabilir olduğu bilinmektedir. Aşağıda Delaunay üçgenlemesinin çalışma şekli ve düzensiz dağılmış olan verilerdeki yapısı anlatılmıştır.

Düzensiz dağılmış veri kümesi için mekansal yakınlık ilişkilerinin oluşturulması

Delaunay üçgenlemesinde bir P_i mekansal gözlemine yakın olan gözlemler P_i gözleminin komşularını oluşturur. Fakat düzensiz dağılımlı bir veri seti için Delaunay üçgenlemesi, veri setinin kenarları-çukurları yakınında veya düşük ve yüksek yoğunluklu bölgeler arasındaki boşluklarda hatalı sonuçlar vermektedir (Kolingerova ve Zalik, 2006).

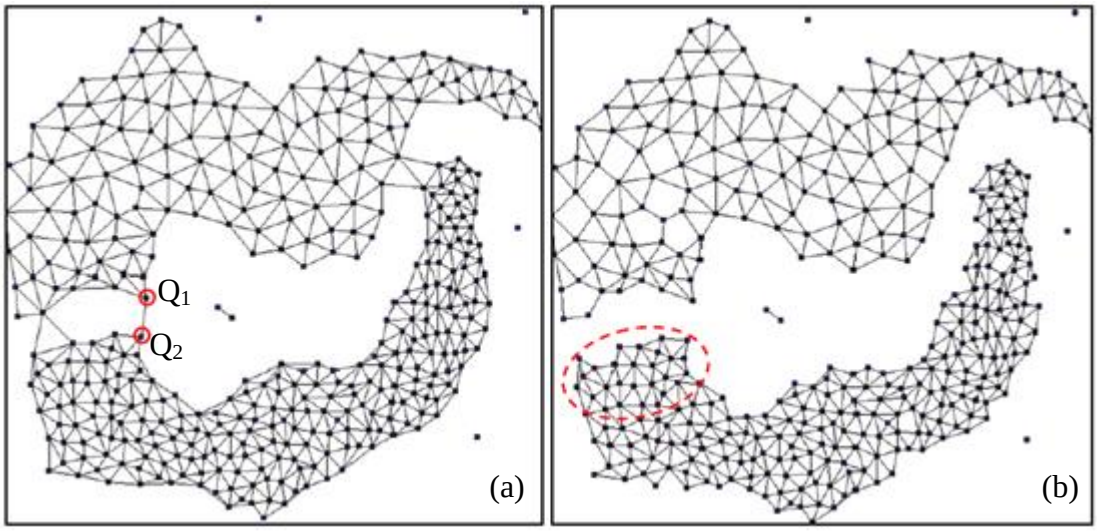


Şekil 4.4.12 Delaunay Üçgenlemesi Örneği (Q. Liu vd., 2012).

Yukarıdaki şekilde Delaunay üçgenlemesi örneği incelendiğinde aralarındaki uzaklık oldukça yüksek olmasına rağmen P_1, P_2, P_3 ve P_4 gözlemleri aynı kenarları paylaşmakta ve mekansal olarak komşu olarak tanımlanmaktadır. Aslında düzensiz dağılmış veri setlerinde mekansal yakınlığın modellemesini Delaunay üçgenlemesine dayanarak doğru bir şekilde yapmak için sayısal bir uzaklık kısıtı getirilmelidir. Bunun için yakın zamanda ASCDT algoritması geliştirilmiştir (Deng vd., 2011). ASCDT algoritmasından yola çıkılarak DBSC algoritmasında ilk olarak global seviyede Delaunay üçgenlemesindeki uzun kenarlar sonrasında ise bölgesel seviyeden silinmiştir. Bu budama işlemi yapıldıktan sonra aynı kenara ait olan gözlemler mekansal komşu olarak tanımlanmıştır. Kenarların global ve bölgesel seviyelerde uzun olup olmadığına karar vermek için öncelikle global ve bölgesel uzaklık eşiğinin bulunması gerekmektedir. SD bir mekansal veri seti ve $D\ddot{U}(SD)$'de SD veri setindeki her bir mekansal P_i nesnenin bir nokta olarak temsil edildiği Delaunay Üçgenlemesi olsun. $Global\ ortalama(D\ddot{U})$ üçgenlemedeki kenar uzunluklarının ortalaması, $b\ddot{o}lgesel\ ortalama(D\ddot{U})$ ise doğrudan P_i'ye gelen kenarların ortalama uzunluğudur. $GlobalSSapma(D\ddot{U})$ Delaunay üçgenlemesindeki tüm kenarların uzunluğunun standart sapması, $B\ddot{o}lgeselSSapma(D\ddot{U})$ ise doğrudan P_i'ye gelen kenarların uzunluğunun standart sapmasıdır.

$$Global\ uzaklık\ eşiği(P_i) = Global\ ortalama(D\ddot{U}) + \alpha.GlobalSSapma(D\ddot{U})$$

$$\alpha = Global\ ortalama(D\ddot{U})/B\ddot{o}lgesel\ ortalama\ (P_i)$$



Şekil 4.4.13 Mekansal Yakınlık İlişkisinin Belirlenmesi: (A) Global Uzun Kenarların Çıkarılması, (B) Bölgesel Uzun Kenarların Çıkarılması (Q. Liu Vd., 2012).

Eğer P_i'ye direk bağlı olan bir kenarın uzunluğu $Global\ uzaklık\ eşiği(P_i)$'den daha büyük ise o köşe global uzun kenar olarak tanımlanır ve Delaunay üçgenlemesinden çıkarılır. Yukarıdaki şekilde gösterilen Delaunay üçgenlemesinden uzun kenarların çıkarılmış hali gösterilmektedir. Global uzun kenarların çıkarılmasından sonra halen daha bölgesel seviyelerde bazı hataların olduğu gözlemlenmektedir. Şekil (a) Q_1 ve Q_2 tarafından bağlanan yüksek ve düşük yoğunluk seviyesine sahip kısımları göstermektedir. Globale göre uzun olmayan kenarlar bölgesel seviyelere göre çok uzun olabileceğinden dolayı bölgesel eşiği geliştirilmiştir.

Global uzun kenarlar çıkarıldıktan sonra herbir G_i alt grafiğinde; $2-Ordermean\ (P_i)$, P_i'den başlayan 2 ya da daha fazla yolla ilişkili olan tüm kenarların ortalama uzunluğudur. $OrtalamaSSapma(P_i)$ ise tüm $BölgeselSSapma(Q_i)$'lerin ortalama değeridir. Q_i'ler P_i'den başlayan 2 ya da daha fazla yolla ilişkili olmalıdır. Bu bilgiler ışığında bölgesel uzaklık eşiği aşağıdaki denklemle hesaplanmaktadır.

$$Bölgesel\ uzaklık\ eşiği(P_i) = 2 - OrderMean(P_i) + \beta OrtalamaSSapma(P_i)$$

G_i alt grağindeki herbir P_i nesnesi için, eğer P_i'nin 2-order komşularına ait olan kenarların uzunluğu $Bölgesel\ uzaklık\ eşiği(P_i)$'den daha büyük ise o kenar bölgesel uzun kenar olarak tanımlanır ve G_i'den çıkarılır. Bölgesel budama işlemi yapıldıktan sonra şekil (b) mekansal yakınlık ilişkisi çok daha iyi gösterilmektedir (β=2). ASCDT algoritmasının aksine DBSC'de bölgesel uzaklık eşiği daha az

düzeyde hassas olduğundan dolayı β parametresi 1 yerine 1'den büyük bir değer alınmaktadır. Sonuç olarak $Bölgesel\ uzaklık\ eşiği$, Delaunay üçgenlemesi tarafından oluşturulan mekansal yakınlık ilişkisininin daha da geliştirilmesi için kullanılmaktadır (Q. Liu vd., 2012).

Değişkenler ile mekansal gözlemlerin kümelenmesi

Global ve bölgesel budama işlemlerinden sonra geliştirilmiş Delaunay üçgenlemesinin (G-DÜ) oluşturulmasıyla mekansal yakınlık ilişkileri ortaya çıkarılmıştır. Bu işlemlerden sonra mekansal nesnelere ait değişkenlerin yakınlıkları da ölçülüp kümeleme işlemi yapılacaktır. DBSC algoritması değişkenler arası yakınlığı ölçme konusunda Öklid uzaklığından yararlanmaktadır. Kümeleme yapılmadan önce işlemler sırasında kullanılan bazı kavramların tanımlanması gerekmektedir.

Mekansal komşular (Spatial Neighbors): G-DÜ'deki hernangi bir P_i gözlemi için mekansal komşular, P_i gözlemine doğrudan bağlı olan gözlemlerdir ve $Komşular(P_i)$ ile gösterilir.

Mekansal Doğrudan ulaşılabilirlik (Spatially Directly Reachable): G-DÜ'deki hernangi bir P_i gözlemi için aşağıdaki koşullar sağlandığında, Q_i gözlemi P_i gözlemi için mekansal doğrudan ulaşılabilirdir. Burada T_l değişkenler arası yakınlığı belirleyen eşik değeridir.

1. $Q_i \in Komşular(P_i)$
2. $Uzaklık(P_i, Q_i) \leq T_l$

Yukarıdaki mekansal doğrudan erişebilirlik kavramının 2. maddesi sadece iki gözlem arasındaki yakın benzerliği dikkate aldığı görülmektedir.

Mekansal ulaşılabilirlik (Spatially Reachable): CLU bir grup kümelenmiş gözlemler topluluğu ve Ort(CLU) ise ortalama değiken değerleri olsun. İki veya daha fazla gözlemi bulunan CLU gözlemler seti için aşağıdaki koşullar sağlandığında, herhangi bir Q_i gözlemi CLU için mekansal ulaşılabilirdir.

1. $Uzaklık(Q_i, Ort(CLU)) \leq T_l$
2. $Q_i \in Komşular(P_i)$ ve $P_i \in CLU$

Mekansal ulaşılabilirlik bir gözlem ile kümelenmiş bir nesneler topluluğu arasındaki yakınlığı tanımlamak için kullanılmaktadır.

Yoğunluk belirleyici (Density İndicator): G-DÜ'deki hernangi bir P_i gözlemi için yoğunluk belirleyici tanımı aşağıdaki gibi yapılmaktadır.

$YB(P_i) = N_{sdr}(P_i) + N_{sdr}(P_i)/N(P_i)$

Burada $N_{sdr}(P_i)$, P_i için mekansal doğrudan ulaşılabilir gözlemler sayısını $N(P_i)$ ise P_i ile komşuluk halinde bulunan toplam gözlemler sayısını belirtmektedir. Yoğunluk belirleyici verilen T_l değişkenler arası yakınlığın eşik değeri ile bir gözlemin yoğunluğunu ölçmektedir. Eğer P_i gözlemine yakın olan birçok gözlem bulunmakta ise $N_{sdr}(P_i)$ değeri büyük olacaktır. Bu yöntem komşuluğun saflığınıda hesaba kattığı için geleneksel yoğunluk kavramlarından oldukça farklıdır. Yoğunluk belirleyici komşuluğun saflığını ölçmede $N_{sdr}(P_i)/N(P_i)$'den yararlanmaktadır. Bu değer bire yaklaştıkça daha saf bir komşuluk, sıfıra yaklaştıkçada tam tersi durum söz konusudur. Bu saflık sayesinde aynı derecede mekansal direk ulaşılabilir komşularının sayısı aynı olan nesneleri bile ayırabilme yeteneğine sahiptir.

***Mekansal kümeleme çekirdeği (Spatial Clustering Core*):** Bir mekansal kümeleme çekirdeği, kümelenmemiş gözlemler arasında en çok yoğunluk belirleyici değerine sahip olan gözlemdir. İki veya daha fazla gözlemin bu şartları sağladığı durumda bu gözlemler arasında en küçük ortalama değişken farklılığına sahip olan gözlem seçilir.

Genişleyen çekirdek (Expanding Core): G-DÜ'deki hernangi bir P_i gözlemi için, eğer komşuları arasında en azından 1 adet mekansal direk ulaşılabilir olan gözlem bulunmakta ise P_i bir genişleyen çekirdek gözlemdir.

DBSC algoritması 3 ana adımdan oluşmaktadır. Bu adımlar ve adımlarda izlenen işlemler aşağıda belirtilmektedir (Q. Liu vd., 2012).

Adım 1: Mekansal komşulukların oluşturulması. Bu adım kendi içinde 3 işlemden meydana gelmektedir ve bu işlemler aşağıda belirtilmektedir.

i. Mekansal veri seti için Delaunay üçgenlemesi oluşturulur. Bu işlemin zaman karmaşıklığı $O(Nlog(N))$'dir.
ii. Global uzaklık eşiği yardımıyla Delaunay üçgenlemesinde bulunan global uzun kenarlar işlemden çıkarılır. Bu işlemin zaman karmaşıklığı $O(N)$'dir.

iii. Bölgesel uzaklık eşiği yardımıyla Delaunay üçgenlemesinde bulunan bölgesel uzun kenarlar işlemden çıkarılır. Bu işlemin zaman karmaşıklığıda $O(N)$'dir.

Adım 2: Yoğunluk belirleyicinin hesaplanması ve T_l parametresinin belirlenmesi. Her gözlem için yoğunluk belirleyici değeri hesaplanır ve bu değere göre gözlemler büyükten küçüğe doğru sıralanır. Bu işlem yaklaşık olarak $O(Nlog(N))$ zaman harcar. DBSC algoritmasında T_l parametresini belirlemek için Liu ve arkadaşları çok başarılı sonuçlar veren bir yöntem geliştirmişlerdir. Bu yöntemde ilk olarak tüm P_i gözlemleri ile bu gözlemlerin en yakın mekansal komşuları arasındaki değişkenlerin farklılıklarını hesaplamıştır. Sonrasında aykırı değerler belirlenip 3 standart sapma kuralına göre çıkarılmıştır. Sonrasında hesaplanan ortalama farklılık T_l olarak alınmıştır. Bu işleminde zaman karmaşıklığı $O(N)$'dir.

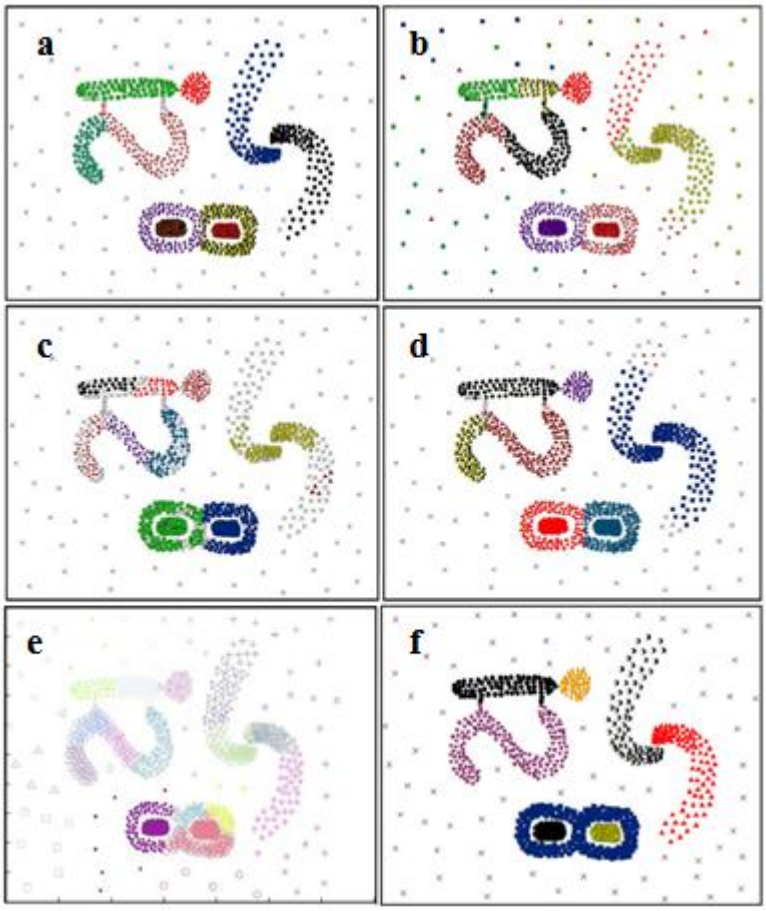
Adım 3: Mekansal kümeleme işleminin uygulanması. Bu adımda kendi içinde 5 adımda ifade edilmektedir.

i. Bir mekansal kümeleme çekirdeği olan P_i gözlemi seçilir. Sonrasında P_i gözlemini komşuları içindeki genişleyen çekirdek gözlemler yoğunluk belirleyici değerlerine göre sıralanır.
ii. Büyükten küçüğe doğru sıralanmış şekilde genişleyen çekirdekler P_i'ye eklenir. Bu çekirdekler P_i'ye göre aynı anda hem mekansal doğrudan ulaşılabilir ve mekansal ulaşılabilir olmalıdır. Bu işlemlerle beraber ön kümeler elde edilir.
iii. Pi'nin her bir K-sıra komşusu için (K≥2), başlangıç kümesine eklenecek ilk gözlem, başlangıç genişleyen çekirdek olarak seçilir. Sonrasında gözlemler iteratif olarak adım ii'ye göre eklenir.
iv. P_i ile başlayan kümeye eklenecek gözlem kalmadığı zaman bir mekansal küme elde edilmiş olur.
v. Belirlenecek gözlem (küme ya da gürültü) kalmayana kadar i-iv adımları tekrarlanır. Tüm gözlemler tanımlandıktan sonra kümeleme işlemi sonlanır.

DBSC algoritması uygulanmadan önce β ve T_l olmak üzere 2 parametrenin kullanıcı tarafından girilmesi gerekmektedir. β parametresi genelllikle 2 veya 3 değeri seçilmektedir (varsayılan β=2). Bunun sebebi bazı uzun kenarların gözden kaçırılmamasıdır. T_l parametresi yüksek bir değer olarak seçildiği durumlarda ise her

kümedeki değişkenlerin varyanslarıda otomatik olarak yükselecektir. Liu ve arkadaşları tarafından T_l parametresi için en uygun değer aralığı, β 2 ile 2,5 arasında olduğu durumlarda 0,85 ile 1,15 arasıdır. DBSC algoritmasının genel zaman karmaşıklığı $O(NlogN)$'dir (Q. Liu vd., 2012).

Şekil 4.14'de DBSC algoritması ile diğer kümeleme algoritmalarının sonuçları karşılaştırılmıştır. DBSC algoritması nesneleri 10 adet kümeye ayırmıştır ve gürültü gözlemleri iyi bir şekilde ortaya çıkarmıştır. Diğer kümeleme algoritmaları DBSC algoritması kadar güzel sonuçlar vememiştir. GDBSCAN algoritmasında sadece 4 küme doğru olarak tanımlanmıştır ve birbirinden ayrı kümeler tek bir kümede toplanmıştır. ASCDT algoritması gözlemlerin değişken değerlerini dikkate almadığından dolayı mekansal olarak birbirine yakın fakat değişkenleri çok farklı olan değerleri yanı kümeye atamıştır. DBSC algoritması, düzgün dağılmayan ve değişik yoğunluklara sahip olan nesneler arasındaki mekansal yakınlık ilişkisini bulma metodu sayesinde diğer kümeleme algoritmalarına göre başarı sağlamaktadır.



Şekil 4.4.14 Algoritmalara Göre Sonuçlarının Karşılaştırılması (A) DBSC (Tl=0,87); (B) K Ortalama; (C) CURE; (D) GDBSCAN (Eps1=42,8, Eps2=0,87); (E) SOM; (F) ASCDT (Q. Liu Vd., 2012).

4.3.4 VDBSCAN

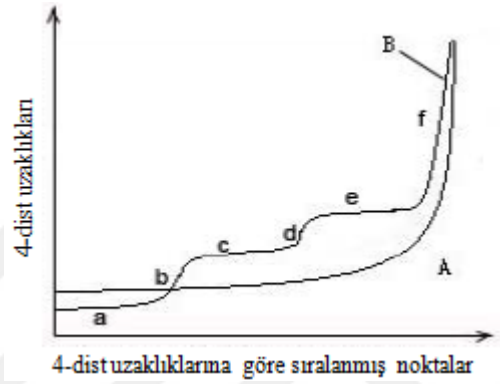
DBSCAN algoritması düzensiz şekilli ve boyutlu kümeleri bulabilen geleneksel bir yoğunluk tabanlı kümeleme algoritmasıdır. DBSCAN algoritması başarılı bir algoritma olmasına rağmen değişen yoğunluklu kümelerle çalışırken hatalara sebep olabilmektedir. Bu problemin üstesinden gelebilen OPTICS ve Jarvis-Patrick algoritmaları geliştirilmiştir. Fakat bu algoritmaların aynı zamanda kümeleme geçerliliğini azaltmaktadır. Tüm bu sorunların üstesinden gelebilmek için Peng Liu, Dong Zhou ve Naijun Wu tarafından 2007 yılında VDBSCAN (Varied Density Based Spatial Clustering of Applications with Noise) algoritması geliştirilmiştir (P. Liu, Zhou, & Wu, 2007). VDBSCAN algoritması, farklı yoğunluk seviyeleri için en uygun parametreleri bulup seçilen herbir parametre ile DBSCAN algoritmasını uygular. Örneğin C_1 ve C_2 iki küme olsun ve bir gürültü gözlem daha yoğun olan C_1 kümesine daha yakın olmasına rağmen C_2 kümesiyle aynı yoğunluğa sahip olsun. Eğer Eps değeri C_2 kümesini bulacak kadar düşük olursa C_1 ve onun etrafındaki gözlemler ayrı kümeler oluşturacaktır. Eğer Eps değeri C_1 kümesini ayrı küme olarak alarak onu çevreleyen nesneleri gürültü olarak alacak kadar yüksek olursa, C_2 ve onu çevreleyen gözlemlerde gürültü olacaktır.

VDBSCAN algoritması ilk olarak her gözlem için k-en yakın komşu uzaklıklarına (k-dist) bakar ve k-dist grafikleri oluşturulur. K-dist grafiğinin oluşmasında ilk önce tüm gözlemler için k-dist değerleri hesaplanır ve küçükten büyüğe sıralanır. Sıralanan bu değerler grafikleştirilir. Sonrasında bu grafik yardımıyla sezgisel olarak yoğunluklar numaralandırılır. Sonrasında herbir yoğunluk seviyesi için Eps_i parametreleri seçilir. Devamında veri seti taranıp, Eps_i değerleri kullanılarak farklı yoğunluklar kümelenir. Genel olarak VDBSCAN algoritması genel olarak Eps_i parametresini seçmek ve kümeleme olmak üzere 2 adımdan oluşmaktadır.

Adım1: Eps_i parametre seçimi

Bu adım kümeleme işlemindeki en önemli adımdır. K-dist grafiği sadece Eps_i değerlerini belirlemek için değil ayrıca veri setinin yoğunluk seviyelerini belirlemek içinde kullanılır. Çok çeşitli yoğunluklara sahip veri kümeleri için, kümenin yoğunluğuna ve noktaların rastgele dağılımına bağlı olarak, bir miktar varyasyon olacağı unutulmamalıdır, fakat aynı yoğunluk seviyesindeki noktalar için varyasyon aralığında keskin bir değişim olmayacaktır. Bu sebeple k-dist grafiğinden birden çok

eğri olacaktır. Eğer grafikte n adet eğri var ise veri seti n adet farklı yoğunluk seviyesine sahiptir. Şekil 3.15 k-dist grafiğine bir örnektir. Bu grafikte A çizgisi tek yoğunluk boyutlu veri setini temsil etmekteyken, B çizgisi ise 3 boyutlu bir veri setini temsil etmektedir. Şekildeki b ve d gibi çizgiler seviye-dönüş (level-turning) çizgileri olarak adlandırılır. Çizgi b, a ve c çizgisini birleştirir ve çizgi d ise c ve e'yi birleştirir. A, c ve e çizgileri ise farklı seviyelerde kalan çizgilerdir. Tıpkı DBSCAN algoritmasında da olduğu gibi f çizgisi aykırı gözlemleri idafe etmektedir ve herhangi 2 çizgiyi birleştirmediğinden dolayı seviye-dönüş çizgisi değildirler (P. Liu vd., 2007).



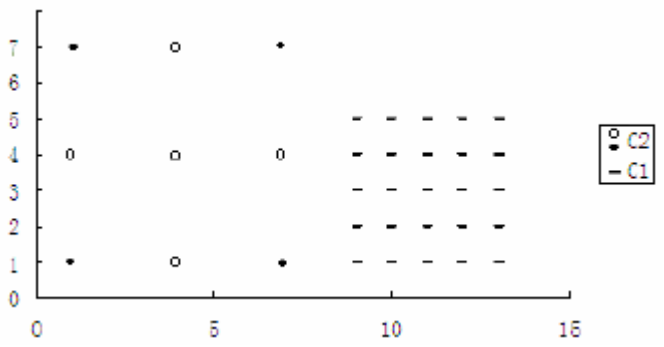
Şekil 4.4.15 k-dist Grafiği (P. Liu vd., 2007).

Şekil 4.4.15'de üç farklı yoğunluk seviyesi bulunmaktadır ve a çizgisi en yoğun b çizgisi ise en az yoğun olan yoğunluk seviyeleridir. Bu grafikte en uygun 3 Eps değerini seçmek için ilk olarak a ile b çizgilerini bir alt grafik olarak düşünüp dönüm noktası Eps_1 olarak seçilir. Aynı şekilde c ve d çizgisi Eps_2 ve son olarak e ve f çizgisi ise Eps_3 parametre değerini bulmada kullanılır.

Adım 2: Değişen yoğunluklu kümeleme

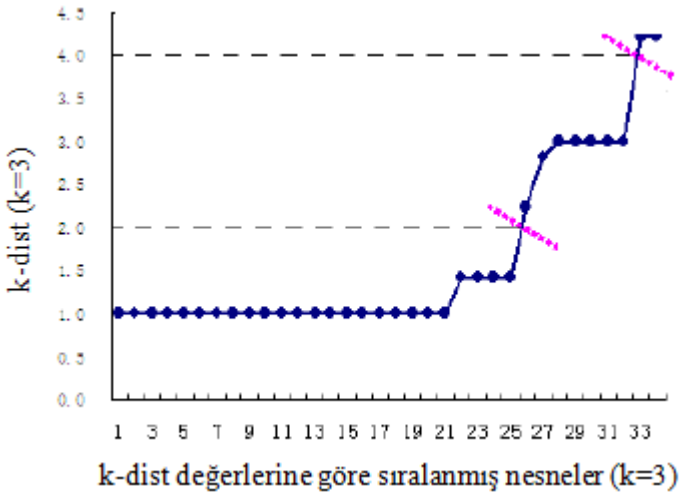
Bu adım Uygun Eps_i değerlerinin seçiminden sonra DBSCAN algoritmasının uygulandığı adımdır. Unutulmamalıdır ki Eps_i k-dist uzaklıklarının sıralanmış halinden meydana gelmektedir ve $Eps_i < Eps_{i+1}$'dir ($i < n$). DBSCAN algoritmasını Eps_{i+1}'e uygulamadan önce, Eps_i ile bazı gözlemler kümelendirilir ve bu kümeler C_{i-t} (t bir doğal sayıdır) ile gösterilir. Burada t ait olunan kümeyi t ise yoğunluk seviyesini ifade etmektedir. Kümelenen gözlemler tekrar işleme alınmamaktadır ve tüm Eps_i işlemlerinden sonra kümelenmeyen gözlemler gürültü olarak tanımlanır (P. Liu vd., 2007).

VDBSCAN algoritmasının işleyişinin daha iyi anlaşılması için Şekil 4.4.16'da gösterilen iki boyutlu veri seti ele alınmıştır. Şekilden de açıkça görüldüğü üzere iki farklı yoğunuk seviyesinin olduğu ve bu iki yoğunluk bölgesinin de tekdüze dağıldığı görülmektedir.



Şekil 4.4.16 Veri Seti Yapısı (P. Liu vd., 2007).

Algoritmada ilk olarak k-dist değerleri hesaplanmaktadır. Bu veri seti için k = 3 olarak alınmıştır. Bu değer için hesaplanan uzaklıklara k(3)-dist denir ve hesaplanan bu uzaklıklar sıralanır. Sıralanan değerler en uygun Eps_i değerlerini bulmak için grafiklendirilir. Şekil 4.4.17'de sıralandırılmış uzaklıkların grafiği gösterilmektedir. Bu grafiğe göre 2 adet keskin olarak düşüş yaşanan nokta olduğu dikkat çekmektedir. Bu değerlerden ilki olan 2 Eps_1 olarak seçilmekte ikinci değer olan 4 ise Eps_2 olarak seçilmektedir. Bu seçimlerden sonra herbir Eps_i değeri için DBSCAN algoritması uygulanacak olup MinPts değeri ise k=3 değeri olarak seçilmektedir (ilk ani düşüşün yaşandığı yer).



Şekil 4.4.17 k-dist Değerine Göre Sıralanmış Nesneler.

Önceden de belirtildiği üzere 2 Eps değeri içinde DBSCAN algoritması uygulanacaktır. Fakat ilk algoritma sonucunda kümelenmeyen nesneler ikinci Eps değerindeki sürece dahil edilecektir. Bir önceki Eps değerindeki kümelenmiş nesneler bir sonraki Eps sürecine dahil edilmezler. Sonuç olarak VDBSCAN algoritması DBSCAN algoritmasının değişen-yoğunluklu veri yapılarıyla çalışma engelini farklı yoğunluklar için farklı Eps değerleri ile ortadan kaldırmaktadır.

4.3.5 DBCLASD

DBCLASD (Distribution Based Clustering of LArge Spatial Databases) algoritması ilk olarak 1998 yılında Xiaowei Xu, Martin Ester, Hans-Peter Kriegel ve Jörg Sander tarafından ICDE konferansında sunulmuştur (Xu vd., 1998). DBCLASD algoritması öncelikle homojen Poisson olarak dağılmış olan noktasal gözlem kümelerini belirlemek için oluşturulmuştur. Kümeleme işleminde algoritmaların çoğunun tümünü birden yerine getiremediği fakat çok önemli olan bazı gereklilikler bulunmaktadır ve bu gereklilikler aşağıda belirtilmektedir. DBCLASD algoritmasının diğer bir amacı ise bu gerekliliklerin sağlanmasıdır (Xu vd., 1998).

1. Kullanıcı tarafında girilmesi gereken parametre sayısının az olması. Bunun sebebi çoğu uygulamada bu parametreler için uygun değerler önceden bilinmemektedir ve tahmin edilmesi zor olabilmektedir.
2. Değişik yapı ve şekillerdeki kümelerin belirlenmesi.
3. Çok büyük veri setlerinde verimli çalışma sağlanması.

DBCLASD algoritması hiçbir girdi parametresi olmadan en iyi küme sayısı ve yapısını büyük veri setlerinden etkilenmeden ortaya koyabilmektedir. DBCLASD algoritması uygulanmadan önce kümelemede kullanılan bazı kavramların tanımlanması gerekmektedir (Xu vd., 1998).

Bir gözlemin en yakın komşusu ve en yakın komşu uzaklığı: q seçilen bir gözlem ve S ise birçok noktadan oluşan bir set olsun. q gözleminin S noktalar seti içerisindeki en yakın komşusu $NN_S(q)$ ile gösterilir ve q noktasına en yakın olan noktadır. Bu noktalar arasındaki uzaklık ise $NNdist_S(q)$ ile gösterilir.

Bir noktalar setinin en yakın komşular uzaklıklarının seti: S noktalardan oluşan bir set ve e_i'de S setinin elemanlarından oluşan bir set olsun. e_i setinin S seti için en yakın komşu uzaklıklarından oluşan set $NNdistSet(S)$ ile gösterilir. Bir başka şekilde tanımlamak gerekirse tüm $NNdist_S(e_i)$ değerleridir.

DBCLASD algoritmasında bir kümenin en yakın komşu uzaklıklarının olasılık dağılımı analiz edilmek ve belirlenmek istenir. Bu analiz bir küme içindeki nesnelerin tekdüze dağılıma sahip olma varsayımına dayanmaktadır. Fakat tüm veri seti için bu varsayım geçerli değildir. N, R veri uzayı üzerinde tekdüze dağılıma sahip olan nesneler olsun ve bu veri uzayının hacmi $Vol(R)$ ile gösterilsin. Bu N adet gözlemden birinin R veri uzayının bir alt uzayı olan S'ye düşmesi olasılığı $Vol(S)/Vol(R)$'dır. R uzayındaki işlem içerisinde olan herhangi bir noktanın en yakın komşu uzaklığı olan D nin bazı x'lerden daha büyük olma olasılığı, N gözlemin hernangi bir q gözlemi etrafındaki x yarıçaplı küre içerisine girmeme olasılığı ile eşittir ve $SP(q,x)$ ile gösterilir (Xu vd., 1998).

$$P(D > x) = (1 - Vol(SP(q,x))/Vol(R))^N \quad (3.2)$$

Sonuç olarak, D'nin x'den büyük olmama olasılığı ise;

$$P(D \leq x) = 1 - P(D > x)$$

$$= 1 - (1 - Vol(SP(q,x))/Vol(R))^N \quad (3.3)$$

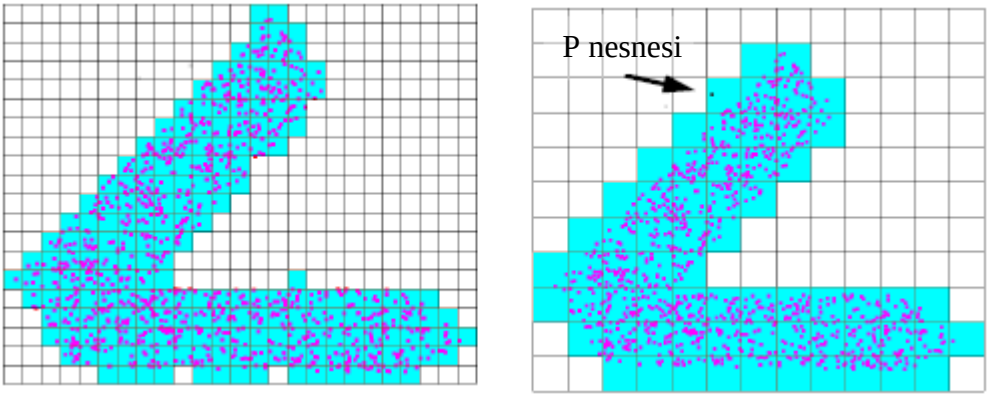
İki boyutlu uzayda dağılım fonsiyonu ise aşağıdaki gibidir.

$$F(x) = P(D \leq x)$$

$$= 1 - ((1 - \pi x\text{^}2) \ / \ (Vol(R)))^N \quad (3.4)$$

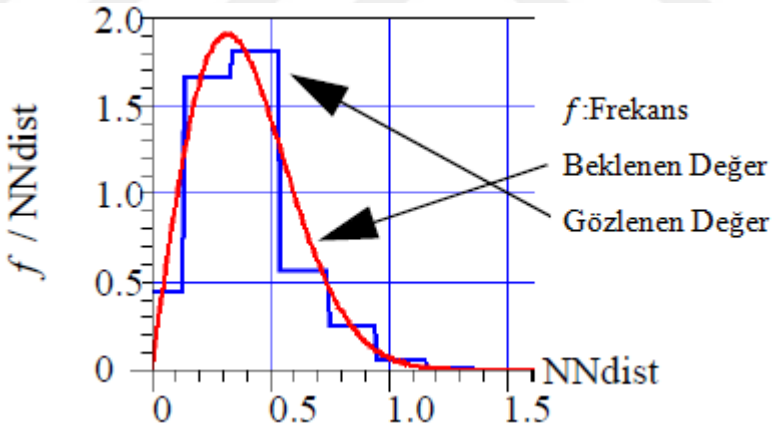
Dağılım fonksiyonu N ve $Vol(R)$ olmak üzere iki adet parametreye sahiptir. Burada N sayısını belirlemek çok kolay olsa bile, değişik şekillere sahip olabilen gözlemler topluluğunun $Vol(R)$ parametresinin nasıl hesaplanacağı açık olarak belli değildir.

S gözlemler topluluğunun kümelenen gözlemlerinin belirli bir alanı bulunmamaktadır. Böyle bir küme alanı belirlemek için DBCLASD algoritması bu alanı belirlemede bir yaklaşım ortaya koymaktadır. Bu yaklaşımda ilk olarak belirlenen alanın şekli kümenin şekline olabildiğince benzer olmalıdır. İkinci olarak ise yaklaşımda kullanılacak olan ızgara yapıları birbirine bağlı olmalıdır. Burada ızgara tabanlı bir yaklaşım kullanılmaktadır. Izgara tabanlı yaklaşımlarda en önemli konulardan biri olan ızgara genişliğinin seçimidir. Eğer ızgara genişliği büyük tutulursa küme tahmini doğruluktan uzaklaşır. Eğer çok küçük seçilirse de tahmin yapısı birbirinden ayrı birçok poligon yapısına bölünebilir.



Şekil 4.4.18 Izgara Genişliğinin Alan Tahmini Üzerindeki Etkisi (Xu vd., 1998).

Şekil 4.4.18, ızgara genişliğinin seçiminin ne kadar önemli olduğunu göstermektedir. DBCLASD algoritmasında ızgara genişliği $NNdistSet(S)$ nesnelerinin maksimum elemanıdır. Dolu bir ızgara hücresi, gözlemler setinden en az bir nokta barındırmalıdır ve S setinin tahmini alanı tüm dolu ızgaraların birleşimidir. Şekil 4.4.19 bir kümenin alanının hesaplanmasında beklenen ve gözlenen uzaklık dağılımlarının karşılaştırılmasını (uyumunu) göstermektedir. Beklenen ve gözlenen değerlerin uyumunu bulmada χ^2 testi kullanılmaktadır (Xu vd., 1998).



Şekil 4.19 Beklenen ve Gözlenen Uzaklık Dağılımlarının Karşılaştırılması (Xu vd., 1998).

Küme: D noktalardan oluşan bir veri seti olmak üzere, C kümesi aşağıdaki özellikleri sağlayan D verisetinin bir alt boyutudur.

(1) $NNdistSet(C)$, yeterli güven seviyesi olan beklenen dağılıma sahip olmalıdır.

(2) C maksimaldir. Yani komşu gözlemlerden oluşan C'nin herbir uzantısı (1) numaralı koşulu yerine getirmez.

(3) C kümesi içindeki herhangi iki a,b noktası birbirine bağlı olmalıdır.

DBCLASD bir artımlı algoritmadır yani bir kümeye bir noktanın atanması, tüm küme veya tüm veri setini dikkate almadan yalnızca şimdiye kadar işlenen noktalara dayanmaktadır. Algoritma, kümeleri onların yakın komşularını kümeye alarak büyütmektedir. Bu büyüme kümelerin beklenen uzaklık dağılımına uyma koşulu altında yapılmaktadır. Bu koşul sağlanmadığı durumlarda küme büyümesi durdurulur. Bir aday nesne süreçteki kümeye atanmadan önce bu küme ile arasında uygun bir ilişki olup olmadığı kontrol edilmelidir.

Aday gözlem üretme: Bir kümenin adayları alan soruşturması yardımıyla oluşturulur. Bir bölge araştırması, çember gibi belirli bir alanın (yarıçap) içine düşen gözlemlerdir. C kümesinin herbir yeni p üyesi için yeni bir alan araştırması yapılır ve o nesnenin etrafındaki gözlemler kümeye alınır. Algoritmada bu alanın yarıçapı m için en uygun değer, kümenin herbir noktasının en yakın komşuya olan daha büyük bir uzaklık beklenmediği durumdaki yarıçap değeri olarak gösterilmektedir. Büyük m seçimi χ^2 testi için daha çok aday olacağı ve algoritmanın verimliliğinde ise düşüş olacağı anlamına gelmektedir. Yarıçap m değerini hesaplama C kümesindeki tekdüze dağılıma sahip gözlemlere dayanmaktadır. A, C kümesinin bir alanı ve N ise bu kümeye ait elemanların sayısı olsun. Bu durumda m için bir koşul aşağıdaki gibidir (Xu vd., 1998).

$$N \times P(NNdist_C(p) > m\ ) < 1) \qquad (3.5)$$

Yukarıdaki 3.4 numaralı denklem burada uygulanır ise aşağıdaki denklem elde edilir.

$$(1 - \pi m^2/A)^N < 1/N$$

$$m > \sqrt{A/\pi(1 - 1/N^{1/N})} \qquad (3.6)$$

Adayların test edilmesi: DBCLASD algoritması adayları üretme ve test etme ile üretilen kümeler arasında bağlılık olduğunu söylemektedir. Kümeleme algoritmalarında bu durum yerine oluşan kümelerin ayrık olması istenir. Tüm kümelerin uzaklıkları beklenen uzaklık dağılımına uyuyor durumda olsa bile, bu durum kümelerin herbir alt boyutu için geçerli olmayabilir. Bu sebeple adayları test etme süreci oldukça önemlidir. DBCLASD algoritması yukarıda bahsedilen bağlılığı ortadan kaldırmak için iki önemli özelliği birleştirmekten yola çıkmaktadır.

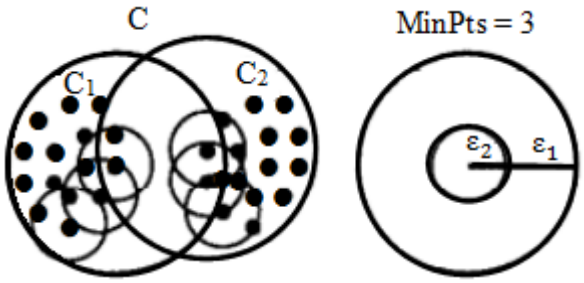
1. Başarısız adaylar hemen işlemden çıkarılmaz daha sonra tekrardan teste sokulurlar.
2. Bir kümeye dahil edilen adaylar daha sonradan başka bir kümeye geçebilir.

Başarısız adaylar bir yerde tutulur. Tüm adaylar süreçten geçtikten sonra başarısız gözlemler yeniden test edilir. Buradakilerin çoğu büyüyen küme sayesinde kümenin uzaklık dağılımına uyarak kümeleneceklerdir. Bu ana kadar anlatılanlar ile oluşan kümeler yukarıda yapılan küme tanımının maksimallik kavramı ile çelişkilidir. Bu nedenle hatalı küme oluşumundan kaçınmak için DBCLASD algoritması komşu kümeleri birleştirmeye çalışmaktadır. Bu birleştirmeyi bir aday başka bir kümeye ait olsa bile yapmaktadır (kümelenen gözlemler tekrardan küme değiştirebilir olmasından dolayı). Bu özellik sayesinde iyi sonuçlar elde edilse bile bir nesnenin birçok küme değiştirebileceğinden dolayı hesaplama zorluğu meydana getirebilir. Adayların test edilmesi sürecinde ise işlemdeki küme adaylarla birlikte genişlemeye başlar. Sonrasında ise en yakın komşu uzaklıkları halen daha beklenen uzaklık dağılımına uyup uymadığı incelenir. Algoritma tüm gözlemler sürece alındığında ve kümeler arası değişim durduğunda sonlanır (Xu vd., 1998). DBCLASD algoritmasının çalışma yapısı genel olarak aşağıdaki adımlarla gösterilmektedir.

1. Bir C kümesi oluşturulur ve veri setindeki bir p noktası bu kümeye alınır.
2. Küme komşu gözlemler ile genişletilir (χ^2 testi testi için minimum 30 gözlem gerektiğinen dolayı bu işlem χ^2 testi yapılmadan k en yakın komşu değerleri kullanılarak yapılır).
3. C kümesindeki herbir nokta için yarıçap değeri hesaplanır ve bu yarıçap değeri kullanılarak oluşturulan çember içinde kalan gözlemler listelenir ve bu listedekiler aday listesine eklenir.
4. Aday listesindekiler sırasıyla listeden çıkarılarak C kümesine atanır. Eğer C kümesinin uzaklık seti beklenen dağılım koşulunu sağlıyorsa p kümede kalır. Aksi durum söz konusu ise p, C kümesinden çıkarılır ve atanmamışlar listesine eklenir.
5. Bu süreç tüm gözlemler işleme sokulana ve işlemlerde kümeler arası bir değişiklik kalmayana kadar devam eder.

4.3.6 OPTICS

OPTICS (Ordering Points to Identify the Clustering Structure) algoritması, Ankerst, Breunig, Kriegel ve Sander tarafından ilk olarak 1999 yılında SIGMOD'99 konferansında sunulmuştur. OPTICS algoritmasında, kümeleme yapısını oluşturmak için nesneler sıralanmaktadır. Bu sıralama, küme yapısının etkili bir şekilde analizini desteklemek için grafiksel olarak görüntülenmektedir (Ankerst vd., 1999). DBSCAN algoritması uygulanırkan başlangıçta kullanıcı tarafından belirlenmesi şart olan Eps ve MinPts adet parametre bulunmaktadır. Bu parametreleri doğru olarak belirlemek oldukça zor olmakla birlikte özellikle veri sayısı büyüdükçe bu durum iyice zorlaşabilmektedir. Ayrıca bu parametrelerin yanlış bir şekilde oluşturulması kullanıcıyı hatalı sonuçlara götürecektir. DBSCAN algoritmasında kullanılan Eps değerinin küme belirlemedeki önemi Şekil 4.4.20'de gösterilmektedir. DBSCAN algoritmasındaki bu dezvantajların ortadan kaldırılması için OPTICS algoritması önerilmiştir (Pasin, 2015). OPTICS algoritmasında kullanıcı tarafından başlangıçta sadece MinPts değerleri girilmektedir.



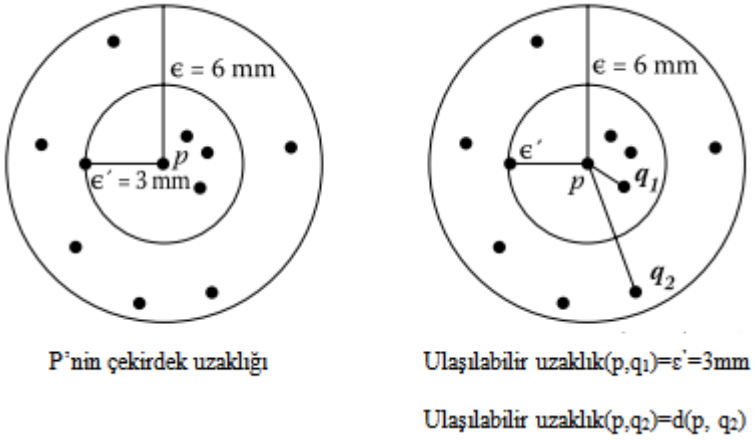
Şekil 4.4.20 Eps1 ve Eps2 Parametrelerine Göre Belirlenen Küme Sayılarının Değişimi.

Şekil incelendiğinde $Eps_1 < Eps_2$ önkoşulu altında C_1 ve C_2 kümelerinin yoğunluğa dayalı olarak kümeleştiği ve C kümesinin Eps_1'e göre bir küme oluşturduğu görülmektedir. Ayrıca C kümesi C_1 ve C_2 kümelerini de içermektedir. Özet olarak burada Eps_1 koşulu altında C kümesi, Eps_2 koşulu altında ise ayrı ayrı C_1 ve C_2 kümesi oluşturulmuştur (Silahtaroğlu, 2004). OPTICS algoritması DBSCAN algoritmasının geliştirilmiş hali olarak düşünülebilir. DBSCAN algoritmasının aksine OPTICS kendisi bir Eps değeri oluşturur. OPTICS algoritması verilen bir veri yapısındaki nesneleri kümelemek yerine nesnelerin yoğunluğa dayalı kümeleme yapısını ortaya koyan bir küme sıralaması geliştirir. Bu sıralama oluşturulurken kullanılan çekirdek uzaklığı (core-distance) ve ulaşılabilir uzaklık (reachability-

distance) adı verilen iki yeni kavram ortaya çıkmaktadır. Bu kavramlar aşağıdaki şekilde tanımlanmaktadır (Ankerst vd., 1999).

Çekirdek uzaklığı (Core-distance): p gözlemi D veritabanının bir elemanı, Eps mesafe değeri ve $N_{Eps}(p)$ p nesnesinin Eps komşusu olsun. p gözleminin çekirdek uzaklığı, p gözlemini çekirdek gözlem yapan (p gözlemi $N_{Eps}(p)$ içinde bir iç gözlem olmak şartıyla) en küçük Eps değeridir. Eğer p noktası çekirdek gözlem değilse p gözleminin çekirdek uzaklığı tanımsızdır (Ankerst vd., 1999).

Ulaşılabilir uzaklık (Reachability-distance): Bir p gözleminin başka bir gözlem olan *o* noktasına olan ulaşabilir uzaklığı en küçük uzaklıktır. Çünkü *o* gözlemi bir çekirdek gözlem olduğunda p, *o* noktasına göre yoğunluğa doğrudan erişebilirdir. Bu durumda o gözleminin ulaşılabilir uzaklığı çekirdek uzaklığından daha küçük olamaz. Bunun nedeni ise daha küçük değerlere sahip olan hiçbir gözlem *o* noktası üzerinden doğrudan yoğunluğa erişebilir olamaz (Ankerst vd., 1999).



Şekil 4.4.21 Çekirdek ve Ulaşılabilir Uzaklık (Harvey J. Miller & Jiawei Han, 2009).

Şekil 4.4.21 incelendiğinde p ile q_1 arasındaki ulaşılabilir uzaklık p'nin çekirdek uzaklığı olan ε' terimine eşit, p ile q_2 arasındaki ulaşılabilir uzaklık ise bu iki nesne arasındaki Öklit uzaklığına eşittir. OPTICS algoritmasının işleyişinin yapısı adımlar halinde aşağıda anlatılmaktadır (Sever, 2015).

1. İlk olarak veri setinden değişkenler seçilir ve bu seçilen gözlemler işleme alınmamışlar ise bu gözlemlerin Eps komşuluğundaki gözlemler bulunarak kaydedilir. Bunların dışında seçilen gözlemin ulaşılabilirlik ve çekirdek uzaklıkları da hesaplanır.

2. Gözlemin çekirdek gözlem olup olmadığı araştırılır. Çekirdek gözlem ise Eps ve MinPts değerleri dikkate alınarak doğrudan erişebilir olan noktalar bulunur ve bir listeye kaydedilirler. Seçilen gözlemin çekirdek gözlem olmadığı durumlarda ise bir sonraki gözlem işleme alınır.
3. Listeye kaydedilen gözlemeler kendilerine en yakın çekirdek gözlemine olan uzaklığa göre sıralanmaktadırlar. Sıralama işleminden sonra listedeki en küçük uzaklığa sahip olan gözlem seçilir ve bu gözlemin Eps komşuluğundaki gözlemlerin çekirdek uzaklıkları hesaplanır. Eğer en düşük uzaklığa sahip olan gözlem çekirdek gözlem ise bulunduğu gözleme önceki adımdan eklenebilecek gözlemler olabilmektedir.
4. Algoritmanın sonunda kümelere gözlem numaralarını atamak için her gözlemin ulaşılabilir ve çekirdek uzaklıklarının kümeleme parametresi olan ε' sayısından büyük olup olmama durumu araştırılır. Gözlemin her iki uzaklığının da bu parametreden büyük olduğu durumlarda gözlem gürültü olarak atanır. Eğer ulaşılabilir uzaklık değeri bu parametreden küçük ya da eşit ise nesne mevcut küme numarasına atanır. Eğer çekirdek uzaklığı bu parametreden küçük ise gözlem bir sonraki küme numarasına atanmaktadır.

OPTICS algoritması yapısal olarak DBSCAN algoritmasına benzerliğinden dolayı algoritmanın hesaplama karmaşıklığı, DBSCAN algoritmasının hesaplama karmaşıklığına eşit olup O(nlogn) olarak ifade edilmektedir. Burada n nesne sayısını ifade etmektedir.

4.4 Izgara Tabanlı Kümeleme Yöntemleri

Izgara Tabanlı Kümeleme (Grid-Based Clustering Algoritms) algoritmaları, veri uzayını sonlu sayıda hücrelere sahip ızgara yapısına böler ve kümeleme işlemleri oluşturulan bu ızgara yapısı üzerinde yapılır. Izgara yapısının oluşturulmasında sonlu sayıda hücre şeklinde veri uzayı bölümlenir ve bu yapıdaki hücrelerden kümeler oluşturulur (Cheng vd., 2014). Izgara tabanlı kümeleme algoritmaları, büyük hacimli ve çok boyutlu veri setlerinde oldukça etkili yöntemlerdir. Izgara tabanlı algoritmaların en önemli faydası özellikle büyük veri setleri için zamanı büyük ölçüde azaltmasıdır. Direk olarak noktaları kümelemek yerine, belirli hücreler içerisinde bulunan veri noktalarının komşu sınırlarlarını kümelemektedir. Çoğu uygulamada hücre sayısı veri nokta sayısından önemli seviyede düşük olacağı için bu

algoritmaların performansı oldukça yüksektir Izgara tabanlı kümeleme algoritmaları en etkili algoritmalar olmasına rağmen hücrelerin eşik değerleri ve ızgaraların büyüklüğü önceden belirlenmektedir ve bu değerlerin seçimi kümeleme sonuçlarını ciddi bir şekilde etkilemektedir. Bu değerlerin iyi bir şekilde belirlenememesi bu algoritmaların dezavantajı olarak görülmektedir (Akın, 2008). Izgara tabanlı kümeleme algoritma adımları aşağıdaki şekilde sıralanabilir.

- Veri uzayı hücrelere bölünerek ızgara yapısı oluşturulur.
- Tüm hücreler için hücre yoğunluğu hesaplanır.
- Hücreler yoğunluklarına göre sıralanır.
- Küme merkezleri oluşturulur.
- Komşu hücrelere geçişler yapılır.

Kümeler oluşturulurken hücre yoğunlukları kullanıldığından dolayı birçok ızgara tabanlı kümeleme algoritması yoğunluk tabanlı olarak da düşünülebilmektedir. Ayrıca bazı ızgara tabanlı algoritmalar, hiyerarşik kümelemeyi de içermektedir. Izgara tabanlı kümeleme algoritmaları bazı durumlara karşı duyarlıdır ve bu durumlar aşağıda ifade edilmektedir (Pasin, 2015).

Düzensizlik (Uniformity): Düzensiz veri dağılımlarında tek düze bir ızgara yapısı kullanmak küme kalitesinin istenilen seviyelere ulaşmasında yeterli olmayabilir.

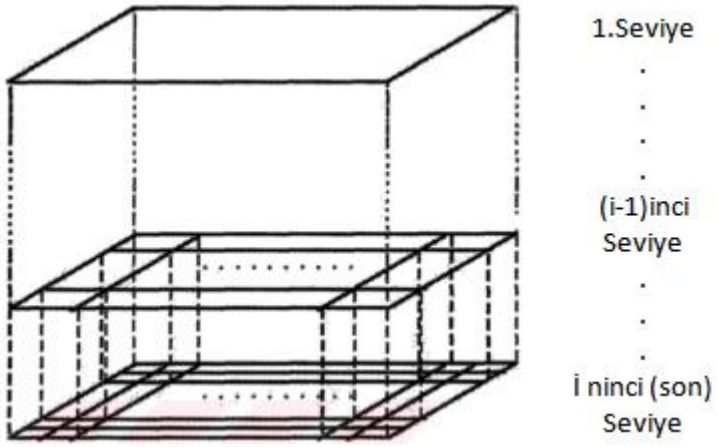
Yersellik (Locality): Veri noktalarının şeklinde ve dağılımında bölgesel değişkenlikler bulunmakta ise ızgara kümeleme yönteminin etkinliği, daha önceden belirlenen hücre sayısı, hücre sınırları ve önemli hücreler için yoğunluk eşik değeri ile sınırlandırılır.

Boyutluluk (Dimensionality): Önceden belirtildiği üzere algoritmanın performansı ızgara yapısının genişliğine bağlı olarak değiştiğinden boyut arttıkça ızgara yapısı da önemli derecede artacaktır ve yüksek boyutlu veriler için kümeler ölçeklenebilir olma özelliğini kaybedecektir.

Veri yapısında düzensizlik olması durumunda adaptif ızgara yöntemlerinden MAFIA algoritması kullanılabilir. Yersellik sorunu olduğunda ise eksen döndürme yöntemleri kullanılmaktadır. Bu yöntemlerden öne çıkanlar NSGC, ADCC, ASGC ve GDILC algoritmalarıdır. Yüksek boyutluluk sorununda ise CLIQUE, MAFIA, ENCLUS algoritmalarının kullanılması önerilmektedir

4.4.1 STING

STING (STatistical Information Grid-based Method), 1997 yılında Wang ve arkadaşları tarafından bölge odaklı sorguları kolaylaştırmak ve yersel veri tabanlarını kümelemek için geliştirilmiştir. STING algoritması girdi gözlemlerinin gömülü uzaysal alanlarının dikdörtgensel hücrelere ayrıldığı ızgara tabanlı, çoklu çözünürlüklü kümeleme tekniğidir (Han vd., 2012). Bu algoritmada çalışılacak mekansal alan dikdörtgen hücrelere bölünmekte ve hiyerarşik bir yapı oluşturulmaktadır. Hiyerarşik yapının temelinde, genellikle farklı seviyelerde hücreler vardır. Üst seviyelerdeki her hücre bir sonraki alt seviyedeki hücreleri oluşturmak için bölünmektedir. En üst seviyedeki (esas) hücre veri uzayına karşılık gelmektedir. Alt seviyelerdeki hücrelerin büyüklüğü ise nesnelerin yoğunluğuna bağlıdır. Şekil 4.4.22'de iki boyutlu uzayda STING algoritmasının hiyerarşik yapısını göstermektedir (Wang vd., 1997).



Şekil 4.4.22 STING Algoritmasının Hiyerarşik Yapısı.

Her bir hücre için

n: Var olan nesne sayısı

m: Hücrede bulunan tüm sayısal değerlerin ortalaması

s: Hücrede bulunan tüm sayısal değerlerin standart sapması

min: Nesnelerin minimum değerleri

max: Nesnelerin maksimum değerleri

dağılım: Nesnelerin dağılım türü

Dağılım tipi çoğunlukla normal, düzgün ve üsseldir. Dağılım tipi önceden biliniyorsa girilir, bilinmediği durumlarda ki-kare gibi testlerle belirlenebilir aksi takdirde gözlem "none" olarak atanmaktadır. Tüm bu istatistiksel bilgiler sorgulama

işlemlerinde kullanılmaktadır. Veriler yüklenirken tüm bu parametreler doğrudan hesaplanır. Yüksek seviyedeki hücrelerin parametreleri, alt seviyelerdeki hücrelerin parametrelerinden kolayca hesaplanabilmektedir (Wang vd., 1997). n, m, s, min, max, parametreleri mevcut hücre parametreleri ve n_i, m_i, s_i, min_i, max_i, parametreleri ise daha düşük seviyedeki hücrelerin parametreleri olsun. Bu durumda n, m, s, min, max ve parametreleri aşağıdaki denklemlerle hesaplanmaktadır.

$$n = \sum_i n_i \tag{4.1}$$

$$m = \sum_i m_i n_i \tag{4.2}$$

$$s = \sqrt{\frac{\sum_i (s_i^2 + m_i^2) n_i}{n} - m^2} \tag{4.3}$$

$$min = \min_i (min_i) \tag{4.4}$$

$$max = max_i (max_i) \tag{4.5}$$

Parametrelerden elde edilen istatistiksel bilgiler gridlere bağlı kalınarak yukarıdan aşağıya doğru kullanılmaktadır. Sıralamak gerekirse ilk olarak sorgulama başlatmak adına var olan hiyerarşik yapıdan bir katman (layer) seçilir. Genel olarak zorunlu olmamakla birlikte bu katman kök hücre olmaktadır ve az sayıda hücreden meydana gelmektedir (orta seviyedeki bir katmanda seçilebilmektedir fakat daha az alan olacağından dolayı seçilmez). Kök katmanı seçildikten sonra bu seviyedeki hücrenin sorguyu karşılama olasılığını elde etmek için güven aralıkları oluşturulur (Bu olasılıklar şartları sağlayan gözlemlerin oranı olarak tanımlanabilir). Oluşturulan bu güven aralıklarına göre hücreler ilişkili veya ilişkisiz olarak işaretlenir ve ilişkisiz hücreler sistem dışına çıkarılır ve ileri aşamalarda bir daha kullanılmaz. İşlemdeki katman ile sorgulama bittikten sonra bir sonraki alt katmana geçilir. Buradaki ana fark tüm hücrelerle işlem yapmak yerine sadece önceki tabakanın ilgili hücrelere bakmamızdır. Son seviyeye ulaşılana dek işlemlere devam edilir ve genelde algoritma tamamlandığında ilgili hücreler ve onların istatistiksel bilgileri ile sorguya tatminkar bir cevap elde edilmiş olur. Çok nadir karşılaşılır fakat bu yanıtı anlamlı düzeyde karşılayacak yeterli düzeyde hücre elde edilemediği durumlarda ise işlem sonunda kalan hücrelerle işlemlere devam edilir ve bu sayede daha anlamlı sonuçlar elde edilebilmektedir. Bu işlemler uygun sonuçlarla tamamlandıktan sonra işlem

öncesinden belirlenen yoğunluk kriterine göre tüm bölgeler analiz edilir. İncelenen bu bölgelerin yoğunluk kriterine uygunluğu belirlenir ve bu kriteri sağlayan bölgeler işaretlenir. Bu süreç tüm uygun hücreler işleme alınana kadar devam eder (Silahtaroğlu, 2004). STİNG algoritmasının adımları aşağıdaki gibi özetlenebilir (Wang vd., 1997).

1. Başlamak için bir katman (seviye) belirlenir.
2. Başlangıç seviyesindeki her bir hücre için sorgu karşılama olasılıkları güven aralıkları hesaplanır.
3. Yukarıdaki bilgiler ışığında hücreler ilişkilendirilir.
4. Başlangıç seviyesi alt seviye ise Adım 6'ya aksi takdirde adım 5'e geçilir.
5. Hiyerarşi yapısında bir seviye aşağı inilir. Bu seviye üst seviyedeki katmanın ilgili hücrelerini oluşturan hücrelerden oluşacağından 2. Adıma gidilir.
6. Bütün ilişkili hücreler için, geçerli hücre ile diğer bütün hücreler için verilen uzaklık incelenir.
7. Birbiri ile ilişkisiz hücre kalmayana kadar Adım 4 tekrarlanır ve artık sadece bir küme söz konusudur.
8. Bütün ilişkili hücreler atanıncaya kadar Adım 4 ve Adım 5 tekrarlanır ve algoritma sonlanır.

Algoritma bu işlemleri $O(n)$ zaman karmaşıklığında yapmaktadır. Burada n toplam nesne sayısını temsil etmektedir. Hiyerarşik yapı oluşturulduktan sonra eğer ağaç hücrenin k adet yaprağı var ise işlem süresi $O(k)$ olup bu değer genellikle toplam nesne sayısından daha küçük bir değere sahiptir. STING algoritmasının diğer kümeleme tekniklerine göre avantajları; verilerin özet bilgisini hücrelerdeki istatistiki bilgileri sorgulamadan bağımsız olarak verir. Kümeleme işlemi veri seti bir kere taranarak yapılmaktadır ve bu özelliği en önemli avantajlarından sayılmaktadır. Izgara yapısı sayesinde algoritma paralel ve kademeli güncelleme işlemlerini kolayca yapabilmektedir. STING algoritmasında ızgara yapısı nedeniyle kümeler arası sınırlar ya dikey ya da yatay olabilmekte, köşegen yapıda sınırlar bulunmamaktadır. Bu nedenle hızlı bir algoritma olmasına karşın bu durum oluşan kümelerin doğruluğunu ve kalitesini azalmaktadır.

4.4.2 Dalga Kümeleme Algoritması

Izgara tabanlı bir kümeleme algoritması olan WaveCluster (Clustering Using Wavelet Transformation, WaveCluster) G. Sheikholeslami ve arkadaşları tarafından

1998 yılında 24. VLDB konferansında sunulmuştur (Sheikholeslami vd., 1998). Hassas bir kümeleme işlemi gerçekleştirdiğinden dolayı çok boyutlu olan büyük verilerin kümelenmesi ve değişik rasgele şekiller içeren küme yapıları için kullanılabilmektedir. Bu algoritma, veri uzayına dalga dönüşümü uygular ve bu dönüşümlerle yoğun bölgeler ortaya konularak kümeler bulunur. Dalga dönüşümü bir sinyal işleme tekniğidir ve bir sinyali farklı frekanslı alt bantlara ayrıştırır. Bu yöntemin mekansal verilere uygulanmasında, mekansal veriler çok boyutlu sinyaller olarak görülür ve bu verileri frekans alanına dönüştürmek için dalga (wavelet) dönüşüm teknikleri uygulanır. Dalga kümeleme algoritmasının mekansal verileri uygulanabileceği düşüncesi, çok boyutlu mekansal veri setindeki nesnelerin d boyutlu bir uzayda temsil edilebilir olmasından gelmektedir. Bu uzaydaki mekansal nesnelerin sasyısal özelliklerini gösteren özellik vektörleri vardır. Özellik vektörleri her bir boyutu bir özelliği yansıtan, özellik uzayı denen mekansal bir alanda gösterilebilir durumdadır. D özellikli bir nesne için özellik vektörü, d boyutlu bir özellik uzayında bir nokta olarak temsil edilir. Veri setinin kümelenmesi yoğun ve seyrek bölgeleri belirleyerek yapılmaktadır ve bu sayede özellik vektörlerinin genel dağılım yapısı keşfedilmiş olur. Özellik uzayına toplanan tüm nesneler d boyutlu bir sinyal oluşturur. Yüksek frekanslı sinyaller özellik uzayında nesnelerin dağılımının ani değişimler olduğu alanları (sınırları) ifade etmektedir. Düşük frekanslı sinyaller ise özellik uzayındaki yoğun bölgeleri bir başka ifade ile kümelerin kendilerini ifade etmektedir (Sheikholeslami, Chatterjee, & Zhang, 2000).

Dalga dönüşümü kullanmanın amacı, veri noktalarını d-boyutlu sinyal olarak almaktır (d boyut sayısı). Dalga kümeleme algoritması, dönüşümler yapıldıktan sonra veri uzayındaki yoğun bölgeleri bu dönüşümü tamamlanmış uzayda aramaktadır (Han vd., 2012). Bir dalganın yüksek frekanslı sinyalleri, küme sınırları gibi daha seyrek bölgelere karşılık gelmekteyken düşük frekanslı sinyallerin parçaları küme içi gibi nesnelerin birbirine daha yakın olduğu bölgelere karşılık gelmektedir. Yukarıda da bahsedildiği üzere algoritma, önce çok boyutlu bir veri ızgara yapısı oluşturmaktadır ve eldeki veriyi bu yapıya yerleştirmektedir. Sonrasında gerçek veri uzayı çok boyutlu sinyaller olarak kabul edilerek ızgarada bulunan tüm hücrelere sinyal işleme tekniği uygulanmaktadır. Bu sebeple dalga kümeleme algoritması hem ızgara tabanlı hem de yoğunluk tabanlı bir algoritma olarak düşünülebilir. Genel

olarak dalga kümeleme algoritmasının adımları aşağıdaki şekilde sıralanabilir (Sheikholeslami vd., 2000):

1. Çok boyutlu özelllik uzayı ızgara yapısına bölünerek tüm gözlemler alt hücrelere yerleştirilir.
2. Dalga dönüşümü oluşturulan özellik uzayına uygulanır.
3. Dönüştürülen özellik uzayının alt bantlarındaki ilişkili kümeler ortaya çıkarılır.
4. Kümeler ilişkili alt uzaylarına göre etiketlenerek atanır.
5. Gözlem seçim tablosu oluşturulur.
6. Gözlemler kümelere atanır.

Dalga kümeleme algoritması denetimsiz kümeleme (unsupervised clustering) yöntemidir. Kümeleri belirlemek için kullanılan filtreler zayıf bilgileri kendi sınırları içine bastırır. Bu filtreler sayesinde veri uzayındaki yoğun bölgeler yakın noktaları daha çok kendine doğru yaklaştırırken uzaktaki gözlemleri daha da uzaklaştırmaktadır. Bu durum sonucunda oluşan kümeler net ve belirgin bir şekildedir ve uç veriler elimine edilmiş olur (Silahtaroğlu, 2004). Dalga kümeleme algoritması yüksek boyutlu ve gürültülü mekasal verilerden etkilenmemekte ve başarılı sonuçlar elde etmektedir. Gelişigüzel şekilli (konkav, konveks veya rasgele) küme yapılılarını bulma konusunda başarılıdır. Mekansal veri madenciliğinde kümelenme algoritmalarında dalga dönüşümünü kullanan başka hiçbir yöntem mevcut değildir. Başlangıçta giriş parametresi olarak küme sayısı bildirmek zorunlu değildir fakat önceden bu konu hakkında bilgi sahibi olunması durumunda bu parametreyi işleme sokmak daha anlamlı sonuçlar doğurmaktadır (Akın, 2008).

Şekil 4.4.23 İki Boyutlu Nitelik Uzayı Örnekleri.

Dalga dönüşüm algoritmasıyla yapılan kümeleme çok katmanlı veya çok çözünürlüklü olduğundan değişik hassasiyet seviyelerinde kümeleme yapılabilir. Yukarıdaki şekilde iki boyutlu veri uzayının hassastan kabaya doğru üç değişik düzeyde yapılmış dalga dönüşümlerini göstermektedir. Bu dönüşümlü şekillerin sol üstündeki alan normal verinin dalga dönüşümü yapılmış halini, sağ üstteki alan yatay kenar, sol alttaki alan dikey kenarlar ve son olarak sağ alttaki alanda ise köşeler gösterilmektedir.

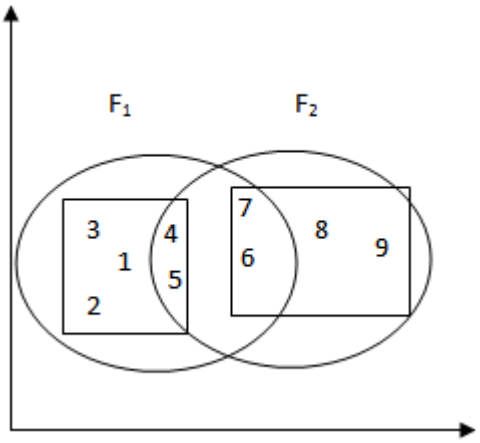
Dalga kümeleme algoritması nesne giriş sırasına duyarlı değildir. Çünkü veri setindeki gözlemler algoritmanın birinci adımında özellik ilişkisine göre hücrelere yerleştirildiğinde bu hücrelerin son hali gözlem sayısından bağımsız olmaktadır ve bu nedenle algoritmanın ilerleyen adımları bu hücreler ile uygulanacağından farklı sıradaki gözlemlerin sonuçlarında bir değişim olmaz (Sheikholeslami vd., 2000). Dalga kümeleme algoritması tüm veritabanını tarayarak gözlemleri ilgili kümelere atadığı için algoritmanın hesaplama karmaşıklığı $O(n)$ olmakla birlikte n veritabanındaki gözlem sayısını göstermektedir.

4.5 Bulanık Kümeleme

Geleneksel kümeleme yaklaşımlarında her gözlem sadece bir kümede olacak şekilde kümeler üretilmektedir. Bu nedenle sert bir kümelemede kümeler ayrıktır ve böyle olması beklenir. Bulanık kümeleme (Fuzz Clustering) ise geleneksel yaklaşımların aksine kümelerin birbirinden belirgin bir şekilde ayrılmadığı durumlarda ya da bazı gözlemlerin hangi kümeye ait olduğu konusunda belirsizlik olduğu durumlarda uygun bir yöntem olarak ortaya çıkmaktadır. Bulanıklık nesnel bir varlıktır. Sistem ne kadar karmaşıksa, onu doğru olarak tanımlamak o kadar zor olur ki bu da daha fazla bulanıklık yaratır (Shekhar vd., 2010). Bulanık kümeleme mekansal veri madenciliğinde belirsiz bir olasılığı tasvir eden bulanık üyelik fonksiyonu temelindeki bulanıklıkları kullanarak kümeleme yapmaktadır (Li et al. 2006). Bu uygulama her gözlemi her küme ile bir üyelik işlevi vasıtasıyla birbirine bağlayarak kümeleme kavramını genişletir. Bu algoritmalar çıktı olarak kümeler üretir fakat ayrık gruplar değildirler. Mekansal veri madenciliğinde bir sınıf ve nesne sırasıyla bir bulanık küme ve bir öğe olarak ele alınır. Her nesneye sınıfta bir üyelik atanır. İlgi evreninde ara sınırları olan birçok gözlem varsa, bir gözleme bir grup üyelik tahsis edilebilir. Üyelik yaklaşık 1'e yaklaştığında, elemanın sınıfa ait olma olasılığı

o kadar yükselmektedir. Örneğin, bir görüntüde toprak, nehir ve bitki örtüsü varsa, bir piksele üç üyelik tahsis edilir.

Normal kümeleme yöntemlerinde küme belirsizliği çoğunlukla öznel varsayım ve nesnel belirsizlikten kaynaklanmaktadır. Bulanık kümelemede her küme tüm gözlemlerin bulanık bir kümesidir. Buna örnek olarak aşağıdaki şekil gösterilebilir.



Şekil 4.4.24 Bulanık Küme Gösterimi.

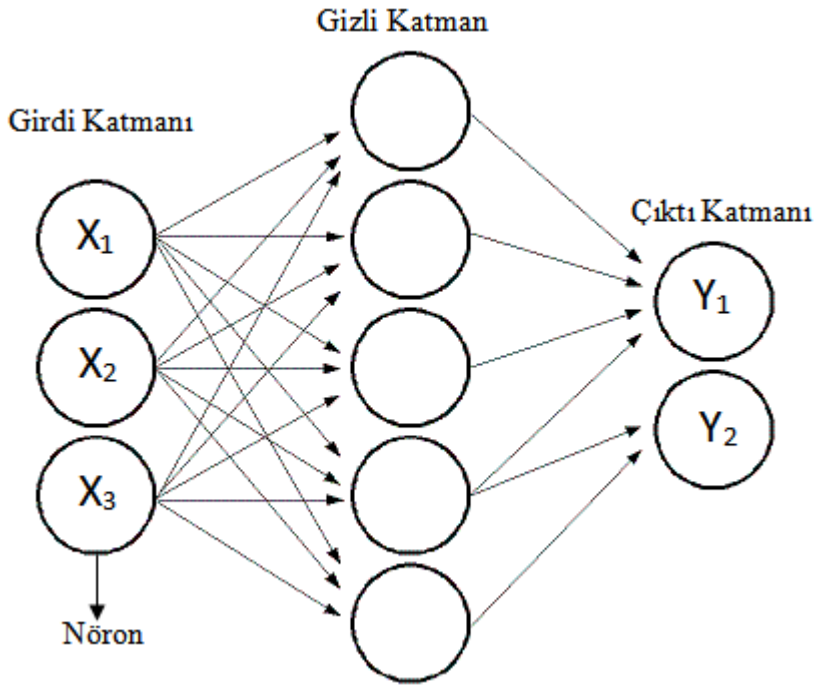
Dikdörtgenler veride iki küme içermektedir D_1={1,2,3,4,5} ve D_2={6,7,8,9}. Fakat bulanık kümeleme daireler içinde gösterilen F_1 ve F_2 adında iki bulanık küme oluşturabilir. Bu kümelerdeki gözlemler her bir küme için [1,0] üyelik değerlerine sahip olacaklar. Örnek ile açıklanırsa 1 numaralı nesne F_1 kümesi için [1,0.9] ile ifade edilmektedir. Burada 1 nesnenin sayısını 0,9 ise kümedeki üyelik değerini ifade etmektedir (Yavuzoğlu, 2009). Bulanık kümeleme mekansal veri madenciliğinde kullanıldığında bilgiyi ortaya çıkarmak daha kolay bir hal almaktadır (Wang ve Wang, 1997). Bu süreç temel adımlarla aşağıdaki gibi gösterilebilir.

1. Bir bulanık küme, her etki faktörü için bulanık değerlendirme matrisini edinir.
2. Tüm bulanık değerlendirme matrisleri, ilgili ağırlık matrisleri ile çarpılır. Ürün matrisi tüm faktörlerin kapsamlı matrisidir.
3. Kapsamlı matris kullanılarak temelinde eşdeğer bir matris olan bulanık benzer matris oluşturulur.
4. Bulanık kümeleme, önerilen maksimum geri kalan algoritmalar vasıtasıyla uygulanır.

En popüler bulanık kümeleme algoritması bulanık c-ortalama (FuzzyC-Means) algoritmasıdır. FCM, yerel minimumdan kaçınmak için k-ortalama algoritmasından daha iyidir.

4.6 Mekansal Kümeleme İçin Yapay Sinir Ağları

Genellikle "sinir ağı" olarak adlandırılan bir yapay sinir ağı, biyolojik sinir ağlarının yapısını ve işlevsel yönlerini taklit etmeye çalışan matematiksel bir model veya hesaplama modelidir. Yapay Sinir Ağları (YSA) bol miktarda komplike bağlantı noktası ile birbirine bağlanmış çok sayıda nörondan oluşur. YSA, yapısını öğrenme aşamasında ağ üzerinden akan harici veya dahili bilgilere dayalı olarak değiştiren uyarlanabilir bir sistemdir. Karmaşık bir ortamda mekansal veri madenciliğinin sınıflandırılması, kümelenmesi ve tahminini yapmak için uygundurlar. Bilgi işleme için harici veri girdisinin dinamik tepkisine bağlı olarak bir giriş katmanı, bir orta katman ve çıktı katmanı içermektedir. Derin bir sinir ağı, giriş ile çıktı katmanı arasında çok sayıda gizli katmana sahiptir ve bu da derin bir öğrenme yöntemi sunar. Çok sayıda nöron, eğitim numunelerinden öğrenerek desenleri belirlemek ve karmaşık bir sistemin doğrusal olmayan işlevlerini oluşturmak için ağa bağlanır.



Şekil 4.4.25 Yapay Sinir Ağları.

YSA'lar dağıtılmış depolama, ilişkisel bellek ve büyük paralel işleme işlevleri olan, oldukça doğrusal olmayan, ultra-büyük ölçekli, sürekli-zamansal ve dinamik sistemlerdir (Shekhar vd., 2010). Çok sayıda kompleks sistemin lineer olmayan fonksiyonlarını tasarlamak için bağlanırlar. Bu fonksiyonlar verilerden fonksiyonların yapısını öğrenmiş olur. Hebb öğrenme kurallarına dayanarak sinir ağları üç tipe ayrılabilir. Tahmin ve model tanımlama için ileri beslemeli ağlar, ilişkisel bellek ve optimizasyon hesaplaması için geri beslemeli ağlar ve kümeleme için kendi kendini organize eden ağlar. YSA'lar belirli bir şartta belirli bir analizin ayırt edici bir özelliğini açıklamak yerine, yapay zekada kullanılan ve bağlantılı yöntem olarak adlandırılan bilgiyi ifade eder. YSA'lar geleneksel yöntemlerle karşılaştırıldığında, model tanımada gürültünün etkisini azaltarak yüksek hata toleransı ve ağa dayanıklılık sağlar. Ayrıca sinir ağlarının kendi kendini düzenleyen ve kendini uyarlayan yetenekleri sayesinde kısıtlama büyük ölçüde rahatlatılır ve daha doğru sonuçlar elde eder.

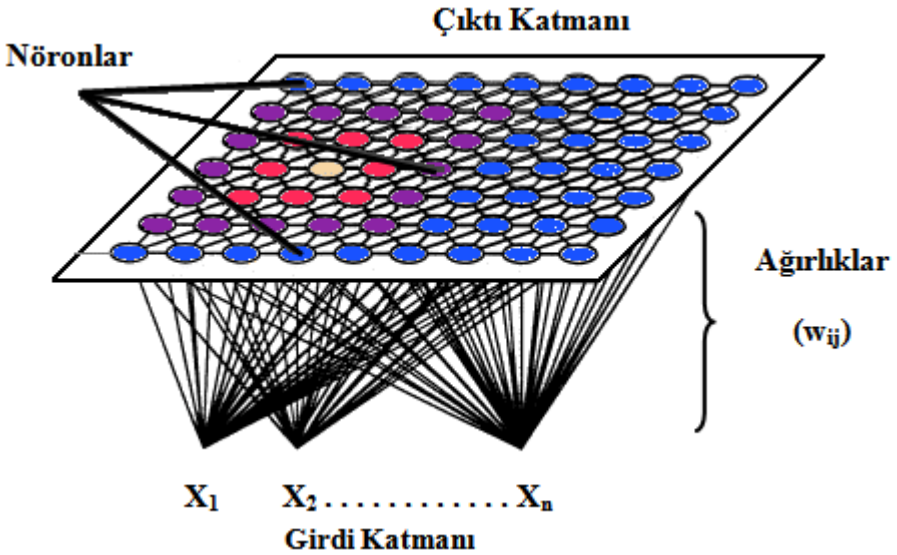
YSA'lar son otuz yıldır geniş çapta kullanılmaktadır. Rekabetçi veya başka bir deyişle, kazanan-tümünü-alır (winner-take-all) sinir ağları da denen bu yöntem genellikle girdi verilerini kümelemek için kullanılır (Jain ve Mao, 1996). Rekabetçi öğrenmede, benzer modeller ağ tarafından gruplanır ve nöron adı verilen tek bir birim tarafından temsil edilir. Bu gruplama işlemi, veri korelasyonlarına dayanarak yapılır. Kümeleme için kullanılan YSA'ların iyi bilinen örnekleri arasında Kohonen'in öğrenme vektörü nicelemesi (LVQ) ve kendi kendini düzenleyen harita (SOM) bulunmaktadır. YSA'ların mimarileri oldukça basittir. Giriş düğümleri ile çıktı düğümleri arasındaki ağırlıklar yinelemeli olarak değiştirilir. Bu işleme öğrenme süreci denir ve bir sonlandırma kriterine ulaşılana kadar devam eder. YSA'lar ayrıca kendi kendine öğrenme, kendi kendini örgütleme ve kendini uyarlama yeteneğine sahiptir. YSA'ların öğrenme prosedürleri, bazı klasik kümeleme yaklaşımlarındaki prosedürlere oldukça benzerlik göstermektedir (Yavuzoğlu, 2009).

Birçok girdi değişkenine sahip karmaşık bir doğrusal olmayan sistem göz önüne alındığında, ağın yakınsaklık, kararlılık, yerel minimum ve parametre ayarlamaları gibi sorunlarla karşılaşabilir (Lu vd., 1996). Örneğin, ağ parametrelerini (orta katmandaki nöronların sayısı) ve eğitim parametrelerini (Öğrenme oranı ve hata eşiği) tanımlamak bilgi gerektirir ve oldukça zordur. YSA eğitim verilerini daha sık taramayı gerektirebilir ve bu nedenle daha fazla zaman alabilir. Buna ek olarak,

keşfedilen bilgi açık kurallar yerine ağ yapısında saklanır. Ara sonuçlar için, keşfedilen bilgi daha rafine edilmiş kurallara dönüştürülmemektedir bu nedenle ağ yapısının kendisi bu aşamada bir tür bilgi olarak ele alınabilir. Bununla birlikte, mekansal veri madenciliğinin nihai sonuç aşamasında keşfedilen bilgiler kolay anlaşılamaz ve karar vermede pek açıklayıcı olmaz (Shekhar vd., 2010). Bu zayıflığın üstesinden gelmek için ileri beslemeli YSA algoritması, her adımda eğitim verisi ve gizli düğümler ekleyerek veri madenciliği sürecinde yavaş eğitim hızlarından, uzun öğrenme sürelerinden ve yerel minimum değerlerden kaçınmak üzere kullanılabilir (Lee, 2000). Sonuç olarak YSA'nın yeteneklerinden tam olarak yararlanmak ve bu eksikliklerinin üstesinden gelmek için diğer yöntemlerle birlikte kullanmak, mekansal veri madenciliğinde kullanımını kolaylaştıracaktır.

4.7 Özdüzenleyici Haritalar

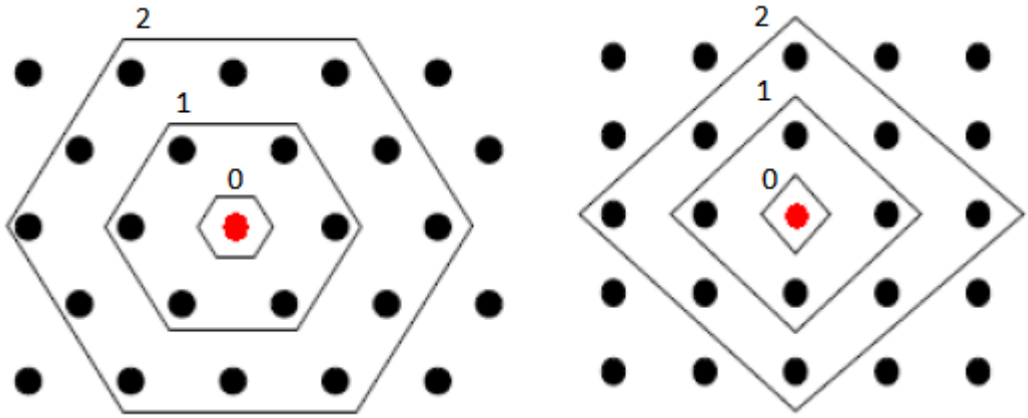
Özdüzenleyici haritalar (Self Organizing Maps, SOM), Tuevo Kohonen tarafından bulunan rekabetçi (competition) öğrenme temeline sahip olan yapay sinir ağının bir koludur (Kohonen, 1995). Özdüzenleyici haritalar literatürde ayrıca Kohonen Ağları olarak da anılmaktadır. Özdüzenleyici haritalar denetimsiz öğrenme yöntemidir bu nedenle kümeleme işlemi için uygun bir teknik olmakla birlikte yaygın olarak kullanılmaktadır (Tamayo vd., 1999). Özdüzenleyici haritalar çok boyutlu verilerin analizini ve sonuçlarının kolayca anlaşılabilir şekilde görselleştirmesini sağlayan bir algoritmadır.



Şekil 4.4.26 Özdüzenleyici Harita Yapısı.

Çoğunlukla bir adet giriş ve çıkış tabakasından oluşmaktadır. Giriş tabakasında tek boyutlu, çıkış tabakasında genellikle 2 boyutlu çeşitli geometrilerde işlemciler bulunmaktadır. Giriş ve çıkış tabakalarındaki her işlemci bir bağ ile bağlıdır. Bu bağlar çıkış işlemci elemanlarının referans vektörlerinde tutulmaktadır (Sarıoğlu vd., 2003). Özdüzenleyici harita ağlarının girdisi, mevcut verinin değişkenleridir ve girdi birimlerinin sayısı da değişken sayısına eşittir. Rekabet eden nöronlar kümeleme birimleridir. Nöronların sayısı kullanıcı tarafından belirlenmektedir ve probleme göre değişmektedir. Özdüzenleyici harita ağlarında, nöronlar arasında topolojik bir komşuluk ilişkisi olduğu varsayılmaktadır. Özdüzenleyici haritaların temsili yapısı yukarıdaki şekilde gösterilmektedir. Şekilde görülen rekabet içindeki nöronların her biri iki boyutlu olan bir uzayda koordinat bilgisine sahiptir. Burada her nöron girdi vektörleri ile aynı boyutta olan bir ağırlık vektörüne (W) sahiptir. Örneğin şekilde girdi verisi X_1, X_2, … , X_n vektörlerinden oluştuğundan her bir nöronun ağırlığı n boyutlu w_1,w_2, … , w_n şeklinde bir ağırlık vektörü olacaktır (Özdoğan, 2009).

Nöron ağırlık vektörleri, temsil ettikleri küme ile ilişkilendirilmiş girdi verileri için bir model teşkil etmektedir. Özdüzenleme (self-organization) işlemi sürecinde, her bir girdi için rekabetçi nöronlar arasından kazanan bir nöron seçilir. Seçilen nöron kendisine en çok benzeyen ağırlık vektörüne sahip olan nörondur. Bu seçim uzaklık ölçülerine dayanmaktadır. Hangi uzaklık ölçütünün kullanılacağı kullanıcı tarafından belirlenir ve kümeleme sonuçlarının başarısını etkilemektedir. Kazanan nörona ve kullanılan komşuluk topolojisi ile yarıçapına göre belirlenen komşu nöronlara ait ağırlık vektörleri güncellenir (Laurene, 1993).



Şekil 4.4.27 Özdüzenleyici Haritalarda Komşuluk İlişkilerinin Dikdörtgen Ve Altıgen Yapıda Gösterimi.

Birbirinden farklı yarıçap değerleri ile dikdörtgen ve altıgen örgü yapıları için örnek komşuluk ilişkileri yukarıdaki şekilde gösterilmektedir. Şekil 4.4.27'de altıgen örgü uygulamasında kırmızı nokta ile belirtilen kazanan nörona ait komşular "1" numaralı çizginin iç bölgesinde kalan altı birimden oluşmaktadır. Dikdörtgen örgü yapısında da benzer olarak kırmızı nörona ait komşular "1" numaralı çizginin iç bölgesinde kalan 4 birimden oluşmaktadır. Yatay, dikey ilişkiler kadar dik açılı ilişkiler de dikkate alacağından dolayı haritanın oluşacağı düzlemde altıgenimsi örgü dizilimi tercih edilmelidir. Eğitim aşaması mümkün olduğu kadar çok sayıda devirle devam etmelidir (Kohonen, 1995).

Etkin hale gelen nöron kazanan nöron (Best-Matching Unit, BMU) olarak anılır. Haykin tarafından ağın ilk olarak kurulmasından itibaren gerçekleşen süreç rekabet, ortak çalışma ve sinaptik uyum olmak üzere üç ana başlık altına toplanmıştır. Özdüzenleyici haritaların özellikleri aşağıdaki belirtilmiştir (Pyle, 2003).

- Özdüzenleyici haritalarda bütün girdi değişkenleri aynı ağırlığa sahiptir. Analizde diğerlerinden daha önemli bazı değişkenler var ise değişkenlere farklı ağırlıklar atanmalıdır. Bu durum tamamen kullanıcı merkezlidir ve sonucu önemli düzeyde etkileyebilmektedir.
- Özdüzenleyici haritalar yalnızca analiz sağlayabilecek görsel olarak anlaşılması kolay haritalar oluşturmaktadır. Değişkenler arasındaki ilişki hakkında bilgiler sunmaz. Bu durumda yukarıdaki maddede olduğu gibi kullanıcı odaklı olarak çözülebilir.
- Özdüzenleyici haritaların tahmin edici özelliği bulunmamaktadır. Haritalar sayesinde kullanıcının vereceği kararların ve çıkarımların daha iyi ve doğru olmasına katkıda bulunur.

Özdüzenleyici haritalar için tanımlı algoritma adımları aşağıdaki şekildedir (Özçalıcı, 2011).

1. Sinaptik ağırlık vektörlerine ilk değerler veri setinden rasgele seçilerek atanır. Öğrenme katsayısı, komşuluk derecesi ve komşuluk fonksiyonu atanır.
2. Bir uzaklık fonksiyonu yardımıyla girdiler ile her bir nöron arası uzaklık hesaplanır. Girdi verisine en yakın olan kazanan nöron (BMU) bulunur.
3. Parametrelere göre vektör güncelleştirmeleri yapılır. Sonrasında her girdi verisi için adım 2-3 tekrardan gerçekleştirilir.

4. Öğrenme ve komşuluk dereceleri güncellenir.
5. Çalışmanın sonlandırılması kontrol edilir. Eğer parametreler uygunda algoritmayı sonlandır.
6. Sonlandırma şartları sağlanana kadar adım 2-5 tekrarlanır.

4.8 Genetik Algoritmalar

Bir genetik algoritma, doğal seleksiyonda evrimsel canlıların hayatta kalmasının evrimsel sürecini taklit ederek mekansal veri madenciliğinde sınıflandırma, kümeleme ve tahminleme için en uygun çözümleri ortaya çıkarmada kullanılmıştır (Buckless ve Petry, 1994). Evrim'in bilgisayar üzerinde simülasyonlarının kullanılması üzerine ilk çalışmalar 1954'de Nils Aall Barricelli tarafından yapılmıştır. Genetik algoritmalar evrimsel algoritmalarının bir alt sınıfıdır. Evrimsel algoritmalar biyolojinin evrim teorisinden esinlendikleri için çaprazlama, mutasyon ve gen gibi terimler yardımıyla optimal çözüme ulaşmayı amaçlamaktadır. Bu terimleri örnekle açıklamak gerekirse gen için yaşlı nüfustan güçlü bireyleri seçerek yeni bir nüfus yaratma; çaprazlama için iki farklı kişinin parçalarını değiş tokuş ederek yeni bir birey yaratma ve mutasyon için bazı bireylerin belirli bir genini değiştirme örnek olarak verilebilir. Bu yinelenen yöntemler, tümdengelimli algoritmaların uyarlanabilir işlevi tarafından rehber alınarak uygulanır. Genetik algoritma çalıştırıldığında, çözülecek problem ilk önce bir başlangıç çözümü oluşturmak için kodlanır. Daha sonra uyarlama değerleri seçim, çaprazlama ve mutasyon yoluyla yeni çözümler üretmek üzere hesaplanır. Bu süreç en iyi çözüm bulunana kadar devam ettirilir. Bu sebeple genetik algoritmalar global optimal çözümü aramada sağlamlık ve alan bağımsızlığı avantajlarına sahiptir. Genetik algoritmalarındaki alan bağımsızlığı özelliği, karşılaşılan farklı alan ve boyutlardaki sorunlara karşı koyabilmekte ve uygulanabilir olduğunu göstermektedir.

Genetik algoritmalar, mekansal veri madenciliğinde bir noktanın optimal olup olmadığını hızlı bir şekilde araştırıp diğer noktalarla karşılaştırarak sonuca ulaşabilir (Shekhar vd., 2010). Genetik algoritma problemin optimal çözümüne ulaşmak için problem çözümlerini bir nesildeki bireyler (kromozom) olarak görmekte ve nesildeki kromozomlardan genetik operatörler yardımıyla yeni bireyler elde etmektedir. Genetik algoritmanın adımsal sürecini olarak göstermeden önce kullanılan bazı terimlerin tanımlamaları yapılmalıdır (Yünel, 2010).

Uygunluk (Fitness): Çözümleri birbiriyle kıyaslayabilmek için tanımlanmış bir fonksiyondur. Problemde elde edilen uygunluk değeri ne kadar büyükse/küçükse çözüm o kadar iyidir.

Seçilim (Selection): Yeni nesil oluşturulurken bireylerin nasıl seçileceğini belirleyen operatördür. Seçilen bireylere çaprazlama ve mutasyon süreçleri uygulanır. Seçilen bireyler ayrıca eşleşme havuzu (matching pool)'na atılır.

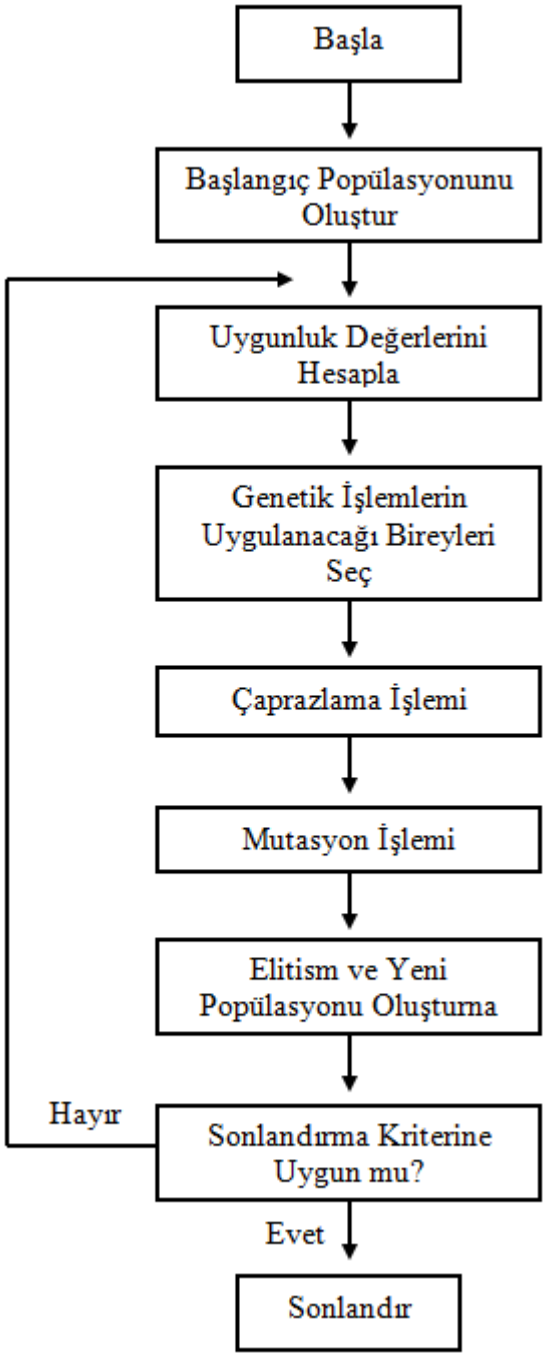
Rulet tekerleği seçimi (Roulette wheel selection): Rulet tekerleği seçilimi ile bireyler, uygunluk (fitness) değerlerinin büyüklüğüyle doğru orantılı bir şekilde seçilme şanslarını arttırırlar. Bir birey kendi uygunluk değerinin nesildeki toplam uygunluk değerine oranı sonucunda oluşan olasılık ile seçilir.

Seçkinlik (Elitism): Seçkinlik operatörü ile bir nesildeki en iyi birey ya da bireyler korunarak bir sonraki nesle aktarılır. Bu uygulamayla yeni neslin uygunluk (fitness) değerlerinin, eski neslin değerlerinden daha iyi olması ve en kötü durumda ise eşit olması sağlanmış olur.

Seçkin çözümler civarında arama (Elit search): Seçkinlik operatörü ile seçilen bireylerden ilk ikisi çaprazlanır ve sonradan ilk çözüm mutasyona uğrar buradan elde edilen çözümlere seçkinlik sınaması yapılır ardından bu bireyler bir sonraki nesle aktarılır.

Çaprazlama (Crossover): Eşleşme havuzundan rastgele alınan iki birey (kromozomun) belirli bir olasılık değeriyle ikiye bölünür ve bölünen bu parçalar karşılıklı olarak yer değiştirir. Bu şekilde iki yeni birey elde edilmiş olur ve bu sürece çaprazlama denmektedir.

Mutasyon (Mutation): Mutasyonun amacı popülasyonda çeşitlilik sağlamaktır ve rasgele seçilmiş gen noktalarının karşılıklı şekilde değiştirilmesi işlemidir. Mutasyon oranına göre bireylerin mutasyona uğrama sıklıkları ayarlanır. Mutasyon oranı çözüme ulaşmada önemli bir yeri bulunmaktadır.



Şekil 4.4.28 Genetik Algoritma Akış Şeması (Yapıcı, 2012).

5. ST-DBSCAN MEKANSAL-ZAMANSAL KÜMELEME ALGORİTMASI

Veri madenciliğindeki çoğu kümeleme algoritması, sıradan veri yapılarında (mekansal ve zamansal olmayan veriler) kümeleri keşfetmeye odaklandığından, mekansal-zamansal verileri kümelemek için kullanılmaları doğru olmaz. Mekansal-zamansal veriler, mekansal veri kümesinin zamansal dilimler halinde saklanan verilere karşılık gelir. Mekansal-zamansal verilerden bilgi keşfi, veri madenciliğinin umut verici bir alt alanıdır çünkü mekansal-zamansal veri yapısı günden güne çoğalmakta ve analiz edilmesi gerekmektedir. Mekansal-zamansal veriler için bilgi keşfi süreci mekansal ve zamansal olmayan veri yapıları ile karşılaştırıldığında daha karmaşıktır. Bunun nedeni ise mekansal-zamansal algoritmalar, veriden istenilen bilgiyi elde etmek için nesnelerin mekansal ve zamansal durumlarını göz önünde bulundurmak zorunda olmasıdır. MZVM'de en çok kullanılan yötemler arasında Mekansal-zamansal kümeleme üst sıralardadır. Mekansal-zamansal kümeleme algoritmaları hava tahminleri, tıbbi görüntüleme ve coğrafi bilgi sistemleri gibi birçok alanda kullanılmaktadır (Birant & Kut, 2007).

DBSCAN algoritmasını temel alan yeni yoğunluk tabanlı kümeleme algoritması olan ST-DBSCAN ilk olarak 2007 yılında Birant ve Kut tarafından oluşturulmuştur. DBSCAN algoritmasının mekansal-zamansal olan veri setleri için geliştirilmiş halidir. DBSCAN algoritması, mekansal verinin benzerliğini ölçmek için yalnızca bir uzaklık parametresi Eps kullanırken ST-DBSCAN algoritması eklenen zamansal değişken benzerliklerini tanımlamak için Eps_1 ve Eps_2 adında 2 adet uzaklık parametresi kullanmaktadır. Eps_1 mekansal verilerde iki gözlemin birbiriyle olan yakınlığını, Eps_2 ise mekansal olmayan ve zamansal değişkenlerdeki benzerliği ifade etmektedir. ST-DBCSAN algoritma yapısına geçmeden önce, DBSCAN algoritmasında da tanıtılan terimlerin tekrardan aşağıda tanımlanmıştır.

Komşuluk (Neighborhood): p ve q gözlemi D veri setinin elemanı olan iki gözlem olsun. Bu iki gözlem arasındaki komşuluk uzaklık fonksiyonları (Manhattan, Euclidean) ile tanımlanmaktadır ve dist(p,q) ile gösterilir.

Eps Komşuluğu (Eps-neighborhood): D veri setinin elamanı olan p noktasının Eps komşuluğu $N_{Eps}(p)$ ile gösterilir ve $N_{Eps}(p) = \{ q \in D \mid d(p,q) \leq \varepsilon \}$ şeklinde tanımlanır. gözlemlerin komşularını belirlerken kullanılan yakınlık mesafesi olan ε yarıçapı içindeki gözlemlerin komşuluğuna denir.

MinPts: Bir küme çevresinde bulunması gereken minimum nokta sayısını gösterir.

Çekirdek Gözlem (Core object): Bir gözlemin çekirdek gözlem olabilmesi için çevresinde Eps değeri koşuluna uyan Minpts sayısı kadar gözlem bulunması gerekmektedir.

Doğrudan yoğunluğa erişilebilirlik (Directly Density-Reachable): Aşağıdaki maddeler sağlandığı takdirde p gözlemi q gözlemi için doğrudan yoğunluk erişilebilirdir.

3. $p \in N_{Eps}(q)$
4. $\left|N_{Eps}(q)\right| \geq MinPts$

Doğrudan yoğunluğa erişebilirlik, çekirdek gözlemler için simetrik bir yapıdadır. Fakat bir çekirdek gözlem ve bir sınır gözlemi içeren durumlarda bu geçerli bir durum değildir.

Yoğunluğa erişebilirlik (Density reachable): Eps ve MinPts koşulları altında eğer $p_1, p_2, \ldots, p_n$ gözlemler zinciri varsa, $p_1 = q$ ve $p_n = p$ ise buradan yola çıkarak p_{i+1}, p_i'den doğrudan yoğunluğa erişebilirdir. Bu durumda p noktası q noktası üzerinden yoğunluğa erişebilir demektir. Yoğunluğa erişebilirlik, doğrudan yoğunluğa erişebilirliğin kanonik bir uzantısıdır. Bu ilişki geçişlidir fakat simetrik bir ilişki değildir.

Yoğunluk bağlantılılık (Density-connected): Bir *o* gözlem olsun ve p ve q gözlemleri Eps ve MinPts değerleri göz önüne alınarak *o* gözlemine yoğunluğa erişebilir durumda ise p gözlemi q gözlemi ile yoğunluk bağlatılıdır denir.

Küme: D gözlemlerden oluşan bir veritabanı olsun. Eps ve MinPts kuşullarını sağlayan bir K kümesi D'nin bir alt kümesidir.

(3) Her bir p ve q için, $p \in K$ ve q noktası p vasıtasıyla yoğunluğa erişebilirdir, $q \in K$

(4) Her bir $p,q \in K$ için p noktası q noktasıyla yoğunluk bağlantısallığına sahiptir.

Gürültü (Noise): $K_1,K_2,\ldots,K_n$, Eps ve MinPts koşullarını sağlamakta olan veri tabanındaki kümeler olsun. Veri tabanındaki herhangi bir K_i kümesine ait olmayan nokta ya da noktalara gürültü denir.

Sınır Gözlemi (Border Object): Eğer p gözlemi, bir çekirdek gözlem değilse ve başka bir çekirdek gözleminden yoğunluğa erişebilirlik var ise, bu gözleme sınır gözlemi denir.

Mekansal-zamansal veri kümelemede karşılaşılan problemler

Bir dizi gözlemin bir küme olarak kabul edilip edilmeyecek kadar benzer olup olmadığını belirlemek için, iki gözlemin ne kadar uzak olduğunu belirten bir uzaklık ölçüsü fonksiyonu (Manhattan, Euclidean ve Minkowski gibi) yardımıyla hesaplanan dist(i,j) değerlerine ihtiyaç duyulur. DBSCAN algoritması, yakınlığı ölçmek için Eps adında yalnızca bir uzaklık parametresi kullanır. Mekansal-Zamansal olan verilerin yapısı iki boyutlu olduğundan dolayı benzerlikleri tanımlamak için Eps_1 ve Eps_2 adında 2 adet uzaklık parametresi tanımlanır. Eps_1 mekansal verilerde iki gözlemin birbiriyle olan yakınlığı için, Eps_2 ise mekansal olmayan verilerde benzerliği hesaplaması için kullanılır. Örneğin $A(x_1,y_1)$ ve $B(x_2,y_2)$ mekansal bir konumu temsil eden iki nokta olsun. Ayrıca t_1, t_2 (Sabah hava sıcaklığı, Akşam hava sıcaklığı) A noktasının, t_3, t_4 ise B noktasının sıcaklık değerleridir. Burada Eps_1 A ve B lokasyonlarının yakınlıkları ile, Eps_2 ise iki noktanın sıcaklık değerlerinin benzerlikleri ile hesaplanmaktadır. $A(x_1,y_1,t_1,t_2)$ ve $B(x_2,y_2,t_3,t_4)$ iki noktasının Eps_1 ve Eps_2 değerleri aşağıdaki denklemler ile hesaplanmaktadır (Birant & Kut, 2007).

$$Eps_1 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5.1)$$

$$Eps_2 = \sqrt{(t_1 - t_3)^2 + (t_2 - t_4)^2} \quad (5.2)$$

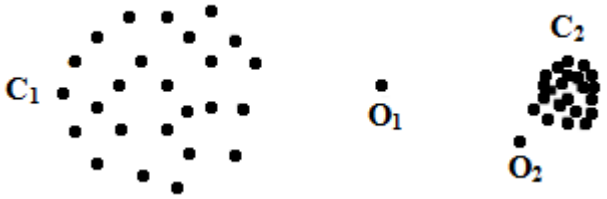
Gürültü Nesnelerini Belirleme Sorunu

Mevcut yoğunluk tabanlı kümeleme algoritmaları, belirli koşullar altında anlamlı ve yeterli sonuçlar verir; ancak farklı yoğunluklu kümeler mevcut olduğunda sonuçları tatmin edici değildir. Aşağıdaki şekilde 52 gözlem içeren bir veri setinde 25'er gözlemden oluşan C_1 ve C_2 kümeleri, o_1 ve o_2 gözlemleri ise gürültüdür. Burada C_2 kümesi C_1 kümesinden daha yoğun yapıda olduğu gözükmektedir. Kümelerin

yoğunlukları farklı olduğundan DBSCAN algoritması yalnızca o_1 noktasını gürültü olarak tanımlar. Bunun nedeni yaklaşık olarak C_1 kümesindeki her bir p gözlemi için, p gözlemi ile en yakın komşusu arasındaki uzaklık o_2 ile C_2 arasındaki uzaklıktan daha büyüktür. Bu nedenle doğru ve anlamlı sonuç için Eps giriş parametresine atanacak uygun bir değer belirlenemez. Örneğin seçilen Eps değeri o_2 ve C_2 arasındaki mesafeden düşükse, C_1'deki bazı gözlemler gürültü gözlemi olarak atanacaktır. Diğer taraftan Eps değeri o_2 ve C_2 arasındaki mesafeden büyükse, o_2 nesnesi gürültü nesnesi olarak atanmaz. Aşağıdaki şekildeki örnekte, farklı yoğunluklu kümeler mevcut olduğu durumda DBSCAN algoritmasının tatmin edici sonuçlar vermediği gösterilmiştir. Bu sorunun üstesinden gelmek için, ST-DBSCAN algoritmasında yoğunluk faktörü (density factor) adında yeni bir kavram önerilmiştir. Her bir küme için kümenin yoğunluğunu gösteren yoğunluk faktörü hesaplanmaktadır. Bu faktörü hesaplamak için yoğunluk uzaklığı (density distance) kullanılmaktadır (Birant & Kut, 2007).

Yoğunluk Uzaklığı (Density Distance): Bir p nesnenin maksimum yoğunluk uzaklığı, onun Eps değeri içindeki komşularını arasındaki maksimum uzaklığı olsun. Benzer bir şekilde Minimum yoğunluk uzaklığı ise p nesnesiyle Eps değeri içindeki komşuları arasındaki minimum uzaklık olsun (Birant & Kut, 2007).

$$\text{Max yoğunluk uzaklığı}(p) = \max\{\text{uzaklık}(p,q) | q \in D \wedge \text{uzaklık}(p,q) \leq \text{Eps}\}$$
$$\text{Min yoğunluk uzaklığı}(p) = \min\{\text{uzaklık}(p,q) | q \in D \wedge \text{uzaklık}(p,q) \leq \text{Eps}\}$$
$$\text{Yoğunluk uzaklığı}(p) = \text{Max yoğunluk uzaklığı}(p) / \text{Min yoğunluk uzaklığı}(p)$$



Şekil 5.5.1 Farklı Yoğunluklarda Kümeler İçeren Veri Seti Örneği (Birant & Kut, 2007).

Yoğunluk Faktörü (Density Factor): Bir C kümesinin yoğunluk faktörü aşağıdaki denklemle hesaplanmaktadır (Birant & Kut, 2007).

$$Yoğunluk\ Faktörü(C) = 1 / \left[\frac{\sum_{p \in C} Yoğunluk\ uzaklığı(p)}{|C|}\right] \qquad (5.2)$$

Bir C kümesinin yoğunluk faktörü o kümenin yoğunluk derecesini göstermektedir. Eğer C bir seyrek küme ise, min yoğunluk uzaklığı yüksek ve yoğunluk uzaklığı küçük olacaktır. Bu nedenle C nin yoğunluk faktörü 1'e yakınsayacaktır. Diğer taraftan C birbirine yakın sıkı bir küme ise minimum yoğunluk uzaklığı azalacak ve yoğunluk uzaklığı oldukça büyük olacaktır. Bu nedenle de C'nin yoğunluk faktörü 0'a yakınsayacaktır.

Bitişik Kümelerin Belirlenmesi Problemi

Mevcut yoğunluk tabanlı kümeleme algoritmalarının uygulanması kümeler birbirinden uzak olduğu durumlarda uygun olmaktadır. Kümeler birbirine yakın ya da bitişik olduğu durumlarda doğru sonuçlar vermemektedir. Eğer komşu gözlemler arasında çok az bir fark bulunmaktaysa, kümenin bir tarafındaki sınır gözlemlerinin değerleri karşı taraftaki diğer sınır gözlemlerin değerlerinden çok farklı olabilir. Bu sorun ST-DBSCAN algoritmasında kümeye yeni gelecek gözlem ile kümenin gözlemlerinin ortalaması karşılaştırılarak çözülmüştür. Eğer yeni gelen gözlem değeri ile küme ortalama değeri arasındaki fark belirlenen eşik değerinden (Δ_ϵ) büyükse yeni gözlem kümeye alınmaz (Birant & Kut, 2007).

ST-DBSCAN algoritmasının uygulanmasında Eps_1, Eps_2, MinPts ve Δ_ϵ parametrelerinin tanımlanması gerekmektedir. Eps_1, mekansal nitelikler (enlem ve boylam) için, Eps_2 ise mekansal olmayan değişkenler için uzaklık parametresidir. MinPts, bir noktanın Eps_1 ve Eps_2 değerleri içerisinde bulunan minimum nesne sayısıdır. Eğer bir bölge yoğunsa MinPts değerinden daha fazla puan içermelidir (Birant & Kut, 2007).

Eps ve MinPts parametrelerini belirlemek için birçok durumda etkili olan basit bir yöntem sunulmuştur. Bu yöntem n veri setinin boyutu olduğunda MinPts≈ln(n) olarak işleme almaktadır ve Eps bu değere bağlı olarak seçilmektedir. Bu yöntemin ilk adımı her nesne için k en yakın komşu uzaklıklarını belirlemektir ve burada k MinPts'ye eşittir. Sonrasında bu k-uzaklık değerleri büyükten küçüğe doğru sıralanır. Sonrasında sıralanan değerlerde ani düşüş olan nokta eşik noktasıdır ve Eps değeri bu nokta tarafından tanımlanan uzaklıktan daha düşük olarak seçilmelidir. Son parametre Δ_ϵ, komşu olan konumlardaki mekansal olmayan nesneler arasındaki farkın çok az düzeyde olmasından kaynaklanan birbirine girmiş kümeleri önlemek için kullanılmaktadır (Birant & Kut, 2007).

```
Algorithm ST_DBSCAN (D, Eps1, Eps2, MinPts, Δ∈)
        // Inputs :
                // D=(o1, o2, . . . , on) Set of objects
                // Eps1 : Maximum geographical coordinate (spatial) distance value.
                // Eps2 : Maximum non-spatial distance value.
                // MinPts : M,n,mum number of points within Eps1 and Eps2 distance.
                // Δ∈ : Threshold value to be included in a cluster
        // Output :
                // C = (C1, C2, . . . , Ck) set of clusters

Cluster_Label = 0

For i =1 to n                                                          // (i)
        İf oi is not in a clıster Then                                 // (ii)
                X = Retrieve_Neighnors (oi, Eps1, Eps2 )               // (iii)

                If |X| < MinPts Then
                        Mark oi as noise                               // (iv)
                Else                    //construct a new cluster      // (v)
                        Cluster_Label = Cluster_Label + 1

                        For j =1 to |X|
                                Mark all objects in X with current Cluster_Label
                        End For

                Push (all objects in X)                                // (vi)

                While not IsEmpty ( )
                        CurrentObj = Pop( )
                        Y = Retrieve_Neighbors (CurrentObj, Eps1, Eps2)

                        İf |Y| >= MinPts Then
                            ForAll Objects o in Y                      // (vii)
                                İf (o is not marked as noise or it is not in a cluster) and
                                        |ClusterAvg ( ) −  o.Value| <= Δ∈ Then
                                          Mark o with current Cluster_Label
                                          Push (o)
                                    End If
                                End For
                            End If
                        End While
                    End If
                End If
            End For
End Algorithm
```

Şekil 5.5.2 ST-DBSCAN Algoritma Yapısı.

Yukarıdaki algoritmada da görüldüğü üzere süreç veri setindeki D(i) ilk nokta seçilir. Bu nokta için tüm süreçler yapıldıktan sonra bir sonraki gözleme geçilir. Eğer seçilen gözlem hiçbir kümeye atanamaz ise (ii), Retrieve_Neighbors fonksiyonu adımına geçilir (iii). Retrieve_Neighbors (Nesne, Eps1, Eps2) çağrısı, Seçilen gözlem için Eps_1 ve Eps_2 parametrelerinden az uzaklığa sahip olan gözlemleri çağırır. Diğer bir deyişle Retrieve_Neighbors fonksiyonu Eps_1, Eps_2 ve MinPts parametrelerini göz

önünde tutarak seçilen gözlem ile yoğunluk erişilebilir olan tüm gözlemleri çağırır. Çağrılan tüm gözlemler Eps-komşuluğunu oluşturmaktadır. Burada komşuluk Retrieve_Neighbours(Nesne,Eps_1,Eps_2) ve Retrieve_Neighbours(object,Eps1) fonsiyonu ile Retrieve_Neighbours(object,Eps2) fonksiyonlarının etkileşimleri anlamına gelmektedir. Eğer Eps-Komşuluğundaki çağrılan tüm gözlemler MinPts parametresinden küçükse, seçilen gözlem gürültü olarak atanır (iv). Bunun anlamı seçilen gözlem küme oluşturmak için minimum gereken gözlem sayısını tutturamamaktadır ve bu nedenle küme oluşturmak için yetersizdir. İlerleyen zamanlarda eğer gürültü olarak tanımlanan gözlemler veri setindeki diğer gözlemlerle direk yoğunluk ulaşılabilir olmasa da, yoğunluk erişilebilir bir ilişki içerisine girerse gürültü olmaktan çıkıp bir kümenin sınır gözlemi olabilirler (Birant & Kut, 2007).

Eğer seçilen gözlem Eps_1 ve Eps_2 parametreleri içinde yeterli sayıda komşu gözleme sahipse ve bu gözlem bir çekirdek gözlem ise yeni bir küme oluşturulur (v). Sonrasında bu çekirdek gözlemin doğrudan yoğunluk erişebilir komşuları da bu oluşturulan kümeye dahil edilir. Sonrasında ise algoritma çekirdek gözlem ile yoğunluk ulaşabilir olan gözlemleri toplar. Toplanan yığın, doğrudan yoğunluk erişilebilir gözlemlerden yoğunluk erişilebilir gözlemleri bulmak için gereklidir. Eğer gözlem gürültü olarak işaretlenmediyse ve bir kümeye ait değilse ayrıca küme ortalaması ile yeni gelecek olan gözlem arasındaki fark Δ_{ϵ} eşik değerinden daha küçükse, gözlem mevcut kümeye yerleştirilir (vii). Seçilen gözlemin işlem süreci tamamlandıktan sonra algoritma bir sonraki gözlemi işleme sokar ve tüm gözlemler bu sürece dahil edilene kadar algoritma sürdürülür. Tüm bunlara ek bir bilgi olarak eğer iki kümesi birbirine çok yakın olursa herhangi bir p noktası iki kümeye de ait olabilir. Böyle durumlarda ST-DBSCAN algoritması bu noktayı ilk tanımlandığı kümeye atar (Birant & Kut, 2007).

DBSCAN algoritmasının ortalama çalışma zamanı karmaşıklığı önceden belirtildiği üzere *O*(n*logn) ‘dir. Burada n veritabanındaki nesnelerin sayısıdır. ST-DBSCAN algoritmasında yapılan değişiklikler ortalama çalışma zaman karmaşıklığını değiştirmemektedir. ST-DBSCAN algoritmasının süreçleri daha iyi anlaşılması için aşağıda 20 gözlemden oluşan veriseti ile örnek uygulama yapılmış ve sonuçlar adım adım gösterilmiştir.

Çizelge 5.1 Örnek Veri Seti.

İller	Enlem	Boylam	Sıcaklık	Tarih	Sayıya Çevrilmiş tarih
Adana	35,34	37,00	8,70	01.01.2017	42736
Mersin	34,60	36,78	9,70	01.01.2017	42736
Hatay	36,20	36,82	9,20	01.01.2017	42736
İstanbul	29,02	40,99	4,50	01.01.2017	42736
Kocaeli	29,92	40,77	4,30	01.01.2017	42736
Sakarya	30,39	40,77	4,00	01.01.2017	42736
Düzce	31,14	41,09	3,90	01.01.2017	42736
Van	43,35	38,47	-3,50	01.01.2017	42736
Manisa	27,40	38,62	4,20	01.01.2017	42736
İzmir	27,74	38,08	5,70	01.01.2017	42736
Adana	35,34	37,00	10,10	02.01.2017	42737
Mersin	34,60	36,78	9,80	02.01.2017	42737
Hatay	36,20	36,82	10,00	02.01.2017	42737
İstanbul	29,02	40,99	4,80	02.01.2017	42737
Kocaeli	29,92	40,77	4,50	02.01.2017	42737
Sakarya	30,39	40,77	4,80	02.01.2017	42737
Düzce	31,14	41,09	3,90	02.01.2017	42737
Van	43,35	38,47	-2,00	02.01.2017	42737
Manisa	27,40	38,62	5,00	02.01.2017	42737
İzmir	27,74	38,08	5,50	02.01.2017	42737

Yukarıdaki çizelgede örnekte kullanılacak olan 20 gözlemlik illerin mekansal, zamansal parametreleri ve sıcaklık bilgileri gösterilmiştir. Algoritma başlangıçta ilk gözlemin diğer gözlemler arasındaki mekansal uzaklık parametresi Eps_1 ve zamansal uzaklık parametresi Eps_2 değerlerini hesaplamaktadır. Aşağıda ilk gözlem ile ikinci gözlem için bu parametrelerin hesaplanması gösterilmiştir.

$$\mathbf{Eps_1} = \mathrm{d}_{1,2} = \sqrt{(\mathrm{Enlem}_1 - \mathrm{Enlem}_2)^2 + (\mathrm{Boylam}_1 - \mathrm{Boylam}_2)^2}$$

$$\mathbf{Eps_1} = \mathrm{d}_{1,2} = \sqrt{(34{,}34 - 34{,}60)^2 + (37{,}00 - 36{,}78)^2}$$

$$\mathbf{Eps_1} = \mathrm{d}_{1,2} = \sqrt{0{,}5476 + 0{,}0484}$$

$$= 0{,}7720$$

$$\mathbf{Eps_2} = \mathrm{d}_{1,2} = \sqrt{(\mathrm{tarih}_1 - \mathrm{tarih}_2)^2 + (\mathrm{sıcaklık}_1 - \mathrm{sıcaklık}_2)^2}$$

$$\mathbf{Eps_2} = \mathrm{d}_{1,2} = \sqrt{(42736 - 42736)^2 + (8{,}70 - 9{,}70)^2}$$

$$\mathbf{Eps_2} = \mathrm{d}_{1,2} = \sqrt{0 + 1} = 1$$

Çizelge 5.2 Öklid Uzaklığı Kullanılarak Hesaplanan Eps_1 Uzaklık Değerleri.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	**0,000**	**0,772**	**0,879**	7,474	6,602	6,222	5,862	8,144	8,104	7,676	**0,000**	**0,772**	**0,879**	7,474	6,602	6,222	5,862	8,144	8,104	7,676
2	**0,772**	**0,000**	**1,600**	6,990	6,150	5,800	5,527	8,912	7,431	6,982	**0,772**	**0,000**	**1,600**	6,990	6,150	5,800	5,527	8,912	7,431	6,982
3	**0,879**	**1,600**	**0,000**	8,303	7,419	7,026	6,621	7,338	8,982	8,553	**0,879**	**1,600**	**0,000**	8,303	7,419	7,026	6,621	7,338	8,982	8,553
4	7,474	6,990	8,303	**0,000**	**0,926**	**1,388**	2,122	14,550	2,871	3,179	7,474	6,990	8,303	**0,000**	**0,926**	**1,388**	2,122	14,550	2,871	3,179
5	6,602	6,150	7,419	**0,926**	**0,000**	**0,470**	**1,261**	13,626	3,313	3,462	6,602	6,150	7,419	**0,926**	**0,000**	**0,470**	**1,261**	13,626	3,313	3,462
6	6,222	5,800	7,026	**1,388**	**0,470**	**0,000**	**0,815**	13,163	3,683	3,776	6,222	5,800	7,026	**1,388**	**0,470**	**0,000**	**0,815**	13,163	3,683	3,776
7	5,862	5,527	6,621	2,122	**1,261**	**0,815**	**0,000**	12,488	4,482	4,541	5,862	5,527	6,621	2,122	**1,261**	**0,815**	**0,000**	12,488	4,482	4,541
8	8,144	8,912	7,338	14,550	13,626	13,163	12,488	**0,000**	15,951	15,615	8,144	8,912	7,338	14,550	13,626	13,163	12,488	**0,000**	15,951	15,615
9	8,104	7,431	8,982	2,871	3,313	3,683	4,482	15,951	**0,000**	**0,638**	8,104	7,431	8,982	2,871	3,313	3,683	4,482	15,951	**0,000**	**0,638**
10	7,676	6,982	8,553	3,179	3,462	3,776	4,541	15,615	**0,638**	**0,000**	7,676	6,982	8,553	3,179	3,462	3,776	4,541	15,615	**0,638**	**0,000**
11	**0,000**	**0,772**	**0,879**	7,474	6,602	6,222	5,862	8,144	8,104	7,676	**0,000**	**0,772**	**0,879**	7,474	6,602	6,222	5,862	8,144	8,104	7,676
12	**0,772**	**0,000**	**1,600**	6,990	6,150	5,800	5,527	8,912	7,431	6,982	**0,772**	**0,000**	**1,600**	6,990	6,150	5,800	5,527	8,912	7,431	6,982
13	**0,879**	**1,600**	**0,000**	8,303	7,419	7,026	6,621	7,338	8,982	8,553	**0,879**	**1,600**	**0,000**	8,303	7,419	7,026	6,621	7,338	8,982	8,553
14	7,474	6,990	8,303	**0,000**	**0,926**	**1,388**	**2,122**	14,550	2,871	3,179	7,474	6,990	8,303	**0,000**	**0,926**	**1,388**	2,122	14,550	2,871	3,179
15	6,602	6,150	7,419	**0,926**	**0,000**	**0,470**	**1,261**	13,626	3,313	3,462	6,602	6,150	7,419	**0,926**	**0,000**	**0,470**	**1,261**	13,626	3,313	3,462
16	6,222	5,800	7,026	**1,388**	**0,470**	**0,000**	**0,815**	13,163	3,683	3,776	6,222	5,800	7,026	**1,388**	**0,470**	**0,000**	**0,815**	13,163	3,683	3,776
17	5,862	5,527	6,621	2,122	**1,261**	**0,815**	**0,000**	12,488	4,482	4,541	5,862	5,527	6,621	2,122	**1,261**	**0,815**	**0,000**	12,488	4,482	4,541
18	8,144	8,912	7,338	14,550	13,626	13,163	12,488	**0,000**	15,951	15,615	8,144	8,912	7,338	14,550	13,626	13,163	12,488	**0,000**	15,951	15,615
19	8,104	7,431	8,982	2,871	3,313	3,683	4,482	15,951	**0,000**	**0,638**	8,104	7,431	8,982	2,871	3,313	3,683	4,482	15,951	**0,000**	**0,638**
20	7,676	6,982	8,553	3,179	3,462	3,776	4,541	15,615	**0,638**	**0,000**	7,676	6,982	8,553	3,179	3,462	3,776	4,541	15,615	**0,638**	**0,000**

Çizelge 5.3 Öklid Uzaklığı Kullanılarak Hesaplanan Eps_2 Uzaklık Değerleri.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	**0,000**	**1,000**	**0,500**	**4,200**	**4,400**	**4,700**	**4,800**	12,200	**4,500**	**3,000**	**1,720**	**1,487**	**1,640**	**4,026**	**4,317**	**4,026**	**4,903**	10,747	**3,833**	**3,353**
2	**1,000**	**0,000**	**0,500**	**5,200**	5,400	5,700	5,800	13,200	5,500	**4,000**	**1,077**	**1,005**	**1,044**	5,001	5,295	5,001	5,886	11,743	**4,805**	**4,317**
3	**0,500**	**0,500**	**0,000**	**4,700**	**4,900**	5,200	5,300	12,700	**5,000**	**3,500**	**1,345**	**1,166**	**1,281**	**4,512**	**4,805**	**4,512**	5,394	11,245	**4,317**	**3,833**
4	**4,200**	5,200	**4,700**	**0,000**	**0,200**	**0,500**	**0,600**	8,000	**0,300**	**1,200**	5,689	5,394	5,590	**1,044**	**1,000**	**1,044**	**1,166**	6,576	**1,118**	**1,414**
5	**4,400**	5,400	**4,900**	**0,200**	**0,000**	**0,300**	**0,400**	7,800	**0,100**	**1,400**	5,886	5,590	5,787	**1,118**	**1,020**	**1,118**	**1,077**	6,379	**1,221**	**1,562**
6	**4,700**	5,700	5,200	**0,500**	**0,300**	**0,000**	**0,100**	7,500	**0,200**	**1,700**	6,181	5,886	6,083	**1,281**	**1,118**	**1,281**	**1,005**	6,083	**1,414**	**1,803**
7	**4,800**	5,800	5,300	**0,600**	**0,400**	**0,100**	**0,000**	7,400	**0,300**	**1,800**	6,280	5,984	6,181	**1,345**	**1,166**	**1,345**	**1,000**	5,984	**1,487**	**1,887**
8	12,200	13,200	12,700	8,000	7,800	7,500	7,400	**0,000**	7,700	9,200	13,637	13,338	13,537	8,360	8,062	8,360	7,467	**1,803**	8,559	9,055
9	**4,500**	5,500	5,000	**0,300**	**0,100**	**0,200**	**0,300**	7,700	**0,000**	**1,500**	5,984	5,689	5,886	**1,166**	**1,044**	**1,166**	**1,044**	6,280	**1,281**	**1,640**
10	**3,000**	**4,000**	**3,500**	**1,200**	**1,400**	**1,700**	**1,800**	9,200	**1,500**	**0,000**	**4,512**	**4,220**	**4,415**	**1,345**	**1,562**	**1,345**	**2,059**	7,765	**1,221**	**1,020**
11	**1,720**	**1,077**	**1,345**	5,689	5,886	6,181	6,280	13,637	5,984	**4,512**	**0,000**	**0,300**	**0,100**	**5,300**	5,600	5,300	6,200	12,100	5,100	**4,600**
12	**1,487**	**1,005**	**1,166**	5,394	5,590	5,886	5,984	13,338	5,689	**4,220**	**0,300**	**0,000**	**0,200**	**5,000**	5,300	**5,000**	5,900	11,800	4,800	**4,300**
13	**1,640**	**1,044**	**1,281**	5,590	5,787	6,083	6,181	13,537	5,886	**4,415**	**0,100**	**0,200**	**0,000**	5,200	5,500	5,200	6,100	12,000	**5,000**	**4,500**
14	**4,026**	5,001	**4,512**	**1,044**	**1,118**	**1,281**	**1,345**	8,360	**1,166**	**1,345**	5,300	**5,000**	5,200	**0,000**	**0,300**	**0,000**	**0,900**	6,800	**0,200**	**0,700**
15	**4,317**	5,295	**4,805**	**1,000**	**1,020**	**1,118**	**1,166**	8,062	**1,044**	**1,562**	5,600	5,300	5,500	**0,300**	**0,000**	**0,300**	**0,600**	6,500	**0,500**	**1,000**
16	**4,026**	5,001	**4,512**	**1,044**	**1,118**	**1,281**	**1,345**	8,360	**1,166**	**1,345**	5,300	**5,000**	5,200	**0,000**	**0,300**	**0,000**	**0,900**	6,800	**0,200**	**0,700**
17	**4,903**	5,886	5,394	**1,166**	**1,077**	**1,005**	**1,000**	7,467	**1,044**	**2,059**	6,200	5,900	6,100	**0,900**	**0,600**	**0,900**	**0,000**	5,900	**1,100**	**1,600**
18	10,747	11,743	11,245	6,576	6,379	6,083	5,984	**1,803**	6,280	7,765	12,100	11,800	12,000	6,800	6,500	6,800	5,900	**0,000**	7,000	7,500
19	**3,833**	**4,805**	**4,317**	**1,118**	**1,221**	**1,414**	**1,487**	8,559	**1,281**	**1,221**	5,100	**4,800**	**5,000**	**0,200**	**0,500**	**0,200**	**1,100**	7,000	**0,000**	**0,500**
20	**3,353**	**4,317**	**3,833**	**1,414**	**1,562**	**1,803**	**1,887**	9,055	**1,640**	**1,020**	**4,600**	**4,300**	**4,500**	**0,700**	**1,000**	**0,700**	**1,600**	7,500	**0,500**	**0,000**

Parametreler :

Eps_1 = 2

Eps_2 = 5

MinPts = 4

1. Adım: İlk gözlem işleme alınır ve yukarıdaki parametreler değerleri içerisinde bulunan yakın gözlemler bulunur. Bulunan bu gözlemler aşağıda gösterilmektedir.

$\mathrm{Eps}_1(\,\mathrm{dist} \leq 2\,) = \{\,1, 2, 3, 11, 12, 13\,\}$

$\mathrm{Eps}_2(\mathrm{dist} \leq 5) = \{1, 2, 3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 20\,\}$

$\mathrm{Eps}_1 \cap \mathrm{Eps}_2 = \{1, 2, 3, 11, 12, 13\}$

Komşu gözlem sayısı = 6 > MinPts olduğundan dolayı 1. gözlem küme bir olarak atanır. Sonrasında 1. gözlemin parametre değerleri içinde olan 2, 3, 11, 12, 13 numaralı gözlemler için aynı süreç uygulanır. Fakat Çizelge5.2'de görüldüğü üzere bu gözlemler için Eps_1 parametre koşulunu sağlayan gözlemler farklılık göstermemektedir. Bu nedenle 2, 3, 11, 12, 13 numaralı gözlemler 1 numaralı kümeye atanır.

Küme 1 : 1, 2, 3, 11, 12 ve 13 numaralı gözlemlerdir. Yani Adana, Mersin ve Hatay illeri sıcaklık derecelerine göre aynı kümeye atanmıştır.

2. Adım : 4 numaralı gözlem sürece dahil edilir ve mekansal ve zamansal komşuları ortaya çıkarılır.

$\mathrm{Eps}_1(\,\mathrm{dist} \leq 2\,) = \{\,4, 5, 6, 14, 15, 16\,\}$

$\mathrm{Eps}_2(\mathrm{dist} \leq 5) = \{1, 2, 3, 4, 5, 6, 7, 9, 10, 14, 15, 16, 17, 19, 20\,\}$

$\mathrm{Eps}_1 \cap \mathrm{Eps}_2 = \{4, 5, 6, 14, 15, 16\}$

Komşu gözlem sayısı = 6 > MinPts olduğundan dolayı 4 numaralı gözlem yeni bir küme oluşturur ve küme 2 olarak işaretlenir. 4 numaralı gözlemin komşuları sırasıyla bu süreçten aynı şekilde geçerler.

5 numaralı gözlem için :

$\mathrm{Eps}_1(\,\mathrm{dist} \leq 2\,) = \{\,4, 5, 6, 7, 14, 15, 16, 17\,\}$

$\mathrm{Eps}_2(\mathrm{dist} \leq 5) = \{1, 3, 4, 5, 6, 7, 9, 10, 14, 15, 16, 17, 19, 20\,\}$

$\mathrm{Eps}_1 \cap \mathrm{Eps}_2 = \{4, 5, 6, \mathbf{7}, 14, 15, 16, \mathbf{17}\}$

5 numaralı gözlem için yapılan işlemler sonucunda Eps_1 ve Eps_2 parametre koşullarını sağlayan gözlemler arasında 7 ve 17 numaralı gözlemler (Düzce) 2

numaralı kümeye dahil edilmiştir. Sonrasında sırasıyla 6, 14, 15, 16 numaralı gözlemler için yapılacak komşuluk araştırmasına 7 ve 17 numaralı gözlemlerde dahil edilmiştir. Fakat Çizelge 5.1.2'de de görüleceği üzere Eps_1 uzaklığı koşulunu sağlayan farklı bir gözlem bulunmamaktadır. Sonuç olarak küme 2 için kesin sonuç 4, 5, 6, 7, 14, 15, 16 ve 17 numaralı gözlemlerdir (İstanbul, Kocaeli, Sakarya ve Düzce).

3. Adım : 8 numaralı gözlem için komşuluk araştırması yapılır.

$\mathrm{Eps}_1(\,\mathrm{dist} \leq 2\,) = \{\,8, 18\,\}$
$\mathrm{Eps}_2(\mathrm{dist} \leq 5) = \{\,8, 18\,\}$
$\mathrm{Eps}_1 \cap \mathrm{Eps}_2 = \{\,8, 18\,\}$

Yapılan araştırma sonucunda komşuluk koşullarını sağlayan gözlemler 8 ve 18 numaralı gözlemlerdir. Gözlem sayısı = 2 < MinPts olduğundan dolayı 8 numaralı gözlem gürültü olarak işaretlenmektedir.

4. Adım : 9 numaralı gözlem için komşuluk araştırması başlatılır.

$\mathrm{Eps}_1(\,\mathrm{dist} \leq 2\,) = \{\,9, 10, 19, 20\,\}$
$\mathrm{Eps}_2(\mathrm{dist} \leq 5) = \{\,1, 4, 5, 6, 7, 9, 10, 14, 15, 16, 17, 19, 20\,\}$
$\mathrm{Eps}_1 \cap \mathrm{Eps}_2 = \{\,9, 10, 19, 20\,\}$

9 numaralı gözlemin komşu sayısı = 4 ≥ MinPts olduğundan dolayı 3 numaralı yeni bir küme oluşturulur ve 9. gözlem bu kümeye dahil edilir. 9 numaralı gözlemin komşuları sırasıyla aynı şekilde bu komşu arama sürecinden geçer. Bu gözlemler için yapılan araştırmada Çizelge 5.1.2'deki uzaklıklara bakıldığında mekansal olarak yakın olan yeni bir gözlem bulunmadığı görülür. Bu nedenle küme 3 için araştırma son bulur ve kesin sonuç 9, 10, 19 ve 20 numaralı gözlemlerdir (İzmir ve Manisa).

4. Adım : Herhangi bir kümeye dahil edilmemiş olan tek gözlem 18 numaralı gözlemdir. Bu gözlem için yapılan araştırmada sonucunda herhangi bir yakın gözlem bulunamamıştır. Bu nedenle 18 numaralı gözlem gürültü olarak işaretlenir ve algoritma sonlandırılır.

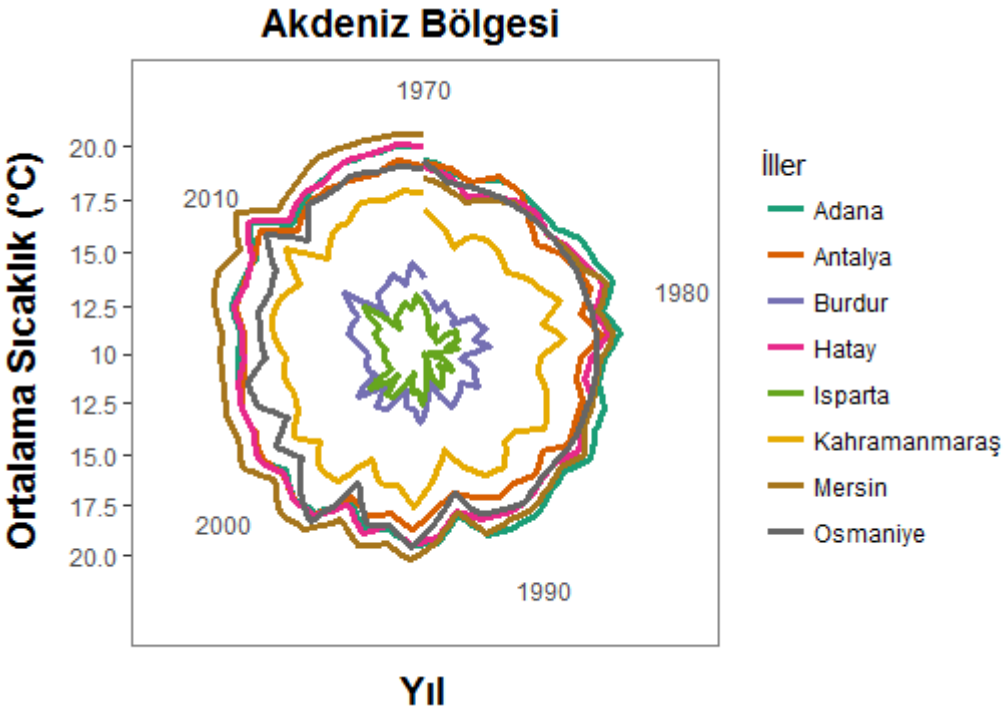
6. UYGULAMA

Bu bölümde Türkiye'deki illerin sıcaklık ve yağış değerlerine göre mekansal-zamansal kümelenmesi ST-DBSCAN kümeleme algoritması kullanılarak uygulanmıştır. Çalışmada 1970-2017 yılları aralığına ortalama sıcaklık (°C) ve toplam yağış miktarı ortalaması (mm) verileri kullanılarak analiz gerçekleştirilmiştir. Veriler Meteoroloji Genel Müdürlüğünden elde edilmiştir. Verilerin analizi için Matlab 13 programı kullanılmış olup sonuçların görselleştirme işlemleri de R Studio programı ile yapılmıştır. Coğrafik bölgelere göre illerin ortalama yıllık sıcaklıkları karşılaştırmalı olarak aşağıdaki şekillerde verilmiştir.

6.1 Veri ve Veri Ön Hazırlık İşlemleri

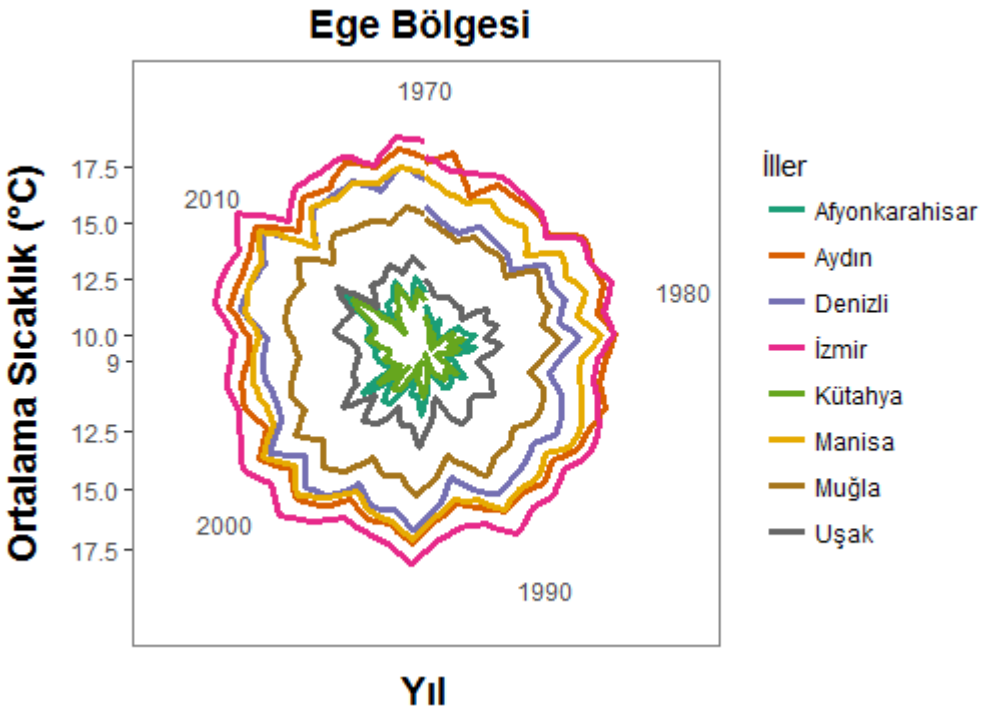
Tezin uygulama aşamasına geçilmeden önce veri talep etme ve veriyi analize hazırlama sürecinde birçok problemlerle karşılaşılmıştır. İlk olarak Meteoroloji Genel Müdürlüğü'nden tüm türkiye istasyonlar bazında aylık olarak 1930-2017 yılları arasında ortalama sıcaklık ve toplam yağış verileri talep edilmiştir. Gelen veriler doğrultusunda istasyonlar noktasal olarak analiz edilmek istenmiştir. Bazı istasyonların düzenli çalışmaması ve bazı istasyonlarında kuruluş tarihleri farklı olduğundan dolayı noktasal olarak istasyonları kümeleme işlemi gerçekleştirmek beklenen şekilde doğru sonuçlar vermemiştir. Bu nedenle herbir ilin en uzun zaman aralığına sahip olan ve en az kayıp gözlem bulunduran istasyonlar seçilip sürece bu veriler ile devam edilmiştir. Elde edilen verinin son hali 1970-2017 yılları arasında herbir ili temsil eden tek bir istasyondan oluşmakta ve aylık ortalama değerleri barındırmaktadır. Aylık olarak sıcaklık ortalamalarının çok değişkenlik göstemesi ve toplam yağış miktarlarınında sıcaklıktan daha fazla bir şekilde aydan aya değişkenlik göstermesi sebebiyle kümeleme sonuçları başarılı sonuçlar vermemiştir ve bir istasyonun gözlemleri birçok farklı kümeye dahil edilmiştir. Bu değişkenlik problemini ortadan kaldırmak için herbir yıla ait 12 ayın ortalama değerleri kullanılmıştır. Verinin son yapısı, 1970-2017 yılları arasında yıllık ortalama sıcaklık ve bir yılda ay başına düşen ortalama toplam yağış miktarını içermektedir.

6.2 Türkiye Coğrafik Bölgelerinin Sıcaklık ve Yağış Seviyelerinin Zamana Göre Değişimleri



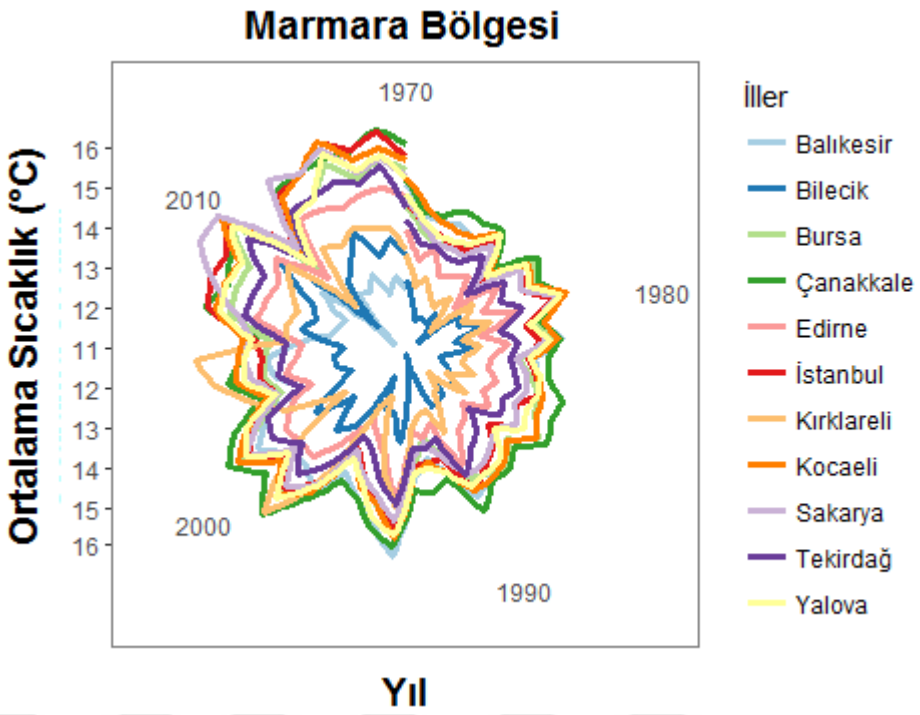
Şekil 6.6.1 Akdeniz Bölgesi İllerinin Ortalama Yıllık Sıcaklıkları.

Akdeniz bölgesi illerinin 48 yıllık ortalama sıcaklık aralığı 10 ile 20 derece gibi geniş bir aralıkta seyretmektedir. Bölgede diğer bir dikkat çeken özellik ise bu süre zarfında Isparta haricindeki tüm illerin ortalama sıcaklık değerlerinde 1 derecelik bir artış olmasıdır. Bölge illeri sıcaklık ortalamalarına göre büyükten küçüğe Mersin (19,43), Adana (19,25), Hatay (18,99), Antalya (18,60), Osmaniye (18,50), Kahramanmaraş (16,78), Isparta (12,04) ve Burdur (13,17) şeklinde sıralanmaktadır. Isparta ve Burdur illerinin sıcaklık ortalaması diğer bölge illerine kıyasla çok belirgin bir şekilde düşüktür. Bölgenin sıcaklık ortalaması 16,79 derecedir. Ayrıca 2010 yılında tüm illerin sıcaklık ortalamasında belirgin bir oranda yükselme olduğu göze çarpmaktadır.



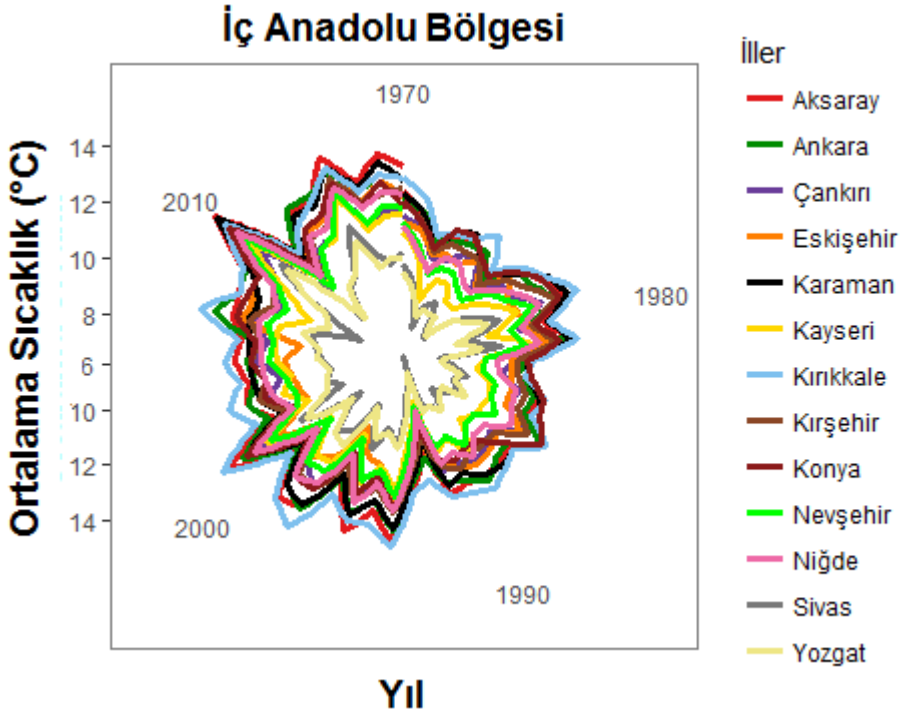
Şekil 6.6.2 Ege Bölgesi İllerinin Ortalama Yıllık Sıcaklıkları.

Ege bölgesi illerinin ortalama sıcaklık aralığı yaklaşık 9 derece ile 18 derece arasında değişmektedir. Bölgede 48 yıllık süre zarfında İzmir, Denizli ve Uşak illeri başta olmak üzere hemen hemen tüm illerde yaklaşık 1 derece sıcaklık artışı yaşanmıştır. Bölge illeri ortalama sıcaklıklarına göre büyükten küçüğe doğru İzmir (17,91), Aydın (17,39), Manisa (16,79), Denizli (16,19), Muğla (15,07), Kütahya (10,73), Afyon (11,22) ve Uşak'tır (12,52) şeklinde sıralanmaktadır. Bölgenin ortalama sıcaklık değeri 14,18 olarak kaydedilmiştir. 2010 yılında tüm türkiyede olduğu gibi Ege bölgesinde de genel olarak sıcaklık artışı meydana gelmiştir. Bölgede Uşak, Kütahya ve Afyon illeri diğer illere göre çok daha düşük sıcaklık değerlerine sahiptir.



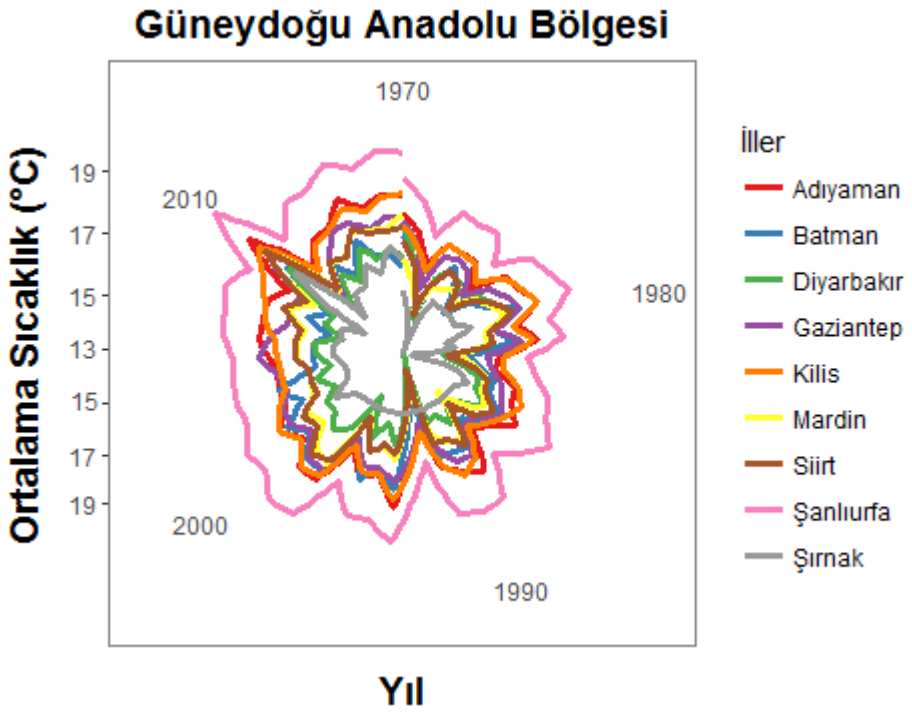
Şekil 6.6.3 Marmara Bölgesi İllerinin Ortalama Yıllık Sıcaklıkları.

Marmara bölgesi illerinin ortalama sıcaklık aralığın 11 ile 16 derece arasında izlediği görülmektedir. Bölgede 2010 yılından sonra tüm illerde ani bir sıcaklık düşüşü gözlenmiştir. Bölge illerinin büyükten küçüğe ortalama sıcaklık sıralaması Çanakkale (15,10), Koceeli (14,87), Yalova (14,69), İstanbul (14,67), Sakarya (14,59), Bursa (14,57), Tekirdağ (14,07), Balıkesir (14,03), Edirne (13,67), Bilecik (12,51), Kırklareli (13,28) ve Edirne (13,67) şeklindedir. Diğer bölgelerle karşılaştırıldığında Marmara bölgesi illerinin ortalama sıcaklıkları arasında çok büyük boyutta bir farklılık bulunmamaktadır (bölge içi benzerlik yüksektir). Fakat 48 yıllık süre zarfında bu bölge illerinin ortalama sıcaklıklarındaki ısınma diğer bölgelere göre dikkat çekici seviyede yüksektir. Bunun nedeni olarak sanayi bölgesi olması gösterilebilir.



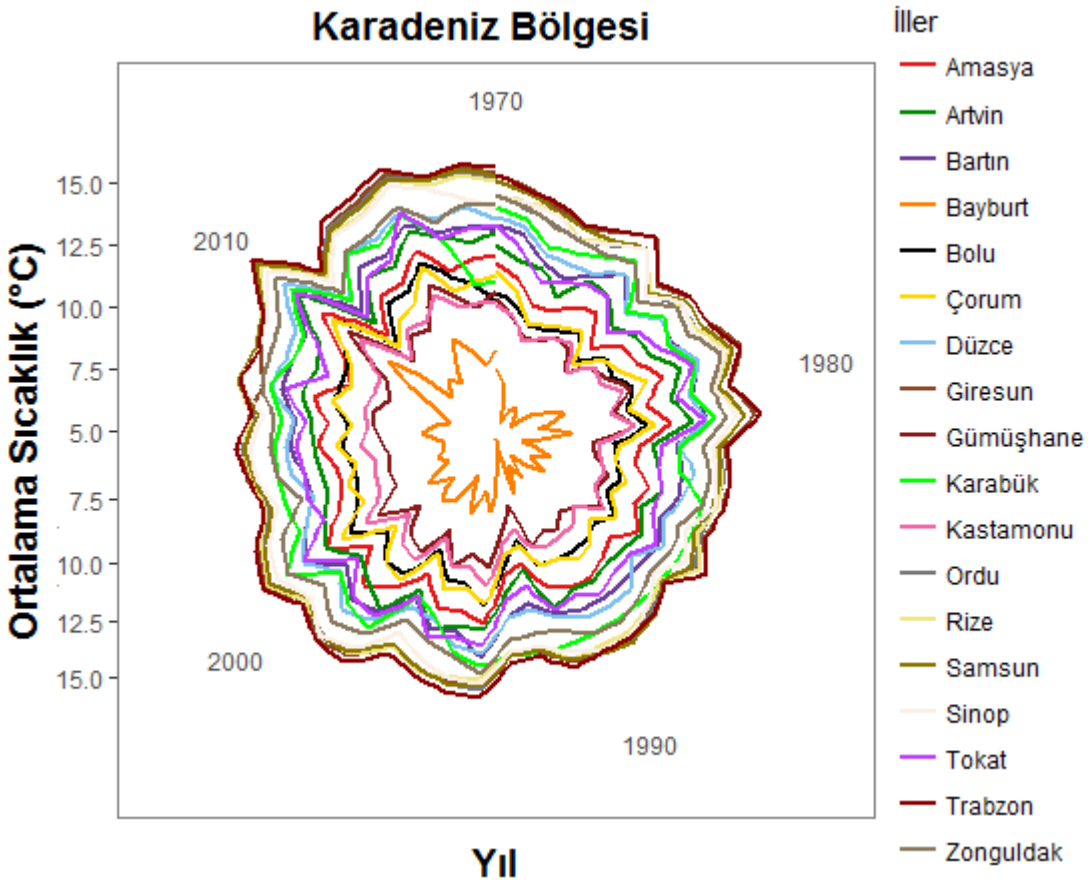
Şekil 6.6.4 İç Anadolu Bölgesi İllerinin Ortalama Yıllık Sıcaklıkları.

İç Anadolu bölgesi illerinin ortalama sıcaklıkları yaklaşık olarak en düşük 6 derece ile en yüksek 14 derece arasında seyretmekte olup bölge ortalaması 11,10 derecedir. Bölge illerinde hemen hemen tüm diğer bölgelerde gözlendiği gibi 48 yıllık zaman zarfı içerisinde sıcaklık artışı meydana gelmiştir. Yine tüm bölgeler ile benzer olarak 2010 yılında yaklaşık 2 derecelik bir artış gözlenmiştir. Bölgede iller arası ortalama sıcaklık farkı çok fazla değildir. Diğer illerden kendini belirgin bir şekilde ayırmış iller düşük sıcaklık derecelerine sahip olan Sivas ve Yozgat'tır. Bölge illeri sıcaklıklarına göre Krıkkale (12,38), Aksaray (12,16), Ankara (11,98), Karaman (11,97), Konya (11,61), Kırşehir (11,45), Çankırı (11,15), Niğde (11,13), Eskişehir (11,11), Nevşehir (10,67), Kayseri (10,42), Yozgat (9,15) ve Sivas (9,12) şeklinde sıralanmaktadır.



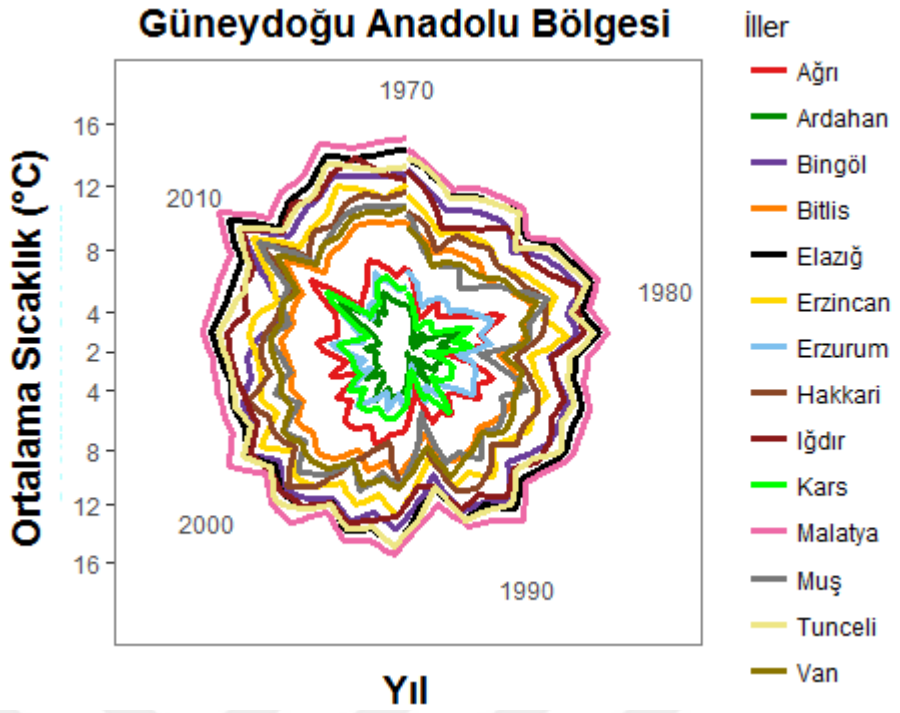
Şekil 6.6.5 Güneydoğu Anadolu Bölgesi İllerinin Ortalama Yıllık Sıcaklıkları.

Günedoğu Anadolu bölgesinde iller yaklaşık olarak 13 derece ile 19 derece arasında seyretmektedir. Güneydoğu Anadolu bölgesi illeri 48 yıllık süre zarfında ısınma seviyesi en net olarak gözüken bölgelerin başında gelmektedir. Özellikle bölgenin en yüksek sıcaklık değerine sahip olan Şanlıurfa (18,54) ilinde ısınma çok daha dikkat çekicidir. Isınma seviyesi neredeyse 2 dereceye tekabül etmektedir. Bölge illerinin sıcaklık dereceleri sırası ile Şanlıurfa (18,54), Adıyaman (17,29), Kilis (17,23), Gaziantep (16,84), Batman (16,31), Siirt (16,25), Mardin (16,18), Şırnak (15,06) ve Diyarbakır (15,75) şeklindedir. Türkiye geneli 2010 yılı sıcaklık artışı bu bölgeyide etkilemiştir. Ayrıca 1992 yılında bölgede yaklaşık olarak 2 derecelik bir soğuma dikkat çekmektedir. Bölgenin ortalama sıcaklığı 16,61 derece olmakla birlikte Türkiye bölgelerinin Akdeniz'den sonra en sıcak 2. bölgesidir.



Şekil 6.6.6 Karadeniz Bölgesi İllerinin Ortalama Yıllık Sıcaklıkları.

Karadeniz bölgesi illerinin ortalama sıcaklık değerleri yaklaşık olarak 5 derece ile 15 derece arasında değişmektedir. Bölge sıcaklık ortalaması 12,38 derecedir. Şekilde de görüldüğü üzere bölgedeki illerden belirgin bir şekilde ayrılmış olan il Bayburt'tur ve bölgenin en soğuk ilidir. Diğer bölgelere nazaran Karadeniz bölgesinde 48 yıllık zaman aralığında sıcaklık artışı daha az seviyelerdedir. Bölge illeri sıcaklık derecelerine göre Trabzon (14,82), Giresun (14,65), Samsun (14,52), Ordu (14,47), Rize (14,44), Sinop (14,22), Zonguldak (13,69), Karabük (13,22), Düzce (13,02), Bartın (12,76), Tokat (12,39), Artvin (12,05), Amasya (11,36), Çorum (10,54), Bolu (10,47), Kastamonu (9,69), Gümüşhane (9,48) ve Bayburt (6,93) şeklindedir.



Şekil 6.6.7 Güneydoğu Anadolu Bölgesi İllerinin Ortalama Yıllık Sıcaklıkları (1970-2017).

Güneydoğu Anadolu bölgesi ortalama sıcaklığı 9,48 derece ille Türkiye'nin en soğuk bölgesidir. Bölge illerinin sıcaklıkları yaklaşık olarak 2 derece ile 16 derece arasında değişmektedir. Şekilde de görüldüğü üzere Erzurum, Kars, Ardağan ve Ağrı illeri bölgenin en soğuk illeri olarak diğer illerden ayrılmaktadır. Bölgede 48 yıllık süre zarfında yaklaşık olarak 1 derecelik bir ısınma gözlenmiştir. Bölge illeri sıcaklıklarına göre Malatya (13,77), Elazığ (13,09), Tunceli (12,80), Iğdır (12,17), Bingöl (11,99), Erzincan (10,87), Hakkari (10,26), Van (9,45), Muş (9,35), Bitlis (9,02), Ağrı (6,14), Erzurum (5,35), Kars (4,85) ve Ardağan (3,70) şeklinde sıralanmaktadır.

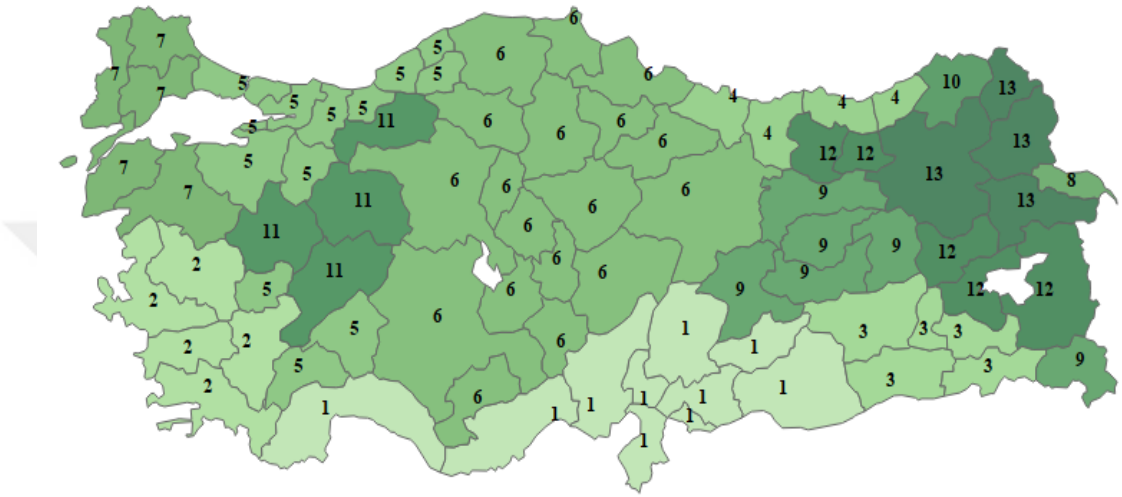
Benzer görselleştirme tekniği bölgelerin aldıkları yağış değerleri içinde yapılmıştır ve yapılan bu grafikler Ek B'de gösterilmiştir.

6.3 Kümeleme Analizi

Bu çalışmada uygulamada kullanılan minimum nokta sayısı olan MinPts, mekansal nitelikler olan enlem boylam için uzaklık parametresi olan Eps_1 ve mekansal olmayan değişkenler için uzaklık parametresi olan Eps_2 parametreleri algoritma sonuçları üzerindeki etkisinin gösterilmesi amacıyla değiştirilerek analiz edilmiştir.

Aşağıdaki şekilde illere göre yıllık ortalama sıcaklık derecelerinin kümelenmesini göstermektedir. Bu uygulamada kullanılan parametre değerleri de aşağıda belirtilmiştir.

Eps_1 = 1,1
Eps2 = 10
MinPts = 5
Δ_e= 2



Şekil 6.6.8 Ortalama Sıcaklık Derecelerine Göre İllerin Kümelenme Sonuçları I (Eps_1 =1,1, Eps2 = 10, MinPts = 5, Δ_e= 2).

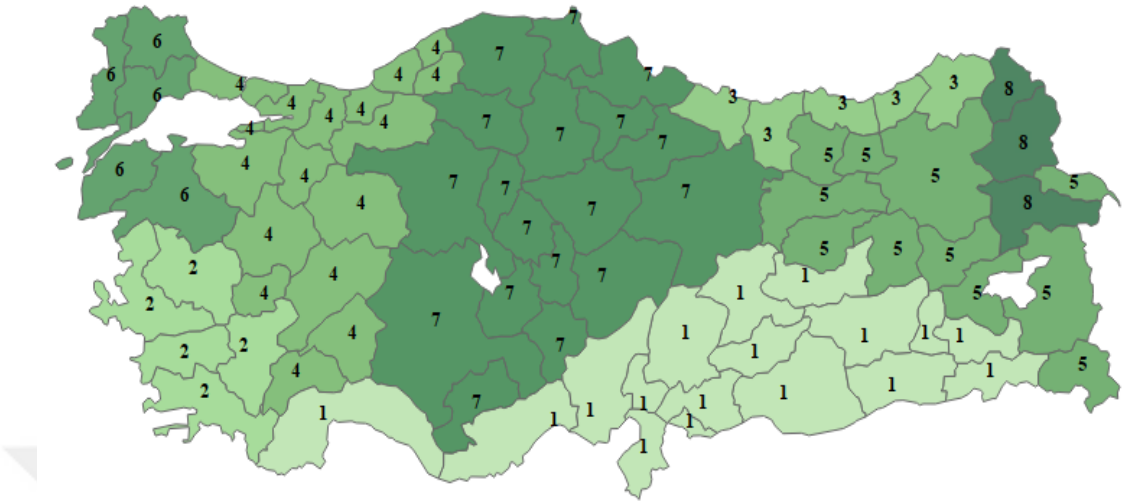
Kümeleme sonuçlarına göre Marmara bölgesinin batısında Tekirdağ, Kırklareli, Edirne, Balıkesir ve Çanakkale illeri ortalama sıcaklığı 14,03 olan 7 numaralı kümede toplanmıştır. Marmara bölgesinin doğusunda ise İstanbul, Kocaeli, Sakarya, Yalova, Bursa, Düzce, Zonguldak, Bartın, Bilecik, Isparta, Burdur, Uşak ve Karabük illeri 7 numaralı küme ile benzer ortalama sıcaklığa sahip olmasına rağmen meknsal yakınlığı sağlayamadıklarından dolayı ortalaması 13,56 olan 5 numaralı kümede toplanmışlardır. Kütahya, Eskişehir, Afyonkarahisar ve Bolu illeri ise ortalaması 10,88 olan 11 numaralı kümede toplanmışlardır. Ege bölgesinde İzmir, Manisa, Denizli, Aydın ve Muğla illeri ortalaması 16,67 olan 2 numaralı kümede toplanmıştır. İç Anadolu bölgesinin tamamı ve orta Karadeniz bölgesi ile Akdeniz bölgesinin iç kısımlarından oluşan iller ortalaması 11,44 olan 6 numaralı kümede toplanmıştır. Akdeniz bölgesinin kıyı kesimleri ile Güneydoğu Anadolu bölgesinin doğusunda bulunan iller Antalya, Mersin, Adana, Osmaniye, Hatay, Kahramanmaraş, Gaziantep, Kilis, Şanlıurfa ve Adıyaman ortalaması 18,03 ile en

yüksek sıcaklığa sahip 1 numaralı kümede toplanmıştır. Malatya, Tunceli, Elazığ, Erzincan, Hakkari ve Bingöl ilerlide ortalaması 12,13 olan 9 numaralı kümede toplanmıştır. Batman, Diyarbakır, Mardin Siirt ve Şırnak illeri ortalaması 15,91 olan 3 numaralı kümeye atanmıştır. Ağrı, Ardağan, Erzurum ve Kars ortalama sıcaklığı 5,01 olan ve en soğuk sıcaklığa sahip olan 13 numaralı kümede toplanmıştır. Gümüşhane ve Bayburt illeri, etrafındaki kümelerle mekansal olarak yakınlık koşulunu sağlamış olsa da mekansal olmayan (sıcaklık ve zaman) uzaklık koşullarını sağlamadığından dolayı ortalaması 8,85 olan 12 numaralı kümede toplanmıştır. Benzer sebepten dolayı Artvin ili de tek başına 10 numaralı kümeyi oluşturmuştur. Aşağıda analiz sonucunda oluşan kümelerin ortalama sıcaklık değerleri verilmiştir.

Çizelge 6.3 Kümelerin Ortalama Sıcaklık Değerleri I (Eps_1 =1,1, Eps2 = 10, MinPts = 5, Δ_e= 2).

Kümeler	**Ortalama Sıcaklık (°C)**
1	18.03
2	16,67
3	15,91
4	14,60
5	13,56
6	11,44
7	14,03
8	12,17
9	12,13
10	12,05
11	10,88
12	8,85
13	5,01

Eps_1 = 1,1
Eps2 = 10
MinPts = 5
Δ_e= 4



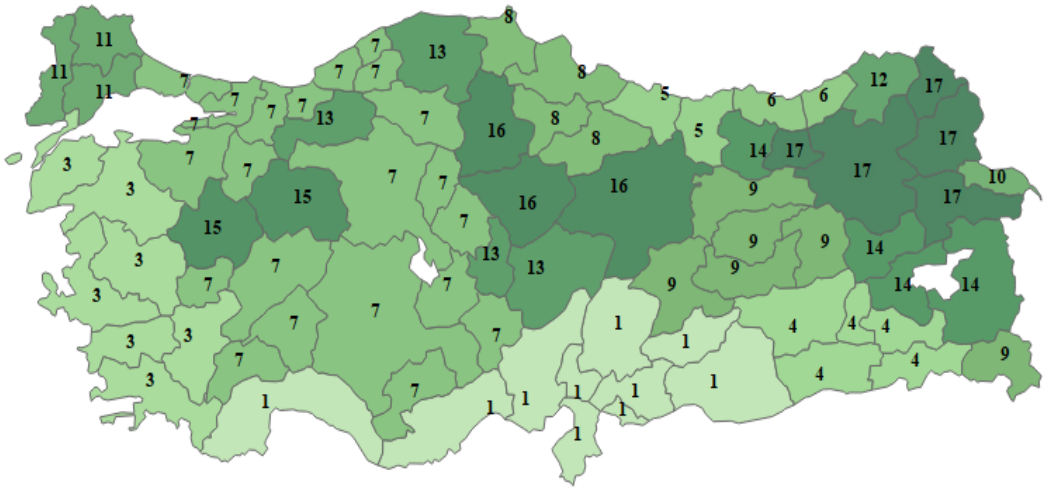
Şekil 6.6.9 Ortalama Sıcaklık Derecelerine Göre İllerin Kümelenme Sonuçları II (Eps_1 =1,1, Eps2 = 10, MinPts = 5, Δ_e= 4).

İllere göre sıcaklık ortalamalarının kümeleme işleminde Δ_e parametresinin büyütülmesi ile elde edilen sonuçlar yukarıda gösterilmektedir. Şekilden de anlaşılacağı üzere Δ_e parametresinin büyütülmesi daha çok gözlemli daha geniş kümelerin elde edilmesine neden olmaktadır. Bunun nedeni gözlemlerin kümeye dahil edilme koşulunun yumuşatılmasıdır. Farklılıkları karşılaştırmak amacıyla yapılan incelemelerde; önceki kümeleme sonucunda 1 ve 11 numaralı kümeler ortalaması 12,93 olan 4 numaralı kümede toplanmıştır. Bunun haricinde dikkat çeken diğer bir küme birleşimi ise 1 numaralı kümenin tamamı, 9 ve 3 numaralı kümeden de bazı iller alınarak ortalaması 16,79 olan 1 numaralı kümeye dahil edilmesidir. Önceki sonuçlarda tek başına küme oluşturan Artvin burada ortalaması 14,09 olan ve Trabzon, Rize, Giresun ve Ordu ilinden oluşan 3 numaralı kümeye dahil edilmiştir. Benzer olarak aynı kümede olan Gümüşhane ve Bayburt illeri bu sonuçlara göre ortalaması 9,79 olan 5 numaralı kümeye dahil edilmiştir.

Çizelge 6.1 Kümelerin Ortalama Sıcaklık Değerleri II (Eps_1 =1,1, Eps2 = 10, MinPts = 5, Δ_e= 4).

Kümeler	Ortalama Sıcaklık (°C)
1	16,79
2	16,67
3	14,09
4	12,93
5	9,79
6	14,03
7	11,44
8	4,89

Eps_1 = 1,2
Eps2 = 5
MinPts = 5
Δ_e= 2



Şekil 6.6.10 Ortalama Sıcaklık Derecelerine Göre İllerin Kümelenme Sonuçları III (Eps_1 =1,2, Eps2 = 5, MinPts = 5, Δ_e= 2).

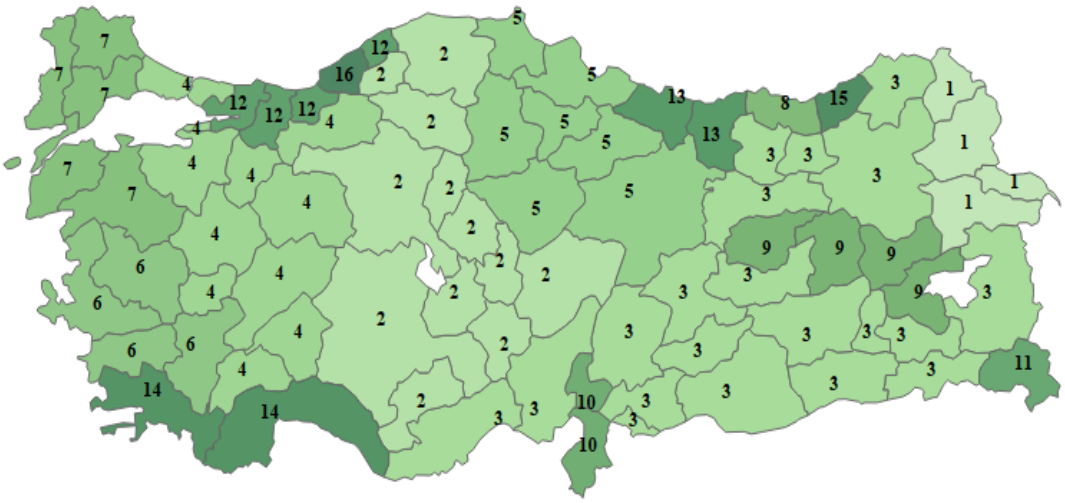
Mekansal maksimum uzaklık parametresi olan Eps_1'in genişletilmesi ve mekansal olmayan uzaklık parametresi Eps_2 değerinin azaltılması ile elde edilen kümeleme sonuçları yukarıdaki şekilde gösterilmektedir. Eps_1 parametresinin genişlemesi kümelerin yayılma genişliğini de arttırırken Eps_2 parametresinin küçültülmesi bu geniş kümelerin bütünlüğünü azaltmıştır.

Çizelge 6.2 Kümelerin Ortalama Sıcaklık Değerleri III (Eps_1 =1,2, Eps2 = 5, MinPts = 5, Δ_e= 2).

Kümeler	Ortalama Sıcaklık (°C)
1	18.03
3	16,07
4	15,91
5	14,57
6	14,63
7	12,79
8	13,12
9	12,13
10	12,17
11	13,67
12	12,05
13	10,31
14	9,33
15	10,92
16	9,60
17	5,39

Türkiye geneli illerin sıcaklık ortalaması kümelenmesinden sonra aşağıda aylık toplam yağış miktarı ortalamasının kümelenme sonuçları verilmiştir. Sıcaklık dereceleri kümeleme analizinde olduğu gibi parametre değerleri değiştirilerek farklı sonuçlar elde edilip yorumlanmıştır.

Eps_1 = 1,1
Eps2 = 20
MinPts = 10
Δ_e= 20



Şekil 6.6.11 Yağış Alma Seviyelerine Göre İllerin Kümelenme Sonuçları I (Eps_1 =1,1, Eps2 = 20, MinPts = 10, Δ_e= 20).

Yukarıdaki parametre değerleri ile yapılan mekansal-zamansal kümeleme analizi sonuçlarına göre illerin aylık toplam yağış miktarı ortalamaları 16 kümeye

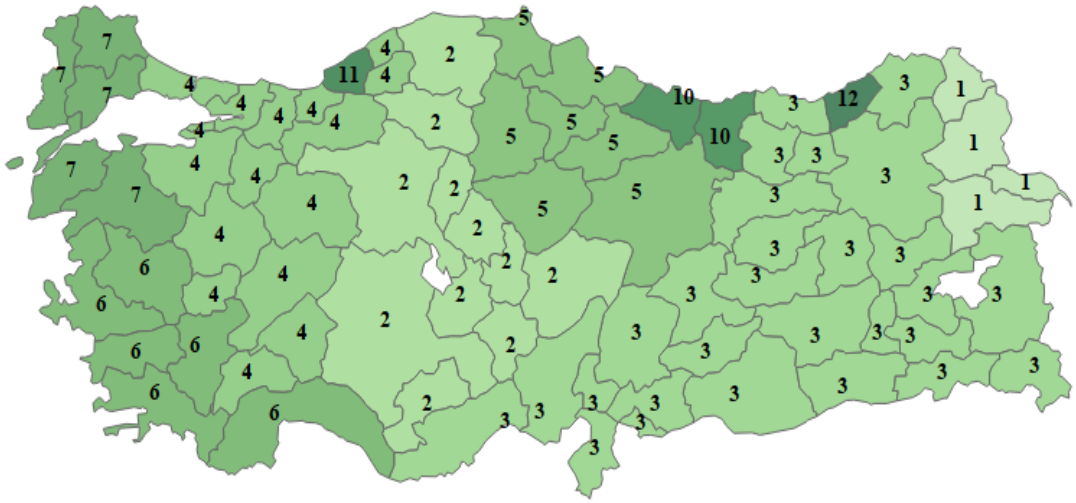
ayrılmıştır. Burada kullanılan Δ_e parametresi, oluşturulmuş bir kümeye dahil olacak yeni bir gözlemin yağış değeri ile küme içindeki gözlemlerin yağış ortalama değeri arasındaki farkın maksimum 20 birim olması gerektiğini ifade etmektedir. Elde edilen kümeleme sonuçları incelendiğinde Marmara bölgesinde Tekirdağ, Edirne, Kırklareli, Çanakkale ve Balıkesir illeri ortalaması 51,42 olan 7 numaralı kümede toplanmıştır. İstanbul ve Yalova ise ortalaması 45,71 olan ve Afyonkarahisar, Bilecik, Bolu, Burdur, Bursa, Kütahya, Isparta, Eskişehir ve Uşak illerini içeren 4 numaralı kümeye dahil edilmiştir. Bartın, Düzce Sakarya ve Kocaeli illeri çevre illere kıyasla daha fazla yağış aldığından dolayı ortalaması 79,72 olan 12 numaralı kümede toplanmıştır. İç Anadolu bölgesinin büyük bir kısmı (Aksaray, Ankara, Çankırı, Karaman, Kayseri, Kırıkkale, Konya, Nevşehir ve Niğde) ve orta Karadeniz bölgesi illerinden Kastamonu ve Karabük ortalaması 33,11 olan 2 numaralı kümede toplanmıştır. Bu illere kıyasla daha fazla yağış alan Samsun, Çorum, Amasya, Sinop, Sivas, Tokat ve Yozgat illeri ortalaması 45,06 olan 5 numaralı kümede toplanmıştır. Ege bölgesinin kıyı kesiminde İzmir, Manisa, Aydın ve Denizli illeri ortalaması 55,35 olan 6 numaralı kümede toplanmıştır. Muğla bölgesinin en fazla yağış alan ili olduğundan bu kümeye dahil edilmemiş kendisi gibi yüksek yağış seviyesine sahip olan Antalya ile ortalaması 92,20 olan 14 numaralı kümeye dahil edilmiştir. Akdeniz bölgesinin kıyı kesiminde Hatay ve Osmaniye çevresindeki illere göre daha fazla yağış aldığından dolayı ortalaması 74,21 olan 10 numaralı kümede toplanmıştır. Bu illerin Çevresindeki iller daha düşük ortalamaya sahip olan (44,35) 3 numaralı kümeye dahil edilmiştir. Ortalaması 44,35 olan 3 numaralı küme, Güney ve Doğu Anadolu bölgesinin büyük bir kısmını ve doğu Karadeniz bölgesinin iç kısmındaki bazı ilerlide kapsayarak büyük bir küme yapısı sergilemektedir. 3 numaralı kümeye göre ortalama yağış alımı çok daha fazla olan (68,69) 9 numaralı küme Bingöl, Bitlis, Muş ve Tunceli illerinden meydana gelmektedir. Doğu Karadeniz bölgesinde aylık toplam yağış miktarı ortalaması 188 olan Rize ili Türkiyenin en çok yağış alan ili olarak kendi başına bir küme oluşturmuştur. Rizenin yanında bulunan Trabzon ilide tek başına bir küme oluşturmuştur. Bunun nedeni batısında Giresun ve Ordu illerinden oluşan ve ortalaması 96,66 olan 13 numaralı kümeden daha düşük seviyede yağış almasıdır (ortalama=69,34). Türkiyenin kuzey doğusundaki Ağrı, Ardağan, Iğdır ve Kars illeri ortalaması 36,57 olan 1 numaralı kümede toplanmaktadır.

Çizelge 6.4 Kümelerin Ortalama Yağış Miktarları I (Eps_1 =1,1, Eps2 = 20, MinPts = 10, Δ_e= 20).

Kümeler	Toplam Yağış Miktarı Ortalaması (mm)
1	36,57
2	33,11
3	44,35
4	45,71
5	45,06
6	55,35
7	51,42
8	69,34
9	68,69
10	74,21
11	63,95
12	79,72
13	96,66
14	92,20
15	188,00
16	101,76

Mekansal zamansal kümeleme analizi kümeye dahil olma eşiği olan Δ_e parametresinin büyümesinin analiz sonuçlarına olan etkisi aşağıda gösterilmektedir.

Eps_1 = 1,1
Eps2 = 20
MinPts = 10
Δ_e= 40



Şekil 6.6.12 Yağış Alma Seviyelerine Göre İllerin Kümelenme Sonuçları II (Eps_1 = 1,1, Eps2 = 20, MinPts = 10, Δ_e= 40).

Bu değişiklik ile yapılan mekansal-zamansal kümeleme analizi sonucuna göre bir önceki analizde 16 olan küme sayısı 12'ye düşmüştür. Sonuçlar incelendiğinde kümeye dahil edilme eşiği olan Δ_e 20'den 40'a çıkarılmasına rağmen 1, 2 ve 5 numaralı kümeler çevresindeki kümelere dahil edilmemeye devam etmektedir.

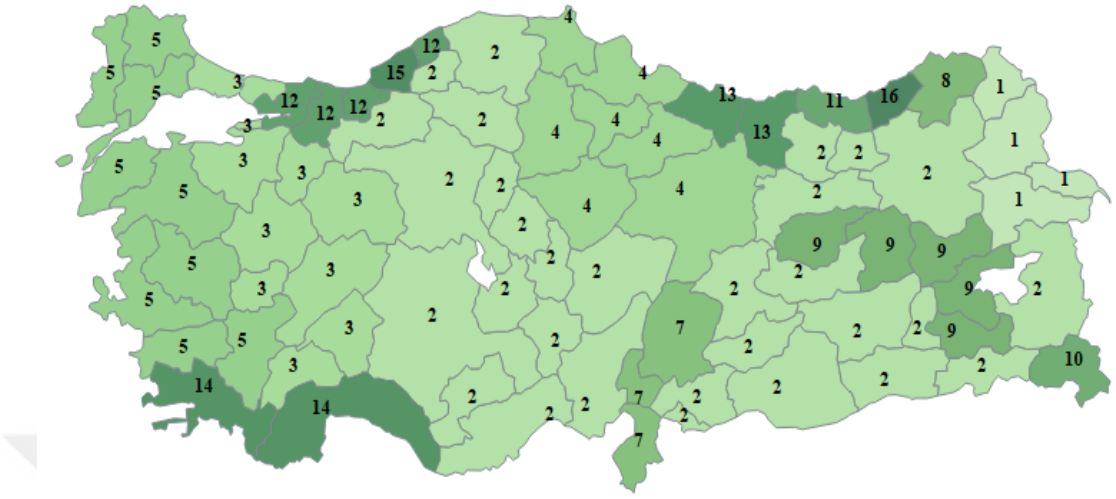
Bunun nedeni maksimum mekansal uzaklık parametresi olan Eps_1 koşulunun sağlanmamasıdır. Bir önceki analizde bulunan ve Sakarya, Kocaeli, Düzce ve Bartın illeri 4 numaralı küme illeri ile aynı kümeye dahil edilmiştir. Burada her ne kadar eşik değeri $\Delta_e = 40$ olarak seçilmiş olsa da ortalama yağışı 101,75 olan Zonguldak ili 4 numaralı kümeye dahil olmayarak tek başına 11 numaralı kümeyi oluşturmuştur. Ordu ve Giresun illeri 10 numaralı kümede toplanmış ve yüksek yağışından dolayı çevresindeki kümelere dahil olmamıştır. Aynı şekilde Türkiye'nin en çok yağış alan ili Rize tek başına küme olmaya devam etmektedir. Bir önceki analizde Ege bölgesi kıyısında İzmir, Aydın, Denizli ve Manisa'dan oluşan kümeye Muğla ve Antalya illeri dahil edilmiştir. Türkiyenin doğusu, Akdeniz bölgesinin doğusu, Rize hariç Karadeniz bölgesinin doğusu değişen parametre ile bütün olarak 3 numaralı kümede toplanmıştır. Ardağan, Kars, Iğdır ve Ağrı illerinden oluşan 1 numaralı küme aynı şekilde korunmaktadır. Bunun nedeni ise analizde illerin sadece bir noktasından alınan enlem ve boylam değerleri kullanılmaktadır ve bu değerler arasındaki uzaklık, maksimum mekansal uzaklık parametresi olan Eps_1 değerinden daha büyüktür. Önceki analizlerde mekansal uzaklık parametresi olan Eps_1 değerinin analiz sonucuna olan etkisinin anlaşılması için değer 1,1'den 1,2 değerine yükseltilmiş ve sonuçları aşağıda gösterilmiştir.

Çizelge 6.3.3 Kümelerin Ortalama Yağış Miktarları II (Eps_1 =1,1, Eps2 = 20, MinPts = 10, Δ_e= 40).

Kümeler	Toplam Yağış Miktarı Ortalaması (mm)
1	38,33
2	32,30
3	51,42
4	53,98
5	45,06
6	67,64
7	51,42
10	96,66
11	101,76
12	188,00

Aşağıda mekansal parametre Eps_1 değerinin büyütülmesinin analiz sonuçlarına etkileri aşağıda gösterilmiştir.

Eps_1 = 1,2
Eps2 = 20
MinPts = 10
Δ_e= 20



Şekil 6.6.13 Yağış Alma Seviyelerine Göre İllerin Kümelenme Sonuçları III (Eps_1 = 1,2, Eps2 = 20, MinPts = 10, Δ_e= 20).

Mekansal yakınlık parametre koşulunun genişletilmesinden sonra yapılan analiz sonuçları (Eps_1=1,2) ile ilk analiz sonuçları (Eps_1=1,1) karşılaştırıldığında ilk göze çarpan farklılık Marmara bölgesinin batısındaki Tekirdağ, Kırklareli, Edirne Balıkesir ve Çanakkale illerinin Ege bölgesinin batısındaki İzmir, Aydın, Manisa ve Denizli illerinde oluşan kümeyle birleşmiş olmasıdır. Bunun nedeni mekansal yakınlık koşulunun (Eps_1) bir nebze olsa da genişletilmesidir. Bir diğer farklılık ise önceki analizde İç Anadolu ve orta Karadeniz bölgesini kaplayan küme ile Güneydoğu ve Doğu Anadolu bölgelerini ayrı kümeler tarafından temsil edilmektedir. Fakat Burada bu iki küme tek bir kümede toplanmıştır. Önceki analiz ile benzer şekilde yine ortalama yağış miktarı 66,23 olan Bingöl, Bitlis, Muş, Siirt ve Tunceli illerinden oluşan 9 numaralı küme ortalaması 37,93 olan 2 numaralı bu kümenin arasında farklı bir küme oluşturmuştur. Eps_1 parametresinin 1,2'ye yükseltilmesi Kars, Iğdır, Van ve Ardağan illerinden oluşan 1 numaralı kümenin ortalamaları benzer seviyelerde olmasına rağmen 2 numaralı kümeye dahil edilmesine yetmemiştir. Akdeniz bölgesinde Hatay ve Osmaniye illerinin oluşturduğu kümeye mekansal uzaklık koşulu genişledikten sonra Kahramanmaraş ili dahil edilmiştir. 4 numaralı küme ile 2 numaralı kümenin ortalama yağış miktarları aynı kümeye dahil edilecek kadar yakın olsalar da mekansal uzaklık koşulunun 1,2 değeri olması iki kümenin birleşmesi için yeterli olmamıştır. Artvin ili mekansal

olmayan parametre olan Eps_2 değeri aynı kaldığından ve çevre kümelerle bu koşulu sağlayamadığından yine tek başına 8 numaralı kümeyi oluşturmuştur. Bu durum 11, 12, 13, 15 ve 16 numaralı kümeler içinde geçerlidir. Aşağıdaki tabloda Analiz sonucunda elde edilen kümelerin ortalama yağış miktarları verilmiştir.

Çizelge 6.6 Kümelerin Ortalama Yağış Miktarları III (Eps_1 =1,2, Eps2 = 20, MinPts = 10, Δ_e= 20).

Küme	Toplam Yağış Miktarı Ortalaması (mm)
1	38,33
2	37,93
3	45,60
4	45,06
5	53,17
7	69,47
8	59,58
9	66,23
10	63,95
11	69,34
12	79,72
13	96,66
14	92,20
15	101,76
16	188,00

7. SONUÇ

Bu çalışmada 1970-2017 yılları arasında ortalama sıcaklık (°C) ve toplam yağış miktarı ortalaması (mm) verileri kullanılarak analiz gerçekleştirilmiştir. Türkiye'deki illerin sıcaklık ve yağış değerlerine göre mekansal-zamansal kümelenmesi ST-DBSCAN kümeleme algoritması kullanılarak uygulanmıştır. Yapılan analizlerde çeşitli parametrelerin algoritmaların performansları üzerindeki etkileri incelenmiştir. Bu sebeple analizler farklı parametre değerleri kullanılarak yapılmış ve sonuçlar karşılaştırılmıştır.

Sıcaklık değerlerine göre yapılan analizlerde ilk olarak **Eps_1**=1,1 – **Eps2**=10 – **MinPts**=5 - **Δ_e**=2 parametreleri kullanılmış ve elde edilen sonuçlara göre iller 13 kümeye ayrılmıştır. İkinci analizde kümeye dahil olma eşik değeri olan Δ_e parametresi 2'den 4'e yükseltilerek bu koşul hafifletilmiş ve daha geniş ve bütünlüğü sağlanmış kümeler elde edilmiştir. Bu değişen parametre sonucunda 13 olan küme sayısı 8'e gerilemiştir. Δ_e parametre değerini yükseltmek her ne kadar küme bütünlüğünü arttırıp küme sayısının azalmasına sebep olsada küme içindeki gözlemlerin benzerliğinide azaltmıştır. Son olarak **Eps_1**=1,2 – **Eps2**=5 – **MinPts**=5 - **Δ_e**= 2 parametreleri kullanılarak yapılan analizde iller sıcaklıklarına göre 17 kümeye ayrılmıştır. 1,1'den 1,2'ye yükselen mekansal uzaklık parametresi Eps_1 kümelerin biraz genişlemesine neden olsa da 10'dan 5'e düşürülen mekansal olmayan değişkenlerin uzaklık parametresi Eps_2 küme içi benzerlik derecesini arttırdığından dolayı küme bütünlüğü yerine parça parça kümeler oluşmasına neden olmuştur.

1970-2017 yılları arasında herbir yılın aylık ortalama toplam yağış miktarlarına göre yapılan analizlerde ise ilk olarak **Eps_1**=1,1 – **Eps2**=20 – **MinPts**=10 - **Δ_e**=20 parametreleri kullanılarak yapılan analizde iller 16 kümede toplanmıştır. Sıcaklık analizinde olduğu gibi ilk analizde 20 olan Δ_e parametresi 40'a yükseltilmiştir. Değişen Δ_e parametresi sonucunda 16 olan küme sayısı 12'ye gerilemiş, kümeler genişlemiş ve bütünleşmiştir fakat küme içi benzerlik katsayısı azalmıştır. Son olarak **Eps_1**=1,2 – **Eps2**=20 – **MinPts**=10 - **Δ_e**=20 parametreleri kullanılmış olup iller 16

kümeye ayrılmıştır. Yükselen Eps_1 değeri ile kümelerin gözlem sayısı ve genişliğide doğru orantılı olarak yükselmiştir.

ST-DBSCAN algoritması kümeye yeni katılacak olan her nesneyi daha önceden hesaplanan küme ortalamasıyla karşılaştırarak kümeye dahil edilme durumunu sınamaktadır. Birbirine bitişik olan kümeleri kolayca ve gürültü gözlemleri kolayca ayırt edebilmektedir. DBSCAN, farklı yoğunluklardaki kütleler mevcut olduğunda bazı gürültü noktalarını algılayamıyor iken, ST-DBSCAN algoritması her kümeye bir yoğunluk faktörü atayarak bu problemi ortadan kaldırmaktadır. Tüm bu sonuçlar eşliğinde ST-DBSCAN algoritmasının verideki mekansal ve zamansal bilgiyi birlikte kullanarak güzel ve anlamlı sonuçlar verdiği söylenebilir. ST-DBSCAN algoritmasının olumsuz yönleri ise aykırı değere karşı hassastır. Bunun yanı sıra bazı durumlarda mekansal olarak aynı istasyonda bulunan noktalar farklı kümelere atanabilmektedir. Doğru parametre seçimi anlamlı analiz sonuçları için kritik derecede önemlidir ve bazı durumlarda Eps_1 ve Eps_2 parametrelerini birbirine uyumlu seçmek zorlayıcı olabilmektedir. Verilerin giriş sırasına göre de sonuçlar değişkenlik göstermektedir yani gözlem sırasına karşı hassas bir algoritmadır.

KAYNAKLAR

Aggarwal, C.C. (2013). *Outlier Analysis*. Springer: Berlin, Germany.

Aggrawal, Rakesh, Johannes Gehrke, Dimitrios Gunopulos ve Prabhakar Raghavan. (1998). *Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications*, Proc. ACM SIGMOD Int. Conference On Management of Data Seattle. USA.

Akın, Y. K. (2008). *Veri Madenciliğinde Kümeleme Algoritmaları ve Kümeleme Analizi*. (Doktora Tezi). Sosyal Bilimler Enstitüsü. Marmara Üniversitesi. İstanbul.

Albayrak, S. ve Yılmaz, Ş.K. (2009). *Veri Madenciliği Karar Ağacı Algoritmaları Ve İMKB Verileri Üzerine Bir Uygulama*, Süleyman Demirel Üniversitesi, İktisadi ve İdari Bilimler Dergisi.

Alexander Hinneburg ve Daniel A. Keim. (1998). *An Efficient Approach to Clustering in Large Multimedia Databases with Noise*, Proc. 4th International Conference on Knowledge Discovery and Data Mining (KDD'98). New York.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *ACM Sigmod Record*, 49–60.

Anselın Luc. (1992).Spatial *Data Analysis with GIS: An Introduction to Application in the Social Sciences*, National Center for Geographic Information and Analysis, University of California.

Argüden, Y. ve ERŞAHİN, B. (2008). *Veri Madenciliği: Veriden Bilgiye, Masraftan Değere*, ARGe Danışmanlık Yayınları, İstanbul.

Atilgan, C. (2014). *Kümeleme algoritmalari ve paralelleştirilmeleri*. (Yüksek Lisans Tezi). Ege Üniversitesi.

Atluri, G., Karpatne, A., & Kumar, V. (2017). *Spatio-Temporal Data Mining: A Survey of Problems and Methods*.

Bailey, T.C. and Gatrell A.C. (1995). *Interactive Spatial Data Analysis*, New York: Wiley, Pearson Education.

Başbozkurt Hakan. (2015). *Mekansal Regresyon Metotları Kullanımı İle Toprağın Bazı Fiziksel Ve Kimyasal Özelliklerinin Analizi* (Yayımlanmış Doktora Tezi). Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya.

Birant, D., & Kut, A. (2007). *ST-DBSCAN: An algorithm for clustering spatial-temporal data*. Data and Knowledge Engineering.

Bittner Thomas. (2000). *Rough Sets in Spatio-temporal Data Mining*. *Temporal, Spatial, and Spatio-Temporal Data Mining First International Workshop (TSDM 2000).*

Buckless BP, Petry FE. (1994). *Genetic algorithms*. IEEE Computer Press, Los Alamitos.

Cevahir Fahrettin, *Bir Perakende Firmasına Ait Veriler Üzerinden Veri Madenciliği Uygulaması* (Yayımlanmış Yüksek Lisans Tezi), Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

Cheng T., Haworth J., Anbaroglu B., Tanaksaranond G., Wang J. (2014). *Spatiotemporal data mining. In Handbook of Regional Science*. Springer: Heidelberg, Germany.

Cheng, H. (2008). *Spatial and temporal data mining with applications to earth science data*. Michigan State University.

Cheng, T., Haworth, J., Anbaroglu, B., Tanaksaranond, G., & Wang, J. (2014). *Handbook of Regional Science. Journal of Regional Science* (C. 54). London: Springer Reference.

Cheng, W.; Wang, W. ve Batista, S. (2014). *Grid-based clustering. Data clustering: algorithms and applications.* NewYork: CRC.

Demiralay M. (2005). *Hiyerarşik Kümeleme Metodları İle Veri Madenciliği Uygulamaları* (Yayımlanmış Y.Lisans Tezi). Marmara Üniversitesi, Fen Bilimleri Enstitüsü. İstanbul.

Deng, M., Liu, Q., Cheng, T., Yan, S. (2011). *An adaptive spatial clustering algorithm based on Delaunay triangulation*. Computers, Environment and Urban Systems 35 (4).

Dönmez Zeynep Seçil. (2008). *Bayi Performans Değerlendirmesinde Bir Veri Madenciliği Uygulaması* (Yayımlanmış Yüksek Lisans Tezi), İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü İstanbul.

E.Schikuta. (1996). *Grid Clustering: An Efficient Hierarchical Clustering Method for Very Large Data Sets*, Proceedings of the 13th International Conference on Pattern Recognition,Vol. 2.

Ester, M., H.P. Kriegel, J. Sander. (1997). *Spatial data mining: A database approach Advances in spatial databases*, Springer-Verlag Berlin, Berlin.

Ester, M., Kriegel, H. P., Sader, J. ve Xu, X. (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise*. *International* Conference of Knowledge Discovery and Data Mining. Portland, USA.

Fischer, M. M., & Wang, J. (2011). *Spatial Data Analysis: Models, Methods and Techniques*. *Springer Briefs in Regional Sciences*.

G. Andrienko, N. Andrienko. (1999). *Data mining with C4.5 and interactive cartographic visualization: User interfaces to data intensive systems*, IEEE Computer Society, Los Alamitos.

Gatrell, a. C., & Bailey, T. C. (1996). *Interactive spatial data analysis in medical geography*. Social Science & Medicine.

Getıs Arthur. (2008). *A History of The Concept of Spatial Autocorrelation: A Geographer's Perspective*, Geographical Analysis.

Gholamhosein Sheikholeslami, Surojit Chatterjee ve Aidong Zhang. (1998). *Wavecluster: A Multi- Resolution Clustering Approach for Very Large Spatial Databases*, Proc. 24th International Conference on Very Large Databases, New York.

Guha, S., Rastogi, R., and Shim, K. (1998). *CURE: An efficient clustering algorithm for large data- bases*. SIGMOD.

Haınıng Robert P. (2003). *Spatial Data Analysis: Theory and Practice, Cambridge University Press*, New York.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. *San Francisco, CA, itd: Morgan Kaufmann*.

Hand, D.J. (1999). Introduction. In: Intelligent Data Analysis: an Introduction, Springer, Berlin, Heidelberg.

Harvey J. Miller, & Jiawei Han. (2009). *Geographic Data Mining and Knowledge Discovery* (2. baskı). Boca Raton: CRC Press.

Haykin, S. (2009). *Neural Networks and Learning Machines, 3rd edition. Chapter 9, Self-Organizing Maps*. Pearson International Edition. New Jersey.

Jain, A.K. & Mao, J. (1996). *Artificial Neural Networks: A tutorial*, IEEE V.

Jiang W, Baker ML, Ludtke SJ, Chiu W. (2001). *Bridging the information gap: computational tools for intermediate resolution structure interpretation*. J Mol Biol.

Jyoti Yadav, Dharmender Kumar. (2014). *Sub space Clustering using CLIQUE: An Exploratory Study*. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics). Eepe.Ethz.Ch*.

Kaufman, L., and Rousseeuw, P. J. (1987), *Clustering by Means of Medoids, Statistical Data Analysis Based on The L1–Norm and Related Methods*, John Wiley & Sons, Inc. New Jersey.

Kintigh, K. W. (1990) *Intrasite Spatial Analysis: A Commentary on Major Methods. In Mathematics and Information Science in Archaeology: A Flexible Framework*. Studies in Modern Archaeology 3. Holos. Bonn.

Kintigh, K. W. and A. J. Ammerman. (1982). *Heuristic Approaches to Spatial Analysis in Archaeology*. American Antiquity.

Kisilevich S., Mansmann F., Nanni M., Rinzivillo S. (2010). *Spatio-Temporal Clustering*. Springer: Berlin, Germany.

Kohonen, T. (1995). *Self-Organizing Maps*. Berlin: Springer.

Kolingerova, I., Zalik, B. (2006). *Reconstructing domain boundaries within a given set of points, using Delaunay triangulation*. Computers & Geosciences 32 (9).

Koperski K., Adhikary J., Han J. (1996). *Spatial data mining: Progress and challenges survey paper*. In Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge

Discovery, Montreal, QC, Canada.

Kökver Yunus. (2012). *Veri Madenciliğinin Nefroloji Alanına Uygulanması* (Yayımlanmış Y.Lisans Tezi). Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü. Kırıkkale.

Larose, D.T. (2005). *Discovering Knowledge in Data: An Introduction to Data Mining*, Wiley Publishing

Laurene Fausett. (1993). *Fundamentals of Neural Networks: Architectures, Algorithms And Applications.* Prentice Hall.

Lee ES. (2000). *Neuro-fuzzy estimation in spatial statistics*. J Math Anal Appl.

Li DR, Wang SL, Li DY. (2006). *Theory and application of spatial data mining*, Science Press. Beijing.

Li, DR., Wang, S., & Li, DY. (2010). *Spatial Data Mining*. *Data Mining and Knowledge Discovery Handbook*.

Liu, P., Zhou, D., & Wu, N. (2007). *VDBSCAN: Varied Density Based Spatial Clustering of Applications with Noise. Proceedings - ICSSSM'07: 2007 International Conference on Service Systems and Service Management*.

Liu, Q., Deng, M., Shi, Y., & Wang, J. (2012). *A density-based spatial clustering algorithm considering both spatial proximity and attribute similarity*. Computers and Geosciences.

Lu H, Setiono R, Liu H. (1996). *Effective data mining using neural networks*. IEEE Trans Knowl Data Eng.

Mennis, J., & Guo, D. (2009). *Spatial data mining and geographic knowledge discovery-An introduction*. Computers, Environment and Urban Systems.

Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel ve Jörg Sander. (1999). *OPTICS: Ordering Points To Identify The Clustering Structure*, ACM SIGMOD International Conference on Management of Data, Philadelphia.

Miller H.J., Han J. (2009). *Geographic Data Mining and Knowledge Discovery*; CRC Press: Sacramento.

N. Cressie, Christopher K. Wikle. (2015). *Statistics for spatio-temporal data*. John Wiley & Sons.

Ng, R. T., & Han, J. (2002). *CLARANS: A method for clustering objects for spatial data mining*. IEEE Transactions on Knowledge and Data Engineering.

Özçalıcı Mehmet. (2011). *Özdüzenleyici Haritalar Yöntemi İle Banka Müşterilerinin Bölümlendirilmesi* (Yayımlanmış Y.Lisans Tezi). Gaziantep Üniversitesi Sosyal Bilimler Enstitüsü, Gaziantep.

Özdoğan Alper. (2009). *Gen Kümeleme İşleminin Özdüzenleyici Haritalar Kullanılarak Gen Ekspresyonu, Motif Sıklık Ve Gen Konum Verilerinden Faydalanılarak Gerçekleştirimi* (Yayımlanmış Y.Lisans Tezi). Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü. Ankara.

Özkan Yalçın, (2008). *Veri Madenciliği Yöntemleri*. Papatya Yayıncılık, İstanbul.

Pasin, Ö. (2015). *Sağlık Alanında Yapılan Araştırmalarda Kümeleme Algoritmalarının Kullanımı : Bir Uygulama*. (Yüksek Lisans Tezi). Sağlık Bilimleri Enstitüsü. Düzce Üniversitesi. Düzce.

Pyle, D. (2003). *Business Modeling and Data Mining*. Morgan Kaufmann Publishers, Singapore.

Rao, K. V., Govardhan, A., & Rao, K. V. C. (2012). Spatıotemporal Data Mınıng: Issues, Tasks And Applıcatıons. International Journal of Computer Science & Engineering Survey.

Sağıroğlu, S., Besdok, E., Erler, M. (2003). *Mühendislikte Yapay Zeka Uygulamaları I: Yapay Sinir Ağları*, Ufuk Kitap Kırtasiye Yayıncılık, Kayseri.

Sander, J., Ester, M., Kriegel, H. P. P., & Xu, X. (1998). *Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications*. *Data Mining and Knowledge Discovery*.

Sever, S. Z. (2015). *Yoğunluk Tabanlı Kümeleme Metodları Kullanılarak Paralel Veri Madenciliği Gerçekleştirilmesi*. Maltepe Üniversitesi. Fen Bilimleri Enstitüsü. İstanbul.

Sheikholeslami, G., Chatterjee, S., & Zhang, A. (2000). *WaveCluster: a wavelet-based clustering approach for spatial data in very large databases*. *The VLDB Journal*.

Shekhar, S., Jiang, Z., Ali, R., Eftelioglu, E., Tang, X., Gunturi, V., & Zhou, X. (2015). *Spatiotemporal Data Mining: A Computational Perspective*. ISPRS International Journal of Geo-Information.

Silahtaroğlu, G. (2004). *Veri Madenciliğinde Kümeleme Analizi ve Öğretim Başarısının Değerlemesine İlişkin Bir Uygulama*. (Doktora Tezi). Sosyal Bilimler Enstitüsü. İstanbul Üniversitesi, İstanbul.

Sudipto Guha, Rajeev Rastogi ve Kyuseok Shim. (1999). *ROCK: A Robust Clustering Algorithm for Categorical Attributes*. 15th International Conference on Data Engineering, (ICDE'99), Austria.

Sumathi, S., Sivanandam, S.N. (2006). *İntroduction To Data Mining And Its Applications*. Berlin: Springer.

Şule Özmen. (2003). *Veri Madenciliği Süreci, Veri Madenciliği ve Uygulama Alanları*, İstanbul Ticaret Üniversitesi. İstanbul.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S., Golub, T.R. (1999). *Interpretting patterns of gene expression with self-organizing maps : Methods and application to hematopoietic differentiation*. Proc. Natl. Acad. Sci USA Vol. 96.

Tobler. W. R. (1970). *A computer movie simulatingurban growth in the Detroit region*. Economic Geography 46: 234-40.

U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. (1996). *From data mining to knowledge discoveryan review*. Advances in knowledge discovery, AAAI Press/The MIT Press, Cambridge.

Wang XZ, Wang SL. (1997). *Fuzzy comprehensive method and its application in land grading*. Geomatics Inf Sci Wuhan Univ.

Wang, W., J. Yang ve R. Muntz. (1997). *STING: A Statistical Information Grid Approach to Spatial Data Mining.* International Conference of Very Large Data Bases (VLDB'99). Athens.

Xu, X., Ester, M., Kriegel, H., & Sander, J. (1998). *A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases*. 14th International Conference on Data Engineering (ICDE ' 98).

Yapıcı Muhammed Mutlu, *Genetik Algoritma Kullanılarak Ders Çizelgeleme Yazılımının Geliştirilmesi* (Yayımlanmış Y.Lisans Tezi). Gazi Üniversitesi Bilişim Enstitüsü. Ankara.

Yavuzoğlu, Ş. Ö. (2009). *An Evaluation of Clustering and Districting Models for Household Socio-Economic Indicators in Address-Based Population Register System*. (Yüksek Lisans Tezi). Orta Doğu Teknik Üniversitesi.

Yue, J., Mao, S., Li, M., & Zou, X. (2014). *An efficient PAM spatial clustering algorithm based on MapReduce*. 2014 22nd International Conference on Geoinformatics.

Yünel Yunus, (2010). *K-Means Kümeleme Algoritmasının Genetik Algoritma Kullanılarak Geliştirilmesi*, Yıldız Teknik Üniversitesi Fen Edebiyat Fakültesi, İstanbul.

Zeren Fatma. (2010). *Mekansal Etkileşim Analizi*, Ekonometri ve İstatistik, 12.

Zhang, T.; Ramakrishnan, R. ve Livny, M. (1996). *BIRCH: an efficient data clustering method for very large databases*. ACM SIGMOD International Conference on Management of Data. Montreal, Canada: ACM.

EKLER

7.1 EK A

Çizelge A.1 Parametre Değerlerine Göre Ortalama Sıcaklık İçin İllerin Kümelenme Sonuçları I.

İller	Ortalama Sıcaklık	Kümeleme Sonuçları		
		Eps_1 = 1,1 **Eps2** = 10 **MinPts** = 5 Δ_e= 2	**Eps_1** = 1,1 **Eps2** = 10 **MinPts** = 5 Δ_e= 4	**Eps_1** = 1,2 **Eps2** = 5 **MinPts** = 5 Δ_e= 2
Adana	19.25	1	1	1
Adıyaman	17,29	1	1	1
Afyonkarahisar	11,22	11	4	7
Ağrı	6,14	13	8	17
Aksaray	12,16	6	7	7
Amasya	11,36	6	7	8
Ankara	11,98	6	7	7
Antalya	18,61	1	1	1
Ardahan	3,70	13	8	17
Artvin	12,05	10	3	12
Aydın	17,40	2	2	3
Balıkesir	14,03	7	6	3
Bartın	12,76	5	4	7
Batman	16,31	3	1	4
Bayburt	6,93	12	5	17
Bilecik	12,51	5	4	7
Bingöl	11,99	9	5	9
Bitlis	9,02	12	5	14
Bolu	10,47	11	4	13
Burdur	13,17	5	4	7
Bursa	14,58	5	4	7
Çanakkale	15,10	7	6	3
Çankırı	11,15	6	7	7
Çorum	10,54	6	7	16
Denizli	16,19	2	2	3
Diyarbakır	15,75	3	1	4
Düzce	13,02	5	4	7
Edirne	13,67	7	6	11
Elazığ	13,10	9	1	9
Erzincan	10,87	9	5	9
Erzurum	5,35	13	5	17
Eskişehir	11,11	11	4	15
Gaziantep	16,84	1	1	1
Giresun	14,68	4	3	5
Gümüşhane	9,48	12	5	14
Hakkari	10,26	9	5	9

Çizelge A.2 Parametre Değerlerine Göre Ortalama Sıcaklık İçin İllerin Kümelenme Sonuçları II.

İller	Ortalama Sıcaklık	Kümeleme Sonuçları		
		Eps_1 = 1,1 **Eps2** = 10 **MinPts** = 5 Δ_e= 2	**Eps_1** = 1,1 **Eps2** = 10 **MinPts** = 5 Δ_e= 4	**Eps_1** = 1,2 **Eps2** = 5 **MinPts** = 5 Δ_e= 2
Hatay	19.00	1	1	1
Iğdır	12,17	8	5	10
Isparta	12,04	5	4	7
İstanbul	14,68	5	4	7
İzmir	17,92	2	2	3
Kahramanmaraş	16,78	1	1	1
Karabük	13,22	5	4	7
Karaman	11,97	6	7	7
Kars	4,85	13	8	17
Kastamonu	9,70	6	7	13
Kayseri	10,42	6	7	13
Kırıkkale	12,38	6	7	7
Kırklareli	13,28	7	6	11
Kırşehir	11,45	6	7	7
Kilis	17,23	1	1	1
Kocaeli	14,87	5	4	7
Konya	11,61	6	7	7
Kütahya	10,73	11	4	15
Malatya	13,77	9	1	9
Manisa	16,79	2	2	3
Mardin	16,18	3	1	4
Mersin	19,43	1	1	1
Muğla	15,07	2	2	3
Muş	9,35	12	5	14
Nevşehir	10,67	6	7	13
Niğde	11,13	6	7	7
Ordu	14,47	4	3	5
Osmaniye	18,50	1	1	1
Rize	14,44	4	3	6
Sakarya	14,60	5	4	7
Samsun	14,52	6	7	8
Siirt	16,25	3	1	4
Sinop	14,22	6	7	8
Sivas	9,12	6	7	16
Şanlıurfa	18,55	1	1	1
Şırnak	15,06	3	1	4
Tekirdağ	14,07	7	6	11
Tokat	12,39	6	7	8
Trabzon	14,82	4	3	6
Tunceli	12,80	9	5	9
Uşak	12,52	5	4	7
Van	9,45	12	5	14
Yalova	14,70	5	4	7
Yozgat	9,15	6	7	16
Zonguldak	13,69	5	4	7

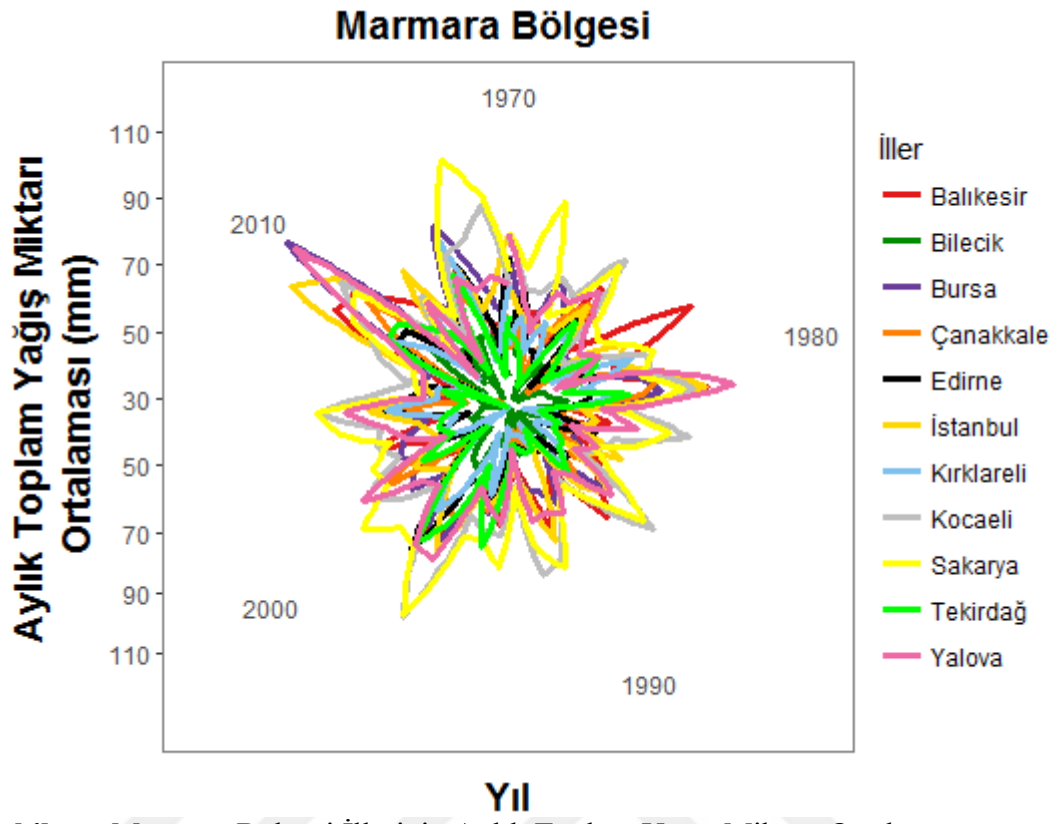
Çizelge A.3 Parametre Değerlerine Göre Toplam Yağış Miktarı İçin İllerin Kümelenme Sonuçları I.

İller	Ortalama Yağış Miktarı	Kümeleme Sonuçları		
		Eps_1 = 1,1 **Eps2** = 20 **MinPts** = 10 Δ_e= 20	**Eps_1** = 1,1 **Eps2** = 20 **MinPts** = 10 Δ_e= 40	**Eps_1** = 1,2 **Eps2** = 20 **MinPts** = 10 Δ_e= 20
Adana	55,02	3	3	2
Adıyaman	56,80	3	3	2
Afyonkarahisar	35,42	4	4	3
Ağrı	43,61	1	1	1
Aksaray	29,01	2	2	2
Amasya	38,48	5	5	4
Ankara	33,96	2	2	2
Antalya	88,44	14	6	14
Ardahan	46,68	1	1	1
Artvin	59,58	3	3	8
Aydın	53,24	6	6	5
Balıkesir	59,38	7	7	5
Bartın	87,41	12	4	12
Batman	39,35	3	3	2
Bayburt	37,25	3	3	2
Bilecik	38,61	4	4	3
Bingöl	76,97	9	3	9
Bitlis	68,37	9	3	9
Bolu	46,75	4	4	2
Burdur	34,69	4	4	3
Bursa	59,29	4	4	3
Çanakkale	51,26	7	7	5
Çankırı	34,27	2	2	2
Çorum	37,69	5	5	4
Denizli	47,49	6	6	5
Diyarbakır	40,29	3	3	2
Düzce	91,41	12	4	12
Edirne	50,02	7	7	5
Elazığ	33,26	3	3	2
Erzincan	31,47	3	3	2
Erzurum	34,07	3	3	2
Eskişehir	30,33	4	4	3
Gaziantep	46,77	3	3	2
Giresun	106,33	13	10	13
Gümüşhane	38,77	3	3	2
Hakkari	63,95	11	3	10
Hatay	77,09	10	3	7
Iğdır	21,92	1	1	1
Isparta	42,97	4	4	3
İstanbul	60,18	4	4	3

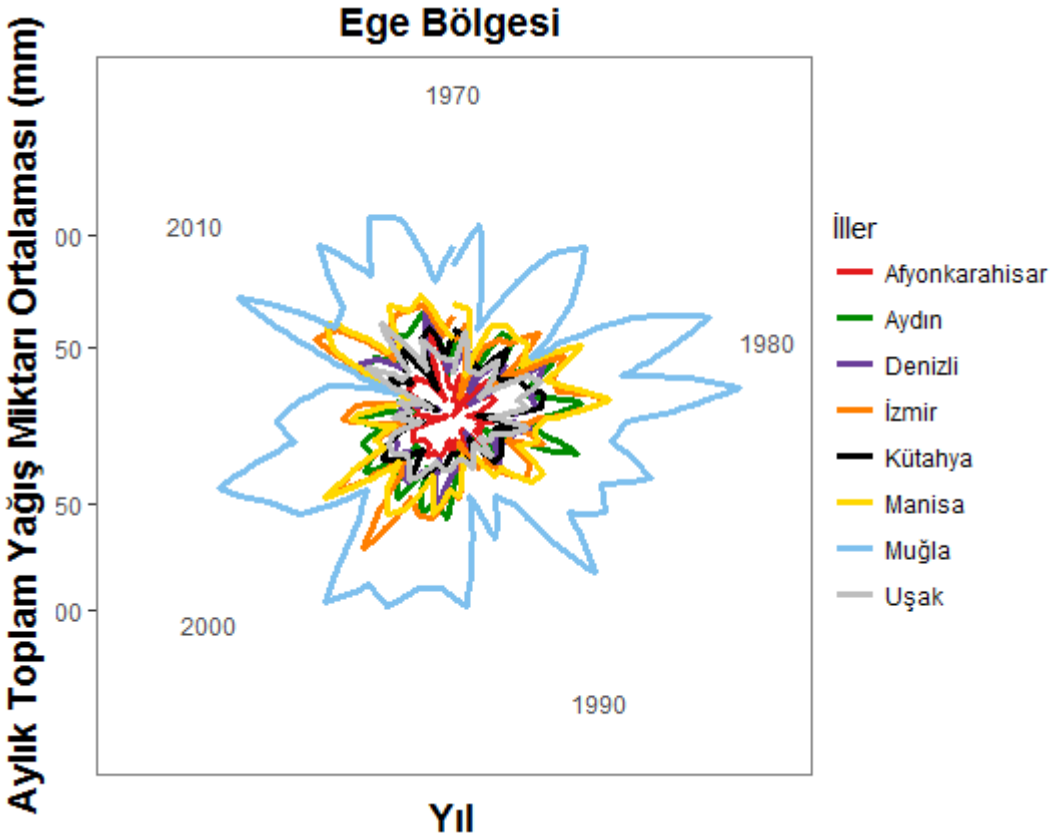
Çizelge A.4 Parametre Değerlerine Göre Toplam Yağış Miktarı İçin İllerin Kümelenme Sonuçları II.

İller	Ortalama Yağış Miktarı	Kümeleme Sonuçları		
		Eps_1 = 1,1 **Eps2** = 20 **MinPts** = 10 Δ_e= 20	**Eps_1** = 1,1 **Eps2** = 20 **MinPts** = 10 Δ_e= 40	**Eps_1** = 1,2 **Eps2** = 20 **MinPts** = 10 Δ_e= 20
İzmir	59.49	6	6	5
Kahramanmaraş	59,99	3	3	7
Karabük	42,01	2	4	2
Karaman	28,04	2	2	2
Kars	41,11	1	1	1
Kastamonu	42,20	2	2	2
Kayseri	33,05	2	2	2
Kırıkkale	32,18	2	2	2
Kırklareli	47,57	7	7	5
Kırşehir	32,02	2	2	2
Kilis	40,42	3	3	2
Kocaeli	68,72	12	4	12
Konya	27,84	2	2	2
Kütahya	46,47	4	4	3
Malatya	31,13	3	3	2
Manisa	61,20	6	6	5
Mardin	53,52	3	3	2
Mersin	49,63	3	3	2
Muğla	95,96	14	6	14
Muş	62,80	9	3	9
Nevşehir	34,88	2	2	2
Niğde	27,82	2	2	2
Ordu	86,98	13	10	13
Osmaniye	71,33	10	3	7
Rize	188,00	15	12	16
Sakarya	71,33	12	4	12
Samsun	58,76	5	5	4
Siirt	56,41	3	3	9
Sinop	57,30	5	5	4
Sivas	37,12	5	5	4
Şanlıurfa	36,24	3	3	2
Şırnak	53,91	3	3	2
Tekirdağ	48,86	7	7	5
Tokat	36,61	5	5	4
Trabzon	69,34	8	3	11
Tunceli	66,61	9	3	9
Uşak	45,87	4	4	3
Van	33,05	3	3	2
Yalova	62,18	4	4	3
Yozgat	49,50	5	5	4
Zonguldak	101,76	16	11	15

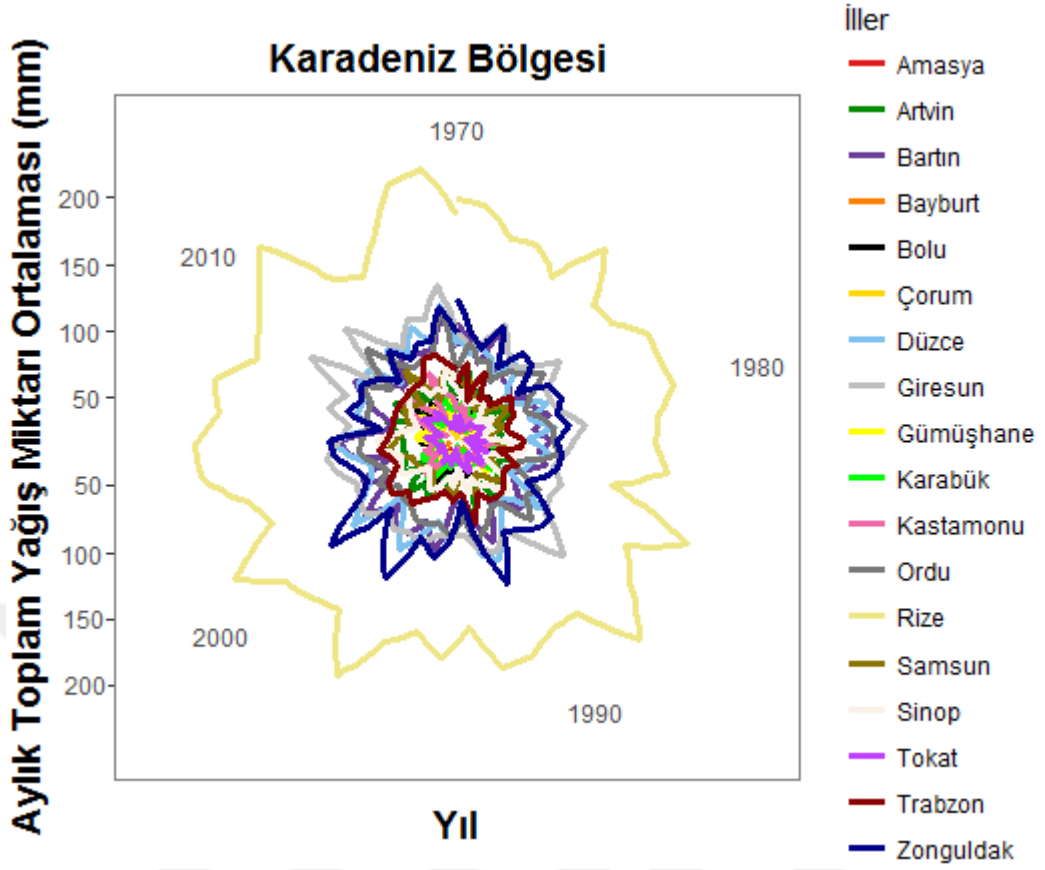
7.2 EK B



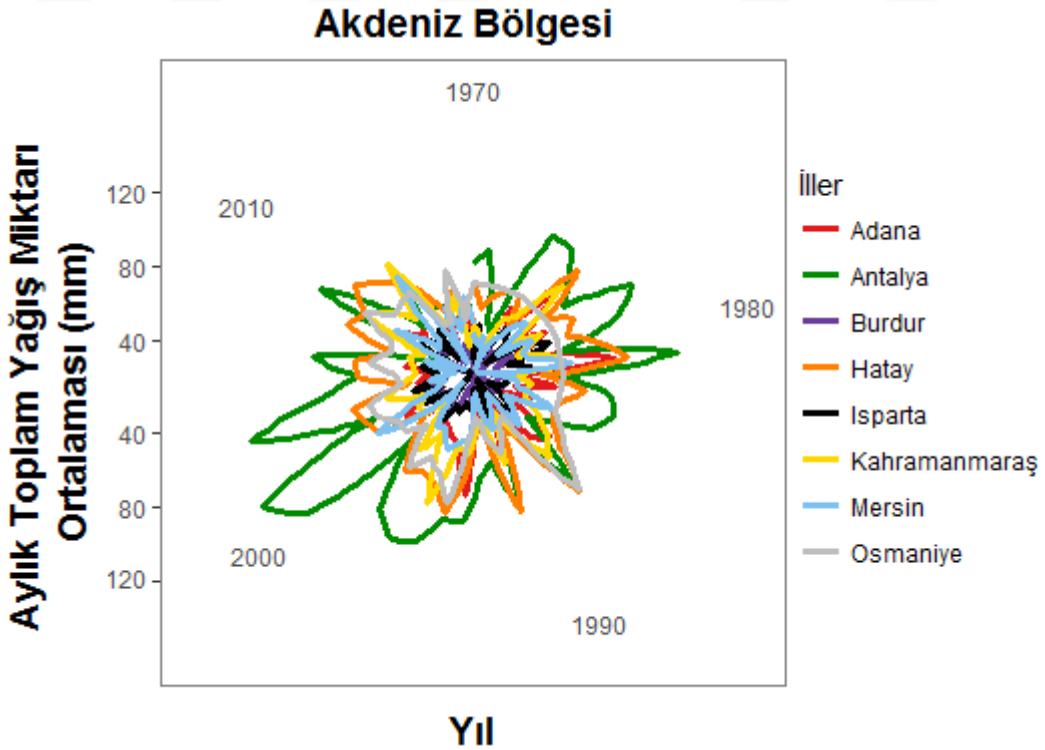
Şekil B.1 Marmara Bölgesi İllerinin Aylık Toplam Yağış Miktarı Ortalaması



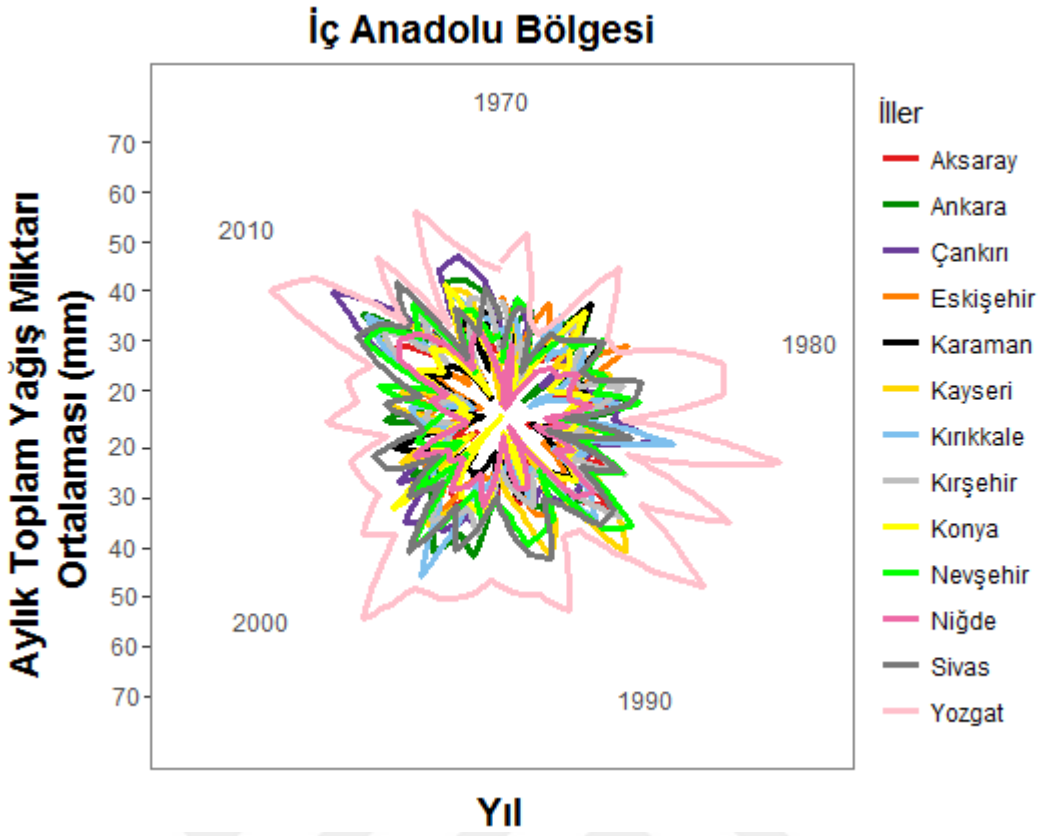
Şekil B.2 Ege Bölgesi İllerinin Aylık Toplam Yağış Miktarı Ortalaması



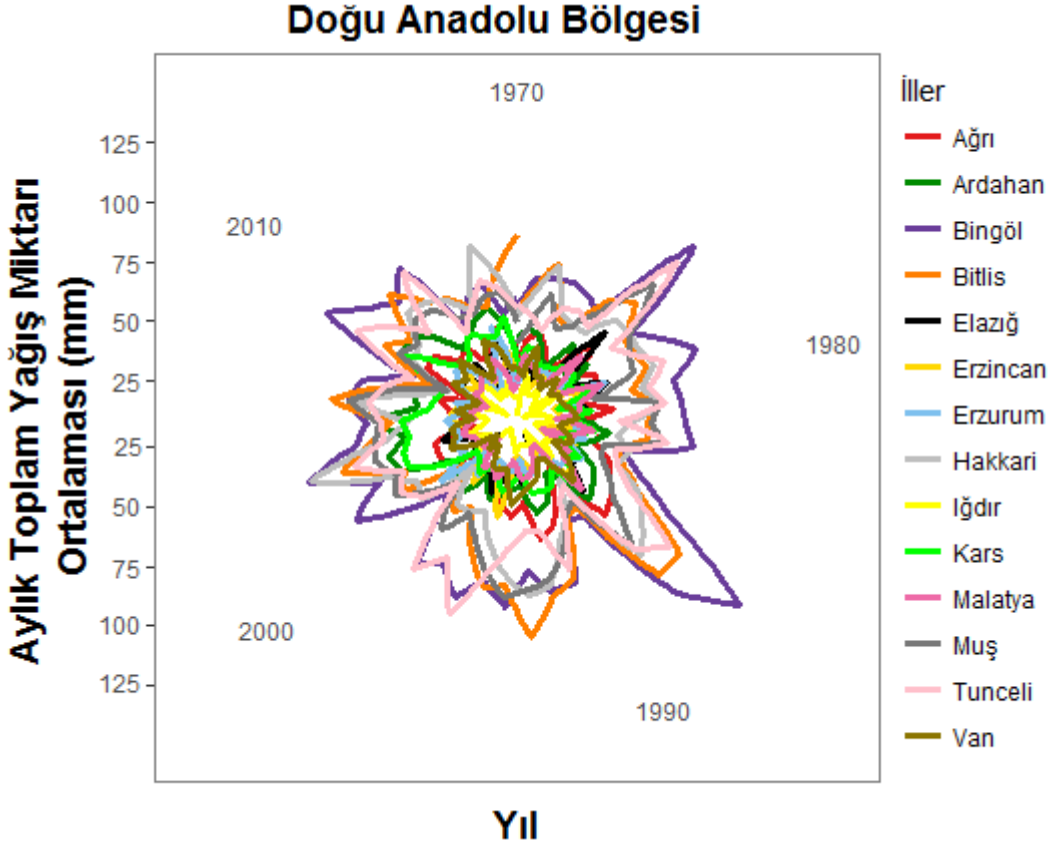
Şekil B.3 Karadeniz Bölgesi İllerinin Aylık Toplam Yağış Miktarı Ortalaması



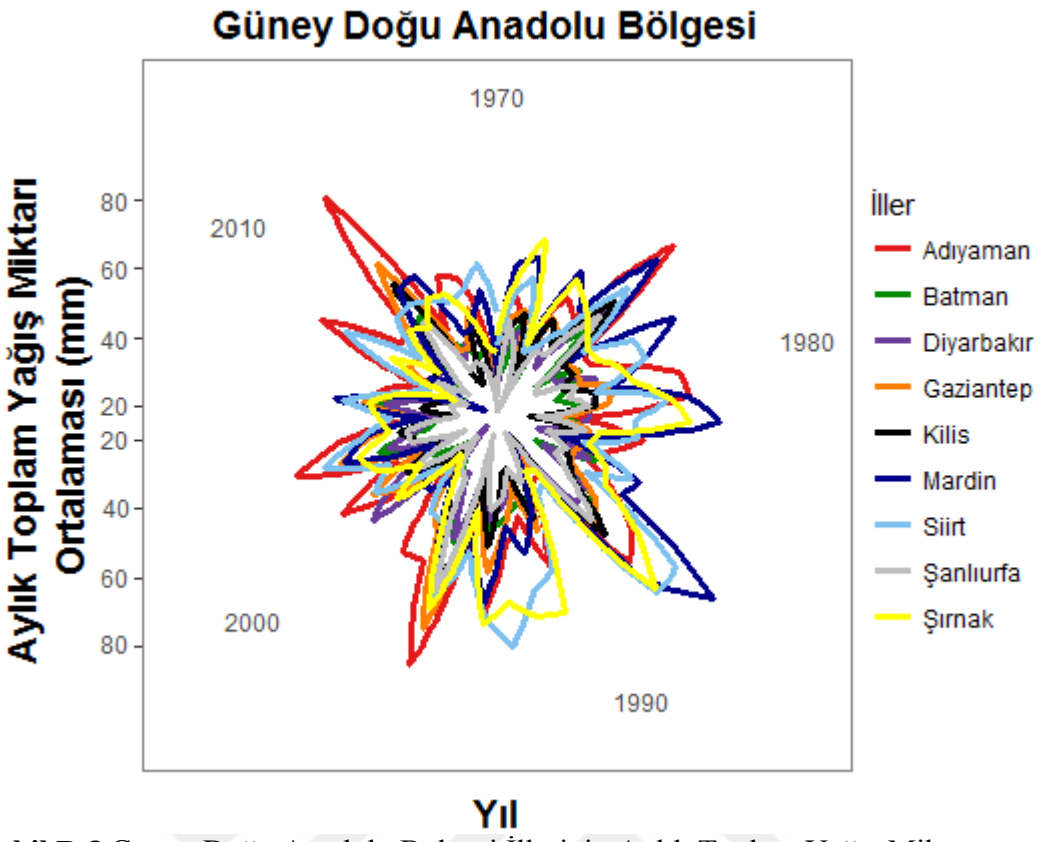
Şekil B.4 Akdeniz Bölgesi İllerinin Aylık Toplam Yağış Miktarı Ortalaması



Şekil B.5 İç Anadolu Bölgesi İllerinin Aylık Toplam Yağış Miktarı Ortalaması



Şekil B.6 Doğu Anadolu Bölgesi İllerinin Aylık Toplam Yağış Miktarı Ortalaması



Şekil B.6 Güney Doğu Anadolu Bölgesi İllerinin Aylık Toplam Yağış Miktarı Ortalaması

ÖZGEÇMİŞ

Adı Soyadı : Turgut Özaltındiş

Doğum Yeri : İstanbul-Bakırköy

Doğum Tarihi : 18.12.1991

İletişim Adresi : turgut.ozaltindis@msgsu.edu.tr

Eğitim Durumu

Lisans : Selçuk Üniversitesi Fen Fakültesi İstatistik Bölümü (2009-2013)

Bilimsel Çalışmalar

1. Afacan, C., Köse A.M., **Özaltındiş, T.**, Özdamar, E.Ö., MSGSÜ Lisans Öğrencilerinin İstatistiğe Yönelik Tutumlarının Kariyer Planı Ve Gelecek Yönelimleri İle İlişkisinin Araştırılması, , XVIII. Uluslararası Ekonometri Yöneylem Araştırması ve İstatistik Sempozyumu, 05 – 07 Ekim 2017, Trabzon/Türkiye, Bildiri Özetleri Kitapçığı.

2. Köse A.M., Afacan C., **Özaltındiş T.**, Özdamar, E.Ö., Grup Karşılaştırmalarına Alternatif Bir Yaklaşım Olarak Anom Testi, XVIII. Uluslararası Ekonometri Yöneylem Araştırması ve İstatistik Sempozyumu, 05 – 07 Ekim 2017, Trabzon/Türkiye, Bildiri Özetleri Kitapçığı.