

T. C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**TIBBİ VERİLERİ ÇİZGE BÖLÜTLEME YÖNTEMİ İLE
SINIFLANDIRMA VE KÜMELENDİRME**

Mehmet YİĞİTER

Tez Yöneticisi
Yrd. Doç. Dr. Ali KARCI

YÜKSEK LİSANS TEZİ
BİYOMÜHENDİSLİK ANABİLİM DALI

ELAZIĞ, 2008

T. C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TIBBİ VERİLERİ ÇİZGE BÖLÜTLEME YÖNTEMİ İLE SINIFLANDIRMA VE KÜMELENDİRME

Mehmet YİĞİTER

YÜKSEK LİSANS TEZİ
BİYOMÜHENDİSLİK ANABİLİM DALI

Bu tez, 07.08.2008 tarihinde aşağıda belirtilen jüri tarafından oybirliği / oyçokluğu ile başarılı / başarısız olarak değerlendirilmiştir.

Danışman: Yrd. Doç. Dr. Ali KARCI

Üye: Doç. Dr. İbrahim TÜRKOĞLU

Üye: Yrd. Doç. Dr. Ahmet ÇINAR

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun/...../..... tarih ve sayılı kararıyla onaylanmıştır.

TEŞEKKÜR

Bu tez de dahil olmak üzere yüksek lisans çalışmalarında ilgi ve yardımlarını esirgemeyen başta değerli danışman hocam Sayın Yrd. Doç. Dr. Ali KARCI olmak üzere Sayın Doç. Dr. İbrahim TÜRKOĞLU'na, Sayın Yrd. Doç. Dr. Mehmet KAYA'ya, KOAH hasta verilerini vererek ve tıbbi konularda destek ve yardımlarını esirgemeyen İnönü Üniversitesi Turgut Özal Tıp Merkezi Göğüs Hastalıkları Anabilim Dalında Doç. Dr. Süleyman HACIEVLİYAGİL ve Dr. Kadir YILDIZ'a, özellikle matrisler ile ilgili olmak üzere kitap temini, konu anlatımı ve problem çözümünde ilgi ve yardımlarını esirgemeyen İnönü Üniversitesi Eğitim Fakültesi İlköğretim Matematik Bölümü Öğretim Üyeleri Doç. Dr. Bilal ALTAY ile Doç. Dr. Celal ÇAKAN'a ve veri madenciliği konusunda kaynak temini ile MATLAB kullanımında yardımcı olmak şeklinde katkıları olan İnönü Üniversitesi Rektörlüğü Enformatik Bölümünde iş arkadaşım ve İnönü Üniversitesi Fen Bilimleri Enstitüsünde doktora öğrencisi olan Fatih M. AVCU ile Hüseyin ARPACI'ya teşekkür ederim.

İÇİNDEKİLER

İÇİNDEKİLER	I
ÇİZELGELER LİSTESİ	III
ŞEKİLLER LİSTESİ	IV
EKLER LİSTESİ	V
KISALTMALAR LİSTESİ	VI
ÖZET	VII
ABSTRACT	VIII
1. GİRİŞ	1
2. VERİ MADENCİLİĞİ	8
2.1. Veritabanı	8
2.2. Veri Madenciliği	8
2.3. Veri Temizleme	10
2.4. Veri Bütünleştirme	11
2.5. Veri İndirgeme	11
2.6. Veri Dönüştürme	11
2.7. Min-Maks Normalleştirilmesi	11
2.8. Z - değeri Standartlaştırma	12
2.9. Veri Madenciliği Uygulama Alanları	12
2.10. Tıpta Veri Madenciliği	12
2.11. İlişki Kuralları	14
2.12. Sınıflandırma	15
2.12.1. Sınıflandırmanın Temel Kuralları	15
2.12.2. Sınıflandırma Yöntemleri	15
2.12.3. Sınıflandırma Aşamaları	16
2.12.3.1. Model Kurma Süreci	16
2.12.3.2. Modelin Uygulanma Süreci	16
2.12.4. Sınıflandırma Modelini Değerlendirme	16
2.13. Kümeleme	17
2.13.1. Kümeleme Yöntemleri	19
3. ÇİZGELER	20
3.1. Çizgelerde Temel Kavramlar: Ayrıtlar ve Düğümmler	20

4. ÇİZGE BÖLÜTLEME YÖNTEMLERİ	22
4.1. Bölüt Oluşturma Algoritmaları	23
4.1.1. Gelişi Güzel Bölütleme Yapan Algoritmalar	23
4.1.1.1. Lineer Algoritma	23
4.1.1.2. Gelişi Güzel Algoritma	24
4.1.1.3. Serpme Algoritması	24
4.1.2. Spektral Çizge Bölütleme	25
4.1.2.1. Laplace matrisi ve Fiedler vektörleri	26
4.1.2.2. Spektral Bölütleme Yöntemleri	27
4.1.2.3. Matrislerin Özdeğeri ve Özvektörünün Hesabı	27
4.1.3. Eylemsizlik Algoritması	32
4.1.4. Çok Seviyeli Çizge Bölütleme	35
5. ÇİZGE BÖLÜTLEME YÖNTEMİ UYGULAMALARI	40
5.1. Örnek 1: Çizge Bölütleme Yönteminin bir alış-veriş verisine uygulanması	40
5.2. Örnek 2: Hepatits veritabanı Çizge Modelinin Oluşturulması	43
5.3. Örnek 3: KOAH veritabanı Çizge Modelinin Oluşturulması	47
5.4. Uygulama 1: Çizge Bölütleme Algoritması ile Hepatits Veritabanının Kümelenmesi	50
5.5. Uygulama 2: Çizge Bölütleme Algoritması ile KOAH Veritabanının Kümelenmesi	52
6. SONUÇ	53
KAYNAKLAR	55
ÖZGEÇMİŞ	57
EKLER	58
Ek 1. Hepatits Veritabanı	58
Ek 2. KOAH Veritabanı	63
Ek 3. Hazırlanan Yazılımın Önemli Algoritmaları	66

ÇİZELGELER LİSTESİ

Çizelge 2.1. Gürültülü veriler barındıran örnek veritabanı	10
Çizelge 2.2. Alışveriş sepet bilgileri	14
Çizelge 2.3. Kümeleme işlemi uygulanacak veri	18
Çizelge 2.4. Algoritmanın uygulanması sonucunda elde edilen kümeler	18
Çizelge 5.1. Örnek veritabanı	40
Çizelge 5.2. Hepatits veritabanında rasgele 7 satır 19 sütunluk bir veri	43
Çizelge 5.3. Hepatits veritabanında rasgele 7 satır 16 sütunluk bir veri	44
Çizelge 5.4. KOAH veritabanında rasgele 9 satır 4 sütunluk bir veri	47

ŞEKİLLER LİSTESİ

Şekil 1.1. Bulanık Kümeler (a) klasik küme, (b) bulanık küme	2
Şekil 1.2. (a) Sinir hücresi yapısı, (b) yapay sinir hücresi	3
Şekil 1.3. (a) İnsan bağışıklık sistemi, (b) yapay bağışıklık algoritmaları	4
Şekil 1.4. Karıncaların yuva-yiyecek arasında en kısa yolu bulması	5
Şekil 2.1. Veritabanlarında bilgi keşfi süreci	9
Şekil 2.2. Kümeleme	17
Şekil 2.3. Hiyerarşik kümelerin görünüşü.....	18
Şekil 2.4. Bir veritabanının kümelenmesi	19
Şekil 4.1. Lineer çizge bölütleme algoritması	24
Şekil 4.2. Eylemsizlik yöntemi ile 2D’de çizge bölütleme	33
Şekil 4.3. Eylemsizlik yöntemine göre aynı ağın iki farklı şekilde bölütlenmesi	34
Şekil 4.4. Bir doğru ile noktaları iki eş parçaya bölütleme	35
Şekil 4.5. Çok seviyeli çizge bölütleme yöntemi	37
Şekil 5.1. Çizelge 5.1’deki örnek veritabanının çizge modeli.....	40
Şekil 5.2. Çizelge 5.3’deki örnek veritabanının çizge modeli	44
Şekil 5.3. Çizelge 5.4’deki örnek veritabanının çizge modeli	47

EKLER LİSTESİ

Ek 1. Hepatits Veritabanı	58
Ek 2. KOAH Veritabanı	63
Ek 3. Hazırlanan Yazılımın Önemli Algoritmaları	66

KISALTMALAR LİSTESİ

ADJ	Adjacent – Bitişik, yakın, komşu
BMI	Body Mass Index - Vücut Kütle İndeksi
DNA	Deoksiribonükleik asit
NP	Non-deterministic Polynomial - belirleyici olmayan polinom
VLSI	Very-large-scale integration – Çok geniş ölçekli tümdevre
YBS	Yapay Bağışıklık Sistemi
YSA	Yapay Sinir Ağları

ÖZET

Yüksek Lisans Tezi

TIBBİ VERİLERİ ÇİZGE BÖLÜTLEME YÖNTEMİ İLE SINIFLANDIRMA VE KÜMELENDİRME

Mehmet YİĞİTER

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Biyomühendislik Anabilim Dalı

2008, Sayfa: 70

Bilgisayar başta olmak üzere bilgi teknolojilerinin hızlı gelişiminin olumlu etkisiyle “Bilgi Çağı” olarak tanımlanan günümüz dünyasında karmaşık biçimlerdeki büyük veri yığınları içinden değerli bilgi parçalarını ayıklamak ve kullanılır hale getirmek “Veri Madenciliği” işlemleri ile olmaktadır. Bilgisayar mühendisliği ve biyomühendislik bilim dallarında veri miktarının artması sonucunda uzmanın yararlı bilgiyi çıkarması zorlaşmaktadır. Bu yararlı bilgiyi çıkarmada, veri toplanması veya üretilmesi için sinyali doğru algılama, verilerin doğru sınıflandırılması ve kümelendirilmesi ile çok boyutlu ilişkilerinin ortaya konularak analizi çok önemlidir.

Bu çalışmada “**tıbbi verileri çizge bölütleme yöntemi ile sınıflandırma ve kümelendirme**” için doğrusal cebir tabanlı bir hesapsal yöntem kullanılarak ve bu yöntemin başarımının ortaya konulması için veri madenciliği alanına uygulaması yapıldı.

Çizge bölütleme yöntemi, başta **tıbbi verilerde sınıflandırma ve kümelendirme** problemleri olmak üzere bir çok bilimsel problemin çözümünün kolaylaştırılması veya bu çözümün bulunması amacıyla yapılan çalışmalar sonucunda ortaya çıkmıştır. Bundan dolayı çizgelerin analizi ve testi yöntemin geliştirilmesi çok önemlidir.

Bu çalışmanın en önemli özelliği, tıbbi verileri çizge bölütleme yöntemi ile sınıflandırma ve kümelendirme algoritmalarının uygulamasının yapılmış olmasıdır.

Anahtar Kelimeler: Çizge bölütleme, sınıflandırma, kümeleme, kümelendirme, tıbbi veriler, veri madenciliği.

ABSTRACT

Master of Science Thesis

THE CLASSIFICATION AND CLUSTERING OF MEDICAL DATA BY GRAPH PARTITIONING METHOD

Mehmet YİĞİTER

Firat University

Graduate School of Natural and Applied Sciences

Department of Bioengineering

2008, Page: 70

In the todays world known as “knowledge age”, to get the important data from the large database with complex formats and to use them for utility it is made by procedure of datamining. Because of the increasing of data amount in the departments of Computer Engineering and Bioengineering, it is difficult to get the utility knowledge by specialist. For this, it is very important the collection of data or the accurate sensation of signal for the accumulation of data, the classification and clustering of data in the appropriate form, by analysis the multi-dimensional togetherness relationship.

In this study, it is built up a calculation method based on linear algebra for “the classification and clustering of medical data by graph partitioning method” and it is used the data mining for the evaluation of this method.

The graph partritioning method has been discovered to the faciliation of solution of some scientific problems especially the classification and clustering of medical data.

For this reason, it is very important to develop this method (the analysing and testing of graphs).

The most important attribute of this study is to give applications for the classification and clustering of medical data by graph partitioning method.

Key Words: Graph partitioning, classification, clustering, medical data, data mining.

1. GİRİŞ

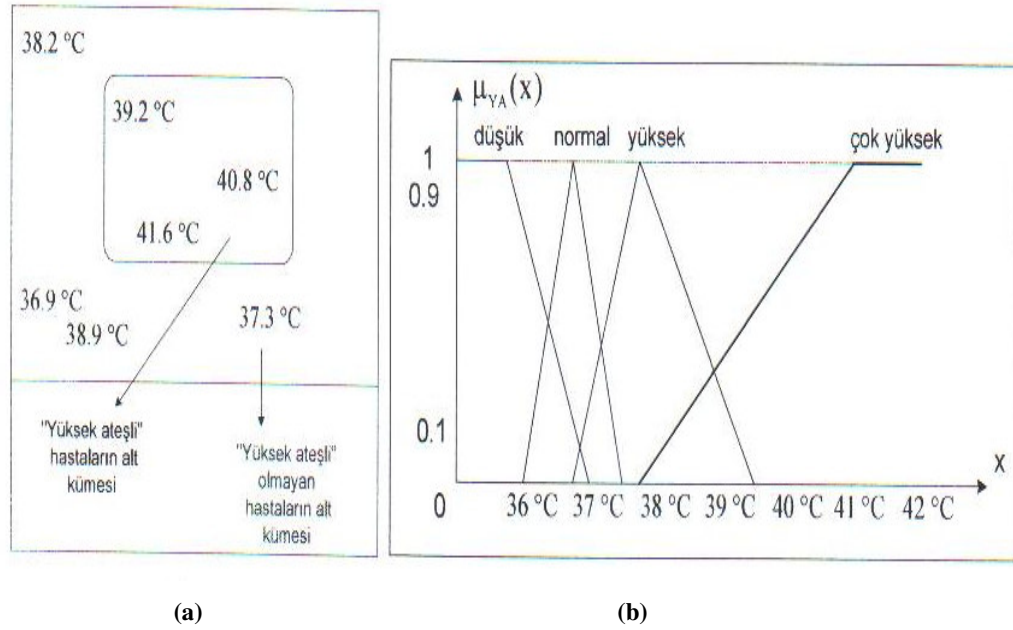
Bilgisayar ve bilgi teknolojilerinin günümüzdeki hızlı gelişiminin olumlu etkisiyle analitik olarak çözülebilen problemlerin sayısal çözümleri üzerinde de geniş çaplı araştırmalar ve çalışmalar yapılmıştır. Analitik olarak çözülemeyen problemlerin hesapsal olarak çözümleri üzerinde de çalışmalar yapılmaktadır. Bulanık mantık, yapay sinir ağları, genetik algoritmalar, yapay bağışıklık sistemleri, karınca koloni algoritması, fidan gelişimi algoritması vb. çok sayıda hesapsal yöntem geliştirilmiştir. Bu yöntemlerin temel özelliği doğadan esinlenerek geliştirilen yöntemler olmasıdır [1].

Bulanık mantık, insanın düşünme tarzı ile hesaplama yapan bir yöntemdir. İnsan düşüncesinde bazı kavramlar arasında keskin geçiş veya keskin sınır yoktur. Bulanık mantıkta; bir nesne sadece bir sınıfa ait, bir duruma sahip, gibi benzer kavramlarda keskin karar vermez. Bir nesne birden fazla sınıfa ait olabilir ve şartlara göre birden fazla duruma sahip olabilir. Bundan dolayı bulanık mantık; verileri ilk önce bulandırma sürecinden geçirir ve bazı sonuçlar elde eder, ondan sonra durulama sürecinden geçirir ve istenilen sonucu/sonuçları elde etmiş olur. Bulanık mantık, Şekil 1.1’de görüldüğü gibi model temelli sistem tasarımı mühendislik deneyimlerine dayandıran bir teknolojidir. Burada klasik kümeler ile bulanık kümeler arasındaki farka bir örnek ile bakılabilir. Hastalar için “yüksek ateş” ölçüsü olarak $T \geq 39^\circ \text{C}$ kabul edilir. Bu durumda, dilsel değişkenlerin bulanık kümelerle ifadesi Şekil 1.1’deki gibi yapılabilir.

Bulanık mantık sistem davranışını tanımlamak için dilsel değişkenler kullanır ve bu sayede tam bir matematiksel modellemeye olan ihtiyaç ortadan kalkar.

Verilen bir sistem için tam bir matematiksel modelin kolayca bulunamadığı durumlarda; doğrusal olmayan, zaman kısıtlamalarının ve çok parametrenin olması durumlarında, problem hakkında tasarım bilgisi varsa veya tasarım sırasında elde edilebildiğinde bulanık mantık kullanılması kolaylıklar sağlar.

Mühendislik tecrübe ve deneyimlerini, bilinen modelleme anlayışı ile birleştirmenin sonucunda sistem performansının artması, var olan tasarımlara yeni fonksiyonlar kazandırması, geliştirme sürecini ve pazara sunma süresini, ileri geliştirme programları kullanarak kısaltması bulanık mantığın kazandırdığı avantajlardır.



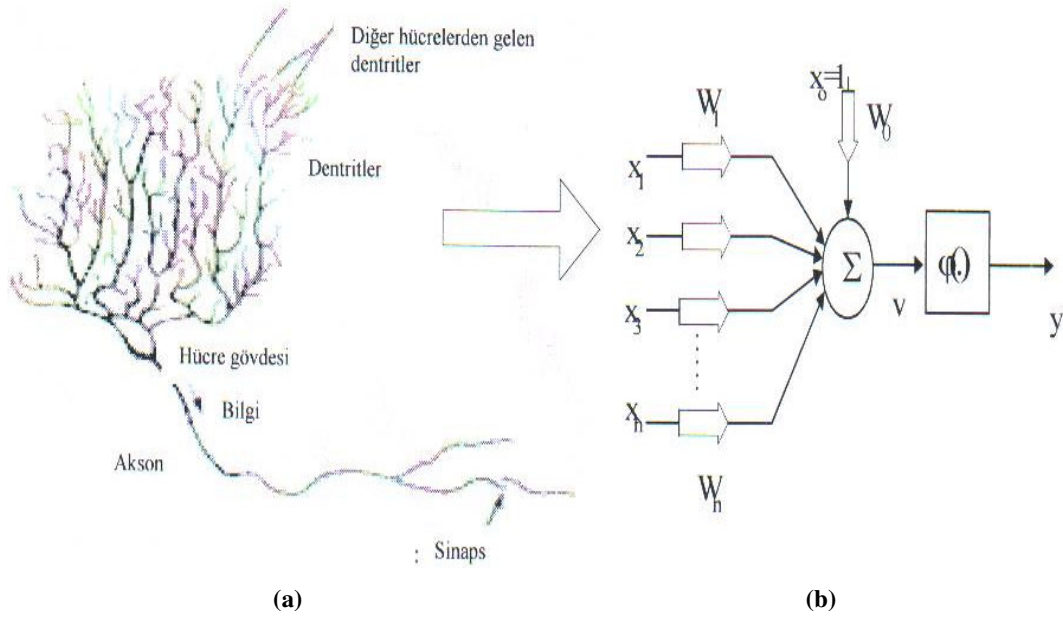
Şekil 1.1. Bulanık Kümeler (a) klasik küme, (b) bulanık küme

Şekil 1.2’de görüldüğü gibi beynin bütün davranışlarını tam olarak modelleyebilmek için fiziksel bileşenlerinin doğru olarak modellenmesi gerektiği düşüncesi ile çeşitli yapay hücre ve ağ modelleri geliştirilmiştir.

Böylece Yapay Sinir Ağları (YSA) olarak adlandırılan yeni ve günümüz bilgisayarlarının algoritmik hesaplama yönteminden farklı bir bilim alanı ortaya çıkmıştır.

Genel olarak YSA, beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan, tek katmanlı ya da çok katmanlı olarak düzenlenebilen ve paralel olarak çalışan çok sayıda doğrusal olmayan yapay hücreden (işlem elemanı) meydana gelen bir sistem yada matematiksel model olarak tanımlanır. Nesne/örüntü tanıma, sinyal işleme, arıza analizi ve tespiti, sistem tanılama (modelleme) ve denetimi vs gibi çeşitli alanlarda yaygın olarak kullanılmaya başlanmış ve kullanıldığı alanlardaki problemlerin çözümüne yeni yaklaşımlar getirmiştir [1].

YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle katmanlar şeklinde düzenlenir. Donanım olarak elektronik devrelerle yada bilgisayarlarda yazılım olarak gerçekleştirilebilir.



Şekil 1.2. (a) Sinir hücresi yapısı, (b) yapay sinir hücresi

Beynin bilgi işleme yöntemine uygun olarak YSA, bir öğrenme sürecinden sonra bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış bir işlemcidir. Hücreler arasındaki ağırlıklar, arzu edilen tasarım amaçlarını sağlayacak şekilde çeşitli öğrenme kuralları ile ayarlanır.

Bu yapısı ile YSA, öğrenme sürecinde bilgiyi toplayan ve ağırlıkları yardımıyla bu bilgiyi saklayan paralel bir işlemcidir.

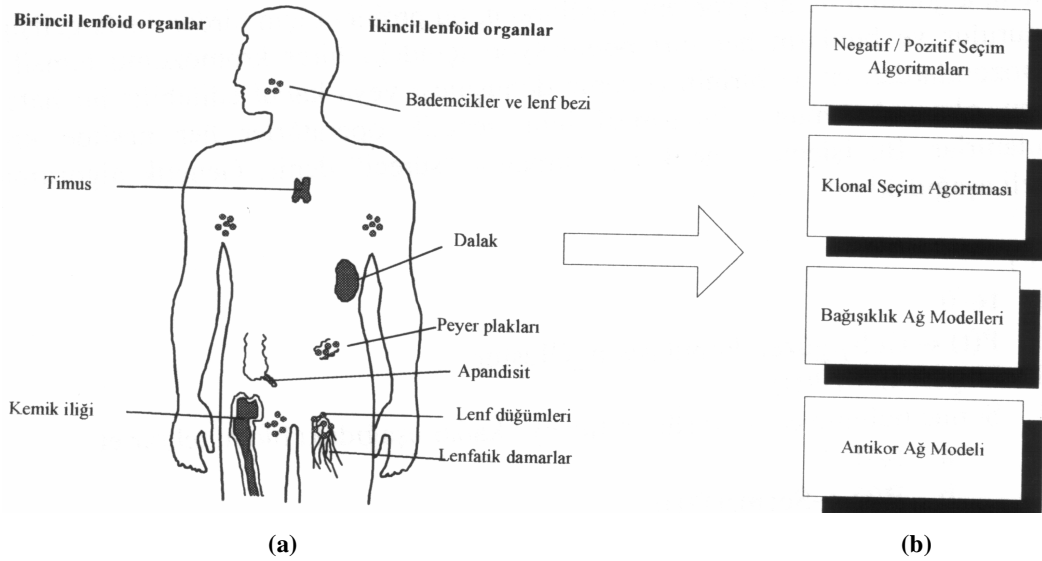
Bugün, çeşitli YSA yapıları ve öğrenme algoritmaları geliştirilmiştir. Herhangi bir uygulama için uygun YSA yapısını ve öğrenme algoritmasını seçmek yada geliştirmek gerekir.

Genetik algoritmalar ise, canlıların çiftleşmesi sonucunda kendi özelliklerini bir sonraki nesle aktarma yeteneklerini, güçlü olanın yaşama şansının yüksek olması ve bazı durumlarda önceden nasıl olacağı bilinmeyen değişimlerin bir hesaplama yöntemi içerisinde bir arada kullanılması sonucunda ortaya çıkmış bir yöntemdir [2].

Genellikle genetik algoritmalar problem çözmek için bir araç olarak kullanılır. Özellikle bilgisayar bilimi veya bilgisayar mühendisliği dışındaki mühendislik bölümleri genetik algoritmayı bir problem çözme aracı veya çözüm arama yöntemi olarak kullanırlar. Bilgisayar mühendisliği bölümünde de genetik algoritma problem çözmede bir araç veya çözüm arama yöntemi olarak kullanıldığı gibi genetik algoritmanın teorisi üzerinde de çalışmalar yapılmaktadır. Zor mühendislik problemlerini çözmek amacıyla yeni yaklaşımlar geliştirmek için, özellikle biyolojik sistemlerden esinlenerek yapılan çalışmalara son yıllarda ilgi gittikçe artmıştır.

Bağışıklık sistemi, insan vücudunun hastalıklara karşı savunma mekanizmasını oluşturan karmaşık bir sistemdir, vücudu yabancı ve zararlı maddelerden korur. Bu sistem vücudumuza giren milyonlarca bakteri, mikrop, virüs, toksin ve parazitlere karşı korunmak için düzenlenmiştir. İnsan vücudu, hastalıklara karşı bir savunma sistemiyle donatılmıştır ve bu yüzden de kendi kendini iyileştirme yeteneğine sahiptir. Hastalığa yol açan maddeler tarafından uyarıldığında, savunma ve bağışıklık sistemi harekete geçer. Bağışıklık sistemine olan ilgi de son birkaç yıldır artarak devam etmektedir.

Bilgisayar bilimcileri, mühendisler, matematikçiler, filozoflar ve diğer araştırmacılar, karmaşıklığı beyne benzeyen bu sistemin özellikle yetenekleri üzerine ilgi duymuş; yeni araştırma alanı olan Yapay Bağışıklık Sistemleri baş göstermiş ve Şekil 1.3’de görüldüğü gibi formsal bir çatı oluşturma çalışmaları başlamıştır.



YBS, yapay sinir ağları ve genetik algoritmalara benzer şekilde, bazı biyolojik sistemlerin özet modelleridir ve anormallik tespiti (virüs tespiti), ağ güvenliği, robotik, kontrol, optimizasyon, örüntü tanıma, makine öğrenmesi, çizelgeleme, hata teşhisi gibi bir çok alanda uygulaması bulunmaktadır [3].

Tanım olarak YBS, teorik bağışıklık ve kompleks problem alanlarına uygulanan gözlemlenmiş bağışık fonksiyonlar, ilkeler ve modellerden esinlenmiş hesapsal sistemlerdir.

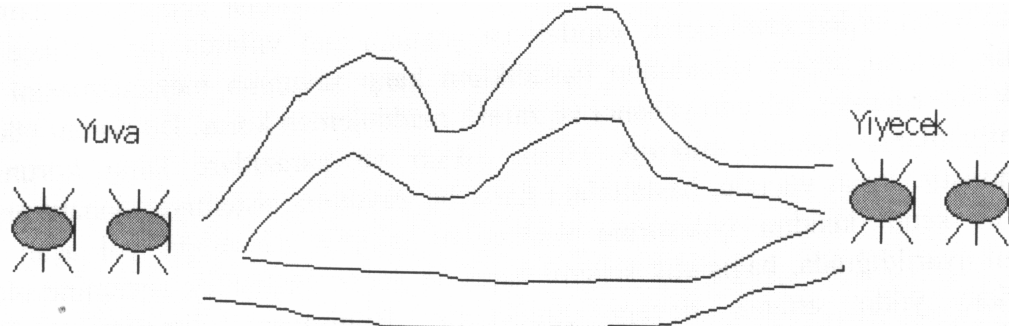
İki temel bileşen (kemik iliği ve timus) ve iki ayrı teori (klonal seçim ve bağışık ağ) bağışıklık sistemini modellemek için kullanılır. Kemik iliği modeli, hücrelerin ve moleküllerin

repertuarını üretmede kullanılır. Timus modeli öz / öz olmayan ayrımı yapmaya yetenekli hücre ve moleküllerin repertuarını üretmede kullanılır.

Karınca koloni optimizasyonu ise, karıncaların davranışlarını esas alır [4].

Arılar, karıncalar ve hatta bakteriler hayatta kalma stratejilerini çok karmaşık grup davranış biçimleri ile gerçekleştirirler.

Karıncalar, kolonilerinin menfaatleri için beraber çalışan sosyal böceklerin en iyi örneklerindendir. Koloni halinde yaşayan karıncalar yiyecek bulmak için ilk olarak öncü karıncaları tek başına gönderirler. Bu öncüler çevreyi araştırarak uygun yiyecek kaynağını bulmaya çalışır. Öncüler yiyecek bulursa, koloninin olduğu yere geri dönerken arkalarında özel bir koku izi bırakarak ilerler. Şekil 1.4'te görüldüğü gibi bu iz sayesinde diğer karıncalar da bu yiyecek kaynağını bulabilirler.



Şekil 1.4. Karıncaların yuva-yiyecek arasında en kısa yolu bulması

Aslında karıncalar yukarıda bahsedilenden çok daha karmaşık bir yöntemi uygularlar. Yiyecek kaynağını başarıyla bulan öncü karınca geri dönerken en kısa yoldan dönmüş olmayabilir, tabii bu da kolonidekilerin karmaşık yollarla kaynağa gitmesine neden olacaktır.

Aynı yiyecek kaynağını keşfeden başka bir öncü karınca belki buraya daha kestirme bir yol bulmuş olabilir.

Bu nedenle bazen koloniler gereksiz derecede uzun bir yoldan gitmek zorunda kalabilirler. Ama kestirme yollardaki izler daha düzenli olarak yenilenir ve bu sayede de karıncalar daha belirgin izi olan yani daha kısa yolu tercih ederek gereksiz derecede uzun yollardan ilerlemek zorunda kalmazlar.

Doğru ve zamanında karar almanın hasta sağlığı üzerindeki etkisi tartışmasız çok önemlidir. Hastane bünyesinde toplanan operasyonel veriler, hasta verileri, uygulanan tedavi yöntemi ve tedavi sürecine dair veriler yöneticiler açısından bakıldığında; hastanedeki servislerin ve programların başarısının görüntülenmesi, kaynakların maliyetlerle göreceli olarak

kullanımı, kaynak kullanımı ve hasta sayıları ile ilgili trendlerin tahmini, harcamalarla ilgili normal olmayan durumların anlık tespiti ve yolsuzlukların engellenmesi, hastanede uygulanan tedavi yöntemlerinin başarısının irdelenmesi açısından önemli bilgileri içermektedir. Bu veriler başarılı tedavi sonuçları almada etken faktörlerin belirlenmesi, ameliyatlarda yüksek risk faktörlerinin sınanması, hasta verilerinin yaş, cinsiyet, ırk ve tedavi yöntemi gibi faktörlere göre sınıflandırılması, hasta sağlığı açısından geriye dönük faktörlerin sınanması, tedavi yöntemi geliştirme vb. amaçlarla kullanılmaktadır. Dünya çapında çok sayıda başarılı uygulama örneği mevcuttur. Örneğin, San Francisco Hearth Institute; hasta sonuçlarının iyileştirilmesi, hastanın hastanede kalma süresinin azaltılması, vb. amaçlarla bir çalışma başlatmış ve kurum bünyesinde toplanan verilerden hastanın geçmişine ait veriler, laboratuvar verileri, medikal verileri vb. bilgiye dönüştürmüştür.

Özellikle tıp alanında olduğu gibi, büyük miktarlardaki veriler içindeki gizli örüntüler, geleneksel çözümleme araçlarıyla bulunamaz. Toplanan veri miktarı büyüdükçe ve toplanan verilerdeki karmaşıklık arttıkça, daha iyi çözümleme tekniklerine olan gereksinim de artmaktadır. Tıbbi veritabanları genellikle heterojendir. Doktorlar; görüntüler, sinyaller ya da diğer klinik bilgilerle ilgili yorumlarını, standartlaştırılması çok güç olan serbest metinler olarak yazmaktadır. Tıptaki temel veri yapıları, fiziksel bilimlerin birçok alanıyla karşılaştırıldığında, matematiksel olarak karakterize edilmeye pek uygun değildir.

Kümeleme analizi bir veri setinin farklı gruplar içerip içermediğini belirlemek ve eğer içeriyorsa, bu grupları tespit etmek için kullanılan çok değişkenli istatistiksel yöntemlerin genel adıdır [5,6]. Çok boyutlu uzayda verilerin özetlenmesi ve tanımlanmasında yol gösterici bir araştırma yöntemi olan kümeleme analizi; nispeten heterojen olan farklı gruplardaki gözlem yapılarını yada nispeten homojen olan benzer gruplardaki gözlemleri uygun yöntemlerle gruplamaya olanak sağlayan bir yöntem olarak bilinmektedir.

Kümeleme analizi; gözlemler arası kümelendirme, değişkenler arası sınıflandırma yada gözlemlerin ve değişkenlerin bir arada sınıflandırılmasını amaçlar.

Bu çalışmada; özellikle tıbbi veriler göz önüne alınarak bir hesapsal yöntem geliştirilmekte ve bu yöntemin performansının ortaya konulması için veri madenciliği alanı kullanılarak bir hesapsal yöntem ortaya konulmaktadır.

Geliştirilecek olan yöntem, etkili şekilde özellikle tıbbi verilerde sınıflandırma ve kümeleme problemlerine uygulanabilecektir. Günümüzde oldukça önemli bir yere sahip olan ve aslında bir arama ve optimizasyon işi olan veri madenciliğine uygulanarak; doğru, ilginç ve anlaşılabilir kuralların etkili şekilde bulunması hedeflenmektedir [7].

Veri madenciliğinin temel konularını sınıflama, kümeleme ve birliktelik çözümlemeleri oluşturmaktadır. Yöntemin veri madenciliğine uygulanmasının sebebi ise, veri madenciliğinde,

özellikle sürekli veri içeren veritabanları kullanıldığında, veriler üzerinde kabuller yapılarak gruplara ayırtırmaktır.

Mühendislik ve bilimsel problemlerin büyük bir kısmına yumuşak hesaplama teknikleri uygulanmaktadır. Bu çalışmada teorisi ve uygulaması yapılan hesapsal yöntem de bu problemlerin büyük bir kısmına uygulanabilecektir. Bundan dolayı birçok bilim dalında, araştırmacılara yeni bir çözüm yöntemi olacaktır.

Bilgisayar mühendisliği ve biyomühendislik bilim dallarında veri miktarının artması sonucunda uzmanın yararlı bilgiyi çıkarması zorlaşmaktadır. Bu yararlı bilgiyi çıkarma yöntemlerinden biri de veri madenciliği konusudur. Özellikle, veri madenciliğinin alt dallarından birliktelik kuralları ve sınıflandırma kuralları veri madenciliğinde sürekli ve belirsiz veriler de içeren veritabanlarında etkili, ilginç, anlaşılabilir kuralların bulunması işi bu yeni yöntemle yapılacaktır. Bu yöntem ile elde edilecek olan sonuçlar, diğer yöntemlerin (genetik algoritmalar, fidan gelişim algoritması vb.) aynı veritabanlarında elde ettiği sonuçlar ile kıyaslandı.

2. VERİ MADENCİLİĞİ

Bilgisayar başta olmak üzere bilgi teknolojilerinin hızlı gelişiminin olumlu etkisiyle “Bilgi Çağı” olarak tanımlanan günümüz dünyasında karmaşık formatlardaki büyük veri yığınları içinden değerli bilgi parçalarını ayıklamak ve kullanılır hale getirmek “Veri Madenciliği” işlemleri ile olmaktadır. Bilgisayar mühendisliği ve biyomühendislik bilim dallarında veri miktarının artması sonucunda uzmanın yararlı bilgiyi çıkarması zorlaşmaktadır. Bu yararlı bilgiyi çıkarmada veri toplanması veya üretilmesi için sinyali doğru algılama; verileri doğru sınıflandırma ve kümeleme ile çok boyutlu ilişkilerinin ortaya konularak analizi çok önemlidir [8,9].

2.1. Veritabanı

Veritabanı (database), herhangi bir konuda birbirleri ile ilişkili verilerin sistematik olarak oluşturduğu yapılardır. Günümüzde eğer herhangi bir alanda veriler sistematik olarak saklanacaksa mutlaka veritabanı sistemlerine ihtiyaç duyulur. Basit bir iş yerindeki müşteri bilgilerinin saklandığı uygulamalardan tutun da çok büyük firmaların verilerinin tutulduğu veritabanı uygulamaları bulunmaktadır. Örneğin marketlerde ürün stoklarının saklandığı, bankalarda müşteri bilgilerinin, okullardaki öğrencilere ait verilerin saklandığı, hastanelerdeki hasta, personel gibi verilerin tutulduğu yapıların hepsi bir veritabanı sistemine ihtiyaç duyar [10].

Veritabanı kendisini tanımlayabilen, kendisi hakkında bilgi içeren, birbirleri ile ilişkili kayıtlardan oluşan bir koleksiyondur [11]. Bir veritabanı, veriler ve kendi yapısını açıklayan metadata denilen iki bölümden oluşur. Metadata, veritabanı içinde veritabanının yapısını açıklayan veridir. Metadata verilerin nasıl tutulacağı, nasıl sıralanacağı ve verilerin birbirleri ile ilişkilerinin nasıl olacağı gibi tanımlardan oluşur [12].

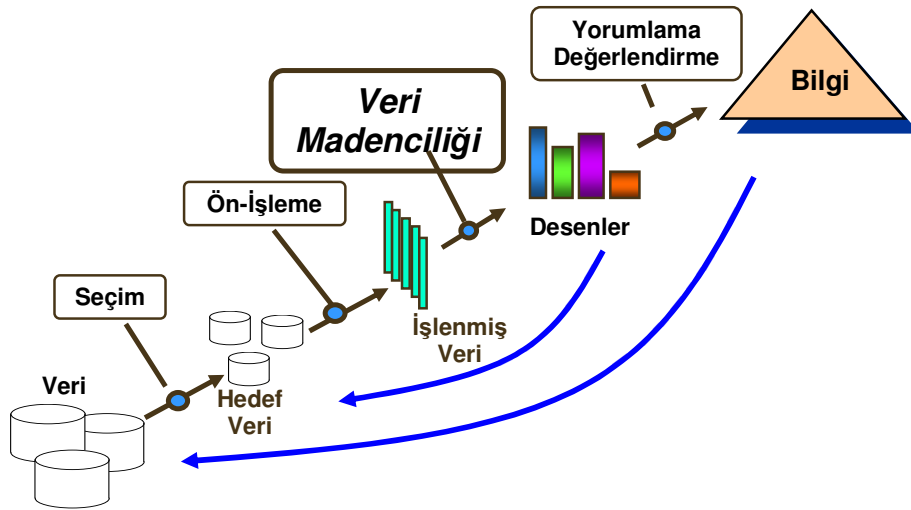
2.2. Veri Madenciliği

Veri madenciliği; veri tabanları, istatistik ve yapay öğrenme konularının kavramlarına dayanır ve onların tekniklerini kullanır. Büyük hacimli veri içerisinden; anlamlı, gizli kalmış ve karar destek sistemi için potansiyel olarak faydalı olabilecek, uygulanabilir bilgilerin (knowledge) çıkarıldığı ve geri planında istatistik, yapay zeka ve veritabanı yönetim sistemlerinin bulunduğu veri analiz tekniğine Veri Madenciliği (Data Mining) adı verilir. Veri

madenciliğinin temel konularını sınıflandırma, kümeleme ve birliktelik çözümlemeleri oluşturmaktadır.

Veri madenciliği, organizasyonların karar aşamaları için yeni bilgiler üreten yada gelecekle ilgili tahminler ve planlar yapmamızı sağlayan bir dizi teknikler ve anlayışlar bütünü olarak tanımlanmaktadır [13]. Temel olarak bilgisayar destekli bir bilgi çözümleme işlemidir. Yani sahip olduğumuz veri kümesinden işimize yarayacak yararlı bilgiyi üretme yöntemidir.

Karar verme fonksiyonunun yerine getirilmesinde sağlam ve güvenilir bilgilere gereksinim duyulur. Karar aşamalarında çok kritik bazı bilgiler vardır ki, sonuçların etkileri bu bilgilerin doğruluğuyla orantılıdır. Bir çok durumda cevabını tam olarak veremediğimiz sorular doğrultusunda karar verebiliriz. Müşterilerimizin ilgi alanları, bize karşı olan bakış açıları, rakip firmalara olan ilgileri, markalarımıza olan bağlılıkları, gelir düzeyleri gibi bilgiler onlara sağladığımız mal ve hizmetlerin kalitesi üzerinde çok net etkiler yapacaktır. Bu tür bilgiler teorik olarak her ne kadar sistemlerimizde kayıt altında olsa da, karar vermede kullanılabilir bir şekilde açık ve net cevaplara ulaşabilmemiz mevcut kullanımdaki sistemlerle çok zordur. Özellikle çok büyük boyutlardaki verilerle karşılaşıldığında, bu veriler içinden anlamlı ve yararlı bilgiyi ortaya çıkarmak ciddi bazı zorluklar taşımaktadır. Geleneksel istatistik yöntemlerle bu tür büyük boyutlardaki verileri çözümlemek çoğunlukla olanaksızdır. Bu tür büyük verileri işlemek için veri madenciliğinin özel yöntemlerine başvurmak söz konusudur.



Şekil 2.1. Veritabanlarında bilgi keşfi süreci [15].

Veri kendi başına değer ifade edebilmesi için amacımız doğrultusunda bilgiye dönüştürülmelidir. Veriyi bilgiye çevirmeye veri analizi denir.

Bilgi de; çözüm üretmek, karar vermek ve/veya bir soruya yanıt vermek için veriden çıkardığımız olarak tanımlanabilir.

Çok büyük veri yığınları altında saklı olan bu bilgilere ulaşmak için uzun yıllar boyu yapıla gelen çalışmaların neticesinde bir dizi metodoloji geliştirilmiştir. Veri madenciliği uzun yıllardır, özellikle batı ülkelerinde üzerinde çalışılan bir konu olmasına rağmen, gerçek hayatta yazılım endüstrisinin son yıllarda üretmiş olduğu ileri teknoloji ürünü yazılımlar ile kullanılmaya başlanmıştır [14].

Veri kalitesine ilişkin bilgiler, bu verilerin belli bir uygulama alanında kullanılabilmesi için uygun olup olmadığının belirlenmesinde kullanıcılara yardımcı olur. Bu sebeple, büyük maliyet ve uzun sürelerde toplanan verilerin mümkün olduğunca yüksek kalite düzeyinde tutulmasına özen gösterilmelidir.

Hedef kitleye doğru şekilde ulaşılarak mesajın doğru verilmesi için geçmiş dönemlere ait bilgilerin veri madenciliği tekniği kullanılarak analiz edilmesi ve yorumlanması gerekmektedir. Geleceğe dönük planlamaların buna göre yapılması, gereksiz zaman ve kaynak israfının önüne geçecektir.

2.3. Veri Temizleme

Veri tabanlarında eksik verilerle veya uygun olmayan verilerin oluşturduğu tutarsız verilerle karşılaşılabilir. Veritabanında yer alan tutarsız ve hatalı veriler **gürültü** olarak değerlendirilmektedir.

ADI SOYADI	KAYIT TARİHİ
Ali Akarsu	?
Ayşe Güler	11.09.2007

Çizelge 2.1. Gürültülü veriler barındıran örnek veritabanı

Bu gibi durumlarda verinin söz konusu sorunlardan temizlenmesi gerekecektir. Bunun için:

- Eksik değer içeren kayıtlar veri kümesinden atılabilir.
- Kayıp değerlerin yerine bir genel sabit kullanılabilir.
- Değişkenin tüm verileri kullanılarak ortalaması hesaplanır ve eksik değer yerine bu değer kullanılabilir.
- Değişkenin tüm verileri yerine, sadece bir sınıfa ait örneklerin değişken ortalaması hesaplanarak eksik değer yerine kullanılabilir.
- Verilere uygun bir tahmin yapılarak, örneğin regresyon yada karar ağacı modeli kurularak eksik değer tahmin edilebilir ve eksik değer yerine kullanılabilir.

2.4. Veri Bütünleştirme

Veri bütünleştirme, farklı veritabanlarından yada veri kaynaklarından elde edilen verilerin birlikte değerlendirmeye alınabilmesi için farklı türdeki verilerin tek türe dönüştürülmesi işlemidir.

2.5. Veri İndirgeme

Veri madenciliği uygulamalarında bazen çözümleme işlemi uzun süre alabilir. Eğer çözümlemeden elde edilecek sonucun değişmeyeceğine inanılıyorsa veri sayısı yada değişkenlerin sayısı azaltılabilir. Veri indirgeme çeşitli biçimlerde yapılabilir:

- Veriyi birleştirme
- Boyut indirgeme
- Veri sıkıştırma
- Örneklem
- Genelleme

2.6. Veri Dönüştürme

Veriyi bazı durumlarda veri madenciliği çözümlemelerine aynen katmak uygun olmayabilir. Değişkenlerin ortalama ve varyansları birbirlerinden önemli ölçüde farklı olduğu takdirde büyük **ortalama** ve **varyansa** sahip değişkenlerin diğerleri üzerindeki baskısı daha fazla olur ve onların rollerini önemli ölçüde azaltır.

Ayrıca değişkenlerin sahip olduğu **çok büyük ve çok küçük değerler** de çözümlemelerin sağlıklı biçimde yapılmasını engeller. Bu nedenle bir dönüşüm yöntemi uygulayarak söz konusu değişkenlerin normalleştirilmesi veya standartlaştırılması uygun bir yol olacaktır.

2.7. Min - Maks Normalleştirilmesi

Verileri 0 ile 1 arasındaki sayısal değerlere dönüştürmek için **min-maks normalleştirme** yöntemi uygulanır.

$$X^* = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

2.8. Z - değeri Standartlaştırma

Bu yöntem, verilerin ortalaması ve standart hatası göz önüne alınarak yeni değerlere dönüştürülmesi esasına dayanmaktadır.

$$X^* = \frac{X - \bar{X}}{\sigma_x}$$

Burada X^* dönüştürülmüş değerleri, X gözlem değerlerini, $\bar{X}$ verilerin aritmetik ortalamasını ve σ_x gözlem değerlerinin sapmasını ifade etmektedir.

2.9. Veri Madenciliği Uygulama Alanları

Veri madenciliği, biyomühendislik ve DNA veri analizlerinde, finans veri analizlerinde, perakende satış verilerinde, telekomünikasyon endüstrisinde, astronomi vb. bir çok alanda uygulanmaktadır [14]. Bunlardan bazıları aşağıdaki gibi sıralanabilir:

- Hastanelere yapılan tedavi taleplerinin bölgelere, zamana ve ihtiyaca göre değerlendirmesi, salgın hastalık riskinin ilk aşamada tespiti, kontrolü ve kaynak planlaması açısından faydalı olur.
- Kaçak enerji kullananların profillerini tespit eden bir model, olası kaçak enerji kullanıcılarını tahmin etmeyi sağlayacak, düşük maliyet ile kaçaklarla etkin mücadele edilmesine olanak tanıyacaktır.
- Karayollarının bölgelere ve zamana göre yoğunluklarını öngörme amaçlı bir çalışma doğru zamanda doğru kaynak planlaması ile örneğin kaza oranlarının asgariye indirilmesini sağlayacaktır.
- Kamu kurumları destek programlarını uygularken, verilecek desteğin doğru miktarda ve doğru hedefleri olan kuruluşlara verilmesini sağlayacak kurumsal risk skorlaması yapılmasıyla uygulanan programların başarısı artar.
- Kredileri tahsis ederken ödememe riski olan profillerin tespit edilmiş olması batık kredi miktarlarını azaltır.

2.10. Tıpta Veri Madenciliği

Tıpta veri madenciliğinden faydalanılarak, akciğerdeki tümörün iyi huylu olup olmadığına dair, karar destek amaçlı bir çalışma yapılmıştır. İstatistiklere göre Amerika’da yılda 160.000’den fazla akciğer kanseri vakasının olduğu ve bunların % 90’ının öldüğü belirlenmiştir.

Bu noktada bu tümörün erken ve doğru teşhisi önem kazanmaktadır. Testlerle elde edilen veriler sayesinde % 40–60 nispetinde doğru teşhis konabilmektedir. İnsanlar, kanser olup olmadıklarından emin olmak için biyopsi yaptırmayı (gerekli olmasına rağmen) tercih etmemektedir. Biyopsi gibi testlerin hem maliyeti yüksektir, hem de bunlar çeşitli riskler taşımaktadır. Farklı yerlerde ve farklı zamanlarda kliniklerde toplanan bu türden test verileri arasında yapılan veri madenciliği çalışmaları teşhiste % 100 veya % 100'e çok yakın oranda doğruluk sağlamıştır.

Başka bir örnek ise, Kore Tıbbî Sigorta Kurumu tarafından hazırlanan bir veritabanı üzerinde yapılan yüksek tansiyon ile ilgili bir çalışmadır. Bu çalışma 1998 yılına ait 127.886 kayıt üzerinde yapılmıştır. İlk aşamada yüksek tansiyona sahip 9.103 kayıt, daha sonra aynı sayıda yüksek tansiyonu olmayan kayıtlar üzerinde çalışılmıştır. Bu örnek 13.689 kayıttan oluşan öğrenme ve 4.588 kayıttan oluşan test setine bölünerek modelin eğitimi yapılmıştır. Bu çalışmalar neticesinde yüksek tansiyon tahmininde dikkate alınan değerler vücut kitle indeksi (BMI), idrar proteini, kan glikozu ve kolesterol değerleridir. Hayat şartlarının (diyet, alınan tuz miktarı, sigara kullanımı vb.) hiçbirinin bu tahminde rolü olmadığı, ayrıca grafiğe dökülen değerlerde de yalnızca yaşın önem arz ettiği gösterilmiştir.

Tıbbi veritabanlarında veri madenciliği ve bilginin bulunması, diğer türdeki veritabanlarındakinden çok farklı değildir. Ancak, tıbbi veride diğer veri türlerinde olmayan bazı özellikler vardır. Bunlar :

1. Çok sayıda yordam, görüntülemeyi bir tanı aracı olarak kullanmaktadır. Bu nedenle, görüntülerden oluşan veritabanlarında etkin bir veri madenciliği gerçekleştirebilmek için yöntemler geliştirmek gerekmektedir. Bu da sayısal veritabanlarındaki veri madenciliğinden hem daha farklı, hem de daha zordur.
2. Tıbbi veritabanları, her zaman heterojendir. Örneğin, bir organa ait görüntü, her zaman, doktorun yorumu (klinik izlenim, tanı) gibi, başka klinik bilgilerle bir aradadır. Bu ise, bu tür verilerin çözümlenmesi için yeni araçlar ve yüksek kapasiteli veri depolama aygıtları gerektirir.
3. Doktorlar, görüntüler, sinyaller yada diğer klinik bilgilerle ilgili yorumlarını, standartlaştırılması çok güç olan serbest metinler olarak yazmaktadır. Bazen aynı hastalık açıklanırken bile farklı adlar kullanılmaktadır. Tıbbi kavramlar arasındaki ilişkileri açıklamak için de farklı dilbilgisi yapıları kullanılmaktadır.
4. Tıptaki temel veri yapıları, fiziksel bilimlerin birçok alanıyla karşılaştırıldığında, matematiksel olarak karakterize edilmeye pek uygun değildir. Veri madencisinin bilgiyi düzenleyebileceği, sınıflandırma, kümeleme, modelleri yada dizi çözümlmeleri gibi karşılaştırılabilir yapılar yoktur.

2.11. İlişki Kuralları

Veritabanı içinde yer alan kayıtların birbirleriyle olan ilişkilerini inceleyerek, hangi olayların eş zamanlı olarak birlikte gerçekleşebileceklerini ortaya koymaya çalışan veri madenciliği yöntemleri bulunmaktadır. Bu ilişkilerin belirlenmesiyle “**ilişki (birliktelik) kuralları**” elde edilir.

Bu çözümleme, veritabanındaki alanlar arasındaki ilişkilerin (bağımlılıkların) belirlenmesiyle ilgilidir.

İlişki kuralları; geçişlerin veritabanındaki bulunan nesneler arasında düzensiz korelasyonlarını (iki bağımsız değişken arasındaki doğrusal ilişkinin yönünü ve kuvvetini) keşfetmek için kullanılabilir.

Örnek olarak bir market müşterilerinin alışveriş alışkanlıklarını belirleyelim. Beş müşterinin alışveriş sepet bilgileri Çizelge 2.2’deki gibi olsun.

MÜŞTERİ	ALIŞVERİŞ SEPETİNDEKİ ÜRÜNLER
1	Makarna, Meyva Suyu, Peynir, Yağ
2	Makarna, Ketçap
3	Bira, Ketçap, Meyva Suyu
4	Makarna, Ketçap, Meyva Suyu, Yağ
5	Bira, Makarna, Ketçap, Yağ

Çizelge 2.2. Alışveriş sepet bilgileri.

Bu verilerden yararlanarak ilişki (birliktelik) çözümlemeleri yapılır. Apriori algoritması ile aşağıdaki sonuçlar elde edilir.

$\{\text{Ketçap, Meyva Suyu}\} \rightarrow \{\text{Yağ}\} \quad (s=0.4, c=1.0)$
 $\{\text{Ketçap, Yağ}\} \rightarrow \{\text{Meyva Suyu}\} \quad (s=0.4, c=0.67)$
 $\{\text{Yağ, Meyva Suyu}\} \rightarrow \{\text{Ketçap}\} \quad (s=0.4, c=0.67)$
 $\{\text{Meyva Suyu}\} \rightarrow \{\text{Yağ, Ketçap}\} \quad (s=0.4, c=0.67)$
 $\{\text{Yağ}\} \rightarrow \{\text{Ketçap, Meyva Suyu}\} \quad (s=0.4, c=0.5)$
 $\{\text{Ketçap}\} \rightarrow \{\text{Yağ, Meyva Suyu}\} \quad (s=0.4, c=0.5)$

Buradaki satırları şöyle yorumlanır:

Ketçap ve Meyve Suyunu birlikte satın alan müşteriler mutlaka Yağ da alıyorlar.
Ketçap ve Yağı birlikte alanlar %67 olasılıkla Meyve Suyunu da alıyorlar.
Yağ ve Meyve Suyunu birlikte satın alan müşteriler %67 olasılıkla Ketçap da alıyorlar.
Meyve Suyu satın alan müşteriler %67 olasılıkla Ketçap ve Yağ da alıyorlar.
Yağ satın alan müşteriler %50 olasılıkla Ketçap ve Meyve Suyu da alıyorlar.
Ketçap satın alan müşteriler %50 olasılıkla Yağ ve Meyve Suyu da alıyorlar.

2.12. Sınıflandırma

“Genç kadınlar küçük araba satın alır; yaşlı, zengin erkekler ise büyük, lüks araba satın alır.”

Amaç bir malın özellikleri ile müşteri özelliklerini eşlemektir. Böylece bir müşteri için ideal ürün veya bir ürün için ideal müşteri profili çıkarılabilir. Örneğin bir otomobil satıcısı şirket geçmiş müşteri hareketlerinin analizi ile yukarıdaki gibi iki kural bulursa genç kadınların okuduğu bir dergiye reklâm verirken küçük modelinin reklâmını verir.

Sınıflandırma (classification), veri madenciliğinde sıkça kullanılan bir yöntem olup, veri tabanlarındaki gizli kalıpları ortaya çıkarmakta kullanılır.

Sınıflandırma, ön tanımlı birkaç sınıfın içine bir veri nesnesini yerleştirmek için kullanılan bir tekniktir. Sınıflandırıcının başarımında iyi belirlenmiş özellikler kilit rolü oynarlar [16].

Veri kümesi içinde ortak özellikleri yada farklılıkları ortaya koyacak biçimde sınıflandırma yapılabilir.

Sınıflandırma bir **öğrenme algoritmasına** dayanır. Tüm veriler kullanılarak eğitime işi yapılmaz. Bu veri topluluğuna ait bir örnek veri üzerinde gerçekleştirilir.

Öğrenmenin amacı, bir **sınıflandırma modelinin** oluşturulmasıdır. Bir başka deyişle sınıflandırma, hangi sınıfa ait olduğu bilinmeyen bir kayıt için bir sınıf belirleme sürecidir.

2.12.1. Sınıflandırmanın Temel Kuralları

- Öğrenme eğitilidir.
- Veri setinde bulunan her örneğin bir dizi niteliği vardır ve bu niteliklerden biri de sınıf bilgisidir.
- Hangi sınıfa ait olduğu bilinen nesneler (öğrenme kümesi - training set) ile bir model oluşturulur.
- Oluşturulan model öğrenme kümesinde yer almayan nesneler (deneme kümesi - test set) ile denenerek başarıları ölçülür.

2.12.2. Sınıflandırma Yöntemleri

- Karar Ağaçları (Decision Trees)
- Örnek Tabanlı Yöntemler (Instance Based Methods): k en-yakın komşu (k nearest neighbor)

- Bayes Sınıflandırıcı (Bayes Classifier)
- Yapay Sinir Ağları (Artificial Neural Networks)
- Genetik Algoritmalar (Genetic Algorithms)
- Cizge Bölütleme

2.12.3. Sınıflandırma Aşamaları

Verilerin sınıflandırılması süreci iki adımdan oluşur.

1. Adım : Model Kurma Süreci

2. Adım : Modelin Uygulanma Süreci

2.12.3.1. Model Kurma Süreci

İlk adım, önceki tanımlanmış veri kümelerine uygun bir modelin ortaya konulmasıdır. Söz konusu model, veritabanındaki kayıtların **nitelikleri** (*attribute*) veya bir başka deyişle **alan isimleri** (*sütun isimleri*) kullanılarak gerçekleştirilir. Sınıflandırma modelinin elde edilmesi için veritabanının bir kısmı **eğitim verileri** olarak kullanılır. Bu veriler veritabanından rasgele seçilir.

2.12.3.2. Modelin Uygulanma Süreci

Test verileri üzerinde sınıflandırma kuralları bu şekilde belirlendikten sonra söz konusu kurallar yeni verilere uygulanarak amaçlanan sonuçlar elde edilir.

2.12.4. Sınıflandırma Modelini Değerlendirme

Sınıflandırma Metodu tarafından oluşturulan modelin başarısını ölçmek için;

- Doğruluk (Accuracy)
- Hata Oranı (Error rate)
- Özgünlük (Specificity)
- Duyarlık, hassasiyet (Sensitivity)

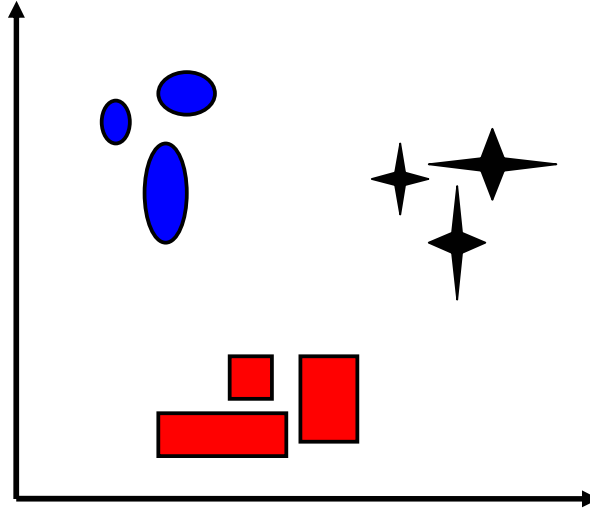
gibi ölçüler kullanılır.

Tıbbi verilerde sınıflandırma; doktorların tecrübesi ile birleştirildiğinde kronik teşhislerde doktorlara zamandan çok büyük kazanç sağlayabilecektir.

2.13. Kümeleme

Küme, birbirine benzeyen nesnelerden oluşan gruptur.

- Aynı kümedeki örnekler birbirine daha çok benzer
- Farklı kümedeki örnekler birbirine daha az benzer



Şekil 2.2. Kümeleme.

Kümeleme; verilerin kendi aralarındaki benzerliklerin göz önüne alınarak gruplandırılması işlemidir. Bu özelliği nedeniyle pek çok alanda uygulanabilmektedir.

Kümeleme; bir eğiticişiz öğrenme ile gerçekleştirilir.

Kümeleme; iki gözlemin/verinin benzerlikleri (yakınlıkları) veya benzemezlikleri (uzaklıkları) temel alınarak yapılır. Amaç, elemanların birbirlerine çok benzediği, ancak özellikleri birbirlerinden çok farklı olan kümelerin bulunması ve veritabanındaki kayıtların bu farklı kümelere (gruplara) bölünmesidir. Diğer bir ifade ile yığın halinde ve karmaşık olan bilgiler içinden benzer olanlarını, bilgilerinin daha anlaşılır olmasının sağlanması bakımından bir başlık altında toplamayı amaçlar.

Kümeleme; benzer karakteristikler ile veri sayfaları veya kullanıcıları birlikte gruplamak için bir tekniktir. Bu teknikler, çeşitli modeller arasından, verinin gösterilmesine en “uygun” olanını bulmak olarak düşünülebilir.

İnsan algılamasıyla ilgisiz veri durumlarında kümeleme problemi çok önemli bir problemdir [17]. Kümeleme problemi özellikler kümesi ile belirlenmiş bir deseni giriş olarak almayı ve çıkış sınıflarından hangisine en iyi uyduğuna karar vermeyi gerektirir [18].

Kümeleme; tahmin ve ilişki kuralları oluşturmak için kullanılabilir.

Örnek: Çizelge 2.3’deki gözlem değerlerini gözönüne alalım. Bu gözlem değerlerinin X1 ve X2 gibi iki değişkeni bulunmaktadır. Bu gözlem değerlerine dayanarak verilerdeki kümelenmeleri ortaya koymak istiyoruz. Kümeleri ortaya koymak üzere birçok veri madenciliği ve istatistiksel yöntem bulunmaktadır. Söz konusu verilere hiyerarşik kümeleme yöntemlerinden “en yakın komşu algoritması”nı uyguladığımızda Çizelge 2.4’de belirtilen kümeler elde edilir:

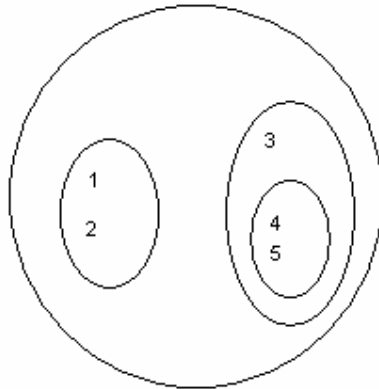
GÖZLEM	X1	X2
1	1	1
2	2	1
3	4	5
4	7	7
5	5	7

Çizelge 2.3. Kümeleme işlemi uygulanacak veri.

	KÜMELER
Küme 1	(1,2)
Küme 2	(4,5)
Küme 3	(3,4,5)
Küme 4	(1,2,3,4,5)

Çizelge 2.4. Algoritmanın uygulanması sonucunda elde edilen kümeler.

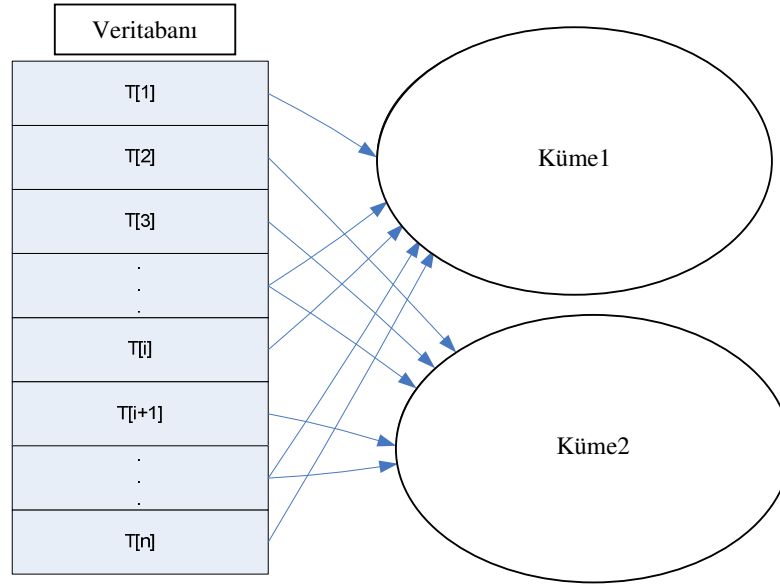
Söz konusu kümeler daha açık bir şekilde aşağıdaki gibi de gösterilebilir. Küme içindeki rakamlar gözlem numaralarını belirtmektedir.



Şekil 2.3. Hiyerarşik kümelerin görünüşü.

2.13.1. Kümeleme Yöntemleri:

- K-Means Kümeleme
- Hiyerarşik Kümeleme
- Yapay Sinir Ağları
- Genetik Algoritmalar
- Çizge Bölütleme



Şekil 2.4. Bir veritabanının kümelenmesi.

3. ÇİZGELER

Çizgeler, bir çok bilimsel problemin çözümünde; özellikle sınıflandırma ve kümelemede kullanılabilir. Günlük hayatta karşılaşılan karışık veya karmaşık problemlerin analizi sonucunda ortaya çıkarlar. Çizgelerin etkili olarak test veya analiz edilmesi için ilk adım olarak problemin çizgesel özellikleri ortaya çıkarılır. Problemin çözümü için de çizgenin ya özellikleri test edilir yada çizgenin analizi yapılır.

Çizgeler, oluşlar ve bu oluşlar arasındaki ilişkilerin grafiksel olarak gösterilmesinde kullanılmaktadır [19]. Özellikle tıp ve mühendislik olmak üzere çoğu bilimsel problem, bu yöntem ile modellenerek ifade edilebilmektedir. Bu şekilde problemin çizge ile modellenmesi üzerine geliştirilmiş etkili algoritmalar olduğu gibi yaklaşım ile problemlere çözüm bulan algoritmalar da geliştirilmiştir. Yaklaşım algoritmalarının geliştirilmesinin sebebi ise, bu tip problemlerin etkili algoritmasının günümüze kadar geliştirilememiş olmasıdır.

Çizge üzerinde yapılan işlemlerin ve algoritmalarının anlatımına geçmeden önce çizge kuramı ve çizgenin bilgisayar ortamında ifade edilmesi konularının açıklanması gerekmektedir.

3.1. Çizgelerde Temel Kavramlar: Ayrıtlar ve Dügümler

Çizgeler, iki sonlu kümeden oluşur. Bunlardan biri sonlu düğümler kümesi, diğeri ise bu düğümler arasındaki ayrıtlar kümesidir. Genel olarak çizge $G=(V,E)$ ile gösterilir. $V=\{v_1, v_2, \dots, v_n\}$ düğümler kümesi ve $E=\{e_1, e_2, \dots, e_m\}$ ($E \subseteq V \times V$) ayrıtlar kümesidir.

Eğer $e_i=(v_j, v_{j+1}) \in E$ ve $(v_{j+1}, v_j) \notin E$ ise bu tip ayrıtlara yönlendirilmiş ayrıt denir ve $e_i=(v_j, v_{j+1})$ şeklinde gösterilir. Aynı zamanda v_j ve v_{j+1} düğümleri birbirine komşu (bitişik) düğümlerdir ve $ADJ(v_j)=v_{j+1}$, $ADJ(v_{j+1})=v_j$ olur. Eğer bir çizgenin ayrıtlarının hepsi yönlendirilmiş ise bu çizgeye yönlendirilmiş çizge denir. Eğer e_i gibi bir ayrıt yönlendirilmemiş ayrıt ise $e_i=(v_j, v_{j+1}) \in E$ ve $e_i=(v_{j+1}, v_j) \in E$ olur. Yönlendirilmemiş ayrıtlar $e_i=\{v_j, v_{j+1}\}$ veya $e_i=\{v_{j+1}, v_j\}$ şeklinde gösterilir. Eğer bir çizgenin bütün ayrıtları yönlendirilmemiş ise bu tip çizgelere yönlendirilmemiş çizge denir. Tez içerisinde, aksi belirtilmediği sürece yönlendirilmemiş çizge yerine çizge terimi kullanılacaktır. $e_i=\{v_j, v_{j+1}\}$ ayrıtı için, v_j ve v_{j+1} düğümleri e_i ayrıtının uç noktalarıdır.

$e_i=\{v_j, v_{j+1}\}$ olsun. Eğer $v_j=v_{j+1}$ ise e_i 'ye döngü (tek çevre) denir. $e_i=\{v_j, v_{j+1}\}$ ve $e_{i+1}=\{v_k, v_{k+1}\}$ olsun; eğer $v_k=v_j$, $v_{k+1}=v_{j+1}$ ve $v_k \neq v_{j+1}$, $v_{k+1} \neq v_j$ ise e_i ve e_{i+1} ayrıtlarına paralel (koşut bağlı) ayrıtlar denir. Uç noktaları farklı olan ayrıtlara tek ayrıt denir. Bir düğüme yalnız

iki ayrıt bağlı ve bu ayrıtların öbür uç düğümleri birbirlerinden farklı ise bu ayrıtlara dizi bağlı ayrıtlar denir.

Bir çizgenin düğümler kümesinin eleman sayısı o çizgenin mertebesini verir. $G=(V,E)$ için eğer $|V|=n$ ise G çizgesinin mertebesi n 'dir. v gibi bir düğüme giren ve çıkan ayrıt sayısına o düğümün derecesi (kertes) denir ve $d(v)$ ile gösterilir.

$v_i \in V$ için eğer $d(v_i)=0$ ise v_i düğümü izole edilmiş bir düğümdür veya bu düğüme tek düğüm denir. Eğer $d(v_i)=1$ ise bu tip düğümlere pendant veya uç düğüm denir.

$G=(V,E)$, bir yönlendirilmemiş çizge olsun. Eğer $V=\emptyset$ veya $V \neq \emptyset$ ve $E=\emptyset$ ise bu tip çizgelere boş çizge denir ve $\forall v_i \in V$ için $d(v_i)=0$ olur. Bir adet tek düğümden veya tek çevreden, veya tek ayrıttan oluşan çizgelere ilkel çizge denir. Eğer bir çizge döngü ve paralel ayrıt içermiyorsa bu tip çizgelere basit çizge (yalın çizge) denir. Sadece çevreleri (döngüleri) içeren çizgelere sözde çizge denir.

Bir çizgede, bir düğüme bağlı olan bütün ayrıtların oluşturduğu kümeye, o düğümün tanımladığı çakışım kümesi denir.

Herhangi bir çizge için, eğer bütün düğümlerin dereceleri eşit ise bu tip çizgelere düzenli çizge denir. Eğer bir çizge paralel ayrıtlar içeriyorsa bu tip çizgelere çoğul çizge denir. Her türlü ayrıtı içeren çizgelere karmaşık çizge denir.

Tam olan bir çizgede her düğüm diğer bütün düğümler ile komşudur ve (3.1) bağıntısı v_i düğümünün komşu düğümler kümesini vermektedir.

$$\forall v_i \in V, \text{Adj}(v_i) = \{v_1, v_2, \dots, v_{i-1}, v_{i+1}, \dots, v_{|V|}\} \quad (3.1)$$

Düzenli bir çizgede, eğer $\forall v_i \in V, d(v_i) = r$ ise du düzenli çizgeye r -düzenli çizge denir.

Eğer $G=(V,E)$ çizgesinin düğümler kümesi $V=V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$ ve E 'deki bütün ayrıtların bir ucu V_1 'de ve diğer ucu V_2 'de ise bu çizgeye iki parçalı çizge denir. İki parçalı ve tam olan çizgeler $K_{m,n}$ şeklinde gösterilirler ($|V_1|=m$ ve $|V_2|=n$).

Eğer $G=(V,E)$ çizgesi k -parçalı ise V kümesi k tane öyle parçalara $V_1, \dots, V_k$ bölünebilir ki, $V_i \cap V_j = \emptyset$ ve $i \neq j$. E içindeki bütün ayrıtların bir ucu V_i 'de ve diğer ucu V_j 'dedir, $i \neq j$. k -parçalı tam bir çizge, basit bir k -parçalı çizge olup $V = \{V_1, \dots, V_k\}$, $\forall v_i \in V_r, \forall v_j \in V_s, r \neq s, 1 \leq r, s \leq k, \{v_i, v_j\} \in E$ dir [19].

4. ÇİZGE BÖLÜTLEME YÖNTEMLERİ

Çizge bölütleme; bilimsel hesaplama, optimizasyon, VLSI tasarımı, yük dengeleme gibi alanlarda uygulanabilen önemli bir yöntemdir. Özellikle bazı mühendislik ve tıbbi problemlerde olmak üzere çizgelerle ifade edilebilecek bilimsel hesaplamalarda; problemlerin çözümünün daha az zaman alması için problem veri kümesinin ayrık olan alt kümelere parçalanması gerekir. Bunu gerçekleştirmek için problemi temsil eden çizgenin bölütlenmesi gerekmektedir. Çizge bölütleme, araştırma alanlarında ve paralel hesaplamalarda çok bilinen bir yöntemdir. VLSI devre tasarımında, yük dengelemede, seyrek matrislerin düzenlenerek mühendislik ve optimizasyon uygulamalarında ortaya çıkan sistemlerin doğrudan çözümünün elde edilmesinde kullanılır.

Çizgeler üzerinde çeşitli işlemler yapılabilmektedir. Bu işlemlerden biri de çizge bölütlemesidir. Bazı bilimsel ve mühendislik problemlerinin çözümünde veriler işlemcilerle öyle bir şekilde dağıtılır ki işlemcilerle düşen yük dengelidir ve işlemciler arası iletişim en aza indirgenmiştir. Bu, bir çizge bölütleme işlemidir. Çizgedeki düğümler yapılacak işi ve düğümler arası ayrıtlar da bu işlerin atandığı işlemciler arasındaki iletişimi gösterir. Optimal bölütleme, işlemcilerle düşen işin mümkün mertebe dengeli ve işlemciler arasındaki iletişimin de minimuma indirgenmesidir. Bu da iletişim ve işlerin yapılması için harcanacak zamanın minimuma indirgenmesini netice verir.

Problemi oluşturan veriler, işlem safhalarında değişmiyorlarsa; bu durumda bölütleme işlemine başlamadan önce çizge oluşturulur ve bu çizgenin bölütleme işlemi çeşitli algoritmalarla göre bir kez yapılır.

Eğer veriler işlem safhasında değişiyorlarsa, bu durumda bölütleme işlemi ilk verilere göre yapılır. Bundan sonra yeni veriler eklenince bölütleme yapılan çizgede bu eklemeye göre düzenleme yapılır veya veri silinmesi söz konusu ise benzer işlem bunun için de tekrar edilir. Bölütleme işlemi, verilerin durumuna göre süreklilik arz eden bir oluşturmaktır [19].

Çizge bölütleme işlemi NP-tam (NP-complete) bir problemdir. Bundan dolayı bölütleme işlemi yapan algoritmanın matematiksel karmaşıklığı (complexity) bir eksponansiyel bağıntıdır (polinom değil) [19]. Günümüze kadar bulunan algoritmalar, hiç biri verilen çizgenin optimum bölütleme işlemini yaptığını garanti edemiyor. Buna rağmen geliştirilen algoritmaların çoğu iyi veya iyiye yakın sonuçlar vermektedir.

Probleme göre verilen çizgenin düğümlerinin koordinatları olabilir veya olmayabilir. Eğer koordinatları varsa, bu durumda bölütleme işlemi biraz daha kolaylaşmaktadır. Çünkü düğümlerin eksen koordinatlarına göre ilk önce bir sıralaması yapılır, ondan sonra bölütleme

(böl ve yönet) işlemi yapılır. Elde edilen parça sayısı yetersiz ise bu parçalar yine aynı yöntem ile diğer eksen yani üzerinde sıralama işlemi yapılmamış veya en az yapılmış eksene göre sıralanır ve bu sıralama işleminden sonra bölütleme işlemi uygulanır. Genelde her parça kendi içinde sıralanır ve ondan sonra o parça ikiye bölütlenir.

Eğer düğümlerin koordinatları yoksa, bu durumda çizgenin bölütleme işlemi için genel olarak üç değişik yöntem uygulanır. Bunlardan biri geliş güzel başlangıç bölütleme işleminin yapılması, ikincisi çizgenin özdeğerleri ve bu özdeğerlere denk düşen özdeğer vektörlerini kullanarak bölütleme işleminin yapılması, üçüncü olarak da eğer çizgenin düğüm sayısı fazla ise ilk önce çizge üzerinde geliş güzel eşleştirme yapılır veya maksimum-ayrıt-önce metodu ile eşleştirme yapılır ve eşleşen düğümler bir düğüme indirgenir. Bu şekilde istenen sayıda düğüm elde edildikten sonra yukarıda belirtilen yöntemlerden biri kullanılarak bölütleme işlemi yapılır ve ondan sonra çizge geri açılarak orijinal çizgenin bölütlenmiş hali elde edilir (çok seviyeli çizge bölütleme metodu).

Bu algoritmalarından şu anda kullanılanların bir listesi:

1. Geliş güzel bölütleme yapan algoritmalar
 - a) Lineer
 - b) Gelişigüzel
 - c) Serpme
2. Spektral Çizge Bölütleme
3. Eylemsizlik
4. Basit Çizge İndirgeme
5. Çok Seviyeli Çizge Bölütleme

4.1. Bölüt Oluşturma Algoritmaları

Bölüt oluşturma algoritmaları çizgenin bölütleme işlemini yapan algoritmalarlardır.

4.1.1. Gelişigüzel Bölütleme Yapan Algoritmalar

Geliş güzel bölütleme algoritmalarının her birinin belli bir algoritmik yapısı vardır. Geliş güzel denilmesinin sebebi yük dengeleme şartını ve minimum kesim ayrıt boyutunu göz önüne almamalarından kaynaklanır.

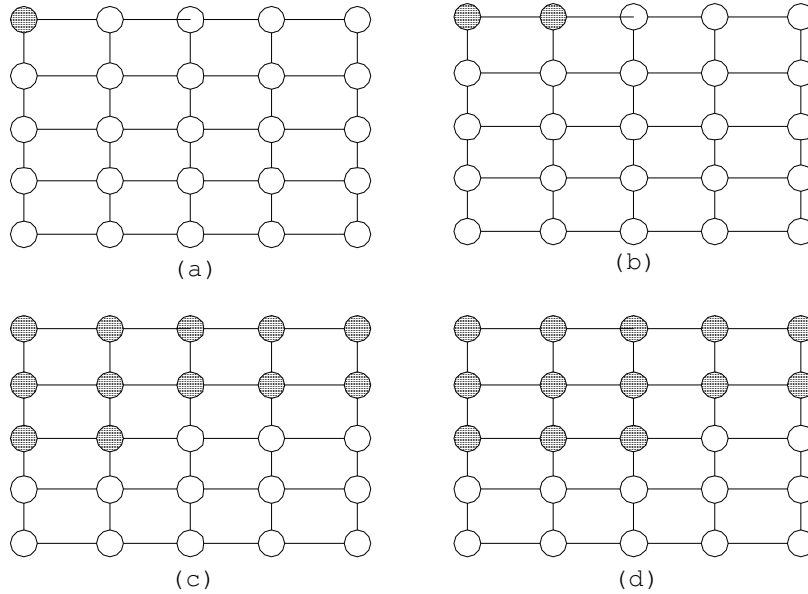
4.1.1.1. Doğrusal Algoritma

Bu algoritmaya göre çizge bir parça olarak alınır ve bu parçadan ilk önce bir düğüm alınır ve ikinci parça olarak kabul edilir. Eğer parçalardaki düğüm ağırlıkları eşit veya birbirine

çok yakın değilse toplam ağırlığı fazla olandan bir düğüm seçilir ve diğer parçaya eklenir, tekrar parçaların ağırlıkları kontrol edilir. Seçilecek olan düğümün en fazla bir tane komşu düğümünün diğer parça içinde olması gerekir. Elde edilen parçalara iyileştirme işlemi uygulanır. Lineer bir algoritmanın çalışma şekli, Şekil 4.1’de görülmektedir.

4.1.1.2. Gelişigüzel Algoritma

Gelişigüzel çizge bölütleme algoritmasında ise çizge gelişigüzel iki parçaya bölünür ve bu parçalara iyileştirme algoritmaları uygulanır. Bu yöntemin iyileştirme kısmı çok yavaş olabilir. Çünkü çizge gelişigüzel iki parçaya bölünmektedir ve bu parçalar bir birinden çok farklı parçalar olabilirler.



Şekil 4.1. Lineer çizge bölütleme algoritması (düğüm ve ayrıt ağırlıkları aynı olan 5x5 ağ çizgesi),
(a) bir parçada bir düğüm, diğer parçada ise geriye kalan bütün düğümler,
(b) bir parçada iki düğüm, diğer parçada ise geriye kalan bütün düğümler,
(c) birinci parçada $\lfloor n/2 \rfloor$ düğüm diğer parçada ise $\lfloor n/2 \rfloor + 1$ düğüm var,
(d) birinci parçada $\lfloor n/2 \rfloor + 1$ düğüm varken, diğer parçada ise $\lfloor n/2 \rfloor$ düğüm vardır [19].

4.1.1.3 Serpme Algoritması

Serpme algoritmasında ise, başlangıçta çizgenin her iki parçasını içerecek kümeler boş durumdadır. Bölütleme işlemi yapılırken, bir düğüm seçilir ve bu seçilen düğüm ağırlık toplamı

az olan kümeye atılır ve eğer her iki kümenin ağırlık toplamaları eşit ise bu durumda seçilen düğümün hangi kümede komşu düğüm sayısı fazla ise, o kümeye katılır ve eğer hem kümelerin ağırlık toplamaları ve hem de seçilen düğümün kümelerde ki komşu ayrıt sayısı eşitse, bu durumda, düğüm geliş güzel bir kümeye kayıt edilir. Bu işlemlerden sonra eğer her iki parça arasındaki fark büyükse iyileştirme işlemi uygulanabilir. Eğer her iki kümenin düğümlerin ağırlıkları toplamı birbirine denk ise ve $W(E_{kesim})$ değeri kabul edilecek kadar küçükse, bu bölütlemenin sonucuna iyileştirme işleminin uygulanmasına gerek kalmaz [19].

4.1.2. Spektral Çizge Bölütleme

Spektral çizge bölütleme yönteminde, numerik metotlar kullanılmaktadır. Bunlardan biri de reel-simetrik matrislerin özdeğerleri ve özdeğer vektörlerinin bulunmasıdır. Bir çizgenin Laplace matrisi reel ve simetrik olan bir matristir. Bu matrisin özdeğer ve özdeğer vektörleri bulunur. Bu matrisin ikinci en küçük özdeğere denk düşen özdeğer vektörü çizgenin iki parçaya bölütlenmesinde kullanılır. Bu yöntemi ilk olarak Fiedler kullanmıştır. Bu ikinci en küçük özdeğerin özdeğer vektörünü çizgenin cebirsel bağıllığı olarak adlandırmıştır. Fiedler'in bu çalışmasından dolayı, bu özdeğere matrisin Fiedler değeri ve özdeğer vektörüne de Fiedler vektörü denilmiştir. Bu vektör çizgenin en küçük ayırıcını verir.

Bu yöntem, bir çok nümerik algoritma için algoritmanın önemli bir parçasını oluşturmaktadır. Spektral bölütleme yöntemi, çizge ve matris bölütlemeye oldukça fazla kullanılmakla birlikte devre dizaynı ve simülasyonlarında da kullanılmaktadır.

$G=(V,E)$ bağlı ve yönlendirilmemiş bir çizge olsun. Çizgenin bölütlenmesi, verilen çizgenin düğümlerinin ayrık birden fazla alt kümeye bölütlenmesidir. Bu alt kümeler A ve $\bar{A}$ olsun ($A \cap \bar{A} = \emptyset$ ve $A, \bar{A} \subset V$). Başlangıçta $|A| \leq |\bar{A}|$ kabul edilmesi yöntemin genelliliğini bozmaz. $E(A, \bar{A})$ çizgenin ayıracı olsun. $(A, \bar{A})$ bölütlemenin kesme boyutu $|E(A, \bar{A})|$ olur ve buna çizgenin ayırıcının boyutu denir ve genelde bölütleme işlemlerinde bunun minimum olması istenir. Kesme oranı $\phi(A, \bar{A})$ aşağıdaki eşitlik ile hesaplanır.

$$\phi(A, \bar{A}) = \frac{|E(A, \bar{A})|}{\min(|A|, |\bar{A}|)} \quad (4.1)$$

Bir çizgenin beklenen en iyi kesme oranına o çizgenin izoperimetrik sayısı denir ve aşağıdaki eşitlik ile hesaplanır.

$$\phi(G) = \min_{|A| \leq n/2} \frac{|E(A, \bar{A})|}{|A|} \quad (4.2)$$

Bir çizgenin bölümleri arasında en fazla 1 düğüm fark varsa bu bölümlere o çizgenin eş iki parçaları denir.

4.1.2.1. Laplace Matrisi ve Fiedler Vektörleri

Bir çizgenin bitişiklik matrisi $|V|=n$ için, $n \times n$ boyutunda olan bir matristir. Eğer $(v_i, v_j) \in E$ ise matrisin (i, j) -elemanı 1 diğer durumlarda sıfırdır. Bir çizgenin derece matrisi de $n \times n$ boyutunda bir kare matristir. (i, i) -elemanı v_i düğümünün derecesine eşit, diğer elemanlar ise sıfırdır. Çizgenin bitişiklik matrisi ile derece matrisinin farkı alınarak Laplace matrisi elde edilir. Laplace matrisi $L(G)=(l_{ij})$, $i, j=1, \dots, n$ 'dir.

$$l_{ij} = \begin{cases} 1 & \text{eğer } \{v_i, v_j\} \in E \\ -\deg(v) & \text{eğer } i = j \\ 0 & \text{diğer durumda} \end{cases} \quad (4.3)$$

Laplace matris özdeğerlerinden biri 0 ve bu özdeğere karşılık bulunan özdeğer vektörü ise bütün elemanları 1 olan bir vektördür. Çünkü Laplace matrisi pozitif yarı tanımlı bir matristir ve bundan dolayı diğer bütün özdeğerleri negatif olmayan değerlerdir. Bundan sonraki işlemlerde Laplace matrisinin ikinci en küçük özdeğere denk düşen özdeğer vektörü bulunur. Daha öncede belirtildiği gibi bu özdeğere Fiedler değeri ve bu özdeğere karşılık bulunan özdeğer vektörüne de Fiedler vektörü denir.

Her hangi bir $\vec{x} \in \mathbb{R}^n$ vektörü için

$$\vec{x}^T L(G) \vec{x} = \sum_{(i,j) \in E} (x_i - x_j)^2 \quad (4.4)$$

ve bununla birlikte, G çizgesinin Fiedler değeri de:

$$\lambda_2 = \min_{\vec{x} \perp (1, 1, \dots, 1)} \frac{\vec{x}^T L(G) \vec{x}}{\vec{x}^T \vec{x}} \quad (4.5)$$

olur. Eğer $\vec{x}$ Fiedler vektörü ise yukarıdaki oran minimum olacaktır. Genel olarak

$$\frac{\vec{x}^T L(G) \vec{x}}{\vec{x}^T \vec{x}} \quad (4.6)$$

oranına $\vec{x}$ vektörünün Rayleigh oranı denir. N.Alon, A.J. Sinclair ve M.R. Jerrum küçük Fiedler değerli çizgelerin iyi bir kesme oranına sahip olduğunu ispatlamışlardır (Alon küçük düğüm ayırıcının varlığını ispatlamıştır). M. Mihail çalışmaları sonucunda, bütün elemanları 1 olan vektöre dik olan ve Rayleigh oranı küçük olan her vektör kullanılarak iyi bir kesme oranı elde edilebilir [19].

4.1.2.2. Spektral Bölütleme Yöntemleri

G çizgesinin Laplace matrisinin Fiedler vektörü $u = (u_1, \dots, u_n)$ olsun. Spektral çizge bölütlemenin ana fikri, bir tane bölütleme değeri s bulunup, Fiedler vektörü iki parçaya bölünür, öyle ki bir parça için $u_i > s$ ve diğer parça için $s \leq u_i$ olur. Bu tip bir çizge kesmesine

Fiedler kesmesi denir. Genel olarak Fiedler kesmesini bulmak için, $(u_1, \dots, u_n)$ ' nin medyanı s olarak alınır ve bu şekilde seçilen s kesme ayırıcına eş iki parça kesmesi denir. s 'nin en iyi kesme oranını verdiği durumda ise, buna kesme oranı, s ' nin değerinin 0 olması durumunda s 'ye işaret kesmesi, s sıralanmış Fiedler vektörü elemanları içinde en uzun aralık içinde bir değere sahipse buna da aralık kesmesi denir [19].

4.1.2.3. Matrislerin Özdeğeri ve Özvektörünün Hesabı

Verilen bir denklem sisteminin abes olmayan bir çözümünün olabilmesi için, bu denklem sisteminin katsayılar matrisi elde edildikten sonra bundan elde edilen vektörlerin bağımsız olması gerekmektedir. $U_1, \dots, U_n$ vektörler olsunlar. Eğer bu vektörler lineer bağımsız iseler:

$$c_1 U_1 + \dots + c_n U_n = 0 \quad (4.7)$$

(4.7) eşitliğindeki katsayılar için, $c_1 = c_2 = \dots = c_n = 0$ eşitliği sağlar. Eğer en az bir tane c_i ($i=1, 2, \dots, n$) sıfırdan farklı ise bu vektörler lineer bağımsız değildir. X bir vektör olsun ve A katsayılar matrisi olsun

$$AX = \lambda X \quad (4.8)$$

eşitliğini sağlayan λ , A matrisinin özdeğeri ve X ' te λ ' ya karşılık gelen özdeğer vektörüdür. (4.8) eşitliği kullanılarak

$$(A - \lambda I)X = 0 \quad (4.9)$$

(4.9) eşitliği elde edilir. Bu eşitliğin determinantı alınınca A matrisinin karakteristik polinomu elde edilir. Bu polinomun kökleri A matrisinin özdeğerleridir. Bu özdeğerler kullanılarak özdeğer vektörleri elde edilebilir. Bilgisayar ortamında bunun çözümü biraz daha farklı olmaktadır. Bunun için geliştirilmiş yöntemler vardır. Bu yöntemler Kuvvet yöntemi, Jacobi yöntemi, Householder yöntemi gibi yöntemlerdir. Kuvvet yöntemi en küçük özdeğere karşılık gelen özdeğer vektörünü bulmaktadır. Jacobi yöntemi bütün özdeğerleri ve bunlara karşılık düşen özdeğer vektörlerini bulmaktadır. Householder yöntemi ise reel simetrik olan bir matrisi üç-diagonal matrisine dönüştürmektedir.

Jacobi yönteminde bütün özdeğerler ve özdeğer vektörleri bulunabildiğinden, bu çalışmada, bu yöntem kullanılmıştır ve burada bu yöntemden söz edilecektir. Jacobi yöntemi simetrik matrislerin özdeğerlerini ve özdeğer vektörlerini bulmada kullanılmaktadır. Bir çizgenin Laplace matrisi de simetrik bir matris olduğundan, bu yöntem kullanılabilir. Bu yöntem bütün simetrik reel matrislerin özdeğer ve özdeğer vektörler probleminin çözümünü garanti etmektedir.

Düzlem Döndürme

Esas algoritmaya geçilmeden önce bazı geometrik bilgilerin verilmesi gerekmektedir. X n -boyutlu bir vektör olsun ve (4.10) bağıntısındaki lineer dönüşümün yapıldığı varsayalım.

$$Y=RX \quad (4.10)$$

$$R = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \dots & \cos\phi & \dots & \sin\phi & \dots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \dots & -\sin\phi & \dots & \cos\phi & \dots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \quad \begin{matrix} \leftarrow p \\ \leftarrow q \end{matrix} \quad (4.11)$$

$\uparrow \quad \uparrow$
 $p \quad q$

R , $n \times n$ boyutunda diyagonal elemanları 1 ve diyagonal dışındaki elemanları sıfır olan bir matristir. Fakat bir fark vardır, p ve q sütun ve satırına denk gelen diyagonal elemanları $\cos\phi$, $r_{pq}=-\sin\phi$ ve $r_{qp}=\sin\phi$. (4.10) bağıntısındaki lineer dönüşümü (4.12) bağıntısındaki gibi olur.

$$\begin{aligned} y_j &= x_j \text{ eğer } j \neq p \text{ ve } j \neq q \\ y_p &= x_p \cos\phi + x_q \sin\phi \\ y_q &= -x_p \sin\phi + x_q \cos\phi \end{aligned} \quad (4.12)$$

Bu dönüşüm n -boyutlu bir uzayın $x_p x_q$ -düzleminde ϕ açısı ile döndürülmesidir. Bunun ters dönüşümü ise aynı uzayın aynı düzlemde $-\phi$ açısı ile döndürülmesidir ve ters dönüşüm $X=R^{-1}Y$. R matrisi ortogonal bir matris olduğundan

$$R^{-1}=R^T \text{ veya } R^T R=I \quad (4.13)$$

olur.

Benzerlik ve Ortogonal Dönüşümler

$$AX=\lambda X \quad (4.14)$$

Bu özdeğer problemi düşünülecek olursa, K matrisi singular olmayan bir matris olsun ve B (4.15) bağıntısına göre tanımlanmış kabul edilerek,

$$B=K^{-1}AK \quad (4.15)$$

(4.15) bağıntısının her iki tarafı $K^{-1}X$ çarpılarak olursa

$$BK^{-1}X=K^{-1}AKK^{-1}X=K^{-1}AX=K^{-1}\lambda X=\lambda K^{-1}X \quad (4.16)$$

elde edilir. Değişken değişimi aşağıdaki gibi yapılırsa

$$Y=K^{-1}X \text{ veya } X=KY \quad (4.17)$$

elde edilir. Eğer (4.17) bağıntıları (4.16)'da yerine yazılırlarsa

$$BY=\lambda Y \quad (4.18)$$

yeni bir özdeğer problemi elde edilmiş olur. (4.14) ve (4.18) bağıntıları karşılaştırılırlarsa, özdeğerleri aynı ve özdeğer vektörleri aynı olmak zorunda olmayan iki tane matrisin elde edildiği görülür (A ve B matrisleri). Bu da benzerlik dönüşümünün özdeğerleri koruduğunu gösterir.

R matrisinin ortogonal olduğu kabul edilsin ve D matrisi şu şekilde tanımlansın.

$$D=R^T A R \quad (4.19)$$

Bu bağıntının her iki tarafı $R^T X$ ile çarpılırsa

$$D R^T X = R^T A R R^T X = R^T A X = R^T \lambda X = \lambda R^T X \quad (4.20)$$

elde edilir. Aynı şekilde değişken değişimi şu şekilde yapılabilir.

$$Y=R^T X \text{ veya } X=RY \quad (4.21)$$

(4.20) ve (4.21) beraber kullanılırsa, yeni bir özdeğer problemi elde edilir.

$$D Y = \lambda Y \quad (4.22)$$

$R^{-1}=R^T$ olduğundan X' ten Y' nin ve Y' den X' in elde edilmesi kolaydır. (4.14) bağıntısı ile (4.22) bağıntısı karşılaştırılırsa, özdeğer probleminde özdeğerlerin değişmediği ve sadece yeni bir matrisin elde edildiği görülür.

Eğer A matrisi simetrik bir matris ise;

$$D^T = (R^T A R)^T = R^T A (R^T)^T = R^T A R = D \quad (4.23)$$

olur. Böylece D matrisi de simetrik bir matristir. Böylece eğer A simetrik ve R ortogonal matrisler ise, A matrisinden D matrisine olan dönüşüm hem özdeğerleri ve hem de simetrikliği korumaktadır. Özdeğer vektörleri arasındaki ilişki ise (4.21)'de verilmiştir.

Dönüşümlerin Jacobi Serileri

Dönüşümlere, reel ve simetrik olan A matrisi ile başlanır ve ortogonal olan $R_1, R_2, \dots, R_n$ matrisleri şu şekilde elde edilirler.

$$\begin{aligned} D_0 &= A \\ D_j &= R_j^T D_{j-1} R_j \quad j = 1, 2, \dots \end{aligned} \quad (4.24)$$

Pratikte D matrisinin diyagonal üzerinde olmayan elemanları sıfıra yaklaşıncı işlem durdurulur ve;

$$\lim_{j \rightarrow \infty} D_j = D = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \quad (4.25)$$

$$D_n = D$$

olur. Buradan;

$$D_n = R_n^T R_{n-1}^T \cdots R_1^T A R_1 R_2 \cdots R_{n-1} R_n \quad (4.26)$$

olur. Eğer R matrisi için şu tanımlama yapılırsa

$$R = R_1 R_2 \cdots R_{n-1} R_n \quad (4.27)$$

olur ve bu tanımlamadan yola çıkılarak, $R^{-1}AR = D$ aşağıdaki sonucu verir.

$$AR = RD = R \text{ diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \quad (4.28)$$

R matrisinin sütunları $X_1, X_2, \dots, X_n$ vektörleri ile gösterilirse, $R = [X_1, X_2, \dots, X_n]$ olur.

Buradan;

$$AR = [AX_1, AX_2, \dots, AX_n] = [\lambda_1 X_1, \lambda_2 X_2, \dots, \lambda_n X_n] \quad (4.29)$$

Görüldüğü gibi R matrisinin sütunları özdeğer vektörlerini göstermektedir.

Jacobi yönteminde, her adımda A matrisinin a_{pq} ve a_{qp} elemanları sıfır yapılır. R_1 ilk ortogonal matris ve $D_1 = R_1^T A R_1$ olsun. R_1 matrisi (4.11)'de görülen matrisin aynısıdır ve tek fark olarak $c = \cos \phi$ ve $s = \sin \phi$ olarak alınacaktır. Her adımda değişikliklerin p ve q satırları ile p ve q sütunlarında yapıldığının ispatlanması lazımdır. $B = A R_1$ olsun.

$$B = \begin{bmatrix} a_{11} & \cdots & a_{1p} & \cdots & a_{1q} & \cdots & a_{1n} \\ a_{p1} & \cdots & a_{pp} & \cdots & a_{pq} & \cdots & a_{pn} \\ a_{q1} & \cdots & a_{qp} & \cdots & a_{qq} & \cdots & a_{qn} \\ a_{n1} & \cdots & a_{np} & \cdots & a_{nq} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \quad (4.30)$$

Bağıntı (4.30)'daki çarpım yapılacak olursa:

$$\begin{aligned} b_{jk} &= a_{jk} & \text{eğer } k \neq p \text{ ve } k \neq q \\ b_{jp} &= c a_{jp} - s a_{jq} & j = 1, 2, \dots, n \\ b_{jq} &= s a_{jp} + c a_{jq} & j = 1, 2, \dots, n \end{aligned} \quad (4.31)$$

olur. $D_1 = R_1^T A R_1$ dönüşümü A matrisinin sadece p ve q satırları ile p ve q sütunlarının değişimine sebep olacaktır. A matrisinin diğer elemanları değişmeden D matrisine geçecektir.

$$\begin{aligned} d_{jp} &= c a_{jp} - s a_{jq} & \text{eğer } j \neq p \text{ ve } j \neq q \\ d_{jq} &= s a_{jp} + c a_{jq} & \text{eğer } j \neq p \text{ ve } j \neq q \\ d_{pp} &= c^2 a_{pp} + s^2 a_{qq} - 2cs a_{pq} \\ d_{qq} &= s^2 a_{pp} + c^2 a_{qq} + 2cs a_{pq} \\ d_{pq} &= (c^2 - s^2) a_{pq} + cs(a_{pp} - a_{qq}) \end{aligned} \quad (4.32)$$

D_1 matrisinin diğer elemanları simetri ile bulunur. Burada önemli olan c ve s değişkenlerinin değerlerinin nasıl bulunacağıdır.

$$\theta = \cot 2\phi = \frac{c^2 - s^2}{2cs} \quad (4.33)$$

Bu eşitlikten (4.32) eşitliğinin son denklemini kullanarak;

$$0 = (c^2 - s^2)a_{pq} + cs(a_{pp} - a_{qq}) \quad (4.34)$$

elde edilir. Bu eşitlik düzenlenerek aşağıdaki biçime getirilir.

$$\theta = \frac{a_{qq} - a_{pp}}{2a_{pq}} \quad (4.35)$$

$t = s/c = \tan \phi$ olur ve (4.35) eşitliği kullanılarak aşağıdaki denklem elde edilir.

$$t^2 + 2t\theta - 1 = 0 \quad (4.36)$$

Buradan:

$$t = -\theta \pm (\theta^2 + 1)^{1/2} = \text{sign}(\theta) / (|\theta| + \theta^2 + 1)^{1/2} \quad (4.37)$$

olur ve buradan c ve s hesaplanabilir.

$$c = 1 / (t^2 + 1)^{1/2} \quad (4.38)$$

$$s = ct$$

Bir çizgenin Laplace matrisinin özdeğer vektörlerin bulunması ile nasıl bölütlenebileceğini ilk olarak Fiedler göstermiştir. Bunu çizgenin iki parçası arasında kalan ayrıtların ağırlık toplamının minimum olması gerektiğini kabul ederek ve çizgenin iki parçaya bölündüğünü düşünerek, bir parçadaki düğümlere +1 değeri atanmış ise diğerine -1 değeri atanarak, aşağıdaki fonksiyonun bu iki parça arasında kalan ayrıt sayısını verdiği ortaya çıkar (4.39). Bu fonksiyonu kullanarak numerik yöntemler ile çizgenin nasıl bölütleneceğinin teorisi şu şekilde olur.

$$f = \frac{1}{4} \sum_{ij} E_{ij} (x_i - x_j)^2 \quad (4.39)$$

Çizgenin derece matrisi D ve bitişiklik matrisi de A olsun.

$$\begin{aligned} \sum_{ij} E_{ij} (x_i - x_j)^2 &= \sum_{ij} E_{ij} (x_i + x_j)^2 - \sum_{ij} 2x_i x_j \\ \Rightarrow \sum_{ij} E_{ij} 2x_i x_j &= \sum_{i,j} x_i A_{ij} x_j = \sum_i x_i \sum_j A_{ij} x_j = x^T A x \\ \Rightarrow \sum_{ij} E_{ij} (x_i + x_j)^2 &= \sum_{ij} 2 = 2|E| = \sum_{i,j} x_i^2 d_i = x^T D x \end{aligned} \quad (4.40)$$

Buradan:

$$\sum_{ij} E (x_i - x_j)^2 = x^T D x - x^T A x = x^T (D - A) x \quad (4.41)$$

D - A matrisi bilindiği gibi Laplace matrisidir.

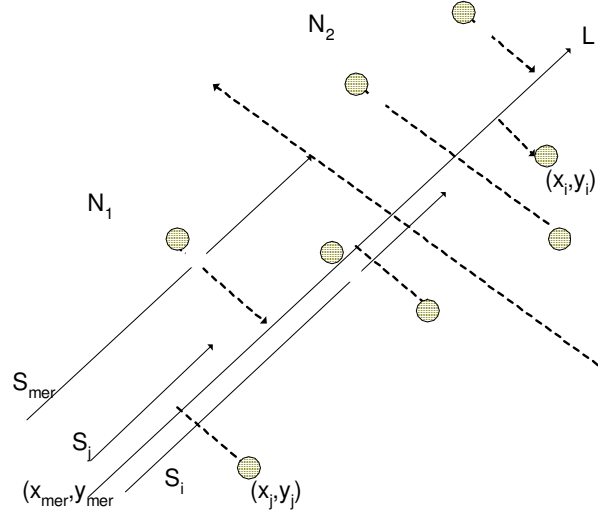
4.1.3. Eylemsizlik Algoritması

Bu yöntem, geometrik bilgileri kullanan, basit ve hızlı olan bir algoritmadır. Bu yöntemde çizgenin yanında bu çizgenin düğümlerinin bir, iki veya üç boyutlu uzayda koordinat bilgisinin olması gerekmektedir. Düğümleri kütesel nokta olarak kabul eden bu yöntem, 2B uzayda noktaları bir doğru ile iki parçaya böler, 3B uzayda ise bir düzlem ile noktaları iki parçaya böler. Bu işlem aşağıdaki algoritmaya göre yapılır.

1. L gibi bir doğru seçilsin ve bu doğrunun denklemi $a(x-x_{mer})+b(y-y_{mer})=0$. Bu doğru (x_{mer}, y_{mer}) noktasından geçer ve $-a/b$ eğimine sahiptir. $a^2+b^2=1$ kabul edilmesi genelliği bozmaz.
2. Her $n_i=(x_i, y_i)$ düğümü için $S_i=-b(x_i-x_{mer})+a(y_i-y_{mer})$ skalar çarpımı yapılır. S_i , (x_{mer}, y_{mer}) noktasından (x_i, y_i) noktasının L doğrusuna iz düşümünün yapılmasında elde edilen noktaya olan uzaklıktır.
3. S_i 'lerin S_{mer} medyanı bulunur.
4. (x_i, y_i) düğümü $S_i \leq S_{mer}$ şartını sağlayıp ve N_1 parçası içinde olsun ve $S_i > S_{mer}$ olan düğümler de N_2 parçasında olsun.

Bu anlatılan yöntem Şekil 14'de görülmektedir. Bu çizgede ayrıtlar gösterilmemiştir çünkü algoritmada çizgenin ayrıtları kullanılmamaktadır. Bunun yerine çizgedeki düğümlerin koordinat sistemindeki birbirlerine olan uzaklıkları önemlidir. Eğer çizgenin ayrıtları düğümler arasındaki bu uzaklıkla uyumlu ise bu durumda çizge bölütlendikten sonra ara kesitte kalan ayrıt sayısı az olacaktır.

Burada önemli olan L doğrusunun yönünün seçilmesidir. Eğer Şekil 4.2'de görülen çizge gibi düğümler dar ve uzun bir bölgede yer alıyorsa bu durumda L doğrusu bu alanın boyu yönündeki eksen yönünde olacaktır. Öyle bir (x, y) nokta seçilir ki bütün noktaların bir bütün olarak minimum eylemsizlik momentinin eksenini olmalıdır. Bundan dolayı bu metoda eylemsizlik yöntemi denilmektedir. a, b, x_{mer} , y_{mer} değerleri öyle seçilmelidir ki $a^2+b^2=1$ ve aşağıdaki değer minimum olmalıdır.



Şekil 4.2. Eylemsizlik yöntemi ile 2B’de çizge bölütleme [19].

$$\begin{aligned}
 X_2 &= \sum_{i=1}^{|N|} (x_i - x_{mer}), \quad Y_2 = \sum_{i=1}^{|N|} (y_i - y_{mer}) \text{ ve} \\
 XY &= \sum_{i=1}^{|N|} (x_i - x_{mer}) (y_i - y_{mer}) \\
 &\sum_{i=1}^{|N|} (\text{nokta nokta olan doğruların uzunluğu})^2 \\
 &= \sum_{i=1}^{|N|} ((x_i - x_{mer})^2 + (y_i - y_{mer})^2 - (-b(x_i - x_{mer}) + a(y_i - y_{mer}))^2) \\
 &= a^2 \left(\sum_{i=1}^{|N|} (x_i - x_{mer})^2 \right) + b^2 \left(\sum_{i=1}^{|N|} (y_i - y_{mer})^2 \right) + 2ab \left(\sum_{i=1}^{|N|} (x_i - x_{mer}) (y_i - y_{mer}) \right) \\
 &= a^2 X_2 + b^2 Y_2 + 2ab XY \\
 &= \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} X_2 & XY \\ XY & Y_2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \\
 &= \begin{bmatrix} a & b \end{bmatrix} M \begin{bmatrix} a \\ b \end{bmatrix}
 \end{aligned} \tag{4.42}$$

x_{mer} ve y_{mer} değerlerinin şu olduğu gösterilebilir.

$$x_{mer} = \sum_{i=1}^{|N|} \frac{x_i}{|N|} \text{ ve } y_{mer} = \sum_{i=1}^{|N|} \frac{y_i}{|N|} \tag{4.43}$$

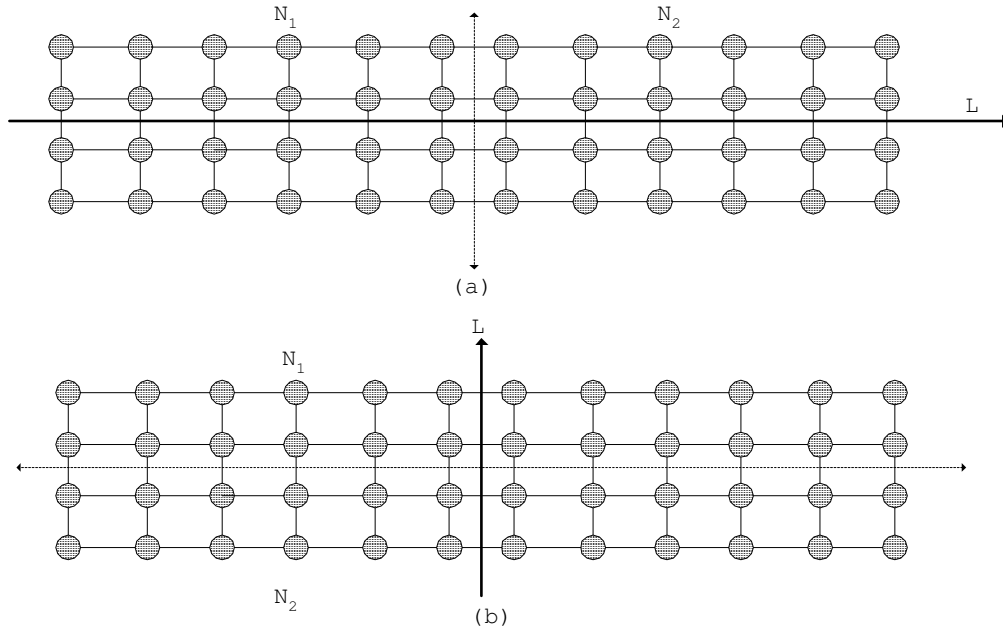
(x_{mer}, y_{mer}) noktası kütlelerin ağırlık merkezidir ve $\begin{bmatrix} a & b \end{bmatrix}$, M matrisinin en küçük özdeğerine denk düşen bir birim özdeğer vektörüdür.

(x,y) düğüm seçiminin iyi yapılması gerekmektedir. Eğer bu düğüm seçimi iyi yapılmaz ise Şekil 4.3.b’de görüldüğü gibi hiç de iyi olmayan bir sonuç elde edilir. Eğer bu seçim iyi yapılırsa, Şekil 4.3.a’da görüldüğü gibi daha iyi bir sonuç elde edilir.

3D ve daha yüksek boyutlu uzaylarda yöntem aynı şekilde işlemektedir, fakat 3D için 3×3 boyutlarında bir matris elde edilir ve bu matrisin en küçük özdeğerine denk düşen birim özdeğer vektörü bulunur. nD için ise $n \times n$ boyutlarında bir M matrisi bulunması gerekir ve aynı zamanda bu matrisin en küçük özdeğerinin birim özdeğer vektörü bulunmalıdır.

Özellikle spektral yöntem düşünüldüğü zaman bu yöntemin kalitesi düşük bölütleme yaptığı görülmektedir. Bu yöntemden sonra iyileştirme algoritmalarından biri uygulandığı zaman daha kaliteli bölütleme sonucu elde edilebilmektedir. Eğer geliştirme algoritmalarından Kernighan-Lin uygulanırsa bunun sonucunda elde edilen bölütleme sadece spektral bölütleme ile elde edilen sonuçtan daha iyidir. Fakat Spektral algoritmadan sonra iyileştirme algoritması uygulandığı zaman daha iyi sonuç elde edilmektedir.

Bir doğru ile noktaları iki eş parçaya bölütlemek için başka bir yöntem de orjinden geçen bir doğru seçmektir. Bu doğrunun belirlenen bir eksen ile yaptığı açı, yük dengeleme şartına göre değiştirilir veya değiştirilmez. Bu yöntemde, özyinelemeli eş iki parçaya bölüt stratejisi kullanılabilir ve bu strateji hızlı bir stratejidir. Bir noktanın (düğümün) bir düzleme olan üyeliği o noktanın iki boyutlu olan bu düzlemdeki yeri ile belirlenir.



Şekil 4.3. Eylemsizlik yöntemine göre aynı ağın iki farklı şekilde bölütlenmesi,

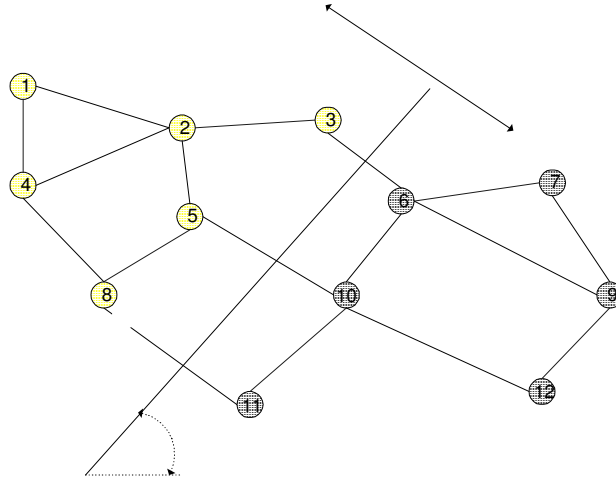
(a) İyi (x,y) seçimi, bölütleme kesmesinin boyutu 4’tür.

(b) Kötü (x,y) seçimi, bölütleme kesmesinin boyutu 12’dir [19].

Şekil 4.4 bu algoritmanın nasıl çalıştığını çok iyi olarak göstermektedir. İlk olarak seçilen bir eksene göre bir açı değeri seçilir ve bu açı değeri ile düzlemde bir doğru çizilir. Eğer bu doğru noktaları iki eş kümeye bölebiliyorsa işlem durdurulur, aksi halde eleman sayısı fazla olan kümeye doğru gidecek şekilde açı değeri değiştirilir. Tekrar eleman sayısı kontrol edilir. Bu işlem en iyi çözüm elde edilinceye kadar devam ettirilir. i düğümünün düzlemdeki yeri (x_i, y_i) ' dir ve her noktaya verilen değer (4.44) eşitliği ile hesaplanır.

$$x'_i = \cos(\alpha)x_i - \sin(\alpha)y_i \quad (4.44)$$

(4.44) eşitliği x eksenini etrafında α açısı kadar döndürme yapar.



Şekil 4.4. Bir doğru ile noktaları iki eş parçaya bölütme [19].

4.1.4. Çok Seviyeli Çizge Bölütme

P -yollu çizge bölütme problemi şu şekilde tanımlanabilir. $G=(V,E)$, $|V|=n$ olan bir yönlendirilmemiş çizge olsun. Bu çizgenin düğümleri öyle p tane $V_1, V_2, \dots, V_p$ kümeye ayrılır ki;

$$\forall i,j \ V_i \cap V_j = \emptyset, i \neq j, |V_i| \leq n/p \text{ ve } \cup V_i = V \quad (4.45)$$

ve uç düğümleri farklı parçalar içinde olan ayrıt sayısı da minimize edilir.

$$M_p: V \{P_a | a \in [1,p]\} \text{ ile } M_p(v) = P_a \Leftrightarrow v \in P_a \quad (4.46)$$

P , n tane elemanı olan bir vektördür ve $P[v]$, 1 ile p arasında bir sayı olur. Uç noktaları farklı parçalarda olan ayrıtların sayısına kesim-ayrıt denir.

Özyinelemeli iki eş parçaya bölütme algoritması p -yollu çizge bölütme probleminde kullanılabilir. İlk etapta çizge 2 parçaya bölütlenir, bundan sonra her parça kendi içinde iki parçaya bölütlenir ve bu işlem $\log(p)$ adım tekrarlanınca p tane parça elde edilir. Böylece p -yollu çizge bölütme işlemi 2-yollu çizge bölütme işleminin bir seri olarak tekrarından

ibarettir. Bu yöntem her zaman iyi sonuç verdiği için değil, basit olduğundan dolayı kullanılır. Çünkü bazı durumlarda bu yöntem iyi sonuç vermeyebilir.

Çok seviyeli çizge bölütleme stratejisi çok basittir. İlk önce $G=(V,E)$ çizgesi eğer çok sayıda düğüme sahipse, istenilen düğüm sayısı elde edilinceye kadar kabalaştırılır ve bu yeni çizgenin iki eş parçaya bölütleme algoritması ile bölütlenmesi yapılır (başlangıç bölütleme), daha sonra bu parçalar orijinal çizge elde edilinceye kadar açılır ve her safhada iyileştirme algoritmaları uygulanır. Şekil 4.5’de bu yöntemin görsel şekli verilmiştir.

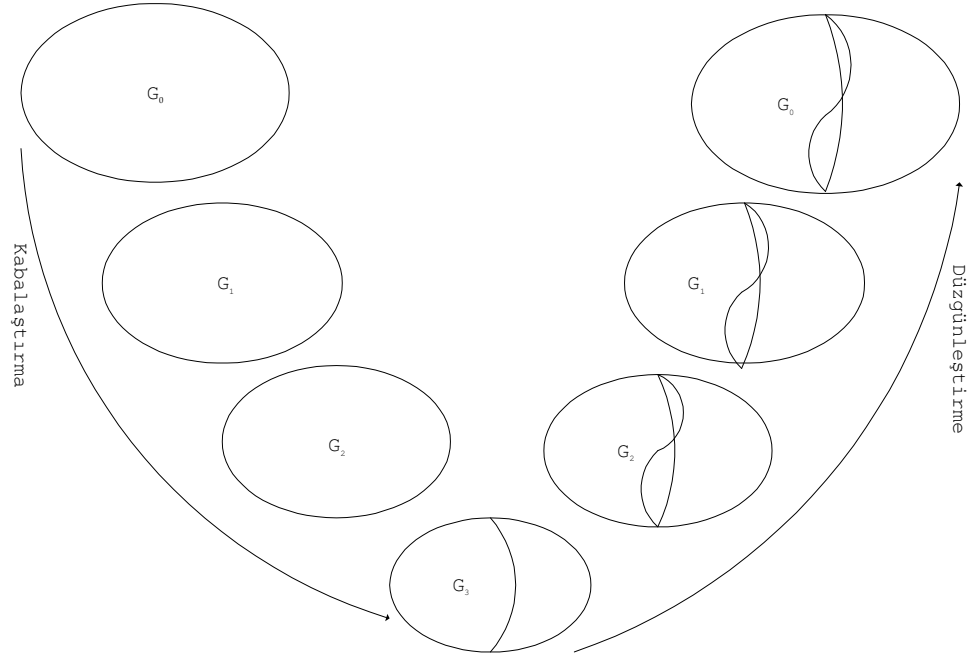
Kabalaştırma safhasında orijinal çizgeden $G_0=(V_0,E_0)$ daha kaba bir çizge $G_1=(V_1,E_1)$ elde edilir öyle ki $|V_0| > |V_1|$ ’dir. Bu işlem $|V_i| > |V_{i+1}|$ şartı göz önünde tutularak devam ettirilir. V_{i+1} kabalaştırılmış çizgenin G_{i+1} düğümler kümesi ve V_i kendisinden kabalaştırılmış çizge elde edilen G_i çizgenin düğümler kümesidir. G_{i+1} çizgesi G_i çizgesinde maksimum eşleştirme kümesi bulunarak, eşleşen düğümler bir hiperdüğüme indirgenerek elde edilir. Eşleştirme yöntemi hem bölütlemenin kalitesini ve hem de kabalaştırma safhasında geçecek zamanı etkiler. Bunlardan biri gelişigüzel eşleştirme algoritmasıdır. Bu algoritmada düğümler gelişigüzel olarak ele alınır ve eşleştirilmemiş düğümler kendisine komşu olan ve eşleştirilmemiş bir düğüm ile eşleştirilirler. Bu eşleştirmeden daha iyi olan diğer bir eşleştirme algoritması ise en ağır ayrıt eşleştirme algoritmasıdır. Düğümler gelişigüzel ele alınır ve bu düğüm kendisine komşu olan düğümlerden eşleşmemiş olan ve ayrıt ağırlığı en fazla olan düğüm ile eşleştirilirler. Bu işlem eşleştirilebilecek düğüm kalmayana kadar devam ettirilir.

Eğer $G_i=(V_i,E_i)$ çizgesinin maksimum eşleştirmesi M_i ve bu çizgenin eşleştirme ayrıtlarının büzüştürülmesi ile $G_{i+1}=(V_{i+1},E_{i+1})$ elde edildiyse,

$$W(E_{i+1})=W(E_i)-W(M_i) \quad (4.47)$$

(4.47) bağıntısı elde edilir ve G_{i+1} çizgesinin toplam ayrıtlar ağırlığı G_i çizgesinin ayrıtların toplam ağırlığından toplam eşleştirme ağırlığının çıkarılması ile elde edilir. Eğer eşleştirme maksimum ise, bu durumda G_{i+1} çizgesinin toplam ayrıtlar ağırlığı daha az olacaktır, bu da demektir ki kesim ayrıtlarının ağırlığı az olacaktır.

Değiştirilmiş ağır ayrıt eşleştirme algoritması ise, kabalaştırılmış çizgelerin ortalama derecesinin daha az olmasını sağlamaya çalışır. Bu algoritmada düğümler gelişigüzel olarak ele alınır. v böyle bir düğüm olsun ve H eşleşmemiş v düğümünün en ağır ayrıtlarla bağlanmış olan komşularının kümesi olsun. Her $u \in H$, W_{v-u} u düğümünü v düğümünün komşularına bağlayan ayrıtların ağırlıkları toplamı olsun. Bu algoritmada v düğümü W_{v-u} ağırlığı en fazla olan düğüm ile eşleştirilir.



Şekil 4.5. Çok seviyeli çizge bölütleme yöntemi [19].

Bu algoritmanın sonucunda kabalaştırılmış çizgede daha az ayrıt ve ortalama ayrıt ağırlığı daha fazla olan bir kaba çizge elde edilir. Daha ileriki seviyelerde eşleştirmede yer alan ayrıtların ağırlığı artar, bu da bu algoritmanın daha etkili olmasını sağlar. Eğer çizgenin ayrıtlarının hepsinin ağırlıkları aynı ise bu algoritma bir önceki algoritmadan daha etkilidir. Bu tip bir çizgenin ilk adımında gelişigüzel eşleştirme algoritmasının sonucu ile ağır ayrıt eşleştirme algoritmasının sonucu aynı olur, çünkü bütün ayrıtlar aynı ağırlıktadır. Fakat değiştirilmiş ağır ayrıt eşleştirmesi daha iyi sonuç verir.

Çizgenin kabalaştırılmış çizgesi elde edildikten sonra, spektral yöntemine göre başlangıç bölütleme yapılır. Elde edilen kabalaştırılmış çizgenin düğüm sayısı orijinal çizgenin düğüm sayısından az olduğundan bölütleme işlemi orijinal çizgenin bölütlenmesine göre daha az zaman alır.

Eğer kabalaştırılmış çizge G_p ise düzgünleştirme işlemi $G_p, G_{p-1}, \dots, G_0$ şeklinde devam eder. Her parçada bulunan hiperdüğümlerin içindeki düğümler de o parçaya ait olduğundan, her hiperdüğüm için o hiperdüğümü oluşturan düğümler yerine konur ve ayrıtlar tekrar düzenlenir. Her düzgünleştirme adımından sonra her parçaya iyileştirme algoritması uygulanır. İyileştirme algoritmalarının amacı, eğer bir düğümün bir parçadan diğer parçaya geçişi kesim ayrıt sayısını azaltıyor ve yük dengeleme şartını bozmuyorsa, bu düğüm o parçaya transfer edilir.

Çok seviyeli çizge bölütleme algoritmalarında iki tip paralellik kullanılır. İlk paralel algoritma, problem çözümünün özyinelemeli doğal yapısından gelmektedir. İlk olarak bir

işlemci orijinal çizgenin iki eş parçasını bulur, ondan sonra iki işlemci her biri bir parçanın iki eş parçalarını bulurlar ve bu işlem p tane parça elde edilinceye kadar devam ettirilir. Bu işlemde $\log(p)$ tane işlemciye ihtiyaç duyar ve toplam bölütleme işlemini seri bölütleme zamanının $\log(p)$ 'ye bölünmesi ile elde edilir.

İkinci paralel algoritma ise iki eş parçayı elde etme adımı safhasında kullanılır. Bu algoritmada orijinal çizgenin bölütlenmesi tek işlemci yerine birden fazla işlemci üzerinde paralel olarak yapılır. Bölütleme adımı paralelleştirildiğinden dolayı, bu algoritmanın toplam zamanı birinci tipteki algoritmadan daha iyidir, çünkü bölütleme algoritmalarında zamanın çoğu bölütleme işlemi için harcanır.

Kabalaştırma esnasında birden fazla kaba çizge elde edilir. $G_{i+1}=(V_{i+1},E_{i+1})$ kaba çizgesi için, $G_i=(V_i,E_i)$ kaba çizgesinin maksimum eşleştirilmesi M_i bulunur ve bu eşleştirmedeki ayrıtlara göre hiperdüğümler oluşturulur ve ayrıtlar düzenlenir. Bu adım, bu çok seviyeli algoritmanın en fazla zaman alan kısmıdır ve bu adımın paralelleştirilmesi gerekir.

Gelişigüzel seri eşleştirme algoritması basit ve etkilidir. Paralel olarak eşleştirmenin hesaplanması zordur (özellikle dağıtık hafızalı sistemlerde). Seri eşleştirme algoritmasının paralelleştirilmesi, işlemciler arasında fazla iletişim gerektirmektedir. Örneğin gelişigüzel eşleştirme algoritması paralelleştirilecek olunursa, her işlemci orijinal çizgenin gelişigüzel bir alt çizgesini içerir. Her yerel v düğümü için her işlemci eşleştirmede olacak bir ayrıt (v,u) seçer. Burada yapılması gereken, yerel olarak v ve u düğümlerini barındıran işlemcilerin iletişimi sonucunda bu düğümlerin eşleşmenin içinde olup olmadığının kontrol edilmesinden sonra eğer eşleşmenin içinde değillerse eşleşmeye eklenirler. Burada dikkat edilmesi gereken bir nokta vardır, o da işlemciler arasında yarış şartının (race condition) ortaya çıkmaması gerekir. Bir işlemci (v,u) ayrıtını kontrol ederken, diğer bir işlemci (u,w) gibi bir ayrıtı kontrol etmek isteyebilir, ancak bu ayrıtlardan biri eşleşmenin içinde yer alabilir. Maksimum bağımsız küme bulunarak eşleştirme yapan algoritmalar da vardır ve bu tip bir algoritma da paralelleştirilebilir. Her durumda işlemciler kendi üzerinde bulunan yerel düğümün komşu düğümleri için diğer işlemcilerle iletişime geçmesi gerekmektedir. Bu da eşleştirme safhasında algoritmanın toplam zamanının önemli bir kısmı burada harcanmaktadır. Eşleştirme yapıldıktan sonra, eşleştirmede yer alan düğümlerin büzüştürülmesi gerekir. Eşleştirmede yer alan her ayrıtın uçları farklı işlemciler üzerinde ise ayrıt büzüştürmeden önce düğüm transferi yapılır. İşlemciler arası iletişim zamanının azaltılması için işlemcilere, işlemciler arasında kalan ayrıtların bir üst sınırı göz önünde tutularak düğümler dağıtılmalıdır. Bu yukarıda verilen algoritmalarındaki iletişim için harcanan zamanın azaltılması anlamına gelir. Bu da başlangıçta bölütleme işleminin yapılması anlamına gelmektedir.

Eşleştirmeyi hesaplamak için başka bir algoritma, her işlemciye, düğümler kümesinin bir alt kümesi dağıtılır ve bu işlemciler kendi üzerinde bulunan düğümler arasında bir eşleştirme yaparlar. Bu algoritmanın avantajı, eşleştirme yapılırken işlemciler arasında iletişime başvurulmaz ve komşuluk listesinin işlemciler arasında transferine ihtiyaç duyulmaz. Bu algoritmanın problemi ise her işlemci üzerinde az sayıda düğüm olması ve bu işlemciler ile düğümler arasında bir eşleştirme yapmalarıdır. Komşuluk listesinin transferine gerek olmadığı halde, her işlemci kendisi üzerinde olan düğümlerin komşu düğümlerinin eşleştirmedeki durumları hakkında bilgi sahibi olmaları kabalaştırma safhasının iyi yapılmasını sağlar. Bu da işlemciler arasında iletişime sebep olur.

Ayrıtların büzüştürülmelerinden sonra elde edilen kabalaştırılmış çizgenin başlangıç bölütlenmesi yapılır. Bölütleme işlemi için daha önce verilen algoritmalarından biri kullanılabilir.

Düzgünleştirme esnasında, kabalaştırılmış çizgenin bölütlenmiş halinin orijinal çizge üzerine izdüşümü alınırken $G_p, G_{p-1}, \dots, G_0$ safhalarından geçer. Her izdüşümü alma adımından sonra elde edilen parçalara iyileştirme algoritması uygulanır. Nasıl ki kabalaştırma safhasında paralel algoritmalar kullanılabildiyse, düzgünleştirme safhasında da paralel algoritmalar kullanılabilir.

5. ÇİZGE BÖLÜTLEME YÖNTEMİ UYGULAMALARI

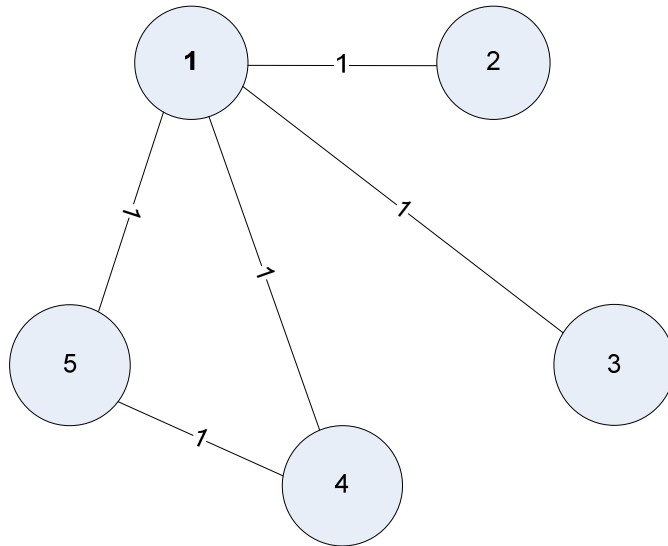
Burada çizge bölütleme algoritmasının önce çok basit bir örnek ile veri madenciliği ilişkisi (birliktelik) problemi için, daha sonra Hepatits ve KOAH veri tabanları için çalışması örnek ve uygulamalarla gösterilecektir.

Örnek 1: Çizge Bölütleme Yönteminin bir alış-veriş verisine uygulanması.

Bu örneğimizde beş müşteri ve beş ürün olduğu ve bunların bir kerede yapmış olduğu alışveriş bilgileri Çizelge 5.1'deki gibi olsun. Burada '0' ilgili ürünün alınmadığını, '1' ise alındığını göstermektedir. Amaç en iyi yoğun nesne kümelerini bulmak olsun.

	A	B	C	D	E
1	0	0	0	1	1
2	0	0	1	1	1
3	0	1	0	1	1
4	1	0	0	1	1
5	1	0	0	1	1

Çizelge 5.1. Örnek veritabanı



Şekil 5.1. Çizelge 5.1'deki örnek veritabanının çizge modeli.

Burada önce birinci müşteri ile diğer müşteriler, sonra ikinci müşteri ile diğer müşteriler, üçüncü müşteri ile diğer müşteriler şeklinde sırasıyla karşılaştırılarak almış oldukları ürünlerin dört tanesi (% 80 ve üzeri) aynı ise 1 değeri verilerek veritabanının çizge modeli Şekil 5.1'deki gibi oluşturulur. Burada % 80 ve üzeri olanlar aynı gruba alınma ihtimali olabileceğinden çizge modeli oluşturulurken ilgili düğümler arasına ayrıt eklendi. Benzerlik oranı % 80'den az olanlara ayrıt eklenmedi. Buna göre bitişiklik matrisi A;

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

Aynı çizgenin derece matrisi D;

$$D = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \text{ olur.}$$

Çizge bölütleme yöntemi kullanılarak verilerin iki gruba ayrılması için çizgenin Laplace matrisinin (L) elde edilmesi lazım. Laplace matrisi :

$$L = D - A = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

şeklinde elde edilir.

$$|L - \lambda I| = \lambda (\lambda - 1)^2 (\lambda - 3) (\lambda - 5) \text{ olup özdeğerler: } [0, 1, 1, 3, 5] \text{ bulunur.}$$

L matrisinin karakteristik polinomu:

$$L(\lambda) = \lambda^5 - 10\lambda^4 + 32\lambda^3 - 38\lambda^2 + 15\lambda \text{ olup,}$$

$$L(\lambda) = 0 \text{ için kökler: } \lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 1, \lambda_4 = 3 \text{ ve } \lambda_5 = 5 \text{ bulunur.}$$

$$\lambda = 1 \text{ için özvektörler uzayı: } D_{\lambda=1}(L) = \{[0, -a-2b, a, b, b]^T ; a, b \in \mathbb{R}\} \text{ bulunur.}$$

$$a = 1, b = 0 \text{ için bir özvektör: } x = [0, -1, 1, 0, 0]^T \text{ vektörü,}$$

$$a = 0, b = 1 \text{ için } y = [0, -2, 0, 1, 1]^T \text{ özvektörü alınabilir.}$$

MATLAB 7.5.0 programı ile de özdeğerler aşağıdaki gibi bulundu:

```
>> L = [4 -1 -1 -1 -1; -1 1 0 0 0; -1 0 1 0 0; -1 0 0 2 -1; -1 0 0 -1 2];  
  
>> eig(L)  
  
ans =  
  
-0.0000  
1.0000  
1.0000  
3.0000  
5.0000  
  
>>
```

Ayrıca internette [20] online çalışan bir program olan Calculator for Eigenvalues and Eigenvectors da özdeğerler ve özvektörleri :

Characteristic polynomial:

$$x^5 - 10x^4 + 32x^3 - 38x^2 + 15x$$

Real eigenvalues:

{0, 1, 1, 3, 5}

Eigenvector of eigenvalue 0: (1, 1, 1, 1, 1)

Eigenvector of eigenvalue 1: (0, -1, 1, 0, 0)

Eigenvector of eigenvalue 1: (0, -2, 0, 1, 1)

Eigenvector of eigenvalue 3: (0, 0, 0, -1, 1)

Eigenvector of eigenvalue 5: (-4, 1, 1, 1, 1)

şeklinde bulundu.

Örnek 2: Hepatits veritabanı Çizge Modelinin Oluşturulması.

Bu örneğimizde Hepatits veritabanında rasgele 7 satırlık bir veri alınarak önce çizge modeli oluşturuldu sonra da özdeğer ve özvektörler hesaplamaları için gerekli matris işlemleri yapıldı.

38, 1, 1, 2, 1, 1, 1, 2, 2, 2, 2, 1, 2, 2.00, 72, 89, 2.9, 46, 1
66, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1.20, 102, 53, 4.3, ?, 1
40, 1, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 2, 0.60, 62, 166, 4.0, 63, 1
38, 1, 1, 1, 2, 2, 2, 1, 1, 2, 2, 2, 2, 0.70, 70, 28, 4.2, 62, 1
27, 1, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1.20, 133, 98, 4.1, 39, 1
31, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1.00, 85, 20, 4.0, 100, 1
42, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0.90, 60, 63, 4.7, 47, 1

Çizelge 5.2. Hepatits veritabanında rasgele 7 satır 19 sütunluk bir veri

Bu veritabanı ile ilgili özellik bilgileri(attribute information):

1. AGE: 10, 20, 30, 40, 50, 60, 70, 80
2. SEX: male, female
3. STEROID: no, yes
4. ANTIVIRALS: no, yes
5. FATIGUE: no, yes
6. MALAISE: no, yes
7. ANOREXIA: no, yes
8. LIVER BIG: no, yes
9. LIVER FIRM: no, yes
10. SPLEEN PALPABLE: no, yes
11. SPIDERS: no, yes
12. ASCITES: no, yes
13. VARICES: no, yes
14. BILIRUBIN: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00
15. ALK PHOSPHATE: 33, 80, 120, 160, 200, 250
16. SGOT: 13, 100, 200, 300, 400, 500,
17. ALBUMIN: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0
18. PROTIME: 10, 20, 30, 40, 50, 60, 70, 80, 90
19. HISTOLOGY: no, yes

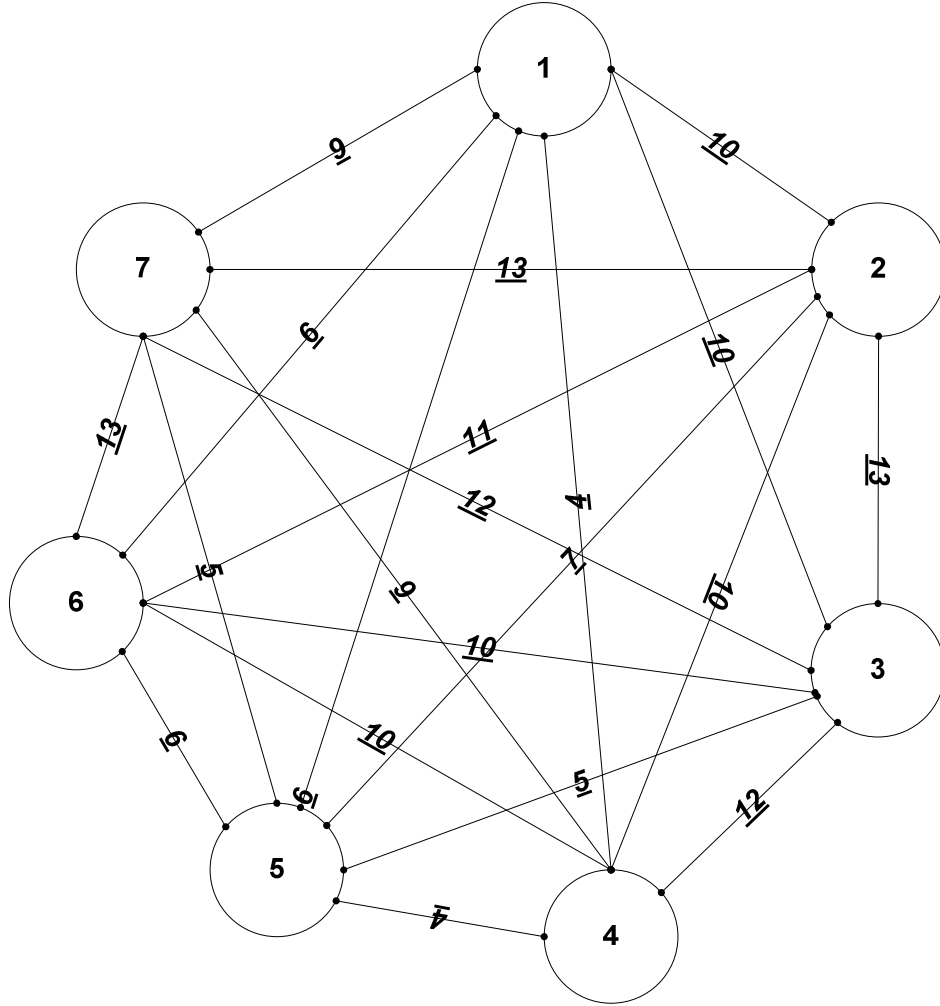
şeklindedir.

Buradaki verilerde ikinci satır sonundaki ? ile gösterilen bilinmeyen değer için veri temizleme yöntemi uygulanarak ortalama değer (60) kullanılacaktır. Birinci sütundaki AGE (yaş) 0-35, 36-55, 56 üzeri olarak, 14. sütundaki Bilirubin 0.39-1.00, 1.01 üzeri olarak, 15. sütundaki Alk Phospate 0-17 yaş arası çocuk için 0-480; 17-100 yaş arası yetişkin için 60-300

olarak, 16. sütundaki SGOT 0-40 normal değer aralığı 40 üzeri yüksek değer olarak, 17. sütundaki Albumin 3.8-5.2 normal değer aralığı olarak değerlendirildi (İnönü Ün.Turgut Özal Tıp Merkezi Klinik Laboratuvar sonuç raporları baz alındı). Ayrıca 2. (sex), 13. (varices) ve 19. (histology) sütundaki veriler her satırda aynı olduğundan (veri temizleme işlemi uygulanmak suretiyle) çıkarılarak aşağıdaki 16 sütun ve 7 satırlık veri üzerinde işleme başlandı.

38, 1, 2, 1, 1, 1, 2, 2, 2, 2, 1, 2.00, 72, 89, 2.9, 46
66, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 1.20, 102, 53, 4.3, 60
40, 1, 2, 1, 2, 2, 2, 1, 2, 2, 2, 0.60, 62, 166, 4.0, 63
38, 1, 1, 2, 2, 2, 1, 1, 2, 2, 2, 0.70, 70, 28, 4.2, 62
27, 2, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1.20, 133, 98, 4.1, 39
31, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1.00, 85, 20, 4.0, 100
42, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0.90, 60, 63, 4.7, 47

Çizelge 5.3. Hepatits veritabanında rasgele 7 satır 16 sütunluk bir veri



Şekil 5.2. Çizelge 5.3'deki örnek veritabanının çizge modeli.

Buna göre bitişiklik matrisi A;

$$A = \begin{bmatrix} 0 & 10 & 10 & 4 & 6 & 6 & 9 \\ 10 & 0 & 13 & 10 & 7 & 11 & 13 \\ 10 & 13 & 0 & 12 & 5 & 10 & 12 \\ 4 & 10 & 12 & 0 & 4 & 10 & 9 \\ 6 & 7 & 5 & 4 & 0 & 6 & 5 \\ 6 & 11 & 10 & 10 & 6 & 0 & 13 \\ 9 & 13 & 12 & 9 & 5 & 13 & 0 \end{bmatrix}$$

Aynı çizgenin derece matrisi D;

$$D = \begin{bmatrix} 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 \end{bmatrix}$$

olur.

Çizge kullanılarak verilerin iki gruba ayrılması için çizgenin Laplace matrisinin(L) elde edilmesi lazım. Laplace matrisi :

$$L = D - A = \begin{bmatrix} 6 & -10 & -10 & -4 & -6 & -6 & -9 \\ -10 & 6 & -13 & -10 & -7 & -11 & -13 \\ -10 & -13 & 6 & -12 & -5 & -10 & -12 \\ -4 & -10 & -12 & 6 & -4 & -10 & -9 \\ -6 & -7 & -5 & -4 & 6 & -6 & -5 \\ -6 & -11 & -10 & -10 & -6 & 6 & -13 \\ -9 & -13 & -12 & -9 & -5 & -13 & 6 \end{bmatrix}$$

şeklinde olur.

MATLAB 7.5.0 programında özdeğerlerin aşağıdaki gibi olduğu görüldü.

```
>> L=[6 -10 -10 -4 -6 -6 -9; -10 6 -13 -10 -7 -11 -13; -10 -13 6 -12 -5 -10 -12; -4 -10 -12 6 -4 -10 -9;  
      -6 -7 -5 -4 6 -6 -5; -6 -11 -10 -10 -6 6 -13; -9 -13 -12 -9 -5 -13 6];  
  
>> eig(L)  
  
ans =  
  
-48.7109  
  7.4143  
 10.6241  
 14.0108  
 18.4247  
 19.8735  
 20.3635  
  
>>
```

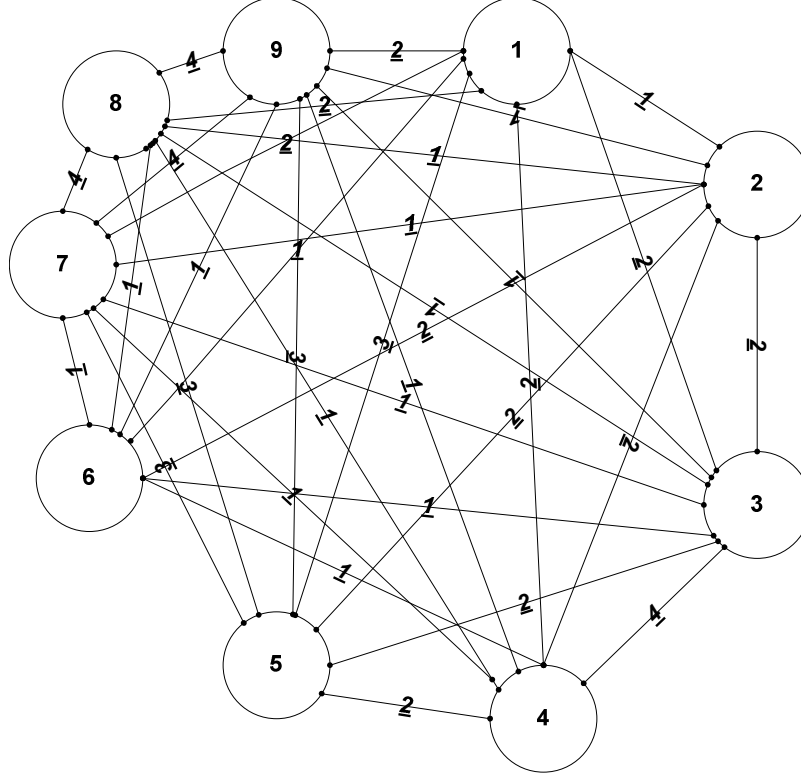
Örnek 3: KOAH veritabanı Çizge Modelinin Oluşturulması.

Bu örneğimizde KOAH veritabanında rasgele 9 satırlık bir veri alınarak önce çizge modeli oluşturuldu sonra da özdeğer ve özvektörler hesaplamaları için matris işlemleri yapıldı.

YAS	CINSIYET	SIGARA	FEV1
48	01	01	104
80	02	00	104
19	02	01	104
30	02	01	104
79	01	01	105
51	02	00	57
60	01	01	57
70	01	01	57
59	01	01	58

Çizelge 5.4. KOAH veritabanında rasgele 9 satır 4 sütunluk bir veri

Burada 1. sütundaki yaş verilerini 0 – 35 aralığı (genç) 1. grup, 36 – 55 aralığı (orta yaş) 2. grup ve 55 üzeri (yaşlı) 3. grup olarak değerlendirildi. Ayrıca FEV1 (Forced Expiration Volum 1 – 1. saniyedeki zorlanmış soluk verme hacmi) verilerinde 70 altındaki değerler olumsuz yani istenmeyen durumu, 80 üstündeki değerler ise istenen normal değerleri belirtmektedir.



Şekil 5.3. Çizelge 5.4'deki örnek veritabanının çizge modeli.

Buna göre bitişiklik matrisi A;

$$A = \begin{bmatrix} 0 & 1 & 2 & 2 & 3 & 1 & 2 & 2 & 2 \\ 1 & 0 & 2 & 2 & 2 & 2 & 1 & 2 & 1 \\ 2 & 2 & 0 & 4 & 2 & 1 & 1 & 1 & 1 \\ 2 & 2 & 4 & 0 & 2 & 1 & 1 & 1 & 1 \\ 3 & 2 & 2 & 2 & 0 & 0 & 3 & 3 & 3 \\ 1 & 2 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 3 & 1 & 0 & 4 & 4 \\ 2 & 2 & 1 & 1 & 3 & 1 & 4 & 0 & 4 \\ 2 & 1 & 1 & 1 & 3 & 1 & 4 & 4 & 0 \end{bmatrix}$$

Aynı çizgenin derece matrisi D;

$$D = \begin{bmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \end{bmatrix}$$

olur.

Çizge kullanılarak verilerin iki gruba ayrılması(sınıflandırılması) için çizgenin Laplace matrisinin(L) elde edilmesi lazım. Laplace matrisi :

$$L = D - A = \begin{bmatrix} 8 & -1 & -2 & -2 & -3 & -1 & -2 & -2 & -2 \\ -1 & 8 & -2 & -2 & -2 & -2 & -1 & -2 & -1 \\ -2 & -2 & 8 & -4 & -2 & -1 & -1 & -1 & -1 \\ -2 & -2 & -4 & 8 & -2 & -1 & -1 & -1 & -1 \\ -3 & -2 & -2 & -2 & 7 & 0 & -3 & -3 & -3 \\ -1 & -2 & -1 & -1 & 0 & 7 & -1 & -1 & -1 \\ -2 & -1 & -1 & -1 & -3 & -1 & 8 & -4 & -4 \\ -2 & -2 & -1 & -1 & -3 & -1 & -4 & 8 & -4 \\ -2 & -1 & -1 & -1 & -3 & -1 & -4 & -4 & 8 \end{bmatrix}$$

şeklinde olur.

MATLAB 7.5.0 programında özdeğerlerin aşağıdaki gibi olduğu görüldü.

```
>> L=[8 -1 -2 -2 -3 -1 -2 -2 -2; -1 8 -2 -2 -2 -1 -2 -1; -2 -2 8 -4 -2 -1 -1 -1 -1;  
      -2 -2 -4 8 -2 -1 -1 -1 -1; -3 -2 -2 -2 7 0 -3 -3 -3; -1 -2 -1 -1 0 7 -3 -3 -3;  
      -2 -1 -1 -1 -3 -1 8 -4 -4; -2 -2 -1 -1 -3 -1 -4 8 -4; -2 -1 -1 -1 -3 -1 -4 -4 8];  
  
>> eig(L)  
  
ans =  
  
-8.1605  
  
6.2447  
  
5.0439  
  
9.3140  
  
9.9217  
  
11.3477  
  
12.2885  
  
12.0000  
  
12.0000  
  
>>
```

Algoritmanın basit şekilde çalışmasını göstermek amacıyla bu örnekler verildi. Sonra da 155 satır ve 19 sütunluk Hepatits veritabanı ile 124 satır ve 4 sütunluk KOAH veritabanı uygulamaları yapıldı.

Uygulama 1: Çizge Bölütleme Algoritması ile Hepatits Veritabanının Kümelenmesi.

Ek 1. olarak verilen Hepatits veritabanı 155 satır, 19 sütun şeklindedir. Veritabanındaki veriler iki kümeden oluşmaktadır. Birinci küme:

7, 31, 32, 36, 68, 72, 77, 87, 88, 89, 92, 95, 99, 101, 105, 107, 110, 112, 119, 121, 128, 130, 132, 135, 139, 142, 144, 145, 147, 148, 151, 155

şeklinde olup geriye kalanlar ikinci kümeyi oluşturmaktadır. Kümedeki sayılar veritabanındaki kayıtların satır numaralarını göstermektedir. Veritabanındaki kayıtların listesi Ek.1.'de verilmiştir.

Bu veritabanı ile ilgili özellik bilgileri(attribute information):

1. AGE: 10, 20, 30, 40, 50, 60, 70, 80
2. SEX: male, female
3. STEROID: no, yes
4. ANTIVIRALS: no, yes
5. FATIGUE: no, yes
6. MALAISE: no, yes
7. ANOREXIA: no, yes
8. LIVER BIG: no, yes
9. LIVER FIRM: no, yes
10. SPLEEN PALPABLE: no, yes
11. SPIDERS: no, yes
12. ASCITES: no, yes
13. VARICES: no, yes
14. BILIRUBIN: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00
15. ALK PHOSPHATE: 33, 80, 120, 160, 200, 250
16. SGOT: 13, 100, 200, 300, 400, 500,
17. ALBUMIN: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0
18. PROTIME: 10, 20, 30, 40, 50, 60, 70, 80, 90
19. HISTOLOGY: no, yes

şeklindedir.

Buradaki verilerde ? ile gösterilen bilinmeyen değerler için veri temizleme yöntemi uygulanarak ortalama değer kullanılacaktır. Birinci sütundaki AGE (yaş) 0-35, 36-55, 56 üzeri olarak, 14. sütundaki Bilirubin 0.39-1.00, 1.01 üzeri olarak, 15. sütundaki Alk Phospate 60-300 normal değer aralığı olarak, 16. sütundaki SGOT 0-40 normal değer aralığı, 40 üzeri yüksek

değer olarak, 17. sütundaki Albumin 3.8-5.2 normal değer aralığı olarak değerlendirildi (İnönü Ün.Turgut Özal Tıp Merkezi Klinik Laboratuvar sonuç raporları baz alındı).

Yapılan Uygulama sonucunda 1. kümedeki kayıtların 13 tanesi ve ikinci kümedeki kayıtların 37 tanesi hatalı veya yanlış kümeye dahil edildiği görüldü. Uygulamanın başarı oranı ise:

1. küme için: $100 - (13/155) * 100 = \% 91.61$
2. küme için: $100 - (37/155) * 100 = \% 76.13$

olarak elde edildi.

Daha önce fidan gelişim algoritması ile yapılan kümelemede başarı oranı % 75 olarak elde edilmişti. Bu yöntemin başarı oranının daha yüksek olduğu görüldü.

Burada Çizge Bölütleme Yöntemi ile Sınıflandırma ve Kümeleme için kullanılan algoritma detayları ve açıklamaları aşağıdaki gibidir. Algoritma1, Hepatits veritabanından verileri okuyarak çizgenin bitişiklik matrisini oluşturmaktadır. Algoritma2 oluşturulan çizgenin derece matrisini oluşturmaktadır. Algoritma3 ilgili çizgenin Laplace matrisini elde etmektedir. Ortagonal dönüşümler ile matrisin diagonal bant haline getirilmesi işlemleri Algoritma4, Algoritma5, Algoritma6 ile gerçekleştirilmektedir. Oluşan bant matrizen özdeğer vektörleri ise Algoritma7 ile elde edilmektedir. Elde edilen özdeğer vektörleri arasında ikinci en küçük özdeğere karşılık gelen özvektör ise Algoritma8 ile elde edilmekte ve Algoritma9 ise yazılımın temel algoritmasıdır.

Uygulama 2: Çizge Bölütleme Algoritması ile KOAH Veritabanının Kümelenmesi.

Burada 124 satır, 4 sütunluk KOAH veritabanının 1.sutundaki yaş verileri 0 – 35 aralığı (genç) 1. grup, 36 – 55 aralığı (orta yaş) 2. grup ve 55 üzeri (yaşlı) 3. grup olarak değerlendirildi. Ayrıca FEV1 (Forced Expiration Volum1 – 1. saniyedeki zorlanmış soluk verme hacimi) verilerinde 70 altındaki değerler olumsuz yani istenmeyen durumu, 80 üstündeki değerler ise istenen normal değerleri belirtmektedir.

Çizge bölütleme yöntemi uygulandıktan sonra sağlıklı sınıf 1-96 satırlar (2. satır hariç) ve hastalıklı sınıf 97-124 satırlar (115, 117, 121 ve 123 hariç) şeklindedir. Buna göre;

Sağlıklı sınıfın hata oranı: $(1/124)*100 = \% 0.80$

Hastalıklı sınıfın hata oranı: $(4/124)*100 = \% 3.22$

Sağlıklı sınıfın başarı oranı: $\% 99.2$

Hastalıklı sınıfın başarı oranı: $\% 96.78$

şeklindedir.

6. SONUÇ

Çizge bölütleme yöntemi, başta tıbbi verilerde sınıflandırma ve kümelendirme problemleri olmak üzere bir çok bilimsel problemin çözümünün kolaylaştırılması veya bu çözümün bulunması amacıyla yapılan çalışmalar sonucunda ortaya çıkmıştır. Bundan dolayı bu yöntemin (çizgelerin analizi ve testi) geliştirilmesi çok önemlidir.

Günümüze kadar çizgeler üzerinde etkili çalışan çok sayıda algoritma geliştirilmiştir. Bunlar P sınıfı algoritmalarıdır. Bunun yanında hala yaklaşımla çözüm bulunan veya yaklaşık çözüm bulunan çizge problemleri de vardır. Bunlara örnek olarak verilen çizgenin bütün düğümlerini kapsayacak minimum ayırıt kümesinin bulunması, çizge bölütleme vb. problemlerdir.

Bu çalışmanın en önemli özelliği çizge bölütleme algoritmaları uygulanmasının yapılmış olmasıdır.

Çizge bölütleme algoritmaları, temel olarak iki grupta toplanabilirler. Bir çizgenin düğümlerinin uzayda koordinatları belli ise, bu durumda çizge bölütleme işlemi daha kolay olmaktadır. Ya sıralama algoritmaları kullanılarak bölütleme yapılır ya da her düğüm bir kütle kabul edilerek bu kütlelerin ağırlık merkezinin bulunması yöntemi ile çizge bölütleme işlemi yapılır. Eğer çizgenin düğümlerinin koordinatları yoksa, bu durumda işlemler biraz daha zorlaşmaktadır. Bu durumda, birden fazla algoritma kullanılabilir. Fakat bunların en iyi sonuç vereni spektral yöntemidir. Bu yöntemde, çizgenin Laplace matrisi oluşturulur ve nümerik metotlar kullanılarak bu çizgenin özdeğer vektörleri bulunur. Yapılacak olan bölütlemeye göre bu özdeğer vektörlerinden biri kullanılır. Bu çalışmada ikinci en küçük özdeğere karşılık gelen özdeğer vektörü kullanılmıştır. Bu vektör ile çizge ikiye bölütlenmiştir.

Çizge bölütleme işlemi bir yaklaşım metodu olduğundan elde edilen sonuç optimum olmayabilir.

Düğüm koordinatları belli olmayan çizgelerin bölütlenmesi için kullanılan yöntemler içinde en iyisi spektral yöntemidir. Çünkü spektral yöntemin global olarak bir güçlülüğü vardır. Bu yöntemde çizgenin şekli veya özelliği hiç önemli değildir. Çünkü çözüm cebirsel olarak yapılmaktadır.

Eğer çizgedeki düğüm sayısı çok fazla ise, bu durumda bilgisayarların hafıza sınırı olduğundan ve işlem zamanının önemli olmasından dolayı, bu çizgenin kabalaştırılmış hali elde edilir ve elde edilen kabalaştırılmış çizgenin bölütlenmesi yapılır. Kabalaştırma işlemi birden fazla seviyeden oluşabilir. Bundan dolayı bu yöntemde çok seviyeli çizge bölütleme yöntemi denir.

Yukarıda anlatılan yöntemlerin hepsinde işlem esnasında veya bölütleme yapıldıktan uzun bir süre sonraya kadar çizgede herhangi bir değişikliğin olmayacağı varsayılmıştır. Eğer bu varsayım doğru değilse, bu durumda artımsal çizge bölütleme yöntemi ile çizge bölütlenir. Bu yöntemin ana fikri, çizge anlatılan yöntemlerden biri ile bölütlenir ve çizgede bir değişiklik olursa, bu değişiklik için tekrar baştan bölütleme yapılmaz, fakat bölütleme sonucu üzerinde işlemler yapılarak yük dengeleme ve minimum kesme ayırıt sayısı şartları sağlanmaya çalışılır.

Çizge bölütleme probleminin en büyük sıkıntısı düğümlerin ve ayırıtların ağırlıkları gelişigüzel olursa ve bir düğümün derecesi de gelişigüzel olursa, bu çizgelerin bölütlenmesi için iyi bir kriterin uygulanamamasından kaynaklanmaktadır. Kriter geliştirme yönünde Fiedler' in yaptığı bir çalışma var, fakat bu çalışmada çizgenin düğüm ve ayırıt ağırlıkları bir kabul edilerek yapılmıştır. Çizge ağırlıkları ile birlikte genel bir çizge değildir. Fiedler bu işi çizgelerin bir alt kümesi için yapmıştır.

Çizge bölütleme yöntemi ile verilerin sınıflandırılması ve kümelendirilmesi işlemi son yıllarda ele alınan bir konu durumuna gelmiştir. Bundan dolayı bu tez çalışmasında verilerin sınıflandırılması ve kümelendirilmesi işlemi için çizge bölütleme yöntemi kullanıldı ve iki tane farklı tıbbi veritabanı üzerinde uygulaması gerçekleştirildi. Birinci uygulamada Hepatits verisi üzerine uygulandı ve elde edilen sonuçların stokastik yöntemlerde elde edilen sonuçlardan daha iyi olduğu gözlemlendi. İkinci uygulamada ise nefes darlığı KOAH veritabanı üzerine uygulaması yapıldı. Bu veriler İnönü Üniversitesi Turgut Özal Tıp Merkezi Göğüs Hastalıkları bölümünden alındı ve alınan verilere başka bir yöntem uygulanmadığından karşılaştırma imkanı bulunamadı. Buna rağmen elde edilen sonuçların iyi olduğu rahatlıkla ifade edilebilir, çünkü başarı oranı % 96'nın üzerinde çıkmıştır.

KOAH veritabanında gürültülü veri olmadığı için başarı oranı % 96'dan daha yüksek olmuştur.

KAYNAKLAR

1. S. Haykin, 1998, Neural Networks: A Comprehensive Foundation, 2. baskı 842 sayfa, Prentice Hall.
2. D. Goldberg, 1989, Genetic Algorithm in Search, Optimization and Machine Learning, Massachusetts, Addison-Wesley Publishing Company mc.
3. L.N. de Castro , J. Timmis, 2002, Artificial Immune Systems: A New Computational Intelligence Approach, 357 sayfa, Springer-Verlag.
4. M Dorigo, T, Stützle, 2004, Ant Colony Optimization, 328 sayfa, MİT Pres.
5. Everitt BS. Statistical Methods in Medical Investigations, Second Edition, John Wiley&Sanss, 1994.
6. Wu JD, Milton DK, Hammond, SK, Spear RC. Hierarchical Cluster Analysis Applied to Workers Exposures in Fiberglas Insolution Manufacturing, Ann. Occup. Hyg., Vol. 43, No.1, pp. 43-55, 1999.
7. A. Karcı, 2004, Novelty in the Generation of Initial Population for Genetic Algorithms, KES2004, 20-24, Wellington, New Zealand.
8. E. Kaya, Tıpta Veri Ambarları Oluşturma ve Veri Madenciliği Uygulamaları
9. Dunham, M.H., Data Mining Introductory and Advencent Topics
10. Y. Kaya, R. Tekin, 2007, Veritabanı ve Uygulamaları, 230 sayfa, Papatya Yayıncılık, İstanbul, Türkiye.
11. S. Elmasri, B. Navathe, 1994, Fundamentals of Database Systems, R, Benjamin / Cumming Publishing.
12. M. Uysal, 1999, SQL Veritabanı Sorgulama Dili, Beta Basım Yayın.
13. Fayyad U., Shapiro G., and Smyth P., 1996, “From Data Mining to Knowledge Discovery in Database”, AI Magazine, 37-54.
14. Han J., 2001, “Data Mining : Concept and Techniques”, Morgan Kaufman.
15. U. Fayyad, et al. (1995), “From Knowledge Discovery to Data Mining: An Overview,” Advances in Knowledge Discovery and Data Mining, U. Fayyad et al. (Eds.), AAAI/MIT Press
16. A. Şengür, 2006, Doktora Tezi: Endoskopik Görüntülerin Değerlendirilmesinde Görüntü işleme Temelli Akıllı Karar Destek Sistemi, Fırat Üniversitesi, Elazığ, Türkiye.
17. K.K. Gundogan, B. Alatas, A. Karcı, “Mining Classification Rules by Using Genetic Algorithms with Non-random Initial Population and Uniform Operator”, Turkish Journal of Electrical Engineering & Computer Sciences, vol.12, pp.43-52, 2004.

18. S. Bandyopadhyay, S.K. Pal, "Classification and Learning Using Genetic Algorithms", Sptinger-Verlag, 2007.
19. A. Karcı, 1998, Yüksek Lisans Tezi: Çizge algoritmaları ve Çizge Bölümleme, Fırat Üniversitesi, Elazığ, Türkiye.
20. İnternet, http://www.arndt-bruenner.de/mathe/scripts/engl_eigenwert.htm, 07.07.2008

ÖZGEÇMİŞ

1964 MALATYA doğumlu. Malatya Hidayet ilkokulu, Kubilay ortaokulu, Ş.K.Ö. Endüstri Meslek Lisesi-1982 ve İstanbul Yıldız (Teknik) Üniversitesi Elektronik ve Haberleşme Mühendisliği -1986 mezunu.

Yedek subay olarak ifa edilen askerlikten hemen sonra ilk olarak 1988'de TÜVASAŞ Türkiye Vagon San. A.Ş. de Elektronik İşlemler Şube Şefi olarak 5 yıl çalıştı.

1993-1994 arası 1 yıl Sakarya Üniversitesi Bilgi İşlem Dairesi Başkanlığı'nda,

1994-1996 arası 2 yıl İnönü Üniversitesi Yapı İşleri ve Teknik Daire Başkanlığı Turgut Özal Tıp Merkezi İnşaatı Kontrol Amirliği'nde Elektronik, Haberleşme ve Bilgisayar Ağı Projeleri Kontrol Mühendisi olarak,

1996-1998 arası 2 yıl İnönü Üniversitesi Turgut Özal Tıp Merkezi Hastanesi'nde Elektronik, Haberleşme ve Bilgisayar Sistemleri ile Tıbbi Cihazlar Sorumlusu olarak çalıştı.

Haziran-1998'den sonra İnönü Üniversitesi Enformatik Bölümü Başkanlığı'nda “Kampus Bilgisayar Ağı Donanım Sorumlusu” olarak ve 1998 – 1999 Eğitim Öğretim Yılından itibaren “Temel Bilgi Teknolojileri”, “Eğitimde Bilgi Teknolojileri Kullanımı”, “Bilgisayar”, “Bilgisayar Donanımı”, “Bilgisayar Ağları ve İletişim” dersleri Okutmanı olarak görev yapmaktadır.

Yabancı dili İngilizce'dir.

Evli ve 4 çocuk babasıdır.

Yayınları:

1. Raylı Taşıtlarda Haberleşme, METRO dergisi, 1991, Sakarya.
2. Temel Bilgi Teknolojileri Ders Notları, İnönü Üniversitesi, 2000, Malatya
3. Bilgisayar Ders Notları, İnönü Üniversitesi, 2000, Malatya
4. Bilgisayar Ağları ve İnternetworking, Seminer Sunumu, İnönü Üniversitesi, 2001, Malatya.
5. M. Demir, M. Yiğiter, A. Karcı, Clustering by Using Saplings Growing up Algorithm, IKECCO-2007 International Kyrgyz - Kazak Electronic and Computer Conference 2007.
6. A. Karcı, M. Demir, M. Yiğiter, Naturel Inspired Computational Intelligence Method: Saplings Growing up Algorithm, IKECCO-2007 International Kyrgyz -Kazak Electronic and Computer Conference 2007.

EKLER

Ek 1. Hepatits Veritabanı.

1	30,2,1,2,2,2,1,2,2,2,2,1.00,85,18,4.0,?,1
2	50,1,1,2,1,2,2,1,2,2,2,2,0.90,135,42,3.5,?,1
3	78,1,2,2,1,2,2,2,2,2,2,2,0.70,96,32,4.0,?,1
4	31,1,?,1,2,2,2,2,2,2,2,2,0.70,46,52,4.0,80,1
5	34,1,2,2,2,2,2,2,2,2,2,2,1.00,?,200,4.0,?,1
6	34,1,2,2,2,2,2,2,2,2,2,2,0.90,95,28,4.0,75,1
7	51,1,1,2,1,2,1,2,2,1,1,2,2,?,?,?,?,1
8	23,1,2,2,2,2,2,2,2,2,2,2,1.00,?,?,?,1
9	39,1,2,2,1,2,2,2,1,2,2,2,2,0.70,?,48,4.4,?,1
10	30,1,2,2,2,2,2,2,2,2,2,2,1.00,?,120,3.9,?,1
11	39,1,1,1,2,2,2,1,1,2,2,2,2,1.30,78,30,4.4,85,1
12	32,1,2,1,1,2,2,2,1,2,1,2,2,1.00,59,249,3.7,54,1
13	41,1,2,1,1,2,2,2,1,2,2,2,2,0.90,81,60,3.9,52,1
14	30,1,2,2,1,2,2,2,1,2,2,2,2,2.20,57,144,4.9,78,1
15	47,1,1,1,2,2,2,2,2,2,2,2,?,?,60,?,?,1
16	38,1,1,2,1,1,1,2,2,2,2,1,2,2.00,72,89,2.9,46,1
17	66,1,2,2,1,2,2,2,2,2,2,2,1.20,102,53,4.3,?,1
18	40,1,1,2,1,2,2,2,1,2,2,2,2,0.60,62,166,4.0,63,1
19	38,1,2,2,2,2,2,2,2,2,2,2,0.70,53,42,4.1,85,2
20	38,1,1,1,2,2,2,1,1,2,2,2,2,0.70,70,28,4.2,62,1
21	22,2,2,1,1,2,2,2,2,2,2,2,2,0.90,48,20,4.2,64,1
22	27,1,2,2,1,1,1,1,1,1,1,2,2,1.20,133,98,4.1,39,1
23	31,1,2,2,2,2,2,2,2,2,2,2,1.00,85,20,4.0,100,1
24	42,1,2,2,2,2,2,2,2,2,2,2,0.90,60,63,4.7,47,1
25	25,2,1,1,2,2,2,2,2,2,2,2,0.40,45,18,4.3,70,1
26	27,1,1,2,1,1,2,2,2,2,2,2,2,0.80,95,46,3.8,100,1
27	49,1,1,1,1,1,1,2,1,2,1,2,2,0.60,85,48,3.7,?,1
28	58,2,2,2,1,2,2,2,1,2,1,2,2,1.40,175,55,2.7,36,1
29	61,1,1,2,1,2,2,1,1,2,2,2,2,1.30,78,25,3.8,100,1
30	51,1,1,1,1,1,2,2,2,2,2,2,2,1.00,78,58,4.6,52,1
31	39,1,1,1,1,1,2,2,1,2,2,2,2,2.30,280,98,3.8,40,1
32	62,1,1,2,1,1,2,?,?,2,2,2,2,1.00,?,60,?,?,1
33	41,2,2,1,1,1,1,2,2,2,2,2,2,0.70,81,53,5.0,74,1
34	26,2,1,2,2,2,2,2,1,2,2,2,2,0.50,135,29,3.8,60,1
35	35,1,2,2,1,2,2,2,2,2,2,2,2,0.90,58,92,4.3,73,1
36	37,1,2,2,1,2,2,2,2,2,1,2,2,0.60,67,28,4.2,?,1
37	23,1,2,2,1,1,1,2,2,1,2,2,2,1.30,194,150,4.1,90,1
38	20,2,1,2,1,1,1,1,1,1,1,2,2,2.30,150,68,3.9,?,1

39 42,1,1,2,2,2,2,2,2,2,2,1.00,85,14,4.0,100,1
 40 65,1,2,2,1,1,2,2,1,1,1,2,0.30,180,53,2.9,74,2
 41 52,1,1,1,2,2,2,2,2,2,2,2,0.70,75,55,4.0,21,1
 42 23,1,2,2,2,2,?,?,?,?,?,4.60,56,16,4.6,?,1
 43 33,1,2,2,2,2,2,2,2,2,2,2,1.00,46,90,4.4,60,1
 44 56,1,1,2,1,2,2,2,2,2,2,2,0.70,71,18,4.4,100,1
 45 34,1,2,2,2,2,2,2,2,2,2,?,?,86,?,?,1
 46 28,1,2,2,1,1,2,2,2,2,2,2,0.70,74,110,4.4,?,1
 47 37,1,1,2,2,2,2,2,1,2,1,2,2,0.60,80,80,3.8,?,1
 48 28,2,2,2,1,1,2,2,1,2,2,2,2,1.80,191,420,3.3,46,1
 49 36,1,1,2,2,2,2,2,1,2,2,2,0.80,85,44,4.2,85,1
 50 38,1,2,1,1,1,1,2,2,2,1,2,2,0.70,125,65,4.2,77,1
 51 39,1,1,2,2,2,2,2,2,2,2,2,0.90,85,60,4.0,?,1
 52 39,1,2,2,2,2,2,2,2,2,2,2,1.00,85,20,4.0,?,1
 53 44,1,2,2,2,2,2,2,2,2,2,2,0.60,110,145,4.4,70,1
 54 40,1,2,1,1,2,2,2,1,1,2,2,2,1.20,85,31,4.0,100,1
 55 30,1,2,2,1,2,2,2,2,2,2,2,0.70,50,78,4.2,74,1
 56 37,1,1,2,1,1,1,2,2,2,2,2,0.80,92,59,?,?,1
 57 34,1,1,2,?,?,?,?,?,?,?,?,?,?,?,1
 58 30,1,2,1,2,2,2,2,2,2,2,2,0.70,52,38,3.9,52,1
 59 64,1,2,1,1,1,2,1,1,2,2,2,2,1.00,80,38,4.3,74,1
 60 45,2,1,2,1,1,2,2,2,1,2,2,2,1.00,85,75,?,?,1
 61 37,1,2,2,2,2,2,2,2,2,2,2,0.70,26,58,4.5,100,1
 62 32,1,2,2,2,2,2,2,2,2,2,2,0.70,102,64,4.0,90,1
 63 32,1,2,2,1,1,1,2,2,2,1,2,1,3.50,215,54,3.4,29,1
 64 36,1,1,2,2,2,2,1,1,1,2,2,2,0.70,164,44,3.1,41,1
 65 49,1,2,2,1,1,2,2,2,2,2,2,0.80,103,43,3.5,66,1
 66 27,1,2,2,2,2,2,2,2,2,2,2,0.80,?,38,4.2,?,1
 67 56,1,1,2,2,2,2,2,2,2,2,2,0.70,62,33,3.0,?,1
 68 57,1,2,2,1,1,1,2,2,2,1,1,2,4.10,?,48,2.6,73,1
 69 39,1,2,2,1,2,2,2,2,2,2,2,1.00,34,15,4.0,54,1
 70 44,1,1,2,1,1,2,2,2,2,2,2,2,1.60,68,68,3.7,?,1
 71 24,1,2,2,2,2,2,2,2,2,2,2,0.80,82,39,4.3,?,1
 72 34,1,1,2,1,1,2,1,1,2,1,2,2,2.80,127,182,?,?,1
 73 51,1,2,2,1,1,1,?,?,?,?,?,0.90,76,271,4.4,?,1
 74 36,1,1,2,1,1,1,2,1,2,2,2,2,1.00,?,45,4.0,57,1
 75 50,1,2,2,2,2,2,2,2,2,2,2,1.50,100,100,5.3,?,1
 76 32,1,1,1,1,1,2,2,2,2,2,2,2,1.00,55,45,4.1,56,1
 77 58,1,2,2,1,2,2,1,1,1,1,2,2,2.00,167,242,3.3,?,1
 78 34,2,1,1,2,2,2,2,1,2,2,2,2,0.60,30,24,4.0,76,1
 79 34,1,1,2,1,2,2,1,1,2,1,2,2,1.00,72,46,4.4,57,1
 80 28,1,2,2,2,2,2,2,2,2,2,2,0.70,85,31,4.9,?,1
 81 23,1,2,2,1,1,1,2,2,2,2,2,2,0.80,?,14,4.8,?,1
 82 36,1,2,2,2,2,2,2,2,2,2,2,0.70,62,224,4.2,100,1
 83 30,1,1,2,2,2,2,2,2,2,2,2,0.70,100,31,4.0,100,1

84 67,2,1,2,1,1,2,2,2,?,?,?,1.50,179,69,2.9,?,1
85 62,2,2,2,1,1,2,2,1,2,2,2,1.30,141,156,3.9,58,1
86 28,1,1,2,1,1,1,2,1,2,2,2,2,1.60,44,123,4.0,46,1
87 44,1,1,2,1,1,2,2,2,1,2,2,1,0.90,135,55,?,41,2
88 30,1,2,2,1,1,1,2,1,2,1,1,1,2.50,165,64,2.8,?,2
89 38,1,1,2,1,1,1,2,1,2,1,1,1,1.20,118,16,2.8,?,2
90 38,1,1,2,1,1,1,1,1,2,2,2,2,0.60,76,18,4.4,84,2
91 50,2,1,2,1,2,2,1,1,1,1,2,2,0.90,230,117,3.4,41,2
92 42,1,1,2,1,1,1,2,2,1,1,2,1,4.60,?,55,3.3,?,2
93 33,1,2,2,2,2,2,?,?,2,2,2,2,1.00,?,60,4.0,?,2
94 52,1,1,2,2,2,2,2,2,2,2,2,1.50,?,69,2.9,?,2
95 59,1,1,2,1,1,2,2,1,1,1,2,2,1.50,107,157,3.6,38,2
96 40,1,1,1,1,1,1,1,1,2,2,2,2,0.60,40,69,4.2,67,2
97 30,1,1,2,1,1,2,2,1,2,1,2,2,0.80,147,128,3.9,100,2
98 44,1,1,2,1,1,2,1,1,2,1,2,2,3.00,114,65,3.5,?,2
99 47,1,2,2,2,2,2,2,2,2,1,2,1,2.00,84,23,4.2,66,2
100 60,1,1,2,1,2,2,1,1,1,1,2,2,?,?,40,?,?,2
101 48,1,1,2,1,1,2,2,1,2,1,1,1,4.80,123,157,2.7,31,2
102 22,1,2,2,2,2,2,2,2,2,2,2,0.70,?,24,?,?,2
103 27,1,1,2,1,2,2,2,1,2,2,2,2,2.40,168,227,3.0,66,2
104 51,1,1,2,1,1,1,2,1,1,1,2,1,4.60,215,269,3.9,51,2
105 47,1,2,2,1,1,2,2,1,2,2,1,1,1.70,86,20,2.1,46,2
106 25,1,2,2,2,2,2,2,2,2,2,2,0.60,?,34,6.4,?,2
107 35,1,1,2,1,2,2,?,?,1,1,1,2,1.50,138,58,2.6,?,2
108 45,1,1,2,1,1,1,2,2,2,2,2,2.30,?,648,?,?,2
109 54,1,1,1,2,2,2,1,1,2,2,2,2,1.00,155,225,3.6,67,2
110 33,1,1,2,1,1,2,2,2,2,2,1,2,0.70,63,80,3.0,31,2
111 7,1,2,2,2,2,2,2,1,1,2,2,2,0.70,256,25,4.2,?,2
112 42,1,1,1,1,1,2,2,2,2,1,2,2,0.50,62,68,3.8,29,2
113 52,1,1,2,1,2,2,2,2,2,2,2,2,1.00,85,30,4.0,?,2
114 45,1,1,2,1,2,2,2,1,1,2,2,2,1.20,81,65,3.0,?,1
115 36,1,1,2,2,2,2,2,2,2,2,2,2,1.10,141,75,3.3,?,2
116 69,2,2,2,1,2,2,2,2,2,2,2,2,3.20,119,136,?,?,2
117 24,1,1,2,1,2,2,2,2,2,2,2,2,1.00,?,34,4.1,?,2
118 50,1,2,2,2,2,2,2,2,2,2,2,2,1.00,139,81,3.9,62,2
119 61,1,1,2,1,1,2,?,?,2,1,2,2,?,?,?,?,2
120 54,1,2,2,1,2,2,1,1,2,2,2,2,3.20,85,28,3.8,?,2
121 56,1,1,2,1,1,1,1,1,2,1,2,2,2.90,90,153,4.0,?,2
122 20,1,1,2,1,1,1,2,2,2,1,1,2,1.00,160,118,2.9,23,2
123 42,1,2,2,2,2,2,2,2,1,2,2,2,1.50,85,40,?,?,2
124 37,1,1,2,1,2,2,2,2,2,1,2,2,0.90,?,231,4.3,?,2
125 50,1,2,2,2,2,2,2,1,1,1,2,2,1.00,85,75,4.0,72,2
126 34,2,2,2,1,1,1,1,1,2,1,2,2,0.70,70,24,4.1,100,2
127 28,1,2,2,1,1,1,?,?,2,1,1,2,1.00,?,20,4.0,?,2
128 50,1,2,2,1,2,2,2,1,1,2,1,1,2.80,155,75,2.4,32,2

129	54,1,1,2,1,1,2,2,2,2,1,2,1.20,85,92,3.1,66,2
130	57,1,1,2,1,1,2,2,2,2,1,1,2,4.60,82,55,3.3,30,2
131	54,1,2,2,2,2,2,2,2,2,2,2,1.00,85,30,4.5,0,2
132	31,1,1,2,1,1,1,2,2,1,2,2,2,8.00,?,101,2.2,?,2
133	48,1,2,2,1,1,1,2,1,2,1,2,2,2.00,158,278,3.8,?,2
134	72,1,2,1,1,2,2,2,1,2,2,2,2,1.00,115,52,3.4,50,2
135	38,1,1,2,2,2,2,2,1,2,2,2,2,0.40,243,49,3.8,90,2
136	25,1,2,2,1,2,2,1,1,1,1,1,1,1.30,181,181,4.5,57,2
137	51,1,2,2,2,2,2,1,1,2,1,2,2,0.80,?,33,4.5,?,2
138	38,1,2,2,2,2,2,2,1,2,1,2,1,1.60,130,140,3.5,56,2
139	47,1,2,2,1,1,2,2,1,2,1,1,1,1.00,166,30,2.6,31,2
140	45,1,2,1,2,2,2,2,2,2,2,2,2,1.30,85,44,4.2,85,2
141	36,1,1,2,1,1,1,1,1,2,1,2,1,1.70,295,60,2.7,?,2
142	54,1,1,2,1,1,2,?,?,1,2,1,2,3.90,120,28,3.5,43,2
143	51,1,2,2,1,2,2,2,1,1,1,2,1,1.00,?,20,3.0,63,2
144	49,1,1,2,1,1,2,2,2,1,1,2,2,1.40,85,70,3.5,35,2
145	45,1,2,2,1,1,1,2,2,2,1,1,2,1.90,?,114,2.4,?,2
146	31,1,1,2,1,2,2,2,2,2,2,2,2,1.20,75,173,4.2,54,2
147	41,1,2,2,1,2,2,2,1,1,1,2,1,4.20,65,120,3.4,?,2
148	70,1,1,2,1,1,1,?,?,?,?,?,1.70,109,528,2.8,35,2
149	20,1,1,2,2,2,2,2,?,2,2,2,2,0.90,89,152,4.0,?,2
150	36,1,2,2,2,2,2,2,2,2,2,2,2,0.60,120,30,4.0,?,2
151	46,1,2,2,1,1,1,2,2,2,1,1,1,7.60,?,242,3.3,50,2
152	44,1,2,2,1,2,2,2,1,2,2,2,2,0.90,126,142,4.3,?,2
153	61,1,1,2,1,1,2,1,1,2,1,2,2,0.80,75,20,4.1,?,2
154	53,2,1,2,1,2,2,2,2,1,1,2,1,1.50,81,19,4.1,48,2
155	43,1,2,2,1,2,2,2,2,1,1,1,2,1.20,100,19,3.1,42,2

Bu veritabanı ile ilgili özellik bilgileri (attribute information):

1. AGE: 10, 20, 30, 40, 50, 60, 70, 80
2. SEX: male, female
3. STEROID: no, yes
4. ANTIVIRALS: no, yes
5. FATIGUE: no, yes
6. MALAISE: no, yes
7. ANOREXIA: no, yes
8. LIVER BIG: no, yes
9. LIVER FIRM: no, yes
10. SPLEEN PALPABLE: no, yes
11. SPIDERS: no, yes
12. ASCITES: no, yes
13. VARICES: no, yes

- 14. BILIRUBIN: 0.39, 0.80, 1.20, 2.00, 3.00, 4.00
- 15. ALK PHOSPHATE: 33, 80, 120, 160, 200, 250
- 16. SGOT: 13, 100, 200, 300, 400, 500,
- 17. ALBUMIN: 2.1, 3.0, 3.8, 4.5, 5.0, 6.0
- 18. PROTIME: 10, 20, 30, 40, 50, 60, 70, 80, 90
- 19. HISTOLOGY: no, yes

Ek 2. KOAH Veritabanı (124 satır, 4 sütun).

S. No	YAS	CINSIYET	SIGARA	FEV1
1	85	02	00	45
2	70	01	01	45
3	65	02	00	49
4	75	02	00	50
5	31	02	00	76
6	64	02	00	76
7	72	02	00	76
8	66	02	00	76
9	69	02	00	76
10	30	02	01	76
11	23	02	01	77
12	65	02	00	77
13	32	02	01	77
14	47	02	00	77
15	63	01	00	77
16	35	02	00	77
17	22	02	01	77
18	60	01	00	78
19	49	02	00	78
20	30	01	01	78
21	74	02	00	78
22	40	01	01	78
23	70	02	00	78
24	59	01	00	78
25	50	02	00	78
26	41	02	00	85
27	30	02	00	85
28	60	01	01	85
29	72	02	00	85
30	41	01	00	86
31	22	02	00	86
32	56	02	00	86
33	45	01	00	86
34	53	02	00	86
35	63	02	00	86
36	52	02	00	86
37	19	01	00	86
38	70	02	00	86
39	23	02	00	86
40	20	02	00	86
41	31	02	00	86
42	35	02	00	86
43	49	02	00	86
44	55	01	01	86
45	46	02	00	86
46	32	02	00	86

47	41	01	00	87
48	65	02	00	87
49	30	02	00	87
50	54	01	00	87
51	49	01	00	87
52	25	01	00	87
53	22	02	00	87
54	59	01	00	87
55	43	01	01	87
56	73	01	00	87
57	78	02	00	87
58	26	02	01	87
59	40	02	00	87
60	68	02	00	87
61	36	01	01	88
62	70	02	00	88
63	56	02	00	88
64	42	02	00	88
65	58	01	00	91
66	49	02	00	91
67	62	02	00	91
68	44	02	00	91
69	34	02	00	91
70	32	02	00	91
71	50	01	00	91
72	56	02	00	91
73	30	02	01	103
74	51	02	01	103
75	26	02	01	103
76	33	02	01	103
77	24	01	01	103
78	56	01	01	104
79	43	01	01	104
80	62	01	01	104
81	42	02	01	104
82	32	01	01	104
83	70	01	01	104
84	30	01	01	104
85	50	01	01	104
86	38	01	01	104
87	60	01	01	104
88	30	01	00	104
89	33	02	00	104
90	46	02	00	104
91	60	02	00	104
92	48	01	01	104
93	80	02	00	104
94	19	02	01	104
95	30	02	01	104
96	79	01	01	105

97	51	02	00	57
98	60	01	01	57
99	70	01	01	57
100	59	01	01	58
101	18	01	00	59
102	30	01	01	59
103	47	01	01	59
104	44	02	00	60
105	41	02	01	61
106	48	01	01	61
107	59	02	00	62
108	38	02	00	63
109	53	02	01	63
110	59	01	01	63
111	39	01	01	67
112	70	01	01	67
113	80	01	01	69
114	39	02	00	70
115	77	01	00	72
116	20	01	01	73
117	65	02	00	73
118	60	01	01	73
119	55	01	01	73
120	43	01	01	74
121	45	01	00	75
122	59	01	01	75
123	54	02	00	76
124	49	01	01	77

Ek 3. Hazırlanan Yazılımın Önemli Algoritmaları

```
Algoritma 1: procedure TForm1.CizgeOlustur;
var
  i, j : integer;
  temp : real;
begin
  for i:=1 to MaxVertex do
    for j:=1 to MaxVertex do
      Matrix[i,j]:=0;

  for i:=1 to (MaxVertex-1) do
    begin
      for j:=i+1 to MaxVertex do
        begin
          temp:=0;
          if (abs(Hepatit[i].AGE-Hepatit[j].AGE)<=35) and
(Hepatit[i].AGE<>0) then
            temp:=temp+1;
          if (Hepatit[i].SEX=Hepatit[j].SEX) and (Hepatit[i].SEX<>0) then
            temp:=temp+1;
          if (Hepatit[i].STEROID=Hepatit[j].STEROID) and
(Hepatit[i].STEROID<>0) then
            temp:=temp+1;
          if (Hepatit[i].ANTIVIRALS=Hepatit[j].ANTIVIRALS) and
(Hepatit[i].ANTIVIRALS<>0) then
            temp:=temp+1;
          if (Hepatit[i].FATIGUE=Hepatit[j].FATIGUE) and
(Hepatit[i].FATIGUE<>0) then
            temp:=temp+1;
          if (Hepatit[i].MALAISE=Hepatit[j].MALAISE) and
(Hepatit[i].MALAISE<>0) then
            temp:=temp+1;
          if (Hepatit[i].ANOREXIA=Hepatit[j].ANOREXIA) and
(Hepatit[i].ANOREXIA<>0) then
            temp:=temp+1;
          if (Hepatit[i].LIVERBIG=Hepatit[j].LIVERBIG) and
(Hepatit[i].LIVERBIG<>0) then
            temp:=temp+1;
          if (Hepatit[i].LIVERFIRM=Hepatit[j].LIVERFIRM) and
(Hepatit[i].LIVERFIRM<>0) then
            temp:=temp+1;
          if (Hepatit[i].SPLEEN=Hepatit[j].SPLEEN) and
(Hepatit[i].SPLEEN<>0) then
            temp:=temp+1;
          if (Hepatit[i].SPIDERS=Hepatit[j].SPIDERS) and
(Hepatit[i].SPIDERS<>0) then
            temp:=temp+1;
          if (Hepatit[i].ASCITES=Hepatit[j].ASCITES) and
(Hepatit[i].ASCITES<>0) then
            temp:=temp+1;
          if (Hepatit[i].VARICES=Hepatit[j].VARICES) and
(Hepatit[i].VARICES<>0) then
            temp:=temp+1;
          if (abs(Hepatit[i].BILIRUBIN-Hepatit[j].BILIRUBIN)<=1.0) and
(Hepatit[i].BILIRUBIN<>0) then
            temp:=temp+1;
          if (abs(Hepatit[i].ALKPHOSPHATE-Hepatit[j].ALKPHOSPHATE)<=300)
and (Hepatit[i].ALKPHOSPHATE<>0) then
```

```

        temp:=temp+1;
        if (abs(Hepatit[i].SGOT-Hepatit[j].SGOT)<=40) and
(Hepatit[i].SGOT<>0) then
            temp:=temp+1;
            if (abs(Hepatit[i].ALBUMIN-Hepatit[j].ALBUMIN)<=5.2) and
(Hepatit[i].ALBUMIN>=3.8) then
                temp:=temp+1;
                if (abs(Hepatit[i].PROTIME-Hepatit[j].PROTIME)<=8) and
(Hepatit[i].PROTIME<>0) then
                    temp:=temp+1;
                    if (Hepatit[i].HISTOLOGY=Hepatit[j].HISTOLOGY) and
(Hepatit[i].HISTOLOGY<>0) then
                        temp:=temp+1;

                if temp<>0 then
                    begin
                        Matrix[i,j]:=temp;
                        Matrix[j,i]:=Matrix[i,j];
                    end;

            end;

        end;
    end;
end;

```

Algoritma 2: procedure TForm1.DereceMatrisiOlustur;

```

var
    i, j : integer;
begin
    for i:=1 to MaxVertex do
        for j:=1 to MaxVertex do
            DereceMatris[i,i]:=0;

    for i:=1 to MaxVertex do
        for j:=1 to MaxVertex do
            if (Matrix[i,j]>0) and (i<>j) then
                DereceMatris[i,i]:=DereceMatris[i,i]+1;
    end;

```

Algoritma 3: procedure TForm1.LaplaceMatrisiOlustur;

```

var
    i, j : integer;
begin
    for i:=1 to MaxVertex do
        for j:=1 to MaxVertex do
            LaplaceMatris[i,j]:=DereceMatris[i,j]-Matrix[i,j];
    end;

```

Algoritma 4: procedure IdentityMatrix;

```
var
  i,j : integer;
begin
  for i:=1 to Max do
    for j:=1 to Max do
      V[i,j]:=0.0;
    for i:=1 to Max do
      V[i,i]:=1.0;
    end;
  end;
```

Algoritma 5: function RMS:real;

```
var
  i : integer;
  sum : real;
begin
  sum:=0.0;
  for i:=1 to Max do
    sum:=sum+Matrix[i,i]*Matrix[i,i];
  RMS:=sqrt(sum/Max);
end;
```

Algoritma 6: procedure ZeroOut;

```
var
  j : integer;
  Theta,t,c,s,r : real;
begin
  Theta:=(Matrix[q,q]-Matrix[p,p])/(2*Matrix[p,q]);
  if Theta<0.0 then
    t:=-1.0
  else t:=1.0;
  t:=t/(abs(Theta)+sqrt(Theta*Theta+1));
  c:=1.0/sqrt(t*t+1);
  s:=c*t;

  for j:=1 to Max do
    begin
      VectorXP[j]:=Matrix[j,p];
      VectorXQ[j]:=Matrix[j,q];
    end;
  end;
```

```

Matrix[p,q]:=0.0;
Matrix[q,p]:=0.0;
Matrix[p,p]:=c*c*VectorXP[p]+s*s*VectorXQ[q]-2*s*c*VectorXQ[p];
Matrix[q,q]:=s*s*VectorXP[p]+c*c*VectorXQ[q]+2*c*s*VectorXQ[p];

for j:=1 to Max do
  if ((j<>p) and (j<>q)) then
    begin
      Matrix[j,p]:=c*VectorXP[j]-s*VectorXQ[j];
      Matrix[p,j]:=Matrix[j,p];
      Matrix[j,q]:=c*VectorXQ[j]+s*VectorXP[j];
      Matrix[q,j]:=Matrix[j,q];
    end;

for j:=1 to Max do
begin
  VectorYP[j]:=V[j,p];
  VectorYQ[j]:=V[j,q];
end;

for j:=1 to Max do
begin
  V[j,p]:=c*VectorYP[j]-s*VectorYQ[j];
  V[j,q]:=s*VectorYP[j]+c*VectorYQ[j];
end;
end;

```

Algoritma 7: procedure FindEigenVectors;

```

var
  Flag : boolean;
  CountS : integer;
  T2 : real;
begin
  GetLaplaceMatrix;
  IdentityMatrix;
  CountS:=0;
  repeat
    T2:=RMS;
    CountS:=CountS+1;
    Flag:=false;

```



```

    for p:=1 to Max-1 do
        for q:=p+1 to Max do
            if (abs(Matrix[p,q])/abs(T2))>Epsilon then
                begin
                    ZeroOut;
                    flag:=true;
                end;
        until ((not(Flag)) or (CountS>50));
    end;

```

Algoritma 8: procedure FindSecondSmallestEigenvector;

```

var
    i : integer;
    m1,m2 : real;
begin
    m1:=100.0;
    m2:=100.0;
    index:=2;
    for i:=1 to Max do
        if Matrix[i,i]<m1 then
            m1:=Matrix[i,i];

    for i:=1 to Max do
        if ((m2>Matrix[i,i]) and (m1<Matrix[i,i])) then
            begin
                m2:=Matrix[i,i];
                index:=i;
            end;
    end;
end;

```

Algoritma 9: procedure SpectralPartition;

```

begin
    GetLaplaceMatrix;
    IdentityMatrix;
    FindEigenVectors;
    FindSecondSmallestEigenvector;
end;

```