T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

BAĞLANIRLIK TABANLI ÖBEKLEME İÇİN JEODEZİK UZAKLIK KESTİRİMİ

Kadir GÜZEL

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Danışman

Doç.Dr. Olcay KURŞUN

Mayıs, 2015

İSTANBUL

Bu çalışma 24/06/2015 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği programında Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Jürisi:

İmza
Doç.Dr. Olcay KURŞUN
İstanbul Üniversitesi
Bilgisayar Mühendisliği

İmza
Prof.Dr. Ahmet SERTBAŞ
İstanbul Üniversitesi
Bilgisayar Mühendisliği

İmza
Doç.Dr. Atakan KURT
İstanbul Üniversitesi
Bilgisayar Mühendisliği

İmza
Yrd. Doç.Dr. Tolga ENSARİ
İstanbul Üniversitesi
Bilgisayar Mühendisliği

İmza
Yrd. Doç.Dr. Okan ŞAKAR
Bahçeşehir Üniversitesi
Bilgisayar Mühendisliği

ÖNSÖZ

Bu çalışmayı hazırlarken her türlü yardımı ve desteği fazlasıyla sağlayan, hiçbir şekilde esirgemeyen saygıdeğer hocam Sayın **Doç. Dr. Olcay KURŞUN** 'a, okul hayatım boyunca bana destek olan tüm arkadaşlarıma ve yaşantım boyunca her daim yanımda olan, bana sevgi, güven ve her türlü desteği veren aileme en içten teşekkürlerimi sunarım. Bu tez 114E071 numaralı "Makine Öğrenmesi Yöntemleriyle Kortikal Katman-4 İmge Gösteriminin Kapasitesinin Ölçülmesi" başlıklı Tübitak 3001 projesi kapsamında desteklenmektedir.

Mayıs, 2015 Kadir GÜZEL

İÇİNDEKİLER

ŞEKİL LİSTESİ

TABLO LİSTESİ

SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
ε	: Epsilon
X	: Rastsal Değişken
d	: Girdi(boyut) sayısı
$\sum$	: Değişinti toplamı
σ	: Sigma
y	: Çıktı
D	: Derece Matrisi
W	: Benzerlik Matrisi
L	: Laplacian Matrisi
Λ	: Lamda (Özdeger)
I	: Ortak Bilgi
H	: Entropi
N	: Düğüm Sayısı
p	: Değişken Sayısı

Kısaltmalar	Açıklama
SÖ	: Spektral Öbekleme
YSÖ	: Yol Tabanlı Spektral Öbekleme
HÖ	: Hibrit Öbekleme
LTÖ	: Laplacian Toplamlı Öbekleme
LBÖ	: Laplacian Birleştirmeli Öbekleme
MST	: Minimum Örten Ağaç
ARI	: Adjusted Rand Index
MI	:Mutual Information
TBA	:Temel Bilesen Analizi
ÇBÖ	:Çok Boyutlu Ölçekleme

ÖZET

YÜKSEK LİSANS TEZİ

BAĞLANIRLIK TABANLI ÖBEKLEME İÇİN JEODEZİK UZAKLIK KESTİRİMİ

Kadir GÜZEL

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç.Dr. Olcay KURŞUN

Spektral öbekleme, son zamanlarda popülerleşen, küresel şekille sınırlı olmayan gelişigüzel/uzatılmış öbekler verebilen bir öbekleme yöntemidir. Çizge tabanlı bu öbekleme yönteminin girdi olarak kullandığı örnekler arasındaki benzerliklerin belirlenmesi için Öklid uzaklığı tabanlı yaklaşımlara ek olarak çizge üzerinde diğer örneklerin dağılımının da etkili olduğu en kısa yol ya da büyük atlamalar yapmayan yollar kullanılarak da bağlanırlık-tabanlı benzerlik ölçütleri de kullanılmaktadır. Bu çalışmada spektral öbeklemenin kullandığı Öklid uzaklığı ile yol-tabanlı spektral öbeklemenin kullandığı en küçük maksimum atlama uzaklığı birleştirilerek, hibrit bir yöntem önerilmiştir. Laplacian matrisleri benzeştirilerek ve birleştirilerek hibrit Laplacian yöntemi oluşturulmuştur ve daha gürbüz olduğu gösterilmiştir.

Mayıs, 2015, 59 sayfa

Anahtar kelimeler: Spektral öbekleme, Yol-tabanlı öbekleme; Floyd-Warshall En kısa yol algoritması; Topluluk öbekleme.

SUMMARY

M.Sc. THESIS

ESTIMATION OF GEODESIC DISTANCES FOR CONNECTIVITY BASED CLUSTERING

Kadir GÜZEL

İstanbul University

Institute of Graduate Studies in Science and Engineering

Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Olcay KURŞUN

Spectral clustering is a recently popular clustering method, not limited to spherical-shaped clusters and capable of finding elongated arbitrary-shaped clusters. This graph theoretical clustering method can use Euclidean distance between each pair of examples as well as connectivity-based similarity measures based on shortest path or paths that do not travel over examples with big distances on the graph. In this thesis, a hybrid method is proposed that utilizes distances used by spectral and path-based spectral clustering algorithms. By combining and appending Laplacian matrices, hybrid methods have been proposed and shown to be more robust than other both methods.

May, 2015, 59 pages

Keywords: Spectral clustering, Path-based clustering; Floyd-Warshall Shortest path algorithm; Ensemble clustering

1. GİRİŞ

İnsanlar tarihin başlangıcından beri kullandıkları eşyaları kümeleme ya da gruplama ihtiyacı duymuşlardır. Bu kümeleme ve gruplama bazen eşyaların özelliklerine göre yapılmış, bazen de özellikleri göz ardı edilerek rastgele yapılmıştır. Özelliklerine göre yapılan gruplamalarda, aynı grupta olan öğelerin baz alınan özelliklere göre benzer, farklı grupta olan öğelerin baz alınan özelliklere göre farklı olması amaçlanmıştır [1].
Bu çalışmada hazırlanan tez beş bölümden oluşmaktadır.

Giriş bölümünde; kümeleme analizi ilgili tanım ve kavramlar, çizge oluşturma teknikleri, kümeleme analizinin kullanım alanları, boyut azaltma yöntemleri, kümeleme analizi yöntemleri, uzaklık/yakınlık ölçüleri ve modellerin birleştirilmesi incelenecektir. Kümeleme analizi ile ilgili literatür taramasına yer verilecektir. Kümeleme analizi ile ilgili önceki yapılan çalışmalar incelenecek ve literatürde kullanılan popüler öbekleme yöntemleri uygulamalı olarak incelenmiştir.

Bölüm 3'te ikinci kısımda bahsedilen literatürde kullanılan K-Merkez, Spektral Öbekleme (SÖ) ve çalışma kapsamında kod uygulaması tarafımızdan geliştirilen Yol Tabanlı Spektral Öbekleme (YSÖ) ile beraber yeni önerilen Hibrit Öbekleme (HÖ) , Laplacian Tanımlı Öbekleme (LTÖ) ve Kritik Noktalı Yol Tabanlı Spektral Öbekleme yöntemlerin gerçek UCI veri kümelerine uygulanmıştır. Çalışmada önerilen öbekleme yöntemlerinden detaylı bahsedilmiştir.

Bölüm 4'te; edilmiş anlamlı sonuçlardan yola çıkılarak önerilen kümeleme algoritmaları uygulamaları karşılaştırılmış ve deney sonuçları ile önerilen algoritmalar arasındaki farklar ortaya koyulmaya çalışılmıştır.

Bu çalışmanın son kısmı olan 5. bölümde ise çalışmada yeni önerilen yöntemlerin kümeleme analizi yöntemlerine göre zayıf ve güçlü yanlarına tartışılmış ve bu araştırmalardan yeni bulgu ve sonuçlar elde edilmiştir.

2. GENEL KISIMLAR

Bu bölümde kümeleme analizinde kullanılacak benzerlik çizge(graf) türleri, boyut azaltma yöntemleri, farklı tür kümeleme yöntemleri ifade edilecek ve kümelemede kullanılan başarı ölçütlerinden bahsedilecektir.

2.1. KÜMELEME ANALİZİ

Kümeleme istatistik, bilgisayar bilimleri, sosyal veya psikolojik biyolojiden değişen uygulamaları araştırmak için çok tercih edilen bir istatistiksel analiz yöntemi olarak karşımıza çıkmaktadır. Kümeleme analizinde herhangi bir eğitim kümesi kullanılmaz. Eğitimsiz öğrenme yapılır. Kümeleme, veri matrisinde yer alan ve doğal gruplamaları kesin olarak bilinmeyen birimleri, değişkenleri ya da birim ve değişkenleri birbiri ile benzer olan alt kümelere (grup, sınıf) ayırmaya yardımcı olan yöntemler topluluğudur. Saklı kalmış örüntülerin keşfedilmesini ve büyük boyutlu veri kümeleri içerisinden en hızlı şekilde bilgiye erişilmesini sağlar [2].

Nesneler küme içerisinde çok benzer biçimde, kümeler arasında farklı olacak biçimde kümeler. Kümeleme işlemi başarılı olursa, bir geometrik çizim yapıldığında nesneler küme içerisinde birbirine çok yakın, kümeler ise birbirinden uzak olacaktır. Kümeleme analizi kendisi belirli bir algoritma değil, ama genel çözüm üretir. Veri nesneleri bir gruptur, aralarında ortak bir payda bulunmaktadır. Ancak, farklı araştırmacılar farklı kümelenme modellerinin her biri için farklı algoritmalar uygularlar.

Kümeleme analizi için başka bir tanım da şu biçimde yapılmaktadır. Birimleri, değişkenler arası benzerlik ya da farklılıklara dayalı olarak hesaplanan bazı ölçülerden yararlanarak homojen gruplara bölmek belirli prototipler tanımlamak amacıyla kullanılan teknikler grubudur. Verilerin gruplar veya kümeler altında toplanarak, benzer özelliklere sahip nesnelerin bir araya gelmesini sağlayan bir veri madenciliği tekniğidir.

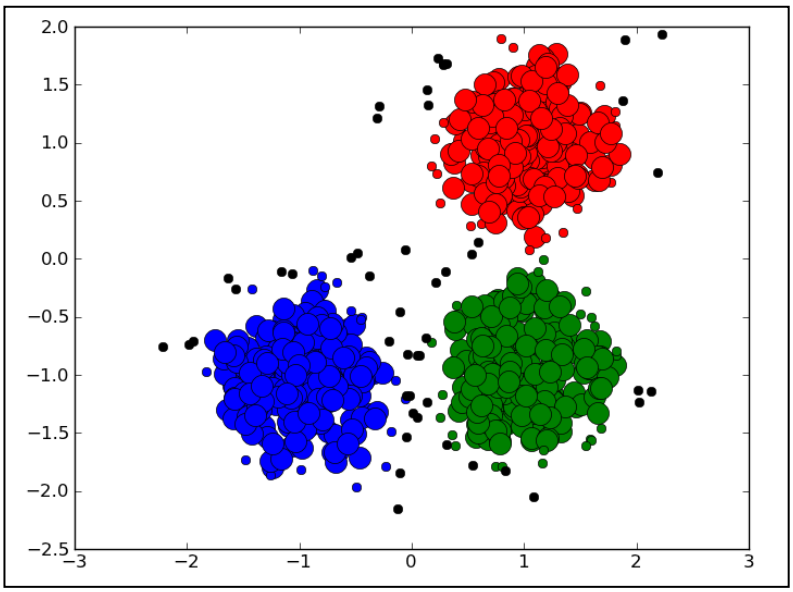
Kümeleme analizi, temel olarak dört değişik amaca yönelik işlev yerine getirir.

a) *n* sayıda birimi, nesneyi, oluşumu *p* değişkene göre saptanan özelliklerine göre olabildiğince kendi içinde türdeş ve kendi aralarında farklı alt gruplara ayırmak,

b) *p* sayıda değişkeni, *n* sayıda birimde saptanan değerlere göre ortak özellikleri açıkladığı varsayılan alt kümelere ayırmak ve ortak faktör yapıları ortaya koymak,

c) Hem birimleri hem de değişkenleri birlikte ele alarak ortak *n* birimi *p* değişkene göre ortak özellikli alt kümelere ayırmak,

d) Birimleri, *p* değişkene göre saptanan değerlere göre, izledikleri biyolojik ve tipolojik sınıflamayı ortaya koymak [3].



Şekil 2.1: Öbeklere ayrılmış örnek veri kümesi.

Yukarıdaki açıklamadan da anlaşılacağı gibi kümeleme analizi çok sayıda değişik işlevi yerine getiren yöntemler topluluğudur. Bu nedenle farklı amaçlar için farklı yöntemler uygulanır. Ayrıca değişkenlerin ölçü birimlerinin ve ölçümleme tekniklerinin farklı olmasından dolayı birimlerinin benzerliklerinin ortaya konmasında da değişik ölçüler kullanılır.

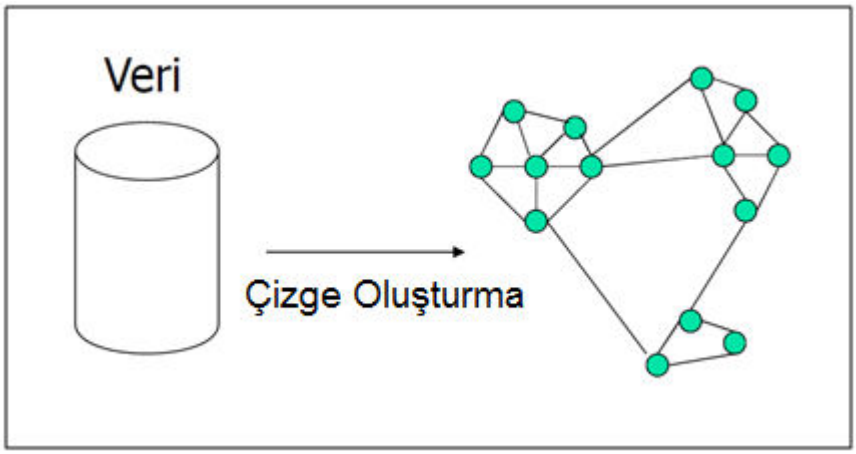
Kümeleme analizinde, heterojen yapıdaki verinin küme sayısı ve küme yapısı araştırılır. Kümelemede amaç, her bir küme içerisindeki gözlemlerin ya da nesnelerin birbirlerine benzer ve kümelerin de birbirinden farklı olacak şekilde en uygun gruplama yapısını bulmaktır. Kümeleme analizi temel olarak ayrıştırma analizinden farklıdır. Ayrıştırma

analizinde gözlemler daha önceden tanımlanmış ve sayısı bilinen gruplara parçalanırken, kümeleme analizinde ne grup sayısı hakkında ne de grupların yapısı hakkında bir ön bilgi bulunmaz. Gözlemlerin kümelere gruplanması için geliştirilen bazı yöntemlerde kümeleme, tüm gözlem çiftleri arasındaki benzerliklerin bulunmasıyla başlar. Bazı durumlarda benzerlikler, uzaklık ölçümlerine dayalı olarak bulunur. Diğer kümeleme yöntemlerinde, küme merkezlerinin seçimi veya küme içi ve kümeler arası değişimin karşılaştırılması yapılır. Değişkenlerin de kümelenmesi mümkündür [4] .

2.2. ÇİZGE NOTASYONU

Çizge teorisinin uygulamaları modern hayatın karmaşık ve geniş kapsamlı birçok probleminin çözümü için kullanılmaktadır. Çizge teorisi problemleri tanımlama ve yapısal olarak ilişkileri belirlemekte de faydalıdır.

Bir çizge düğüm olarak adlandırılan noktalar ve her biri bu noktaları veya sadece noktanın kendisini birleştiren ve kenar olarak adlandırılan çizgiler topluluğudur. Örnek olarak şehirleri düğüm (vertex) ve onları bağlayan yolları kenar (edge) olarak gösteren yol haritaları verilebilir. Bir çizgeyi tanımlamak için öncelikle düğümlerin ve ayrıtların kümesini tanımlamamız gerekir. Daha sonra hangi ayrıtların hangi düğümleri bağladığını belirtmeliyiz. Bir kenar her iki ucunda da bir düğüm olacak şekilde tanımlandığından çizgedeki tüm ayrıtların uç noktalarını bir düğüm ile ilişkilendirmek gerekir.



Şekil 2.2: Veri kümesinden çizge (graf) oluşturulması.

- Veri kümesi $D=\{X_1, X_2, ..., X_p\}$
- Veri kümesi ağırlıklı, yönsüz, bağlı bir çizge ile temsil edilir.

2.2.1. Çizge Oluşturma Türleri

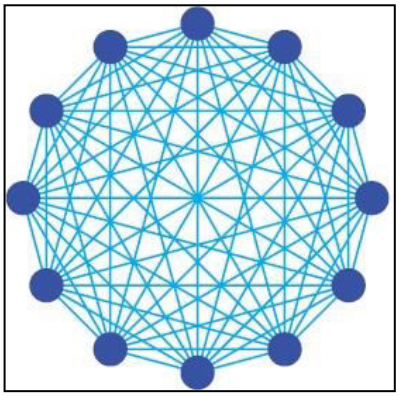
Epsilon-komşu çizge: Düğümler arası ikili mesafeleri ε daha küçük olan düğümlerin birbirine bağlayarak oluşturulan çizgelerdir. Bağlı olan tüm noktaları arasındaki mesafeler (çoğu ε de) aşağı yukarı aynı ölçek olduğundan, ağırlık kenarları çizge hakkında fazla bilgi vermez. Bu nedenle, ε-komşuluk grafiği genellikle ağırlıklandırılmamış çizge olarak kabul edilir [2].

***K*-en yakın komşu çizge:** Bu çizge oluşturmadaki temel hedef v_i düğümünün istenen sayıdaki *(k)* komşuları arasında v_j var ise v_i ile v_j birbirlerine bağlanırlar. Veri kümesindeki nesnelerin *k* en yakın komşuları arasında ayrıtlar oluşturulur.

Şekil 2.3: Swiss-Roll veri kümesi için 5-komşulu çizge oluşturma.

Oluşan ilişki simetrik değildir. İşaretlemiş olduğu komşuluklara en yakın düğümü bünyesine katarak ilerler.

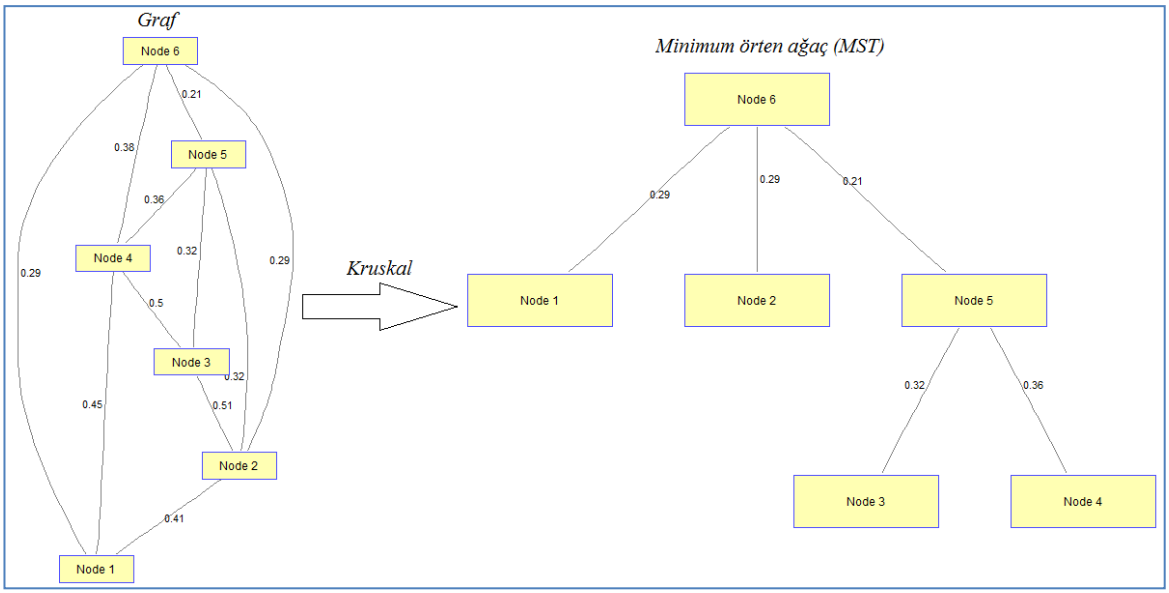
Tam bağlı çizge: Bütün nesneler arasında benzerlik hesaplanır ve bu benzerlik ile ağırlıklandırılmış ayrıtlar oluşturulur. Bu tip çizgelerde benzerlik matrisi Gauss fonksiyonu kullanılarak oluşturulur. Oluşan ilişki simetriktir.



Şekil 2.4: Tam bağlı çizge oluşturulması.

2.2.2. Minimum Örten Ağaç (Minimum Spanning Tree)

Minimum örten ağaç çevrim içermeyen bağlantılı bir çizgedir. Bir ağaç üzerinde N tane düğüm ve *N-1* tane kenar (edge) bulunur. Bu yüzden, bir ağaç üzerinde bir düğümden başka bir düğüme gitmek için sadece tek bir yol mevcuttur. Minimum örten ağaç (MST) ise, üzerinden bir çizgedeki tüm düğümlere ulaşılabilen ağaçlar içerisinde, toplam ağırlığı en düşük olan ağaçtır. Bütün düğümleri dolaşan en kısa yolu verir.

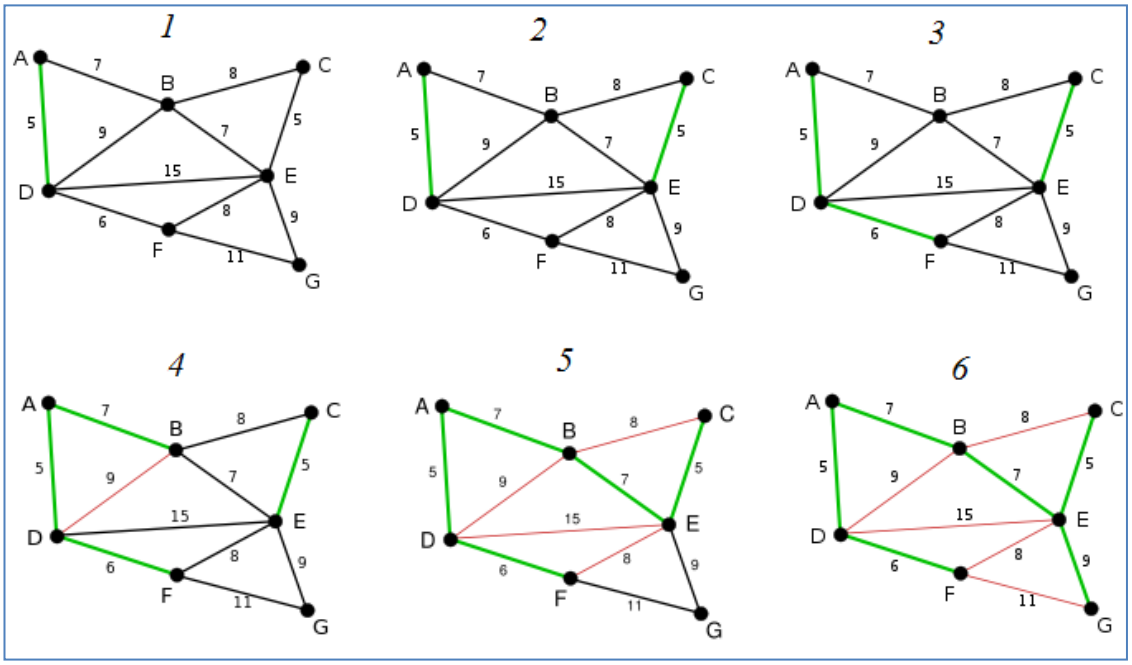


Şekil 2.5: Kruskal algoritması kullanılarak Matlab'ta MST uygulaması.

Greedy(Açgözlü) algoritmaları kullanılarak minimum örten ağaç bulunur. 3 tane Greedy algoritması vardır. Bunlar Kruskal, Pirim-Jarnik, Djkistra bunlardan en fazla kullanılanlar Kruskal ve Pirim-Jarnik'tir[29]. Günlük hayattan bir örnek MST 'i anlamamıza daha fazla yardımcı olacaktır. Bir kablolu TV şirketi düşünelim. Bu şirket

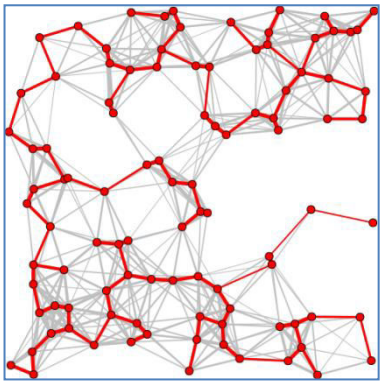
bir mahalleye kablo döşemek istesin ve belirli evlere uğramak zorunda olsun. Bütün evleri dolaşması için gerekli kabloları en az maliyetle döşemesi gerekir. Bunun için de bütün evleri dolaşan en az maliyetli yolu bulması gerekir. Aksi takdir de gereksiz kablo döşemiş olacak ve zarara uğrayacaktır. İşte minimum örten ağaç burada devreye giriyor. Minimum örten ağaç yardımıyla bütün evleri dolaşan en kısa yol bulunur ve işlem en az maliyetle çözülmüş olur

Kruskal algoritması bağlı düğümler içerisinde en kısa şekilde tüm düğümleri dolaşmayı sağlar. Algoritma en küçük yolları alıp bir döngü(loop) yapmadan tüm düğümleri en kısa yoldan nasıl dolaşılabileceğini bulur. Tüm düğümlerin bağlı ve yolların çift yönlü tek ağırlıklı olması gerekmektedir yani A düğümünden B düğümüne gitmenin maliyeti x ise B düğümünden A düğümüne gitmenin de maliyetinin x olması gerekmektedir.



Şekil 2.6: Kruskal algoritması adımları.

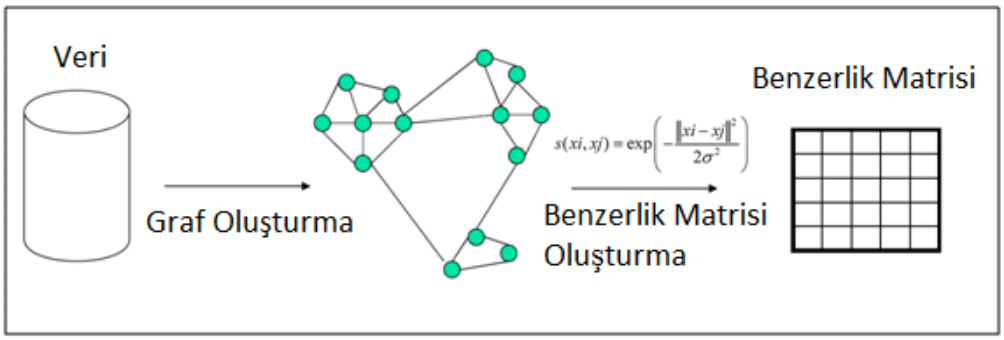
Kruskal algoritmasında bütün yollar listelenip küçükten büyüğe doğru sıralanır. Liste çıkarıldıktan sonra sırasıyla en küçükten en büyüğe doğru komşuluklar işaretlenir. Bu işaretleme sırasında düğüm grupları ve grupların birbiri ile ilişkisine dikkat edilir. Yani şayet listedeki iki düğüm harfi de aynı node ise bu bağlantı atlanır. Tüm komşuluklar bittikten sonra minimum örten ağaç ortaya çıkmış olur.



Şekil 2.7: Minimum örten ağaç örneği.

2.2.3. Benzerlik Matrisi

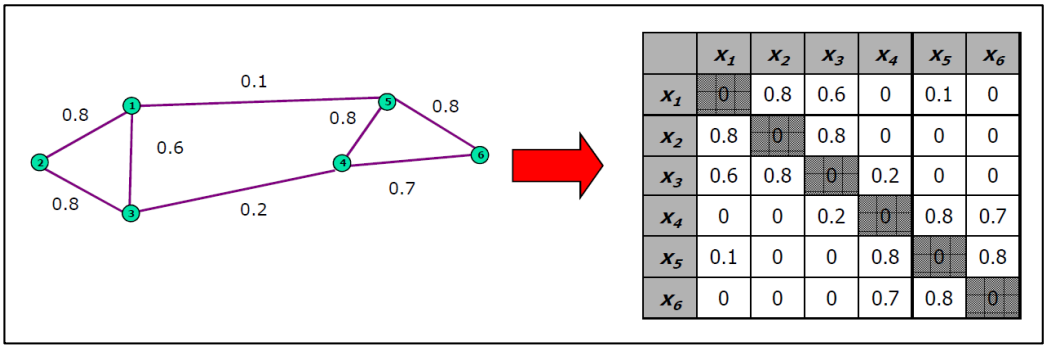
Bir veri setinde yer alan birimlerin kümelenmesi işlemi bu birimlerin birbirleriyle olan benzerlikleri ya da birbirlerine olan uzaklıkları kullanılarak gerçekleştirilmektedir. Öklid uzaklığı formülleri standartlaştırılmış verilerle değil, işlenmemiş verilerle hesaplama yapılır. Öklid uzaklıkları kümeleme analizine sıra dışı olabilecek yeni nesnelerin eklenmesinden etkilenmezler. Ancak boyutlar arasındaki ölçek farklılıkları Öklid uzaklıklarını önemli ölçüde etkilemektedir. Öklid uzaklık formülü en yaygın olarak kullanılan uzaklık hesaplama formülüdür.



Şekil 2.8: Veri kümesinden çizge (graf) oluşturulması.

$$s(xi,xj)=\exp\left(-\frac{\|xi-xj\|^2}{2\sigma^2}\right) \tag{2.1}$$

En çok kullanılan 1. denklemdeki Gaussian fonksiyonu ile her düğümün diğer düğümlere olan Öklid uzaklıklarına ters orantılı bir şekilde benzerlikler belirlenir.



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0.8	0.6	0	0.1	0
x_2	0.8	0	0.8	0	0	0
x_3	0.6	0.8	0	0.2	0	0
x_4	0	0	0.2	0	0.8	0.7
x_5	0.1	0	0	0.8	0	0.8
x_6	0	0	0	0.7	0.8	0

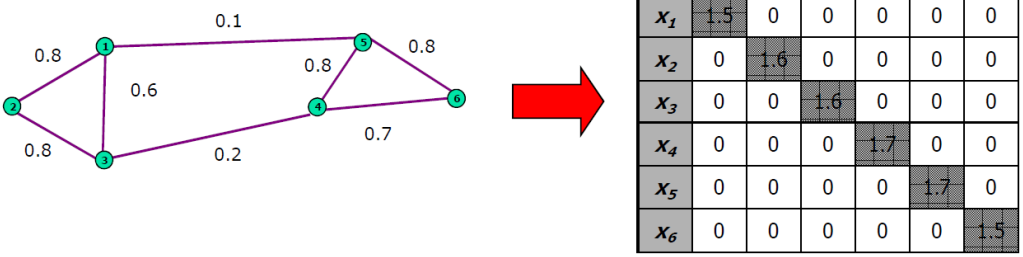
Şekil 2.9: Benzerlik matrisinin oluşturulması.

Benzerlik matrisi *(W)* simetrik bir matris elde edilir. Bu matrisin boyutu *N*×*N* matris (*N*: düğüm sayısı). $W=[w_{ij}]$: x_i ve x_j düğümleri arasındaki ayrıtın ağırlığı matrise yazılarak benzerlik matrisi oluşturulur.[5]

2.2.4. Derece Matrisi

Derece matrisi *(D)* *N*×*N* diagonal bir matristir. Çizgedeki bir düğümden diğer düğümlere olan ayrıtların 2. denklemde de gösterildiği gibi ağırlıklarının toplamı alınarak hesaplanır.

$$D(i,i) = \sum w_{ij} \tag{2.2}$$



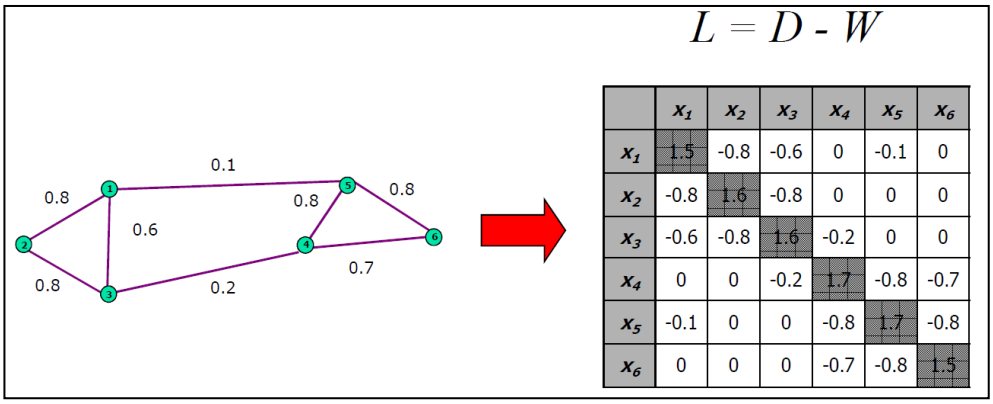
	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.5	0	0	0	0	0
x_2	0	1.6	0	0	0	0
x_3	0	0	1.6	0	0	0
x_4	0	0	0	1.7	0	0
x_5	0	0	0	0	1.7	0
x_6	0	0	0	0	0	1.5

Şekil 2.10: Derece matrisinin oluşturulması.

Matematiksel alanında derece matrisi, her köşe her düğüme bağlı kenarları sayısı derecesi hakkında bilgi içeren bir köşegen matristir [6]. Yönlendirilmiş çizgede terim derece (her köşenin de gelen kenarların sayısı) veya her köşenin de giden kenar sayısına sevk edebilir.

2.2.5. Laplacian Matris

Laplacian matrisi (L) ; $N \times N$ simetrik bir matristir. Laplacian matrisinin özdeğerler ve özvektörleri çizge yapısı hakkında bilgi verir.



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.5	-0.8	-0.6	0	-0.1	0
x_2	-0.8	1.6	-0.8	0	0	0
x_3	-0.6	-0.8	1.6	-0.2	0	0
x_4	0	0	-0.2	1.7	-0.8	-0.7
x_5	-0.1	0	0	-0.8	1.7	-0.8
x_6	0	0	0	-0.7	-0.8	1.5

Şekil 2.11: Laplacian matrisinin oluşturulması.

D(G), G çizgesinin köşegen elemanları noktaların derecelerinden oluşan bir köşegen matris olmak üzere *L(G)=D(G)-W(G)* şeklinde tanımlanan matris *G* çizgesinin Laplacian matrisi olarak adlandırılır. Bilgisayar bilimlerinin de içinde bulunduğu pek çok bilim ve mühendislik alanında kullanılan çizge teorisi (graph theory) açısından önemli bir matristir. Laplacian matrisinin özelliği her düğümün derecesini (node order) ve diğer düğümlerle olan komşuluk ilişkisini (adjacency list) tutmasıdır.

Özellikleri:

- Özdeğerler pozitif gerçel sayılardır
- En küçük özdeğer sıfırdır.
- Özdeğerler ve özvektörler çizge yapısı hakkında bilgi verir.

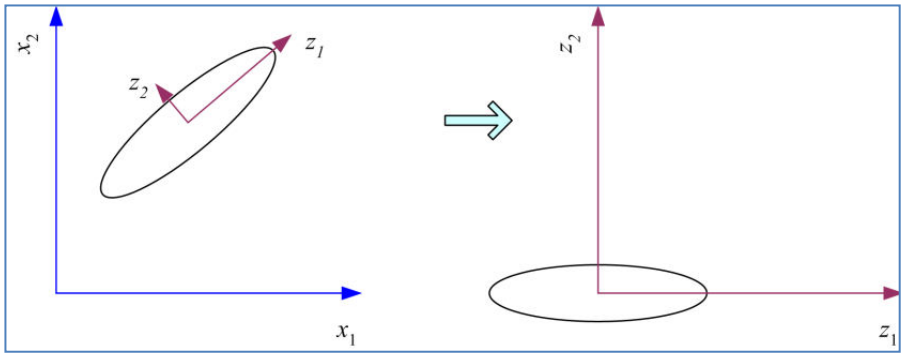
2.3. BOYUT AZALTMA YÖNTEMLERİ

Çoğu öğrenme algoritmalarinda karmaşıklık, veri örnekleme büyüklügü (N) ve girdi boyut sayısına (p) bağlıdır. Gereken bellek ve yapılacak işlem sayısını azaltmak için problemin boyut sayısını azaltmak isteriz. Bir girdinin gereksiz olduğunu anladığımızda onu elde etmek yada ölçmek için gereken uğraş ve bedelden tasarruf ederiz. Veri kümesi daha az değişkenle açıklandığında bu verileri üreten süreci daha iyi anlayabiliriz, bu da bilgi çıkarımı yapabilmemizi kolaylaştırır.

Boyut azaltmak için iki temel yöntem vardir. Öznitelik seçimi (Feature Selection) ve öznitelik çıkarımı (Feature Extraction). Öznitelik seçiminde *d* değişkeninden en çok bilgi içeren *k* tanesi bulunur ve diğerleri atılır (*d*--*k*). Öznitelik çıkarımında ise asıl d değişkeni birleştirilerek *k*<*d* tane yeni değişken oluşturulur. Bu tez çalışmada en iyi bilinen ve en çok kullanılan boyut azaltma yöntemleri temel bileşen analizi (PCA) ve çok boyutlu ölçekleme (MDS) kullanılmıştır.

2.3.1. Temel Bileşen Analizi (Principal Component Anlaysis)

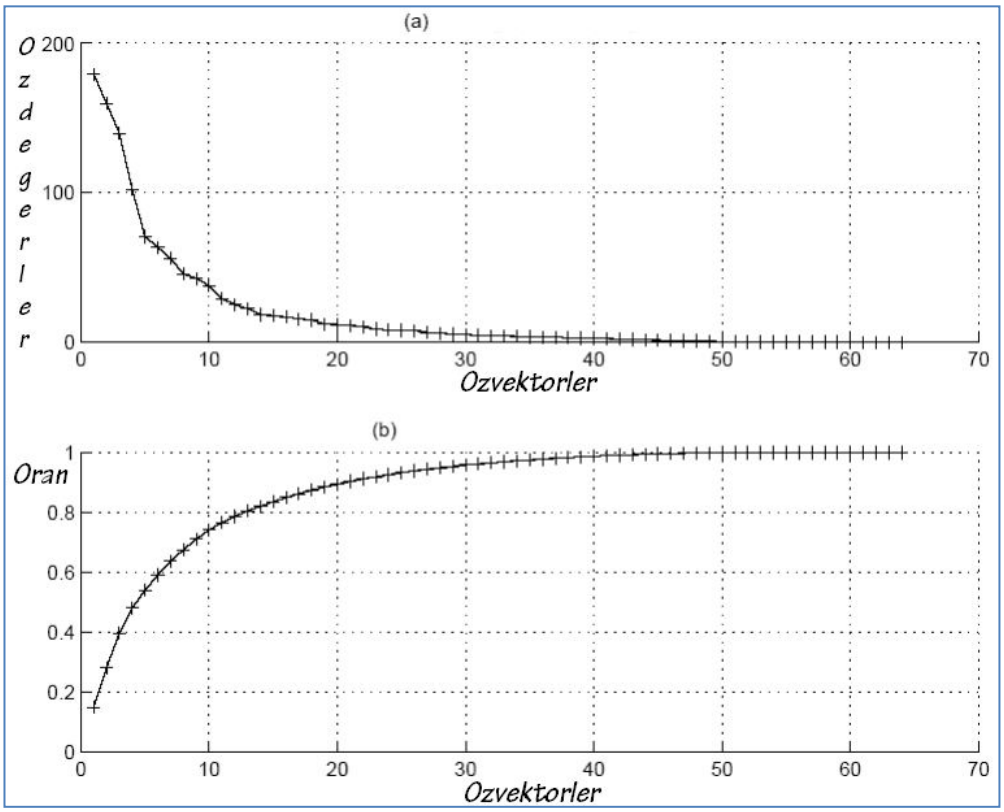
Orijinal *P* değişkeninin varyans yapısını daha az sayıda ve bu değişkenlerin doğrusal bileşenleri olan yeni değişkenlerle ifade etme yöntemidir. Aralarında korelasyon bulunan *P* sayıda değişkeni açıkladığı yapıyı, aralarında korelasyon bulunmayan ve sayıca orijinal değişken sayısından daha az sayıda (*p*>*k*) orijinal değişkenlerin doğrusal bileşenleri olan değişkenlerle ifade etme yöntemine denir. Veri matrisinde yer alan *P* değişkenin doğrusal bileşenlerini bulmak için kovaryans matrisinin ya da korelasyon matrisinin özdeğerleri ve özvektörleri kullanılır. Eğer değişkenler aynı birim veya karşılaştırılabilir birimlerdeyse, değişken varyansları aynı boyuttaysa varyans-kovaryans matrisi kullanılır. Bu durumlar sağlanmadığında varyans-kovaryans matrisi yerine korelasyon matrisi kullanılır. Çok boyutlu veri kumelerinde kullanilan bir boyut dusurme teknigidir. Kovaryans matrisinin özdeğerlerini bulurak veri kümesindeki varyansı maksimize ederek çalışır.



Şekil 2.12: Temel bileşen analizi örneklemi ortalar ve eksenleri en yüksek değişinti yönlerine rastlayacak biçimde döndürür.
*Eger Z_2yöneyinin değişintisi çok düşükse bu boyut göz ardı edilebilir ve böylece boyut sayısı ikiden bire iner.

Tüm özdeğerler sıfırdan büyük de olsa bazı özdeğerlerin değişintiye az katkısı olduğunu görüp onları atabiliriz. Bu durumda değişintinin, örneğin yüzde 90 dan fazlasını açıklayan en büyük k bileşenini alabiliriz. Özdeğerler azalan şekilde sıralanmış kabul edersek aşağıdaki açıklanan oran ile k seçilir.

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$



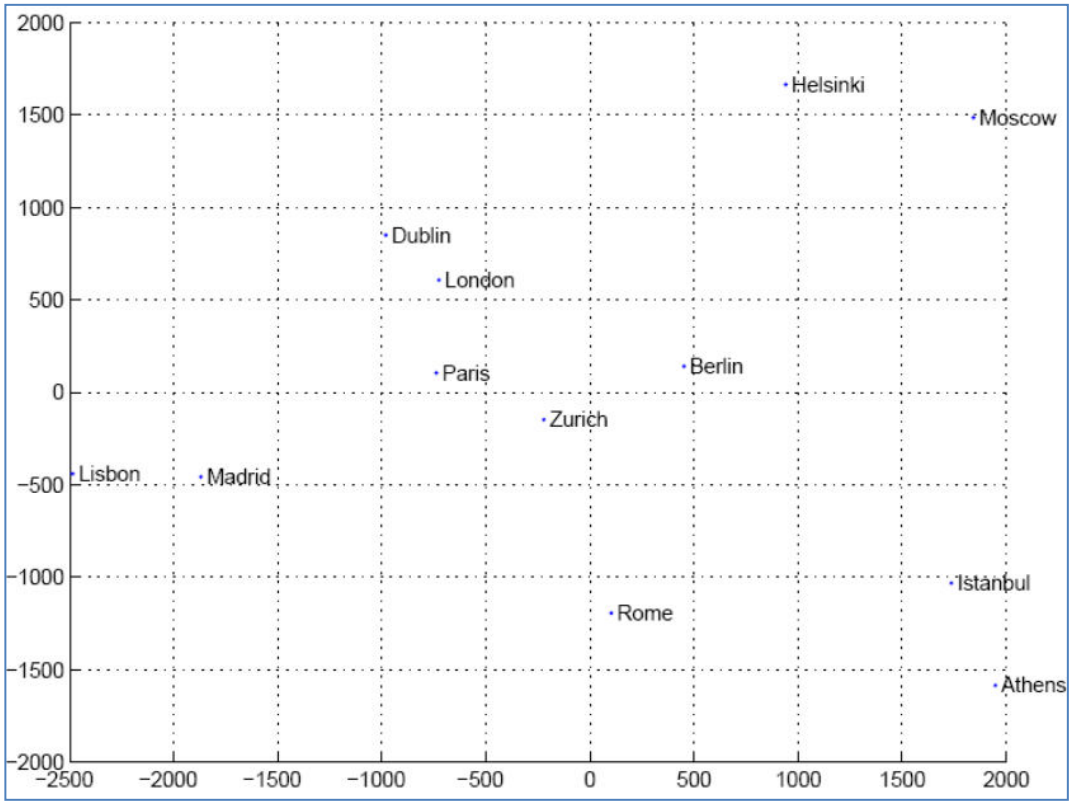
Şekil 2.13: UCI veritabanından alınan Optdigits[25] veri üzerinde (a) hesaplanan çizit ve (b) açıklanan değişinti oranı gösterilmektedir.

Yukarıdaki şekilde 10 sınıf ve 64 boyutlu girdiye sahip bir elle yazılmış rakamlar veri kümesidir. Ilk 20 özvektör değişintinin yüzde 90'ını açıklamaktadır. Temel bileşen analizinin üç temel amacı vardır

- Verilerin boyutunu azaltmak
- Tahminleme yapmak
- Veri setini, bazı ön analizler için görüntülemek.

2.3.2. Çok Boyutlu Ölçekleme (Multi Dimensional Scaling)

Elimizde *N* her nokta çifti arasindaki uzaklık bilgisi olsun ama bu noktaların kaç boyutlu bir uzayda, nerede konumlandıklarını ya da uzaklıkların nasıl hesaplandığını bilmiyor olalım. Çok boyutlu ölçekleme bu noktaları daha düşük boyutlu bir uzayda öyle bir biçimde yerlestirir ki bu yeni uzayda noktalar arasındakiöklid uzaklıkları verilen uzaklıklara olabildiğince yakın olur.Dolayısıyla bilinmeyen boyutlu bir uzaydan başka bir uzaya eşleme oluşur.

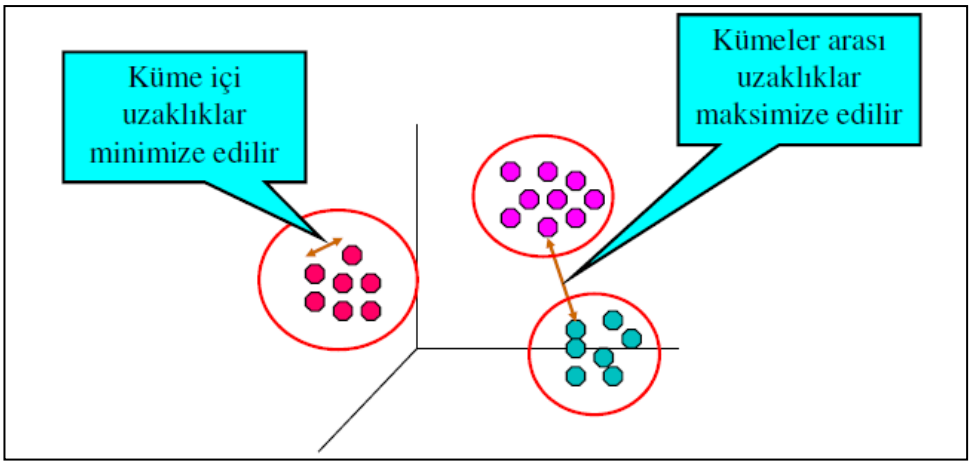


Şekil 2.14: Çok boyutlu ölçeklemeyle oluşturulan Avrupa haritası[22].

Şekil 2.14'te gösterilen kent çiftleri Arasındaki karayolu uzaklıkları girdi olarak verilmiş ve CBO bu kentleri aralarındaki uzaklıkları olabildiğince koruyacak biçimde iki boyuta yerleştirmiştir. Çok boyutlu ölçeklemenin en bilinen örneği, bir ülkedeki kentler arasındaki karayolu uzaklıklarını alıp bunları iki boyuta eşleyerek ülkenin haritasınıçıkarmaktır. Çok boyutlu ölçekleme boyut azaltma için kullanılabilir. *D* boyutlu uzayda nokta çiftleri arasındaki öklid uzaklıklarıyla *N*×*N* dizeyi oluşturup çok boyutlu ölçekleme girdi olarak veririz. Sonucunda çıkan düşük boyutlu uzaydaki konumu düşük girdili gösterimi olur[22].

2.4. ÖBEKLEME YÖNTEMLERİ

Öbekleme analizinde uzaklık/yakınlık ölçüleri kullanılarak oluşturulan uzaklık/yakınlık matrisindeki değerlerinden faydalanılarak birimlerin kümelere (gruplara) atanması yapılır. Araştırmacı hangi benzerlik/uzaklık ölçüsünü kullanacağına karar verdikten sonra, kümeleme işleminin nasıl olacağına karar vermek zorundadır. Birimlerin benzerliklerine göre kümelere dâhil edilmesinde kullanılabilecek çeşitli yaklaşımlar vardır. Bu yaklaşımlardan biri, en çok benzer iki birimi aynı gruba atamakla başlayıp tüm birimlerin aynı gruba atanması ile biten hiyerarşik bir yaklaşımdır. Bir başka yaklaşım ise tüm verilerin ortalama değerlerine en yakın değerlere sahip birimlerin aynı kümeye atanmasını esas alan yaklaşımdır.



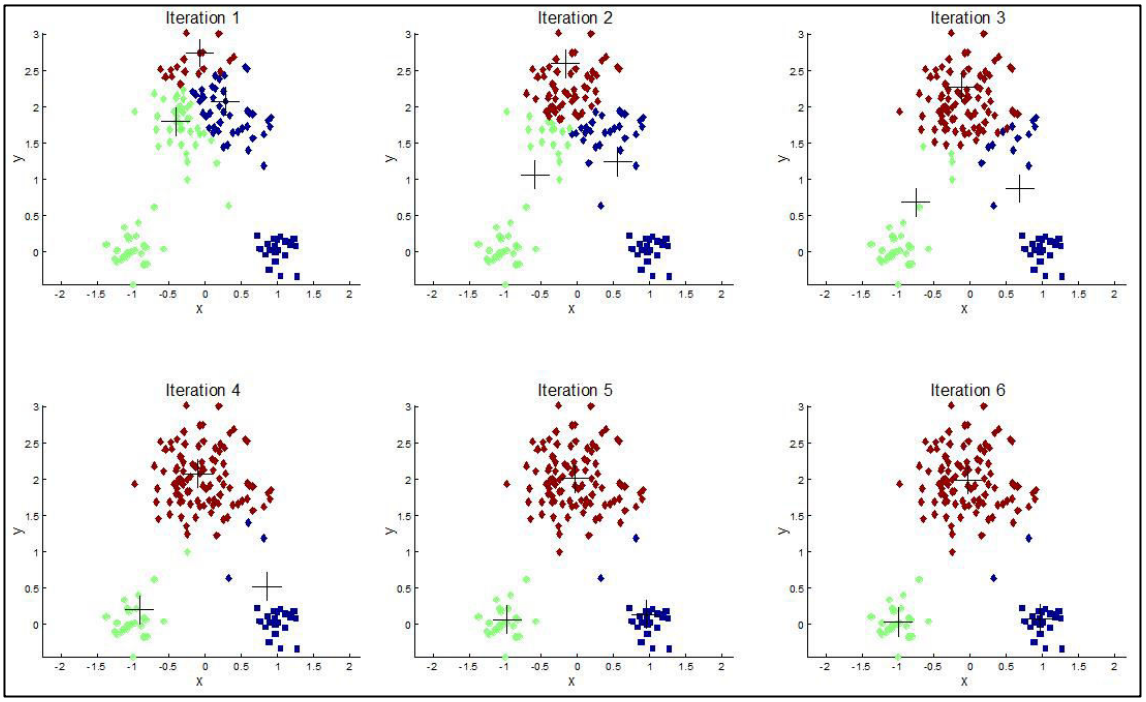
Şekil 2.15: Kümeleme yöntemlerinin amacı.

En çok kullanılan bu iki yaklaşım dışında diğer yaklaşımlar da mevcuttur. Tüm yaklaşımlarda en önemli ölçüt, kümeler arası farklar ile kümeler içi benzerliklerin maksimum olmasını sağlamaktır. En çok kullanılan kümeleme algoritmaları *K*-merkez, Spektral kümeleme ve hiyerarşik kümeleme altında 3 kategoride toplanmaktadır [7].

2.4.1. *K*- Merkez

En eski kümeleme algoritmalarından olan *K*-merkez, 1967 yılında J.B. MacQueen tarafından geliştirilmiştir. En yaygın kullanılan gözetimsiz öğrenme yöntemlerinden birisi olan *K*-merkezin atama mekanizması, her verinin sadece bir kümeye ait olabilmesine izin verir. Bu nedenle, keskin bir kümeleme algoritmasıdır. Merkez noktanın kümeyi temsil etmesi ana fikrine dayalı bir yöntemdir.

K-merkez algoritmasının genel mantığı *n* adet veri nesnesinden oluşan bir veri setini, giriş parametresi olarak verilen *K* adet kümeye bölümlemektir. Algoritmaya *K*-means adı verilmesinin nedeni, algoritmanın çalışmasından önce sabit bir küme sayısına ihtiyaç duyulmasıdır. Küme sayısı *K* ile gösterilir ve elemanlarının birbirlerine olan yakınlıklarına göre oluşacak grup sayısını ifade eder. Buna göre *K* önceden bilinen ve kümeleme işlemi bitene kadar değeri değişmeyen sabit bir pozitif tam sayıdır. Bazı kümeleme algoritmaları bazı verilerde daha iyi sonuçlar vermesine rağmen *K*–merkez kümeleme algoritması her çeşit veride kabul edilebilir sonuçlar verir. Algoritmanın en büyük dezavantajı yerel optimumlarda kalarak genel optimumlara ulaşamamasıdır.



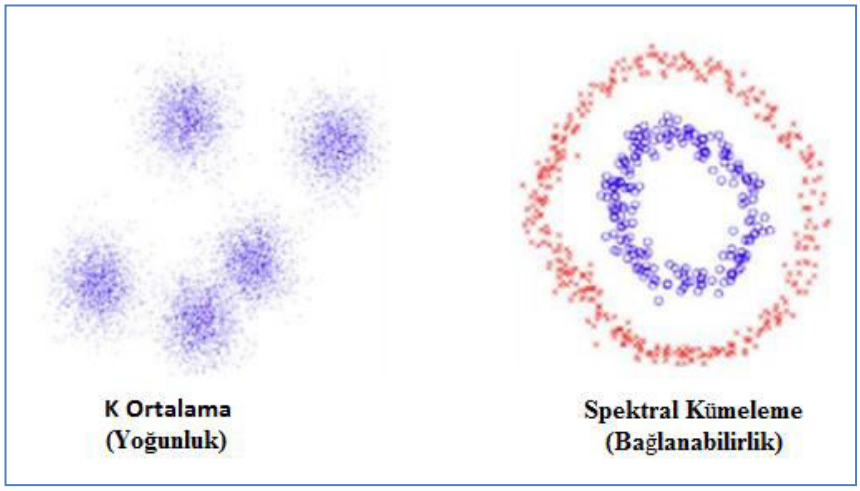
Şekil 2.16: *K*-merkez algoritması adımları.

K-merkez algoritması adından da anlaşılabileceği gibi giriş uzayını *K* adet merkezle ifade etmeye çalışan bir yöntemdir[8]. Merkezlere ilk değer ataması rastgele olarak yapıldıktan sonra merkez değerlerinin güncellenmesi için iki farklı yöntem kullanılır. Birinci yöntemde giriş kümesindeki her bir örneğin hangi merkeze yakın olduğu hesaplanır. Aynı merkeze yakın olan örneklerin ortalaması alınarak merkezin değeri güncellenmiş olur. Durma koşulu sağlanana kadar bu işlem tekrar edilir. İkinci yöntemde giriş kümesinden bir örnek seçilir ve bu örneğin merkezlere olan uzaklığına bakılır. Örneğin en yakın olduğu merkez bulunarak bu merkezin değeri güncellenir. Her

bir örnek için bu işlem tekrarlanır. Merkez değeri güncellenirken merkezle örnek arasındaki mesafe değeri her adımda azalan bir öğrenme katsayısıyla çarpılarak kullanılır. Bu sayede ilk adımlarda merkezlerin yer değiştirmesi büyük miktarlarda olurken zamanla yer değiştirme azalır. Bu çalışmada birinci yaklaşım tercih edilmiştir.

2.4.2. Spektral Öbekleme

Son yıllarda, spektral öbekleme popüler kümeleme algoritmalarından biri haline gelmiştir. Standart lineer cebir yazılım tarafından verimli bir şekilde çözünebilir hale gelmiş ve uygulaması basittir. Bu yöntem k-merkez algoritması gibi geleneksel kümeleme algoritmalarını geride bırakıyor. İlk bakışta spektral kümelenme biraz gizemli görünür ve ne için çalıştığı, ne yaptığı belirgin değildir.



Şekil 2.17: *K*-Merkez için yoğunluk (Compactness) önemli iken Spektral için bağlanabilirlik (Connectivity) önemlidir.

Spektral kümeleme sonuçları geleneksel yaklaşımları sık sık geride bırakır, spektral kümeleme gayet basit uygulanır ve standart doğrusal cebir yöntemleri ile verimli çözülebilir [2].

Spektral öbeklemede verilen N örnekten oluşan $X_1, \dots, X_N$ data kümesini kullanarak benzerlik matrisini $S \epsilon R^{NxN}$ Gaussian fonksiyonu kullanılarak oluşturur

$$S_{ij} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right). \tag{2.3}$$

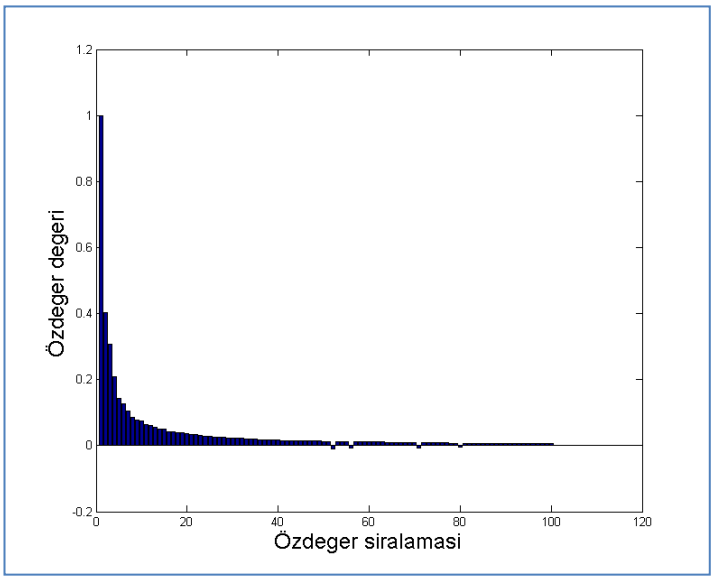
Denklem 1 ile her düğümün diğer düğümlere olan Öklid uzaklıklarına ters orantılı bir şekilde benzerlikler belirlenir. Spektral öbeklemenin birçok farklı türleri bulunmaktadır [2]. Bu çalışmada tercih edilen normalize edilmiş spektral öbekleme yönteminde [10, 11] çizge Laplacian matrisi,

$$L = I - D^{-1/2} S D^{-1/2}, \tag{2.4}$$

şeklinde elde edilir ve burada D köşegen matrisi şöyle hesaplanır:

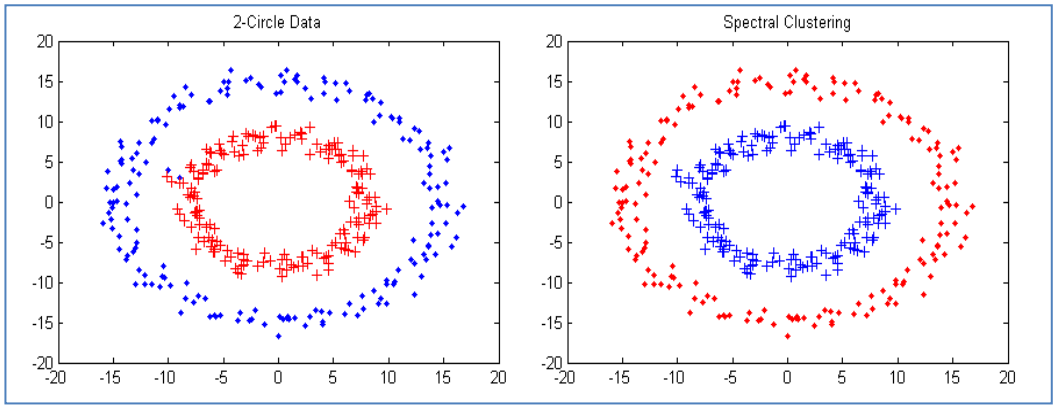
$$D_{ii} = \sum^{n} S_{ij}. \tag{2.5}$$

L matrisinden istenen boyut sayısınca *K* özvektör seçilerek *N*×*K* şeklinde yeni bir veri matrisi oluşturulur ve *K*-merkez öbekleme gibi bir algoritma ile bu yeni *K* boyutta öbeklemeye gidilir.



Şekil 2.18: Benzerlik matrisinin özdeğerlerin histogramı.

K sayısı olarak kaç boyut kullanılması gerektiğini Şekil 1'de gösterildiği gibi yüksek özdeğerler belirlenerek karar verilir. Ancak burada genelde kullanılabilecek sezgisel bir diğer yaklaşım da istenilen öbek sayısınca boyut seçmektir.



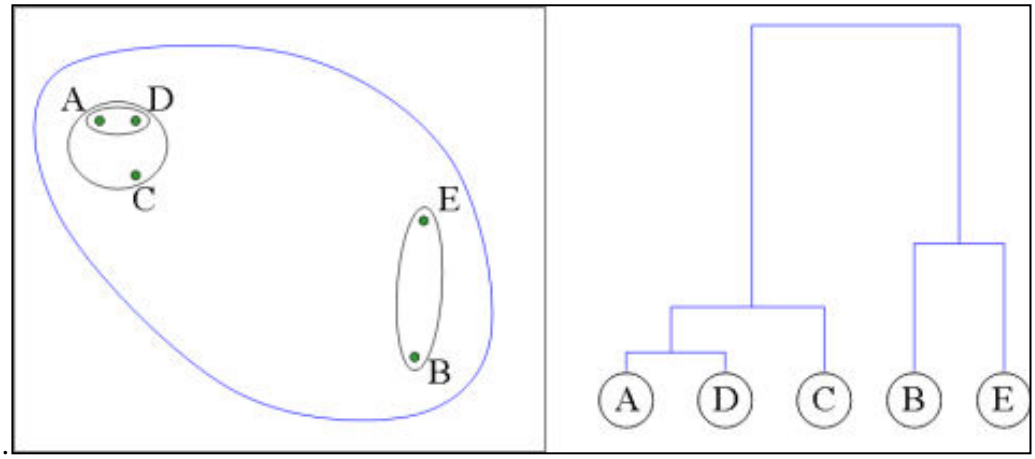
Şekil 2.19: 2-Circle veri kümesinde Spektral Öbekleme (SÖ) yönteminin öbeklemede başarılı olduğu görülmektedir.

2.4.3. Hiyerarşik Öbekleme Yöntemleri

Hiyerarşik kümeleme (HÖ) yöntemleri, veri setinin birimlerinin birbirlerine olan uzaklık değerlerini kullanarak, veri setindeki birimlerin hiyerarşik ayrıştırmasını yapar. Hiyerarşik ayrıştırma sırasında, dendogram olarak bilinen ağaç diyagramı kullanılır. Ağaç diyagramı, hiyerarşik kümeleme yöntemiyle elde edilen kümelerin görselleştirilmesini sağlar. Küme sayısına görsel olarak karar verilir. Gruplayıcı ve bölücü olmak üzere iki yöntem mevcuttur. Gruplayıcı hiyerarşik yöntemde her birim veya her gözlem başlangıçta bir küme olarak kabul edilir. Daha sonra en yakın iki küme (veya gözlem) yeni bir kümede toplanarak birleştirilir. Böylece her adımda küme sayısı bir azaltılır. Bölücü hiyerarşik yöntemde ise süreç gruplayıcı hiyerarşik yöntemin tam tersidir. Bu yöntemde tüm gözlemlerden oluşan büyük bir küme ile ise başlanır. Benzer olmayan gözlemler ayıklanarak daha küçük kümeler oluşturulur. Her gözlem tek basına küme oluşturana kadar isleme devam edilir [19].

Hiyerarşik kümeleme yöntemi dört adımdan oluşan bir algoritma ile ifade edilebilir:

1. n tane birey, n tane küme olmak üzere işleme başlanır.
2. En yakın iki küme (d_{ij} değeri en küçük olan alınır) birleştirilir.
3. Küme sayısı bir indirgenerek yinelenmiş uzaklıklar matrisi bulunur.
4. 2 ve 3 nolu adımlar n-1 kez tekrarlanır [20].



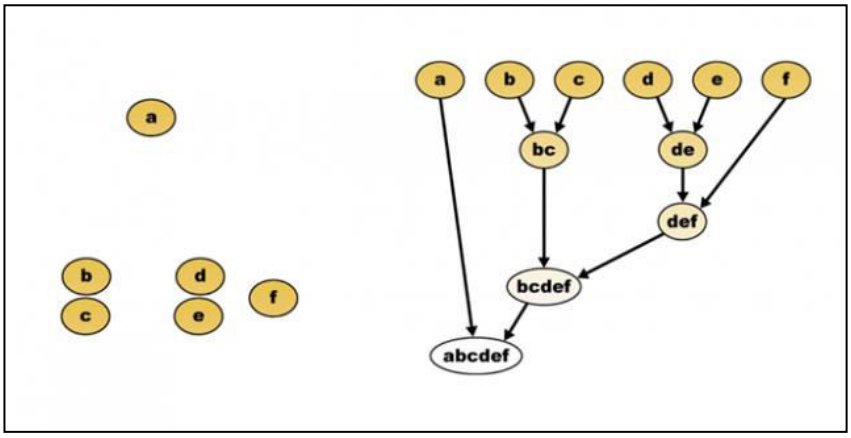
Şekil 2.20: Dendrogram örneği.

Hiyerarşik kümeleme yöntemlerinin çıktıları dendrogramlar ile sunulmaktadır. Hiyerarşik kümeleme yönteminin "dendrogram" ile gösterimi Şekil 2.3.3-1'de görülmektedir. Analizlerde birçok kümeleme yöntemi denemek sonuçları karşılaştırmak için fayda sağlayabilir. Verilerin özelliklerine bağlı olarak, bazı kümeleme yöntemi diğerlerine göre daha uygun kümeler oluşturabilir. En çok kullanılan 3 hiyerarşik kümeleme yöntemleri şunlardır;

1. Tek Bağlantı Kümeleme Yöntemi (Single Linkage)
2. Tam Bağlantı Kümeleme Yöntemi (Complete Linkage)
3. Ortalama Bağlantı Kümeleme Yöntemi (Average Linkage)

Dendogramın yapısının oluşturulmasına göre hiyerarşik kümeleme yöntemi ikiye ayrılır. Bu yöntemler, birleştirici hiyerarşik kümeleme yöntemleri ile bölücü (divisive) kümeleme yöntemleridir.

Hiyerarşik yapı oluşturulurken dendogramın kökünden birimlere doğru iniliyor ise bu yönteme bölücü hiyerarşik kümeleme yöntemi adı verilir. Bu durumun tersinde ise, yani kümeleme işlemi yapılırken her bir birim ayrı bir küme olarak düşünülüp ana küme elde ediliyorsa bu yönteme birleştirici hiyerarşik kümeleme yöntemi adı verilir.



Şekil 2.21: Birleştirici hiyerarşik kümeleme.

Hiyerarşik kümeleme yöntemleri iteratif yöntemlerdir. Bu işlemlerin en büyük olumsuzluğu, bir adım gerçekleştirildikten sonra bir daha tekrar aynı adıma geri dönülememesidir. Bu yüzden yanlış kararları doğrulamaya izin vermemektedir.

Birleştirici hiyerarşik yöntemlerin algoritması genel olarak şu iteratif süreçlerinden geçer

- Veri setindeki her bir birim ayrı bir küme olarak kabul edilir.
- Her bir birimin diğerlerine olan uzaklıklarından ya da benzerliklerinden oluşan (birim sayısı N olmak üzere) $N \times N$ kare matrisi oluşturulur. $D=\{d_{ik}\}$
- Oluşturulan $N \times N$ kare matrisi en yakın küme çiftleri araştırılır
- En çok benzer olan kümeler birleştirilir. $N \times N$ kare matrisi birleştirilen kümelere göre yeniden düzenlenir. Daha açık olarak söylemek gerekirse, birleştirilen kümelere ait satırlar ve sütunlar silinir ve yeni oluşturulan küme veya kümeler ve diğer kümelerin oluşturdukları $(N-1) \times (N-1)$ kare matris oluşturulur.
- 3. ve 4. adımlar $N-1$ kez tekrar edilir ve dendogramın kök kısmı oluşunca kümeleme işlemi sona erer.

2.4.3.1. Tek Bağlantı Öbekleme Yöntemi (Single Linkage)

Single linkage yöntemi için girdi, birim çiftleri arasındaki uzaklık ya da benzerliklerin oluşturduğu $N \times N$ kare matristir. Kümeler, her biri ayrı bir küme olarak kabul edilen

birimlerin birleştirilmesiyle oluşturulur. En yakın komşular, yani minimum uzaklığa ya da maksimum benzerliğe sahip olan küme çiftleri bir araya getirilir.

En yalın hiyerarşik kümeleme yöntemidir. Aynı zamanda en yakın komşuluk yöntemi olarak ta bilinir. Bu yöntemde uzaklıklar matrisinden faydalanılarak, birbirine en yakın birim veya kümeler birleştirilir, birleştirmelere bütün birimler herhangi bir kümeye atanıncaya kadar devam edilir. Birleştirme yapılırken kümelerin eleman sayısının birden fazla olması koşulu yoktur. Bir birim yalnız başına bir küme oluşturabilir. İki küme arasındaki uzaklık, bir kümedeki bir gözlem ve diğer kümedeki bir gözlem arasındaki minimum mesafedir.

2.4.3.2. Tam Bağlantı Öbekleme Yöntemi (Complete Linkage)

Complete linkage single linkage yöntemiyle benzer biçimdedir. Single linkage yönteminde başlangıçtaki $N \times N$ kare matrisi oluşturulduktan sonra birleştirilen iki kümenin diğer kümelere hesaplanmasında minimum uzaklıklar dikkate alınırken, complete linkage yönteminde maksimum uzaklıklar dikkate alınır.

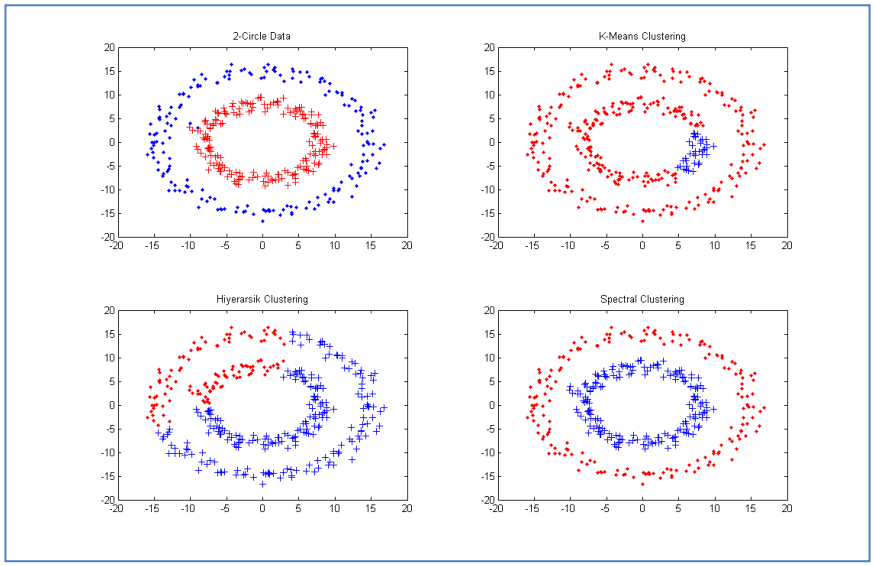
2.4.3.3. Ortalama Bağlantı Öbekleme Yöntemi (Average Linkage)

Uzaklıklardan ya da benzerliklerden oluşan $N \times N$ kare matriste minimum uzaklıkta olan kümelerin birleştirilmesiyle oluşturulan yeni kümenin diğer birimlere olan uzaklıkları yeni oluşturulan kümenin diğer kümelerin birbirlerine olan uzaklıklarının ortalaması hesaplanılarak bulunur. Elde edilen yeni matriste ise, birbirine en az uzak olan kümeler birleştirilir.

Yeni oluşturulan kümenin diğer kümelere olan uzaklığı ise, U V ve W kümeler olmak üzere aşağıdaki formül ile hesaplanır:

$$d_{(uv)w} = \sum_i \sum_k \frac{d_{ik}}{N_{UV} N_W} \quad (2.6)$$

Bu yöntemde işleme tek bağlantı ve tam bağlantı yöntemlerinde olduğu gibi başlanır. Ancak kümeleme kriteri olarak bir küme içindeki birim ile diğer küme içindeki birimler arasındaki ortalama uzaklıklar kullanılır. Ortalama bağlantı tekniğinde kümeler küçük varyanslar ile birbirlerine bağlıdır. Bu teknik tek bağlantı ve tam bağlantı teknikleri arasında sonuçlar vermesi nedeniyle bir alternatif yöntem olarak önerilmektedir [9].

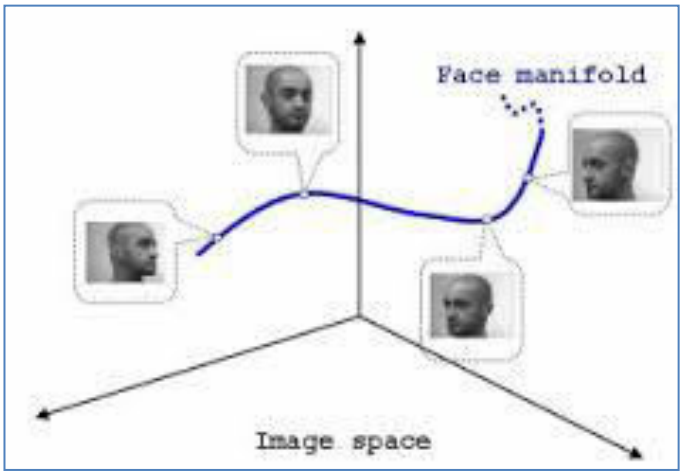


Şekil 2.22: 2-Circle veri kümesine K-merkez , Hiyerarşik Öbekleme, Spektral Öbekleme sonuçları , farklı renkler farklı öbekleri göstermektedir.

Şekil 2.22'de gösterilen 2-Circle veri kümesinde üstün küre yapısına uymayan, yol-tabanlı türdeki öbeklerin spektral öbekleme yöntemi ile kolaylıkla tespit edildiği gösterilmiştir.

2.4.4. Yol Tabanlı(En Küçük Maksimum Atlamalı) Spektral Öbekleme

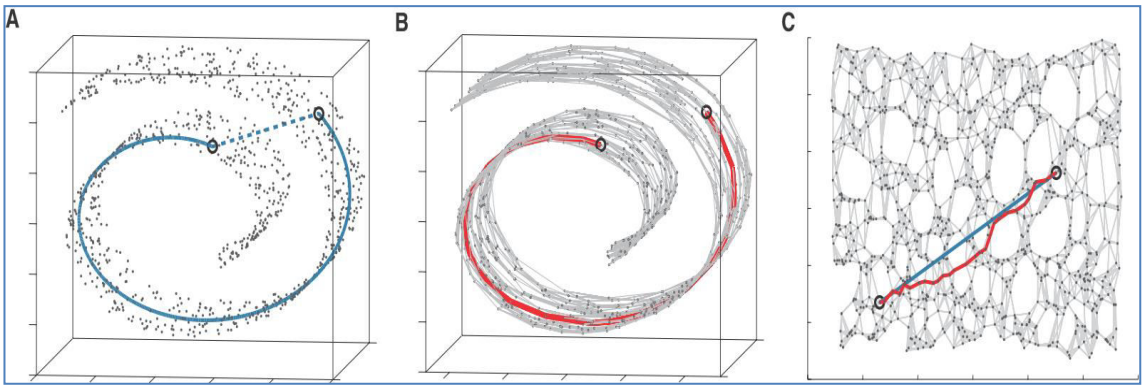
Birçok uygulamada ve problemlerde Öklid uzaklığı benzerlik ölçüsü olarak kullanılır. Ama bazı problemlerde bu varsayım geçerli olmayabilir. Yüz tanıma problemlerini ele alalım:



Şekil 2.23: Resim uzayındaki oluşan yüz gezinmesi.

Yüzlerin iki boyutta, örneğin 100×100 imgeler olduğunu varsayarsak her yüz 10.000 boyutlu uzayda bir nokta olacaktır. Simdi bir kişinin başını sağdan sola yavaşça çevirdiğini düşünelim; bu sırada kaydedilen imgeler bu 10.000 boyutlu uzayda bir gezinge (Manifold) izleyecektir. Ama bu gezinge doğrusal değildir. İnsanlar yüzlerini çevirirlerken yüz resimlerinin 10.000 boyuttaki uzayda oluşturduğu gezingeler bir alt uzay tanımlar ve istediğimiz bu alt uzayı modellemektedir[22].

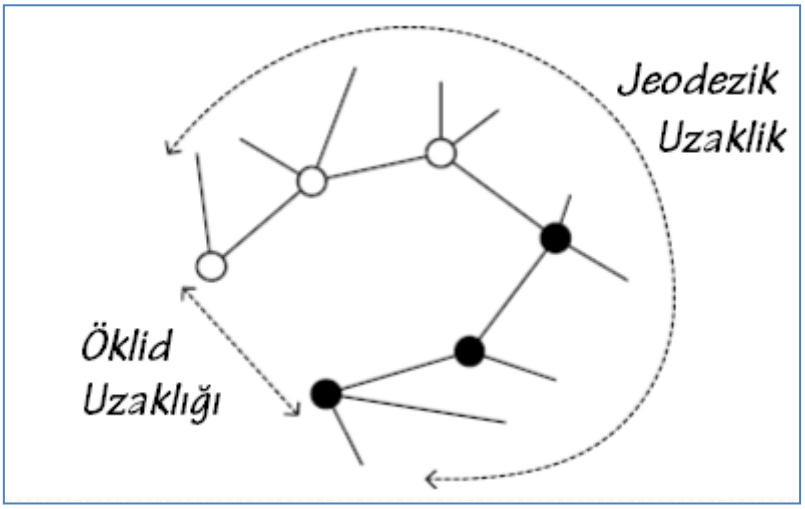
İki yüz resmi arasındaki benzerlik imge noktalarındaki değerlerinin farklarının toplamı olarak yazılamaz ve bu yüzden Öklid uzaklığı iyi bir benzerlik ölçüsü değildir. Farklı iki kişinin ayni açıdan çekilmiş imgeleri arasındaki Öklid uzaklığı, ayni kişinin farklı iki açıdan çekilmiş imgeleri arasındaki Öklid uzaklığından daha küçük olabilir. İstenen bu değildir. Önemli olması gereken jeodezik uzaklık denilen gezingenin (Manifold) tanımladığı alt uzay boyunca olan uzaklıktır [21]. Es ölçümsel öznitelik esleme (ISOMAP) bu uzaklığı kestirip çok boyutlu ölçekleme (MDS) ile boyut azaltır.



Şekil 2.24: Spiral veri kümesi için (c) Öklid uzaklıkları ve jeodezik uzaklık bulunması.

Verideki nokta çiftleri arasındaki jeodezik uzaklıklar kullanılır. Girdi uzayında yakin olan komsu noktalar arasında Öklid uzaklığı kullanılabilir; bakış arasındaki çok küçük değişiklikler için alt uzayın yerel olarak doğrusal olduğu varsayılabilir.

Uzak noktalar için yerölçümsel uzaklık ise alt uzayda üzerinden geçilen noktalar arasındaki yerel uzaklıkların toplamı olarak kestirilebilir. Bunu yaparken tanımladığımız çizgenin düğümleri *N* veri noktasına, ayrıtlar da komşuluklara karşılık gelir[22].



Şekil 2.25: Jeodezik uzaklık ile Öklid uzaklık farkı.

İki nokta arasındaki jeodezik uzaklık, karşılık gelen düğümlerin arasında çizge içindeki en kısa yol uzunluğu olarak hesaplanır. Yakin olmayan iki nokta için çizge üstünde ara düğümler üzerinden sıçrayarak gidileceğinden uzaklık, alt uzay üzerindeki yerel Öklid uzaklıklarının toplamı olarak (Şekil 2.25) kestirilen uzaklık olacaktır.

Yol-tabanlı Spektral Öbekleme (YSÖ) yönteminde, öncelikle veri kümesindeki düğümler arasındaki Öklid mesafeleri kullanılarak bir çizge oluşturulur. Bu çizgedeki iki düğüm arasındaki benzerliğe temel olarak iki düğüm arasındaki tüm yollar arasından yol içerisindeki maksimum atlamanın en küçük olduğu yol seçilir ve bu seçilen yol üzerindeki maksimum atlama uzaklık olarak kabul edilir. Sonra, bu jeodezik uzaklıklar yine Gaussian benzerlik fonksiyonu kullanılarak, normal spektral öbeklemedeki gibi affinity matrisi oluşturulur [30].

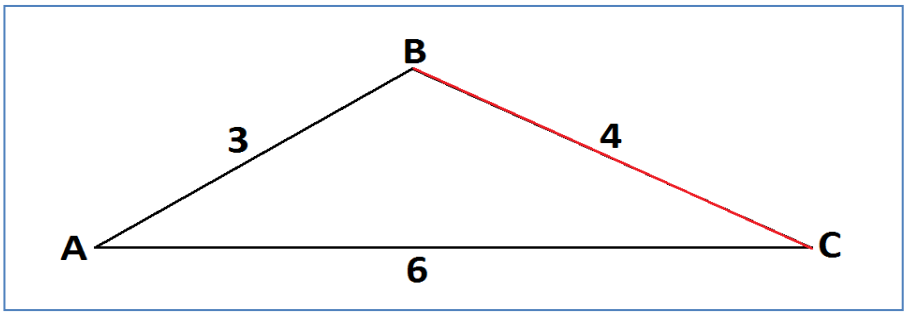
Bu hesaplamayı tüm düğüm ikilileri için dinamik programlama yaklaşımı kullanan Floyd-Warshall [12] algoritması $O(N^3)$ karmaşıklıkla yapabilmektedir. Algoritmanın her adımında matris üzerinde çeşitli işlemler yapılarak en kısa yol bulunmaya çalışılır. Algoritma matris üzerinde çalışan ve düğümler arası mesafeleri her adımda güncelleyen bir algoritma olduğu için iteratif (iterative, döngü ile) yazılabilir. Aşağıda müspette kodlar (pseude codes) verilmiştir.

```
int yol[][];
procedure FloydWarshall ()
for i = 1 to N
   for j = 1 to N
      if(i'den j'ye bir yol varsa)
         yol[0][i][j] = i ile j arasındaki mesafe
      else
         yol[0][i][j] = sonsuz
for k = 1 to N
   for i = 1 to N
      for j = 1 to N
         yol[k][i][j] = min(yol[k-1][i][j], yol[k-1][i][k]
+ yol[k-1][k][j])
```

Sonucun içinde bulunacağı ve anlık olarak bir düğümden diğerine ne kadar maliyet ile bulunduğunu tutan matris ilk olarak komşuluk listesini tutar ve doğrudan gidilemeyen düğümlere olan mesafe sonsuz olarak tutulur. Algoritmada görüldüğü üzere *n* adet düğümlü bir çizge için iki boyutlu bir dizi oluşturulur. Bu dizinin içerisindeki değerlere ilk olarak komşuluk listesindeki değerler atanır ve doğrudan ulaşılamayan düğümler için sonsuz değeri doldurulur. Ardından Matrisi dolaşan bir döngü ile (*i ve j*) yollar güncellenir. Bu matrisin taranması işlemi düğüm sayısı *(n)* kadar tekrar eder (yukarıdaki *k* döngü değişkeni tekrara yaramaktadır). Bu tekrar aslında üzerinden atlanma ihtimali olan düğümü belirtmektedir. Yani örneğin *k = 3* için 3. düğüm marifetiyle ulaşılan düğümler belirlenir.

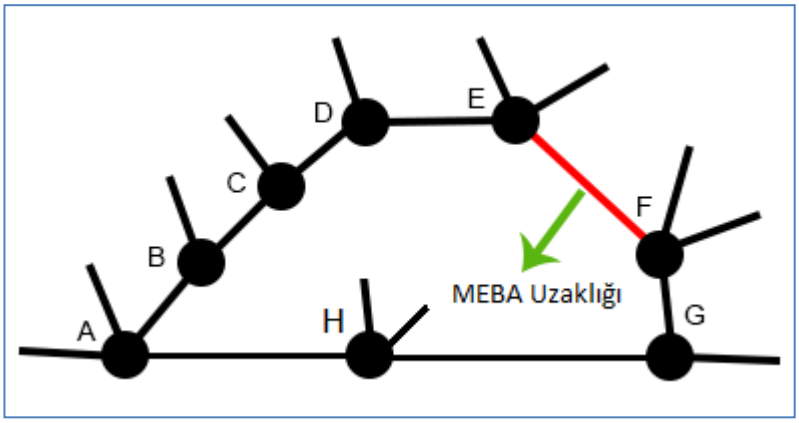
```
for k ← 1 to N
  for i ← 1 to N
     for j ← 1 to N
       if dist[i][j] > MAX(dist[i][k] , dist[k][j])
           S[i][j] ← MAX(dist[i][k] , dist[k][j])
       end if
     end
  end
end
```

Algoritma 1: Floyd-Warshall algoritmasında ufak bir değişiklikle i-j düğümleri arasındaki tüm yollar içerisinden en küçük maksimum atlamalı yol kolaylıkla tespit edilebilir.



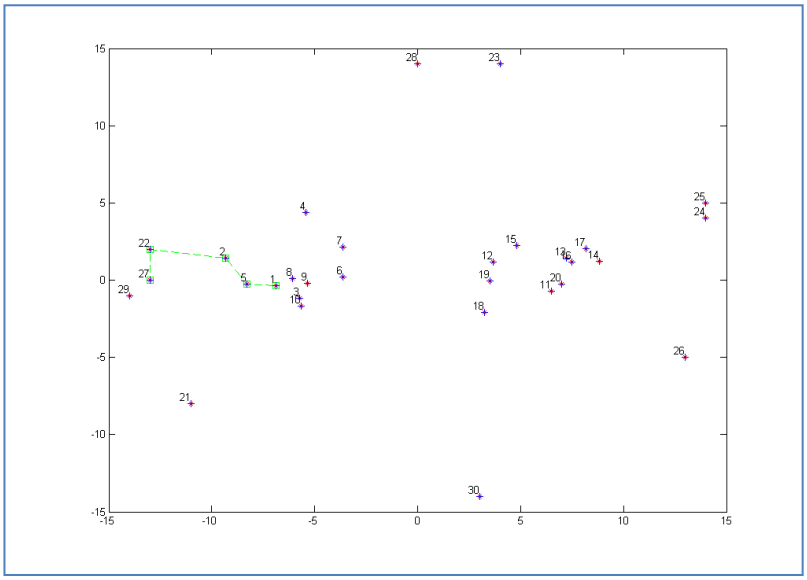
Şekil 2.26: A'dan C'ye B üzerinden gitmek için öyle bir yol seçilmelidir ki o yol üzerindeki maksimum atlama minimum olsun. Bu da yolu B üzerinden tercih ettirir ve A-C uzaklığını normal spektral öbeklemedeki gibi 6 değil de 4 olarak belirlemiş olur.

Verilen pseudo kod parçasında görüldüğü gibi tüm düğümler dinamik programlama tekniği ile atlama taşı olarak kullanılarak düğümler arası oluşacak en zayıf benzerlik oranının olabildiğince güçlü olması sağlanmaktadır [13].



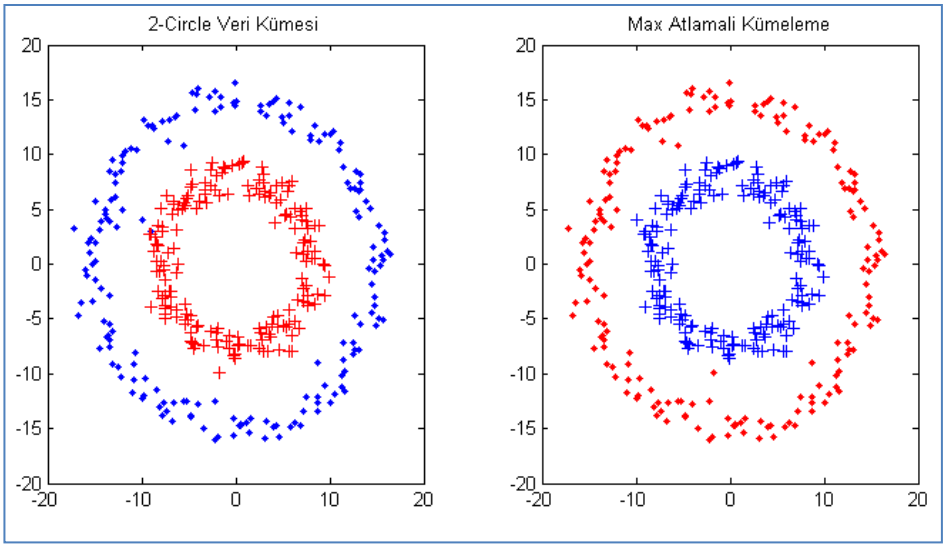
Şekil 2.27: A-G arası Minimum En Büyük Atlama Uzaklığı.

İki düğüm arasındaki tüm yollar arasından yol içerisindeki maksimum atlamanın en küçük olduğu yol seçilir ve bu seçilen yol üzerindeki maksimum atlama uzaklık olarak kabul edilir. Bu algoritmaya göre A'dan G'ye gidilen yoldaki tüm düğümler arası en kısa yol olan H dugumu uzerinden gitmek yerine minumum en büyük atlama (MEBA) olan E-F uzaklığı kullanılır [31].

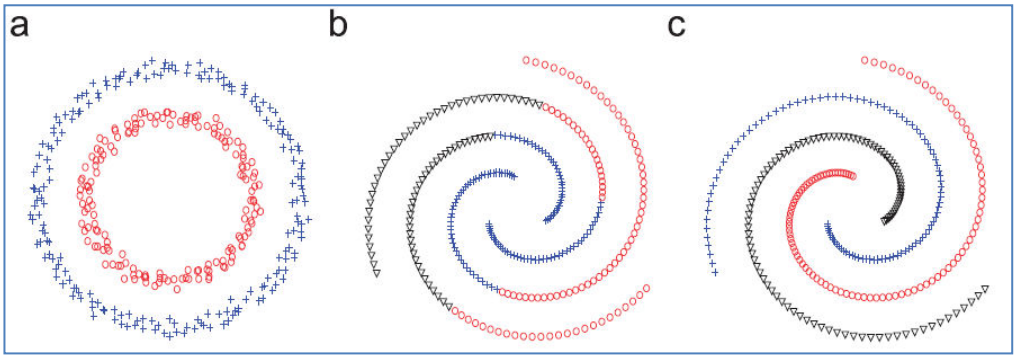


Şekil 2.28: Maksimum atlama kullanılarak 1. Düğümden 27. düğüme giden atlamalı yollar gösterilmektedir.

Böylece uzaklık matrisinde Öklid uzaklıkları yerine (Şekil 2.26) verinin uzaydaki dağılımını da kullanan bir uzaklık matrisi çıkartılmaktadır. Şekil 2.28'de verilen 2-Circle(Çember) veri kümesinde YSÖ da aynı derecede başarılı sonuçlar vermektedir. Ancak, bazı problemlerde YSÖ, bazı problemlerde ise SÖ birbirlerine üstünlük gösterebilmektedir. Bu sebeple bu çalışmada bu iki yöntemin birleştirilerek daha gürbüz çalışan bir hibrit öbekleme algoritması önerilmiştir.



Şekil 2.29: 2- Circle veri kümesinde En Küçük Maksimum Atlamalı (YSÖ) yönteminin öbeklemede başarılı olduğu görülmektedir.



Şekil 2.30: (a) 2-Circle (Çember) veri kümesi için spektral öbekleme (b) 3-Spiral veri kümesi için spektral öbekleme (c) 3-Spiral veri kümesi için yol tabanlı öbekleme.

Şekil 2.30'da gösterilen 3-Spiral veri kümesi için yol tabanlı öbekleme yöntemi spektral öbeklemeden daha iyi çaliştığı görülmektedir.

2.5. BAŞARI ÖLÇÜTLERİ

2.5.1. Karşılıklı Bilgi (Mutual Information)

Karşılıklı bilgi (MI) olasılık ve bilgi teorisinde karşılıklı bağımlığı gösteren ölçüdür. Korelâsyon katsayısı gibi reel değerli rassal değişkenler ile sınırlı değildir. MI daha genel ve ortak dağıtım *P(X, Y)* faktörlü marjinal dağılım *P (X) P (Y)* ürünlere ne kadar benzediğini belirler. Karşılıklı bilgi ölçümü en yaygın kullanılan yöntemdir[23]. Logaritma iki tabanındayken en çok kullanılan ölçü birimi bit olmaktadır. Matematiksel olarak, karşılıklı bilgi iki soyut rassal değişken olan *X* ve *Y* için şu şekilde ifade edilir:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y)\, log\left(\frac{p(x,y)}{p_1(x)p_2(y)}\right) \tag{2.7}$$

Burada $p(x,y)$ olarak ifade edilen, *X* ve *Y*'nin bağıl olasılık dağılım fonksiyonu, $p_1(x)$ve $p_2(y)$ marjinal olasılık fonksiyonlarıdır.

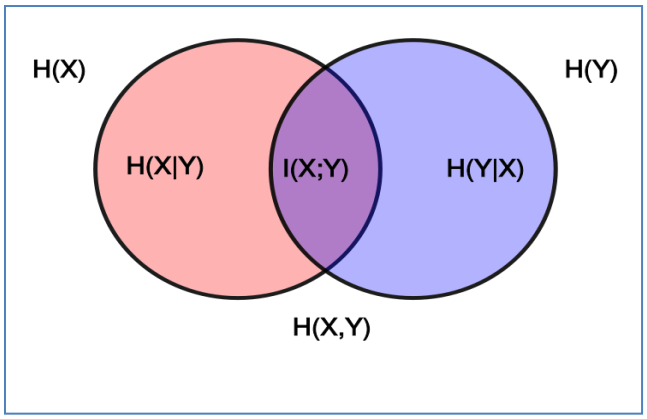
Sürekli durumda ise, çift integral ile yer değiştirilir.

$$I(X;Y) = \int_Y \int_X p(x,y) log\left(\frac{p(x,y)}{p_1(x)p_2(y)}\right)\, dx\, dy \tag{2.8}$$

Bu durumda ise, $p(x,y)$ bağıl yoğunluk fonksiyonuna dönüşmüştür. *X* ve *Y* için ise, $p_1(x)$ve $p_2(y)$ marjinal olasılık yoğunluk fonksiyonuna dönüşmüştür[23].

Bu tanımlar çelişki içermektedir çünkü temel alınan log fonksiyonu belirtilmemiştir. Bunu düzeltmek için, *I* fonksiyonu parametreleştirilirse $I(X,Y,b)$, *b* burada temel alınan fonksiyondur. Başka bir alternatif ise, en çok kullanılan birim olan "bit" temel almak olabilir, bu durumda 2 tabanında işlem yapılmış olacaktır.

Karşılıklı bilginin *X* ve *Y* değişkenlerinin birbiri hakkında ne kadar bilgi sahibi anlamına geldiğinin belirtmiştik. Eğer iki değişken birbirleri hakkında çok az bilgi içeriyorsa, karşılıklı bilgileri 0'a yakındır. Diğer durumda ise, eğer *X* ve *Y* aynı bilgileri taşıyorlarsa, yani *X* değerlerini belirlerken *Y*'den bilgi içeriyorsa bu durumda karşılıklı bilgi 0'dan farklıdır.



Şekil 2.31: Karşılıklı bilgi (MI) ve entropi Venn şeması.

Karşılıklı bilgi, *X* ve *Y*'nin bağıl dağılımları ölçer ve bu dağılımın sonuçlarını inceler. Sadece ve sadece iki değişken birbirinden bağımsız iseler, $I(X,Y) = 0$ olacaktır. Aslında bunu anlamak hiç de zor değildir. Eğer *X* ve *Y* birbirinden bağımsız iseler, $p(x,y)$ değeri gittikçe $p_1(x)p_2(y)$ 'e yaklaşacaktır. Bu durumda

$$log\left(\frac{p(x,y)}{p(x)p(y)}\right) = log\,1 = 0 \tag{2.9}$$

Daha da fazlası, karşılıklı bilgi 0'dan büyüktür *(I(X;Y) ≥ 0)* ve simetriktir *(I(X; Y) = I(Y; X)).*

Entropi, bir sistemin veya durumun rastgeleliği, belirsizliği ve beklenmeyen durumun ortaya çıkma olasılığının, sayısını veya değerini gösterir. H ile gösterilir.

Karşılıklı bilgi (MI), şu şekilde de ifade edilebilir:

$$I(X;Y) = H(X) - H(X|Y) \tag{2.10}$$

$$= H(Y) - H(Y|X)$$

$$= H(X) + H(Y) - H(X,Y)$$

$$= H(X,Y) - H(X|Y) - H(Y,X)$$

Burada *H(X)* ve *H(Y)* marjinal entropi değerleridir. *H(X|Y)* ve *H(Y|X)* ise şartlı entropilerdir. *H(X,Y)* ise bağıl entropi değeridir. $H(X) \geq H(X|Y)$ olacağı için, bu az önce belirttiğimiz pozitifliği destekler niteliktedir.

Eğer *H(X)* bir değişkenin kararsızlığı hakkında bilgi veriyorsa, *H(X|Y) Y* değişkeninin *X* hakkında bilmediklerini söylemektedir.

2.5.2. Düzeltilmiş Rand İndeksi (Adjusted Rand Index ARI)

Dış kriterlere göre kümelenme sonuçlarını karşılaştırmak amacıyla bir ölçü gereklidir. Rand indeksi [26] veri kümelemede özellikle de iki veri kümelemelerinin arasındaki benzerliğin bir ölçüsüdür.

Rand indeksi form elemanlarını gruplama için ayarlanmış olacak şekilde tanımlanabilir, bu düzeltilmiş Rand indeksidir. Matematiksel açıdan, Rand indeksi doğruluk ile ilgilidir, ancak sınıf etiketleri kullanılmaz.

Düzeltilmiş rand indeks rand endeksinin başka bir versiyonudur. Rand indeksi sadece 0 ile +1 arasında bir değer verebilir olmasına rağmen endeks beklenen indeksi az ise, düzeltilmiş rand indeksi negatif değerler elde edebilirsiniz.

X\Y	Y_1	Y_2	$\cdots$	Y_s	Toplam
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
Toplam	b_1	b_2	$\cdots$	b_s	

Şekil 2.32: ARI İhtimal Tablosu.

N elemanlı bir dizi S göz önüne alındığında, iki gruplaşma (kümelenmemeler) bu noktalarda:

$$X = \{X_1, X_2, \ldots, X_r\} \tag{2.11}$$

$$Y = \{Y_1, Y_2, \ldots, Y_s\},$$

Her giriş n_{ij}, X_i ve Y_j arasındaki örtüşen (2.12) ortak nesne sayısını ifade etmektedir.

$$n_{ij} = |X_i \cap Y_j|. \tag{2.12}$$

X ve Y arasında örtüşme sayıları ihtimal tablosuna girilerek aşağıdaki (2.13) hesaplama formülü ile Düzeltilmiş rand indeks hesaplanır.

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{2.13}$$

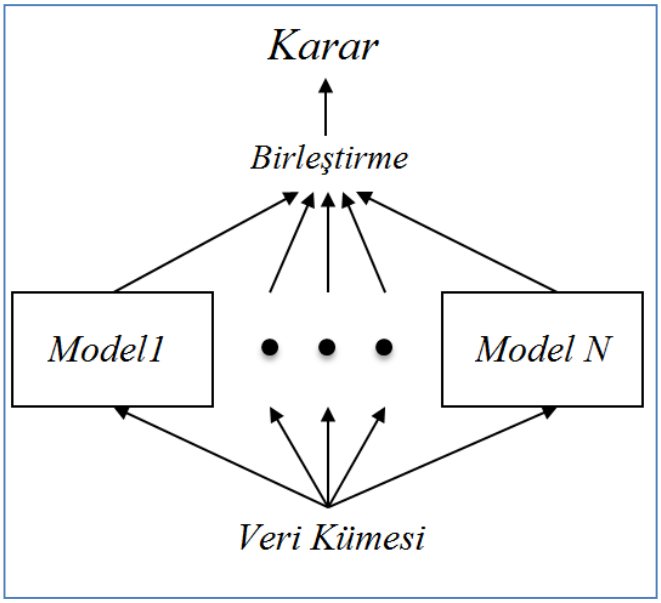
2.6. MODELLERİN BİRLEŞTİRİLMESİ

Bir uygulamada farklı öğrenme algoritmalarından herhangi birini ve her algoritmada model üst parametre değerlerinden herhangi birini kullanabiliriz. Örneğin bir sınıflandırma uygulamasında olasılık tabanlı bir sınıflandırıcı ya da çok katmalı bir algılayıcı kullanabiliriz. Eğer çok katmanlı algılayıcı kullanırsak saklı birim sayısı su ya da bu olabilir. Olası tüm uygulamalar üzerinde en başarılı olacağına emin olduğumuz

bir algoritma yoktur ve bu yüzden her uygulamada olası algoritmaların hepsini geçerleme kümesi üzerinde deneyip en başarılı olanı seçmek gerekir [22].

Makine öğrenmesinde aktif alanlardan birisi de model birleştirmektir. Temel fikir tek bir öğrenici eğitip kullanmak yerine çok sayıda öğreniciyi eğitip sonuçlarını birleştirmektir [27,28]**.** Öğrenici topluluklarının işe yaraması (tek bir öğrenici ile elde edilen sonuçlardan daha yüksek başarıya sahip olmaları) için temel öğrenicilerin birbirlerinden farklı sonuçlar üretmeleri gereklidir [28].

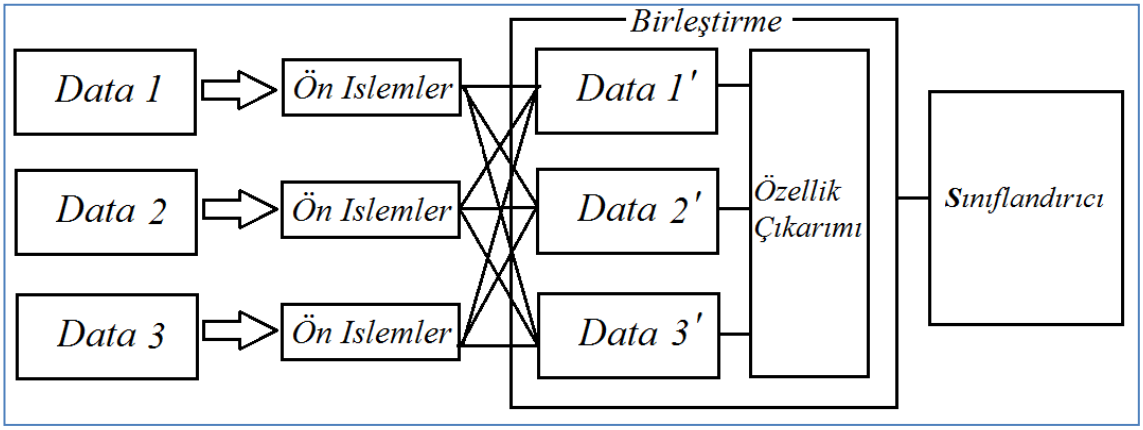
Tüm temel öğrenicilerin aynı sonucu üretmeleri durumunda bu aynı kararları birleştirmenin işe yaramayacağı açıktır. Bununla birlikte temel öğrenicilerin tekil performanslarının yüksek olması öğrenici toplulukların başarısında önemli bir etkendir. Dolayısıyla temel öğrenicilerin birbirlerinden farklı sonuçlar üretmelerini ve temel öğrenicilerin tekil performanslarının yüksek olmasını istemek aslında birbirine karşıttır. Temel öğrenicilerin performansı arttıkça sonuçları arasındaki farklılık azalacaktır.



Şekil 2.33: Model birleştirme akışı.

Bir modelin geçerleme hatasını en aza indirmek için ayarlamalar yapılabilir. Ama hem bu ayarlamalar maliyetler maliyetlidir hem de ne kadar ayarlasak da en basarîli algoritmalar bile bazı örneklerde hata yapacaktır. Ama baksa model bu örnekler de doğru sonuç verebilir. Öyleyse birden çok temel modeli birleştirerek basarîyi

artırabiliriz. Son yıllarda bellek hesaplama maliyetlerinin azalmasıyla bu biçimde birden çok modeli birleştiren yaklaşımlar daha çok kullanılır olmuştur[28].

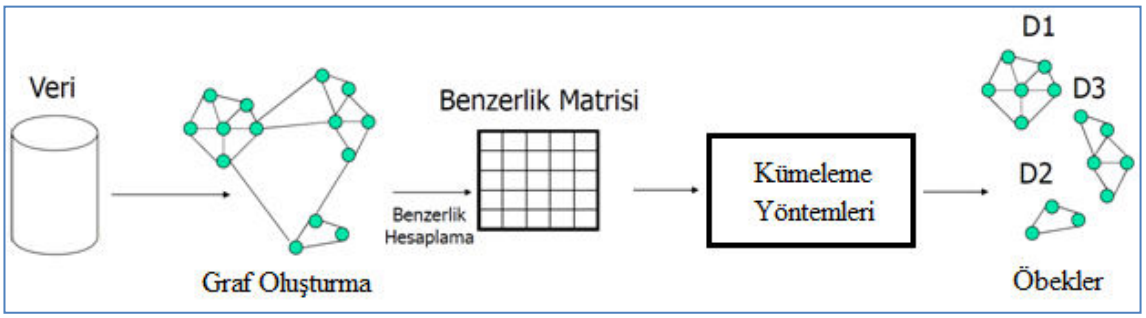


Şekil 2.34: Veri kümesi (View) birleştirme akışı.

Model birleştirmenin yanında veri kümesi(View) on islemeler ile beraber birleştirerek daha basarîli sınıflandırma sonuçları elde edilebilir. Fakat model veya veri kümelerini (View) birleştirmenin başarıyı her zaman artırmaz. Bir yerine birden çok modeli birleştirmek bellek ve hesap gereksinimini her zaman artırır ama temel modeller dikkatli eğitilip kararları zekice birleştirilmez ise başarıda bir artma görmeden yalnızca karmaşıklığı ve maliyeti artırmış oluruz.

3. MALZEME VE YÖNTEM

Bu bölümde literatürde kullanılan popüler öbekleme yöntemleri ve tez kapsamında önerilen öbekleme yöntemlerinden bahsedilecektir. İkinci kısımda bahsedilen *K*-Merkez, Spektral Öbekleme (SÖ) ve kod uygulaması bizim tarafımızdan geliştirilen Yol Tabanlı Spektral Öbekleme(YSÖ) ile yeni önerilen Hibrit Öbekleme (HÖ) ve Laplacian Tanımlı Öbekleme(LTÖ) yöntemlerin gerçek UCI veri kümeleri üzerinde uygulanma adımları ve deney sonuçlarından bahsedilecektir.

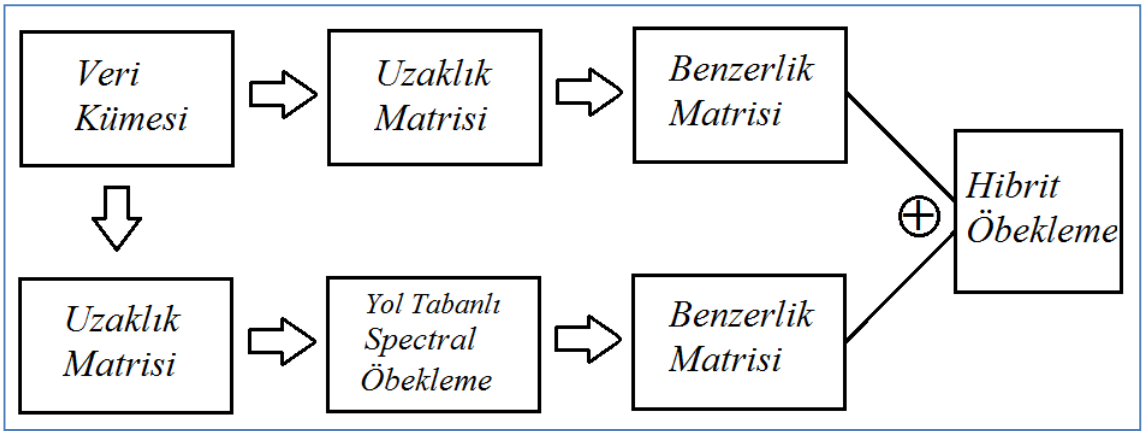


Şekil 3.1: Öbekleme akışı.

Şekil 3.1'de gösterilen kümeleme problemi akışında benzerlik matrisinden öbekleri oluşturacak kümeleme yöntemleri için önerilen algoritmaların detayları ve sonuçları verilecektir.

3.1. HİBRİT ÖBEKLEME

Hibrit öbekleme yöntemimizde, Spektral Öbekleme (SÖ) ve Yol-tabanlı Spektral Öbekleme (YSÖ) yöntemlerini bir arada kullanmak için bu iki yöntemden oluşan boyutlar yan yana konarak (K boyut birinden, K boyut diğerinden toplam 2K boyut olarak) kullanılabilirdi ancak bu yaklaşım hibrit yöntemin daha çok boyut kullanması sebebiyle SÖ ve YSÖ ile karşılaştırmalarda eşitsizlik çıkartacağından tercih edilmemiştir.

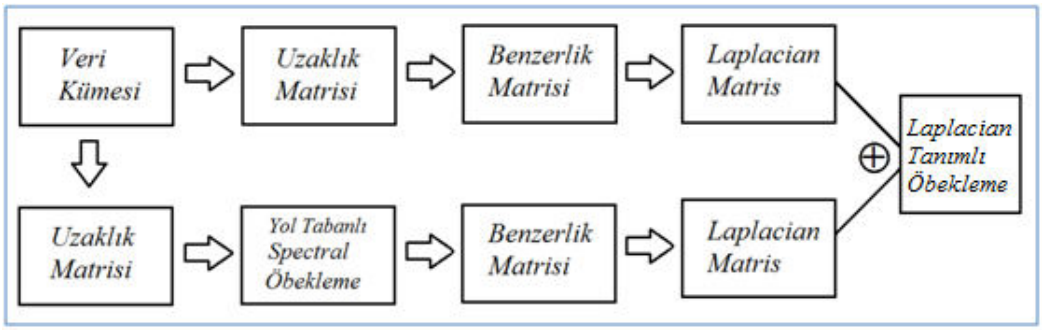


Şekil 3.2: Hibrit Öbekleme akışı.

Bu çalışmada önerilen yaklaşım, spektral öbeklemeden Öklid uzaklığı temel alınarak oluşturulmuş olan benzerlik matrisi ve en küçük maksimum atlamalı yol yöntemi ile elde edilen benzerlik matrislerinin toplanarak hibrit bir benzerlik matrisi olarak kullanılmasıdır. Normal spektral öbekleme yöntemindeki gibi bu benzerlik matrisi üzerinden Laplacian matrisi çıkartılıp özvektörleri alınmış ve *K*-merkez ile bu özvektörler üzerinden çıktı öbekler bulunmuştur.

3.2. LAPLACİAN TANIMLI ÖBEKLEME

Her bir düğümünden diğer düğümlere olan ayrıtların ağırlıklarının benzeştirilmesi, köşegen matrisi olan (*D*) derece matrisi ile hesaplanır ve bu derece matrisinden Öklid uzaklıklarına ters orantılı bir şekilde hesaplanan simetrik benzerlikler matrisi (*W*) çıkartılarak Laplacian matris hesaplanır.



Şekil 3.3: Laplacian Tanımlı Öbekleme akışı.

Önerilen yeni yöntemde Spektral Öbekleme (SÖ) ve Yol-tabanlı Spektral Öbekleme (YSÖ) yöntemlerinden oluşan Laplacian matrisleri benzeştirerek hibrit bir Laplacian

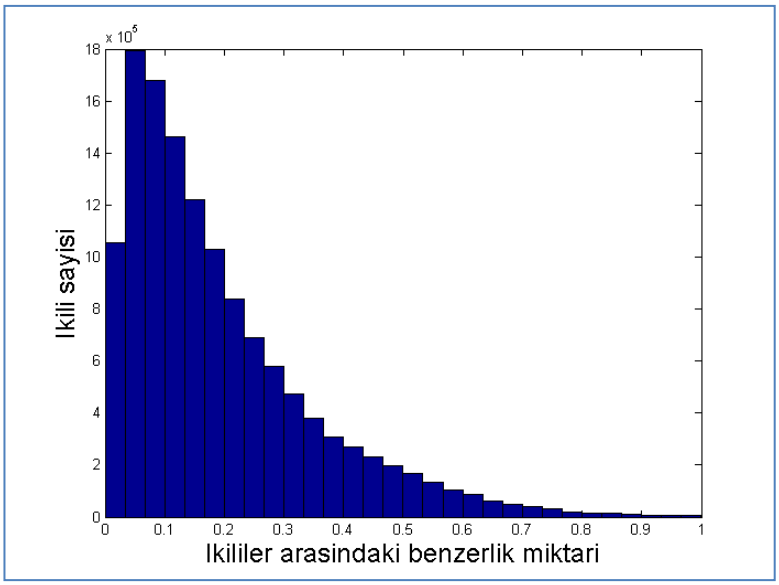
matrisi oluşturulmuştur. Ve bu oluşan yeni matrisin özvektörleri alınarak *K*-merkez ile bu özvektörler üzerinden çıktı öbekleri bulunmuştur.

3.3. OPTİMAL SİGMA SEÇİLİMİ

Öbekleme başarı kriteri olarak ARI (Adjusted Rand Index) [14] kullanılmıştır. ARI öbekler ile sınıf etiketlerinin ne kadar uyuştuğunu ölçmektedir. Kullanılan tüm algoritmalardaki benzerlik fonksiyonunu için Gaussian fonksiyonundaki sigma (σ) parametresi için optimal değer Şekil 4'teki gibi veri kümesini iyi temsil edebilecek (örnek ikililerinin benzerliklerinin 0-1 aralığında dağılabileceği, 0'a ya da 1'e aşırı yığılma olmayacağı) yaklaşık bir aralık seçilerek bu aralıkta eğitim kümesi üzerinde en iyi öbekleme sonucunu (en iyi ARI değerini) veren optimal sigma (σ) araması yapılır.

$$s(xi, xj) = \exp\left(-\frac{\|xi - xj\|^2}{2\sigma^2}\right) \tag{3.1}$$

YSÖ yönteminde esas alınan en küçük maksimum atlama uzaklığı Öklid uzaklığından daha küçük olduğundan, YSÖ için optimal sigma değerinin SÖ' ye göre daha büyük çıkması beklenmelidir.



Şekil 3.4: Optimal sigma (σ) parametresi seçimi için yapılan doğrusal aramayı hızlandırabilmek için benzerliklerin histogramını dikkate almak gerekmektedir.

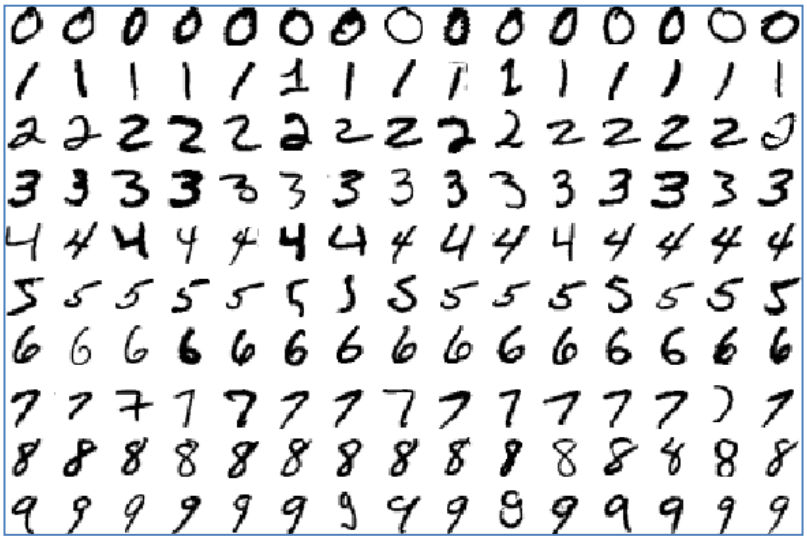
3.4 KULLANILAN VERİ KÜMELERİ

3.4.1. COIL Veri Kümesi

Şekil 3.5: COIL veri kümesi.

COIL [15] veri kümesinde 100 farklı cismin 72 farklı açıdan (her seferinde 5'er derece döndürülerek) çekilmiş toplam 7200 resim bulunmaktadır. 128x128'lik bu resimler boyut sayısını azaltmak için kübik spline yöntemi ile 32x32 = 1024 boyuta indirgenmiştir. Sonuçta 100 öbek arandığı için 100 özvektör kullanılmıştır.

3.4.2. MNIST Veri Kümesi



Şekil 3.6: MNIST veri kümesi.

Chang ve Yeung kullandıkları yol-tabanlı öbekleme yöntemlerini [13] anlatırken MNIST veri kümesini [16] kullanmışlardır. Bizim çalışmamızda da bu veri kümesi aynı

şekilde kullanılmıştır. El yazısı rakam kümesinden oluşan MNIST [16] veri kümesi 28x28 resimlerden (784 boyut) oluşmaktadır. Toplam 60.000 resim içerisinden [13] çalışmasında olduğu gibi iki sınıflı dört farklı veri kümesi oluşturulmuştur. Bu sınıf ikilileri 1-2, 1-4, 1-7 ve 8-9'dur. Her sınıftan 400'er örnek seçilmiştir. Her problem için, sınıf-uyumlu 2 öbek arandığı için 2 özvektör kullanılmıştır.

3.4.3. Çok Özellikli Rakam Veri Kümesi

UCI makine öğrenme deposunda [18] bulunan "Çok özellikli rakam veri kümesi" [17] deneylerde kullanılmıştır. Bu veri kümesi '9' ile '0' arası el yazısı rakamlarının sınıf başına 200 örnek alınarak sayısallaştırılması ile oluşturulmuştur.

Veri setlerinin özellikleri:

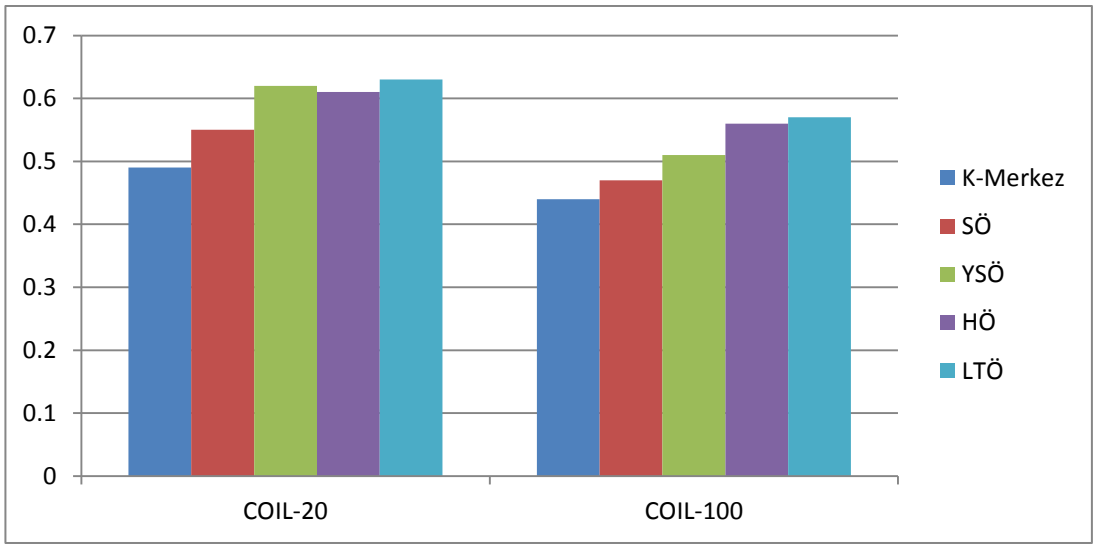
1. mfeat-fou: 76 Fourier katsayıları;

2. mfeat-fac: 216 Profil korelasyonları;

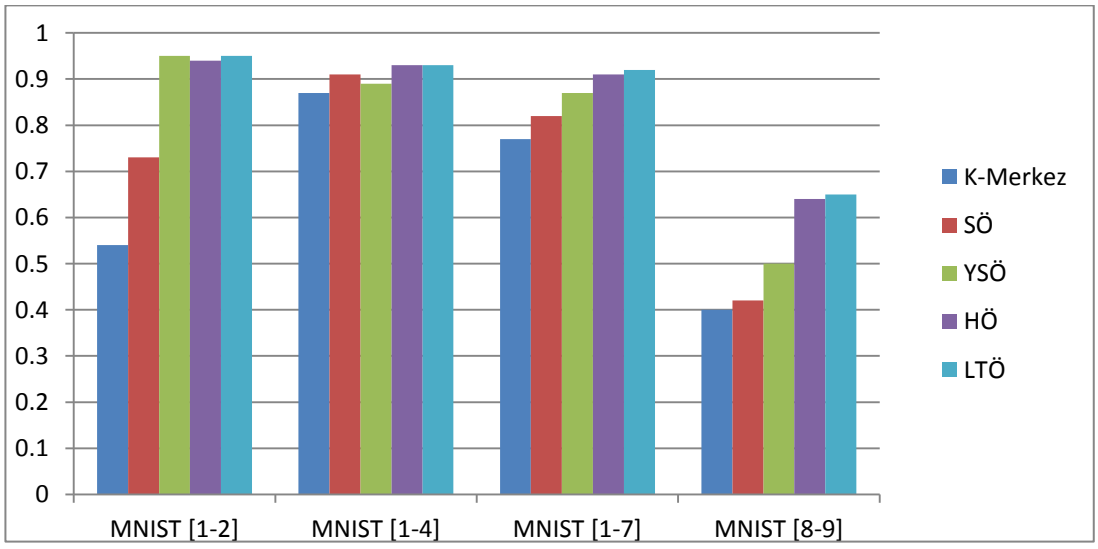
3. mfeat-zer: 47 Zernike momentleri;

4.BULGULAR

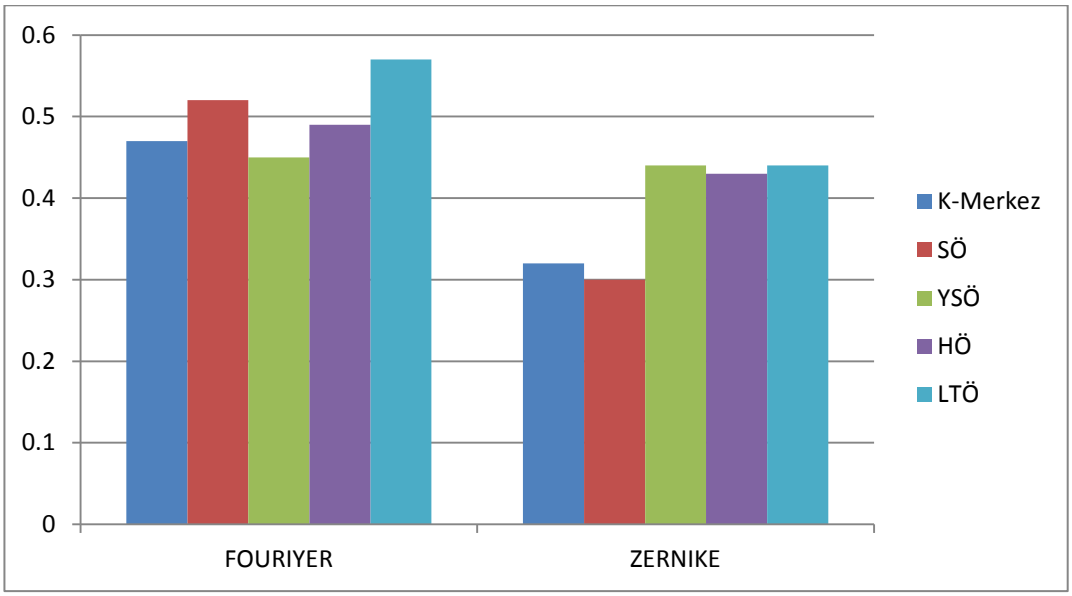
Literatürdeki yöntemler basit geometrik problemlere uygulandıktan sonra gerçek dünya verilerine tez kapsamında önerilen yöntemler ile beraber uygulandı. Bu deneyler sonucunda aşağıdaki sonuçlar elde edilmiştir:



Şekil 4.1: COIL-20 ve COIL-100 veri kümeleri için elde edilen sonuçlar.



Şekil 4.2: MNIST 1-2, 1-4, 1-7 ve 8-9 veri kümeleri için elde edilen sonuçlar.



Şekil 4.3: Fouriyer ve Zernike veri kümeleri için elde edilen sonuçlar.

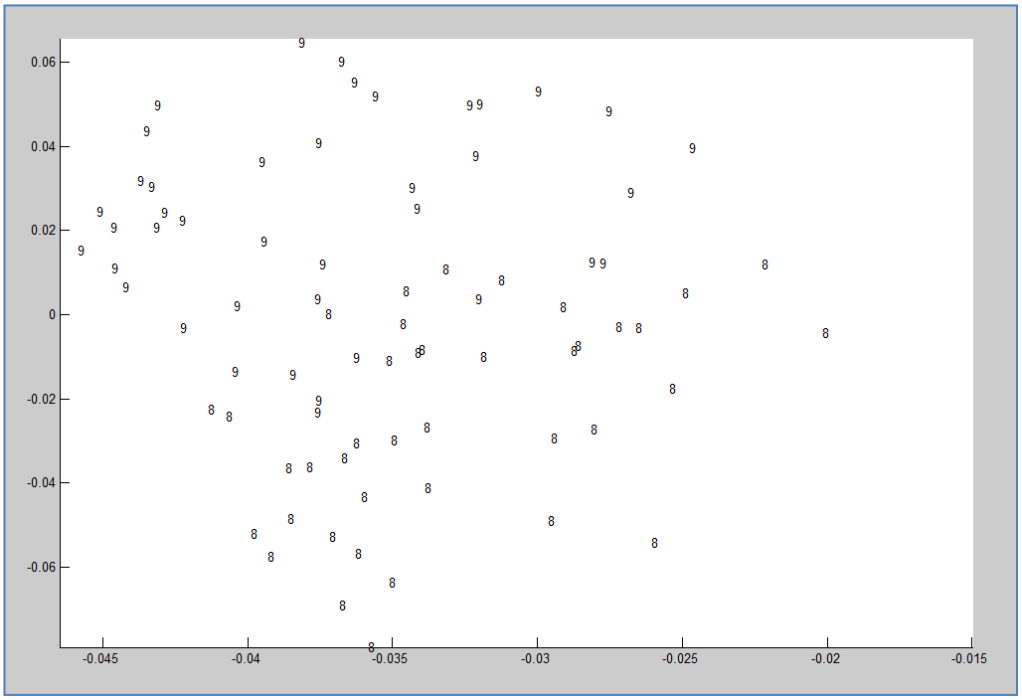
Kullanılan veri kümelerinde yapılan deneyler ile literatürde bulunan öbekleme yöntemleri ve yeni önerilen yöntemler uygulanarak yukarıdaki grafiklerdeki sonuçlar elde edilmiştir. Bu sonuçlara göre önerilen Laplacian matrislerini benzeştiren yeni öbekleme yöntemi diğer öbekleme yapan yöntemlere göre sınıfları daha iyi ayrıştırma yaptığı sonucu elde edilmiştir.

Tablo 4.1: Veri kümeleri için elde edilen Adjusted Rand Index sonuçları.

Veri Kümeleri	K-Merkez	SÖ	YSÖ	HÖ	LTÖ
COIL-20	0.49	0.55	0.62	0.61	0.63
COIL-100	0.44	0.47	0.51	0.56	0.57
MNIST [1-2]	0.54	0.73	0.95	0.94	0.95
MNIST [1-4]	0.87	0.91	0.89	0.93	0.93
MNIST [1-7]	0.77	0.82	0.87	0.91	0.92
MNIST [8-9]	0.40	0.42	0.50	0.64	0.65

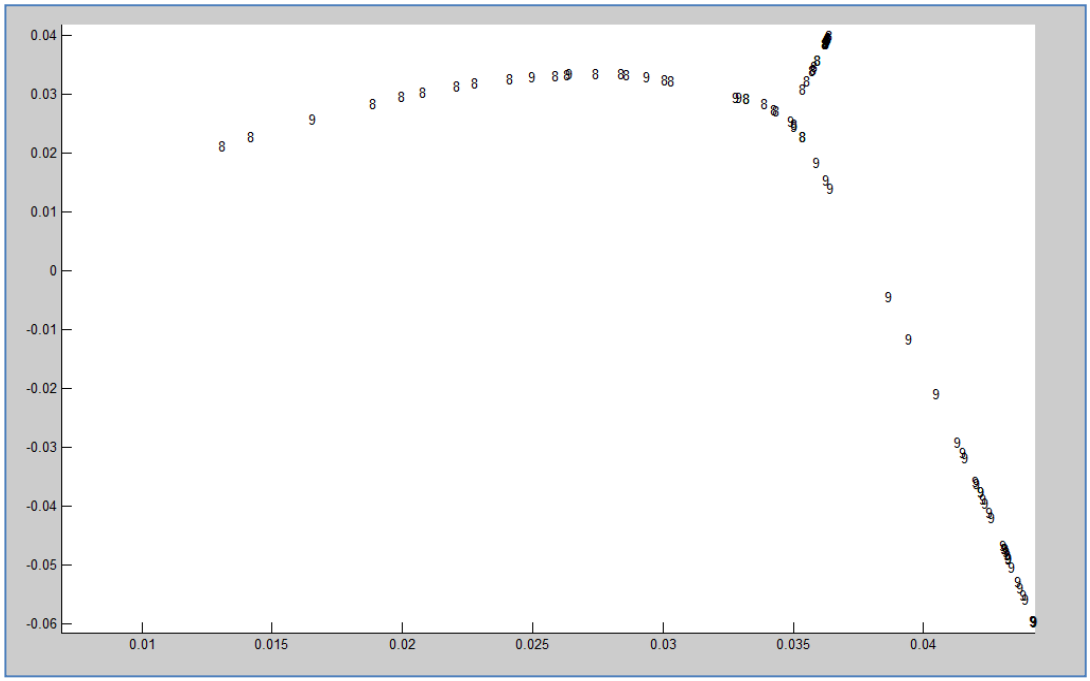
FOURIYER	0.47	0.52	0.45	0.49	0.57
ZERNIKE	0.32	0.30	0.44	0.43	0.44

Yukarıda gösterilen tabloda ki sonuçlar; uygulanan algoritmalar n=10 kez çalıştırılarak alınan ortalama sonuçları göstermektedir. Yukarıda gösterilen tabloda ki sonuçlara göre Spektral öbeklemeye göre yol-tabanlı (bağlanırlık tabanlı) verinin tüm örneklerinin dağılımını da hesaba katan uzaklık ölçümleri daha avantajlı olabilmektedir. Önerilen Laplacian matrislerini benzeştiren yeni öbekleme yöntemi ve önerilen benzerlik matrislerini toplayarak öbekleme yapan hibrit örnekleme sınıfları daha iyi ayrıştırmaktadır.

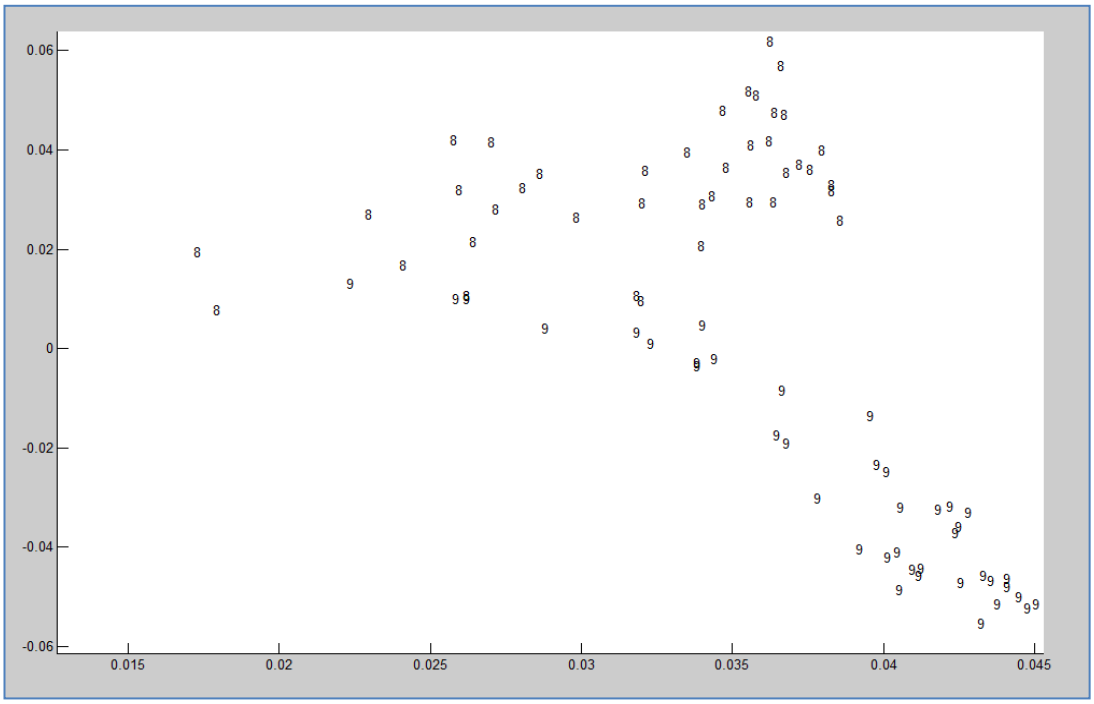


Şekil 4.4: MNIST [8-9] veri kümesi için Spektral Öbekleme projeksiyonları.

Spektral öbekleme ile elde edilen sınıflar için ilk iki özvektör değerleri yukarıdaki şekil 4.4 de gösterilmiştir. Bu projeksiyonda veri kümesinde bulunan 8 ve 9 etiketli sınıfların birbirleri içerisine heterojen olarak dağıldığı gözlenmektedir.

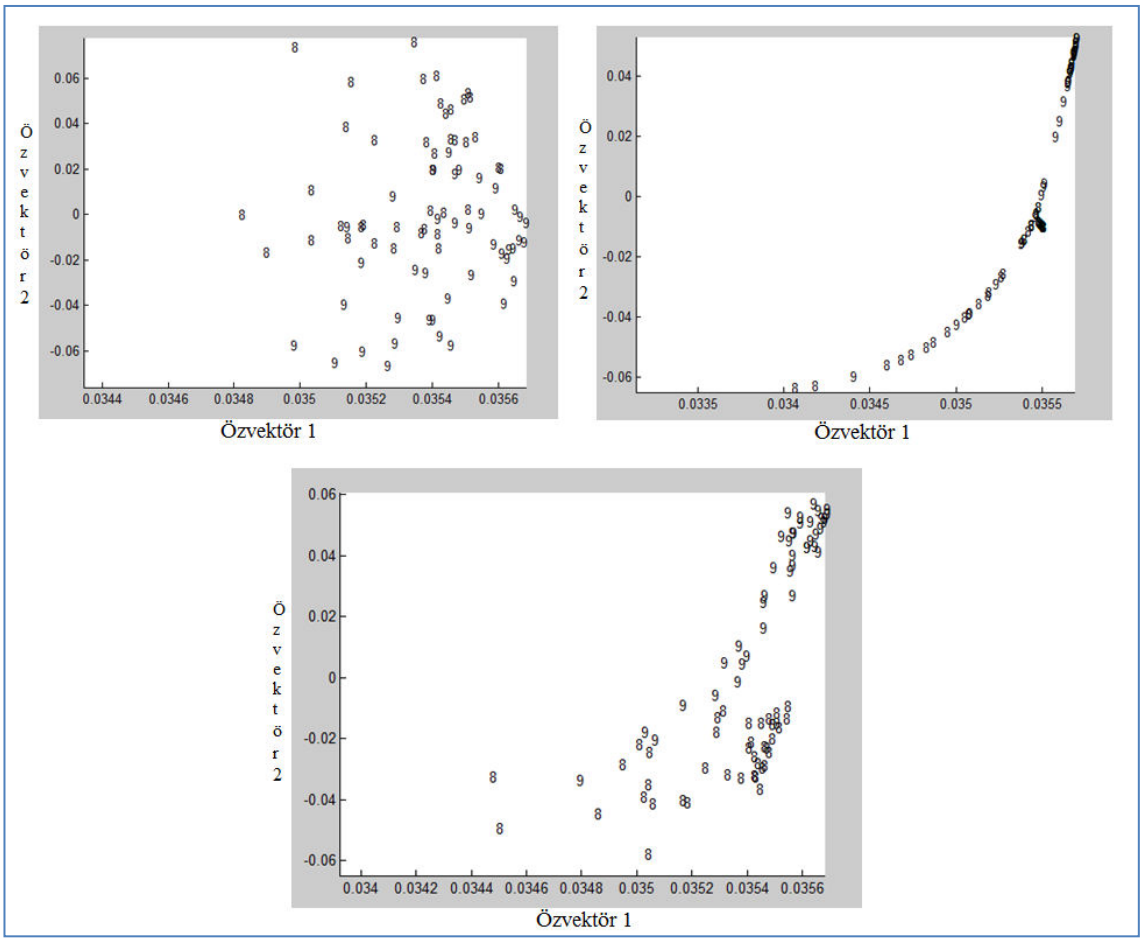


Şekil 4.5: MNIST [8-9] veri kümesi için Yol-tabanlı Spektral Öbekleme projeksiyonları.



Şekil 4.6: MNIST [8-9] veri kümesi için Hibrit Spektral Öbekleme projeksiyonları.

Yukarıdaki şekillerdeki izdüşüm değerlerinden görüldüğü gibi (yalnızca rastgele 80 örnek gösterilmiştir) önerilen hibrit Spektral öbekleme yöntemi sınıfları daha iyi ayrıştırmaktadır.



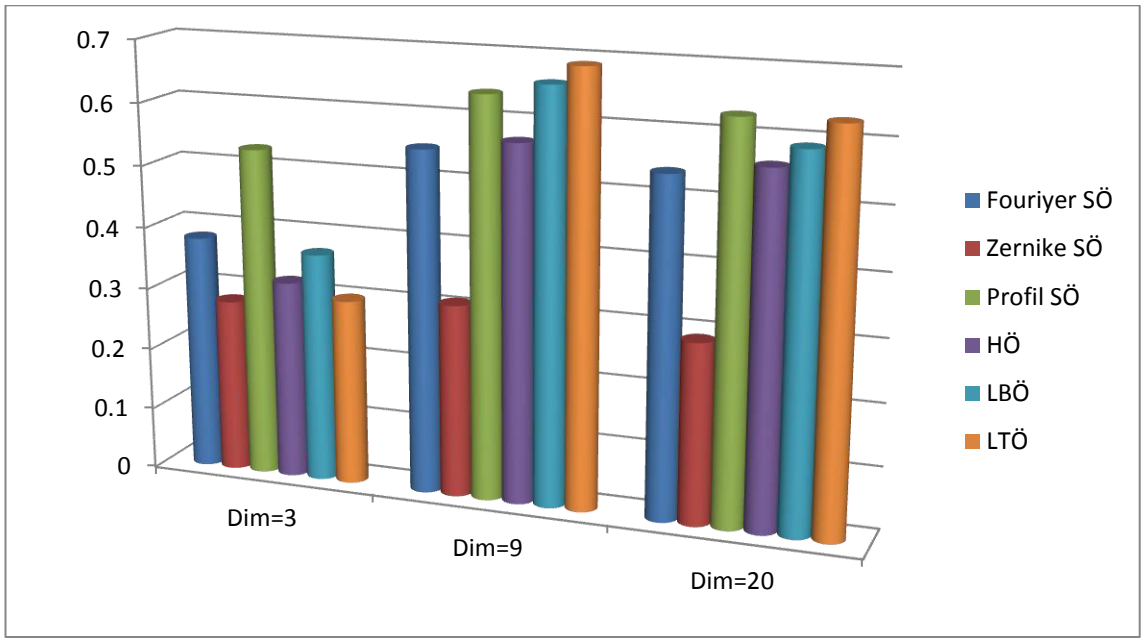
Şekil 4.7: Optimal sigma seçilmeme durumunda oluşan Spektral Öbekleme (Üst Sol), Yol-tabanlı Spektral Öbekleme (Üst Sağ), Hibrit Spektral Öbekleme (Alt) projeksiyonları.

Sekil 4.4, 4.5 ve 4.6 da ki projeksiyonlarda optimal sigma seçilimi yapılarak sigma (σ) değerleri 0.03 ve 0.07 seçilerek daha iyi ayrışabilen sınıflar elde edilmiştir. Sekil 4.7 de spektral öbekleme için sigma değeri 0.001 ve yol tabanlı spektral öbekleme için sigma değeri 0.003 olarak optimal olmayacak şekilde seçildiğinde elde edilen projeksiyonlardaki sınıflar karışık, heterojen olarak dağılmıştır. Dolayısıyla sınıflar arası öbekleme başarısı düşmüştür. Kullanılan tüm algoritmalardaki benzerlik fonksiyonunu Gaussian fonksiyonundaki sigma (σ) parametresi için optimal değer veri kümesini iyi temsil edebilecek yaklaşık bir aralık seçilmelidir.

Bulgulara ek olarak Şekil 2.33 gösterilen Ensemble yöntemler (Minimum Spanning Tree + En Kısa Yol + Model Birleştirme(OR) + MDS + CCA + KNN) kullanılarak gerçek veri kümelerini birleştirerek daha gürbüz çalışan bir öğrenme yapılmak istenmiştir. Bu kullanılan yöntemleri kullanılarak yapılan deneylerden istenilen başarı oranları yakalanılamamıştır. Bu amaçla Tablo 4.2'de gösterilen yeni yöntemlere gidilerek model birleştirmede daha iyi çalışan yeni yöntemler önerilmiştir

Tablo 4.2: Veri kümeleri için kullanılan öbekleme yöntemlerinin ARI sonuçları.

Boyutlar	**Fouriyer SÖ**	**Zernike SÖ**	**Profil SÖ**	**HÖ**	**LBÖ**	**LTÖ**
Dim=3	0.38	0.28	0.53	0.32	0.37	0.30
Dim=7	0.47	0.33	0.65	0.56	0.65	0.60
Dim=8	0.46	0.30	0.65	0.57	0.65	0.58
Dim=9	0.55	0.31	0.64	0.57	0.66	0.69
Dim=10	0.54	0.31	0.64	0.59	0.65	0.68
Dim=12	0.53	0.29	0.65	0.56	0.66	0.67
Dim=14	0.51	0.30	0.63	0.57	0.65	0.66
Dim=15	0.53	0.31	0.67	0.55	0.58	0.66
Dim=17	0.54	0.30	0.67	0.55	0.57	0.67
Dim=19	0.54	0.29	0.62	0.56	0.58	0.63
Dim=20	0.54	0.29	0.63	0.56	0.59	0.63



Şekil 4.8: Fouriyer, Zernike ve Profil modellerinin birleştirilmesi sonucu farklı boyutlar için farklı öbekleme yöntemleri ile elde edilen sonuçlar.

Yukarıda gösterilen tablodaki sonuçlar; uygulanan algoritmalar n=10 kez çalıştırılarak alınan ortalama sonuçları göstermektedir. İlk 3 sütün sırasıyla Fouriyer, Zernike ve Profil veri kümeleri için Spektral öbekleme sonuçları farklı boyutlar için elde edilmiştir. Hibrit örneklemede (HÖ) 3 ayrı modeller birleştirilerek Ensemble öğrenme yapılmıştır. Yani 3 modelin benzerlik matrisleri toplanarak birleştirilmiştir.

Laplacian birleştirmeli öbeklemede (LBÖ) incelen 3 veri kümesinin Laplacian matrisinin özdeğerleri yan yana toplanması ve bu matrisin K-Merkez algoritmasında öbeklenmesi ile elde edilmiştir.

Laplacian tanımlı öbeklemede (LTÖ) incelen 3 veri kümesinin Laplacian matrisleri benzeştirilerek birleştirilmiştir. Birleşen bu matris Spektral öbekleme algoritmasında kümelere ayrılarak sonuçlar elde edilmiştir.

Tablo 2'de alınan sonuçlara göre boyut dim=9 olarak seçildiğinde uygulanan tez kapsamında önerilen Ensemble model birleştirme algoritmaları literatürde var olan tek model üzerinden öbekleme yapan Spektral öbeklemeden daha gürbüz yöntemler olduğu deneylerden gösterilmektedir.

5.TARTIŞMA VE SONUÇ

Spektral öbeklemenin girdi olarak kullandığı benzerlik matrisini hesaplamak için yalnızca Öklid uzaklığı kullanmanın avantajlı olduğu durumlar olduğu gibi yol-tabanlı (bağlanırlık tabanlı) verinin tüm örneklerinin dağılımını da hesaba katan uzaklık ölçümleri de avantajlı olabilmektedir. Bu çalışmada, bunları beraber kullanan hibrit bir yöntem önerilmiştir. Tablo 1'de öbek-sınıf uyumunu ölçen ARI indeks değerlerinden ve Şekil 4.1'de projeksiyon değerlerinden görüldüğü gibi, önerilen hibrit yöntem sınıfları daha iyi ayrıştırmaktadır.

Uygulaması yapılan Yol tabanlı Spektral öbekleme (YSÖ) ve hibrit yöntemin daha büyük veri kümelerinde daha verimli çalışması için Kritik noktalı yol tabanlı Spektral öbekleme önerilmiştir. Bu algoritma verim acısından Spektral öbeklemeden iyi çalışmış olmasına rağmen başarı açısından geri kalmaktadır. Bu yöntem ileriki çalışmalarda geliştirilmek istenmektedir.

Tablo 2'de alınan projeksiyon sonuçlara göre en optimum çalışan boyutta veya diğer boyutlarda boyut seçimi yapıldığında tez kapsamında önerilen Ensemble model birleştirme algoritmaları literatürde var olan tek model üzerinden öbekleme yapan Spektral öbeklemeden daha gürbüz yöntemler olduğu deneylerden gösterilmektedir.

KAYNAKLAR

[1]. Science_Clustering, Wikipedia, *The Free Encyclopedia*, (2015). [Online; accessed 04-February-2015].

[2]. Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 395-416.

[3]. Özdamar, K. (1999). Paket programlar ile istatistiksel veri analizi. *Kaan Kitabevi, Eskişehir*, *535*.

[4]. Galımberti, G., Soffritti, G. (2007). Model-Based Methods to Identify Multiple Cluster Structures In a Data Set, *Computational Statistics & Data Analysis*, 52: 520-536.

[5]. Similarity_matrix, Wikipedia, *The Free Encyclopedia*, (2015). [Online; accessed 04-February-2015].

[6]. Chung, F., Lu, L., & Vu, V. (2003). Spectra of random graph with given expected degrees. *Proceedings of the National Academy of Sciences*, *100*(11), 6313-6318.

[7]. Blashfield, R.K., Aldenferder, M.S. (1978). The Literature on Cluster Analysis, *Multivariate Behavioral Research*, 13, 271-295.

[8]. Basser, P. J., Pajevic, S., Pierpaoli, C., Duda, J., & Aldroubi, A. (2000). In vivo fiber tractography using DT-MRI data. *Magnetic resonance in medicine*, 44(4), 625-632.

[9]. Hubert, L. (1974). Approximate evaluation techniques for the single-link and complete-link hierarchical clustering procedures. *Journal of the American Statistical Association*, *69*(347), 698-704.

[10]. Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, *2*, 849-856.

[11]. Chung, F. R. (1996). Spectral graph theory (CBMS regional conference series in mathematics, No. 92).

[12]. Floyd-Warshall algorithm Wikipedia, *The Free Encyclopedia*, 2015. [Online; accessed 16-February-2015].

[13]. Chang, H., & Yeung, D. Y. (2008). Robust path-based spectral clustering.*Pattern Recognition*, *41*(1), 191-203.

[14]. Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, *66*(336), 846-850.

[15]. COIL-100 Dataset (2015), http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php, [Online; accessed 01-February-2015].

[16]. MNIST Dataset (2015), http://yann.lecun.com/exdb/mnist/, [Online; accessed 01-February-2015].

[17]. Van Breukelen, M., Duin, R. P., Tax, D. M., & Den Hartog, J. E. (1998). Handwritten digit recognition by combined classifiers. *Kybernetika*, *34*(4), 381-386.

[18]. Asuncion, A., & Newman, D. J. (2007). UCI Machine Learning Repository [http://www. ics. uci. edu/~ mlearn/MLRepository. html]. Irvine, CA: University of California. *School of Information and Computer Science*.

[19]. Everitt, S.B., Landau, S., Leese M. (2001). Cluster Analysis. *Oxford University Press Inc*, 122, New York.

[20]. Tatlıdil, H. (1996). Uygulamalı çok değişkenli istatistiksel analiz. *Cem Web Ofset, Ankara*, *329*

[21]. Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*(5500), 2319-23

[22]. Alpaydin, E. (2004), Introduction to machine learning. *Cover, Copyright Page, Table of Contents for*, 1-327.

[23]. Mutual Information Wikipedia, *The Free Encyclopedia*, 2015. [Online; accessed 24-February-2015].

[24]. Rand indexWikipedia, *The Free Encyclopedia*, (2015). [Online; accessed 24-February-2015].

[25]. Handwritten Dataset (2015), https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits, [Online; accessed 01-February-2015].

[26]. Lawrence, H., & Phipps, A. (1985). Comparing partitions. *Journal of Classification*, *2*(1), 193-218.

[27]. Brown, G. (2010). Ensemble learning. In *Encyclopedia of Machine Learning*(pp. 312-320). Springer US.

[28]. Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 242-263.

.

[29]. Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, *7*(1), 48-50.

[30]. Fischer, B., & Buhmann, J. M. (2003). Bagging for path-based clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(11), 1411-1415.

[31]. Fischer, B., & Buhmann, J. M. (2003). Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(4), 513-518

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı	Kadir Güzel
Uyruğu	T.C.
Doğum tarihi, Yeri	21.11.1984, Bayburt
Telefon	05369254198
E-mail	kadirguzell@hotmail.com
Web adres	-

Eğitim

Derece	Kurum/Anabilim Dalı/Programı	Yılı
Yüksek Lisans	İ.Ü. Fen Bilimleri Enstitüsü/ Mühendislik Fakültesi / Bilgisayar Mühendisliği	2013-…
Lisans	Karadeniz Teknik Üniversitesi / Bilgisayar Mühendisliği	2004-2009
Lise	Erzurum Lisesi	1999-2002

Makaleler / Bildiriler

Guzel, K., & Kursun, O. (2015, May). Improving spectral clustering using path-based connectivity. In *Signal Processing and Communications Applications Conference (SIU), 2015 23th* (pp. 2110-2113). IEEE.