MİMAR SİNAN GÜZEL SANATLAR ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SPEKTRAL KÜMELEME VE FARKLI UZAKLIK FONKSİYONLARININ SPEKTRAL KÜMELEMEYE ETKİSİNİN İNCELENMESİ

YÜKSEK LİSANS TEZİ

Hazırlayan: Mustafa EROĞLU

Anabilim Dalı: MATEMATİK

Programı: MATEMATİK YÜKSEK LİSANS

Tez Danışmanı: Dr. Öğr. Üyesi Gülay İlona TELSİZ KAYAOĞLU

ŞUBAT 2023

SPEKTRAL KÜMELEME VE FARKLI UZAKLIK FONKSİYONLARININ SPEKTRAL KÜMELEMEYE ETKİSİNİN İNCELENMESİ

ÖZET

Bu tezde spektral kümeleme ve farklı uzaklık fonksiyonlarının spektral kümelemeye etkisi incelenecektir. Spektral kümelemede kullanılan Öklid metriği yerine farklı uzaklık fonksiyonları ele alarak daha başarılı bir kümeleme elde edip edilemeyeceğini örnek veri setleri üzerinden incelenecektir.

Birinci bölümde spektral kümelemenin tarihinden bahsedilecek ve spektral kümelemeye giriş yapılacaktır. İkinci bölümde benzerlik grafları anlatılacaktır. Üçüncü bölümde Laplasyen matrisleri anlatılacaktır. Dördüncü bölümde graf kesitleri incelenecektir. Beşinci bölümde spektral kümeleme algoritmalarını anlatılacaktır. Altıncı bölümde ise Python ile 3 veri seti üzerinde farklı uzaklık fonksiyonları ele alınarak uygulama yapılacaktır.

Anahtar Kelimeler: Spektral Kümeleme, Uzaklık Fonksiyonları

SPECTRAL CLUSTERING AND INVESTIGATION OF THE EFFECT OF DIFFERENT DISTANCE FUNCTIONS IN SPECTRAL CLUSTERING

SUMMARY

In this thesis, spectral clustering and the effect of different distance functions in spectral clustering will be examined. Instead of the Euclidean metric used in spectral clustering, it will be examined on sample data sets whether a more successful clustering can be obtained by considering different distance functions.

In the first chapter, the history of spectral clustering will be mentioned and an introduction to spectral clustering will be made. In the second part, similarity graphs will be explained. In the third chapter, Laplacian matrices will be explained. In the fourth chapter, graph partitioning will be examined. In the fifth chapter, spectral clustering algorithms will be explained. In the sixth chapter, different distance functions will be handled with Python on 3 data sets and an application will be made.

Keywords: Spectral Clustering, Distance Functions

Önsöz

Öncellikle bana her türlü desteğini esirgemeyen tez danışmanım Dr. Öğr. Üyesi Gülay İlona Telsiz Kayaoğlu'na sonsuz teşekkür ederim.

Ayrıca bu çalışmayı yapmama ilham olan Dr. Öğr. Üyesi Gülay İlona Telsiz Kayaoğlu, Dr. Öğr. Üyesi Özlem Yılmaz, Doç. Dr. Özgür Martin ve Prof. Dr. İlker Birbil'e teşekkür ederim.

Her zaman yanımda olan aileme, Mimar Sinan Güzel Sanatlar Üniversitesi'nden. Umutcan Erdur'a ne kadar teşekkür etsem azdır.

Mustafa Eroğlu

İçindekiler

Bölüm 1

SPEKTRAL KÜMELEMEYE GİRİŞ

Bu bölümde spektral kümelemenin tarihi ve spektral kümelemeye giriş anlatılacaktır.

Spektral kümeleme, bir grafikteki noktaları benzer özelliklere sahip gruplara ayırmak için kullanılan bir yöntemdir. Bu yöntem, grafikteki noktaların komşuluk ilişkilerine göre gruplandırılmasını sağlar ve bu gruplara ayırma işlemini gerçekleştirilirken, graf teorisi ve lineer cebir gibi matematiksel kavramları kullanılır.

1970 yılında Kenneth M. Hall yaptığı çalışmaların sonucunda bir grafın yapısı ile matrisin spektral özellikleri arasındaki ilişkiyi ortaya koyması ile başlamıştır diyebiliriz. Kenneth M. Hall r boyutlu Öklid uzayında n düğümün yerleştirmesi problemini incelemiştir [1].

Ayrıca Donath ve Hoffman 1973 yılında yayınlanan çalışmalarında parçalanış bulmak için grafların benzerlik matrisinin özvektörlerini kullanmayı önermişlerdir [2].

Yine 1973 yılında Fiedler grafı iki eş parçaya ayırmanın Laplasyen matrisinin ikinci özvektörü ile yakından ilişkili olduğunu göstermiştir [3].

Spektral kümeleme kavramı ise, matematikçi F. R. K. Chung tarafından 1997 yılında yayınlanan bir makalede tanımlanmıştır [4]. Bu makalede, Chung, grafiklerdeki noktaları bir araya getirme işlemini "kümeleme" olarak adlandırmış ve bu işlemin spektral bir yöntemle gerçekleştirilebileceğini ileri sürmüştür. Spektral kümeleme Shi, Malik ve Ng, Jordan, Weiss tarafından bir makine öğrenmesi yöntemi olarak gündeme getirilmesiyle tekrar popülerlik kazanmıştır [5],[6].

Spektral kümeleme, ilk olarak sosyal ağ analizi gibi alanlarda kullanılmıştır. Ancak zaman içinde, bu yöntem birçok farklı alanda da kullanılmaya başlanmıştır. Örneğin,

makine öğrenimi, veri madenciliği ve görüntü işleme gibi alanlarda da spektral kümeleme yöntemi kullanılmaktadır. Ayrıca kümeleme istatistik, bilgisiyar bilimi, biyoloji, psikoloji ve birçok alanda keşifsel veri analizi için en yaygın kullanılan tekniklerden biridir.

Spektral kümeleme yöntemi, grafikteki noktaların konumlarını belirlemek için bir Laplasyen operatörü kullanır. Bu operatör, noktalar arasındaki ilişkileri inceler ve bunları bir matris olarak gösterir. Daha sonra, bu matrisin en küçük özvektörleri hesaplanır ve bu özvektörler, graftaki noktaları benzer özellikleri olan gruplara ayırmak için kullanılır.

Spektral kümeleme algoritması, ayrıntıları diğer bölümlerde açıklanacak olan şu adımlardan oluşmaktadır.:

1. Benzerlik grafı hesaplanır. Bu aşamada ε-komşuluk grafı, k-NN grafı veya tam bağlantılı graftan biriyle noktalar arası benzerlik ilişkisi kurulur. Benzerlik Grafları Bölüm 2'de açıklanmıştır.

2. Veriler daha düşük boyutlu uzaya yansıtılır. Bunu yapabilmek için Laplasyen matris bulunur. Bu matrisin nasıl oluşturulduğu Bölüm 3'te verilmiştir.

3. Laplasyen matrisinin özdeğer ve özvektörleri hesaplanır. k-means algoritmasıyla veri kümelenir. 4. bölümde özdeğer ve özvektörlerin kümeleme ile ilişkisi teorik olarak anlatılmıştır.

Hiyerarşik kümeleme ve k-means gibi geleneksel algoritmalar ile karşılaştırıldığında spektral kümelemenin birçok avantajı vardır. Spektral kümeleme ile geleneksel algoritmalar karşılaştırıldığında daha iyi performans gösterir. Buna son bölümde değinilecektir.

Bölüm 2

BENZERLİK GRAFLARI

$x_1, \ldots, x_n$ veri kümesi ve x_i ile x_j veri noktaları arasındaki s_{ij} benzerliği verilsin. Kümelemenin amacı aynı kümedeki noktaların benzer ve farklı kümedeki noktaların birbirinden farklı olmasıdır. Eğer sadece verilerin birbirlerine olan benzerliği biliniyorsa veri $G = (V, E)$ benzerlik grafı şeklinde temsil edilebilir. Graftaki her v_i küşesi x_i veri noktasını temsil eder. İki köşe bağlıdır eğer x_i ve x_j arasındaki s_{ij} benzerliği pozitiftir veya belirli bir eşik değerden büyüktür. Graftaki kenarlar s_{ij} ile ağırlaklandırılır. Kümeleme problemi graf parçalama problemine dönüştürülebilir. Burada amaç farklı kümeler arasındaki kenarlar düşük ağırlığa ve aynı küme içindeki kenarlar yüksek ağırlığa sahip olacak şekilde grafı parçalamaktır. İlk önce bazı temel graf teoriyle ilgili tanımlar verilecektir.[7]

2.1 Tanımlar

$G = (V, E)$, köşe kümesi $V = \{v_1, \ldots, v_n\}$, kenar kümesi $E \subseteq \{\{u, v\} \,|u, v \in V\}$ olan yönsüz çizge olsun. G grafının ağırlıklı olduğunu varsayıyoruz yani v_i ve v_j köşeleri arasındaki her kenarın negatif olmayan $w_{i_j} \geq 0$ ağırlığına sahiptir. Grafın ağırlıklı benzerlik matrisi $W = (w_{ij})_{i,j=1,2,\ldots,n}$ matrisidir. Eğer $w_{ij} = 0$ ise bu v_i ve v_j köşelerinin bir kenarla bağlı olmadığı anlamına gelir. G yönsüz olduğu için $w_{ij} = w_{ji}$ sağlanmalıdır. Bir v_i köşesinin derecesi aşağıdaki gibi tanımlanır.

$$d_i = \sum_{j=1}^{n} w_{ij}$$

Aslında yukardaki toplam v_i köşesine komşu tüm köşeler için de doğrudur. Çünkü komşu olmayan v_j köşeleri için $w_{ij} = 0$'dır. D derece matrisi köşegen üzerinde $d_1, \ldots, d_n$ derecelerine sahip olan köşegen matris olarak tanımlanır.

$A \subset V$ köşelerinden oluşan bir altküme verildiğinde, $V \backslash A$ tümleyenini $\overline{A}$ ile gösteriyoruz.

Tanım 2.1.1. *Her girdisi için $v_i \in A$ ise $f_i = 1$ aksi durumda $f_i = 0$ olacak şekilde tanımlanan $\mathbb{1}_A = (f_1, \ldots, f_n)' \in \mathbb{R}^n$ vektörüne indikatör vektörü denir.*

Tanım 2.1.2. *$A, B \subset V$ altkümeleri için ağırlıklı yakınlık matrisi:*

$$W(A, B) := \sum_{i \in A, j \in B} w_{ij} \tag{2.1.1}$$

olarak tanımlarız.

Tanım 2.1.3. *Bir kümenin büyüklüğü:*
Bir $A \subset V$ altkümesinin büyüklüğünü ölçmenin iki farklı yolu vardır:

$$|A| = A\text{'daki köşe sayısı}$$
$$vol(A) = \sum_{i \in A} d_i$$

$|A|$,A'nın boyutunu köşe sayısıyla, $vol(A)$ ise A'daki köşelere bağlı tüm kenarların ağırlıklarını toplayarak ölçer.

Tanım 2.1.4. *Bir grafın $A \subset V$ altkümesindeki herhangi iki köşe, tüm ara noktalar da A'da olacak şekilde bir yol ile birleştirilebiliyorsa bağlantılıdır denir.*

Tanım 2.1.5. *A bağlantılıdır ve A ve $\overline{A}$'daki köşeler arasında bağlantı yoksa A altkümesi bağlantılı bileşendir denir.*

Tanım 2.1.6. *$A_i \cap A_j = \emptyset$ ve $A_1 \cup \ldots \cup A_k = V$ sağlanıyorsa $A_1, \ldots, A_k$ V grafının bir parçalanışıdır.*

2.2 Farklı Benzerlik Grafları

Benzerlik grafını oluştururken amacımız veri noktaları arasındaki yerel komşuluk ilişkilerini modellemektir.[7] Verilen $x_1, \ldots, x_n$ veri noktaları kümesinin s_{ij} ikili benzerlikler veya d_{ij} ikili mesafelerle bir grafa dönüştürmenin birçok yolu vardır. Bunlardan en çok kullanılan üçü aşağıda açıklanmıştır: **ε-komşuluk grafı**: Herhangi iki nokta arasındaki mesafe ε'dan küçük ise birleştirilir,aksi durumda birleştirilmez.

$$w_{ij} = \begin{cases} 1, & \|x_i - x_j\| < \varepsilon \text{ ise} \\ 0, & \text{aksi durumda} \end{cases}$$

k-NN grafı:Burada amacımız v_j, v_i'nin k-en yakın komşuluğunda veya v_i, v_j'nin k-en yakın komşuluğundaysa v_i köşesi ile v_j köşesini birleştirilir.

$$w_{ij} = \begin{cases} 1, & x_i \in kNN(x_j) \text{ veya } x_j \in kNN(x_i) \text{ ise} \\ 0, & \text{aksi durumda} \end{cases}$$

Tam bağlantılı graf(Gaussian benzerlik grafı): Burada bütün noktalar birbirleriyle bağlanmakta ve pozitif s_{ij} ile ağırlaklandırılmaktadır.

$$w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

Burada $\sigma > 0$ kullanıcı tarafında seçilen parametredir.

Bölüm 3

LAPLASYEN MATRİSLERİ

Spektral kümeleme için ana araçlar Laplasyen matrislerdir. Spektral graf teorisi bu matrisleri inceleyen ana alandır. Bu bölümde literatürde yer alan farklı Laplasyen matrislerinin tanımı verilecek ve en önemli özellikleri ispatlanacaktır.
$G = (V, E)$ yönsüz, ağırlıklı graf olsun. G'nin köşe sayısı n olsun. G'nin ağırlık matrisi $W = (w_{ij})_{i,j=1,2,\ldots,n}$ olsun ve w_{ij}, v_i köşesi ile v_j köşesi arasındaki ağırlığı temsil etsin. Ağırlıklar ilgili şu varsayımlar yapılacaktır. $w_{ij} \geq 0$, $w_{ii} = 0$, $w_{ij} = w_{ji}$.
G'nin derece matrisi $D = (d_i)_{i=1,2,\ldots,n}$ köşegen matristir ve köşegen üzerinde i'nci terimi d_i'dir [7].

3.1 Tanımlar

Tanım 3.1.1. *Bir $f : V \to V$ lineer fonksiyonu verilsin. Eğer bir $v \in V$ ve $\lambda \in \mathbb{R}$ için $f(v) = \lambda v$ oluyorsa, v'ye f'nin bir özvektörü ve eğer $v \neq 0_V$ ise λ'ya da v'nin bir özdeğeri denir. (Eğer $v \neq 0_V$ bir özvektörse λ'ya v'nin özdeğeri denir.)*

Tanım 3.1.2. *Bir matrisin bütün özdeğerlerinin kümesine bu matrisin spektrumu denir.*

Tanım 3.1.3. *Bir özdeğere karşılık gelen lineer bağımsız özvektörlerin sayısına özvektörün "geometrik katlılığı" denir.*

Tanım 3.1.4. *Bir özdeğerin karakterisitk denklemdeki kuvvetine (spektrumdaki görünme sayısına) özdeğerin cebirsel katlılığı denir.*

3.2 Normalize Edilmemiş Laplasyen Matrisi

Normalize edilmemiş Laplasyen matrisi

$$L = D - W$$

olarak tanımlanır. Aşağıdaki önerme, spektral kümeleme için önemlidir.

Önerme 3.2.1. ***(L'nin özellikleri)****L matrisi aşağıdaki özellikleri sağlar:*

i. Her $v \in \mathbb{R}^n$ *vektörü aşağıdaki eşitliği sağlar*

$$v^\top Lv = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(v_i - v_j)^2 \tag{3.2.1}$$

ii. L simetriktir ve pozitif yarı tanımlıdır.

iii. L'nin en küçük özdeğeri 0*'dır, karşılık gelen özvektör sabit bir* $\mathbb{1}$ *vektörüdür.*

vi. L'nin n tane negatif olmayan, reel değerli özdeğeri vardır ve $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ *sağlanır [7].*

Kanıt. i. $d_i = \sum_{j=1}^{n} w_{ij}$ tanımından

$$\begin{aligned}
v^\top Lv &= v^\top Dv - v^\top Wv \\
&= \sum_{i=1}^{n} d_i v_i^2 - \sum_{i,j=1}^{n} v_i v_j w_{ij} \\
&= \frac{1}{2}(\sum_{i=1}^{n} d_i v_i^2 - 2\sum_{i,j=1}^{n} v_i v_j w_{ij} + \sum_{j=1}^{n} d_j v_j^2) \\
&= \frac{1}{2}(\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} v_i^2 - 2\sum_{i,j=1}^{n} v_i v_j w_{ij} + \sum_{j=1}^{n}\sum_{i=1}^{n} w_{ji} v_j^2) \\
&= \frac{1}{2}(\sum_{i,j=1}^{n} w_{ij} v_i^2 - 2\sum_{i,j=1}^{n} v_i v_j w_{ij} + \sum_{i,j=1}^{n} w_{ij} v_j^2) \\
&= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(v_i - v_j)^2
\end{aligned}$$

ii. D ve W simetrik olduğundan L simetriktir. Her $v \in \mathbb{R}^n$ için $v^\top Lv \geq 0$ ve (i)'den dolayı L pozitif yarı tanımlıdır.

iii. Her $v \in \mathbb{R}^n$ için $v^\top Lv \geq 0$ olduğundan her v özvektörü için de $v^\top Lv = \lambda v^\top v \geq 0$ olur. Her v için $v^\top v > 0$ olduğundan $\lambda \geq 0$'dır. Buradan L'nin her λ özdeğeri için $\lambda \geq 0$ doğrudur. Bundan dolayı da en küçük özdeğer 0'dır.$Lu = \lambda u$ eşitliğinden $\lambda = 0$ ise $Lu = \lambda u = 0$'dır. L'nin her satırının toplamı $d_i - \sum_{j=1}^{n} w_{ij} = 0$ sağlandığından $u = \mathbb{1}$'dir.

iv. Her $n \times n$ simetrik matris n tane özdeğere sahiptir. (3)'den dolayı en küçük özdeğer 0 olduğundan $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ sağlanır. □

Laplasyen matrisi köşegenleştirilebilir olduğundan cebirsel katlığı geometrik katlılığına eşittir, bu nedenle bundan sonra sadece "katlılık" terimini kullanacağız.
G grafının bağlantılı bileşen sayısı ilgili teoremin ispatı [2] nolu kaynakta bulunabilir.

Teorem 3.2.1. *Laplasyen matrisinin sıfır özdeğerlerinin sayısı(yani, 0 özdeğerinin katlılığı), G grafının bağlantılı bileşenlerinin sayısına eşittir.[2]*

3.3 Normalize Edilmiş Laplasyen Matris

Literatürde normalize edilmiş Laplasyen matris olarak iki farklı matris vardır. Bu iki matris aşağıdaki gibi tanımlanır:

$$L_{sym} := D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}$$
$$L_{rw} := D^{-1}L = I - D^{-1}W$$

Yukarıda ispatlanan teoremin benzeri L_{sym} ve L_{rw}matrisleri için de elde edilebilir.

Önerme 3.3.1. *(L_{sym} **ve** L_{rw}**'nin özellikleri)**Normalize edilmiş Laplasyen matris aşağıdaki özellikleri sağlar:*
i. Her $v \in \mathbb{R}^n$ aşağıdaki eşitliği sağlar

$$v^\top L_{sym} v = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left(\frac{v_i}{\sqrt{d_i}} - \frac{v_j}{\sqrt{d_j}}\right)^2 \tag{3.3.1}$$

ii. L_{rw}'nin λ özdeğerine karşılık gelen u özvektörüdür ancak ve ancak L_{sym}'nin λ özdeğerine karşılık $w = D^{-1/2}u$ özvektörüdür.
iii. L_{rw}'nin λ özdeğerine karşılık gelen u özvektörüdür ancak ve ancak eğer λ özdeğeri ve u özvektörü genelleştirilmiş özdeğer problemini $Lu = \lambda Du$ çözer.
iv. L_{rw}'nin 0 özdeğerine karşılık gelen $\mathbb{1}$ özvektörüdür. L_{sym}'nin 0 özdeğerine karşılık gelen $D^{-1/2}\mathbb{1}$ özvektörüdür.
v. L_{sym} ve L_{rw} matrisleri pozitif yarı tanımlıdır ve n tane negatif olmayan, reel özdeğerleri vardır ve $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ sağlanır [7].

Teorem 3.3.1. *L_{sym} ve L_{rw} Laplasyen matrislerinin sıfır özdeğerlerinin sayısı(yani, 0 özdeğerlerinin katlılığı),G grafının bileşenlerinin sayısına eşittir [7].*

Bölüm 4

GRAF KESİTLERİ

S benzerlik matrisi ve W ağırlıklı yakınlık matrisi verilsin. Grafın parçalanışını bulmanın yolu bu bölümde açıklanacak olan optimizasyon problemini çözmektir. Bu mincut yaklaşımı olarak bilinmektedir.
Ağırlıklı yakınlık matrisi $W(A,B) = \sum_{i\in A, j\in B} w_{ij}$ ve $\overline{A}$, A'nın tümleyeni olsun. k tane altküme için, mincut yaklaşımı $cut(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k} W\left(A_i, \overline{A}_i\right)$'yi minimize eden $A_1,\ldots,A_k$ parçalınışı seçmektir.
Optimizasyon probleminde kullanılan en yaygın iki amaç fonksiyonu oransal kesim RatioCut ve normalize edilmiş kesim Ncut'tır. RatioCut'ta bir grafın A altkümesinin boyutu $|A|$ ile ölçülür. Ncut'ta ise $vol(A)$ ile ölçülür. $A_1,\ldots,A_k$ parçalınışı için RatioCut ve Ncut tanımlarını verelim:

$$RatioCut(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W\left(A_i,\overline{A_i}\right)}{|A_i|} = \sum_{i=1}^{k}\frac{cut\left(A_i\overline{A_i}\right)}{|A_i|}$$

$$Ncut(A_1,\ldots,A_k) := \frac{1}{2}\sum_{i=1}^{k}\frac{W\left(A_i,\overline{A_i}\right)}{vol(A_i)} = \sum_{i=1}^{k}\frac{cut\left(A_i\overline{A_i}\right)}{vol(A_i)}$$

4.1 Oransal kesim (RatioCut) yaklaşımı:

Oransal kesim (Ratiocut) yaklaşımının öncelikle bir grafı iki kümeye bölme problemi üzerinde inceleyelim:

4.1.1 $k = 2$ durumu:

Teorem 4.1.1. *u_A fonksiyonu;*

$$(u_A)_i = \begin{cases} \sqrt{\dfrac{|\overline{A}|}{|A|}}, v_i \in A \\ -\sqrt{\dfrac{|A|}{|\overline{A}|}}, v_i \in \overline{A} \end{cases}$$

biçiminde tanımlanmak üzere;

$$u_A^\top L u_A = \frac{1}{n} Ratiocut(A)$$

dır.

Kanıt. $v^\top Lv = \dfrac{1}{2}\sum_{i,j=1}^{n} w_{ij}(v_i - v_j)^2$ eşitliği 3.2 bölümünde ispatlanmıştı.

Buradan $u_A^\top L u_A = \dfrac{1}{2}\sum_{i,j=1}^{n} w_{ij}(u_A(i) - u_A(j))^2$ olur.

u_A'nin tanımından $i,j \in A$ veya $i,j \in \overline{A}$ ise $u_A(i) - u_A(j) = 0$.

$u_A^\top L u_A$'yi aşağıdaki gibi yazılabilir:

$$u_A^\top L u_A = \frac{1}{2}\sum_{i\in A, j\in\overline{A}} w_{ij}(u_A(i) - u_A(j))^2 + \frac{1}{2}\sum_{i\in\overline{A}, j\in A} w_{ij}(u_A(i) - u_A(j))^2$$

yukarıdaki eşitlikten

$$\begin{aligned}
&= \frac{1}{2}\left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 \cdot cut(A) + \frac{1}{2}\left(\sqrt{\frac{|A|}{|\overline{A}|}} + \sqrt{\frac{|\overline{A}|}{|A|}}\right)^2 \cdot cut(\overline{A}) \\
&= cut(A)\cdot\left(\sqrt{\frac{|\overline{A}|}{|A|}} + \sqrt{\frac{|A|}{|\overline{A}|}}\right)^2 \qquad [cut(A) = cut(\overline{A})] \\
&= cut(A)\cdot\left(\sqrt{\frac{|\overline{A}|}{|A|}}^2 + 2\cdot\sqrt{\frac{|\overline{A}|}{|A|}}\cdot\sqrt{\frac{|A|}{|\overline{A}|}} + \sqrt{\frac{|A|}{|\overline{A}|}}^2\right) \\
&= cut(A)\cdot\left(\frac{n-|A|}{|A|} + \frac{|A|+n-|A|}{|\overline{A}|} + 1\right) \\
&= cut(A)\cdot\left(\frac{n}{|A|} - 1 + \frac{n}{|\overline{A}|} + 1\right) \\
&= n\cdot cut(A)\cdot\left(\frac{1}{|A|} - \frac{1}{|\overline{A}|}\right) \\
&= n\cdot RatioCut(A) \quad \square
\end{aligned}$$

elde edilir.

Teorem 4.1.2. $u_A \cdot \mathbb{1}_n = 0$ *eşitliği sağlanır.*

Kanıt. (..) ifadesi yarine konursa;

$$\begin{aligned} u_A \cdot \mathbb{1}_n &= |A|\sqrt{\frac{|\overline{A}|}{|A|}} - |\overline{A}|\sqrt{\frac{|A|}{\overline{A}}} \\ &= \sqrt{\frac{|\overline{A}| \cdot |A|^2}{|A|}} - \sqrt{\frac{|A| \cdot \overline{A}^2}{|\overline{A}|}} \\ &= \sqrt{|\overline{A}| \cdot |A|} - \sqrt{|\overline{A}| \cdot |A|} \\ &= 0 \end{aligned}$$ □

elde edilir.

O halde kümelemeyi bir optimizasyon problemi olarak düşünürsek, $u_A \cdot \mathbb{1}_V$ koşulu altında ratiocut değerini minimum yapan, dolayısıyla $\frac{1}{n} u_A^\top L u_A$ minimum yapan A kümesini bulmak istenmektedir. Yani

$$\begin{aligned} \underset{A \subseteq V}{\text{argmin}} RatioCut(A) &= \underset{A \subseteq V}{\text{argmin}} \frac{1}{n} u_A^\top L u_A \\ &= \underset{A \subseteq V}{\text{argmin}} u_A^\top L u_A \end{aligned}$$

Ancak bu NP-zor bir problemdir. Bu nedenle problem basitleştirilerek $u \cdot \mathbb{1}_V = 0$ koşulu altında $\underset{A \subseteq V}{\text{argmin}} u^\top L u$ problemi ele alınacaktır.

Aşağıdaki verilen Rayleigh-Ritz teoremine göre bu probleme minimum değerini veren özvektör, G'nin Laplasyen matrisinin en küçük ikinci özdeğeri olan "Fiedler özdeğerine" karşılık gelen özvektördür.

Spektral kümelemede, Fiedler vektörü, bir grafiği iki kümeye bölmek için kullanılan özel bir vektördür. Adını, kavramı 1973 tarihli makalesinde tanıtan Miroslav Fiedler'den almıştır. Fiedler vektörü, grafiğin Laplace matrisinin ikinci en küçük özdeğerine karşılık gelen özvektör olarak tanımlanır. Laplasyen matrisi, köşeler arasındaki bağlantılar hakkındaki bilgileri kodlayan grafiğin bir matris temsilidir. Fiedler vektörü, her tepe noktasını Fiedler vektörünün pozitif veya negatif değerine karşılık gelen kümeye atayarak grafiği iki kümeye bölmek için kullanılmaktadır. Çünkü aynı işaretli değerler için $(v_i - v_j)^2$ daha küçük değerler alacaktır.

Teorem 4.1.3. ***(Spektral Teorem - simetrik matrisler için)***
$M : \mathbb{R}^n \to \mathbb{R}^n$ bir simetrik matris, ise
i.M'nin $v_1, \ldots, v_n$ özvektörleri vardır, öyle ki $\{v_1, \ldots v_n\}$ $\mathbb{R}^n$ için bir ortonormal bir

bazdır.[12]

ii.M'nin $\lambda_1, \ldots, \lambda_n$ reel özdeğerleri vardır.

Teorem 4.1.4. ***(Rayleigh - Ritz)***

M reel ve simetrik matris olsun. M'nin özvektörleri $v_1, v_2 \ldots, v_n$ ve $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ ise R_M fonksiyonu v_1'de λ_1 maksimum değerini alır. Ayrıca, R_M fonksiyonu v_n'de minimum değerini alır.

Kanıt. $u \in \mathbb{R}^n$ ve $||u||^2 = 1$ olsun. M simetrik ve reel bir matris olmak üzere, yukarıda verilen Spektral Teorem'den, öyle $c_1, c_2, \ldots, c_n \in \mathbb{R}$ vardır öyle ki $u = c_1v_1 + c_2v_2 + \ldots + c_nv_n$ sağlanır.

$(c_1^2 + c_2^2 + c_n^2 = 1)$

$R_M(u)$'yu aşağıdaki gibi yazılabilir:

$$\begin{aligned} R_M(u) = u^\top Mu &= (c_1v_1^\top + \ldots + c_nv_n^\top)M(c_1v_1 + \ldots + c_nv_n) \\ &= (c_1v_1^\top + \ldots + c_nv_n^\top)(\lambda_1c_1v_1 + \ldots + \lambda_nc_nv_n) \end{aligned}$$

Yine Spektral Teorem'den, $\forall i \neq j$, $v_i \cdot v_j = 0$ ve $\forall i, v_i \cdot v_i = 1$ olduğundan:

$$R_M(u) = \lambda_1c_1^2 + \ldots + \lambda_nc_n^2$$

olur. Bundan dolayı, R_M'yi maksimize etmek için $c_1 = 1$ ve $c_2 = c_3 = \ldots = c_n = 0$ seçtiğimizde $u = v_1$'de λ_1 değerini alır

ve R_M'yi minimize etmek için $c_n = 1$ ve $c_1 = c_2 = \ldots = c_{n-1} = 0$ seçtiğimizde $u = v_n$'de λ_n değerini alır. □

4.1.2 Keyfi k için:

$u_j = (u_{1,j}, \ldots, u_{n,j})^\top$'yi tanımlayalım:

$$u_{i,j} = \begin{cases} \dfrac{1}{\sqrt{|A_j|}}, \text{eğer } v_i \in A_j \\ 0, \qquad \text{aksi halde} \end{cases} \qquad (i = 1, \ldots, n; j = 1, \ldots, k)$$

Teorem 4.1.5. *$U = [u_1, u_2, \ldots, u_k] \in \mathbb{R}^{n \times k}$ matris olsun.U'nin sütunları birbirine ortonormal olduğundan $U^\top U = I$ olur.*

Kanıt. Her $i \in \{1, 2, \ldots, k\}$ için U'nun sütunları birbirine ortonormal olduğundan $\|u_i\|^2 = 1$ ve her $i \neq j$ için $u_i \cdot u_j = 0$ olur. Yani $U^\top U$ matrisi için diyagonalde 1, diyagonal haricinde ise 0 olması gerekir. Buradan $U^\top U = I$ olur. □

Teorem 4.1.6. $u_j^\top Lu_j = \dfrac{cut(A_j, \overline{A_j})}{|A_j|}$

Kanıt. Her $v \in \mathbb{R}^n$ için $v^\top Lv = \sum_{i,j} w_{ij}(v_i - v_j)^2$ eşitliği bölüm ... da ispatlanmıştı. $u_{i,j}$'nin tanımından $s,t \in A_j$ veya $s,t \in \overline{A_j}$ ise $u_{s,j} - u_{t,j} = 0$ olur.

$$\begin{aligned}
u_j^\top L u_j &= \frac{1}{2}\sum_{s,t} w_{st}(u_{s,j} - u_{t,j})^2 \\
&= \frac{1}{2}\Big(\sum_{v_s \in A_j, v_t \in \overline{A_j}} w_{st}(u_{s,j} - ut, j)^2 - \sum_{v_s \in \overline{A_j}, v_t \in A_j} w_{st}(u_{s,j} - ut, j)^2\Big) \\
&= \frac{1}{2}\Big(\sum_{v_s \in A_j, v_t \in \overline{A_j}} w_{st}\Big(\frac{1}{\sqrt{|A_j|}}\Big)^2 - \sum_{v_s \in \overline{A_j}, v_t \in A_j} w_{st}\Big(-\frac{1}{\sqrt{|A_j|}}\Big)^2\Big) \\
&= \frac{1}{2}\Big(\sum_{v_s \in A_j, v_t \in \overline{A_j}} w_{st}\Big(\frac{1}{|A_j|}\Big) - \sum_{v_s \in \overline{A_j}, v_t \in A_j} w_{st}\Big(\frac{1}{|A_j|}\Big)\Big) \\
&= \frac{1}{2}\frac{1}{|A_j|}\Big(\sum_{v_s \in A_j, v_t \in \overline{A_j}} w_{st} - \sum_{v_s \in \overline{A_j}, v_t \in A_j} w_{st}\Big) \\
&= \frac{1}{2}\frac{1}{|A_j|}\Big(W(A_j, \overline{A_j}) + W(\overline{A_j}, A_j)\Big) \\
&= \frac{1}{2}\frac{1}{|A_j|} 2 \cdot W(A_j, \overline{A_j}) \\
&= \frac{cut(A_j, \overline{A_j})}{|A_j|}
\end{aligned}$$

□

Teorem 4.1.7. $u_j^\top L u_j = (U^\top LU)_{jj}$

Kanıt. $U^\top LU$ matrisinin jj. elemanı $U^\top L$ nin j. satırı ile U matrisinin j. sütunu olan u_i nin çarpılması ile elde edilir. $U^\top L$ nin j. satırı ise $U^\top$ nin j. satırının yani $u_j^\top$ nin L ile çarpılması ile elde edilir. O halde $(U^\top LU)_{jj} = u_j^\top L u_j$ □

Sonuç 4.1.1. $RatioCut(A_1, \ldots, A_k) = Tr(U^\top LU)$

Kanıt. $RatioCut(A_1, \ldots, A_k) = \sum_{i=1}^k u_i^\top L u_i = \sum_{i=1}^k (U^\top LU)_{ii} = Tr(U^\top LU)$ □

Yani RatioCut'ı minimize etmekle $Tr(U^\top LU)$'yi minimize etmek aynı şeydir. Bu problemi aşağıdaki şekilde yazılabilir:
$U^\top U = I$ koşulu altında $\min_{B_1,\ldots,B_k} Tr(U^\top LU)$
Yukarıdakine benzer şekilde, U matrisinin girdilerinin keyfi değerler almasına izin verilerek problem basitleştirilirse:
$U^\top U = I$ koşulu altında $\min_{U \in \mathbb{R}^{n \times k}} Tr(U^\top LU)$ elde edilir. Bu durumda yine Rayleigh-Ritz Teoremin'den U'nin sütunlarını L'nin ilk k özvektörü olarak seçilir.

4.2 Normalize edilmiş Kesim (NCut) Yaklaşımı:

Bu bölümde [7] nolu kaynaktan yararlanılarak oransal kesim yerine normalize edilmiş kesim temelli yaklaşım incelenmiştir.

4.2.1 $k = 2$ durumu:

Bir h_A fonksiyonu tanımlayalım:

$$(h_A)_i = \begin{cases} \sqrt{\dfrac{vol(\overline{A})}{vol(A)}}, & v_i \in A \text{ ise} \\ \sqrt{\dfrac{vol(A)}{vol(\overline{A})}}, & v_i \in \overline{A} \text{ ise} \end{cases}$$

Teorem 4.2.1. *h_A yı kısaca h ile göstermek üzere; $(Dh)^\top \mathbb{1} = 0$ dir.*

Kanıt.

$$\begin{aligned}
(Dh)^\top \mathbb{1} &= h^\top D\mathbb{1} \\
&= (h_1, \ldots, h_n) \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \\
&= h_1 d_1 + \ldots + h_n d_n \\
&= \sum_{v_i \in A} h_i d_i + \sum_{v_i \in \overline{A}} h_i d_i \\
&= \sqrt{\frac{vol(\overline{A})}{vol(A)}} \sum_{v_i \in A} d_i - \sqrt{\frac{vol(A)}{vol(\overline{A})}} \sum_{v_i \in \overline{A}} d_i \\
&= \sqrt{\frac{vol(\overline{A})}{vol(A)}} \cdot vol(A) - \sqrt{\frac{vol(A)}{vol(\overline{A})}} \cdot vol(\overline{A}) \\
&= \sqrt{vol(A)vol(\overline{A})} - \sqrt{vol(A)vol(\overline{A})} \\
&= 0
\end{aligned}$$

□

Teorem 4.2.2. $h^\top Dh = vol(V)$

Kanıt.

$$h^\top Dh = (h_1, \ldots, h_n) \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_n \end{pmatrix}$$

$$\begin{aligned} &= h_1^2 d_1 + \ldots + h_n^2 d_n \\ &= \sum_{v_i \in A} h_i^2 d_i + \sum_{v_i \in \overline{A}} h_i^2 d_i \\ &= \frac{vol(\overline{A})}{vol(A)} \sum_{v_i \in A} d_i + \frac{vol(A)}{vol(\overline{A})} \sum_{v_i \in \overline{A}} d_i \\ &= \frac{vol(\overline{A})}{vol(A)} \cdot vol(A) + \frac{vol(A)}{vol(\overline{A})} \cdot vol(\overline{A}) \\ &= vol(\overline{A}) + vol(A) \\ &= vol(V) \end{aligned}$$

□

Teorem 4.2.3. $\dfrac{h^\top Lh}{h^\top Dh} = NCut(A, \overline{A})$

Kanıt. $h^\top Lh = \dfrac{1}{2} \sum_{i,j=1} w_{ij}(h_i - h_j)^2$ eşitliğini ispatladık. h'in tanımından $i, j \in A$ veya $i, j \in \overline{A}$ ise $h_i - h_j = 0$ olur.

$$\begin{aligned} h^\top Lh &= \frac{1}{2} \sum_{i,j=1} w_{ij}(h_i - h_j)^2 \\ &= \frac{1}{2} \Big(\sum_{i \in A, j \in \overline{A}} w_{ij}(h_i - h_j)^2 + \sum_{i \in \overline{A}, j \in A} w_{ij}(h_i - h_j)^2 \Big) \\ &= \frac{1}{2} \Big[\sum_{i \in A, j \in \overline{A}} w_{ij} \Big(\sqrt{\frac{vol(\overline{A})}{vol(A)}} - \Big(-\sqrt{\frac{vol(A)}{vol(\overline{A})}} \Big) \Big)^2 + \sum_{i \in \overline{A}, j \in A} w_{ij} \Big(-\sqrt{\frac{vol(A)}{vol(\overline{A})}} - \sqrt{\frac{vol(\overline{A})}{vol(A)}} \Big)^2 \Big] \\ &\Big(\sqrt{\frac{vol(A)}{vol(\overline{A})}} + \sqrt{\frac{vol(\overline{A})}{vol(A)}} = c \quad \text{diyelim} \Big) \\ &= \frac{1}{2} \Big(c \cdot \sum_{i \in A, j \in \overline{A}} w_{ij} + c \cdot \sum_{i \in \overline{A}, j \in A} w_{ij} \Big) \\ &\Big(W(A, B) = \sum_{i \in A, j \in B} w_{ij} \quad \text{tanımından ve } W(A, \overline{A}) = W(\overline{A}, A) \text{'dan} \Big) \\ &= \frac{1}{2} \Big(c \cdot W(A, \overline{A}) + c \cdot W(\overline{A}, A) \Big) \\ &= c \cdot W(A, \overline{A}) \end{aligned}$$

$$\Big(vol(V) = vol(A) + vol(\overline{A}) \quad \text{eşitliğinden}\Big)$$
$$= \Big(\frac{vol(V) - vol(\overline{A})}{vol(\overline{A})} + \frac{vol(V) - vol(A)}{vol(A) + 2}\Big) \cdot W(A, \overline{A})$$
$$vol(V)\Big(\frac{1}{vol(A)} + \frac{1}{vol(\overline{A})}\Big) \cdot W(A, \overline{A})$$

$h^\top Dh = vol(V)$ gösterdik.

$$\frac{h^\top Lh}{h^\top Dh} = \frac{vol(V)\Big(\frac{1}{vol(A)} + \frac{1}{vol(\overline{A})}\Big) \cdot W(A, \overline{A})}{vol(V)}$$
$$= W(A, \overline{A}) \cdot \Big(\frac{1}{vol(A)} + \frac{1}{vol(\overline{A})}\Big)$$
$$\Big(NCut(A, \overline{A}) \quad \text{tanımından}\Big)$$
$$= NCut(A, \overline{A}) \qquad \square$$

$Ncut$ minimize etme problemini tekrar yazalım:

$Dh \perp \mathbb{1}, h^\top Dh = vol(V)$ koşulu altında $\min_A h^\top Lh$

h'in keyfi reel değer alabilsin. Yine benzer şekilde bu problem yerine aşağıdaki daha basit bir problemle ilgilenilecektir:

$Dh \perp \mathbb{1}, h^\top Dh = vol(V)$ koşulu altında $\min_{h \in \mathbb{R}^n} h^\top Lh$

$h : D^{1/2}g$ yazılırsa:

$g \perp D^{1/2}\mathbb{1}, \|g\|^2 = vol(V)$ koşulu altında $\min_{g \in \mathbb{R}^n} g^\top D^{-1/2} L D^{-1/2} g$

$D^{-1/2}LD^{-1/2} = L_{sym}$, ,$L_{sym}$'nin ilk özvektörü $D^{1/2}\mathbb{1}$'dir ve $vol(V)$ sabittir.

Rayleigh-Ritz Teoreminden g,L_{sym}'nin ikinci özvektörüdür.

$h = D^{-1/2}g$ yazalım ve L_{sym}-L_{rw} özelliklerinden h,L_{rw}'nin ikinci özvektörüdür.

4.2.2 Keyfi k durumu:

$h_j = (h_{1,j}, \ldots, h_{n,j})^\top$'yi tanımlayalım.

$$h_{i,j} = \begin{cases} \dfrac{1}{\sqrt{vol(A_j)}}, & v_i \in A_j \text{ise} \\ 0, & \text{aksi halde} \end{cases} \qquad (i = 1, \ldots, n; j = 1, \ldots, k)$$

Teorem 4.2.4. *$H = [h_1, h_2, \ldots, h_n] \in \mathbb{R}^{n\times k}$ matris olsun. H'nin sütunları birbirine ortonormal olduğundan $H^\top H = I$ olur.*

Kanıt. Her $i \in \{1, 2, \ldots, k\}$ için H'nin sütunları birbirine ortonormal olduğundan $\|h_i\|^2 = 1$ ve her $i \neq j$ için $h_i \cdot h_j = 0$ olur. Yani $H^\top H$ matrisi için diyagonalde 1, diyagonal haricinde ise 0 olması gerekir. Buradan $H^\top H = I$ olur. □

Teorem 4.2.5. $h_j^\top D h_j = 1$

Kanıt.

$$\begin{aligned}
h_j^\top D h_j &= (h_{1,j}, \ldots, h_{n,j}) \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{pmatrix} h_{1,j} \\ \vdots \\ h_{n,j} \end{pmatrix} \\
&= h_{1,j}^2 d_1 + \ldots + h_{n,j}^2 d_n \\
&= \sum_{v_\ell \in A_j} h_{\ell,j}^2 d_\ell + \sum_{v_\ell \in \overline{A_j}} h_{\ell,j}^2 d_\ell \\
&= \frac{1}{vol(A_j)} \sum_{v_\ell \in A_j} d_j \\
&= \frac{1}{vol(A_j)} vol(A_j) \\
&= 1
\end{aligned}$$

□

Teorem 4.2.6. $h_j^\top L h_j = \dfrac{cut(A_j, \overline{A_j})}{vol(A_j)}$

Kanıt.

$$\begin{aligned}
h_j^\top L h_j &= \frac{1}{2} \sum_{s,t} w_{st} (h_{s,t} - h_{t,j})^2 \\
&= \frac{1}{2} \Big(\sum_{v_s \in A_j, v_t \in \overline{A_j}} w_{st} (h_{s,t} - h_{t,j})^2 - \sum_{v_s \in \overline{A_j}, v_t \in A_j} w_{st} (h_{s,t} - h_{t,j})^2 \Big) \\
&= \frac{1}{2} \Big(\sum_{v_s \in A_j, v_t \in \overline{A_j}} w_{st} \Big(\frac{1}{\sqrt{vol(A_j)}} - 0 \Big)^2 - \sum_{v_s \in \overline{A_j}, v_t \in A_j} w_{st} \Big(0 - \frac{1}{\sqrt{vol(A_j)}} \Big)^2 \Big) \\
&= \frac{1}{2} \Big(\sum_{v_s \in A_j, v_t \in \overline{A_j}} w_{st} \Big(\frac{1}{vol(A_j)} \Big) - \sum_{v_s \in \overline{A_j}, v_t \in A_j} w_{st} \Big(\frac{1}{vol(A_j)} \Big) \Big) \\
&= \frac{1}{2} \frac{1}{vol(A_j)} \Big(\sum_{v_s \in A_j, v_t \in \overline{A_j}} w_{st} - \sum_{v_s \in \overline{A_j}, v_t \in A_j} w_{st} \Big)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\frac{1}{vol(A_j)}\Big(W(A_j, \overline{A_j}) + W(\overline{A_j}, A_j)\Big) \\
&= \frac{1}{2}\frac{1}{vol(A_j)} 2 \cdot W(A_j, \overline{A_j}) \\
&= \frac{cut(A_j, \overline{A_j})}{vol(A_j)}
\end{aligned}
$$

Böylece normalize edilmiş kesim yaklaşımı için de oransal kesim yaklaşımındakine benzer sonuca ulaşılmış olur. □

Bölüm 5

SPEKTRAL KÜMELEME ALGORİTMALARI

Bu bölümde ise önceki bölümlerde açıklanan teorik alt yapı kullanılarak elde edilmiş olan iki spektral kümeleme algoritmasından bahsedilecektir. Veri kümesi n noktadan oluştuğunu varsayalım. $s_{ij} = s(x_i, x_j)$ negatif olmayan ve simetrik olan benzerlik fonksiyonuyla ikili benzerlikleri ölçüyoruz. $S = (s_{ij})_{i,j=1,\ldots,n}$ benzerlik matrisi olsun.

5.1 Normalize Edilmemiş Spektral Kümeleme Algoritması

Normalize edilmemiş spektral kümeleme algoritmasının adımları aşağıda verilmiştir [7]:
Girdi: $S \in \mathbb{R}^{n\times n}$ benzerlik matrisi, k küme sayısı
1. Benzerlik grafını oluştur. W ağırlıklı yakınlık matrisi olsun.
2. $L = D - W$'yi hesapla.
3. L'nin (ilk en küçük k özdeğere karşılık gelen) ilk k özvektörünü hesapla. $u_1, \ldots, u_k$ ilk k özvektör olsun.
4. $U = [u_1, u_2, \ldots, u_n] \in \mathbb{R}^{n\times k}$ matrisini oluştur.
5. $i = 1, \ldots, n$ için $y_i \in \mathbb{R}^k$, U'nun i.satıra denk gelen vektör olsun.
6. k-means algoritmasıyla $(y_i)_{i=1,\ldots,n} \in \mathbb{R}^k$ noktalarını $C_1, \ldots, C_k$ kümelerine kümele.
Çıktı: $A_1, \ldots, A_k$ kümeleri öyle ki $A_i = \{j \mid y_j \in C_i\}$
Normalize edilmiş spekral kümeleme algoritmasının ise iki versiyonu vardır.

5.2 Shi ve Malik'in Normalize Edilmiş Spektral Kümeleme Algoritması

Girdi: $S \in \mathbb{R}^{n \times n}$ benzerlik matrisi, k küme sayısı

1. Benzerlik grafını oluştur. W ağırlıklı yakınlık matrisi olsun.

2. $L = D - W$'yi hesapla.

3. Genelleştirilmiş özdeğer probleminin (ilk en küçük k özdeğerine karşılık gelen) ilk k genelleştirilmiş özvektörünün bul.

(*Teorem 3.3.1 (iii)*' ye göre bu özvektörler normalleştirilmiş Laplacian L_{rw}'nin özvektörleridir, bu nedenle normalize edilmiş spektral kümeleme olarak adlandırılır.)

4. $U = [u_1, u_2, \ldots, u_n] \in \mathbb{R}^{n \times k}$ matrisini oluştur.

5. $i = 1, \ldots, n$ için $y_i \in \mathbb{R}^k$, U'nun i.satıra denk gelen vektör olsun.

6. k-means algoritmasıyla $(y_i)_{i=1,\ldots,n} \in \mathbb{R}^k$ noktalarını $C_1, \ldots, C_k$ kümelerine kümele.

Çıktı: $A_1, \ldots, A_k$ kümeleri öyle ki $A_i = \{j \mid y_j \in C_i\}$

Bu algoritma [5] nolu çalışmaya dayanmaktadır.

5.3 Ng, Jordan ve Weiss'in Normalize Edilmiş Spektral Kümeleme Algoritması

Girdi: $S \in \mathbb{R}^{n \times n}$ benzerlik matrisi, k küme sayısı

1. Benzerlik grafını oluştur.W ağırlıklı yakınlık matrisi olsun.

2. $L_{sym} = D^{-1/2} L D^{-1/2}$'yi hesapla.

3.L_{sym}'nin (ilk en küçük k özdeğerine karşılık gelen) ilk k özvektörünü hesapla. $u_1, \ldots, u_k$ ilk k özvektör olsun.

4. $U = [u_1, u_2, \ldots, u_n] \in \mathbb{R}^{n \times k}$ matrisini oluştur.

5.U'nun satırlarının normu 1 olacak şekilde normalize et. $t_{ij} = u_{ij} / \sum_k (u_{ik}^2)^{1/2}$ olsun. $T \in \mathbb{R}^{n \times k}$ matrisini oluştur.

6. $i = 1, \ldots, n$ için $y_i \in \mathbb{R}^k$, T'nin i.satıra denk gelen vektör olsun.

7. k-means algoritmasıyla $(y_i)_{i=1,\ldots,n} \in \mathbb{R}^k$ noktalarını $C_1, \ldots, C_k$ kümelerine kümele.

Çıktı: $A_1, \ldots, A_k$ kümeleri öyle ki $A_i = \{j \mid y_j \in C_i\}$

Bu algoritma ise [6] nolu çalışmaya dayanmaktadır.

Yukarıda belirtilen üç algoritma, üç farklı Laplasyen matris kullanmaları dışında benzer görünmektedir. Her üç algoritmanın da ortak noktası veri noktalarını farklı veri noktalarına dönüştürmesidir. Laplasyen matrisinin özelliklerinden dolayı bu dönüşüm faydalıdır. Özellikle, k-ortalamalar algoritması bu kümeleri rahat bir şekilde tespit eder.

Bölüm 6

FARKLI METRİK FONKSİYONLARININ SPEKTRAL KÜMELEMEYE ETKİSİ

Kümeleme algoritmaları, veri noktalarını benzer özelliklere sahip olan gruplara ayırmak için veri noktaları arasındaki uzaklıkları ölçen bir uzaklık fonksiyonu kullanılır, bu standart durumda Öklid uzaklığıdır. Bununla birlikte en sık kullanılan kümeleme algoritmalarından K-means kümeleme algoritmasında Öklid uzaklığı yerine farklı uzaklık fonksiyonları kullanılarak elde edilen sonuçların karşılaştırıldığı [8],[9] gibi çalışmalar mevcuttur. Bu tezde ise Spektral kümeleme algoritması farklı uzaklık fonksiyonları ile ele alınarak sonuçları değerlendirilmiştir.

Kümelemede farklı uzaklık fonksiyonlarının kullanılmasının birkaç nedeni vardır. Örneğin, Öklid uzaklığı tüm boyutlardaki farklılıklara duyarlıyken, Manhattan uzaklığı yalnızca aynı boyuttaki koordinatlar arasındaki farkı dikkate alır. Bu, boyutlar eşit derecede önemli olmadığında Manhattan mesafesinin daha uygun olabileceği anlamına gelir. Ayrıca, kümeleme algoritmasının performansını optimize etmek için farklı uzaklık fonksiyonları kullanılabilir. Bazı uzaklık fonksiyonları, hesaplama açısından daha verimli olabilirken, diğerleri verilerdeki gürültüye veya aykırı değerlere karşı daha dayanıklı olabilir. Özetle, kümelemede, verilerin doğasına, verilerin ilgilenilen yönlerine ve kümeleme algoritmasının istenen performansına bağlı olarak farklı uzaklık fonksiyonları kullanılabilir. Bu bölümde K-means algoritmasının başarılı şekilde ayıramadığı bazı veri kümeleri ele alınmış ve spektral kümeleme algoritmasında Öklid uzaklığının yanı sıra farklı uzaklık fonksiyonları da kullanarak daha iyi bir kümeleme yapılıp yapılmayacağı incelenmiştir.

6.1 Uzaklık Fonksiyonları

Bu bölümde Spektral kümeleme algoritmasnda kullanılacak olan farklı uzaklık fonksiyonları tanıtılacaktır: Öklid Uzaklığı

$x, y \in \mathbb{R}^n$ için Öklid uzaklığı

$$d_2(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Manhattan Uzaklığı (Cityblock)

$x, y \in \mathbb{R}^n$ için Manhattan uzaklığı

$$d_1(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Chebyshev Uzaklığı

$x, y \in \mathbb{R}^n$ için Minkowski uzaklığı

$$d_\infty(x, y) = \max_i |x_i - y_i|$$

Canberra Uzaklığı

$x, y \in \mathbb{R}^n$ için Canberra uzaklığı

$$d_{can}(x, y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

Cosine Uzaklığı

$x, y \in \mathbb{R}^n$ için Cosine uzaklığı

$$d_{cos}(x, y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$$

Bray-Curtis Uzaklığı

$x, y \in \mathbb{R}^n$ için Bray-Curtis uzaklığı

$$d_{bray}(x, y) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} (x_i + y_i)}$$

6.2 Farklı Uzaklık Fonksiyonları İçin Elde Edilen Sonuçlar

Bu kısımda üç veri kümesi Bölüm 3'de açıklanan Normalize edilmemiş Laplasyen matris;

$$L = D - W$$

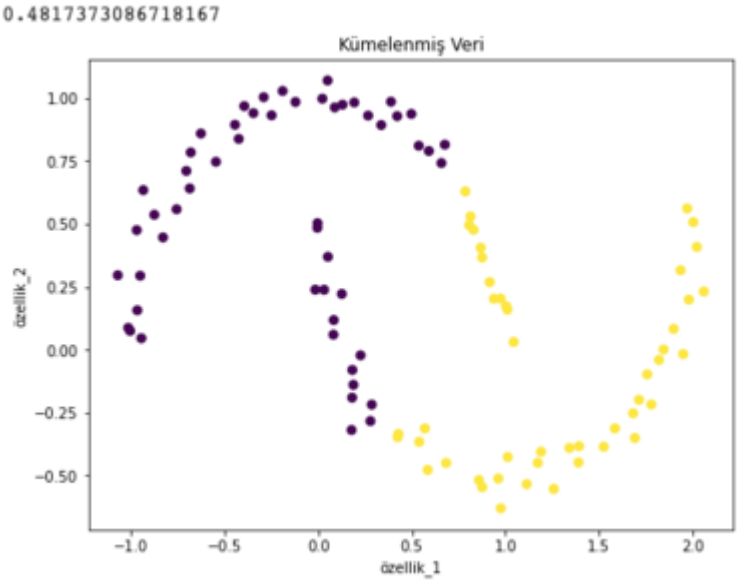
ve normalize edilmiş Laplasyen Matrisler;

$$L_{sym} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$
$$L_{rw} := D^{-1} L = I - D^{-1} W$$

kullanılarak spektral kümeleme algoritması ile kümelenirken 6 farklı uzaklık fonksiyonları kullanılmıştır. Kümeleme başarısını ölçmek için, kümelenen verilerin bulunduğu kümedeki uygunluğunu bulmak için geliştirilen ve temeli [10] makalesine dayanan "Silhoutte skoru" kullanılmıştır.

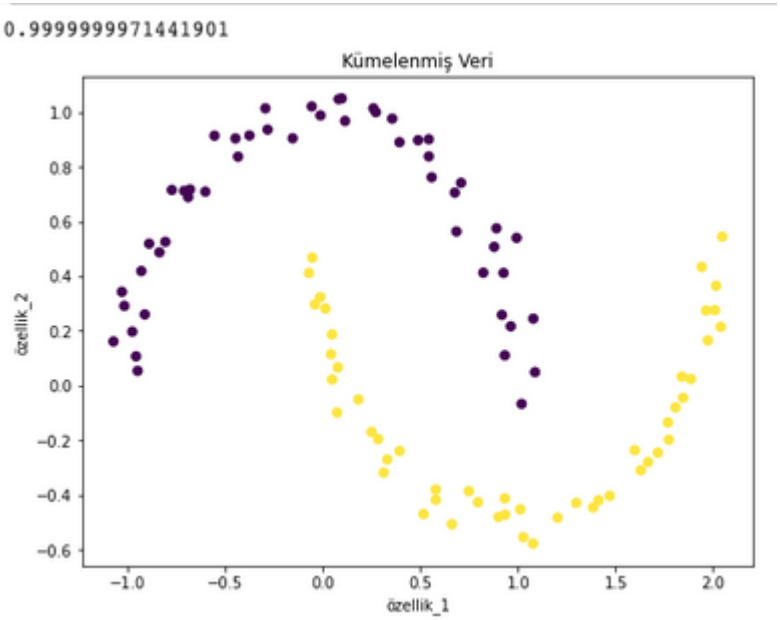
6.2.1 Noisy Moons veri kümesi:

Bu veri kümesi iç içe iki ay şeklinden oluşmaktadır. En sık kullanılan kümeleme algoritmalarından K-Means algoritması bu kümeyi aşağıdaki gibi kümelemektedir:



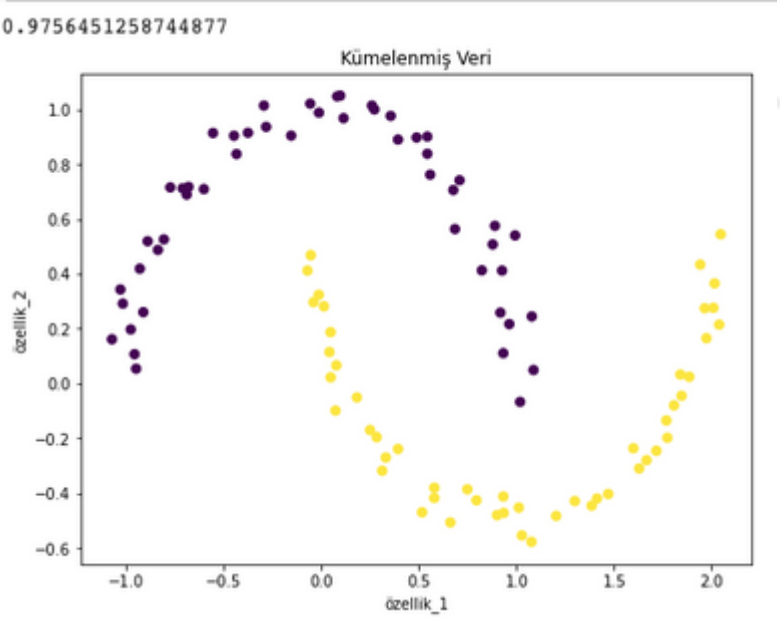
Şekil 6.1: k-means

Spektral kümeleme algoritması bu veri setini ayırmada daha başarılı olmakla birlikte farklı uzaklık fonksiyonlarının bu kümelemeye etkisi incelenmiş ve Normalize edilmemiş L laplasyen matrisi için aşağıdaki sonuçlara ulaşılmıştır:

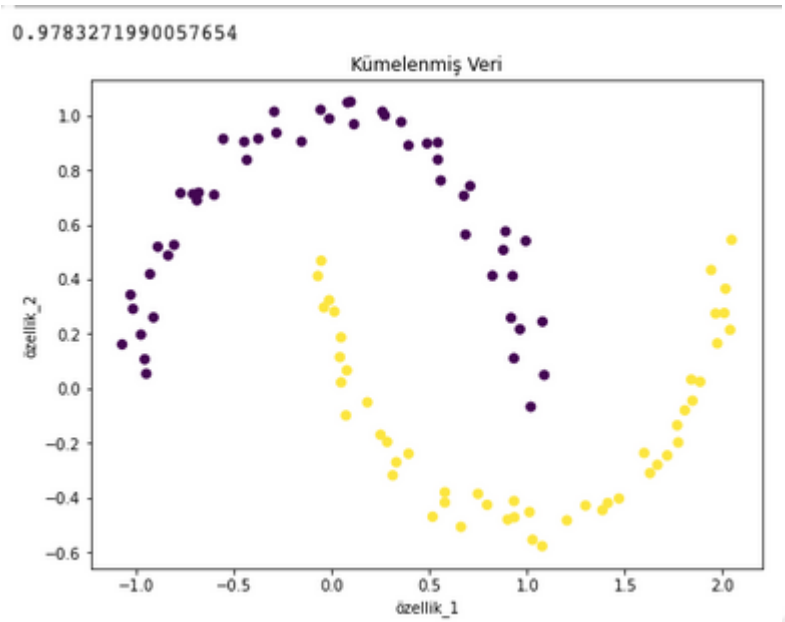


Şekil 6.2: Euclidean - L

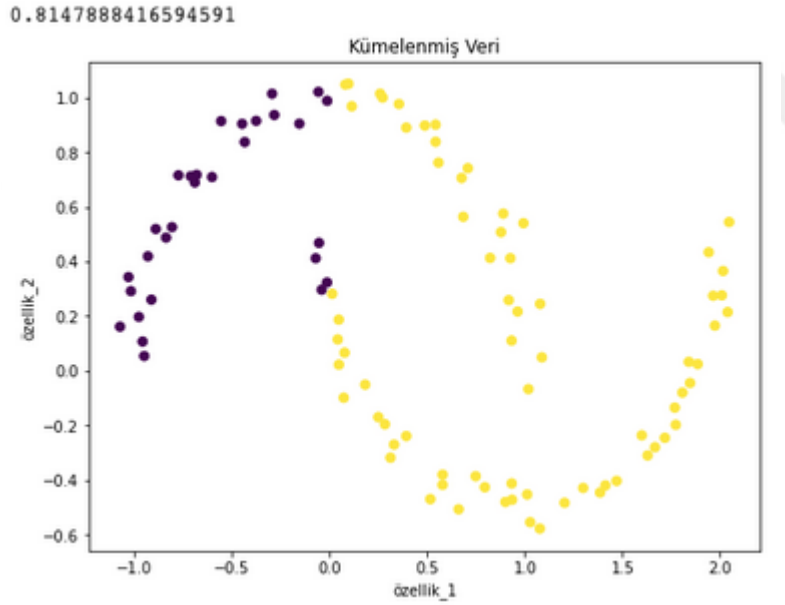
(Şeklin sol üst köşesinde yer alan sayılar kümelemenin başarısını ölçmeye yarayan araçlardan biri olan ”Silhoutte skoru”nu göstermektedir.)



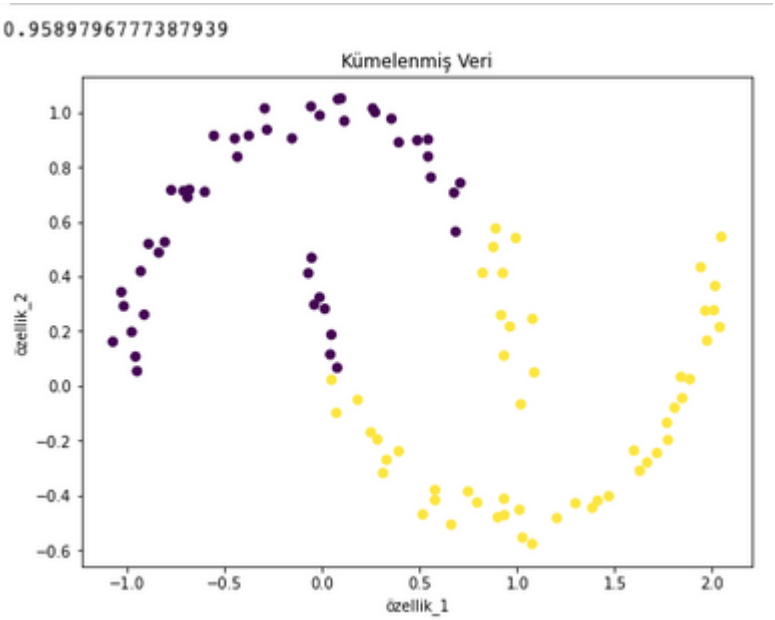
Şekil 6.3: Cityblock - L



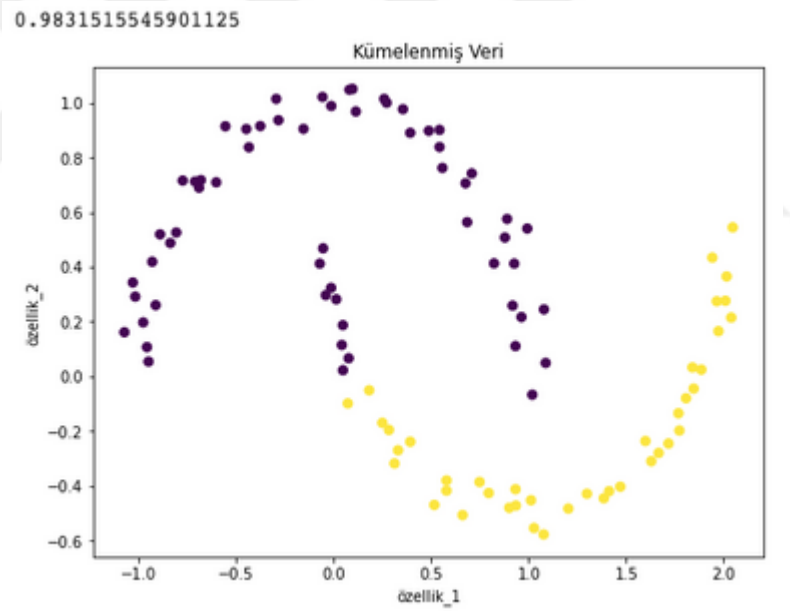
Şekil 6.4: Chebyshev - L



Şekil 6.5: Canberra - L



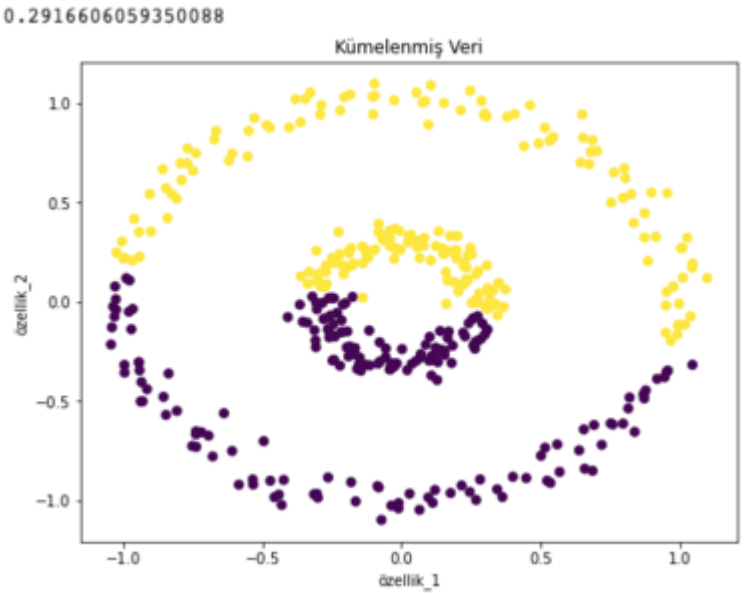
Şekil 6.6: Cosine - L



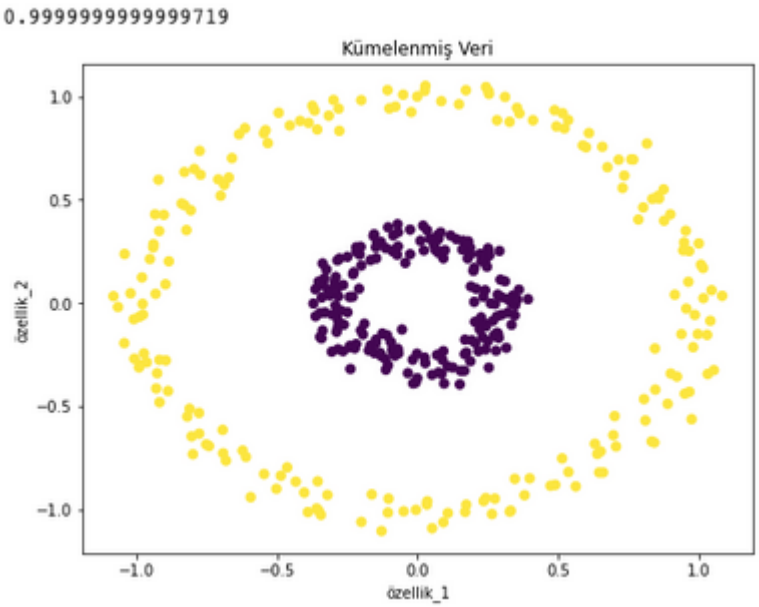
Şekil 6.7: Bray-Curtis - L

6.2.2 İç İçe Çember veri kümesi:

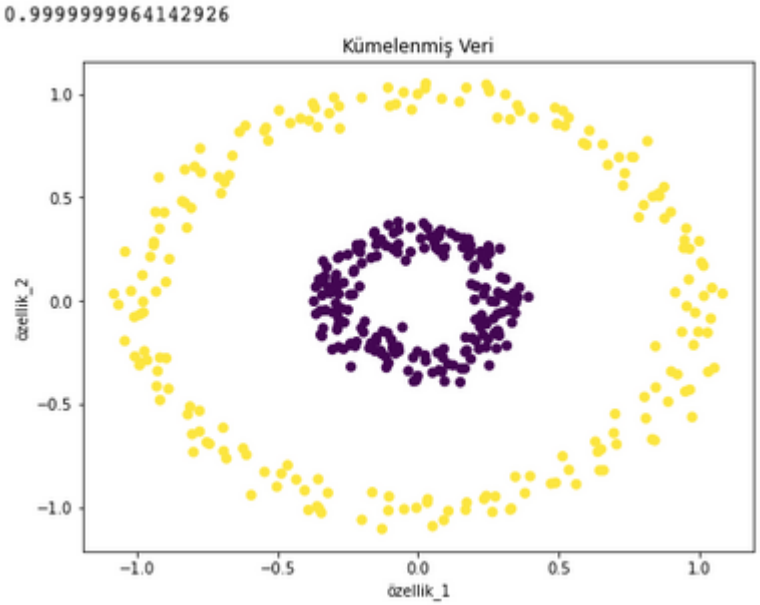
Bu veri kümesi iç içe iki çember şeklinden oluşmaktadır. Yukarıda görüldüğü gibi K-Means algoritması bu kümeyi de başarılı bir şekilde kümeleyememektedir:
Bu veri seti için Normalize edilmemiş L laplasyen matrisi kullanılarak elde edilen sonuçlar ise aşağıda verilmiştir:



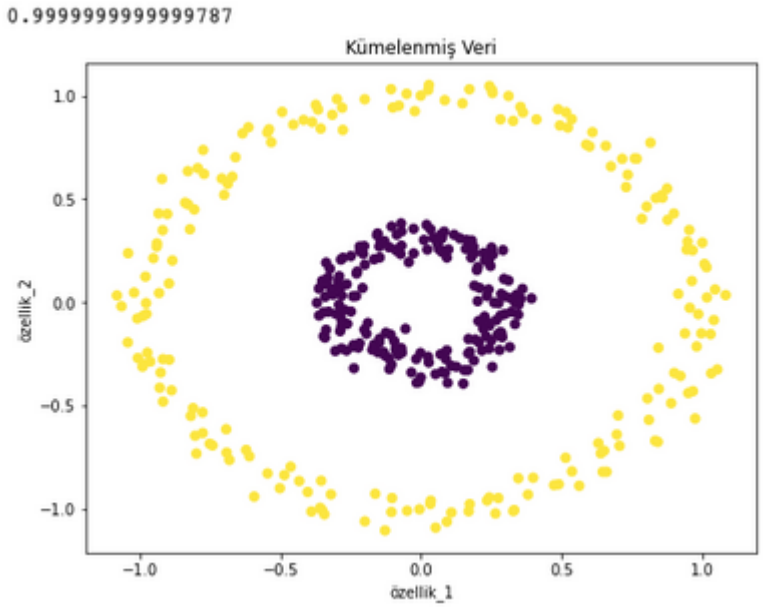
Şekil 6.8: k-means



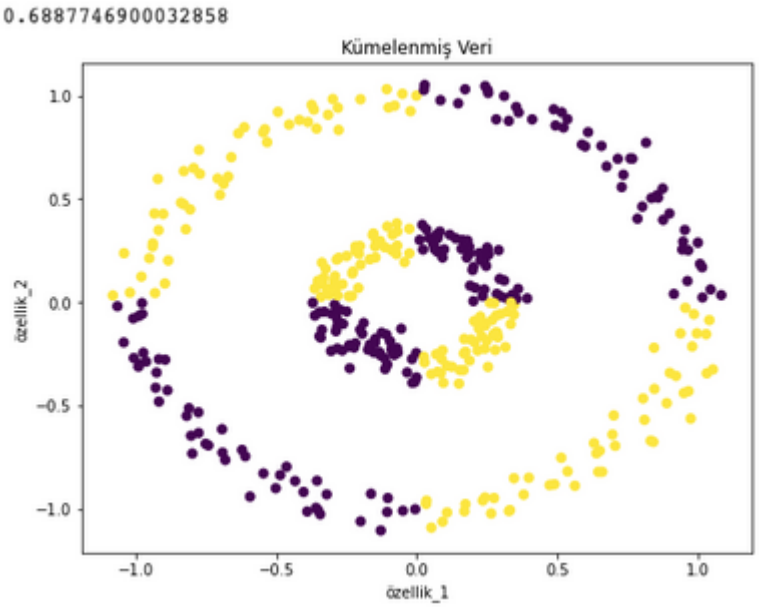
Şekil 6.10: Cityblock - L



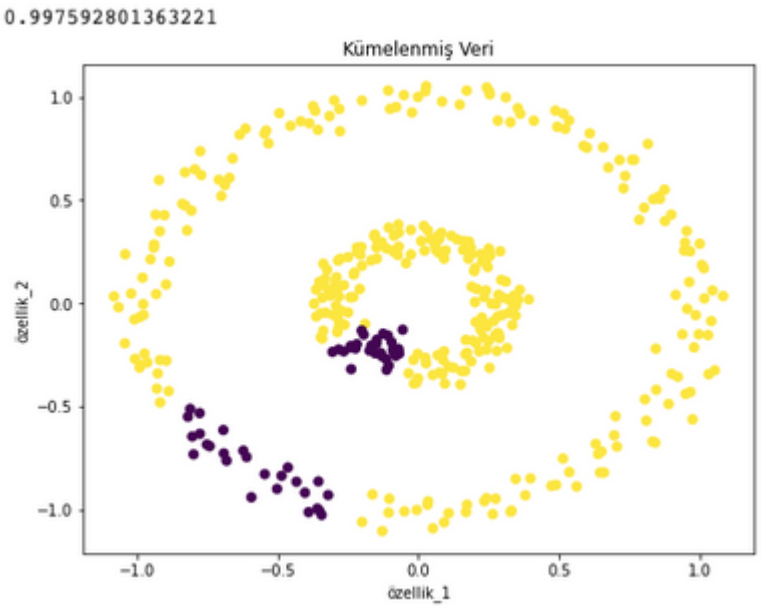
Şekil 6.9: Euclidean - L



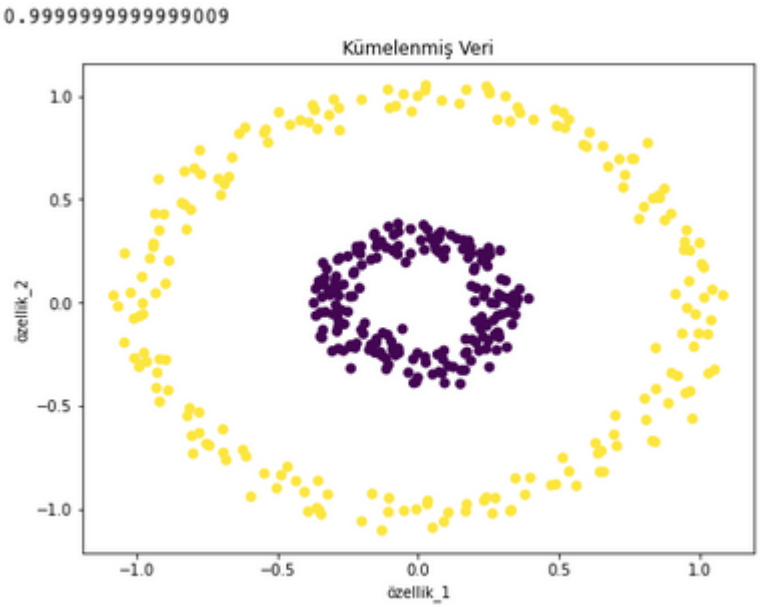
Şekil 6.11: Chebyshev - L



Şekil 6.12: Canberra - L



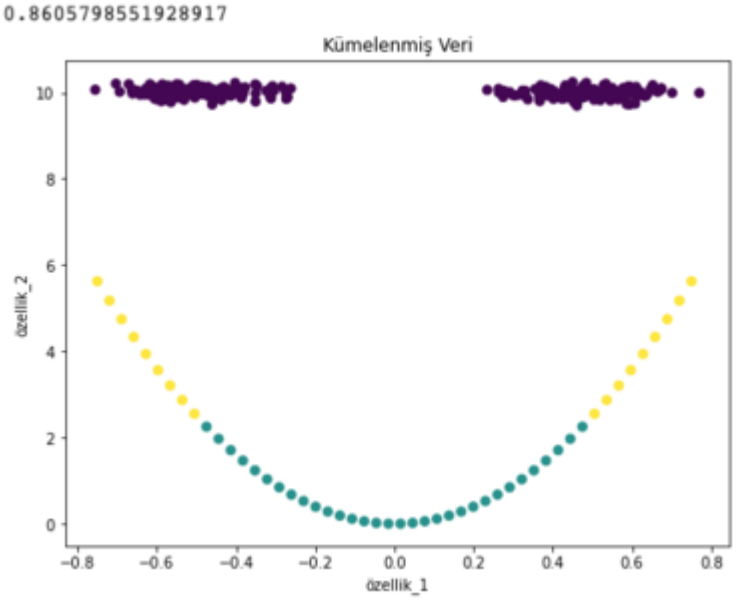
Şekil 6.13: Cosine - L



Şekil 6.14: Bray-Curtis - L

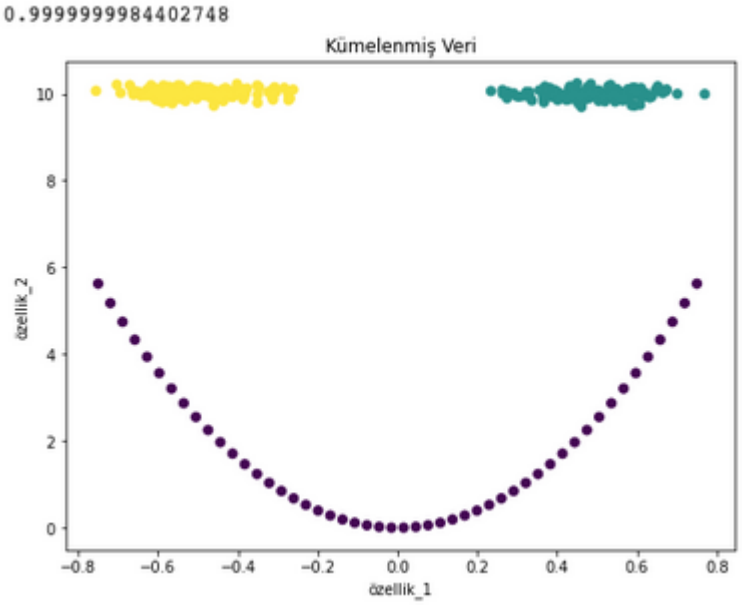
6.2.3 Gülen Yüz veri kümesi:

Bu veri kümesini ise K-Means algoritması bu kümeyi aşağıdak gibi kümelemektedir:

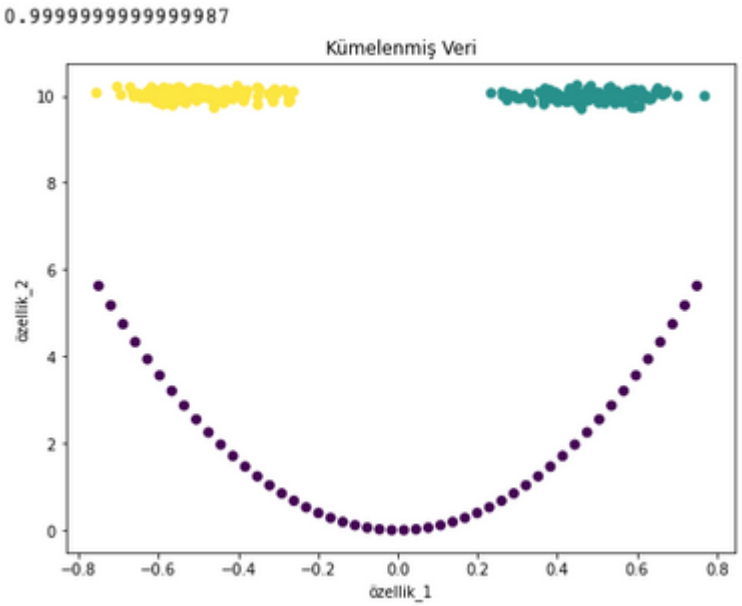


Şekil 6.15: k-means

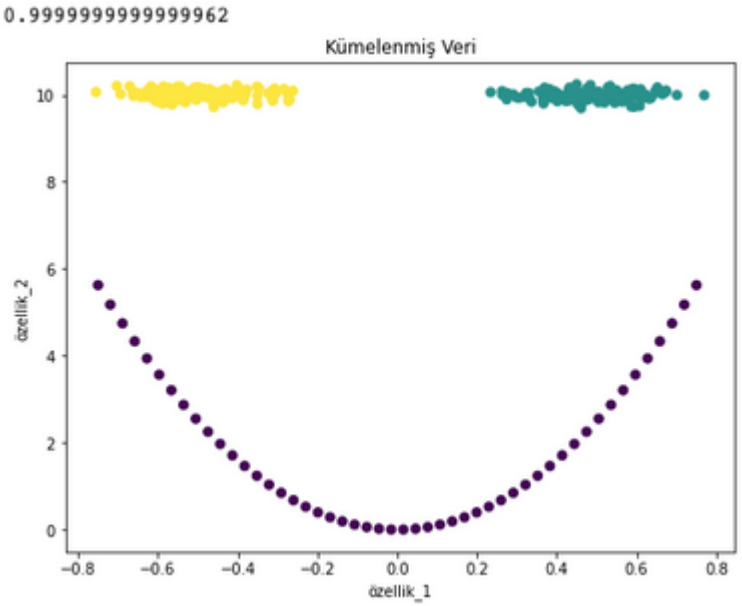
Bu veri kümesi üzerinde de 6 farklı uzaklık fonksiyonlarının bu kümelemeye etkisi incelenmiş, Normalize edilmemiş L laplasyen matrisi kullanılarak elde edilen sonuçlar aşağıda sunulmuştur:



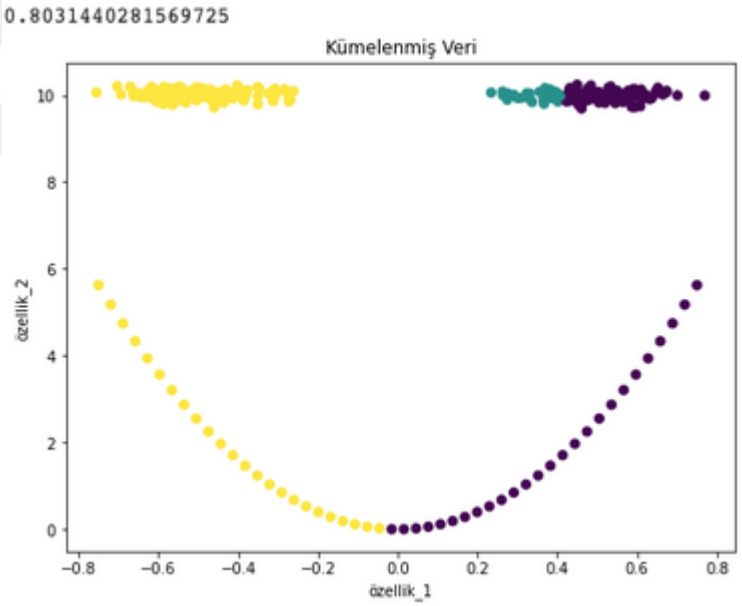
Şekil 6.16: Euclidean - L



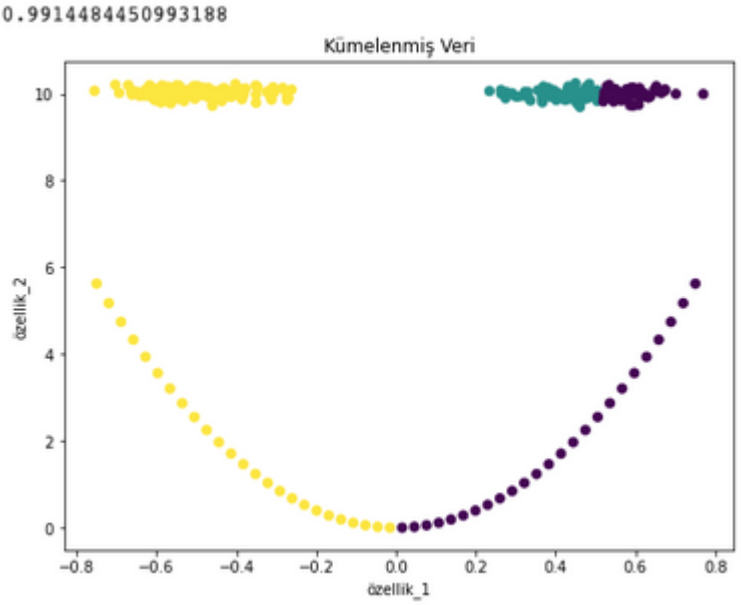
Şekil 6.17: Cityblock - L



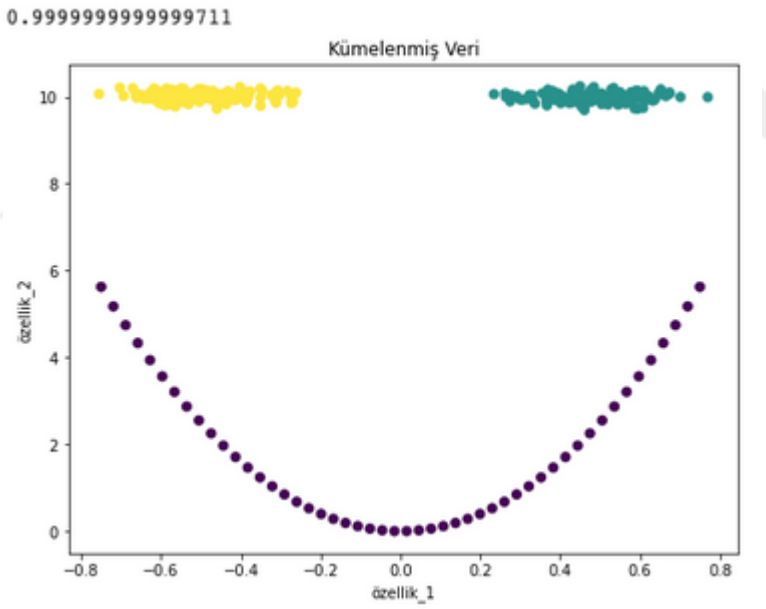
Şekil 6.18: Chebyshev - L



Şekil 6.19: Canberra - L



Şekil 6.20: Cosine - L



Şekil 6.21: Bray-Curtis - L

Sonuçlar toplu olarak aşağıda verilmiştir:

	L	L_{sym}	L_{rw}
Bray Curtis	0.9831515545901125	0.9663808194002818	0.9831515545902012
Canberra	0.8147888416594591	0.8262316211390218	0.8315216020471056
Chebyshev	0.9783271990057654	0.980107386167194	0.9783271990057086
Cityblock	0.9756451258744877	0.9999999999999978	0.9756451258744733
Cosine	0.9589796777387939	0.9498972746091934	0.9155276577911508
Euclidean	0.9999999971441901	0.9999999964387821	0.9999999974217948

Tablo 6.1: Noisy Moons

	L	L_{sym}	L_{rw}
Bray Curtis	0.9999999999999009	0.8384318920094864	0.9999999999999247
Canberra	0.6887746900032858	0.6712992342698152	0.6588678674039116
Chebyshev	0.9999999999999787	0.9999999999904378	0.9999999999999604
Cityblock	0.9999999999999719	0.9999999999999792	0.9999999999999989
Cosine	0.997592801363221	0.9774739853603731	0.9822451540571193
Euclidean	0.9999999964142926	0.9999999977729033	0.9999999963393109

Tablo 6.2: İç içe çember

	L	L_{sym}	L_{rw}
Bray Curtis	0.9999999999999711	0.9999999999999964	0.9999999999999972
Canberra	0.8031440281569725	0.7969235362610836	0.8081768659048145
Chebyshev	0.9999999999999962	0.9999999999999966	0.9999999999999974
Cityblock	0.9999999999999987	0.9999999999999964	1.0
Cosine	0.9914484450993188	0.9901801064498377	0.9901801064498377
Euclidean	0.9999999984402748	0.9999999976211893	0.9999999975880882

Tablo 6.3: Gülen Yüz

Burada kümelemede yaygın olarak kullanılan K-ortalamalar kümeleme algoritmasının başarılı şekilde ayıramadığı kümeler tercih edilmiştir. Sonuç olarak;

Noisy Moons veri seti için;

- Cityblock - L_{sym} kombinasyonu 0.9999999999999978 Silhoutte skoru ile en iyi sonucu vermiştir,
- En kötü sonucu veren kombinasyon Canberra - L kombinasyonu olmuştur (Silhoutte skoru: 0.8147888416594591),

İç İçe Çember veri seti için;

- Cityblock - L_{rw} kombinasyonu 0.9999999999999989 Silhoutte skoru ile en iyi sonucu vermiştir,
- En kötü sonucu veren kombinasyon Canberra - L_{rw} kombinasyonu olmuştur (Silhoutte skoru: 0.6588678674039116),

Gülen Yüz veri seti için;

- Yine Cityblock - L_{rw} kombinasyonu 1.0 Silhoutte skoru ile en iyi sonucu vermiştir,
- En kötü sonucu veren kombinasyon Canberra - L_{sym} kombinasyonu olmuştur (Silhoutte skoru: 0.7969235362610836).

  Dolayısıyla bu 3 veri setinin incelenmesi sonucunda Cityblock en iyi, Canberra ise en kötü sonuçları vermiştir.

Bölüm 7

SONUÇ

Bu tezin birinci bölümünde spektral kümelemenin tarihçesinden bahsedilmiş, 2.,3. ve 4. bölümlerde spektral kümeleme algoritmalarının teorik alt yapısı tanım ve teoremlerle açıklanmıştır. 5. bölümde bu teoriye dayanarak oluşturulmuş kümeleme algoritmaları verilmiştir.

Son bölümde ise kümelemede veri noktaları arasındaki uzaklığın hesaplanmasında kullanılan standart uzaklık fonksiyonu "Öklid uzaklığı" yerine farklı uzaklık fonksiyonları alınarak daha başarılı bir kümeleme elde edilip edilemeyeceği, 3 veri seti üzerinde incelenmiştir.

Bu tezin spektral kümeleme algoritmalarının altında yatan matematiksel temelin anlaşılması açısından fayda sağlaması amaçlanmış, ayrıca spektral kümelemede standart olarak kullanılan Öklid metriği yerine farklı uzaklık fonksiyonlarının alınmasının kümelemeyi nasıl etkilediği 3 veri seti ve 6 uzaklık fonksiyonu üzerinden incelenmiştir. Karşılaştırılan uzaklık fonksiyonlarının ve veri setlerinin sayısı arttırılarak daha iyi bir bakış açısı kazanılabilecektir. Farklı uzaklık fonksiyonlarının neden daha iyi ya da daha kötü sonuç verdiğinin matematiksel olarak açıklanması ise bu alana büyük katkı sağlayacaktır.

Kaynakça

[1] Hall, K. M. (1970) *An r-Dimensional Quadratic Placement Algorithm*, Management Science, 17(3), 219–229.

[2] Donath, W. E., Watson, T. J.(1973) *Lower Bounds for the Partitioning of Graphs*, IBM Journal of Research and Development, Volume: 17(5), 420 - 425.

[3] Fiedler, M. (1973) *Algebraic connectivity of graphs*, Czechoslovak Mathematical Journal, Vol. 23 (1973), No. 2, 298–305.

[4] Chung, F.R.K. (1997) *Spectral Graph Theory*, American Mathematical Society. CBMS Regional Conference Series in Mathematics in American Mathematical Society, 212, 92.

[5] Shi, J., and J. Malik (1997) *Normalized cuts and image segmentation*, Proc. of IEEE Conf. on Comp. Vision and Pattern Recognition, Puerto Rico.

[6] Ng, A. Y., M. I. Jordan, and Y. Weiss (2001) *On Spectral Clustering: Analysis and an algorithm*, Advances in Neural Information ProcSystems 14.

[7] von Luxburg, U. (2007) *A Tutorial on Spectral Clustering*, Statistics and Computing, Volume 17(4), 395-416.

[8] Koplik,G. (2017) *Spectral Clustering Theory and Implementation* https://gjkoplik.github.io/spectral_clustering/

[9] Singh, A., Yadav, A. and Rana, A., (2013) *K-means with Three different Distance Metrics*, International Journal of Computer Applications 67(10), 13-17.

[10] Ghazal, T.M. et al. (2021) *Performances of K-Means Clustering Algorithm with Different Distance Metrics*, Intelligent Automation Soft Computing, 30(2), 735-742.

[11] Rousseeuw, P.J. (1987) *Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis*, Comput. Appl. Math. 20, 53-65.

[12] Rodgers,B. (2018) *A proof of the spectral theorem for symmetric matrices (Optional)* https://mast.queensu.ca/~br66/419/spectraltheoremproof.pdf