DEU FMD 26(77) (2024) 237-241

Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi
Dokuz Eylul University Faculty of Engineering Journal of Science and Engineering
Elektronik/Online ISSN: 2547-958X

ARAŞTIRMA MAKALESI / RESEARCH ARTICLE

Farklı Uzaklık Fonksiyonlarının Spektral Kümeleme Algoritmasının Performansına Etkisi

Effect of Different Distance Measures on the Performance of Spectral Clustering Algorithm

Gülay İlona Telsiz Kayaoğlu [1*], Mustafa Eroğlu [2]

[1] Mimar Sinan Güzel Sanatlar Üniversitesi Fen Edebiyat Fakültesi Matematik Bölümü, İstanbul, TÜRKİYE
[2] Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı, İstanbul, TÜRKİYE
Sorumlu Yazar / Corresponding Author *: gulay.telsiz@msgsu.edu.tr

Öz

Makine öğrenmesinin bir kolu olan denetimsiz öğrenme problemlerinde kullanılan kümeleme algoritmaları, veri noktalarını benzer özelliklere sahip olan gruplara ayırmak için veri noktaları arasındaki uzaklıkları ölçen bir uzaklık fonksiyonu kullanır, ve bu, standart durumda Öklid uzaklığıdır. Bununla birlikte en sık kullanılan kümeleme algoritmalarından k-ortalamalar (k-means) kümeleme algoritmasında Öklid uzaklığı yerine farklı uzaklık fonksiyonları kullanılarak elde edilen sonuçların karşılaştırıldığı [1],[2] gibi çalışmalar mevcuttur. Bu çalışmada ise Spektral kümeleme algoritması farklı uzaklık fonksiyonları ile ele alınarak sonuçlar değerlendirilmiştir. K-ortalamalar algoritmasının başarılı şekilde ayıramadığı veri kümeleri tercih edilmiş ve spektral kümeleme algoritmasında Öklid uzaklığının yanı sıra farklı uzaklık fonksiyonları da kullanarak daha iyi bir kümeleme yapılıp yapılmayacağı incelenmiştir.

Anahtar Kelimeler: *Denetimsiz Öğrenme, Spektral Kümeleme, Uzaklık Fonksiyonları*

Abstract

Clustering algorithms used in unsupervised learning problems, which is a branch of machine learning, use a distance function that measures the distances between data points to separate data points into groups with similar characteristics, and this is known as the Euclidean distance in the standard case. However, there are studies such as [1] and [2] in which the results obtained by using different distance functions instead of Euclidean distance in the K-means clustering algorithm, which is one of the most frequently used clustering algorithms, are compared. In this study, the Spectral clustering algorithm is handled with different distance functions and its results are evaluated. The datasets that the k-means algorithm could not separate successfully were preferred and it was examined whether a better clustering could be made by using different distance functions in addition to the Euclidean distance in the spectral clustering algorithm.

Keywords: *Unsupervised Learning, Spectral Clustering, Distance Functions*

EXTENDED ABSTRACT

Introduction

Spectral Clustering is a method used to categorize points with similar characteristics on a graph. This technique relies on mathematical concepts such as graph theory and linear algebra to sort points based on their proximity relationships in the graph.

The Spectral clustering method utilizes a Laplacian operator to determine the positions of points in a graph. This operator examines relationships among points and represents them as a matrix. Subsequently, the eigenvectors corresponding to the smallest eigenvalues of this matrix are calculated, and these eigenvectors are employed to partition the points in the graph into groups with similar characteristics.

Materials and Methods

In this study, the impact of different distance functions on Spectral Clustering is explored by considering three different Spectral Clustering algorithms and six different distance functions.

Spectral Clustering Algorithms:

Three algorithms were employed: the unnormalized Spectral Clustering Algorithm," "Shi & Malik's" normalized Spectral Clustering algorithm, and "Ng, Jordan & Weiss's" normalized Spectral Clustering algorithm. The difference between these algorithms lies in the Laplacian matrices used. The Laplacian matrices used in these algorithms are given as follows:

$$L = D - W \quad (1)$$

$$L_{rw} = D^{-1}L = I - D^{-1}W \quad (2)$$

$$L_{\text{sym}} = D^{-1/2}LD^{-1/2} \quad (3)$$

The steps of the algorithms to be used are as follows [3]:

DOI:10.21205/deufmd.2024267706

Geliş Tarihi / Received: 12.04.2023
Kabul Tarihi / Accepted: 11.08.2023

Atıf şekli / How to cite:
Kayaoglu Telsiz, G.I.., Eroglu, M. (2024). Farklı Uzaklık Fonksiyonlarının Spektral Kümeleme Algoritmasının Performansına Etkisi. DEUFMD, 26(77), 237-241.

Input: Similarity matrix $S \in \mathbb{R}^{n\times n}$, number k of clusters to construct.

Step 1. Construct a similarity graph
Step 2. Compute the Laplacian matrix
Step 3. Compute the first k eigenvectors $u_1, u_2 \dots, u_k$ (the eigenvectors corresponding to the smallest k eigenvalues).
Step 4. Let U be the matrix containing the vectors $u_1, u_2, \dots, u_k$ as columns.
Step 5. For $i = 1, \dots, n$ let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i -th row of U. Cluster the points $(y_i)_{i=1,\dots,n}$ with the k-means algorithm into clusters $C_1, C_2, \dots, C_k$

Output: Clusters $A_1, \dots, A_k$ with $A_i = \{j \mid y_j \in C_i\}$

Distance Functions:

The study employs six different distance functions:
Euclidean Distance, Manhattan Distance (Cityblock), Chebyshev Distance, Canberra Distance, Cosine Distance, Bray-Curtis Distance.

Results

Six distinct distance functions, as defined in section 2.2, are employed to assess clustering performance. The "Silhouette score," rooted in [4], evaluates the adequacy of clustered data within clusters, with a score ranging from -1 to +1.

A score close to 1 signifies well-clustered data. Results reveal that the Canberra distance function produced the lowest score across all three datasets, while the Manhattan (Cityblock) distance function consistently achieved the highest clustering success.

Conclusion

This study explores the consequences of employing different distance functions, departing from the standard Euclidean metric in spectral clustering, on clustering across three datasets and six distance functions. While the Canberra distance function resulted in the lowest score, the Manhattan (Cityblock) distance function consistently achieved the highest clustering success.

To provide a more comprehensive perspective, expanding the comparison to include additional distance functions and datasets is recommended. Furthermore, a detailed exploration into the theoretical reasons behind the superior or inferior performance of various distance functions would significantly contribute to advancing this field.

---

1. Giriş

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.

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 özdeğerlerine karşılık gelen özvektörler hesaplanır ve bu özvektörler, graftaki noktaları benzer özellikleri olan gruplara ayırmak için kullanılır.

2. Materyal ve Metot

Bu çalışmada farklı uzaklık fonksiyonlarının spektral kümelemeye etkisi incelenirken üç farklı spektral kümeleme algoritması ve 6 farklı uzaklık fonksiyonu ele alınmıştır.

2.1. Spektral Kümeleme Algoritmaları

Bu üç veri kümesi ve farklı uzaklık fonksiyonları kullanılarak, "Normalize Edilmemiş Spektral Kümeleme Algoritması" ile "Shi & Malik'in" ve "Ng, Jordan & Weiss'in" normalize edilmiş spektral kümeleme algoritmaları olmak üzere üç farklı algoritma ile kümelenerek sonuçlar karşılaştırılmıştır. Bu algoritmalar arasındaki fark kullanılan Laplasyen matrislerin farklı olmasıdır. Bu algoritmalarda kullanılan Laplasyen matrisler sırasıyla aşağıda verilmiştir;

$$L = D - W \quad (1)$$
$$L_{rw} = D^{-1}L = I - D^{-1}W \quad (2)$$
$$L_{\text{sym}} = D^{-1/2}LD^{-1/2} \quad (3)$$

Burada W ağırlık matrisi, D derece matrisidir.

Kullanılacak algoritmaların adımları ise şu şekildedir [3]:

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

1. Adım: Benzerlik grafını oluşturma
2. Adım: Laplasyen matrisini hesaplama
3. Adım: Laplasyen matrisin en küçük ilk k özdeğerine karşılık gelen $u_1, u_2 \dots, u_k$ özvektörlerini hesaplama
4. Adım: Sütunları $u_1, \dots, u_k$ olan $U \in \mathbb{R}^{n\times k}$ matrisini oluşturma.
5. Adım: $i = 1, \dots, n$ için $y_i \in \mathbb{R}^k$, U matrisinin i.satıra denk gelen vektör olsun. K-ortalamalar ile $(y_i)_{i=1,\dots,n}$ noktalarını $C_1, C_2, \dots, C_k$ biçiminde kümeleme.

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

2.2. Uzaklık Fonksiyonları

Çalışmada kullanılan 6 farklı uzaklık fonksiyonu aşağıda verilmiştir:

$x, y \in \mathbb{R}^n$ için Öklid (Euclidean) uzaklığı:

$$d_2(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (4)$$

$x, y \in \mathbb{R}^n$ için Manhattan uzaklığı (Cityblock):

$$d_1(x,y) = \sum_{i=1}^{n} |x_i - y_i| \quad (5)$$

$x, y \in \mathbb{R}^n$ için Chebyshev Uzaklığı:

$$d_\infty(x,y) = \max_i |x_i - y_i| \quad (6)$$

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

$$d_{\text{can}}(x,y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|} \quad (7)$$

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

$$d_{\cos}(x,y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \quad (8)$$

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

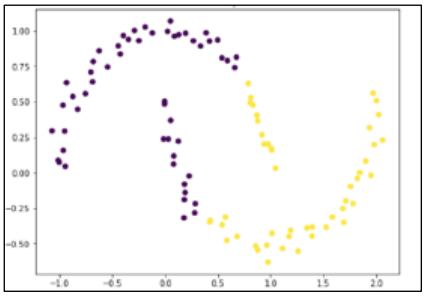
$$d_{\text{bray}}(x,y) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} (x_i + y_i)} \quad (9)$$

3. Bulgular

Bu kısımda üç veri kümesi 2.1.'de açıklanan üç Laplasyen matris (L, L_{sym}, L_{rw}) kullanılarak spektral kümeleme algoritması ile kümelenirken 2.2.'de tanımlanmış olan 6 farklı uzaklık fonksiyonu kullanılmıştır. Kümeleme başarısını ölçmek için, kümelenen verilerin bulunduğu kümedeki uygunluğunu bulmak için geliştirilen ve temeli [4] makalesine dayanan "Silhoutte skoru" kullanılmıştır. Bu değer, -1 ile +1 arasında değişmekte olup değerinin 1'e yakın olması verilerin iyi kümelenmiş olduğunu gösterir.

3.1. Noisy Moons Veri Kümesi:

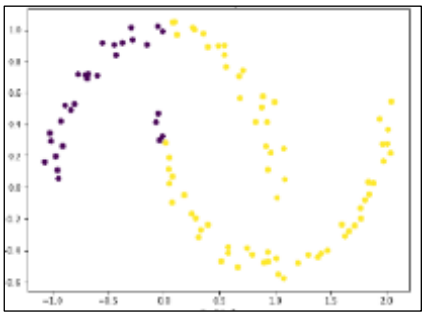
Bu veri kümesi iç içe iki ay şeklinden oluşmaktadır. K-means algoritması bu kümeyi aşağıdaki gibi kümelemektedir:



Şekil 1. Noisy moons veri kümesinin k-means ile kümelenmesi (Silhoutte skoru: 0.481737)

Figure 1. Clustering of noisy moons dataset with k-means (Silhoutte score: 0.481737)

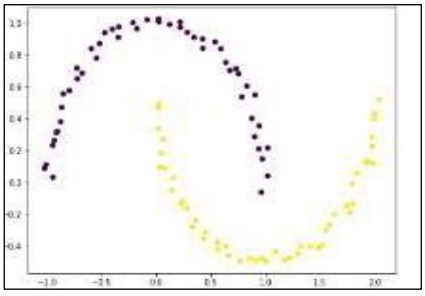
Bu veri kümesi için en kötü sonucu veren kombinasyon Canberra - L kombinasyonu olmuştur;



Şekil 2. Noisy moons veri kümesi için Canberra - L kombinasyonu (Silhoutte skoru: 0.81479)

Figure 2. Canberra - L combination for the noisy moons dataset (Silhoutte score: 0.81479)

En iyi sonucu veren kombinasyon ise Cityblock - L_{sym} kombinasyonu olmuştur:

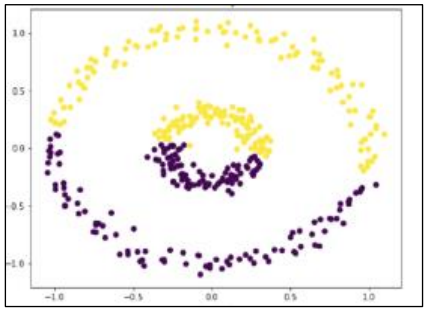


Şekil 3. Noisy moons veri kümesi için Cityblock - L_{sym} kombinasyonu (Silhoutte skoru: 0.99999)

Figure 3. Cityblock - L_{sym} combination for the noisy moons dataset (Silhoutte score: 0.99999)

3.2. İç İçe Çember Veri Kümesi:

İç içe iki çemberden oluşan bu veri setini k-means algoritması aşağıdaki gibi kümelemektedir:

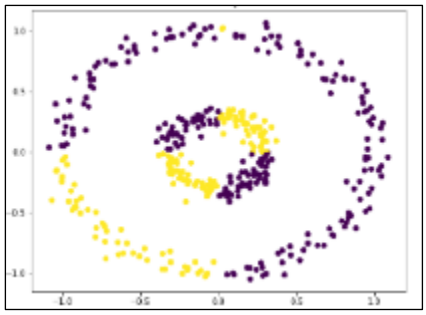


Şekil 4. İç içe çemberler veri kümesinin k-means ile kümelenmesi (Silhoutte skoru: 0.291660)

Figure 4. K-means result on the nested circles dataset (Silhoutte score: 0.291660)

Görüldüğü gibi k-means ile kümeleme yapıldığında iki çember iki ayrı küme olarak kümelenmemiştir.
Spektral kümeleme kullanıldığında ise aşağıdaki sonuçlara ulaşılmıştır:

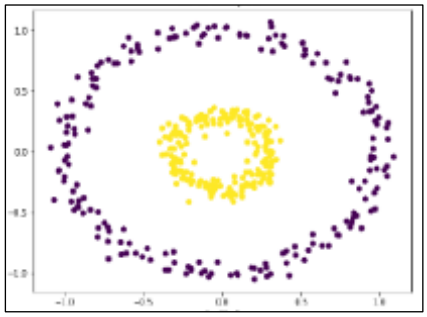
En kötü sonucu veren kombinasyon Canberra - L_{rw}kombinasyonu olmuştur;



Şekil 5. İç içe çemberler veri kümesi için Canberra - L_{rw} kombinasyonu (Silhoutte skoru: 0.658868)

Figure 5. Canberra - L_{rw} combination for the dataset of nested circles (Silhoutte score: 0.658868)

En iyi sonucu veren kombinasyon ise Cityblock - L_{rw} kombinasyonu olmuştur:

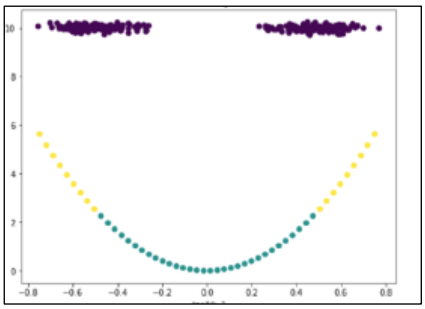


Şekil 6. İç içe çemberler veri kümesi için Cityblock - L_{rw} kombinasyonu (Silhoutte skoru: 0.99999)

Figure 6. Cityblock - L_{rw} combination for the dataset of nested circles (Silhoutte score: 0.99999)

3.3. Gülen Yüz Veri Kümesi:

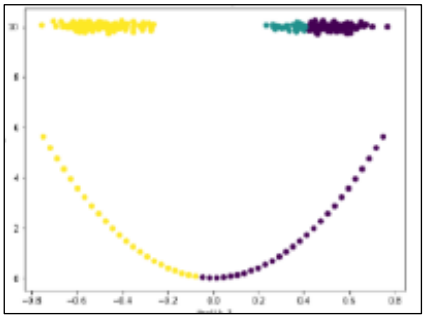
Bu veri kümesini k-means algoritması aşağıdaki gibi kümelemektedir:



Şekil 7. Gülen yüz veri kümesinin k-means ile kümelenmesi (Silhoutte skoru: 0.8605798)

Figure 7. Clustering of smiley face dataset with k-means (Silhoutte score: 0.8605798)

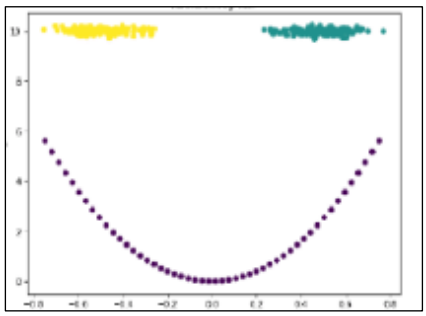
Bu veri kümesi için en kötü sonucu veren kombinasyon Canberra - L_{sym} kombinasyonu olmuştur;



Şekil 8. Gülen yüz veri kümesi için Canberra - L_{sym} kombinasyonu (Silhoutte skoru: 0.7969235)

Figure 8. Canberra - L_{sym} combination for the smiley face dataset (Silhoutte score: 0.7969235)

En iyi sonucu veren kombinasyon ise Cityblock - L_{rw} kombinasyonu olmuştur:



Şekil 9. Gülen yüz kümesi için Cityblock - L_{rw} kombinasyonu (Silhoutte skoru: 1.0)

Figure 9. Cityblock - L_{rw} combination for smiley face dataset (Silhoutte score: 1.0)

Sonuç olarak spektral kümeleme algoritması kullanılarak yapılan kümelemede Canberra uzaklık fonksiyonu üç veri kümesinde de en kötü skoru vermiş, Manhattan (Cityblock) uzaklık fonksiyonu ise üç veri kümesinde de en iyi kümeleme başarısına ulaşmıştır.

Bu üç veri setinin üç Laplasyen matrisi ve 6 uzaklık fonksiyonu kullanılarak kümelenmesi sonucunda elde edilen kümelenmiş verinin Silhoutte skorları toplu olarak aşağıdaki tablolarda verilmiştir:

Tablo 1. Noisy moons veri kümesi için elde edilen sonuçlar (Silhoutte skorları)

Table 1. Results obtained for the noisy moons dataset (Silhoutte scores)

Uzaklık Fonksiyonu	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 2. İç içe çemberler veri kümesi için elde edilen sonuçlar (Silhoutte skorları)

Table 2. Results obtained for the nested circles dataset (Silhoutte scores)

Uzaklık Fonksiyonu	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 3. Gülen yüz veri kümesi için elde edilen sonuçlar (Silhoutte skorları)

Table 3. Results obtained for the smiley face dataset (Silhoutte scores)

Uzaklık Fonksiyonu	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

4. Sonuçlar

Bu çalışmada spektral kümelemede standart olarak kullanılan Öklid metriği yerine farklı uzaklık fonksiyonlarının alınmasının kümelemeyi nasıl etkilediği üç veri seti ve 6 uzaklık fonksiyonu üzerinden incelenmiştir.

Canberra uzaklık fonksiyonu üç veri kümesinde de en kötü skoru vermiş, Manhattan (Cityblock) uzaklık fonksiyonu ise üç veri kümesinde de en iyi kümeleme başarısına ulaşmıştır.

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 teorik olarak açıklanması ise bu alana büyük katkı sağlayacaktır.

Etik kurul onayı ve çıkar çatışması beyanı

Hazırlanan makalede etik kurul izni alınmasına gerek yoktur.

Hazırlanan makalede herhangi bir kişi/kurum ile çıkar çatışması bulunmamaktadır.

Yazar katkılarının beyanı

Bu çalışma birinci yazarın danışmanlığındaki ikinci yazarın "Spektral Kümeleme ve Farklı Uzaklık Fonksiyonlarının Spektral Kümelemeye Etkisinin İncelenmesi" başlıklı yüksek lisans tezinden üretilmiştir.

Fikir oluşturma ve teorik alt yapının kurulmasında 1. yazar, Python ile analizlerin gerçekleştirilmesi ise 2. yazar tarafından gerçekleştirilmiştir. Bu bağlamda yazarlar çalışmaya farklı yönlerden eşit oranda katkı sağlamışlardır.

Kaynaklar

[1] Singh, A., Yadav, A., Rana, A., 2013. K-means with Three different Distance Metrics, International Journal of Comp. Applications, Cilt. 67(10), s.13-17. DOI:10.5120/11430-6785

[2] Ghazal, T.M. et al., 2021. Performances of K-Means Clustering Algorithm with Different Distance Metrics, Intelligent Automation & Soft Computing, Cilt. 30(2), s. 735-742. DOI:10.32604/iasc.2021.019067

[3] von Luxburg, U., 2007, A Tutorial on Spectral Clustering, Statistics and Computing, Cilt. 17(4), s. 395-416. DOI:10.1007/s11222-007-9033-z

[4] Rousseeuw, P.J. (1987) Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis, Comput. Appl. Math. Cilt. 20, s. 53-65. DOI:10.1016/0377-0427(87)90125-7