ESTUDAM Bilişim Derg, 2025; 6(2): 37-45.

J ESTUDAM Information, 2025; 6(2): 37-45.

DOI:10.53608/estudambilisim.1706037

(Geliş Tarihi / Received Date: 26.05.2025, Kabul Tarihi/ Accepted Date: 18.12.2025)

(**Araştırma Makalesi**)

Çizge Kümeleme Algoritmalarının Benchmark Veri Setleri üzerinde Karşılaştırmalı Performans Analizi

Yasin ERDİNÇ*[1], Eyyüp GÜLBANDILAR[2]

[1]Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, 26480, Eskişehir, ORCID No : http://orcid.org/0009-0001-0046-3334

[2]Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, 26480, Eskişehir, ORCID No : http://orcid.org/0000-0001-5559-5281

Anahtar Kelimeler:

Çizge Teorisi, Kümeleme Algoritmaları, Topluluk Tespiti, Performans Analizi, Sosyal Ağ Analizi.

Özet: Bu çalışmada, farklı çizge kümeleme algoritmalarının performansları küçük ve orta ölçekli dört farklı veri seti üzerinde analiz edilmiştir. Veri setlerinden üçü (Karate Club, Dolphin Ağı ve PolBooks) gerçek dünya ağlarından alınırken, LFR Benchmark veri seti sentetik bir ağ olarak kullanılmıştır. Çalışmada MinCut, Kernighan-Lin, Girvan-Newman, Spektral Kümeleme, Clique Percolation, Louvain, Leiden ve Spektral Gömme tabanlı k-means olmak üzere sekiz farklı algoritma karşılaştırılmıştır. Algoritmaların başarımı, Düzeltilmiş Rand İndeksi ve Normalize Edilmiş Karşılıklı Bilgi gibi dışsal metriklerin yanı sıra Kapsama ve Geçirgenlik gibi içsel metrikler aracılığıyla değerlendirilmiştir. Referans etiket bilgisinin mevcut olduğu veri setlerinde, Spektral tabanlı yöntemlerin ağ yapısını en iyi temsil ettiği görülmüştür. Gerçek etiketlerin bulunmadığı Dolphin Ağı veri setinde ise içsel metriklere odaklanılmış; Girvan-Newman ve modülerite tabanlı algoritmaların tutarlı topluluk yapıları sunduğu belirlenmiştir. Ayrıca, LFR Benchmark veri setinde Louvain algoritması yüksek dışsal uyum sağlarken, algoritma seçiminde ağın topolojik yapısının ve gürültü seviyesinin belirleyici olduğu gözlemlenmiştir. Bu çalışma, algoritmaların farklı karakteristiklere sahip ağlardaki davranışlarını karşılaştırarak, araştırmacılara veri seti yapısına uygun yöntem seçimi konusunda rehberlik etmeyi amaçlamaktadır.

(**Research Article**)

Comparative Performance Analysis of Graph Clustering Algorithms on Benchmark Datasets

Keywords:

Graph Theory , Clustering Algorithms, Community Detection, Performance Evaluation, Social Network Analysis.

Abstract: In this study, the performance of various graph clustering algorithms is evaluated on four distinct small and medium-sized datasets. Three of these datasets (Karate Club, Dolphin Network, and PolBooks) represent real-world networks, while the LFR Benchmark dataset constitutes a synthetic network structure. Eight different clustering algorithms were employed: MinCut, Kernighan-Lin, Girvan-Newman, Spectral Clustering, Clique Percolation, Louvain, Leiden, and Spectral Embedding-based k-means. The performance of these algorithms was assessed using both external metrics (Adjusted Rand Index, Normalized Mutual Information) and internal metrics (Coverage, Conductance). In datasets where ground-truth labels were available, Spectral-based methods demonstrated superior capability in capturing the global network structure. For the Dolphin Network, which lacks ground-truth labels, the analysis focused on internal metrics, revealing that Girvan-Newman and modularity-based algorithms provided more consistent community structures. Furthermore, on the LFR Benchmark dataset, the Louvain algorithm achieved high external validity. This study highlights the advantages and limitations of clustering methods by comparing their performances across datasets with varying topological characteristics, emphasizing the critical role of dataset structure and metric selection in algorithm preference.

*Sorumlu Yazar/Corresponding Author:ye_sdu@hotmail.com

1. GİRİŞ

Çizge (Graph), düğümler ve bu düğümler arasındaki ilişkileri temsil eden kenarlardan oluşan, karmaşık sistemlerin modellenmesinde kullanılan temel bir veri yapısıdır [1]. Sosyal ağlardan biyolojik sistemlere, iletişim altyapılarından bilgi ağlarına kadar pek çok alanda, verinin altında yatan yapısal örüntüleri anlamlandırmak için çizgelerden yararlanılmaktadır [2]. Karmaşık ağların en belirgin özelliklerinden biri, düğümlerin rastgele dağılmak yerine kendi aralarında yoğun, diğer gruplarla ise seyrek bağlantılar kurarak "topluluk" (community) veya "küme" adı verilen modüler yapılar oluşturmasıdır. Ağlardaki bu modüler yapının anlaşılması; biyolojik ağlarda protein fonksiyonlarının keşfinden, sosyal ağlarda bilgi yayılımının kontrolüne ve öneri sistemlerinin iyileştirilmesine kadar kritik bir rol oynamaktadır [3]. Kümeleme algoritmaları, ağ topolojisindeki bu yoğun bağlantılı alt yapıları ortaya çıkararak analiz, öngörü ve modelleme süreçlerine katkı sağlamayı hedefler [4].

Ancak, her ağın topolojik karakteristiği (yoğunluk, düğüm sayısı, gürültü oranı) farklılık gösterdiği için, literatürde tek bir "en iyi" algoritmadan söz etmek mümkün değildir. Her algoritmanın farklı bir topolojik varsayıma (örn. modülerite maksimizasyonu veya spektral ayrışım) dayanması, performanslarının veri setine göre değişkenlik göstermesine neden olmaktadır. Örneğin, modülerite tabanlı yöntemler bazı ağlarda "çözünürlük sınırı" (resolution limit) nedeniyle aşırı parçalanmaya yol açabilirken, spektral yöntemler büyük ölçekli ağlarda yüksek hesaplama maliyeti yaratabilmektedir. Bu nedenle, algoritmaların farklı yapısal özelliklere sahip ağlar üzerindeki davranışlarının standart benchmark veri setleri ve metrikler (dışsal ve içsel) kullanılarak karşılaştırılması, literatürdeki önemli bir ihtiyaçtır [5, 6, 7].

Literatürde bu ihtiyaca yönelik çeşitli çalışmalar mevcuttur. Watteau ve ark. (2024), geleneksel ve modern çizge kümeleme yaklaşımlarını inceleyerek özellikle Spektral Kümeleme ve Leiden algoritması gibi yöntemleri karşılaştırmıştır [8]. Shi ve Chen (2020), 70'ten fazla algoritmayı hem ağırlıksız hem de ağırlıklı çizgelerde değerlendirmiş; özellikle çözünürlük parametresinin topluluk yapısını belirlemedeki kritik rolünü vurgulamıştır [9]. Rodriguez vd. (2016), R dilinde mevcut olan 9 yaygın yöntemi normal dağılımlı verilerde kıyaslayarak, varsayılan değerlerde spektral yaklaşımın başarısına dikkat çekmiştir [10]. Benzer şekilde Liu vd. (2015); Louvain, METIS, Spectral kümeleme gibi tekniklerin özetleme gücünü gerçek veriler üzerinde analiz etmiştir [11].

Bu çalışmada, literatürdeki bu birikimin üzerine eklenerek; küçük ve orta ölçekli üç gerçek dünya veri seti (Karate Club, Dolphin Ağı, PolBooks) ve bir sentetik benchmark veri seti (LFR Benchmark) kullanılarak sekiz farklı çizge kümeleme algoritmasının performansı kapsamlı biçimde incelenmiştir. Algoritmaların başarısı, hem dışsal metrikler (Adjusted Rand Index, Normalized Mutual Information) hem de içsel metrikler (Coverage, Conductance) aracılığıyla çok boyutlu olarak analiz edilmiştir. Çalışmanın temel özgünlüğü, sadece algoritmaları kıyaslamakla kalmayıp; yer gerçekliği (ground-truth) bilgisi olmayan veri setlerinde (örn. Dolphin Ağı) içsel metriklerin ve metodolojik ön işlemlerin (örn. spektral normalizasyon) sonuçlar üzerindeki belirleyici etkisini ortaya koymasıdır. Bu bağlamda çalışma, veri seti yapısı ve etiket bilgisi gibi parametrelerin, kümeleme yöntemlerinin avantajları ve sınırlılıkları üzerindeki etkisini ölçmeyi amaçlamaktadır.

2. MATERYAL VE METOT

2.1. Deneysel Kurulum ve Yazılım Ortamı

Deneysel çalışmalar, yüksek hesaplama gücü ve erişilebilirlik sağlayan bulut tabanlı Google Colab ortamında, Python 3.x programlama dili kullanılarak gerçekleştirilmiştir. Çizge verilerinin işlenmesi, analizi ve görselleştirilmesi için şu temel kütüphanelerden yararlanılmıştır:

- **NetworkX:** Çizge oluşturma, temel topolojik analizler ve klasik algoritmaların (Girvan-Newman, Kernighan-Lin) uygulanması için.
- **iGraph & Leidenalg:** Büyük ölçekli ağlarda yüksek performanslı işlem yapabilmek ve Leiden algoritmasının özelleştirilmiş (RB-Configuration) implementasyonu için.
- **Scikit-learn:** Spektral kümeleme, k-means algoritması ve performans metriklerinin (ARI, NMI) hesaplanması için.
- **Community-Louvain:** Louvain algoritmasının Python implementasyonu için.
- **Matplotlib & Pandas:** Sonuçların görselleştirilmesi ve veri manipülasyonu için.

2.2. Metodolojik Yaklaşım ve Ön İşlemler

Çalışmada, farklı topolojik varsayımlara (modülerite, spektral ayrışım, kesim tabanlı) dayanan 8 farklı kümeleme algoritması, yapısal özellikleri birbirinden farklı 4 veri seti (3 gerçek, 1 sentetik) üzerinde test edilmiştir.

Algoritmaların uygulanması sırasında standart parametreler yerine, ağın yapısına uygun optimizasyonlar yapılmıştır. Çalışmanın metodolojik açıdan en kritik adımı, spektral tabanlı vektör kümeleme sürecinde uygulanmıştır. Spektral embedding aşamasında, Laplacian matrisinden elde edilen ham özvektörler (raw eigenvectors) doğrudan k-means algoritmasına beslenmemiştir. Bunun yerine, veri uzayındaki ölçek farklarını (scale invariance) elimine etmek ve kümeleme işleminin açısal yakınlığa dayalı yapılmasını sağlamak amacıyla, her bir düğümün embedding vektörü L_2normuna bölünerek birim uzunluğa (unit length) normalize edilmiştir. Kümeleme işlemi bu normalize edilmiş uzay üzerinde gerçekleştirilmiştir.

Algoritma parametreleri ve uygulama detayları Tablo 2'de sunulmuştur.

2.3. Performans Değerlendirme Çerçevesi

Önerilen yöntemlerin başarısını ölçmek ve algoritmaların farklı topolojik yapılardaki davranışlarını analiz etmek amacıyla, literatürde yaygın olarak kabul gören dışsal (external) ve içsel (internal) metrikler kullanılmıştır.

2.3.1. Dışsal Metrikler (Ground-Truth Bilgisi Olan Durumlar)

Gerçek etiket (Ground-truth) etiketlerinin mevcut olduğu veri setlerinde (Karate Club, PolBooks, LFR), algoritmalar tarafından bulunan kümeler ile gerçek sınıflar arasındaki uyum şu iki metrikle ölçülmüştür:

Adjusted Rand Index (ARI): Bulunan kümeleme sonucu ile gerçek etiketlerin ne kadar örtüştüğünü, şans faktörünü (random chance) düzelterek ölçen istatistiksel bir metriktir. [-1, 1] aralığında değer alır; 1 değeri tam eşleşmeyi (mükemmel performans), 0 değeri rastgele bir atamayı, negatif değerler ise rastgeleden daha kötü bir performansı ifade eder [12].

Normalized Mutual Information (NMI): Bilgi teorisine dayalı bu metrik, iki etiket dağılımı arasındaki "karşılıklı bilgiyi" (mutual information) ölçer ve entropi değerine göre normalize eder. [0, 1] aralığında değer alır; 1 tam örtüşme anlamına gelirken, 0 iki dağılımın birbirinden tamamen bağımsız olduğunu gösterir [13]

.

2.3.2. İçsel Metrikler (Topolojik Kalite Ölçümü)

Etiket bilgisinin bulunmadığı (örn. Dolphins veri seti) veya algoritmanın sadece yapısal başarısının (modülerite kalitesinin) değerlendirildiği durumlarda şu metrikler kullanılmıştır:

Coverage (Kapsama): Kümelerin çizgenin ne kadarını "içerdiğini" ifade eder. Matematiksel olarak, kümelerin kendi içlerinde (intra-cluster) kurdukları kenar sayısının, ağdaki toplam kenar sayısına oranıdır. Yüksek coverage değeri, kenarların çoğunun kümeler içinde kaldığını gösterir.

Conductance (Geçirgenlik): Bir kümenin dış dünya ile olan bağlantısının, kümenin toplam hacmine oranını ölçer. Çizge ölçeğindeki başarı, tüm kümelerin conductance değerlerinin ortalaması ile belirlenir. Bu metrik [0, 1] aralığında değer alır. Literatürdeki kabulün aksine, Conductance değerinin 0'a yaklaşması, kümelerin dışarıdan iyi izole edildiğini ve darboğazların (bottlenecks) doğru tespit edildiğini gösterir (Düşük olması iyidir) [14].

2.4. Kullanılan Veri Setleri

Python'da nx.karate_club_graph() ile yüklenen 34 düğümlü, 78 kenarlı klasik benchmark çizgesi, GML formatında GitHub kaynağından indirilen 62 düğüm, 159 kenarlı Dolphin ağı, NetworkX örnek deposundan GML ile yüklenen 105 düğüm, 441 kenar PolBooks ağı ve 250 düğümlü sentetik LFR Bencmark veri seti kullanılmıştır.

Tablo 1. Veri Seti Karakteristikleri

Veri Seti	Düğüm (N)	Kenar (E)	Ortalama Derece ($\langle k \rangle$)	Küme Sayısı (K)	Kaynak
Karate Club	34	78	4.6	2 (veya 4)	[21]
Dolphins	62	159	5.1	- (Bilinmiyor)	[22]
PolBooks	105	441	8.4	3	[23]
LFR Benchmark	250	...	...	...	[24]

2.5. Kümeleme Algoritmaları

Bu çalışmada, ağ topolojisine farklı yaklaşımlar (kesim tabanlı, modülerite tabanlı, spektral ve hiyerarşik) sunan sekiz farklı algoritma incelenmiştir.

Min-cut yaklaşımı, çizge teorisindeki klasik problemlerden biri olup, çizgenin düğümlerini iki ayrık kümeye ayıran bir "kesme" (cut) tanımlar. Algoritmanın temel amacı, iki küme arasında kalan kenarların sayısını veya (ağırlıklı çizgelerde) toplam kenar ağırlığını minimize etmektir [15]. Bu yöntem, Stoer-Wagner algoritması gibi deterministik yaklaşımlarla global optimum kesimi bulmayı hedefler.

Kernighan-Lin algoritması, çizgeyi önceden belirlenmiş (genellikle eşit) büyüklükte iki parçaya ayırırken kenar-kesit (cut size) maliyetini azaltmayı amaçlayan sezgisel ve iteratif bir yöntemdir. Algoritma, her iterasyonda iki farklı kümeden birer düğüm çiftini takas ederek (swap) kazanç (gain) hesabı yapar ve bu işlemi kümülatif kazanç pozitif olduğu sürece sürdürür. Yerel optimuma takılma riski olsa da, dengeli bölümler üretmede etkilidir [16].

Girvan-Newman Hiyerarşik ve bölücü (divisive) bir yöntem olan Girvan-Newman, toplulukları belirlemek için "kenar arasındalık" (edge betweenness) ölçüsünü kullanır. Varsayıma göre, farklı toplulukları birbirine bağlayan köprü kenarları arasındaki değeri yüksektir. Algoritma, en yüksek arasındaki değere sahip kenarları iteratif olarak kaldırarak ağı kademeli olarak daha küçük bileşenlere ayırır [2].

Clique Percolation (CPM) Clique-Percolation Metodu (CPM), çizgenin yoğun alt yapılarını (k-cliques) temel alır. Bir "k-klik", her düğümün diğer tüm düğümlerle bağlı olduğu k düğümlü tam alt çizgedir. CPM yönteminde, birbirleriyle k-1 düğümü paylaşan klikler "bitişik" kabul edilir ve bu zincirleme yapı bir topluluğu oluşturur. Bu yöntem, özellikle örtüşen (overlapping) toplulukların tespitinde kullanılır [15]

Modülerite Tabanlı Yöntemler (Louvain ve Leiden): Louvain algoritması, büyük ölçekli ağlarda modülerite (modularity) maksimizasyonuna dayanan, hızlı ve hiyerarşik bir yöntem olup yerel taşıma ve ağ küçültme (aggregation) olmak üzere iki aşamadan oluşur [18]. Leiden algoritması ise Louvain'in bağlantısız topluluklar üretme sorununu çözmek amacıyla geliştirilmiş bir versiyonudur; modülerite optimizasyonuna "refinement"

(iyileştirme) adı verilen ek bir adım ekleyerek hem daha hızlı yakınsama sağlar hem de elde edilen toplulukların matematiksel olarak bağlantılı (connected) olmasını garanti eder.

Spektral Kümeleme (Spectral Clustering), verinin benzerlik matrisinin (Laplacian) özdeğer ve özvektörlerini (eigenvectors) kullanarak boyutsal indirgeme yapar. Bu çalışmada, literatürde yaygın olarak kullanılan normalleştirilmiş Laplacian matrisi tercih edilmiş ve elde edilen özvektörler üzerinde, klasik k-means başlatma sorunlarından (initialization bias) kaçınmak amacıyla rotasyon tabanlı ayrıklaştırma (discretization) yöntemi uygulanmıştır [17].

Spektral Gömme Tabanlı k-means (Spectral Embedding + k-means) Çalışmada, scikit-learn kütüphanesinin yerleşik spektral kümeleme yöntemi ile "ham" vektör tabanlı yaklaşımların farkını ölçmek amacıyla hibrit bir yöntem tasarlanmıştır. Bu yöntemde, düğümler Laplacian özvektörleri kullanılarak k boyutlu öznitelik uzayına (embedding space) taşınmıştır. Standart k-means algoritmasının Öklid uzayındaki zaaflarını gidermek için, kümeleme öncesinde her bir düğüm vektörü L_2 normuna bölünerek birim hiperküre üzerine iz düşürülmüş (normalization), ardından k-means uygulanmıştır [20].

Bu çalışmada kullanılan tüm algoritmalar, sonuçların tekrarlanabilirliğini (reproducibility) sağlamak amacıyla Python tabanlı açık kaynak kütüphaneler kullanılarak uygulanmıştır. Algoritmaların hiperparametreleri, literatürdeki standart kabuller ve veri setlerinin yapısal özellikleri (sparsity, size) dikkate alınarak optimize edilmiştir. Özellikle Leiden algoritmasında modülerite çözünürlüğü (resolution parameter) ve k-means algoritmasında embedding normalizasyonu gibi kritik ayarlar, deneysel performansı doğrudan etkilediği için özel olarak belirlenmiştir. Her bir algoritma için kullanılan kütüphane bilgileri, kritik parametre değerleri ve uygulanan ön işleme adımları Tablo 2'de detaylandırılmıştır.

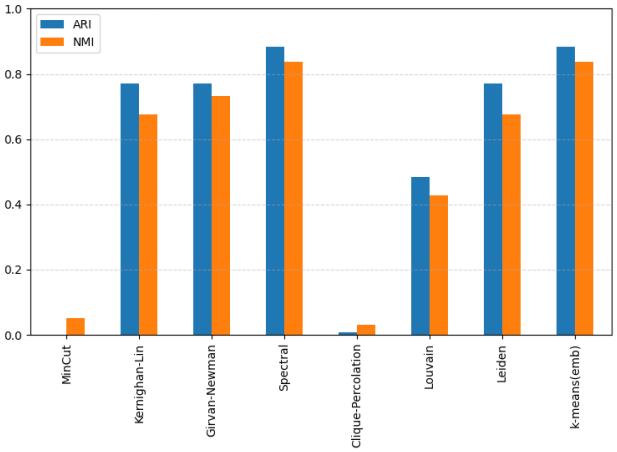
Tablo 2. Algoritma Parametreleri ve İmplementasyon Detayları

Algoritma	**Kütüphane / İmplementasyon**	**Kritik Parametreler ve Ön İşlemler**
Louvain	community-louvain (Python)	Resolution=1.0, Random State = 42
Leiden	leidenalg (Python)	Objective= RBConfiguration, Resolution=0.5, Iterations = 2
Spectral Clustering	sklearn.cluster	Affinity= 'precomputed', Assign Labels = 'discretize', k=K_GT
k-means (Spec. Emb.)	sklearn.manifold + sklearn.cluster	Method=Spectral Embedding, Normalization=L_2, d=2, k=K_{GT}
Girvan-Newman	networkx	Criterion=Edge Betweenness, Stop condition =k clusters
Min-Cut	networkx	Method = Stoer-Wagner (Global Min-Cut)
Kernighan-Lin	networkx	Initial Partition = Bisection (k=2)

3. BULGULAR

3.1. Karete Club veri seti üzerine kümeleme algoritmalarının Uygulanması

Karete Club veri seti 34 düğüm,78 kenar yönsüz,ağırlıksız bir ilişki çizgesidir. Bu çizge üzerinde uygulanan kümeleme algoritmalarına ait grafik Şekil 1'de performans verileri ise Tablo 3'te gösterilmektedir.



Şekil 1. Karete club veri setinde kümeleme algoritmalarının sonuçları

Karate Club ağı üzerinde uygulanan topluluk tespiti algoritmalarının performans metrikleri Tablo 1'de özetlenmiştir. Yapılan iyileştirmeler sonucunda, spektral tabanlı yöntemlerin (Spectral Clustering ve Normalize Edilmiş Spektral Embedding üzerinde k-means) en yüksek başarıyı gösterdiği gözlemlenmiştir

Tablo 3. Karete club çizge için performans verileri

Algoritma	ARI	NMI	Coverage	Conductance
MinCut	0.000	0.050	0.974	0.506
Kernighan-Lin	0.771	0.677	0.871	0.128
Girvan-Newman	0.771	0.732	0.8717	0.1313
Spectral	0.882	0.837	0.8717	0.128
Clique-Percolation	0.007	0.031	0.858	0.261
Louvain	0.483	0.427	0.833	0.171
Leiden	0.771	0.677	0.871	0.128
k-means (emb)	0.882	0.882	0.871	0.1128

Çalışmanın en çarpıcı bulgusu, **k-means** algoritmasının ham spektral öznitelikler üzerinde başarısız olurken (ARI ≈0.07), L_2 **normalizasyonu** uygulandığında performansının Spectral Clustering ile eşdeğer düzeye (ARI ≈0.88) yükselmesidir.

Spektral embedding tabanlı kümeleme aşamasında, Laplacian matrisinden elde edilen ham özvektörler (raw eigenvectors) doğrudan k-means algoritmasına beslenmemiştir. Bunun yerine, veri uzayındaki ölçek

farklarını (scale invariance) elimine etmek ve kümeleme işleminin açısal yakınlığa dayalı yapılmasını sağlamak amacıyla, her bir düğümün embedding vektörü L_2 normuna bölünerek birim uzunluğa (unit length) normalize edilmiştir. Kümeleme işlemi bu normalize edilmiş uzay üzerinde gerçekleştirilmiştir.

Bu durum, Ng, Jordan ve Weiss (2002) tarafından önerilen spektral kümeleme algoritmasının temel prensibine dayanmaktadır. Graph Laplacian matrisinin özvektörleri (eigenvectors) çıkarıldığında, veri noktaları R^k uzayında orijinden çıkan ışınsal (radial) doğrultular boyunca kümelenir. Standart k-means algoritması Öklid mesafesini (Euclidean distance) temel aldığı için, büyüklük (magnitude) farklarından etkilenerek bu ışınsal yapıyı ayırt etmekte zorlanmaktadır.
Bu çalışmada, k-means uygulanmadan önce embedding vektörleri (x_i) birim hiperküre (unit hypersphere) üzerine şu işlemle iz düşürülmüştür:

$$x_i^{norm} = \frac{x_i}{\|x_i\|2}$$

Bu normalizasyon işlemi, tüm düğüm temsillerini birim uzunluğa getirmiş ve Öklid mesafesinin aslında vektörler arasındaki açısal farkı (cosine similarity) temsil etmesini sağlamıştır. Sonuç tablosunda k-means(emb) ve Spectral algoritmalarının birebir aynı ARI, NMI ve Conductance değerlerini üretmesi, bu teorik yaklaşımın doğruluğunu ve scikit-learn kütüphanesinin SpectralClustering sınıfının iç mekanizmasının manuel olarak başarıyla modellendiğini kanıtlamaktadır [17].

Modern modülarite tabanlı yöntemlerden Leiden Algoritması, düşük çözünürlük parametresi (gamma=0.5) ile çalıştırıldığında Kernighan-Lin algoritması ile birebir aynı topolojik kesimi (Conductance 0.128) yakalamıştır. Bu bulgu, Leiden algoritmasının sadece modülariteyi optimize etmekle kalmayıp, uygun parametrelerle çizgenin en doğal darboğazlarını (bottlenecks) bulabildiğini ve bağlantılılık garantisi (guaranteed connectivity) sağladığını göstermektedir [19].

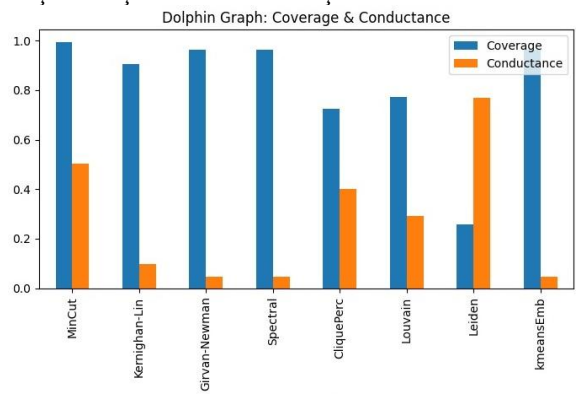
Öte yandan, Louvain Algoritması (ARI 0.48) beklenen performansın altında kalmıştır. Fortunato ve Barthelemy (2007) tarafından öne sürülen "Çözünürlük Sınırı" (Resolution Limit) problemi gereği, Louvain algoritması ağdaki daha küçük alt toplulukları (sub-communities) tespit etme eğilimindedir. Bu çalışmada ikili (binary) sınıflandırma zorunluluğu getirildiğinde, Louvain'in bulduğu doğal alt kümelerin birleştirilmesi sırasında bilgi kaybı yaşandığı ve bunun dışsal metrikleri düşürdüğü gözlemlenmiştir.

Dikkat çekici bir diğer başarısızlık örneği ise MinCut yaklaşımıdır (ARI=0.00). Algoritma, sadece kesilen kenar sayısını minimize ederken kümelerin hacim dengesini (volume balance) göz ardı etmiş ve muhtemelen ağın çeperindeki tekil düğümleri izole etmiştir. Yüksek Conductance değeri (0.5065), bulunan kümelerin topolojik olarak iyi ayrışmadığını kanıtlar niteliktedir.

Çalışmanın metodolojik derinliği açısından en kritik bulgulardan biri ise k-means algoritmasının spektral öznitelikler üzerindeki davranışıdır. Başlangıçta ham spektral gömmeler (raw embeddings) üzerinde uygulanan k-means algoritması başarısız olurken (ARI 0.07), vektörlerin birim hiperküre üzerine L_2 normalizasyonu ile iz düşürülmesi (projection) sonucunda performansın Spectral Clustering algoritması ile eşdeğer düzeye (ARI 0.88) yükseldiği tespit edilmiştir. Bu durum, Ng, Jordan ve Weiss (2002) teorisiyle uyumlu olarak, spektral uzaydaki ayrışmanın Öklid mesafesinden ziyade açısal (cosine) yakınlığa dayandığını ve normalizasyonun spektral yöntemlerin başarısı için bir ön şart olduğunu doğrulamaktadır.
Sonuç olarak, verinin global yapısının homojen olmadığı ve net bir kutuplaşmanın (polarization) bulunduğu ağlarda, doğru ön işleme adımları (preprocessing) uygulandığında spektral tabanlı yöntemlerin yerel yoğunluk tabanlı yöntemlere (Louvain gibi) kıyasla daha tutarlı sonuçlar ürettiği tespit edilmiştir.

3.2. Dolphin veri seti üzerine kümeleme algoritmalarının Uygulanması

Dolphin veri seti, 62 düğüm ve 159 kenardan oluşan, ancak yer gerçekliği (ground-truth) etiketlerinin analiz sürecine dahil edilmediği (unsupervised) bir sosyal ağdır. Bu çizge üzerinde uygulanan kümeleme algoritmalarının performans metrikleri Tablo 4'te, görselleştirilmiş sonuçlar ise Şekil 2'de sunulmuştur.



Şekil 2. Dolphin veri setinde kümeleme algoritmalarının sonuçları

Sonuçlar incelendiğinde, algoritmaların "matematiksel optimum" ile "sosyolojik anlamlılık" arasında bir ödünleşim (trade-off) yaşadığı görülmektedir:

Dengesiz Kesim (Trivial Solution) Problemi: Girvan-Newman, Spectral Clustering ve k-means (emb) algoritmalarının tamamı, virgülden sonraki hassasiyete kadar birebir aynı skorları (Conductance ≈ 0.045) üretmiştir. İlk bakışta bu değerler "mükemmel ayrışma" gibi görünse de görselleştirme sonuçları bu algoritmaların ağın merkezindeki yoğun yapıyı bölmek yerine, ağın çeperindeki (periphery) zayıf bağlı birkaç düğümü izole ederek dengesiz bir kesim (trivial cut) yaptığını göstermektedir. Coverage değerlerinin çok yüksek (0.96) olması, ağın büyük kısmının tek bir dev kümede toplandığını doğrulamaktadır.

Dengeli Dağılım: Kernighan-Lin algoritması, çizgeyi dengeli parçalara ayırma (bisection) kısıtlaması

nedeniyle, dengesiz kesim problemine düşmeden ağın ana omurgasını en iyi bölen yöntem olmuştur. 0.90 Kapsama ve 0.09 geçirgenlik değerleri, algoritmanın yunus popülasyonunu iki ana aile grubuna ayıran en kararlı (stable) kesimi bulduğunu işaret etmektedir.

Modülerite ve Parçalanma: Louvain algoritması (Cov ≈0.77), ağı ikili (binary) yapıdan ziyade daha küçük ve çoklu alt topluluklara ayırma eğilimi göstermiştir. Bu durum, yunus ağının içindeki daha sıkı arkadaşlık gruplarını (cliques) ortaya çıkarması açısından sosyolojik olarak anlamlıdır.

Parametre Duyarlılığı: Leiden algoritmasının Dolphin ağında oldukça düşük bir Kapsama oranı (Cov ≈0.26) sergilemesi, kullanılan çözünürlük (resolution) parametresinin bu ağın seyrek (sparse) yapısı için agresif kaldığını ve ağın aşırı parçalanmasına (over-segmentation) neden olduğunu göstermektedir.

Sonuç olarak, Dolphin ağı gibi belirgin bir kutuplaşmanın olmadığı ancak yoğun alt grupların bulunduğu ağlarda, sadece Conductance metriğine odaklanmanın yanıltıcı olabileceği; dengeli kesim yöntemlerinin (Kernighan-Lin) yapıyı daha iyi koruduğu tespit edilmiştir

Tablo 4. Dolphin çizge için performans verileri

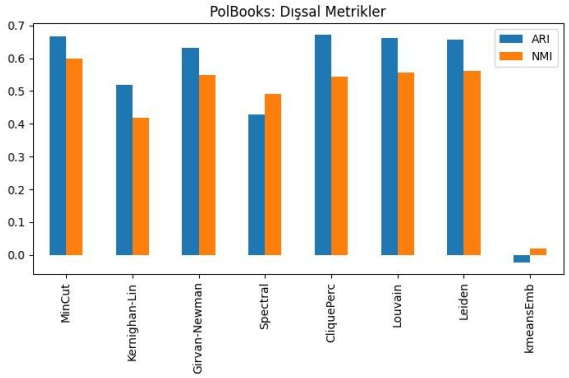
Algoritma	Coverage	Conductance
MinCut	0.993711	0.501577
Kernighan-Lin	0.905660	0.096181
Girvan-Newman	0.962264	0.045308
Spectral	0.962264	0.045308
Clique-Percolation	0.723270	0.399974
Louvain	0.773585	0.293416
Leiden	0.257862	0.768353
k-means (emb)	0.962264	0.045308

3.3. Polbooks veri seti üzerine kümeleme algoritmalarının Uygulanması

Polbooks veri seti, 2004 ABD Başkanlık seçimleri sırasında Amazon.com üzerinde siyasi kitapların satış verilerinden Valdis Krebs tarafından derlenen; 105 düğüm ve 441 kenardan oluşan orta ölçekli bir ağdır. Düğümler "Liberal", "Muhafazakâr" ve "Nötr" olmak üzere üç sınıfa ayrılmıştır. Bu ağ üzerinde uygulanan algoritmaların performans metrikleri Tablo 5'te, görselleştirilmiş kümeleme yapıları Şekil 3'te sunulmuştur.

Polbooks ağı, Dolphin ağından farklı olarak çok güçlü bir yapısal kutuplaşma (polarization) içermektedir. Bu durum, algoritmaların davranışlarını doğrudan etkilemiştir:

MinCut ve Kutuplaşma Başarısı: Dolphin ağında "aşikar çözüm" üreterek başarısız olan MinCut algoritması, Polbooks ağında en yüksek ikinci ARI skoruna (0.667) ve en iyi NMI skoruna (0.597) ulaşmıştır. MinCut'ın ürettiği çok düşük Conductance (0.0431) değeri, bu ağda Liberal ve Muhafazakâr kitaplar arasındaki bağlantının "pamuk ipliğine bağlı" olduğunu ve algoritmanın bu doğal darboğazı (bottleneck) mükemmel tespit ettiğini göstermektedir. Burada düşük conductance, trivial bir kesimi değil, gerçek kutuplaşmayı yansıtmaktadır.



Şekil 3. Polbooks veri setinde kümeleme algoritmalarının sonuçları

Tablo 5. Polbooks çizge için performans verileri

Algoritma	ARI	NMI	Coverage	Conductance
MinCut	0.667	0.597	0.9569	0.0431
Kernighan-Lin	0.518	0.418	0.9320	0.0680
Girvan-Newman	0.630	0.548	0.9501	0.0499
Spectral	0.427	0.492	0.8073	0.1927
Clique-Percolation	0.671	0.543	0.9093	0.0907
Louvain	0.660	0.556	0.8934	0.1066
Leiden	0.656	0.560	0.8889	0.1111
k-means (emb)	−0.022	0.018	0.2993	0.7007

Clique-Percolation (CPM) Performansı: En yüksek ARI skorunu (0.671) elde eden CPM, kitapların birlikte satın alınma (co-purchasing) davranışındaki yoğun alt grupları (cliques) başarıyla yakalamıştır. Bu durum, benzer görüşteki okuyucuların oluşturduğu sıkı öbeklenmelerin, global kesim yöntemlerinden ziyade yerel yoğunluk yöntemleriyle daha iyi modellendiğini göstermektedir.

Louvain ve Leiden Kararlılığı: Modülerite tabanlı bu iki algoritma, hem yüksek dışsal başarı (ARI ≈ 0.66) hem de dengeli içsel metrikler sunarak en güvenilir yöntemler olduklarını kanıtlamıştır.

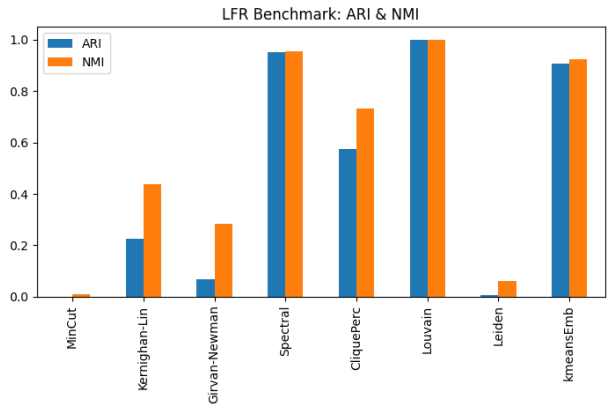
Spektral Yöntemlerde Ayrışma: Standart Spectral Clustering orta seviye bir başarı (ARI=0.427) gösterirken; k-means (emb) yaklaşımı negatif ARI değeri (-0.022) ve çok düşük Coverage (0.29) ile tamamen başarısız olmuştur. Bu negatif bulgu, 3 sınıflı (Liberal-Nötr-Muhafazakâr) ve heterojen yapılarda, basit spektral gömme tekniklerinin yetersiz kaldığını; özellikle "Nötr" sınıfının embedding uzayında diğer sınıflarla karıştığını

(non-linearly separable) ve k-means'in küresel varsayımlarının çöktüğünü işaret etmektedir.

Sonuç olarak, Polbooks gibi net kutuplaşmaya sahip ağlarda MinCut gibi kesim tabanlı algoritmalar ile CPM gibi yoğunluk tabanlı algoritmaların, genel amaçlı yöntemlerden daha iyi sonuç verebildiği görülmüştür.

3.4. LFR Benchmark veri seti üzerine kümeleme algoritmalarının Uygulanması

LFR (Lancichinetti-Fortunato-Radicchi) Benchmark, topluluk tespit algoritmalarının performansını ölçmek için literatürde standart kabul edilen sentetik bir ağ modelidir. Bu çalışmada kullanılan 250 düğümlü LFR ağı; düğüm derecelerinin ve topluluk büyüklüklerinin güç yasasına (power-law) göre dağıldığı, gürültü parametresinin (mixing parameter, μ) kontrol altında tutulduğu heterojen bir yapıya sahiptir. Algoritmaların bu kontrollü ortamdaki performans verileri **Tablo 6**'da, görsel sonuçlar ise **Şekil 4**'te sunulmuştur.



Şekil 4. LFR Benchmark veri setinde kümeleme algoritmalarının sonuçları

LFR veri seti üzerindeki sonuçlar, algoritmaların "yapısal gürültüye" ve "parametre optimizasyonuna" verdikleri tepkileri net bir şekilde ayrıştırmıştır:

Modülerite ve Global Optimizasyonun Zaferi: Louvain algoritması, ARI=1.000 ve NMI=1.000 skorlarına ulaşarak, sentetik olarak oluşturulan yer gerçekliği (Ground-Truth) yapısını hatasız bir şekilde tespit etmiştir. Benzer şekilde Spectral Clustering (ARI=0.952) ve normalize edilmiş k-means (emb) (ARI=0.908) yöntemleri de çok yüksek başarı göstermiştir. Bu durum, ağdaki toplulukların lineer olmayan uzayda (manifold) veya modülerite düzleminde net sınırlarla ayrıştığını kanıtlamaktadır.

Leiden Paradoksu ve Parametre Duyarlılığı: Karate Club verisinde başarılı olan Leiden algoritması, bu veri setinde dramatik bir performans düşüşü (ARI=0.006) yaşamıştır. Çok düşük Coverage (0.14) ve çok yüksek Conductance (0.85) değerleri, algoritmanın ağı atomize ettiğini (aşırı parçaladığını) göstermektedir. Bu başarısızlığın temel nedeni, Karate Club için optimize edilen düşük çözünürlük parametresinin (resolution=0.5), daha karmaşık ve gürültülü LFR yapısına uymamasıdır. Bu bulgu, Leiden gibi gelişmiş algoritmaların varsayılan veya statik parametrelerle her veri setinde çalışmayacağını, hiperparametre optimizasyonunun (grid search) zorunlu olduğunu ortaya koymaktadır.

Dengesşz Çözüm Tekrarı: MinCut algoritması, 0.998 gibi neredeyse tam Coverage değerine rağmen 0.001 ARI skoru üretmiştir. Bu, algoritmanın yine ağın %99'unu tek bir küme yapıp, sadece 1-2 düğümü keserek (trivial cut) "matematiksel hile" yaptığını gösterir.

Hiyerarşik Yöntemlerin Yetersizliği: Girvan-Newman (ARI=0.066) ve Kernighan-Lin (ARI=0.226) gibi klasik yöntemler, LFR ağının karmaşık topolojisi ve gürültü seviyesi karşısında yetersiz kalarak, modern spektral ve modülerite tabanlı yöntemlerin gerisinde kalmıştır.
Sonuç olarak; LFR gibi iyi tanımlanmış yapılarda Louvain ve Spectral yöntemler "altın standart" performans gösterirken, parametreleri veriye özel ayarlanmayan yöntemlerin (Leiden örneği) yanıltıcı sonuçlar doğurabileceği gözlemlenmiştir.

Tablo 6. LFR Benchmark Çizge için Performans Verileri

Algoritma	ARI	NMI	Coverage	Conductance
MinCut	0.001	0.008	0.9980	0.3333
Kernighan-Lin	0.226	0.438	0.9590	0.0427
Girvan-Newman	0.066	0.284	0.9922	0.0476
Spectral	0.952	0.956	0.9239	0.0720
Clique-Percolation	0.574	0.732	0.6666	0.5119
Louvain	1.000	1.000	0.9356	0.0618
Leiden	0.006	0.062	0.1423	0.8581
k-means (emb)	0.908	0.925	0.9025	0.0922

4. TARTIŞMA VE SONUÇ

Bu çalışmada, çizge kümeleme (graph clustering) probleminin karmaşıklığını ve algoritmik çeşitliliğini analiz etmek amacıyla; topolojik özellikleri birbirinden farklı dört veri seti (Karate Club, Dolphin Ağı, PolBooks ve LFR Benchmark) üzerinde sekiz farklı algoritmanın (MinCut, Kernighan-Lin, Girvan-Newman, Spektral, Clique Percolation, Louvain, Leiden ve k-means embedding) performansı karşılaştırmalı olarak değerlendirilmiştir.

Elde edilen deneysel bulgular, algoritmaların başarısının tek bir metrikle ölçülemeyeceğini, veri setinin yapısal karakteristiğinin (kutuplaşma, modülerite, gürültü oranı) en belirleyici faktör olduğunu ortaya koymuştur:

Global Kutuplaşma ve Spektral Yöntemler: Net bir kutuplaşmanın (polarization) bulunduğu Karate Club ağında, Spektral Kümeleme yöntemi (ARI ≈ 0.88) yerel yöntemlere kıyasla daha başarılı olmuştur. Bu durum,

çizgenin Laplacian özvektörlerine dayalı global optimizasyonun, ağdaki cebirsel kopuş noktalarını (algebraic connectivity) tespit etmede üstün olduğunu göstermektedir.

Dengesiz Kesim (Trivial Solution) Riski: Yer gerçekliği etiketinin bulunmadığı Dolphin ağında, MinCut ve Spektral gibi yöntemlerin çok düşük Conductance değerlerine ulaşmasına rağmen, görsel analizde ağın sadece uç noktalarını (outliers) kestiği ve "dengesiz kesime çözüm" ürettiği tespit edilmiştir. Buna karşılık, dengeli bölme (partitioning) kısıtlamasına sahip Kernighan-Lin algoritması (Cov ≈ 0.91), ağın sosyolojik yapısını en iyi yansıtan yöntem olarak öne çıkmıştır.

Yoğun Alt Yapıların Tespiti: Siyasi kitaplardan oluşan PolBooks ağında, Clique-Percolation (ARI ≈ 0.67) ve Louvain/Leiden (ARI ≈ 0.66) algoritmaları, benzer görüşteki sıkı öbeklenmeleri (dense subgraphs) yakalamada başarılı olmuştur.

Parametre Duyarlılığı ve Modülerite: Kontrollü bir deney ortamı sunan sentetik LFR Benchmark veri setinde, Louvain algoritması (ARI=1.0) kusursuz bir performans sergilerken; parametreleri Karate Club ağına göre (düşük çözünürlük) ayarlanan Leiden algoritmasının performansı dramatik şekilde düşmüştür (ARI ≈0.006). Bu bulgu, modern algoritmaların varsayılan ayarlarla her veride çalışmayacağını ve hiperparametre optimizasyonunun (grid search) kritik önemini kanıtlamaktadır.

Çalışmanın en özgün bulgularından biri, spektral gömme (embedding) tabanlı k-means yaklaşımında yapılan iyileştirmedir. Ham özvektörler üzerinde başarısız olan k-means algoritması, düğüm vektörlerinin L_2 normu ile birim hiperküreye normalize edilmesi sonucunda, Spektral Kümeleme algoritması ile eşdeğer (ARI ≈0.88) ve yüksek performansa ulaşmıştır. Bu sonuç, vektör uzayındaki ayrışmanın Öklid mesafesinden ziyade açısal (cosine) yakınlığa dayandığını deneysel olarak doğrulamaktadır.

Sonuç olarak bu çalışma; ağ analizinde "tek bir en iyi algoritma" olmadığını, algoritma seçiminin verinin büyüklüğüne, etiket bilgisinin varlığına ve aranılan yapının (dengeli kesim vs. yoğun topluluk) niteliğine göre yapılması gerektiğini göstermektedir. Gelecek çalışmalarda, sadece topolojik yapıyı değil, düğüm özniteliklerini de öğrenme sürecine dahil eden Çizge Sinir Ağları (Graph Neural Networks- GNN) tabanlı yöntemlerin bu veri setleri üzerindeki başarısının incelenmesi hedeflenmektedir.

Etik Hususlar

Etik kurallara uyum

Yazar olarak insan gönüllüleri ve deneysel hayvan içeren çalışmalarda gerçekleştirilen tüm prosedürleri, kurumsal ve / veya ulusal araştırma komitesinin etik standartlarına ve 1964 Helsinki deklarasyonuna ve daha sonraki değişikliklerine veya karşılaştırılabilir etik standartlara uygun çalıştığımızı deklare ederiz.

Finansman

Bu çalışma için herhangi bir finansal destek alınmamıştır.

Çıkar çatışması

Yazar çalışma ile ilgili herhangi bir kurum ya da kişilerle çıkar çatışmasının olmadığını beyan eder.

KAYNAKÇA

[1] Newman, M. (03 2010). *Networks: An Introduction*. doi:10.1093/acprof:oso/9780199206650.001.0001

[2] Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. Proceedings of the National Academy of Sciences, 99(12), 7821-7826. https://doi.org/10.1073/pnas.122653799

[3] Fortunato, S. (2010). Community detection in graphs. Physics Reports, 486(3-5), 75-174. https://doi.org/10.1016/j.physrep.2009.11.002

[4] Newman, M. E. J. (2006). Modularity and community structure in networks. Proceedings of the National Academy of Sciences, 103(23), 8577-8582. https://doi.org/10.1073/pnas.0601602103

[5] Yang, J., Leskovec, J. (2016). Defining and evaluating network communities based on ground-truth. Knowledge and Information Systems, 42(1), 181–213. https://doi.org/10.1007/s10115-013-0670-5

[6] Lancichinetti, A., & Fortunato, S. (2009). Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Physical Review E, 80(1), 016118. https://doi.org/10.1103/PhysRevE.80.016118

[7] Fortunato, S., & Hric, D. (2016). Community detection in networks: A user guide. Physics Reports, 659, 1-44. https://doi.org/10.1016/j.physrep.2016.09.002

[8] Watteau, T., Bonnefoy, A., Illouz-Laurent, S., Jusseau, J., & Iovleff, S. (2024). Advanced Graph Clustering Methods: A Comprehensive and In-Depth Analysis. arXiv preprint arXiv:2407.09055.

[9] Shi, L., & Chen, B. (2020). Comparison and benchmark of graph clustering algorithms. arXiv preprint arXiv:2005.04806.

[10] Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. D. F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PloS one*, *14*(1), e0210236.

[11] Liu, Y., Shah, N., & Koutra, D. (2015). An empirical comparison of the summarization power of graph clustering methods. *arXiv preprint arXiv:1511.06820*.

[12] Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*(1), 193–218. https://doi.org/10.1007/BF01908075

[13] Strehl, A., & Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and

Data Mining (pp. 186–195). https://doi.org/10.1145/775047.775067

[14] Emmons, S., Kobourov, S., Gallant, M., & Borner, K. (05 2016). Analysis of Network Clustering Algorithms and Cluster Quality Metrics at Scale. *PLOS ONE*, *11*. doi: 10.1371/journal.pone.0159161

[15] Aggarwal, C., & Wang, H. (01 2010). Managing and Mining Graph Data (Vol. 40). doi:10.1007/978-1-4419-6045-0

[16] Kernighan, B. W., & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, *49*(2), 291–307. doi:10.1002/j.1538-7305.1970.tb01770.x

[17] Von Luxburg, U. V. (2007). *A tutorial on spectral clustering*. *Statistics and Computing, 17*(4), 395–416. https://doi.org/10.1007/s11222-007-9033-z

[18] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008*(10), P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

[19] Traag, V. A., Waltman, L., & van Eck, N. J. (2019). *From Louvain to Leiden: Guaranteeing well-connected communities*. *Scientific Reports*, 9,5223. https://doi.org/10.1038/s41598-019-41695-z

[20] MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 281–297). University of California Press.