

Karmaşık Ağların Sınıflandırma Sorusuna Yönelik Düşük-Dereceli Temsiliyeti Low-Rank Representations Towards Classification Problem of Complex Networks

Murat Çelik
Bilgisayar Mühendisliği Bölümü
Hacettepe Üniversitesi
Ankara, Türkiye
b21827263@cs.hacettepe.edu.tr

Ali Baran Taşdemir
Bilgisayar Mühendisliği Bölümü
Hacettepe Üniversitesi
Ankara, Türkiye
alibaran@tasdemir.us

Lale Özkahya
Bilgisayar Mühendisliği Bölümü
Hacettepe Üniversitesi
Ankara, Türkiye
ozkahya@cs.hacettepe.edu.tr

Özetçe —Karmaşık ağlar, sosyal ilişkiler, beyin aktiviteleri, moleküler yapıları temsil edip, çizge olarak özelliklerinin anlaşılabilirliği ve tahmin yürütülebilirliği için kapsamlı olarak incelenmiştir. Bu ağların modelleri ve onlara yönelik algoritmalar arama motorları, tavsiye sistemleri gibi günlük uygulamalarda kullanılmaktadır. Genelde, bu ağlar düğümlerinin düşük-boyutlu Öklidyen gömülmesi işlemiyle inşa edilerek modellenmektedir. Burada düğümlerin Öklid uzayında aralarındaki mesafe, her çiftin arasında bir bağlantı olma ihtimaline dair bilgi verir. Bu çalışmada, bu düşük-dereceli temsil yöntemlerinin gerçek hayat ağlarının sınıflandırma sorusuna yönelik performansı incelenmiştir.

Anahtar Kelimeler—Çizge gömmesi, çizge temsili, düşük-boyutlu gömme, düşük-dereceli temsil.

Abstract—Complex networks representing social interactions, brain activities, molecular structures have been studied widely to be able to understand and predict their characteristics as graphs. Models and algorithms for these networks are used in real-life applications, such as search engines, and recommender systems. In general, such networks are modelled by constructing a low-dimensional Euclidean embedding of the vertices of the network, where proximity of the vertices in the Euclidean space hints the likelihood of an edge (link). In this work, we study the performance of such low-rank representations of real-life networks on a network classification problem.

Keywords—Graph embeddings, graph representations, low-dimensional embedding, low-rank representation.

I. GİRİŞ

Karmaşık ağlar, günümüzde, sosyal ilişkiler, beyin aktiviteleri, moleküler yapıları temsil edip, çizge olarak özelliklerinin anlaşılabilirliği ve tahmin yürütülebilirliği için kapsamlı olarak incelenmiştir. Karmaşık ağlar için iyi modeller oluşturulması, sosyal ağların da modern insan etkileşimini anlamada rolü düşünüldüğünde, oldukça önemli bir konudur [1]–[3]. Son yıllarda, çizgenin düğümlerinin Öklid uzayında düşük-boyutlu gömmelerinin elde edilmesi makine öğrenmesine yönelik uygulamalarda sıklıkla çalışılmış ve bu şekilde, gömmenin geometrisi sayesinde çizgeye dair yapısal özelliklerin korunması

denenmiştir [4]. Özel olarak, düğüm gömme (node embedding) yöntemi girdi olarak n düğümlü bir çizgedeki her düğümü $x \in \mathbb{R}^k$ ve $k \ll n$ olacak şekilde bir vektör ile temsil eder. Bu şekilde gömme yöntemi makine öğrenmesi ile beraber öbekleştirme (clustering), sınıflandırma, bağlantı tahmini (link prediction) gibi birçok konuda kullanılmaktadır [4]. Bu anlamda, iki düğümün arasında kenar (edge/ link) olma olasılığı, onları temsil eden vektörlerin geometrik yakınlıklarından çıkarılabilmektedir.

Bu gömme yöntemlerinden yaygın kullanılan ikisi TSVD (Truncated Singular Value Decomposition) ve PCA (Logistic Principal Component Analysis) gibi verilen ağın (ya da çizgenin) komşuluk matrisini parçalama (decomposition) yöntemleridir [5]. Son yıllarda, derin sinir ağları kullanılarak da çizge gömmesini öğrenme yöntemi sıklıkla kullanılmıştır [6]–[9]. Bu yöntemlerin çözmeye çalıştığı soru, verilen büyük bir ağın bir yandan yapısal özelliklerini koruyarak orijinalinden daha küçük boyutlarda matrisler ile temsil edilmesini sağlamaktır. Hala tam cevaplanmamış sorulardan birisi, bu temsiliyetin ne kadar küçük bir matris ile ağın yapısal bilgilerini koruyarak yapılabileceğidir [10]–[13]. Örneğin, üçgenlerin sıklıkla görülmesi Facebook ağı gibi sosyal ağlarda, önemli bir karakteristik olup, bunun TSVD ve PCA gibi çizge gömme teknikleriyle ne kadar korunabileceği araştırılmış [10], [11], [14], ikili (binary) düğüm sınıflandırma sorusu, düğümlerin buldukları topluluk etiketleri çizge karakteristiği olarak alınarak uygulanmıştır.

Buradaki çalışmanın temel katkısı, yaygın kullanılan TSVD ve LPCA çizge gömme tekniklerinin orijinal çizgenin birçok sınıflı (multiclass) sınıflandırma problemindeki performansını incelemektir. Çizge sınıflandırma sorusu, birçok farklı alanda çalışılan bir problem olup, bazı durumlarda yeterli ağ bilgisi olmadan sınıf bilgisinin çıkarılmaya çalışılması gibi zorluklar taşır. Bonner vd. [15] çizge sınıflandırması için, ağın topoloji özneliklerini derin sinir ağlarını kullanarak bir öğrenme yöntemi geliştirmiştir. Rastsal çizge modellerinin çeşitliliği açısından kapsamlı olan çalışmalardan birisi Rossi ve Ahmed [16] tarafından yapılmıştır. Bu çalışmada da, rastsal ağ modelleri ile inşa edilen ağ grupları olmak üzere belirgin olarak farklı yapılara sahip 9 ağ sınıfı sınıflandırma sorusuna dahil edilmiştir. Çizge gömme yöntemleri ile düşük boyutlarda temsil

TABLO I: Sınıflandırma işleminde kullanılan özniteliklerin her ağ grubu için ortalama değerleri.

Veri Kümesi	Düğüm Sayısı	Kenar Sayısı	Yoğunluk	Maks. Derece	Ort. Derece	Maks. k-Çekirdek	Ort. Kümelenme Kat sayısı	Üçgen Sayısı	Ort. Özvektör Merkeziliği
Barabasi-Albert	1000	35409	0.0709	278	71	40	0.127	767320	0.0247
Biyoloji	2715	27895	0.0098	339	20	37	0.256	272918	0.0096
Beyin	487	12561	0.2358	245	48	35	0.520	398615	0.0610
Chung-Lu	5201	21437	0.0045	167	8	16	0.072	9449	0.0105
Ekonomi	2283	100090	0.1574	863	120	111	0.437	5518546	0.0255
Enzim	60	94	0.0948	6	4	3	0.287	21	0.0989
Erdos-Renyi	1000	37583	0.0752	99	75	62	0.075	410869	0.0310
Facebook	6314	231472	0.0138	987	76	64	0.258	1910524	0.0086
Retweet	5586	6396	0.0005	1972	2	4	0.013	236	0.0055

edilecek şekilde yeniden inşa edilen (reconstruct) ağların, sınıf özelliklerini büyük oranda koruduğu görülmüştür. Deneyler sonucunda, LPCA yönteminin TSVD yöntemine göre daha iyi performans verdiği görülmüştür. Rastsal ağların başarılı olarak temsil edilebilmesi için gerçek dünya ağlarından çok daha büyük dereceli temsillere ihtiyaç duyduğu gözlemlenmiştir.

II. METOT

Sınıflandırma işleminde, biyoloji, ekonomi, sosyal ağlar gibi farklı gerçek ağ grupları yanında rastsal yöntemlerle üretilen farklı ağ grupları, sınıflandırma işleminin sınıflarını teşkil eder. Çizge gömme yöntemleri ile verilen orijinal ağların daha düşük boyutta çizgeler tarafından temsilinin sınıf özelliğini ne kadar koruduğu, TSVD ve LPCA yöntemlerinin performansları üzerinden karşılaştırılmıştır. Sınıflandırma işlemi, çizgelere yönelik önemli rol taşıdığı bilinen çizge öznitelikleri kullanılarak yapılmıştır.

A. Çizge Gömme Yöntemleri

a) TSVD: TSVD (Kesik Tekil Değer Çözümü - Truncated Singular Value Decomposition), tekil değer çözümü (SVD) kullanılarak komşuluk matrisinin (adjacency matrix) yakınsandığı bir yöntemdir. TSVD yöntemi, komşuluk matrisini yeniden inşa etmek için kullanılmıştır. Bir çizge için komşuluk matrisi $A \in \{0, 1\}^{N \times N}$, ortonormal matris $Z \in \mathbb{R}^{N \times k}$ olsun. Z ortonormal matrisinin sütunları, en büyük değerli k özdeğerine karşılık gelen komşuluk matrisi özvektörlerini içerir. W ise elemanları en büyük k özdeğere karşılık gelen köşegen matrisidir. TSVD gömmeleri (embeddings) X ve Y , $X = Zs(W)\sqrt{|W|}$ ve $Y = Z\sqrt{|W|}$ ile hesaplanır. $s()$ fonksiyonu, işaret fonksiyonunu temsil etmektedir. Son olarak, yeniden inşa edilmiş komşuluk matrisinin hesaplanması için $\sigma(XY^T)$ uygulanmıştır. Bu işlemde σ fonksiyonu, değerleri 0 ve 1 arasında sınırlayan bir kırpma fonksiyonudur (clipping function).

b) LPCA: Komşuluk matrisine yakınsamak için LPCA'ya (Lojistik Ana Bileşenler Analizi – Logistic Principal Component Analysis) dayalı bir yöntem kullanılmıştır. Bir kayıp (loss) fonksiyonu yardımıyla $\sigma(XY^T)$ (yakınsama) ve A (komşuluk matrisi) arasındaki hata en aza indirilmiştir. $A \in \{0, 1\}^{N \times N}$, ve $X, Y \in \mathbb{R}^{N \times k}$ değerleri için kayıp fonksiyonu:

$$L = \sum_{i=1}^N \sum_{j=1}^N -\log l\left(\tilde{A}_{i,j} [XY^T]_{i,j}\right)$$

$l(x) = (1 + e^{-x})^{-1}$ lojistik fonksiyonunu (sigmoid) ve \tilde{A} kaydırılmış komşuluk matrisini temsil etmektedir. Komşuluk matrisi A üzerindeki 0 değerleri -1 ile değiştirilmiştir ($\tilde{A} = 2A - 1$). X ve Y matrisleri $[-1, 1]$ arasında rastgele değerler ile başlatılmıştır. SciPy kütüphanesi [17] ile "L-BFGS-B" [18] algoritması kullanılarak 100 iterasyon ile kayıp fonksiyonunu en aza indiren X ve Y gömmeleri (embeddings) elde edilmiştir. Son olarak $\sigma(XY^T)$ ile komşuluk matrisi yeniden inşa edilmiştir.

Bu iki çizge gömme yönteminde de, işlemin derecesi (rank) elde edilen matrislerin boyutunda görülen k değeridir. Bu derece tanımının, aşağıdaki çizgelerin düğümünün derecesi tanımıyla karışmamasına anlatımda gereken dikkat gösterilmiştir.

B. Çizge Öznitelikleri ve Modelleri

Kullanılan öznitelikler, Tablo I'de de görülen yoğunluk, maksimum derece, ortalama derece, maksimum k -çekirdek, ortalama kümelenme katsayısı, toplam üçgen sayısı, ortalama özvektör merkeziliğidir. Aşağıda, bu çizge özniteliklerinin tanımları verilmektedir. Bir çizge $G = (V, E)$ şeklinde V düğüm kümesi ve E kenar kümesi ile tanımlanır. Bu kümelerin eleman sayılarını $n = |V|$ ve $m = |E|$ parametreleri temsil edecektir. Her düğümün bağlantısı olan komşu düğümlerin sayısı, o düğümün derecesi (degree) olup, maksimum, minimum ve ortalama (n üzerinden) değerleri çizgeler üzerine tanımlanan renklendirme, eşleştirme problemleri gibi alanlarda önemli rol oynayan niteliklerdendir. Algoritmik çalışma süresini etkileyen önemli çizge parametrelerinden biri olan çizge yoğunluğunun değeri $\frac{m}{\binom{n}{2}}$ şeklinde hesaplanır. Bu bağlamda, bir çizgenin n^2 seviyesinde kenar içermesi yoğun, aksi halde de seyrek, olduğu anlamına gelir.

Bir çizgenin kümelenme katsayısı (clustering coefficient), çizgedeki düğümlerin kümelenme eğilimini ölçer [3]. Özvektör merkeziliği bir düğümün diğer düğümler üzerindeki etkisini ölçer. Bir düğümün özvektör merkeziliği, komşuluk matrisinin en büyük özdeğerine ait özvektöründeki bu düğüme karşılık gelen hanenin değeridir. Eğer bir düğümün derecesi yüksekse, bu düğümün özvektör merkeziliği yüksek olacaktır. Diğer bir önemli ağ parametresi de çizgenin k -çekirdeğidir [19], [20]. Bir çizgenin k -çekirdeği, içerisindeki her düğümün en az k komşusunu da içeren olabilecek en büyük altçizgen düğüm sayısıdır.

C. Sınıflandırma İşlemi

Makine Öğrenmesi modeli olarak Rastsal Orman (RF), Lineer Model Olasılıksal Dereceli Azalma (SGD), K-En Yakın

Komşu (KNN) ve Destek Vektör Makineleri (SVM) Sınıflandırıcıları kullanılmıştır. Eğitim aşamasında tabakalı, 10 katlamalı çapraz doğrulama ile F1 skorları elde edilmiştir, ve bu skorlar yardımıyla optimizasyon çalışması yapılmıştır. *F1 skoru*, kesinlik (precision) ve hassasiyet (recall) değerlerinin harmonik ortalaması alınarak hesaplanan istatistiksel bir ölçüttür. Bu skorlar, Rastsal Orman (RF), Lineer Model Olasılıksal Dereceli Azalma (SGD), K-En Yakın Komşu (KNN) ve Destek Vektör Makineleri (SVM) için sırasıyla 0.87, 0.77, 0.86, ve 0.89 olarak gözlemlenmiştir. Eğitim veri setinin 20%'si test veri seti olarak belirlenerek optimize edilmiş modeller üzerinde ikinci bir F1 skoru elde edilmiştir. Sonuçlar arasında 0.92'lik bir F1 skoru elde edilen Destek Vektör Makineleri (SVM), sınıflandırma için en uygun model olarak gözlemlenmiştir. Bu sebeple, ilerleyen bölümlerde çizge gömme yöntemlerinin performansları SVM ile elde edilen F1 skor sonuçları üzerinden karşılaştırılmıştır.

III. DENEYSEL SONUÇLAR

Deneyler için 9 farklı sınıftan toplam 198 çizge ile çalışılmıştır [21]. Bu ağ grupları ve eleman sayıları: Biyoloji Ağları (bio, 30), Beyin Ağları (bn, 10), Ekonomi Ağları (econ, 16), Retweet Ağları (rt, 29), Facebook Ağları (fb, 30), Enzim Ağları (enz, 30), Barabási–Albert Ağları (ba, 13), Chung-Lu Ağları (cl, 24), Erdős–Rényi Ağları (er, 16) olup, ağların öznitelikleri ortalama değer olarak Tablo-I'de verilmiştir. Bu ağ gruplarından son üçü, ismiyle anılan rastsal ağ modelleri ile oluşturulmuş olup, farklı ağ yapılarını temsil etmekte kullanılırlar.

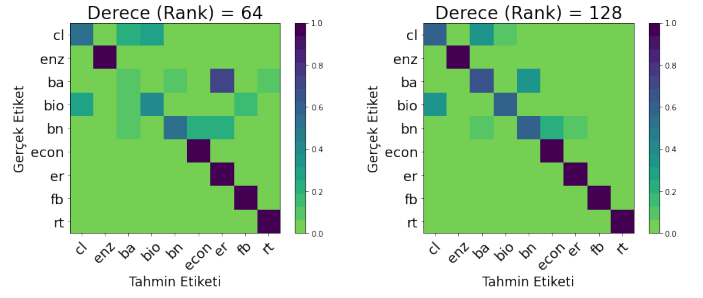
TABLE II: SVM sınıflandırıcısı ile bulunan F1-skorlarının, çizge gömme yöntemlerinin farklı dereceleri (rank) kullanıldığında, tüm ağ grupları üzerinden alınan ortalama değerleri.

TSVD		LPCA	
Derece	F1 skoru	Derece	F1 skoru
16	0.48	5	0.51
32	0.52	16	0.82
64	0.70	32	0.92
128	0.81	64	0.97

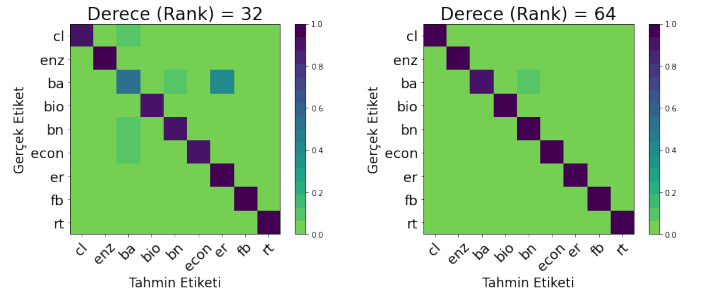
Rastsal ağ modellerinin aralarındaki temel ayrım, iki düğüm arasına eklenen kenarın olasılığının modele göre farklılık göstermesidir. BA modelinde [1], her adımda bir düğüm eklenirken, yeni kenarların her biri, önceden varolan düğümlerin dereceleriyle doğru orantılı olarak belirlenir. ER modelinde [22], düğüm ikililerinin sayısı $\binom{n}{2}$ olup, her çift belirlenmiş sabit bir p olasılığıyla kenara dönüşür. CL modeli [23], ER modelini eşit olmayan kenar olasılıklarıyla, verilmiş bir düğüm derece (komşu sayısı) dağılımı beklentisine uyacak şekilde geliştirir.

A. Sınıflandırma Performansları

Eğitim aşamasında SVM en yüksek F1 skorunu sağladığından, test aşaması da bu sınıflandırıcının değerleri performans açısından referans alınmıştır. Tüm ağlar üzerinden ortalama F1 skoru karşılaştırıldığında, Tablo II'de görülen değerler sırasıyla TSVD (derece= 64 ve 128 için) 0.70 ve 0.81, LPCA (derece= 64 ve 128 için) 0.92 ve 0.97 bulunmuştur. Bu açıdan LPCA TSVD'den aynı derece (rank) için daha yüksek kesinlik (accuracy) sağlamaktadır.



Şekil 1: TSVD yöntemiyle elde edilen ve tüm ağ grupları üzerinden sınıflandırma kesinliğini (accuracy) değerlendiren hata matrisi.



Şekil 2: LPCA yöntemiyle elde edilen ve tüm ağ grupları üzerinden sınıflandırma kesinliğini (accuracy) değerlendiren hata matrisi.

Şekil 1 ve 2'de görülen hata matrisleri karşılaştırıldığında, aynı şekilde LPCA'nın sınıflandırma kesinliği performansının TSVD'den daha üstün olduğu görülmektedir. Ek olarak, BA, CL gibi ağlarda sınıflandırma kesinliği düşüktür. Özellikle, LPCA'nın BA grubu için performansı diğer gruplara bakıldığında daha zayıf kalmaktadır. Bu anlamda, sınıflandırma sorusu üzerinde, bu yöntemlerin gerçek hayat ağları için gösterdikleri başarıyı rastsal ağlarda sağlayamadığı sonucu çıkmıştır.

TABLE III: Test edilen ağ grupları için 92% F1 skoruyla başarılı sınıflandırma sağlayan (minimum) parçalanma boyutunun, LPCA yöntemi kullanıldığında elde edilen değerleri. (Ortalama derece, verilen çizgelerdeki düğümlerin ortalama derecesini belirtir.)

Veri Kümesi	Düğüm Sayısı	Ortalama Derece	Parçalanma Boyutu
Biyoloji	2715	19.58	5
Enzim	60	3.47	5
Retweet	5585	2.34	5
Beyin	487	47.50	16
Chung-Lu	5201	7.96	16
Ekonomi	2283	120.02	16
Erdos-Renyi	1000	75.17	32
Facebook	6314	76.10	32
Barabasi-Albert	1000	70.82	64

Tablo III'de, test edilen ağ grupları için, gruptaki tüm ağların 92% F1 skoruyla sınıflandırılmasını sağlayan (minimum) parçalanma boyutunun, LPCA yöntemi kullanıldığında elde edilen değerleri listelenmiştir. Bu değerlere bakıldığında, LPCA yönteminin Biyoloji, Enzim ve Retweet ağları için

oldukça küçük bir derece (rank) değeriyle çizge gömme işleminde başarılı sonuçlar verdiği gözlemlenmektedir. Bu ağlar, Tablo I’de de görüldüğü üzere yoğunluğu ve ortalama derecesi düşük ağlardır. Genel olarak rastsal ağlarda (BA, CL, ER) aynı seviyede kesinlik değerlerine ulaşmak için daha yüksek derece (rank) gerektiği görülmektedir.

Hesaplama Zamanı: Karşılaştırma amaçlı Ekonomi ve BA rastsal ağ grubundan birer ağ seçilerek LPCA ve TSVD yöntemlerinin hesaplama hızı Tablo IV’de karşılaştırılmıştır. TSVD için hesaplama süresinin LPCA’dan çok daha kısa sürdüğü görülmüştür. LPCA için harcanan bu sürenin, iterasyon sayısının artması halinde daha da artacağı öngörülmektedir. Burada verilen iki örnek ağ dışındaki büyük ağlar için aynı fark görülmekte ve TSVD’yi hesaplama zamanı açısından tercih edilir kılmaktadır.

TABLO IV: İki yöntemin hesaplama sürelerinin (saniye olarak) econ-beause ($n = 0.5K$, $m = 44K$) ve ba_1k_150k ($n = 1K$, $m = 150K$) ağları üzerinde örnek amaçlı karşılaştırılması.

Derece	econ-beause		ba_1k_150k	
	TSVD	LPCA	TSVD	LPCA
16	0.40	3.54	1.75	15.98
32	0.44	3.86	1.65	14.33
64	0.36	2.60	1.70	15.19

IV. SONUÇLAR

Bu çalışmada, çizge gömme yöntemleri ile düşük boyutlarda temsil edilecek şekilde yeniden inşa edilen ağların, sınır özelliklerini büyük oranda koruduğu görülmüştür. Öte yandan, benzeri çalışmalarda, ağların analizinde önem taşıyan üçgen dağılımı gibi bazı özelliklerin burada sunulan yöntemlerle yeterli kadar korunmadığı görülmektedir. Bu açıdan, sunulan yöntemler yeniden inşa işleminde farklı çizge özelliklerinin korunmasına yönelik gelişime ihtiyaç duymaktadır.

BİLGİLENDİRME

Bu çalışma, TÜBİTAK 118E283 no’lu proje kapsamında desteklenmiştir.

KAYNAKLAR

- [1] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [2] D. Chakrabarti and C. Faloutsos, “Graph mining: Laws, generators, and algorithms,” *ACM computing surveys (CSUR)*, vol. 38, no. 1, pp. 2–es, 2006.
- [3] D. J. Watts and S. H. Strogatz, “Collective dynamics of ‘small-world’ networks,” *Nature*, vol. 393, no. 6684, p. 440, 1998.
- [4] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Advances in neural information processing systems*, vol. 30, 2017.
- [5] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, “Distributed large-scale natural graph factorization,” in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 37–48.
- [6] S. Cao, W. Lu, and Q. Xu, “Deep neural networks for learning graph representations,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.
- [7] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

- [8] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.
- [9] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “Line: Large-scale information network embedding,” in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.
- [10] C. Seshadhri, A. Sharma, A. Stolman, and A. Goel, “The impossibility of low-rank representations for triangle-rich complex networks,” *Proceedings of the National Academy of Sciences*, vol. 117, no. 11, pp. 5631–5637, 2020.
- [11] S. Chanpuriya, C. Musco, K. Sotiropoulos, and C. Tsourakakis, “Node embeddings and exact low-rank representations of complex networks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 185–13 198, 2020.
- [12] A. Loukas, “What graph neural networks cannot learn: depth vs width,” *arXiv preprint arXiv:1907.03199*, 2019.
- [13] V. Garg, S. Jegelka, and T. Jaakkola, “Generalization and representational limits of graph neural networks,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 3419–3430.
- [14] A. Stolman, C. Levy, C. Seshadhri, and A. Sharma, “Classic graph structural features outperform factorization-based graph embedding methods on community labeling,” *arXiv preprint arXiv:2201.08481*, 2022.
- [15] S. Bonner, J. Brennan, G. Theodoropoulos, I. Kureshi, and A. S. McGough, “Deep topology classification: A new approach for massive graph classification,” in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 3290–3297.
- [16] R. A. Rossi and N. K. Ahmed, “Complex networks are structurally distinguishable by domain,” *Social Network Analysis and Mining*, vol. 9, no. 1, p. 51, 2019.
- [17] E. Jones, T. Oliphant, P. Peterson *et al.*, “Scipy: Open source scientific tools for python,” 2001.
- [18] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A limited memory algorithm for bound constrained optimization,” *SIAM Journal on scientific computing*, vol. 16, no. 5, pp. 1190–1208, 1995.
- [19] A. E. Saryüce, C. Seshadhri, and A. Pinar, “Parallel local algorithms for core, truss, and nucleus decompositions,” *arXiv.org e-Print archive*, <https://arxiv.org/abs/1704.00386>, 2017.
- [20] P. Liu and A. E. Saryüce, “Analysis of core and truss decompositions on real-world networks,” *MLG workshop at Knowledge Discovery and Data Mining Conference, Anchorage, AK, USA*, 2019.
- [21] R. A. Rossi and N. K. Ahmed, “The network data repository with interactive graph analytics and visualization,” in *AAAI*, 2015. [Online]. Available: <http://networkrepository.com>
- [22] P. Erdős and A. Rényi, “On random graphs i,” *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.
- [23] F. Chung and L. Lu, “The average distances in random graphs with given expected degrees,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 25, pp. 15 879–15 882, 2002.