

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AĞ BAĞLANTILI VERİLER İÇİN GÖZETİMSİZ İKİLİ
ÖZNİTELİK OLUŞTURMA YÖNTEMİ

ARZU KAKIŞIM
DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2019

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

AĞ BAĞLANTILI VERİLER İÇİN
GÖZETİMSİZ İKİLİ ÖZNİTELİK
OLUŞTURMA YÖNTEMİ

ARZU KAKIŞIM
DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
PROF. DR. İBRAHİM SOĞUKPINAR

GEBZE
2019

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**UNSUPERVISED BİNARY FEATURE
CONSTRUCTION METHOD FOR
NETWORKED DATA**

ARZU KAKIŞIM
**A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY**
DEPARTMENT OF COMPUTER ENGINEERING

THESIS SUPERVISOR
PROF. DR. İBRAHİM SOĞUKPINAR

GEBZE
2019

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 03/07/2019 tarih ve 2019/30 sayılı kararıyla oluşturulan jüri tarafından 23/07/2019 tarihinde tez savunma sınavı yapılan Arzu KAKIŞIM'ın tez çalışması Bilgisayar Mühendisliği Anabilim Dalında DOKTORA tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI)

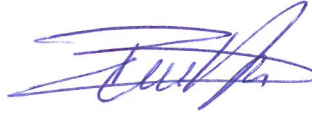
: Prof. Dr. İbrahim SOĞUKPINAR



ÜYE : Prof. Dr. Nevcihan DURU



ÜYE : Prof. Dr. Banu DİRİ



ÜYE : Dr. Öğr. Üyesi Yakup GENÇ



ÜYE : Dr. Öğr. Üyesi Burcu YILMAZ



ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve/..... sayılı kararı.

ÖZET

Ağ bağlantılı veriler, ağ nesnelerinden ve bağlantılardan oluşan verilerdir. Ağ nesneleri, yüksek boyutlu öznitelikler ve bu nesneler arasındaki ilişkileri gösteren bağlantılar ile karakterize edilir. Fakat, geleneksel öznitelik seçim ve öznitelik çıkarım yöntemleri yalnızca bu ağlara ait öznitelik bilgisini dikkate almakta ve ağ bağlantılı verilere ait bağlantı bilgisini göz ardı etmektedirler. Bu tez çalışmasında, ağ verilerine ait bağlantı ve öznitelik bilgisi birlikte kullanılarak, ağ verileri için yeni bir öznitelik çıkarım yöntemi geliştirilmiştir. Ağın bağlantı bilgisinden yararlanarak her ağ nesnesi için öznitelikleri yeniden yapılandıran NetBFC olarak adlandırdığımız bir deneimsiz ikili öznitelik çıkarım yöntemi geliştirilmiştir. Önerilen yöntem, ağdaki benzer nesneleri birbirleri ile ilişkilendirerek, aynı grupta olma olasılığı yüksek nesneler arasındaki benzerlikleri artırır. Yöntem, ağa bağlı verilerin seyrekliğinden kaynaklanan zorluğun üstesinden gelmek için ağ nesnelerini, kendisine benzerlik gösteren diğer ağ nesneleri ile gruplandırarak ve bu grup içerisinde sıklıkla gözlemlenen nitelikler ile ilişkilendirerek her bir nesne için yerel öznitelik zenginleştirmesini ve yerel öznitelik seçimini mümkün kılar. Ek olarak, önerilen yöntem, kümeleme algoritmalarının performansını düşüren alakasız ve gereksiz nitelikleri ortadan kaldırmak için bir öznitelik eleme aşaması uygular. Gerçek dünyada verileri üzerinde gerçekleştirilen deneyler sonucunda, bu tez çalışması kapsamında önerilen yöntemin literatürde var olan diğer temel yöntemler ile kıyaslandığında ağ bağlantılı veriler için daha iyi kümeleme performansı sağladığı gözlemlenmektedir.

Anahtar Kelimeler: Öznitelik Çıkarımı, Öznitelik Seçimi, Bağlantı Oluşturma, Ağ Verisi, Sosyal Ağlar.

SUMMARY

Networked data is data composed of network objects and links. Network objects are characterized by high dimensional attributes and by links indicating the relationships among these objects. However, traditional feature selection and feature extraction methods consider only attribute information, thus ignoring link information. In the presented work, we propose a new unsupervised binary feature construction method (NetBFC) for networked data that reconstructs attributes for each object by exploiting link information. By exploring similar objects in the network and associating them, our method increases the similarities between objects with high probability of being in the same group. The proposed method enables local attribute enrichment and local attribute selection for each object by aggregating the attributes of similar objects in order to deal with the sparsity of networked data. In addition, this method applies an attribute elimination phase to eliminate irrelevant and redundant attributes which decrease the performance of clustering algorithms. Experimental results on real-world data sets indicate that NetBFC significantly achieves better performance when compared to baseline methods.

Key Words: Feature Extraction, Feature Selection, Link Reconstruction, Social Network, Networked Data.

TEŞEKKÜR

Doktora eğitimim boyunca bana destek vererek, tez yazımı sürecini tamamlamamda emeği geçen başta değerli tez danışmanım Prof. Dr. İbrahim SOĞUKPINAR olmak üzere tez izleme jürimde yer alarak bilgi birikimini benimle paylaşan Dr. Yakup GENÇ ve Prof. Dr. Nevcihan DURU'ya şükranlarımı sunarım. Tez savunma jürimde yer alan Prof. Dr. Banu DİRİ ve Dr. Burcu YILMAZ'a göstermiş oldukları desteklerden dolayı en içten teşekkürlerimi sunarım. Bu çalışmayı, bütün çalışmam boyunca yanımda olan ve tecrübelerini benimle paylaşan değerli eşim Dr. Can KAKIŞIM'a, annem Sevim GÖRGÜLÜ ve babam Satılmış GÖRGÜLÜ'ye en içten sevgilerimle ithaf ediyorum.

İÇİNDEKİLER

	Sayfa
ÖZET	v
SUMMARY	vi
TEŞEKKÜR	vii
İÇİNDEKİLER	viii
SİMGELER VE KISALTMALAR DİZİNİ	x
ŞEKİLLER DİZİNİ	xi
TABLolar DİZİNİ	xiii
1. GİRİŞ	1
1.1. Problemin Önemi	1
1.2. Tezin Amacı, İçeriği ve Katkısı	4
1.3. Tezin Organizasyonu	6
2. GÖZETİMSİZ ÖĞRENME	7
2.1. Kümeleme	8
2.1.1. Geleneksel Kümeleme Yöntemleri	9
2.1.2. K-means Algoritması	13
2.2. Boyutsallık Laneti	14
2.3. Boyut İndirgeme	14
2.3.1. Öznitelik Seçimi	15
2.3.2. Gözetimsiz Öznitelik Seçim Stratejileri	17
2.3.3. Gözetimsiz Öznitelik Seçim Yöntemleri	19
2.3.4. Öznitelik Çıkarımı	23
2.3.5. Gözetimsiz Öznitelik Çıkarım Yöntemleri	23
2.3.6. Öznitelik Oluşturma	28
2.3.7. Öznitelik Oluşturma Yöntemleri	29
3. AĞ BAĞLANTILI VERİ	32
3.1. Ağ Yapısı	32
3.2. Ağ Bağlantılı Veri	34
3.2.1. Ağ Bağlantılı Verinin Zorlukları	35
3.2.2. Ağ Bağlantılı Veri Örnekleri	36

3.3. Homofili ve Sosyal Etki	38
3.4. Ağ Bağlantılı Veri için Öznitelik Seçim Yöntemleri	41
3.5. Ağ Bağlantılı Veri için Öznitelik Çıkarım Yöntemleri	45
4. AĞ BAĞLANTILI VERİ İÇİN GÖZETİMSİZ ÖZNİTELİK OLUŞTURMA	49
4.1. Yöntem: NetBFC	49
4.1.1. Tanımlamalar	51
4.1.2. Öznitelik Üretme	54
4.1.3. Öznitelik Seçimi	58
4.1.4. Çizge Yapılandırma	59
4.1.5. Önerilen Yöntemin Algoritması	61
5. DENEYSEL SONUÇLAR	64
5.1. Veri Seti	64
5.2. Değerlendirme Kriterleri	66
5.3. Deneysel Kurulum	67
5.4. Performans Değerlendirmesi	68
5.5. Parametre Analizi	76
5.6. Öznitelik Seçimi Analizi	81
5.7. Çalışma Zamanı Analizi	81
6. SONUÇ VE GELECEK ÇALIŞMALAR	86
KAYNAKLAR	88
ÖZGEÇMİŞ	98
EKLER	99

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler ve Açıklamalar

Kısaltmalar

Acc	: Doğruluk
AANE	: Hızlandırılmış Nitelikli Ağ Gömüleme
BIRCH	: Hiyerarşileri Kullanarak Dengeli İteratif Azaltma ve Kümeleme
CGPFC	: Küme tabanlı GP Özellik İnşaata
CLARA	: Büyük Uygulamalar için Kümeleme
CLICK	: Bağlantı Çekirdekleri ile Küme Tanımlama
CURE	: Temsilcileri Kullanarak Kümeleme
DBSCAN	: Gürültü lü Uygulamaların Yoğunluğa Dayalı Mekansal Kümelenmesi
LANE	: Etiket Bilgili Nitelikli Ağ Katıştırma
LSI	: Gizli Anlamsal İndeksleme
LUFS	: Bağlantılı Denetimsiz Öznitelik Seçimi
MST	: Minumum Yayılım Ağacı
NetBFC	: Ağ Bağlantılı Veriler için İkili Öznitelik Oluşturma
NDFS	: Negatif Olmayan Ayırt Edici Öznitelik Seçimi
NetFS	: Ağ Bağlantılı Veri İçin Öznitelik Seçimi
NMF	: Negatif Olmayan Matris Ayırıştırma Yöntemi
NMI	: Normalizde Edilmiş Ortak Bilgi
PCA	: Temel Bileşenler Analizi
WWW	: Geniş Dünya Ağı
XML	: Genişletilebilir İşaretleme Dili

ŞEKİLLER DİZİNİ

<u>Şekil No:</u>	<u>Sayfa</u>
1.1: Ağ yapısı ve ağ bağlantılı veri örneği.	2
2.1: Gözetimsiz Öğrenme Süreci Örneği.	7
2.2: Kümeleme Görevi Örneği.	9
2.3: Boyutsallık lanetinin grafiksel temsili.	15
3.1: Ağ bağlantılı veri yapısına ait örnek gösterim.	35
3.2: Zamana göre ağ yapısında gözlenen homofili eğilimi.	39
3.3: Zamana göre ağ yapısında gözlenen sosyal etki eğilimi.	40
4.1: NetBFC yönteminin çalışma sürecine ait gösterim.	50
4.2: Öznitelik ve bağlantı yakınlığı örneği.	53
4.3: Öznitelik yakınlığına göre hedef düğüme ait üst- k -yakınlık komşularının belirlenmesi örneği.	56
4.4: Bağlantı yakınlığına göre hedef düğüme ait üst- k -yakınlık komşularının belirlenmesi örneği.	57
4.5: Öznitelik-bağlantı yakınlığına göre üst- k -yakınlık komşularının belirlenmesi örneği.	57
4.6: Hedef düğüm için öznitelik üretme ve seçme süreci örneği.	60
4.7: Çizge Yapılandırma Örneği.	61
4.8: NetBFC yöntemi ile öznitelik ve bağlantı oluşturma süreci örneği.	62
5.1: NetBFC yöntemi ile ve NetBFC yöntemi olmadan çalıştırılan yöntemlerin kümeleme Acc değerleri.	75
5.2: NetBFC yöntemi ile ve NetBFC yöntemi olmadan çalıştırılan yöntemlerin kümeleme NMI değerleri.	75
5.3: Tüm verisetleri için farklı α ve β parametreleri ile elde edilen kümeleme Acc değerleri.	77
5.4: Farklı k ve z parametre değerleri ile WebKB verisi için NetBFC yönteminin kümeleme performansı.	78
5.5: Farklı k ve z parametre değerleri ile Cora verisi için NetBFC yönteminin kümeleme performansı.	78

5.6: Farklı k ve z parametre değerleri ile Citeseer verisi için NetBFC yönteminin kümeleme performansı.	79
5.7: Farklı k ve z parametre değerleri ile BlogCatalog verisi için NetBFC yönteminin kümeleme performansı.	79
5.8: Farklı k ve z parametre değerleri ile verisetlerinin boyutsallığında gözlenen değişimler.	80
5.9: Farklı k ve z parametre değerleri ile tüm verisetlerine ait kümeleme Acc değişimleri.	80
5.10: Farklı parametre ayarlamaları ile verisetleri için NetBFC yöntemine ait zaman analizi.	83
5.11: Tüm verisetleri için yöntemlere ait zaman analizi.	85

TABLÖLAR DİZİNİ

<u>Tablo No:</u>	<u>Sayfa</u>
2.1: Geleneksel kümeleme yöntemleri.	10
5.1: Veri setine ait detaylı bilgiler.	66
5.2: Yöntemlere ait K-means kümelemesi Acc değerleri.	69
5.3: Yöntemlere ait K-means kümelemesi NMI değerleri.	70
5.4: NetBFC yöntemine ait parametreler $k = 250$ ve $z = 250$ iken tüm verisetleri için elde edilen kümeleme performans değerleri.	72
5.5: Tüm verisetler için farklı k ve z değerleri ile elde edilen ortalama kümeleme performans değerleri.	73
5.6: Tüm verisetler için farklı k ve z parametreleri ile istenilen boyutta elde edilen ortalama kümeleme performans değerleri.	74
5.7: Cora verisetinin üç farklı grubuna ait orjinal ve NetBFC ile yapılandırılmış öznitelik dağılım örneği.	81

1. GİRİŞ

Bu bölümde, tez çalışması için seçilen problemin öneminden, tezin amacı, içeriği ve katkısından, ve son olarak tezin organizasyonundan bahsedilecektir.

1.1. Problemin Önemi

Ağ bağlantılı veriler, korelasyon ve bağımlılıkları olan veriler olarak tanımlanmaktadır (Kolaczyk, 2009). Örneğin, bir makalenin içerik bilgisinin yanı sıra, bu makalenin atıfta bulunduğu ve bu makaleye atıfta bulunan makalelere ait bilgi de mevcuttur. Birbirlerine atıf yapan makaleler arasında içeriksel bir yakınlık olduğu varsayılır ve her bir makalenin bir nesne, her bir atıfın bir bağlantı olduğu bir alıntı ağı gözlemlenir. Bu nedenle, bağımsız ve aynı şekilde dağıtıldığı varsayılan geleneksel veriden farklı bir yapıya sahiptir. Ağ bağlantılı verilerde gözlenen korelasyon ve bağımlılıklar, bağlantılara sahip olan e-posta ağları (Diesner vd., 2005) ve protein-protein etkileşimi ağları (Stelzl vd., 2005) gibi birçok ağda doğal olarak meydana gelmektedir. İnsanlar arasındaki sosyal etkileşimi (Tang vd., 2009), karmaşık sistemler arasındaki durumların ilişkisini (Logendran ve Sriskandarajah, 1996), bağlantılı web sayfalarını (Broder vd., 2000), program akışlarını ve bağımlılıklarını (Bhattacharya vd., 2012) temsil etmek için de ağ yapıları kullanılmaktadır. Bu nedenle, ağ verileri iletişim ağları, bilgi ağları ve sosyal ağlar olmak üzere birçok alanda karşımıza çıkmakta, biyolojiden sosyolojiye çok farklı ve çeşitli bilim dallarında ve mühendislik problemlerinde sıklıkla kullanılmaktadır. Şekil 1.1’de nesnelerden ve bağlantılardan oluşan bir ağ yapısı ve nesneleri tanımlayabilecek öznitelikler (tweet, yer bilgisi, e-posta vb.) örneklendirilmiştir.

Günümüzde, çeşitli bilgi sistemlerinin, çevrimiçi kütüphanelerin ve sosyal medya uygulamalarının yükselişi ve yaygın kullanımı ile birlikte kullanıcılar tarafından benzeri görülmemiş bir oranda büyük miktarda ağ bağlantılı veriler üretilmektedir. Bu nedenle, bu tür büyük verilerden faydalı bilgilerin çıkarılması, tüketicilere ürün öneri hizmetleri sunan öneri sistemleri (Berman vd., 2012) ve reklam için doğru alıcıları bulmayı amaçlayan içerik sağlayıcıları (Nicholas ve Carswell, 2012) gibi birçok uygulama için çok önemli hale gelmektedir. Hızlı miktarda veri artışı ile birlikte, bu artışa

paralel olarak ağ bağlantılı verilere ait çözülmesi gereken problemlerin önemi ve miktarı da günden güne artmaktadır. Ayrıca, bu ağlara ait nesne, nesnelere ait içerik ve nesneler arasındaki bağlantı sayıları arttıkça ağ verisine ait graf yapısı da karmaşık bir yapıya dönüşmektedir. Bu nedenle, ağ verisi analizi için yeni teorilerin ve algoritmaların geliştirilmesi çok önemli bir çalışma konusu haline gelmiştir (Tang ve Liu, 2014; Wei vd., 2015).



Şekil 1.1: Ağ yapısı ve ağ bağlantılı veri örneği.

Ağ bağlantılı verilerin genellikle çok fazla sayıda nesneye ve çok fazla sayıda özneliliğe sahip olmasından dolayı, bu ağ içerisindeki nesnelere, nesneler arasındaki ilişkilere ve içeriğe ait detaylı analiz zorlaşmaktadır. Bu yapıdaki verilerin analizi için kullanılan en önemli yaklaşım, ağ nesnelerini daha küçük ve anlamlı gruplar içerisinde analiz etmektir. Ağdaki nesneler, belli özellikler açısından birbirleri ile benzerlik gösteren nesneler ile gruplandırılmaktadır. Böylelikle, ağ nesneleri küme olarak adlandırılan belli gruplara ayrılmaktadır. Ağ verisini kümeleme görevi, farklı disiplinlerde farklı problemleri adresleyecek şekilde uygulanmaktadır. Örneğin; dünya çapında bir geniş ağ olan WWW (World Wide Web), hizmet kalitesini artırmak için kullanıcıları coğrafik bölge ve ilgi alan benzerliklerine göre gruplara ayırmakta ve her bir gruba uygun bir sunucu tahsis etmektedir (Krishnamurthy ve Wang, 2000). Aynı şekilde, pazarlama şirketlerinin yeni bir ürünü pazarlamak için kullandığı öneri sistemlerinde, var olan ağ verisi temel alınarak, benzer müşterileri gruplandırılma yaklaşımı sıklıkla kullanılmaktadır (Nicholas ve Carswell, 2012).

Ağ verisinin temel doğasından kaynaklanan bazı zorlayıcı problemlerden dolayı,

K-means (Lloyd, 1982) gibi geleneksel kümeleme yöntemleri ile anlamlı gruplar elde edilemediği durumlar oluşmaktadır. Bu problemlerden en önemlisi, ağ nesnelerinin birbirinden çok farklı özellikler göstermesi ve bu nedenle ağ verisine ait öznitelik sayısının çok büyük olmasıdır. Bu durumda, ağ nesnelere ait öznitelik uzayı yüksek boyutlulukla karakterize edilmektedir. Yüksek boyutluluk, boyutlandırma laneti olarak adlandırılan bir zorluğa sebep olarak, kümeleme yöntemlerini ya çalışma süresi açısından ya da doğruluk açısından etkisiz hale getirmektedir (Bellman, 1957; Duda vd., 2012). Bu zorluğun üstesinden gelebilmek için, ağ verisine öznitelik çıkarımı ve öznitelik seçimi gibi boyut küçültme yöntemleri uygulanmaktadır. Öznitelik seçimi yöntemleri, ilgili özniteliklerin bir alt kümesini seçerek öznitelik sayısını azaltırken, öznitelik çıkartma yöntemleri verileri daha düşük boyutlu yeni bir uzaya yansıtmaktadır. Fakat, düz veriye uygulanan mevcut öznitelik seçimi ve öznitelik çıkarma yöntemlerinin çoğu Paatero ve Tapper (1994); Cai vd. (2010); Yang vd. (2012), ağa ait bağlantı verilerini göz ardı ederek, yalnızca nesnelere ait öznitelik uzayını kullanmaktadır. Fakat, ağ verileri iki tür bilgi içermektedir: ağı oluşturan nesneler arasındaki ilişkileri temsil eden yapısal veriler ve her nesneyi tanımlayan öznitelik verileri. Bu durumda, aralarında korelasyon olan bu iki verinin öznitelik seçimi ya da çıkarımı gibi öğrenme görevlerinin performansını artıracak şekilde nasıl düzgün bir şekilde entegre edileceği önemli bir problem olarak karşımıza çıkmaktadır (Kumar ve Daumé, 2011; Tang ve Liu, 2014; Le ve Lauw, 2014; Wei vd., 2015). Ağ verisine ait diğer bir problem, ağ nesnelere ait öznitelik uzayının seyrek (sparse) gözlemlenmesidir. Bu seyreklik, bazı veri nesnelerinin çok az özneliği olduğu ve bazı özniteliklerin yalnızca tek bir nesne için mevcut olduğu sosyal medya verilerinde özellikle görülmektedir (Marsden ve Friedkin, 1993). Bu durumda, kümeleme görevlerinde ağ nesneleri arasındaki yakınlığı uygun şekilde hesaplamak zorlaşmaktadır. Ayrıca, ağdaki örnekler ait nitelikler ve bağlantılar ilgisiz, gereksiz ve gürültülü olabilir. Bu türde nitelikler ve bağlantılar, özneliğin kendisini veya öznitelik ile nesne arasındaki ilişkiyi açıklamaya yardımcı olmayan ve istenmeyen öznitelikler ve bağlantılardır. Bu nitelikler ve bağlantılar, öğrenme yöntemlerinin başarısının düşmesine veya aşırı öğrenmesine neden olabilmektedir. Diğer bir problem ise, ağ nesnelerini etiketleme sürecinde ortaya çıkmaktadır. Veri etiketleme işlemi, günden güne artan ve değişen ağ verisi için oldukça zaman alıcı ve pahalı bir iş haline gelmektedir (Gundecha ve Liu, 2012). Gözetimli öznitelik seçimi ya da

çıkarmı yöntemleri, eğitim aşaması için önemli miktarda etiketlenmiş veri gerektirir. Bu nedenle, etiket bilgisine ihtiyaç duymadan öğrenme sağlayan gözetimsiz yöntemler giderek daha önemli hale gelmektedirler.

Öznitelik seçimi yöntemleri ile seçilen öznitelikler, düşük miktarda özniteliğe sahip ağ nesnelerinin doğru kümelenmesi için yeterli bilgiyi taşımayabilirler. Az sayıda özniteliğe sahip bir ağ nesnesi için düşük boyutta seçilen özniteliklerden sadece birkaçı bu nesneye ait olabilir (Hoseini ve Mansoori, 2016). Bu durumda, öznitelik çıkarmı yöntemlerinin kullanımı performans açısından daha etkili olabilmektedir. Fakat, bazı uygulamaların fiziksel işlemlerini yorumlamak için özniteliklerin fiziksel anlamı da önemli olmaktadır. Öznitelik çıkarma yöntemleri bir haritalama fonksiyonu ile öznitelik uzayını daha düşük boyutlu başka bir uzaya yansıttığı için orijinal özniteliklerin fiziksel anlamı kaybolmaktadır. Bu nedenle, fiziksel uzayı değiştirmeden öznitelik uzayını yeniden yapılandırarak boyut azaltımı sağlayan yöntemlere de ihtiyaç duyulmaktadır.

1.2. Tezin Amacı, İçeriği ve Katkısı

Bu tezin amacı, ağ bağlantılı veri üzerinde çalıştırılan ve boyutluluk lanetinden dolayı beklenen performansı gösteremeyen kümeleme algoritmalarının performansını artıracak şekilde, yüksek boyutlulukla karakterize edilmiş ağ verisinin öznitelik uzayını istenilen daha düşük boyutlu ve anlamlı bir uzaya dönüştüren yeni ve etkili bir öznitelik oluşturma yöntemi geliştirmektir. Böylelikle, ağ verisini alt gruplara ayırarak analiz etme problemi adreslenmekte ve ağ verisinden daha doğru ve anlamlı alt kümeler elde edilmesini sağlayacak bir ön işlem süreci önerilmektedir.

Bu tez çalışması kapsamında, öğrenme görevlerinin performansını artırma hedefi ile, ağ verisine ait bağlantı ve öznitelik verisinin birlikte kullanılması amaçlanmıştır. Bu iki verinin aralarındaki korelasyondan faydalanarak, sadece öznitelik uzayını kullanan geleneksel öznitelik seçimi ve çıkarmı yöntemlerinden daha başarılı bir yöntemin geliştirilmesi hedeflenmiştir. Ayrıca, ağ verisine bağlı seyreklik, ilgisiz, gereksiz ve gürültülü bağlantılar ve veri etiketleme problemlerine odaklanılarak, bu tez kapsamında önerilen yöntemin bu zorluklara sahip veri üzerinde başarılı sonuçlar üretmesi bek-

lenmektedir.

Bu tez çalışmasında, ağ bağlantılı verileri üzerinde çalışan NetBFC (Kakisim ve Sogukpınar, 2019) olarak adlandırılan bir gözetimsiz öznitelik oluşturma yöntemi önerilmiştir. Önerilen yöntem, nesneler arasındaki bağlantı ve nitelik yakınlığını bütünleştirerek ağ nesneleri için yapısal ve içeriksel olarak anlamlı, yeni ve düşük boyut-salılıkta bir öznitelik alanı oluşturur. Nesneler arasındaki yapısal ve içeriksel yakınlığını temel alarak, aynı grupta olma olasılığı yüksek olan benzer nesneleri ilişkilendirmekte ve bu nesnelere ait öznitelik uzayını, gruptaki nesneler için daha ayırıcı olan öznitelik-leri temel alarak benzeştirmektedir. Yöntem, seyrek verinin getirdiği zorluk ile baş edebilme için ağ üzerindeki verilere bir öznitelik birleştirme işlemi uygulayarak her bir nesne için yerel öznitelik zenginleştirme süreci uygulamaktadır. Sonrasında yöntem, her bir nesne için yerel bir öznitelik seçim süreci uygulamakta ve bu süreç içerisinde hiçbir nesne tarafından seçilmeyen öznitelikleri ortadan kaldırmaktadır. Bu durum, yüksek boyutsallığa sahip verinin daha düşük bir boyutta temsil edilmesini sağlamaktadır. Yöntem, orijinal öznitelik alanını yeni düşük boyutlu bir alana yansıtmak yerine, sadece nesnelere ait öznitelik değerlerini değiştirdiği ve orijinal alan üzerinde öznitelik eliminasyonu gerçekleştirdiği için özniteliklerin fiziksel anlamı korunmaktadır. Öneri-len yöntem iteratif çalışan bir yöntemdir. Bu nedenle, her iterasyonda ağ bağlantılı veri için farklı boyutsallıkta farklı öznitelik temsilleri elde edilebilmektedir.

Tez çalışmasında sunulan yöntem, dört gerçek dünya veri setinde test edilmiş ve literatürde oldukça bilinen yöntemler ile karşılaştırılmıştır. Deneysel sonuçlar, NetBFC yöntemi ile ağ bağlantılı veri üzerinde, literatürde sıklıkla kullanılan temel öznitelik seçimi ve çıkarımı yöntemlerine kıyasla daha iyi performans elde edilebileceğini göstermektedir.

Bu tez çalışmasının katkıları aşağıdaki şekilde özetlenebilir:

- Önerilen yöntem ile öznitelik oluşturma sürecinde ağ verisine ait iki farklı veri olan bağlantı ve öznitelik verisi birlikte kullanılmıştır.
- Ağ verisine ait bağlantı ve öznitelik verisi arasındaki korelasyonun miktarına göre, bağlantı ya da öznitelik verisinin öznitelik oluşturmadaki katkısı, parametreler sağlanarak azaltılabilmekte ya da artırılabilir.
- Yöntem yinelemeli olarak çalışmakta ve istenen miktarda boyut azaltımı gerçe-

leyebilmektedir.

- Önerilen yöntem ile oluşturulan yeni öznitelik uzayı, diğer öznitelik seçim ve çıkarım yöntemleri ile birlikte uygulanabilmekte, bu sayede bir ön işleme yöntemi olarak kullanılabilir.

1.3. Tezin Organizasyonu

Tez çalışması altı bölümden oluşmaktadır. Birinci bölüm giriş bölümü olup, problemin kısaca tanımı verilmiş, problemin öneminden bahsedilmiş ve sonrasında ise tez çalışmasının amacına, içeriğine ve katkılarına yer verilmiştir.

İkinci bölümde, bilimsel alt yapıya ait kavramlara ve geleneksel veri üzerinde çalışan mevcut öznitelik seçim ve çıkarım stratejilerine ve yöntemlerine yer verilmiştir.

Üçüncü bölümde, ağ bağlantılı verinin tanımına ve var olan örneklerine değinilmekte, ayrıca ağ bağlantılı veri için geliştirilmiş mevcut öznitelik seçim ve çıkarım yöntemleri anlatılmaktadır.

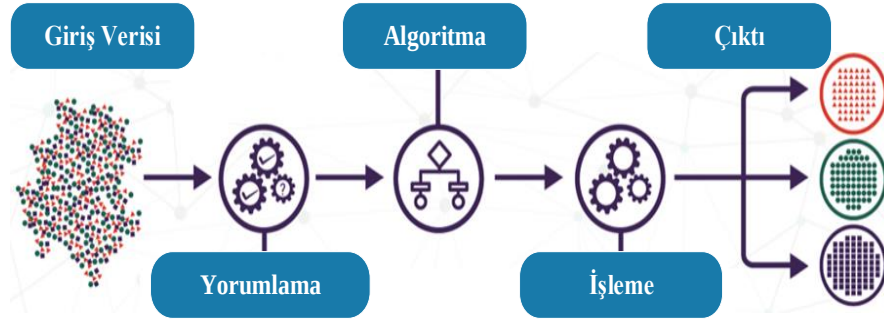
Dördüncü bölümde, tez kapsamında önerilen ağ verisi için gözetimsiz ikili öznitelik oluşturma yöntemi NetBFC'ye ait tanımlamalar yapılmakta ve yöntemin algoritması açıklanmaktadır.

Beşinci bölümde, gerçek dünya verisi olan dört farklı ağ verisi üzerinde NetBFC yönteminin performansına ait deneysel sonuçlar, mevcut seçilmiş öznitelik seçim ve çıkarım yöntemleri ile karşılaştırılarak verilmektedir.

Altıncı bölüm sonuç bölümüdür. Bu bölümde elde edilen sonuçlar özetlenmiştir. Bu konu çerçevesinde gelecekte yapılabilecek çalışmalar için önerilere yer verilmektedir.

2. GÖZETİMSİZ ÖĞRENME

Gözetimsiz öğrenme, yalnızca girdi verilerinizin olduğu ve bu girdi verilerine karşılık gelen çıktı değişkenlerinin olmadığı durumlar için kullanılan bir öğrenme türüdür. Gözetimsiz öğrenmenin amacı, veriler hakkında daha fazla bilgi edinmek için verilerdeki temel yapıyı veya dağılımı bir algoritma ile modellemektir. Öğrenme sürecinin doğruluğunu gözetleyen bir öğretmen yoktur. Şekil 2.1’de girdi verisinden çıktı verisine kadar olan gözetimsiz öğrenme süreci örneklendirilmiştir. Günümüzde, sosyal medyadan e-ticarete, biyoinformatikten, ulaşımaya kadar birçok farklı alanda çok büyük miktarda veri üretilmektedir. Veri boyutundaki artış nedeniyle, veri örneklerinin insanlar tarafından manuel olarak etiketlenmesi oldukça zor ve pahalı bir görev haline gelmiştir. Bu nedenle, gözetimsiz bir şekilde otomatik etiketleme yapmak veri madenciliğinde çok önemli bir süreç haline gelmiştir.



Şekil 2.1: Gözetimsiz Öğrenme Süreci Örneği.

En bilinen gözetimsiz öğrenme yaklaşımlarından biri kümeleme yaklaşımıdır. Veri kümeleme, en popüler ve yaygın veri etiketleme tekniklerinden biridir (Jain vd., 1988). Kümeleme işleminde amaç, verideki benzer örneklerin küme adı verilen bir grupta, benzemeyen örneklerin de farklı kümelerde olmasını sağlamaktır. Bu durumda, küme içerisindeki nesneler arasındaki benzerliğin fazla, kümeler arası benzerliklerin ise uzak olması beklenir. Kümeleme görevi görüntü işleme, örüntü tanıma, örüntü sınıflandırma, ağ analizi vb. birçok makine öğrenmesi ve veri madenciliği görevlerinde sıklıkla kullanılmaktadır.

Veri örneklerin çok fazla sayıda özneliğe diğer bir deyiş ile yüksek boyutluluğa sahip olduğu durumlarda, kümeleme algoritmaları, boyutsallığın laneti olarak bilinen

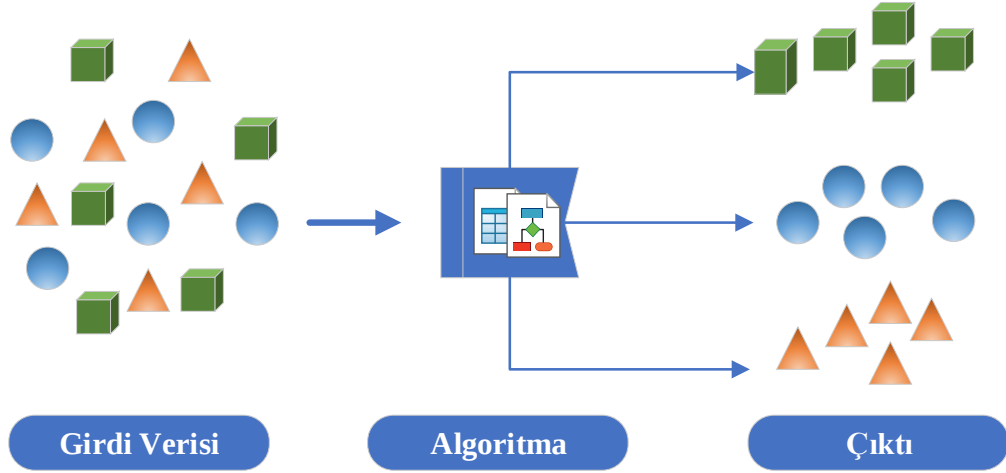
bir problem ile karşılaşmaktadırlar (Bellman, 1957; Duda vd., 2012). Boyutsallığın laneti problemi, çoğunlukla öğrenme görevlerini doğruluk açısından etkisiz hale getirebilmektedir. Bu nedenle, gözetimsiz öğrenme algoritmaların yüksek boyutlu veriler üzerinde daha etkili ve verimli bir şekilde kullanılabilmesi için etiketsiz veri üzerinde çalışan boyut azaltma yöntemlerine ihtiyaç duyulur. Boyut azaltma, bir dizi ana değişken elde ederek, dikkate alınan rastgele değişken sayısını azaltma işlemidir (Fodor, 2002). Boyut azaltmak için öznel- telik seçimi ve öznitelik çıkarımı olmak üzere iki farklı yaklaşım kullanılmaktadır. Öznitelik seçimi yöntemleri, ilgili özniteliklerin bir alt kümesini seçerek öznitelik sayısını azaltırken, öznitelik çıkarım yöntemleri bir haritalama fonksiyonu ile verileri daha düşük boyutlu yeni bir alana yansıtmaktadır.

Bu bölümde, gözetimsiz öğrenme kapsamında, bir gözetimsiz öğrenme süreci olan kümeleme görevi, boyutsallık laneti, öznitelik seçimi ve öznitelik çıkarımı yaklaşımları ve var olan genel yöntemler anlatılacaktır.

2.1. Kümeleme

Gözetimsiz öğrenmenin en önemli problemi olarak kabul edilen kümeleme, bilinmeyen alandaki veri ile ilgilenen bir ileri öğrenme görevidir. Kümeleme işleminin temel amacı benzer nesneleri birlikte, benzerlik göstermeyen nesneleri ayrı gruplamaktır. Şekil 2.2’de anlamsal olarak üç farklı gruba ait (üçgen, küp, daire) olan örneklerin kümeleme yaklaşımı ile gruplandırılması örneklendirilmiştir. Bir kümeleme algoritması ile gözetimsiz olarak nesneler anlamsal yakınlığına göre gruplandırılmaktadır. Buradaki zorluk, veriye ait grup sayısı veya grup yapısı ile ilgili önsel bir bilgiye sahip olunmamasıdır. Gruplandırma, benzerlikler veya mesafeler (farklılıklar) temelinde yapılır. Bu nedenle, kümeleme yöntemleri genellikle nesneler arasındaki uzaklığı, yoğunluğu veya hiyerarşiyi temel almaktadır. Benzerlik ve uzaklık ölçüsü seçiminde çoğu zaman çok fazla öznellik vardır. Değişkenlerin doğal yapısı (ayrık, sürekli, ikili) ve ölçüm ölçekleri (nominal, sıra, aralık, oran) hangi ölçme tekniğinin seçilmesi gerektiğini belirleyen hususlardır. Genellikle, nicel (sayısal değişkenler) veri özelliklerine sahip veriler arasındaki ilişkiyi tanımak için uzaklık ölçüsü tercih edilirken, nitel (kategorik değişkenler) veri özellikleriyle uğraşan yöntemler benzerlik ölçüsünü tercih etmek-

tedir (Xu ve Wunsch, 2005). Minkowski, Kosinüs ve Pearson korelasyon mesafesi mesafe fonksiyonları için, Jaccard ve Hamming benzerliği de benzerlik fonksiyonları için örnek verilebilir. Matematiksel sunuma ve detaylı bilgiye Xu ve Wunsch (2005)'un makalesinden ulaşılabilir.



Şekil 2.2: Kümeleme Görevi Örneği.

Standart kümeleme süreci aşağıda verilen birkaç süreci içermektedir (Xu ve Wunsch, 2005):

- Öznitelik çıkarma ve seçme: orijinal veri setinden veriyi en iyi temsil eden öznitelikler çıkarılır ve seçilir;
- Kümeleme algoritması tasarımı: kümeleme algoritması problemin özelliklerine göre tasarlanır;
- Sonuç değerlendirme: kümeleme sonucu ve algoritmanın geçerliliği değerlendirilir.

2.1.1. Geleneksel Kümeleme Yöntemleri

Literatürde farklı yaklaşımlara sahip birçok kümeleme yöntemi vardır. En tipik algoritmaları içerecek şekilde, Tablo 2.1'de geleneksel kümeleme yöntemleri 6 kategoride örneklenmiştir (Xu ve Wunsch, 2005).

Tablo 2.1: Geleneksel kümeleme yöntemleri.

Kümeleme algoritması (Yaklaşım)	Örnek Algoritmalar
Bölümleme temelli	K-means, K-medoids, CLARA
Hiyerarşiye dayalı	BIRCH, CURE, ROCK
Bulanık teoriye dayalı	FCM, FCS
Yoğunluğa dayalı	DBSCAN, OPTICS
Çizge teorisine dayalı	CLICK, MST

Bölümleme temelli kümeleme algoritmalarının temel fikri, veri noktalarının merkezini, karşılık gelen kümenin merkezi olarak görmektir. Genel yaklaşım, öğrenme süreci içinde bir kümede etiketlenmiş noktalar ile o kümenin merkezi olarak belirlenmiş nokta arasındaki mesafeyi en aza indirmektir. K-means (Lloyd, 1982) ve K-medoids (Park ve Jun, 2009) algoritmaları en bilinen bölümleme temelli algoritmalarlardır. K-means yöntemi, veri noktalarının merkezi tarafından temsil edilen küme merkezini, yinelemeli hesaplamayla güncellemekte ve yinelemeli işlem yakınsama kriterlerini karşılanıncaya kadar devam ettirmektedir. K-means'den farklı olarak, K-medoids algoritması merkez noktaları için veri noktalarını (medoids) seçer. K-means yönteminde merkezler gerçek bir veri noktasına karşılık gelmediğinden, uçdeğerlere (outlier) karşı hassastır. Bu durum, istenilen başarıda merkez noktası hesaplamayı zorlaştırabilir. K-medoids bu durumlar için daha avantajlı olsa da, çalışma zamanı açısından K-means daha verimlidir. Diğer sık kullanılan bölümleme temelli yöntemlerden biri de CLARA (Clustering Large Applications) (Kaufman ve Rousseeuw, 2009) algoritmasıdır. Bu yöntem büyük veri setlerini gruplamak için sıklıkla tercih edilir. K-medoids algoritmasında olduğu gibi veri setinin tamamı için merkez bulmak yerine, CLARA veri setinden küçük bir örnek alır ve numune için en uygun merkez (medoid) setini üretmek için Kmedoids algoritmasını uygular.

Hiyerarşik yöntemler, verileri hiyerarşi gibi görünen farklı seviyelere böler. Her veri noktasını başlangıçta ayrı bir küme olarak temsil eder ve en benzer iki kümeyi yalnızca bir küme kalıncaya kadar birleştirir. Yada, tersi bir işlem uygular. Bu tür kümeleme, veri görselleştirmesi ve özetlemesi işlemleri için elverişlidir. Hiyerarşik kümeleme yaklaşımları, her bir nesneyi öncesinde ayrı bir küme olarak değerlendirip ve her aşamada benzer kümeleri sırayla birleştirerek aşağıdan yukarıya doğru çalıştığında

aglomeratif (agglomerative), öncesinde bütün veri noktalarını bir kümede gruplayıp ardından bu kümeleri ardışık olarak bölerek yukarıdan aşağıya doğru çalıştığında ise bölücü (divisive) hiyerarşik kümeleme olarak adlandırılır. BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) (Zhang vd., 1996), CURE (Clustering Using Representatives) (Guha vd., 1998) ve ROCK (Robust Clustering using Links) (Guha vd., 2000) en bilinen hiyerarşik kümeleme yöntemleridir. BIRCH, kümeleme işlemini, bir düğümü bir alt küme küme olarak değerlendirip bir kümeleme öznitelik ağacı oluşturarak gerçekleştirir. Yeni bir veri noktası geldiğinde ağacı dinamik olarak büyütür. CURE ve ROCK yöntemleri büyük ölçekli kümeleme problemleri için tercih edilir. Veri noktalarını ayrı olarak kümelemek için rastgele örnekleme tekniğini kullanırlar ve sonuçları işlemin en sonunda birleştirirler.

Bulanık olmayan kümelemede (sert kümeleme), veriler her kümenin yalnızca tam bir kümeye ait olabileceği farklı kümelere bölünür. Bulanık teoriye dayalı, bulanık kümelemede veri noktaları çoklu kümelere ait olabilir. Bu tür kümeleme algoritmalarının temel fikri, nesneler arasındaki ilişkiyi daha makul bir şekilde tanımlamak için nesnelerin küme etiketine ait (0, 1) ayrık değerini, [0, 1] sürekli bir aralıkta olacak şekilde atamaktır. FCM (Fuzzy C-Means) (Bezdek vd., 1984) ve FCS (Fuzzy C-Sheels) (Dave ve Bhaswan, 1992) yöntemleri en bilinen bulanık kümeleme yöntemlerindendir. FCM yöntemi çalışma süreci açısından K-means yöntemi ile benzerlik göstermektedir. K-means yönteminde her nokta bir merkeze atanırken, FCM’de her nokta farklı ağırlıklarla aynı anda ikiden fazla farklı kümeye atanabilir. Diğer yöntem FCS, doğrusal alt alanlarda kümeleri arayan FCM yönteminden farklı olarak, doğrusal olmayan alt alanlarda uzanan kümeleri tespit etmeyi amaçlamaktadır. Hiper küp şeklindeki kümeleri tespit etmeye çalışır. Kullandığı mesafe ölçüsü de hiper küpler arasındaki mesafeleri hesaplamaktadır.

Yoğunluğa dayalı kümelemede, birbirinden daha geniş alanlarla ayrılmış veri alanındaki yoğun alanlar küme olarak tanımlanmaktadır. Gürültü alanları içindeki yoğunluk, kümelerin herhangi birindeki yoğunluktan daha düşük olmalıdır. Yoğunluğa dayalı kümeleme, S-şekli gibi rastgele şekilli kümeleri yakalayabilmektedir. En bilinen örneği DBSCAN (Density-based Spatial Clustering of Applications with Noise) (Ester vd., 1996) ve OPTICS (Ordering Points to Identify Cluster Structure) (Ankerst vd., 1999) yöntemleridir. DBSCAN, veri kümesindeki her nokta için, bu veri nok-

tasının etrafında n boyutlu bir şekil oluşturur ve daha sonra bu şeklin içine kaç veri noktasının düştüğünü sayar. DBSCAN, kümedeki her bir noktadan geçerek ve yakındaki diğer veri noktalarının sayısını sayarak kümeyi yinelemeli olarak genişletir. OPTICS yöntemi, değişen yoğunluklardaki verileri işleyecek şekilde DBSCAN temel alınarak geliştirilmiştir.

Çizge teorisine dayalı kümelemede, düğümün veri noktası ve kenarların veri noktaları arasındaki ilişkiyi tanımlayan bağlantı olarak kabul edildiği çizgeler üzerinden kümeleme gerçekleştirilir. En bilinen çizge tabanlı kümeleme yöntemleri için CLICK (CLuster Identification via Connectivity Kernels) (Sharan ve Shamir, 2000) ve MST (Minimum Spanning Tree) (Zahn, 1970) yöntemi örnek olarak verilebilir. CLICK yöntemi, minimum kesim (minimum cut) hesaplamalar ile yinelemeli olarak ağırlıklı bir çizgeyi bileşenlere ayırmayı amaçlamaktadır. Algoritma, giriş verisini ağırlıklı bir benzerlik çizgesi ile temsil eder. Bu çizgede, düğümler veri nesnelerine karşılık gelir ve kenar ağırlıkları veri nesneleri arasındaki benzerlik değerlerinden elde edilir. Yöntemin daha verimli çalışmasını sağlamak için genellikle bir eşik değerine göre benzerliği yüksek düğümler arasında kenarlar tutulur ve eşik değerinden düşük olan kenarlar çizgeden çıkarılır. MST kümeleme yöntemi, giriş verilerine ait öncelikle Kruskal algoritmasını kullanarak bir MST (minimum yayılım ağacı) oluşturur. Ardından bir eşik değeri ve adım boyutu ile uzunlukları eşik değerinden büyük olan kenarları bu yayılım ağacından çıkarır. Daha sonra küme içi mesafe ile kümeler arası mesafe hesaplanır ve eşik değeri güncellenir. Yeni (güncellenmiş) eşik değeri ile süreç yinelemeli olarak tekrarlanır.

Bölümleme temelli yöntemler özellikle K-means nispeten düşük zaman karmaşıklığına sahiptir. Bu yöntemler, dışa dönük (non-convex) olmayan veriler için uygun değildir. Uç değerlere (outlier) nispeten duyarlıdır. Küme sayısının giriş olarak önceden verilmesi gerekmektedir ve küme sayısına duyarlı kümeleme sonuçları üretmektedir. Hiyerarşik tabanlı yöntemlerden olan BIRCH ve CURE yöntemleri düşük zaman karmaşıklığına sahip olsada, hiyerarşik yöntemler nispeten daha yüksek zaman karmaşıklığına sahiptir. Aynı şekilde, hiyerarşik yöntemler için de küme sayısı bir ön değer olarak verilmelidir. Bu yöntemler, rastgele şekilde ve rastgele türde niteliklere sahip veriler için daha uygundur. Yoğunluk temelli yöntemler de rastgele şekildeki veriler için daha verimli kümeleme gerçekleştirir. Fakat, veri alanlarındaki yoğunluk eşit

olmadığında düşük kalitede bir kümelendirme sonucuna sebep olabilir. Özellikle, veri hacmi büyük olduğunda büyük boyutlu bir belleğe ihtiyaç duyulmaktadır. Çizge tabanlı yöntemler genellikle yüksek doğrulukla kümeleme sonucu elde etmektedir. Fakat, çizgenin karmaşıklığı arttıkça, hesaplama zamanı önemli ölçüde artmaktadır.

2.1.2. K-means Algoritması

K-means kümeleme algoritması (Lloyd, 1982), en popüler gözetimsiz öğrenme yöntemlerinden biridir. K-means algoritmasının kullandığı ana fikir, her bir küme için bir tane olacak şekilde k tane çekim merkezi (centroid) tanımlamaktır. Farklı konumlarda tahsis edilen çekim merkezleri farklı sonuçlara neden olduğu için merkezler akıllıca bir şekilde yerleştirilmelidir. Bu nedenle, başlangıç adımı için sezgisel olarak bu çekim merkezlerini birbirinden mümkün olduğunca uzağa koymak önemlidir. Bir sonraki adım, belirli bir veri setindeki her noktayı ona en yakın çekim merkezi ile ilişkilendirmektir. Bu durumda, ilk k adet grup elde edilmiş olur. Bu noktada, oluşan gruplar temel alınarak, k adet çekim merkezi yeniden hesaplanmaktadır. Veri setindeki her nokta için çekim merkezlerine olan uzaklıklara göre gruplar belirlenmekte ve merkezlerin yeri yeniden hesaplanmaktadır. Bu döngü, çekim merkezleri başka bir noktaya hareket etmeyene veya önceden belirlenen bir yineleme sayısına ulaşılan kadar devam eder.

Kümeleme görevi uygulanacak n sayıda örnekten oluşan $X = \{x_1, x_2, \dots, x_n\}$ veri setinde x_i örneği D boyutlu bir örneği temsil etsin. K-means algoritması, X veri setini $K = \{C_1, C_2, \dots, C_k\}$ olacak şekilde k adet kümeye bölmeyi amaçlamaktadır. Algoritma, bir hedef fonksiyonu en aza indirmeyi, bu durumda bir kare hata fonksiyonu J 'yi en aza indirmeyi amaçlamaktadır. Hedef fonksiyon Denklem 2.1'de verilmektedir. Burada, μ_j merkezi küme C_j ile uzaklık temelinde ilişkilendirilmiş tüm örneklerin ortalaması hesaplanarak bulunur.

$$J = \sum_{j=1}^K \sum_{x_k \in C_j} \|x_k - \mu_j\|^2 \quad (2.1)$$

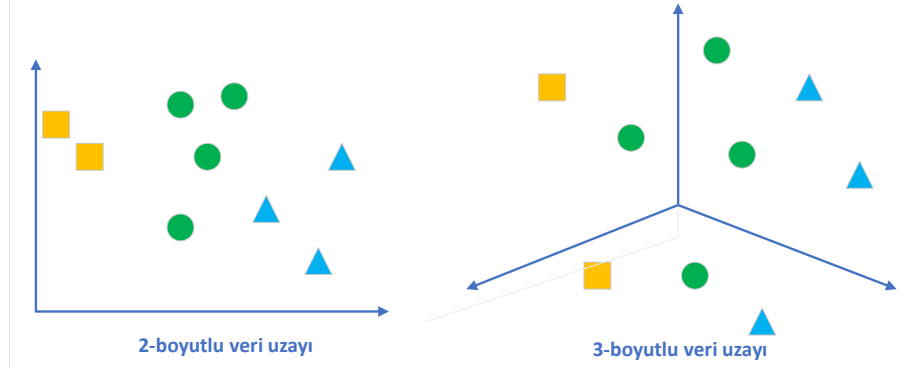
K-means yöntemi, kümelere ait çekim merkezlerini güncellemek için genellikle

iki farklı yaklaşım kullanmaktadır. İlk yaklaşım, tüm veri örneklerini en yakın komşu örnekler ile gruplandırarak çekim merkezlerini yeniden hesaplamaktır (Forgy, 1965). İkinci yaklaşım (MacQueen, 1967) ise, her örnek için en yakın komşular hesabı yaptıktan sonra çekim merkezlerini güncellemektir. K-means algoritması sadece local optimumu bulan açgözlü bir algoritmadır. Ayrıca, bulduğu sonuç, başlangıçtaki çekim merkezi atamalarına göre değişebilmektedir. Bu nedenle, en iyi sonucu elde etmek için genellikle K-means algoritması farklı başlangıç değerleri ile birden fazla kez çalıştırılmaktadır.

2.2. Boyutsallık Laneti

Boyutsallık laneti, ilk olarak Bellman (Bellman, 1957) tarafından öne sürülen, çok değişkenli makine öğrenmesi sürecinde ortaya çıkan en önemli sorunlardan biridir. Öklid uzayına ekstra boyutlar eklenmesiyle, ilişkili hacimdeki üssel artıştan kaynaklanan sorunu tanımlayan bir terimdir. Boyutluluk arttığında, alanın hacmi o kadar hızlı artar ki, mevcut veriler seyrekleşir. Bu durumda, istatistiksel öneme sahip olan herhangi bir öğrenme yöntemi ile güvenilir bir sonuç elde etmek için büyük miktarda veri ihtiyacı ortaya çıkmaktadır ve gereken veri miktarı genellikle boyutsallıkla birlikte üssel olarak artar. Hughes fenomenine göre (Hughes, 1968), veriye ait öznitelik sayısı artış gösterdiğinde ve öznitelik sayısı optimum sayıda öznitelige ulaşana kadar arttığında, bir istatistiksel öğrenme yönteminin performansının da aynı şekilde artış gösterir. Fakat öznitelik sayısı veri örneklerinin sayısını geçmeye başladığında, öğrenme yönteminin performansının düştüğünü belirtir. Şekil 2.3’de 2-boyutlu ve 3-boyutlu veri uzayı örnek bir gözlem kümesi temel alınarak temsil edilmiştir. Boyut miktarının 1-boyut artması ile gözlemlenen örneklerin birbirinden nasıl uzaklaştığı gösterilmektedir. Boyutsallığın artışı ile bu uzaklaşma üstselsel olarak artmaktadır.

Boyutsallık laneti ve yoğun seyrek veri uzayı, uzaklık ve benzerlik ölçüsüne göre gruplama yapan kümeleme yöntemlerinin gruplama stratejilerinin etkili olmasını engellemektedir. Çünkü, nesnelere ait öznitelik dağılımı arttıkça, verimli bir şekilde nesneler arasındaki benzerliği hesaplamak zorlaşır.



Şekil 2.3: Boyutsallık lanetinin grafiksel temsili.

2.3. Boyut İndirgeme

Teorik olarak nesnelere ait daha fazla özniteliğe sahip olmak daha fazla ayırt edici güce neden olmalıdır düşüncesi oluşsa da pratikte bu veriler üzerinde gerçekleştirilen öğrenme görevleri, yüksek boyutsallık sebebi ile beklenen çıkarımları yapmakta zorlanmaktadır. Yüksek boyutsallık sorunun üstesinden gelmek için, öznitelik seçimi ve öznitelik çıkarımı olarak kategorize edilen boyut azaltma amacı ile kullanılan öznitelik oluşturma yöntemleri önerilmektedir (He vd., 2006; Dy, 2008; Yang vd., 2012; Li vd., 2012). Öznitelik oluşturma yöntemleri, daha basit ve anlaşılır modeller oluşturmak, temiz, anlaşılır veri hazırlamak ve bu sayede veri madenciliği ve öğrenme görevlerinin performansını artırmak amacı ile kullanılmaktadır. Ayrıca, bellek depolama gereksinimlerini ve hesaplama maliyetlerini önemli ölçüde azaltmak için de bu yöntemler sıklıkla kullanılmaktadır.

2.3.1. Öznitelik Seçimi

Öznitelik seçim yöntemleri ile orijinal öznitelik uzayı üzerinde bir değişiklik yapmadan orijinal öznitelik uzayını en iyi temsil eden öznitelik altkütmesi seçilmektedir (Fodor, 2002). Bu yöntemler, veri örneklerine ait etiket bilgisi varlığına göre, gözetimli, gözetimsiz ve yarı gözetimli yöntemler olarak sınıflandırılır. Gözetimli yöntemler, yeterli etiket bilgisi mevcut olduğunda uygulanabilirken, gözetimsiz yöntemler herhangi bir etiket bilgisine ihtiyaç duymazlar. Bununla birlikte, gerçek dünya verilerinde

çok sayıda etiketsiz örneğe rağmen bazı veri örnekleri için etiket bilgisi mevcut olabilmektedir. Gözetimli ve gözetimsiz öznitelik seçim yöntemleri bu senaryoda yer alan örneklerden tam olarak yararlanamamaktadır. Çünkü gözetimli yöntemlerde, etiketli numunelerin az sayıda olması, öznitelikler arasındaki korelasyon bilgisinden faydalanmak için yetersiz olurken, gözetimsiz yöntemler etiketleri tamamen görmezden gelmektedir. Bu durumlarda, hem etiketli hem de etiketsiz örnekleri kullanabilen yarı gözetimli yöntemler tercih edilir. Gözetimli ve yarı gözetimli öznitelik seçim yöntemleri genellikle sınıflandırma veya regresyon problemleri için kullanılırken, gözetimsiz yöntemler genellikle kümeleme problemleri için kullanılmaktadır.

Öznitelik seçim yöntemleri, verilen bir problem için asgari performans düşüşü ile veriyi doğru şekilde tanımlayan bir öznitelik alt kümesi bulmayı amaçlamaktadır. Araştırmacılar, yüzlerce ya da binlerce özniteliğin olduğu durumlarda, bu özniteliklerin çok büyük bir kısmının öğrenme görevlerinde önemli bir rol oynamadığını ve çoğunun alakasız ya da gereksiz öznitelikler olduğunu göstermişlerdir (Yang ve Pedersen, 1997). Bu doğrultuda, öznitelik seçimi yöntemleri veriye ait önemli, diğer bir deyişle, alakalı özniteliklerin tespit edilmesi ve gereksiz olanların elenmesi yaklaşımlarını kullanırlar. Sezgisel olarak, bir öznitelik bir hedef hakkında bazı bilgiler içeriyor ise bu öznitelik alakalı öznitelik olarak tanımlanır. Bir özniteliğin alakalık derecesini ideal bir Bayes sınıflandırıcısını temel alarak tahmin etmeye çalışabiliriz. N örneğe ve D özniteğe sahip bir veri matrisi $X = \{x_1, x_2, \dots, x_N\}$ olsun. $F_1, F_2, \dots, F_D$ öznitelikler olmak üzere, her örneğin öznitelik vektörü $F = \{f_1, f_2, \dots, f_D\} \in R^{N \times D}$ olarak temsil edilir. Eğer F_i özniteliği tüm öznitelikleri içeren öznitelik uzayı S 'den çıkarıldığında $S_i = F - F_i$, Bayes sınıflandırıcısının tahmin etme doğruluğunda önemli bir düşüş gözlenir ise, bu öznitelik için şiddetle alakalı (strong relevance) bir özniteliktir diyebiliriz. Eğer bir öznitelik şiddetle alakalı bir öznitelik değil ise, fakat bir öznitelik alt kümesi olan S_i kümesine dahil olduğunda $S_i \cup F_i$, sınıflandırıcının başarısını olumlu yönde etkiliyor ise bu özniteliklere zayıf alakalı öznitelikler denmektedir. Eğer bir öznitelik ne güçlü ne de zayıf şekilde alakalı değilse, alakasız öznitelik olarak tanımlanır.

Öznitelik seçim yöntemleri için önemli olan bir diğer yaklaşım gereksiz özniteliklerin öznitelik uzayından çıkarılmasıdır. Bir özniteliğin öğrenme süreci için gerekli olup olmadığı öznitelikler arasındaki korelasyon temel alınarak öğrenilir. Eğer iki öznitelik birbirleri ile tamamen korelasyon içerisinde ise bu özniteliklerden biri gerek-

siz öznitelik olarak tanımlanabilir. Örneğin; $F_i = F_j$ durumu var ise, bu öznitelikten birinin seçilmesi yeterli olacaktır ve diğeri gereksiz bir öznitelik olacaktır. Bu doğrultuda, öznitelik seçim yöntemlerinin amacı en optimum öznitelik altkümelerini bulmaktır. Ancak, olası öznitelik altkümelerinin sayısı, boyutsallığın artmasıyla birlikte üssel olarak artar. Optimal bir altkümenin bulunması genellikle zordur ve öznitelik seçimiyle ilgili birçok problemin NP-zor problemler olduğu gösterilmiştir (Kohavi ve John, 1997). Öznitelik seçimi algoritmaları genellikle öznitelik altkümelerinin alanı boyunca bir arama yapar. Bu arama sürecinde arama sürecini etkileyen üç temel süreç bulunmaktadır (Langley, 1994). Bu süreçler sırasıyla başlama noktası ve öznitelik altkümesi oluşturma, değerlendirme stratejisi ve durma kriteri olarak bilinmektedir. Öznitelik alt kümesi arama sürecinde, aramanın başlatılacağı noktayı seçmek, aramanın yönünü etkileyebilir. Bu nedenle, sezgisel olarak üç genel yaklaşım bulunmaktadır. Birinci yaklaşım ile, arama sürecine boş bir küme ile başlayıp kademeli olarak arama uzayı boyunca ileri giderek, kümeye öznitelikler eklenmektedir. Bu yaklaşıma sıralı ileri seçim denmektedir. İkinci yaklaşım da, arama tüm öznitelikler ile başlatılır ve öznitelikler arama sürecinde art arda kaldırabilir. Bu yaklaşıma sıralı geriye doğru seçim denmektedir. Bu durumda, arama, arama alanı içinde geriye doğru ilerler. Diğer bir alternatif ise uzayın orta noktasına düşen bir yerden başlamak ve bu noktadan dışarıya doğru hareket etmektir.

Öznitelik altkümelerinin nasıl değerlendirileceği öznitelik seçim algoritmalarındaki farklılaşmayı sağlayan en önemli faktördür. Farklı seçim stratejileri açısından, gözetimli öğrenme sürecinde kullanılan özellik seçimi yöntemleri sarmalayıcı, filtre ve gömülü yöntemler olarak kategorize edilmektedir. Aşağıdaki alt bölümde, gözetimsiz öğrenme sürecinde sıklıkla kullanılan öznitelik seçim stratejileri anlatılacaktır.

2.3.2. Gözetimsiz Öznitelik Seçim Stratejileri

Öznitelik seçim yaklaşımları sarmalayıcı, filtre ve gömülü yöntemler olarak kategorize edilmektedir. Gözetimsiz öğrenmede, sarmalayıcı model, seçilen özniteliklerin kalitesini değerlendirmek için bir kümeleme algoritmasını kullanır. Bir öznitelik alt kümesi ile başlar ve seçilen alt kümeyi kullanarak kümeleme kalitesini değerlendirir.

İstenen kalitede öznitelik alt kümesi bulunana kadar bu süreci devam ettirir. İlk öznitelik altkümesi oluşturduktan sonra, öğrenme algoritması, öğrenme performansına dayalı olarak bu özniteliklerin kalitesini değerlendirmek için bir kara kutu gibi davranır. Bu durum, sarmalayıcı yöntemleri genel ve basit bir yaklaşım haline getirir. Fakat, öznitelikler önceden seçilmiş bir öğrenme algoritması için optimize edildiğinden başka bir öğrenme yöntemi için uygun olmayabilir. Diğer bir yandan, sarmalayıcı yöntemler kaba kuvvet çalışan yöntemlerdir ve büyük miktarda hesaplama gerektirir, çünkü arama işlemi sırasında çok sayıda kümeleme işlemi gerçekleştirilir. Etkili arama stratejileri ile hızlanma sağlanabilir. Fakat, d boyutsallığında bir öznitelik uzayı için arama alanı 2^d olmaktadır ve d 'nin büyük olduğu durumlarda ayrıntılı arama süreci artan boyutsallıkla olanaksız hale gelmektedir. Arama stratejisi ve kullanılan kümeleme algoritması kombinasyonu değiştirilerek kümeleme için farklı sarmalayıcı öznitelik seçim yöntemleri önerilmektedir (Dy ve Brodley, 2004; Kim vd., 2002).

Filtre yöntemleri, özniteliklerin kalitesini test etmek için herhangi bir kümeleme algoritmasını kullanmamaktadır (Dy, 2008). Özniteliklerin önemini değerlendirmek için verinin sadece belirli özniteliklerini temel alırlar. Filtre yöntemleri, genel olarak, özniteliğin önemini tespit etmek için bazı öznitelik değerlendirme kriterlerini kullanarak öznitelikleri puanlar. Bu puanlama sürecinde, her öznitelik diğer özniteliklerden bağımsız tek başına puanlanabilir ya da birden fazla öznitelik ile birlikte puanlanabilir. Sonrasında, az puana sahip olan öznitelikler filtrelenerek, kalan öznitelik kümesi hedef öznitelik kümesi olarak seçilmektedir. Günümüzde, olasılıksal mesafe ölçüleri, Pearson korelasyonu, entropi veya diğer bilgi-teorik ölçümler gibi birçok farklı değerlendirme kriteri kullanan filtre yöntemleri önerilmiştir (Biesiada ve Duch, 2007). Filtre yöntemleri oldukça hızlı çalışır. Bu nedenle diğer yöntemlerin hesaplama karmaşıklığından dolayı verimli olamadığı yüksek boyutlu problemler için daha kullanışlı olmaktadır. Bununla birlikte, öznitelik seçimi aşamasını yönlendiren belirli bir öğrenme algoritmasının bulunmamasından dolayı, seçilen öznitelikler hedef öğrenme algoritmaları için optimal olmayabilir. Benzer şekilde, beklenen riske göre global olarak optimize edilmiş bir değerlendirme kriter fonksiyonu tanımlamak genellikle zordur.

Filtre ve sarmalayıcı modellerin dezavantajının üstesinden gelmek ve daha iyi kümeleme kalitesi elde etmek için, etkin filtreleme kriterlerini ve sarmalayıcı modelin arama yaklaşımını birlikte kullanan gömme modelleri önerilmektedir (Zhu vd.,

2007). Gömme modelleri, farklı aday altkümeleri seçmek için filtreleme kriterlerini kullanırlar. Sonrasında, her aday alt kümenin kümelenme kalitesini arama stratejisi ile değerlendirmektedir. En yüksek kümeleme kalitesine sahip alt küme seçilmektedir. Gömme yaklaşımlı öznitelik seçim yöntemleri genellikle filtre tabanlı yöntemlerden daha iyi kümeleme kalitesi sağlarlar. Fakat, sarmalayıcı yaklaşımda kullandıklarından çalışma zamanı açısından filtre tabanlı yöntemlerden daha az verimli olmaktadır.

2.3.3. Gözetimsiz Öznitelik Seçim Yöntemleri

Sınıf etiketleri kullanmadan, özelliklerin ayırt edilebilirliğini değerlendirmek gözetimli yöntemlere kıyasla daha zor bir süreç gerektirmektedir. Gözetimsiz öznitelik seçimindeki temel sorun, verilere gömülü olan ayırt edici bilgilerin nasıl etkili bir şekilde ortaya çıkarılacağıdır. Bu problem için kullanılan en genel yaklaşımlardan biri verinin yerel yapısını veya benzerliğini korumaktır. Benzerlik koruma yaklaşımları, orijinal verilerin yerel yapısını en iyi koruyan öznitelikleri seçmeyi amaçlamaktadır. Diğer bir deyişle, bir öznitelik yerel yapıyı iyi muhafaza edebiliyor ise bu özneliğin şiddetle alakalı öznitelik olduğu sonucuna varılır.

Filtreleme tabanlı benzerlik koruma yaklaşımlarının en tipik örneği SPEC (Spectral Feature Selection) (Zhao ve Liu, 2007) yaklaşımıdır. SPEC, bir benzerlik matrisi S 'den türetilmiş bir matrisin spektrumuyla öznitelik tutarlılığını tahmin ederek özneliğin alakalılığını hesaplamaya çalışmaktadır. SPEC, Radyal-Bazlar Fonksiyonu RBF'yi, Denklem 2.2'de olduğu gibi, iki örnek x_i ve x_j arasındaki benzerliği hesaplamak için benzerlik fonksiyonu olarak kullanır.

$$S_{ij} = e^{-\frac{\|x_i - x_j\|}{2\sigma^2}} \quad (2.2)$$

SPEC, örneklerin arasındaki benzerliği tutan benzerlik matrisi S 'e dayalı olarak oluşturulan bir çizge yapısına göre özniteliklerin nasıl seçileceğini araştırır. Bu nedenle, özneliğin alakalı öznitelik olup olmadığına karar vermek için bir öznitelik ağırlıklandırma yaklaşımı kullanılmaktadır. Ağırlıklandırma işlemi, Laplas (Laplacian) matrisleri üzerinden yürütülür. Graf G , benzerlik matrisi S temel alınarak oluşturulmuş

bir yakınlık çizgesi, W bu çizgeye ait komşuluk matrisi ve W 'e ait derece (köşegen) matrisi $\bar{D}$ olmak üzere, W ve $\bar{D}$ matrisi kullanılarak Laplas matrisi ve normalize edilmiş Laplas matrisi Denklem 2.3'de gösterildiği gibi elde edilir.

$$L = \bar{D} - W; \quad \mathcal{L} = \bar{D}^{-\frac{1}{2}} L \bar{D}^{-\frac{1}{2}} \quad (2.3)$$

SPEC yönteminde temel alınan ana fikir, çizgede birbirine yakın olan örneklerle (düğümlere) ait özniteliklerin, çizge yapısı içerisinde de yakınlık göstermesi diğer bir değişle her iki yapıda da tutarlılık göstermesi gerektiğidir. Bu durumda, benzer örnek grubunda (kümede) benzer şekilde gözlenen öznitelikler alakalı öznitelikler olarak belirlenir. Her özniteliğin ağırlığı, grafiğin spektrumuyla normalleştirilmiş kesim fonksiyonundan türetilmiş üç farklı fonksiyon $\varphi = \{\varphi_1, \varphi_2, \varphi_3\}$ kullanılarak hesaplanır. $\hat{f}_i$ normalize edilmiş ağırlıklı öznitelik vektörünü ve λ ve ϵ sırasıyla özvektör (eigenvector) ve özdeğerleri (eigenvalue) temsil etmek üzere, öznitelik ağırlıklandırma fonksiyonları aşağıdaki denklemlerde verilmektedir.

$$\varphi_1(F_i) = \hat{f}_i^T L \hat{f}_i = \frac{\hat{f}_i^T L \hat{f}_i}{\hat{f}_i^T \bar{D} \hat{f}_i}; \quad (2.4)$$

$$\varphi_2(F_i) = \frac{\sum_{j=1}^{n-1} \alpha_j^2 \lambda_j}{\sum_{j=1}^{n-1} \alpha_j^2} = \frac{\hat{f}_i^T L \hat{f}_i}{1 - \hat{f}_i^T \epsilon_0}; \quad (2.5)$$

$$\varphi_3(F_i) = \sum_{j=1}^{k-1} (2 - \lambda_j) \alpha_j^2 \quad (2.6)$$

SPEC algoritması, yukarıda verilen ağırlıklandırma fonksiyonlarını kullanarak her bir öznitelik vektörü $\hat{f}_i$ ' yi değerlendirir ve hesaplanan ağırlıklara göre öznitelik seçimi yapar.

SPEC yönteminin özel bir türevi olan diğer bir öznitelik seçim yöntemi Laplas skoru yöntemidir. Laplas skoru (He vd., 2006) ve geliştirilmiş halleri (Zhao vd., 2008; Yang vd., 2012) en bilinen benzerlik koruma yöntemleridir. Laplas skoru yaklaşımları, graf modelini temel alarak, benzerlik koruma gücüne göre özniteliklerin önemini değ-

erlendirmektir. SPEC'e benzer bir şekilde Laplas matrisi ve normalize edilmiş Laplas matrisi elde edildikten sonra her bir öznitelik için Denklem 2.7'de verilen öznitelik ağırlıklandırma fonksiyonu kullanılmaktadır.

$$F_i = \frac{\hat{f}_i^T L \hat{f}_i}{\hat{f}_i^T \bar{D} \hat{f}_i}; \quad \hat{f}_i = f_i - \frac{f_i^T \bar{D} 1}{1^T \bar{D} 1} 1 \quad (2.7)$$

Farklı benzerlik koruma kriterlerine dayanarak farklı benzerlik koruma algoritmaları önerilmektedir (Zhao vd., 2009; Cai vd., 2010). Zhao vd. (2009) farklı kümelerin en iyi şekilde ayrılmasına neden olacak öznitelikleri bulmayı hedefleyen M3FS yöntemini (Maximum Margin Feature Selection with Manifold Regularization) önermişlerdir. Özniteliklerin kümeleme üzerindeki etkisini ölçmek amacı ile her bir öznitelik için ölçek faktörü (scale factor) tanımlamışlardır. Cai vd. (2010) çok kümeli yapıyı koruma yaklaşımı ile MCFS (Multi-Cluster Feature Selection) olarak adlandırılan gözetimsiz bir öznitelik seçim yöntemi önermişlerdir. MCFS yöntemi verinin çok-kümelili yapısını en iyi koruyan öznitelikleri alakalı öznitelikler olarak değerlendirmektedir. Yöntem en alakalı öznitelik altkümelerini seçmek için verinin spektral analizi ile birlikte l_1 -düzenlenmiş regresyon modelini kullanır. Laplas grafinin üst özvektörlerini kullanarak, spektral kümeleme ile etiket bilgisini kullanmadan veri örneklerini kümeleyerek, verinin çoklu küme yapısını koruyan öznitelikleri bulmaktadır.

Son zamanlarda, benzerlik koruma yaklaşımlarından farklı olarak, sözde etiket tabanlı yaklaşımlar olarak adlandırılan kümeleme tabanlı öznitelik seçim yöntemleri de önerilmektedir. Veri örneklerine ait sınıf etiketleri bulunmadığından, bu yöntemler belirli kümeleme yöntemleriyle sözde etiketler üretmeye çalışmaktadır. Sözde etiketlerin tahmin edilmesinde rol oynayan öznitelikleri alakalı öznitelikler olarak seçmektedirler. Girdi verilerinin sınıf etiketinin doğrusal bir sınıflandırıcı tarafından tahmin edilebileceği varsayımı altında, Yang vd. (2011) tarafından UDFS (Unsupervised Discriminative Feature Selection) olarak adlandırılan bir sözde etiket tabanlı öznitelik seçimi yöntemi önerilmiştir. Yöntem, ayırmacı (discriminative) analiz ve $l_{2,1}$ -norm küçültme (minimization) yaklaşımını birlikte kullanmaktadır. $l_{2,1}$ -norm küçültme (minimization) yaklaşımı ile öznitelikler arasındaki korrelasyondan faydalanarak, öznitelik seçim sürecini kuvvetlendirmektedir. UDFS, her bir $x_i \in R^D$ örneği bir düşük boyutlu uzaya

$G_i \in R^s$ haritalayan bir doğrusal bir sınıflandırıcı $W \in R^{D \times s}$ olduğunu varsayar. $G_i = P_i' G$ eşitliğini sağlayan bir seçim matrisi $P_i \in \{0, 1\}^{n \times (k+1)}$ olduğunda, her bir örnek x_i için Denklem 2.8'de gösterildiği gibi bir yerel ayırıcı puan hesaplanır. $\tilde{X}$ matrisi, veri matrisi X 'in merkezlenmiş matris halini (simetrik ve eşkuvvetli) temsil etmektedir.

$$DS_i = tr[W' X' P_i \tilde{X}_i' (\tilde{X}_i \tilde{X}_i' + \lambda I_d)^{-1} \tilde{X}_i P_i' X W] \quad (2.8)$$

UDFS, X 'de bulunan tüm örnekler için en yüksek yerel ayırt edici puanı sağlayan W 'yi kullanma eğilimi gösterir. Ayrıca, yöntem $l_{2,1}$ -norm küçültme yaklaşımını yapısına ekleyip, Denklem 2.9'da verilen hedef fonksiyonunu minimize ederek öznitelik seçimini gerçekleştirir. α öğrenme modelinin seyrekliğini kontrol eden bir ayarlama parametresidir.

$$\min_{W'W=I_D} \sum_{i=1}^n tr[G_i' H_{k+1} G_i - DS_i] + \alpha ||W||_{2,1} \quad (2.9)$$

Yöntem, yukarıdaki denklemde verilen $l_{2,1}$ -norm tabanlı seyrek regresyon problemini optimize ederek özniteliklerin önem derecelerini bulmaya çalışmaktadır. Matris W 'nin $l_{2,1}$ -normu Denklem 2.10'da verildiği gibi hesaplanmaktadır.

$$||W||_{2,1} = \sum_{i=1}^m \sqrt{\sum_{j=1}^n W_{ij}^2} = \sum_{i=1}^m ||w_i||_2 \quad (2.10)$$

Diğer önerilen sözde etiket tabanlı yaklaşımlardan biri de NDFS (Nonnegative Discriminative Feature Selection) (Li vd., 2012) yöntemidir. NDFS, yöntemi negative olmayan spektral analiz ile öznitelik seçimini birlikte kullanır. Öncelikle örneklerle ait sözde küme etiketlerini öğrenmek için spektral kümeleme işlemi gerçekleştirir. Spektral kümeleme ile öğrenilen küme etiketleri, öznitelik seçimini yönlendirmek için kullanılır. Öznitelikler ve sözde etiketler arasında doğrusal bir dönüşüm olduğunu varsayarak, sözde sınıf etiketleriyle en alakalı olan öznitelikler seçilir. Yinemeli bir şekilde her aşamada bir küme gösterge matrisi ve buna bağlı bir öznitelik seçim matrisi oluş-

turulur. En ayırıcı özniteliklerin tespiti için öznitelik seçim matrisi üzerinde $l_{2,1}$ -norm küçültme problemi optimize edilmektedir.

Sözde etiket yaklaşımları bazı dezavantajlara sahiptir. Küme etiketleri genellikle yeterince doğru değildir. Sahte etiketlerde yer alan yanlış bilgiler, öznitelik seçimi daha da yanlış yönlendirebilmektedir. Ayrıca, amaç fonksiyonunda birçok farklı parametre kullanılmaktadır ve bu parametreler için en uygun değerlerin gözetimsiz ortamlarda tahmin edilmesi zaman isteyen bir süreçtir. UDFS ve NDFS yöntemleri, verilerin genellikle ideal olarak temiz olmadığı, aykırı değerler (outlier) ve gürütülü (noisy) verilerin olduğu veri setlerinde başarısız olabilmektedir. UDFS ve NDFS'nin hesaplama karmaşıklığından dolayı, örneğin metin verileri ve genetik veriler gibi yüksek boyutlu veriler için çoğu zaman bu yöntemler istenilen performansa ulaşamamaktadırlar.

2.3.4. Öznitelik Çıkarımı

Öznitelik çıkarımı, bazı işlevsel haritalama fonksiyonları ile orijinal özniteliklerden yeni bir öznitelik uzayı oluşturma işlemidir. Veriye ait $F_1, F_2, \dots, F_n$ olmak üzere n adet öznitelik olduğu varsayıldığında, öznitelik çıkarımı işlemi sonucunda $m < n$ olmak üzere veriye ait yeni bir öznitelik uzayında m adet yeni öznitelik $B_1, B_2, \dots, B_m$ elde edilmektedir. Öznitelik çıkarım yöntemleri, öznitelik uzayını yeni bir uzaya yansıtmak için bir haritalama fonksiyonu $B_i = \phi_i(F_1, F_2, \dots, F_n)$ kullanmaktadırlar.

Öznitelik çıkarımı yaklaşımlarının hedefi, bazı performans ölçütlerine göre bir miktar dönüşümle minimum bir dizi yeni öznitelik oluşturmaktadır. Dönüşümler için farklı haritalama fonksiyonları kullanılabilir. Genel olarak, dönüşümler doğrusal veya doğrusal olmayan dönüşümler halinde kategorize edilmektedir.

Öznitelik çıkarım yaklaşımı genel olarak boyutsallık azaltma problemini adreslemektedir. Öznitelik seçim yöntemlerinde olduğu gibi özniteliklerin bir alt kümesini seçmek yerine, bu yaklaşım öznitelikleri, orijinal verinin bilgisini mümkün olduğunca muhafaza ederek, daha düşük bir boyutlu yeni bir uzaya haritalar. Boyutsallık azaltma yöntemleri, öznitelik seçim yöntemlerine kıyasla daha çok gözetimsiz çalışan yöntemleri içermektedir.

2.3.5. Gözetimsiz Öznitelik Çıkarım Yöntemleri

En yaygın kullanılan öznitelik çıkarım yöntemlerinden biri bir boyut azaltma yöntemi olan, PCA olarak adlandırılan Temel Bileşen Analizi yöntemidir (Principal Component Analysis) (Kramer, 1998; Shaw, 2009). PCA, veriyi geometrik olarak asıl bileşenler adı verilen daha küçük boyuttaki yeni öznitelik uzayına yansıtarak boyut-sallığı azaltmaktadır. Bu yöntemin temel hedefi, orjinal niteliklerin doğrusal kombinasyonlarından oluşan, birbirlerine dik ve verilerdeki maksimum varyasyon miktarını yakalayan temel bileşenleri yani yeni öznitelikleri ortaya çıkartmaktır. PCA bir öz ayrıştırma (eigen decomposition) gerçekleştirmeye dayanan ve bu nedenle spektral boyut azaltma olarak bilinen bir öznitelik çıkarım yöntemidir. Geleneksel olarak, PCA yöntemi genel olarak kovaryans matrisi veya korelasyon matrisi olmak üzere kare simetrik bir matris üzerinde gerçekleşir. Kare matris üzerinde öz ayrıştırma uygulanarak öz vektörler (eigen vector) ve bunlara karşılık gelen öz değerler (eigen values) elde edilir. Öz vektörler öz değerlerine göre azalan düzende sıralanır ve ilk k öz vektörleri seçilir. Böylelikle orijinal n boyutlu veri k boyutlu yeni bir uzaya yansır. Matematiksel olarak ifade edilir ise, $n \times n$ 'lik bir C matrisi ve sıfır olmayan bir vektör q için Denklem 2.11'i sağlayan λ değerleri C 'nin özdeğerleri olarak tanımlanmaktadır. Aşağıdaki denklemi sağlayan q vektörleri de özvektörleri temsil etmektedir.

$$(C - \lambda I_n)q = 0 \quad (2.11)$$

C 'nin sıfır olmayan özdeğerlerinin sayısı, C 'nin maksimum doğrusal olan bağımsız sütun veya satır sayısını ifade eden $rank(C)$ olarak temsil edilir. Eğer C matrisi n adet doğrusal olarak bağımsız olan özvektöre $[q_1, q_2, \dots, q_n]$ sahip ise, C aşağıdaki denklemde verildiği şekilde üç matrisin çarpımı olarak ifade edilebilir. C 'nin özdeğerleri, Λ köşegen girdileri azalan sırada olacak şekilde $(\lambda_1, \lambda_2, \dots, \lambda_n)$ olarak temsil edilir. $Q = [q_1, q_2, \dots, q_n]$ ise C 'nin özvektörlerinin matrisidir. Bu işlem, C 'nin öz ayrıştırması olarak bilinir.

$$C = Q\Lambda Q^{-1} \quad (2.12)$$

Özayırıştırma işlemi bir kovaryans matrisi üzerinde gerçekleştirilebilir. m veri örnek- lerinin sayısı ve n özniteliklerin sayısı olmak üzere bir $m \times n$ 'lik A matrisi verildiğinde, bir kare matris olan kovaryans matrisi $Cov(A)$ Denklem 2.13'de olduğu gibi hesaplanır. n örnekli bir yapıda x ve y olmak üzere iki rastgele değişkeni olan bir örneğin kovaryansı $Cov(x, y)$, $\bar{x}$ ve $\bar{y}$ değişkenlerinin ortalamasını ifade etmektedir.

$$Cov(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \quad (2.13)$$

Kovaryans matrisine uygulanan özayırıştırma işleminden sonra, orjinal veri matrisi A , AQ 'e eşit bir diğer matris olan A' dönüştürülür. A' matrisinin her sütunu orjinal özniteliklerin doğrusal bir birleşimidir ve bu sütunlar asıl bileşenler olarak adlandırılır. Burada, i 'ninci yeni özniteliğin varyansı λ_i 'ye ve yeni özniteliklerin varyansının toplamı orijinal özniteliklerin varyansının toplamına eşit olacaktır. $Cov(A)$ 'nın özvektörleri sırasıyla yeni bir ortogonal eksen kümesi tanımlar ve en büyük özdeğerle ilişkilendirilen özvektör, verilerin en fazla varyansa sahip olduğu yönü belirtir.

PCA yöntemi yeni ve daha bilgilendirici öznitelikler bulmak için yararlı olan nispeten basit, parametrik olmayan bir yöntemdir ve düşük varyanslı öznitelikleri elimine ederek boyutsallığın azaltılmasında kullanılmaktadır. Asıl bileşenler birbirine dik olduğundan, her asıl bileşen, diğer asıl bileşenlerle ilişkili değildir diğer bir deyişle gereksiz bilgi içermez. PCA, mümkün olduğunca az asıl bileşen ile değişkenler arasındaki varyasyonun en yüksek yüzdesini muhafaza etmeye çalışır. Bu nedenle, çoğu zaman ilk birkaç asıl bileşen toplam varyansın büyük bir yüzdesini oluşturur ve bu durum verinin oldukça düşük boyutta temsil edilebilmesini sağlar.

PCA'ye benzer diğer bir yöntem ise SVD olarak bilinen Tekil Değer Ayrışımı (Singular Value Decomposition) yöntemidir (Demmel, 1997; Berry vd., 1999). SVD, kare olmayan matrisler için özvektörler mevcut olmadığından herhangi bir $m \times n$ boyutlu matrise uygulanabilen öz ayırıştırmanın genelleştirilmiş bir halidir. Matematiksel olarak ifade edilir ise, SVD $m \times n$ boyutlu bir matrisin aşağıdaki denklemde verilen üç matris çarpımının çarpanlara ayrılması işlemidir. Denklem 2.14'de verilen U matrisi $m \times m$ 'lik sol tekil vektörleri, Σ matrisi $m \times n$ 'lik köşegen üzerinde negatif olmayan gerçek sayılardan oluşan dikdörtgen köşegen matrisi ve V matrisi $n \times n$ 'lik sağ tekil vektörleri tem-

sil eder. Σ 'nın köşegen girdileri $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$, A matrisinin tekil değerleridir.

$$A = U\Sigma V^T \quad (2.14)$$

Tekil değerlerin hesaplanması işlemi aslında AA^T ve $A^T A$ matrislerinin özvektör ve özdeğerlerinin bulunması işlemi ile eşitlik göstermektedir. Burada, aşağıdaki denklemlerde görüldüğü üzere AA^T 'nin özvektörleri U matrisinin sütunlarını oluştururken, $A^T A$ 'nin özvektörleri de V matrisinin sütunlarına karşılık gelir. Bu durumda, Σ 'nın içindeki tekil değerler AA^T ve $A^T A$ matrislerinden gelen özdeğerlerin kare kök değerleri olmaktadır.

$$AA^T = (U\Sigma V^T)(V\Sigma^T U^T) = U\Sigma^2 U^T \quad (2.15)$$

$$A^T A = (V\Sigma^T U^T)(U\Sigma V^T) = V\Sigma^2 V^T \quad (2.16)$$

PCA ve SVD yaklaşımlarının arasındaki fark PCA yöntemi kovaryans matrisi hesabında her değişkenin ortalama değerini ortadan kaldırırken, SVD orjinal veriyi kullanmaktadır. Bu nedenle, özellikle seyrek veriler ve ortalama değerin önemli olduğu veriler için SVD yöntemi daha çok tercih edilmektedir.

Öznitelik seçiminde diğer en bilinen yaklaşımlardan biri NMF olarak bilinen Negatif Olmayan Matris Ayrıştırma (Non-Negative Matrix Factorization) yöntemidir (Paatero ve Tapper, 1994). NMF, birçok nitelik olduğunda ve nitelikler belirsiz veya zayıf tahmin edilebilir olduğunda başarılı bir çıkarım yapabilmektedir. NMF, nitelikleri birleştirerek anlamlı desenler, konular veya temalar üretebilir. Yöntem, bir negatif elemanı olmayan $A \in \mathbb{R}^{m \times n}$ matrisinin, $k \ll \min(m, n)$ olmak üzere negatif elemanları olmayan $W \in \mathbb{R}^{m \times k}$ ve $H \in \mathbb{R}^{k \times n}$ matrisi olarak çarpanlarına ayrılması işlemini uygular.

$$A \approx WH \quad (2.17)$$

Negatif olmayan W ve H matris faktörlerini bulmak için NMF yöntemi ile Denklem 2.17'de verilen bir doğrusal olmayan optimizasyon problemi çözülmektedir. Bu-

rada kullanılan $||.||$, matris elemanlarının mutlak karelerinin toplamının karekökü- nün (Frobenious norm) hesaplanması demektir.

$$\min_{W,H} f(W, H) = \frac{1}{2} ||A - WH||^2 \quad (2.18)$$

Matris W ve H 'ın hesaplanması için NMF yönteminin kullandığı birden fazla algoritma mevcuttur. Bu algoritmalar genelde Çarpımsal Güncelleme (Multiplicative Update) (Lee ve Seung, 2001), Alternatif En Küçük Kareler (Alternating Least Squares) (Paatero ve Tapper, 1994) ve Gradyan İniş (Gradient Descent) (Lin, 2007) algoritmaları olarak üç yaklaşıma ayrılmaktadır. NMF algoritmalarının çoğu W ve H çarpanlarını rastgele seçer. NMF algoritmalarının yakınsaması ve yakınsama kalitesi çarpanların başlangıç değerlerine bağlı olabileceğinden, algoritmalar genellikle farklı başlangıç çarpanları ile birkaç kez tekrarlanır. Başlangıç çarpanları tanımlandıktan sonra, W ve H matrislerinin bulunması için bu algoritmalar tanımlanmış bir iterasyon miktarınca bu matrislerin değerlerini günceller. Bu algoritmalar en genel olanı Çarpımsal Güncelleme algoritmasıdır. Bu algoritma, n .ci iterasyonda Denklem 2.19 ve 2.20'de verilen şekilde W ve H matrislerini güncellemektedir.

$$H_{[i,j]}^{n+1} = H_{[i,j]}^n \frac{((W^n)^T V)_{[i,j]}}{((W^n)^T W^n H^n)_{[i,j]}} \quad (2.19)$$

$$W_{[i,j]}^{n+1} = W_{[i,j]}^n \frac{(V(H^{n+1})^T)_{[i,j]}}{(W^n H^{n+1} (H^{n+1})^T)_{[i,j]}} \quad (2.20)$$

Alternatif En Küçük Kareler yaklaşımı bir matrisi sabitler ve diğerini iyileştirir. Denklem 2.21 ve 2.22'de W ve H matrisleri Alternatif En Küçük Kareler yaklaşımına göre güncellenmektedir.

$$W^{n+1} = \arg \min_{(W \geq 0)} f(W, H_n) \quad (2.21)$$

$$H^{n+1} = \arg \min_{(H \geq 0)} f(W_{n+1}, H) \quad (2.22)$$

Denklem 2.23 ve 2.24’de ise Gradyan İniş algoritmasına göre matrislerin nasıl güncellendiği gösterilmektedir.

$$W^{n+1} = \max(0, W_n - \alpha \Delta_W f(W_n, H_n)) \quad (2.23)$$

$$H^{n+1} = \max(0, H_n - \alpha \Delta_H f(W_n, H_n)) \quad (2.24)$$

2.3.6. Öznitelik Oluşturma

Öznitelik oluşturma, eksik bilgilerden kaynaklı öznitelikler arasındaki korrelasyonun tam olarak ortaya çıkartılamadığı durumlarda eksik bilgileri keşfetme amacı ile uygulanan bir işlemdir (Motoda ve Liu, 2002). Veri örneklerine ait ek öznitelikler çıkartarak veya yaratarak örneklerin öznitelik alanı zenginleştirilir. Yeni oluşturulan tüm öznitelikler, orjinal öznitelikler uzayında tanımlanır, bu nedenle öznitelik oluşturma sırasında öznitelik uzayına yeni bir öznitelik eklenmez. Öznitelik oluşturmada temel hedef, öğrenme algoritmasının başarısını artıracak şekilde örneklerin öznitelik vektörlerini anlamlı bir şekilde değiştirilmesidir. Bu durum, bir örneğe ait özniteliğin sayısının arttırılmasını bazen ise azaltılmasını gerektirebilir (Sondhi, 2009). Öznitelik oluşturma yöntemleri, öğrenme algoritmasının doğruluğunu artırmak, anlaşılabilirliği artırmak, verilere ait daha doğru kümeler elde etmek ve gizli bilgileri ortaya çıkarmak gibi bir çok durum için kullanılabilir.

Sezgisel olarak, üstel olarak artan birçok orjinal öznitelik kombinasyonu olabilir ve tüm kombinasyonlar gerekli ve faydalı olmayabilir. Bu durumda, gereksiz ve alakasız öznitelikleri kaldırmak için öznitelik oluşturma süreci içerisinde genellikle öznitelik seçim yaklaşımları kullanılmaktadır. Böylelikle, öznitelik oluşturma sürecinde, yüksek boyutsallığa sahip veriler için daha düşük boyutta temsiller elde edilebilmektedir. Farklı öznitelik oluşturma yöntemleri, farklı süreçlere sahip olmakla birlikte, kavramsal olarak herhangi bir öznitelik oluşturma yönteminin genellikle aşağıda belirtilen süreçleri sırasıyla yerine getirmektedir (Sondhi, 2009):

- İlk öznitelik alanı F^0 girdi olarak alınır.

- F^0 , yeni bir öznitelik alanı oluşturmak için F^N 'e dönüştürülür.
- F^N 'den bir F^i öznitelik alt kümesi seçilir.
 - Bazı fayda kriterlerine dayanarak tahmin görevi için F^i 'nin yararlılığı tespit edilir.
 - Sonlandırma kriterlerine ulaşılır ise 3. adıma geri dönülür.
 - Sonlandırma kriterlerine ulaşılmaz ise $F_T = F^i$ olarak atanır.
- F_T yeni oluşturulan öznitelik uzayı olur.

Öznitelik oluşturma sürecinde, öznitelik dönüşümü için yaygın olarak kullanılan yaklaşım, orjinal öznitelik değerlerine bir dizi işleç uygulamaktır. İşleç seçimi, alan bilgisine ve öznitelik türlerine göre değişebilir. Yaygın olarak kullanılan operatörlerden bazıları şunlardır:

- Mantıksal işleçler: Ve, Yada, Olumsuzlama vb.
- Nominal işleçler: Kartezyen Çarpımı, MofN vb.
- Matematiksel İşleçler: Minimum, Maksimum, Toplama, Çıkarma, Çarpma, Bölme, Ortalama, Eşitlik, Eşitsizlik vb.

Öznitelik dönüşümündeki büyük zorluk, veri için doğru işleç grubunun seçilmesinde ve uygun şekilde uygulanmasında yatmaktadır. İşleçleri isteğe bağlı seçmek, performansı düşürebilecek çok sayıda alakasız öznitelik üretebilir. İşleçler genellikle yinelemeli olarak uygulanır. Öznitelik oluşturma yöntemleri, öznitelik alanını tanımlamak ve aramak için kullandıkları tekniklere göre sınıflandırılır.

2.3.7. Öznitelik Oluşturma Yöntemleri

İlk önerilen yöntemler büyük ölçüde karar ağaçlarına dayanmaktadırlar. Son zamanlarda ise daha çok Mantıksal Programlama ve Genetik Programlamaya dayalı yöntemler önerilmektedir. FRINGE (Bagallo ve Haussler, 1990) ilk önerilen öznitelik oluşturma yöntemlerinden biridir. Orjinal öznitelikleri kullanarak bir karar ağacı oluşturur ve bu ağacı faydalı yüksek seviyeli öznitelikler bulmak için analiz eder. Mevcut

öznitelik alanındaki olumsuz öznitelik çiftlerini öznitelik alanından çıkarır ve mantıksal işlemler (ve, olumsuzlama) ile öznitelik alanlarını birleştirilerek yeni öznitelikler oluşturulur. Elde edilen yeni öznitelikleri orjinal öznitelikler ile birleştirip, bu yeni uzayda karar ağacını tekrar oluşturur. Bu süreç, hiçbir yeni aday öznitelik kalmayana kadar yinelenir. Yöntemin temel amacı, öznitelik oluşturma süreci sırasında, başlangıç uzayını mantıksal işlemler ile genişleterek DNF temsillerini öğrenmektedir. FRINGE, daha önce oluşturulan hipotez karar ağacına dayanarak yeni öznitelikler inşa ettiği için hipotez odaklı bir öznitelik oluşturma yaklaşımına sahiptir. Karar ağacı kavram öğrenmesini kullanarak Boole özniteliklerini oluşturan bir diğer yöntem CITRE (Matheus ve Rendell, 1989) yöntemidir. FRINGE'den farklı olarak bazı farklı mantıksal işlemler (ve, yada) kullanmıştır. Ayrıca, CITRE sadece ağacın yapraklarında değil, ağacın tamamındaki temsilleri tarar. Diğer bir algoritma FICUS (Markovitch ve Rosenstein, 2002) ise daha esnek bir şekilde farklı ekleme ve çıkarma fonksiyonları (birleştir, ekle, değiştir vb.) kullanarak öznitelik uzayını zenginleştirmektedir. Algoritma, en iyi üretilen öznitelik kümelerini korur ve bu kümeleri yinelemeli olarak geliştirir. Her yineleme sırasında, algoritma tanımlanmış öznitelik alanı üzerinde bir tarama yapar ve mevcut özniteliklere yapıcı işlemler uygulayarak yeni öznitelikler oluşturur.

Öznitelik oluşturma yaklaşımlarında karar ağacı yaklaşımından farklı olarak sıklıkla kullanılan bir diğer yaklaşım genetik programalama yaklaşımıdır. Genetik programlama, bir birey popülasyonu ile başlayan, bazı uygunluk fonksiyonlarına (fitness function) dayanarak bunları değerlendiren ve yüksek puan alan bireylere bir dizi mutasyon ve geçit operatörü uygulayarak ve düşük puan alanları ortadan kaldırarak yeni bir popülasyon oluşturan evrimsel algoritma tabanlı bir tekniktir. Krawiec (2002) tarafından makine öğrenmesi sınıflandırıcılarının performansını artırmak için genetik programlama (GP) tabanlı bir öznitelik oluşturma prosedürü önerilmiştir. Evrimsel süreçte, orjinal öznitelik uzayından yeni öznitelikler türetilir ve en iyi performans gösteren öznitelik alt kümeleri muhafaza edilir. Mutasyon operatörü (toplama, çıkarma, bölme vb.), bir bireyin alt ifadesini rastgele seçer ve onu bir başka rasgele oluşturulmuş alt ifadeyle değiştirir. Çaprazlama operatörü, rastgele, bir çift ve bu çiftlere ait iki alt ifadeyi seçer, ve sonrasında bu ifadeleri çiftler arasında yer değiştirir. Diğer benzer bir çalışma Otero vd. (2003) tarafından sunulmuştur. Bu çalışmada, bilgi kazancını (information gain) bir kriter olarak kullanarak, öznitelikler için bir puan hesabı yapmaktadır. Yüksek puana

sahip öznitelikler, orjinal öznitelik kümesine eklenmiştir. Tran vd. (2017) tarafından yüksek boyutlu verileri sınıflandırmak için kullanılan algoritmaların performansını artırmak için CGPFC (Cluster-based GP Feature Construction) yöntemi önerilmiştir. Bu yöntem ile GP yaklaşımına öznitelik kümeleme yaklaşımı eklenerek, gereksiz öznitelikleri korelasyona bağlı olarak bir grupta kümelemeyi amaçlamışlardır. Öznitelik kümeleme yaklaşımı ile elde edilen öznitelik kümelerinin her birinden en başarılı öznitelikler seçilir ve tek bir yeni üst düzey öznitelik oluşturmak için kullanılır.

Son zamanlarda, k-means kümeleme algoritmalarının performansını artıracak şekilde gözetimsiz öznitelik oluşturan yöntemler de önerilmektedir. Lensen vd. (2017) tarafından önerilen çalışmada, her bir birey orjinal özniteliklerinin sadece tek birinin yapılandırıldığı birden fazla öznitelik temsillerine sahip olacak şekilde bir çoklu-ağaç gösterimi ile temsil edilmektedir. Öznitelik yapılandırması için toplama, çıkarma, minimum, maksimum vb. standart aritmetik işlemler kullanılmaktadır. Bu öznitelik temsillerinin alt kümelerini seçmek için kullanılan uygunluk fonksiyonu küme içi varyasyonu, diğer bir deyişle aynı kümedeki örnekler arasındaki mesafeyi en aza indirmeyi hedeflemektedir.

3. AĞ BAĞLANTILI VERİ

Bu bölümde, ilk olarak ağ bağlantılı veriye özel kavramlara değinilecek ve ağ bağlantılı veri tanımlanacaktır. Devamında, ağ bağlantılı veri yapısı için geliştirilmiş en bilinen öznitelik seçim ve çıkarım yöntemlerine yer verilecektir.

3.1. Ağ Yapısı

Ağ yapısı, insanlar, nesneler ve durumlar arasındaki etkileşimler ile ortaya çıkan öge ve bağlantılardan oluşan ilişki veya hiyerarşinin gözlemlenebildiği bir yapıdır. Bizlerin de bireyler olarak bir sosyal ilişkiler ağının nesnesi olduğumuz düşünülür ise her yer ve her alanda ağların doğal olarak ortaya çıktığı anlaşılmaktadır. Ağlara ait nesneler, elektrik güç ızgaraları, internet, otoyollar, metro sistemleri veya sinir ağlarında olduğu gibi somut nesneler yada tanışma ağları veya sosya ağlarda olduğu gibi soyut bir alanda tanımlanan varlıklar olabilir.

Ağ analizi, çizge teorisi olarak bilinen ayrık matematik dalı içerisinde yer almaktadır. Leonhard Euler tarafından 1736'da (Euler, 1736) Königsberg köprüsü problemine (Rusya'nın Königsberg şehrinde bulunan Pregel nehrinin ayırdığı dört toprak parçasını birleştiren yedi köprüünün her birini bir kez tam olarak geçen bir gezi bulma) bir çözüm yayınlamasından sonra çizge teorisi alanı yeni problem ve sunulan çözümler ile birçok heyecan verici gelişmelere tanık olmuştur. Örneğin; bir boru şebekesi içerisindeki kaynaktan bir lavaboya birim zamandaki maksimum akış miktarının ne olacağı (Goldberg ve Tarjan, 1988) veya bir harita üzerindeki bölgelerin, komşu bölgeler farklı renklerle boyanarak nasıl boyanabileceği (Leighton, 1979) pratikte karşılığı olan bazı sorulara çizge teorisi ile çözüm üretilmeye çalışılmıştır.

Son yıllarda, yapısı zamanla evrilebilen, ne tamamen düzenli ne de tamamen rastgele bağlantılardan oluşan yüzlerce, binlerce yada milyonlarca ögeden oluşan *karmaşık ağlar* olarak adlandırılan yeni bir ilgi ve araştırma alanı oluşmuştur. Üzerinde son zamanlarda sıklıkla çalışılan bu ağlar, kafes ve rastgele çizge yapıları gibi matematiksel ağ modellerinin çoğunda gözlemlenmeyen yapısal özellikler göstermektedir. Bu özellikler önemli ve çözümlenmesi zor olan özellikler olarak bilinmektedir. Bu ağlar,

ağ seyrekliği, düğüm derece dağılımı, ortalama yol uzunluğu ve kümeleme katsayısı gibi özellikler ile rastgele çizge yapılarından farklılaşmaktadır (Wang ve Chen, 2003; Akal, 2014). Örneğin; rastgele ağlar her düğümün aynı sayıda kenara sahip olduğu homojen bir yapıda gözlenirken (Erdős ve Rényi, 1960), karmaşık ağlarda çoğu düğümün çok az sayıda kenara, ağdaki az sayıdaki düğümün ise çok sayıda kenara sahip olduğu homojen olmayan bir yapı gözlemlenmektedir (Barabási vd., 1999). Bunların yanı sıra, karmaşık ağlar genellikle bir topluluk yapısı veya bir hiyerarşik yapı içermektedirler. Birçok kompleks ağın, küçük dünya modeliyle tanımlanabilir olduğu belirtilmektedir Watts ve Strogatz (2018). Watts-Strogatz olarak da bilinen bu model, her düğüme birkaç adımda diğer tüm düğümlerden erişilebildiğini göstermektedir. Bu ağlar üçgen yapılara göre oluşmuştur ve düğümlerin derece dağılımı Poisson dağılımına uymaktadır.

Karmaşık ağların gerçek dünyadaki örneklerine *gerçek-dünya ağları* denilmektedir. Sosyal ağlar, biyolojik ağlar, teknoloji ağları, bilgi ağları vb. ağlar gerçek-dünya ağları olarak tanımlanmaktadır. Ağların matematiği üzerine yapılan son çalışmalar, gerçek-dünya ağların özelliklerinin gözlemlenmesi ve onları ait yeni modellerin geliştirilmesine odaklanmaktadır. Bu nedenle gerçek-dünya ağlarına ait ağ verileri karmaşık ağları analiz etmek için sıklıkla kullanılırlar Watts ve Strogatz (2018) tarafından sunulan çalışma günümüzde bu alanda yapılan son çalışmaların en önemli kaynağıdır. Çalışmada, aynı zamanda farklı ağ türlerinden gelen verileri eşzamanlı olarak inceleyerek birçoğunun ortak olduğu özellikler ve bu özellikleri yansıtan matematiksel gelişmelere vurgu yapılmıştır.

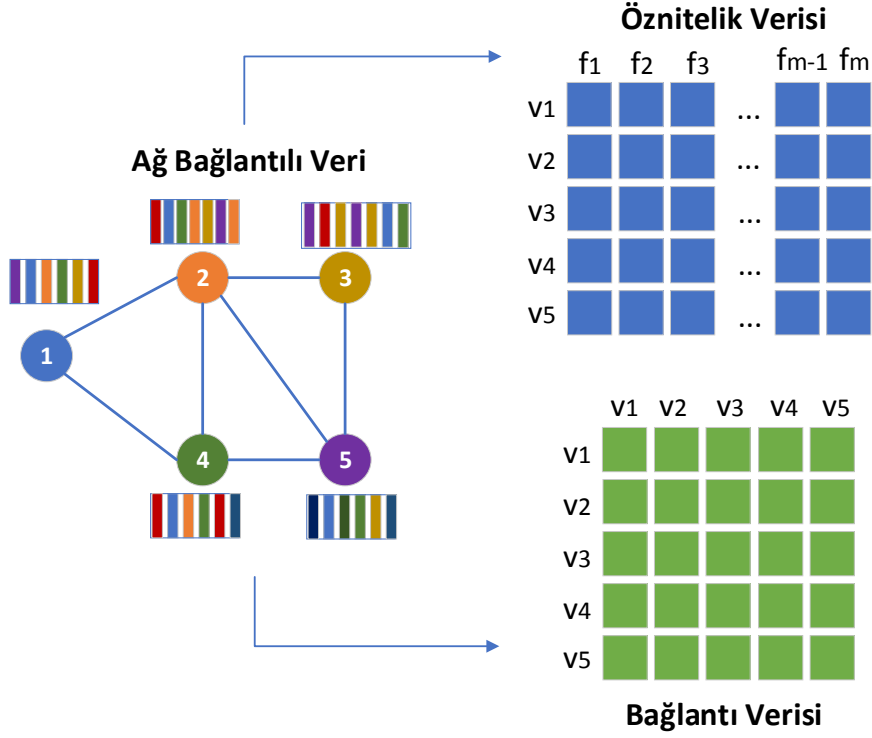
Son zamanlarda, karmaşık ağlar temelinde gerçek-dünya ağlarına ait birçok problem ve bu problemler için çözümler önerilmektedir. Bu problemler, ağ örnekleme, bağlantı tahmini ve ağ parçalama olarak örneklendirilebilir. Örneğin; birçok uygulamanın var olan problemlerine pratikte çözüm önermek için ağdan belli bir miktarda örneğin alınması gerekir (Rezvanian ve Meybodi, 2015). Özellikle, geniş ölçekli yapılarından dolayı WWW (World Wide Web) veya Facebook sosyal ağını tamamen modellemek pratik değildir. Bunun yerine, mümkün olduğunca orjinaline benzer daha küçük bir ağı örnek olarak almak ve bu ağ üzerinde çözüm üretmek gerekir. Bu problem çizge örnekleme (graph sampling) olarak bilinir. Çizge örnekleme bazı önemli zorluklara sahip bir problemdir. Örneğin; belirli bir ağ için en iyi örnekleme yöntemini

bulmak, örneklenecek çizge için örnek boyutuna karar vermek, örneklenen çizgenin kalitesini ölçmek vb. problemler çizge örnekleme problemini daha zor bir problem haline getirebilmektedir (Shakibian ve Charkari, 2016). Bağlantı tahmini problemi, ağa ait iki nesnenin arasında bir bağlantı oluşma olasılığını incelemektir. Karmaşık ağların kapsamında diğer önemli bir problem olan bağlantı tahmini, bir öneri sistemindeki müşterilere ürünleri önermek yada bir protein ağında protein-protein etkileşimini tahmin etmek gibi problemlerin çözümü için önemli bir hale gelmektedir (Martínez vd., 2017). Diğer önemli bir problem olan ağın parçalanması problemi, karmaşık ağların temel problemlerinden biridir. Örneğin; pazarlama ve öneri sistemleri için benzer kullanıcıları bulmak veya sosyal ağlarda benzer ilgi alanlarına sahip kullanıcıları bulmak gibi problemler çizge parçalama problemleri olarak karşımıza çıkmaktadır (Newman, 2006).

3.2. Ağ Bağlantılı Veri

Ağ bağlantılı veri, ağlar üzerinden toplanan veri olarak tanımlanır. Bu veri, Twitter ve Facebook gibi sosyal ağlar (arkadaşlıklar ile bağlı kullanıcılar) veya biyolojik ağlar (protein etkileşimleri) gibi birçok platformda ortaya çıkmaktadır (Li ve Liu, 2017). Her nesnenin diğer nesnelerden bağımsız olarak belli bir öznelilikler kümesi ile tanımlandığı geleneksel veriden farklı olarak, ağ verisindeki varlıklar birbirleriyle farklı bağlantı türleri ile ilişkili olan korrelasyonlu bir veri türüdür. Ağ bağlantılı veri, ağ içerisindeki her nesneyi tanımlayan öznelilik kümelerinden ve nesneler arasındaki ilişkiyi tanımlayan bağlantı yapısından oluşmaktadır. Bağlantı yapısı, daha çok kompleks ağ yapısı olarak modellenenbilmektedir. Ağın topolojisi üzerine geliştirilen matematiksel modellere, bir de ağ içerisinde paylaşılan anlamsal bilgiler eşlik eder. Bu durum, yeni bir veri madenciliği çalışma alanının gelişimini sağlamıştır.

Karmaşık ağların matematiksel ifadesi için çizge teorisi kullanılmaktadır. Çizge teorisine göre, karmaşık ağdaki her bir nesne çizgenin bir düğümünü, nesneler arasındaki her bir ilişki ise bu nesnelere karşılık gelen düğümler arasındaki bir kenarı temsil etmektedir. V düğüm kümesi ve E kenarlar kümesi olmak üzere bir çizge (graf) $G = (V, E)$ olarak taemsil edilmektedir. Çizge yapısı bir ağın ağ topolojisini temsil eder. Ağ



Şekil 3.1: Ağ bağlantılı veri yapısına ait örnek gösterim.

bağlantılı veriler ise düğümleri özniteliklere sahip bir çizge olarak tanımlanır. Matematiksel bir ifade ile ağ bağlantılı veriler $G = (V, E, F)$ çizgesi ile temsil edildiğinde, $V = \{v_1, v_2, \dots, v_N\}$ bağlantılı verinin nesnelerinin çizge temsiliinde düğümlerini, $E \subseteq V \times V$ ise çizgenin bağlantılarını diğer bir deyişle nesneler arasındaki ilişkileri, ve $F = \{f_1, f_2, \dots, f_D\} \in R^{N \times D}$ ise nesnelerin özniteliklerini temsil etmektedir. f_{v_i}, v_i düğümünün D boyutsallığında olan öznitelik vektörü anlamına gelir. v_i ve v_j düğümleri arasındaki ilişki ise e_{v_i, v_j} olarak temsil edilmektedir. G çizgesinin bağlantı yapısı L olmak üzere, eğer v_i ve v_j düğümleri arasında bir bağlantı var ise $l_{v_i, v_j} = 1$, diğer durumda 0 değerindedir.

Şekil 3.1’de ağ bağlantılı verinin öznitelik verisi ve bağlantı yapısı örneklendirilmiştir. Beş farklı ağ nesnesinin arasındaki ilişkiler bağlantı verisi olarak tanımlanmakta ve bu bağlantıları temsil eden çizge yapısı bir komşuluk matrisi ile ifade edilmektedir. Her bir nesnenin öznitelik vektörlerinden oluşan öznitelik matrisi ise öznitelik verisi olarak tanımlanmakta ve öznitelik matrisi ile ifade edilmektedir.

3.2.1. Ağ Bağlantılı Verinin Zorlukları

Ağ bağlantılı veriler, bağımsız ve aynı şekilde dağıtıldığı varsayılan düz verilerden farklılaşmaktadır. Fakat, öğrenme ve madencilik algoritmalarının çoğu, verilerin bağlantı bilgisini görmezden gelirken, çizge algoritmalarının çoğu öznitelik bilgisini görmezden gelir. Öte yandan, bağlantılı veriler aşağıdaki belirtildiği gibi dört açıdan zordur.

- Nesneler, yüksek boyutsallıkla karakterize edilir. Bu nedenle, birçok öğrenme görevi, boyutsallık laneti nedeniyle hesaplamalı olarak zordur veya etkisiz hale gelir (Liao vd., 2017).
- Bağlantılı verilerdeki öznitelikler çoğunlukla seyrek gözlenir. Bu seyreklik, öğrenme algoritmalarının benzerlik ölçümlerini ve hesaplama karmaşıklığını büyük ölçüde etkiler (Dhillon ve Modha, 2001).
- Bağlantılı verilerdeki nitelikler ve bağlantılar ilgisiz, gereksiz ve gürültülü olabilir. Bu nitelikler ve bağlantılar, öğrenme yöntemlerinin başarısız olmasına yada aşırı öğrenmesine neden olur (Tang ve Liu, 2013).
- Veri etiketleme, çok sayıda nesne için zaman alıcı ve pahalı bir iştir. Bu nedenle, gözetimsiz yöntemler çok önemli hale gelmektedir (Yang vd., 2012).
- Bağlantılı verinin öznitelik verileri ile bağlantı yapısı uygun bir şekilde (veriler arasındaki korelasyonu öğrenecek şekilde) entegre edilmelidir (Li vd., 2016).

Yukarıda belirtilen zorluklardan dolayı, geleneksel öznitelik seçim ve çıkarım algoritmalarından farklı olarak ağ bağlantılı veriye odaklanan yeni yöntemlere ihtiyaç duyulmaktadır.

3.2.2. Ağ Bağlantılı Veri Örnekleri

Ağ bağlantılı veriler, sosyal ağlar (social network), e-posta ağları (e-mail network), alıntı ağları (citation network), iş birliği ağları (collaboration network) ve internet sayfaları ağları (web network) gibi birçok farklı ağda gözlenir. Bu bölümde, en

yaygın ağ bağlantılı verilere yer verilecektir.

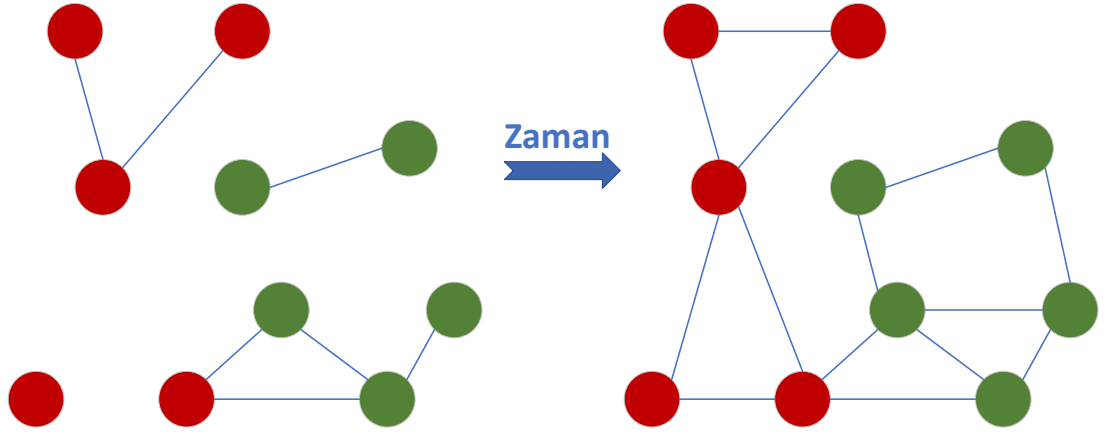
- *Sosyal Medya Verisi (Social Media Data)*: Sosyal medya verileri, bireylerin sosyal medya faaliyetlerinden toplanan tüm ham bilgileri ifade eder. Arkadaşlar, aile, iş arkadaşları veya müşteriler ile bağlantıda kalmak için kullanılan internet tabanlı sosyal medya sitelerinden (Facebook, Twitter, LinkedIn vb.) elde edilebilmektedir. Sosyal medya verileri, bireylere ait paylaşımları (tweet, yorum vb.), kişisel bilgileri, ilgi alanlarını, ve hatta tıklamalara (clicks) ait meta verileri içerir. Bu veriler, bireyler veya organizasyonlar gibi sosyal aktörlerden ve bu sosyal aktörlerin paylaştığı içerikten oluşan sosyal bir yapıya aittir. Bu sosyal yapı içerisindeki ilişkiler ve sosyal aktörler bir ağ topolojisi ile temsil edilirken, sosyal aktörleri tanımlayan veriler öznitelik verilerini oluşturur (Tufekci, 2014).
- *Haberleşme verisi (Information Data)*: Haberleşme verisi, e-posta veri kümesi içindeki e-posta iletişimine ait veridir (Bird vd., 2006). Haberleşme verisi, e-posta kullanarak mesaj alış-verişi ile etkileşim halinde olan bir grup arkadaş, bir şirketin çalışanlarını veya bir organizasyona ait müşterileri içerebilmektedir. Ayrıca, gönderilen veya alınan e-posta mesajlarına ait e-postanın gönderilme zamanı, başlık bilgisi, içerik bilgisi vb. bilgileri de içermektedir. E-posta verisine ait ağda her düğüm bir e-posta adresini ve iki düğüm arasındaki her bir kenar bu iki e-posta adresi arasındaki bir e-posta alış-verişini temsil eder. Bir e-posta ağı, bir e-posta sunucusunun genellikle günlük dosyalarından elde edilir.
- *İşbirliği verisi (Collaboration Data)*: Yayınlanan makalelere, dergilere veya kitaplara ait bilimsel işbirliği kapsayan bir veridir. Bir makalenin fikri ya da yöntemi, az ya da çok, belli referanslardan ilham alınarak oluşuyor ise, bu referanslar makalenin fikrinin veya yönteminin kaynakları ve nedenleri olarak değerlendirilebilir. Bu makalelere ait referanslar bir alıntı ağı oluşturmaktadır ve alıntı ağı içerisindeki bağlantılar nedensel ilişkiler olarak değerlendirilmektedir (Ding, 2011). Böyle bir ağda, her makale çizgede bir düğümü ifade eder. Eğer bir makale başka bir makaleden alıntı yapmış ise bu iki makale arasında, çizge de kenara karşılık gelen bir bağlantı oluşur. Bu veri türünde, her makalenin içerik bilgisi de yazarı tanımlayan bir öznitelik vektörü ile temsil edilir.
- *İnternet verisi (Internet Data)*: Geniş dünya ağı (WWW)'dan elde edilen özel-

likle internet sayfaları arasındaki bağlantıları ve içerikleri temsil eden bir ağ türüdür. Birbirine köprülerle bağlı internet sayfaları arasındaki bağlantılar bir çizgede kenarları temsil ederken, sayfalar ise çizgedeki düğümleri karşılık gelir. İnternet sayfalarında, bir sayfadan diğerine geçebilmek için köprüler kullanılmaktadır. Bir köprü içeren belge, kaynak belge olarak düşünüldüğünde, örneğin, Wikipedia veya Google gibi çevrimiçi bir sayfada, metindeki birçok kelime ve terim, okuyucuyu bu kelime ve terimlerin tanımlarına götüren köprüler halindedir. Bu durumda, kaynak sayfa ile sonraki sayfa arasında aslında nedensel ilişkiler olduğu düşünülmektedir. Her nedensel ilişki çizge de bir kenarı temsil ederken, kaynak belge ve köprü ile ulaşılan belge içerisindeki içerik verisi, internet verisine ait öznelik verisini temsil eder.

Sosyal medya verileri, çoğunlukla yapısız veri (unstructured data) formundadır (Shi vd., 2016). Önceden tanımlanmış bir veri modeline sahip olmayan veya ilişkisel tablolarda tutulmayan bir veridir. Sosyal medya verisi, metin ağırlıklı bir veri olmakla birlikte tarihler, sayılar ve semboller gibi veriler de içerebilir. Bu veri türü, veritabanlarında tutulan veya anlamsal olarak etiketlenmiş verilerle karşılaştırıldığında, analiz edilmesini zorlaştıran düzensizliklere ve belirsizliklere sahiptir. Ayrıca, sosyal medya verileri oldukça büyük miktarda gereksiz ve gürültülü veri içerebilmektedir. İşbirliği verisi de yapısız bir veri olarak değerlendirilir. Fakat, sosyal medya verisine kıyasla, daha düzenli bir metin bilgisi ve daha anlamlı ilişkiler içermektedir. Haberleşme verisi ve internet verisi, daha çok yarı-yapılı (semi-structured data) bir veri formuna sahiptir. Yarı-yapılı veri ilişkisel veritabanları veya diğer veri tablo biçimleriyle ilişkili veri modellerinin resmi yapısına uymayan, ancak yine de semantik unsurları ayırmak ve kayıtların hiyerarşilerini anlamak için, etiketler veya başka işaretler içeren yapısal bir veri biçimidir. Bu nedenle, kendini tanımlayan bir yapıya sahip olduğu söylenebilir. İnternet verisi genellikle XML biçiminde muhafaza edilir. XML'deki nitelikler nesne türleri olarak kabul edilebilir ve nesne örnekleri, niteliklerin içeriği analiz edilerek belirlenebilir. Nitelikler arasındaki bağlantılar nesne ilişkilerini temsil eder.

3.3. Homofili ve Sosyal Etki

Ağ bağlantılı verilerde, nesnelerin özelliklerinin yanısıra nesneler arasındaki ilişkilerde gözlemlendiğinden geleneksel (düz) veri yapılarından farklı bir yapıya sahiptir. Çoğu zaman, bireylere ait özellikler ağda rastgele dağılmaz ve çoğunlukla altta yatan ağ yapısı ile bağlantılıdır. Çok farklı ağlarda bile gözlemlenen bu durum, özellikle birbirine komşu olan düğümlerin özellikleri arasında bir korelasyon olduğunu göstermektedir (Fiore ve Donath, 2005; Fowler vd., 2011). Sosyal ağlarda ise bu durum, insanların kendilerine benzer olduğunu düşündükleri diğerleriyle ilişki kurma eğiliminde oldukları anlamına gelir. Benzerlikler, yaş grubu, cinsiyet, köken, hobiler vb. bir çok farklı özellikte ortaya çıkabilmektedir. Bireylerin kendilerine benzer olduğunu düşündükleri diğerleriyle ilişki kurma eğilimi *homofili* (homophily) olarak bilinilmektedir (McPherson vd., 2001). Şekil 3.2’de bir ağ yapısının zamanla homofoli eğilimine göre nasıl evrildiği örneklendirilmiştir. Aynı özelliklere sahip düğümler aynı renk ile gösterilmiştir. Zaman içerisinde, aynı özelliklere sahip aynı renkte olan nesneler arasında homofoli eğilimden kaynaklı olarak yeni bağlantılar oluştuğu görülmektedir.

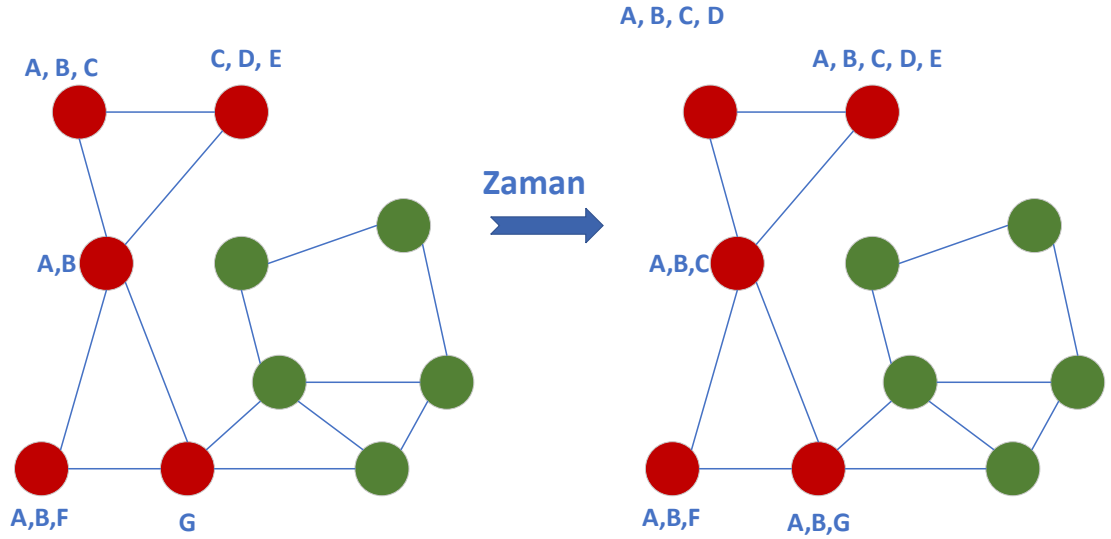


Şekil 3.2: Zamana göre ağ yapısında gözlenen homofili eğilimi.

Kossinets ve Watts (2009) zaman içinde gelişen etkileşimlerin, niteliklerin ve ilişkilerin kaydedildiği ağ verilerini kullanarak, büyük bir üniversite topluluğunda homofili durumunu analiz etmişler ve oldukça benzer çiftlerin yeni bağlar oluşturmak için ortalama eğilimden daha yüksek eğilim gösterdiğini keşfetmişlerdir. Thelwall (2010) ise MySpace sosyal ağındaki homofili eğilimini analiz etmişlerdir ve benzer

toplumsal duygu ifadesi seviyelerine sahip olanların arkadaş olma eğiliminde olduğunu gözlemlemişlerdir.

Bireyler ve arkadaşlarının sosyal davranışlarını temel alan bazı teorik ve ampirik çalışmalar da yapılmıştır. Backstrom vd. (2006) bir çevrimiçi topluluktaki üyelik problemini incelemiş ve bir çevrimiçi topluluğa katılan bir kullanıcının hareketi ile o topluluğa üye olan arkadaş sayısı arasında bir ilişki olduğunu gözlemlemişlerdir. Marlow vd. (2006) Flickr sosyal ağındaki etiket kullanım problemini göz önüne almış ve bir kullanıcı tarafından yerleştirilen bir dizi etiketi ve sonrasında kullanıcının arkadaşları tarafından yerleştirilen etiketler arasında bir benzerlik olduğunu göstermişlerdir. Birbirleri ile sosyal bağlantıları olan bireylerin eylemlerinde benzerlik gösterme eğilimi olduğu gözlemlenmiş ve bu kavram *sosyal etki* (social influence) olarak tanımlanmıştır. Şekil 3.3’de bir ağ yapısındaki nesnelere ait öznitelik uzayının zaman içerisinde sosyal etki eğilimi ile nasıl evrildiği örneklendirilmiştir. Verilen başlangıçtaki ağ yapısında, kırmızı renk ile gösterilmiş düğümlere ait öznitelik kümesi verilmiştir. Bu düğümlerin öznitelik kümeleri bağlantılı olduğu düğüm kümelerinin öznitelik kümelerine göre zaman içerisinde değişim gösterdiği ve aralarındaki benzerliğin arttığı gösterilmektedir.



Şekil 3.3: Zamana göre ağ yapısında gözlenen sosyal etki eğilimi.

Homofili ve sosyal etki kavramları bireylerin ayrışması, bireylerin topluluk gelişimi, sosyal hareketliliği vb. bir çok çeşitli sosyo-psikolojik durumları açıklamak için

kullanılmıştır. Son yıllarda ise bu eğilimler, ağları, büyük toplulukları anlamak ve özellikle ağ bağlantılı veriler üzerine yaklaşım geliştirmek için sıklıkla kullanılmaktadır. Çünkü, analiz edilen ağ yapısı ve bu ağ yapısı içerisindeki üyelerin davranışları, özellikleri ve ilişkileri bir dinamik süreç içerisinde evrilir. Bu dinamik süreçteki değişim, büyüme veya azalma homofili ve sosyal etki eğilimlerinden etkilenmektedir. Bu nedenle, veriler arasındaki korrelasyonu ortaya çıkarmak için ağ bağlantılı verinin homofili ve sosyal etki eğilimleri araştırılmakta ve bu eğilimlerin oluşmasına etki eden benzerlikler analiz edilmektedir (Liao vd., 2017; Huang vd., 2017a).

3.4. Ağ Bağlantılı Veri için Öznitelik Seçim Yöntemleri

Geleneksel öznitelik seçimi yöntemleri (Mitra vd., 2002; Guyon ve Elisseeff, 2003; He vd., 2006; Zhao vd., 2009; Nie vd., 2010; Yang vd., 2011; He vd., 2012; Yang vd., 2012; Bennasar vd., 2015; Bharti ve Singh, 2015; Abualigah vd., 2017; Hu vd., 2018) veri nesnelerinin bağımsız ve aynı şekilde dağıtıldığını varsaymakta ve veri nesneleri arasındaki ilişkileri belirten bağlantı yapısını gözardı etmektedir. Bu nedenle, bazı yeni çalışmalar, ağ verisine ait öznitelikler arasındaki korrelasyonu ağ verisine ait bağlantı yapısından faydalanarak bulmayı hedeflemektedir (Zhu vd., 2007; Qi vd., 2012; Tang ve Liu, 2012; Chang vd., 2015; Hoseini ve Mansoori, 2016).

Ağ bağlantılı veri için ilk önerilen yaklaşımlar daha çok gözetimli öznitelik seçimi yaklaşımlarıdır. Öncü çalışmalardan ilki (Gu ve Han, 2011), bağlantı bilgisini kullanmak için doğrusal bir regresyon modele graf düzenleme (graph regularization) aşaması eklemiş ve graf üzerindeki düğümler arasındaki benzerlikten yararlanarak öznitelik seçimi yapmaya çalışmışlardır. FsNet (Feature Selection in Network) olarak adlandırılan çalışma, içerik bilgisini öğrenmek için doğrusal regresyon (linear regression) ve bağlantı bilgisini birlikte kullanmışlar ve bağlantı bilgisinden faydalanmak için bir önceki yöntemde olduğu gibi graf düzenleme yöntemini kullanmaktadır. Ardından, öznitelik seçim aşaması için Laplas Düzenleştirilmiş En Küçük Kareler (Laplacian Regularized Least Squares) yöntemini problem için özelleştirerek, eğitim hatasını en aza indirecek şekilde bir öznitelik alt kümesi seçimi yapılmaktadır. Ortaya çıkan optimizasyon problemini çözmek için Proksimal Gradyan İniş (Proxi-

mal Gradient Descent) algoritması çalıştırılmıştır. Diğer bir öncü yöntem ise (Tang ve Liu, 2012) yarı gözetimli çalışan LinkedFS (Linked Feature Selection) yöntemidir. LinkedFS yöntemi, arkadaşlık, takip, takipçi gibi çeşitli sosyal ilişkileri geleneksel öznitelik seçimine entegre etmekte ve ortaya çıkan bütünleşik formülasyonları var olan bir öznitelik seçim algoritmasına ($l_{2,1}$ -norm düzenleme tabanlı) (Liu vd., 2009) dahil etmektedir.

LUFS (Linked Unsupervised Feature Selection) yöntemi (Tang ve Liu, 2014) gözetimsiz öznitelik seçim yöntemlerinin ilk bilinen örneklerindendir. LUFS, bağlantılı veri üzerinde öznitelik seçimi yapmak için sözde etiket temelli bir öznitelik seçimi yöntemi olan UDFS yöntemine bir sosyal boyut yaklaşımını dahil etmiştir. UDFS yöntemini temel aldığı için sözde etiket tabanlı bir yaklaşıma sahiptir. UDFS'den farklı olarak, hem bağlantı hem de öznitelik değeri verilerinden kısıtlamalar çıkararak sözde sınıf etiket bilgisi arar. Öznitelik seçimi sürecinde, bağlantı yapısından ve öznitelik uzayından gelen bazı kısıtlamalarla tutarlı olan öznitelik alt grubu seçilir. Bağlantı yapısından ve öznitelik uzayından gelen kısıtlamalar sırasıyla sosyal boyut düzenlemesi ve spektral analiz yoluyla elde edilir. LUFS, nesneler arasındaki bağlantı bilgisinden yararlanmak için sosyal boyut kavramını kullanır. Sosyal boyuttan yararlanarak bağlantılı veriler arasında ilişkileri gruplar halinde ortaya koymayı amaçlamaktadır. Sosyal boyutlar, veri örneklerini gruplayan kümeleme ve topluluk bulma yöntemleri ve gruplar arasındaki mesafeyi maksimize etmeye çalışan Doğrusal Farklılık Analizi (Linear Discriminant Analysis) yaklaşımı ile tespit edilmektedir. İlk olarak sosyal boyut gösterge matrisi H 'ı elde etmek için bir çizge parçalama algoritması olan bir modülerlik maksimize etme yöntemi (Newman, 2004) kullanır ve ağ verisinin bağlantı yapısını temsil eden çizgeye ait gruplar elde edilir. Sonrasında, D öznitelik sayısını, s sözde sınıf sayısını temsil etmek üzere bir haritalama fonksiyonu (bir doğrusal sınıflandırıcı) $W \in R_{D \times s}$ 'yi ve n nesne sayısını temsil etmek üzere sözde sınıf etiket gösterge matrisi $Y = W^T X \in R^{s \times n}$ 'ni oluşturur. Bu temsilleri sosyal boyut kavramı ile bütünleştirerek, aşağıdaki denklemlerde verildiği şekilde S_w , S_b ve S_t olmak üzere sosyal boyut dağılım matrislerini elde eder.

$$S_w = YY^T - YH(H^T H)^{-1/2}(H(H^T H)^{-1/2})^T Y^T \quad (3.1)$$

$$S_b = YH(H^T H)^{-1/2}(H(H^T H)^{-1/2})^T Y^T \quad (3.2)$$

$$S_t = YY^T \quad (3.3)$$

Sosyal boyut düzenlenmesinden gelen bağlantı yapısına dair kısıtlama ile spektral analiz yoluyla öznitelik yapısına ait kısıtlama sırasıyla Denklem 3.4 ve Denklem 3.5’de gösterildiği gibi elde edilir.

$$\max_W Tr((S_t)^{-1} S_b) \quad (3.4)$$

$$\min_W Tr(YY^T) \quad (3.5)$$

Hem bağlantı bilgilerinden hem de öznitelik-değer verisinden gelen kısıtlamaları göz önünde bulundurarak, LUFs yöntemi aşağıdaki optimizasyon problemini çözerek öznitelik seçimini gerçekleştirmektedir.

$$\min_{W,s} Tr(YY^T) - \alpha Tr((S_t)^{-1} S_b) \quad (3.6)$$

Bir diğer ağ bağlantılı veri üzerinde gözetimsiz olarak öznitelik seçimi yapan yöntem POPFS (Partial Order Preserving) (Wei vd., 2015) olarak adlandırılan Kısmi Düzen Koruma yöntemidir. Yöntem, ağ nesneleri ve komşuları arasındaki benzerliği muhafaza ederek öznitelik seçimi yapmaktadır. Bu çalışmada, iki farklı yaklaşım ile öznitelik seçimi yapılmaktadır. Birinci yaklaşım da basitçe her bir öznitelik için, bu özniteliği paylaşan komşu çiftlerin sayısı temelinde puan hesabı yapılmakta ve sonrasında puanlara göre öznitelikler sıralanarak seçim yapılmaktadır. Bu yaklaşımda, öznitelikler ayrı ayrı değerlendirilmekte ve bu nedenle öznitelikler arasındaki korelasyon dikkate alınmamaktadır. Diğer yaklaşımda, yöntem ağ yapısı içerisinde aralarında bağlantı olmayan ama aynı grupta olma ihtimali yüksek olan benzer düğümleri tanımlayan bir anlamlı öznitelik kümesini arar. Bu amaçla, L_i düğüm i ’nin komşularını ve U_i düğüm i ile bağlantılı olmayan düğümleri temsil etmek üzere, POPFS her düğüm için Denklem 3.7’de gösterildiği gibi kısmi sıra üçlüsü (partial order triplets) ($j >_i k$) kümeleri tanımlar. Sonrasında, farklı öznitelik alt kümeleri arasından en ayırıcı olan

öznitelik kümesini bulmak için $\mathbf{w} = \{w_1, w_2, \dots, w_D\} \in \{0, 1\}^D$ öznitelik seçim matrisi olmak üzere Denklem 3.8’de verilen optimizasyon problemini çözer. Optimizasyon problemini çözmek için yöntem, Stokastik (Alt) Gradyan İniş (Stochastic Gradient Descent) algoritmasını kullanmaktadır.

$$\Omega = \{(i, j, k) | i \in V, j \in L_i, k \in U_i\} \quad (3.7)$$

$$\max_w L(>) = \sum_{(i,j,k) \in \Omega} l((j >_i k) | \mathbf{w}) \quad (3.8)$$

Bir diğer en yaygın öznitelik seçim yöntemi NetFS (Feature Selection for Networked Data) yöntemidir (Li vd., 2016). NetFS ağ örnekleri arasındaki korelasyonları ortaya çıkarmak için gizli temsilleri (latent representation) kullanmakta ve gizli temsilleri öğrenme sürecini öznitelik seçim sürecine katmaktadır. NetFS, bağlantı bilgisinden gizli temsilleri simetrik negatif olmayan matris ayrıştırma yöntemi (SymNMF) (He vd., 2011) ile elde etmektedir. SymNMF yöntemi ile ağ içerisindeki bağlantılı her örnek, gizli özniteliklerin bir karışımından oluşur. c gizli faktörlerin sayısı ve n örneklerin sayısı olmak üzere örneklere ait gizli temsiller $U \in R^{(n \times c)}$ Denklem 3.9’de verildiği gibi elde edilmektedir. Burada A ağın komşuluk matrisini ifade etmektedir.

$$\min_{U \geq 0} \|A - UU^T\|_F^2 \quad (3.9)$$

NetFS, gizli temsilleri öğrenme sürecini öznitelik bilgisi X ile bütünleştirerek öznitelik seçimi aşamasına eklemektedir. Bu nedenle, U matrisini, çok değişkenli bir doğrusal regresyon ile modellemek için Denklem 3.10’da gösterildiği gibi bir kısıtlama olarak kullanmaktadır. Burada, W bir dönüşüm parametre matrisi olmak üzere $W(i, :)$ değeri i . özneliliğin önemini gösterir.

$$\min_W \|XW - U\|_F^2 \quad (3.10)$$

$$\min_{U \geq 0, W} \|XW - U\|_F^2 + \|W\|_{2,1} + \|A - UU^T\|_F^2 \quad (3.11)$$

NetFS gizli temsiller üzerinden her bir özneliliğin ağırlığını hesaplayacak şekil-

de problemi Denklem 3.11’de verilen bir optimizasyon problemine dönüştürür ve optimizasyon probleminin çözümü ile öznitelik alt kümelerini seçer.

3.5. Ağ Bağlantılı Veri için Öznitelik Çıkarım Yöntemleri

Öznitelik çıkarımı üzerine yapılan son çalışmalar, genellikle bağlantı yapısı ve içerik bilgilerini birleşik bir alana yerleştirerek ortak bir temsil sağlamaya odaklanmıştır (Zhu vd., 2007; Qi vd., 2012). Bu konuda yapılan öncü çalışmalardan ilki (Zhu vd., 2007) web sayfalarını sınıflandırma başarısını artırmak için web sayfalarına ait içerik bilgisi ile web sayfaları arasındaki köprüler sayesinde oluşan ilişkileri birlikte kullanmıştır. Web sayfalarını bir çizge olarak temsil etmişler ve bağlantı bilgisini (köprüleri) tutan komşuluk matrisi ile web sayfalarının içerisindeki terimlerin frekanslarını tutan belge-terim matrisi üzerinde ortak bir faktörizasyon gerçekleştirmişlerdir. Bu sayede, hem içerik hem de bağlantı bilgisini kullanan bir yöntem geliştirmeyi amaçlamışlardır. Bu iki farklı matris temel alınarak, web sayfaları için düşük boyutlu bir uzayda yeni bir temsil üretmek için, Ortak Bağlantı-İçerik Matris Ayrıştırma (Joint Link-Content Matrix Factorization) olarak adlandırılan bir yöntem geliştirmişler. Yöntem, çizgeyi temsil eden komşuluk ve içeriği temsil eden belge-terim matrisi için ayrı ayrı olarak matris ayrıştırma yaklaşımını uygulamaktadır. Matematiksel olarak ifade edilir ise, öznitelik F ve komşuluk matrisi A verildiğinde, terim ve bağlantıların gizli temsillerine ait $S \times S$ boyutsallığında bir U matrisi, $N \times S$ boyutsallığında V matrisi ve $N \times S$ boyutsallığında yeni bir öznitelik matrisi X elde edilmektedir. Önerilen yönteme ait amaç fonksiyonu aşağıdaki denklemde verilmektedir. Yöntem, α , γ ve β parametreleri ile bağlantı ve içerik dengesine ait kararlılığı düzenlemektedir. Yöntemde, Denklem 3.12’de verilen minimizasyon problemini çözümlmek için Eşlenik Gradyan Yöntemi (Conjugate Gradient Method) ve Yarı-Newton Yöntemi (Quasi-Newton) olmak üzere bir kaç farklı gradyan yöntemi kullanılmıştır.

$$\min_{U,V,X} J(U, V, X) = \|A - XU X^T\|^2 + \alpha \|F - XV^T\|^2 + \gamma \|U\|^2 + \beta \|V\|^2 \quad (3.12)$$

Hem bağlantı hem de içerik bilgisini kullanarak gizli temsilleri öğrenmeyi temel alan diğer bir yaklaşım (Qi vd., 2012), çoklu ortam ağları için LSI (Latent Semantic

Indexing) yaklaşımını kullanmış ve çoklu ortam nesneleri için oluşturduğu gizli temsiller üzerinden yeni öznitelik vektörleri elde etmiştir. Diğer bir yöntem ise PLANE (Probabilistic Latent Document Network Embedding) (Le ve Lauw, 2014) yöntemidir. PLANE yöntemi, ağ verilerine ait birleşik bir düşük boyutlu temsil elde etmek için bir belgenin çeşitli gösterimlerini (kelimeler, bağlantılar, konular ve koordinatlar) bir araya getiren olasılıksal bir ilişkiyel konu modeli kullanmaktadır. Yöntem, konu modelleri ile her belgenin bir konu dağılımı ile ilişkili olduğu ve her konunun bir kelime dağılımı ile ilişkili olduğu bir yapı oluşturarak ağ verilerini temsil etmektedir.

Son zamanlarda, ağ bağlantılı veriye ait bağlantı ve içerik bilgisini düşük boyutsalıkta bir bütünleşik uzaya transfer eden gömme tabanlı yaklaşımlar (embedding methods) önerilmektedir. Huang vd. (2017b,a) tarafından LANE (Label Informed Attributed Network Embedding) ve AANE (Accelerated Attributed Network Embedding) olarak adlandırılan iki öznitelikli ağ gömme yöntemi önerilmiştir. LANE yöntemi, nesneler arasındaki yapısal yakınlığı (graf temelindeki) ve öznitelik vektörleri arasındaki benzerlikleri etiket bilgisi ile düzgün bir şekilde modellemekte ve nesneler için düşük boyutsalıkta yeni bir temsil elde etmektedir. Yöntem, öncelikle, sırasıyla U^G ve U^F olan gizli yapısal bilgi ve gizli öznitelik bilgisi matrislerini düğümler arasındaki benzerliğe dayalı olarak ortaya çıkarır. Sonrasında, bu matrisler arasındaki korelasyonu temel olarak gizli öznitelik bilgisini gizli yapısal bilgi ile ilişkilendirir. Eğer nesneler için etiket bilgisi mevcut ise yöntemin gözetimli çalışması için, nesneler arasındaki etiket bilgisine dayanan yakınlığı başka bir gizli temsil U^Y matrisine yansıtır. Yöntem, son olarak öğrenilen tüm gizli temsilleri birleşik bir gömme temsili matrisi H oluşturur. LANE, U^G ve U^F 'yi elde etmek için normalize edilmiş graf Laplas yöntemine (Chung ve Graham, 1997) dayanarak, öznitelik çıkarım problemini bir maksimizasyon problemi dönüştürmekte ve Denklem 3.13 ve 3.15 verilen amaç fonksiyonları ile nesneler arasındaki yakınlıkları modellemektedir.

$$\max_{U^G} J_G = Tr(U^{G^T} L^G U^G) \quad (3.13)$$

$$\max_{U^F} J_G = Tr(U^{F^T} L^F U^F) \quad (3.14)$$

Verilen maksimizasyon problemi, özdeğer optimizasyonu ile çözülmektedir. S ,

Cosinüs benzerliği yöntemi elde edilmiş düğümler arasındaki benzerliği tutan bir yakınlık matrisidir. D , S 'ye ait köşegen matrisi olmak üzere L Laplas matrisi, $D^{-1/2}SD^{-1/2}$ işlemi ile elde edilir. U^G ve U^F matrisleri Denklem 3.15'de verildiği gibi ilişkilendirildikten sonra, Denklem 3.16'de verilen optimizasyon problemi çözülerek hem bağlantı hem içerik bilgisinin yansıtıldığı ağ bağlantılı veri için yeni bir birleşik gömme temsili H elde edilmektedir.

$$\rho_1 = \text{Tr}(U^{F^T} U^G U^{G^T} U^F) \quad (3.15)$$

$$\max_{U(\cdot), H} J = (J_G + J_A + \rho_1) \quad (3.16)$$

Benzer bir yaklaşıma sahip olan AANE yöntemi ise aynı şekilde öncelikle bağlantı yapısı ve içerik bilgisine dair farklı gömme temsilleri elde etmekte ve sonra bu temsillerden ortak bir temsil öğrenmektedir. AANE yöntemi LANE yönteminden farklı olarak, amaç fonksiyonu öznitelik dağılımı benzer ve birbirine bağlı olan düğümler arasındaki yakınlığı artıracak şekilde, diğer bir deyişle gömme temsillerindeki uzaklığı minimize edecek şekilde bir hata fonksiyonunu minimize etmektedir. S düğümler arasındaki öznitelik yakınlığını tutan yakınlık matrisi ve h_i i. düğüme ait H vektör temsili olmak üzere, ağ bağlantılı veriye ait gömme temsili H aşağıdaki denklemde verildiği gibi bulunmaktadır.

$$\min_H J = \|S - HH^T\| + \sum_{(i,j) \in E} e_{ij} \|h_i - h_j\|_2. \quad (3.17)$$

Ağ bağlantılı verilere ait düşük boyutlu temsiller elde etmek için önerilen bir diğer gömme yöntemi ise Gat2Vec (Sheikh vd., 2019) yöntemidir. Gat2Vec yöntemi, ağ bağlantılı veriye ait içerik ve bağlantı bilgisini bir ikili graf üzerinde bütünleştirerek tek bir temsil elde etmektedir. Bu graf temsili üzerinde rastgele yürüyüş algoritması çalıştırılarak her bir düğüm için belirlenen uzunluklarda yollar elde edilmektedir. Bu yollar, bir dizi metin olarak değerlendirilerek literatürde kullanılan Word2Vec algoritması (Mikolov vd., 2013) çalıştırılmaktadır. Yöntem, sadece ağ bağlantısı üzerinde çalışan DeepWalk (Perozzi vd., 2014) algoritmasının ağ bağlantılı veriler için geliştiril-

miş bir halidir.

Bağlantı ve içerik bilgisini birlikte kullanmayı temel alarak kümeleme performansını artırmayı hedefleyen bir diğer yaklaşım Çok-Görüşlü Öğrenme (Multi-View Learning) yaklaşımıdır. Çok görüşlü öğrenme yaklaşımları (De Sa, 2005; Kumar ve Daumé, 2011) her nesne için farklı kaynaklardan gelen her bir görünümün modellenmesi için ayrı bir fonksiyon öğrenmeyi ve daha iyi performans elde etmek için bu fonksiyonları birlikte optimize etmeyi amaçlamaktadırlar. De Sa (2005) tarafından spektral kümeleme algoritması için veriye ait iki farklı uzayı iki bağımsız görüş olarak değerlendirip bütünleştiren bir yöntem önerilmiştir. Yöntem, her iki görüşü bir iki-taraflı çizgeye (bipartite graph) yansıtır. Bu çizgenin kenarları, düğümleri bir görünümünden diğer görünüme bağlar. Daha sonra, bu iki taraflı çizge üzerinde standart spektral kümeleme problemi çözülmektedir. Diğer bir çok-görüşlü spektral kümeleme yöntemi ise Kumar ve Daumé (2011) tarafından önerilmiştir. Önerilen yöntem, farklı görüşlerden öğrenilen özvektörlerden hesaplanan ikili benzerliklerin yakın olması için bir ortak düzenleme (co-regularization) yaklaşımı kullanmaktadır. Bu amaçla, iki görüşe sahip veri için elde edilecek yeni temsile ait örneklerin ikili görünümdeki benzerliklerinin tüm görüşler arasında benzer olmasını sağlar. Bunun için, iki görüş arasındaki uzaklığı ölçen bir maliyet fonksiyonu kullanmaktadır. v ve w verinin iki ayrı görüşünü temsil etmek üzere, Denklem 3.18'de verilen maliyet fonksiyonu hesaplanmaktadır. $U^{(v)}$ ve $U^{(w)}$ iki görüşe ait özdeğer vektör matrisidir.

$$D(U^{(v)}, U^{(w)}) = -Tr(U^{(v)}U^{(v)T}U^{(w)}U^{(w)T}) \quad (3.18)$$

Yöntem, v ve w görüşlerinin kümelenmeleri arasındaki yukarıdaki maliyeti en aza indirmeyi amaçlamaktadır. Bu nedenle, iki yapı için aşağıdaki ortak maksimizasyon problemini çözmeyi hedefler.

$$\max_{U^{(v)}, U^{(w)}} = Tr(U^{(v)T}L^{(v)}U^{(v)}) + Tr(U^{(w)T}L^{(w)}U^{(w)}) + Tr(U^{(v)}U^{(v)T}U^{(w)}U^{(w)T}) \quad (3.19)$$

4. AĞ BAĞLANTILI VERİ İÇİN GÖZETİMSİZ ÖZNİTELİK OLUŞTURMA

Bu tez çalışması kapsamında, ağ bağlantılı veriler üzerinde gerçekleştirilen kümeleme görevlerinin başarısını artırmak için bir öznitelik oluşturma yöntemi geliştirilmiştir. Geliştirilen yöntem gözetimsiz ikili öznitelik oluşturma yaklaşımına sahiptir ve NetBFC (Binary Feature Construction Method for Networked Data) olarak adlandırılmıştır.

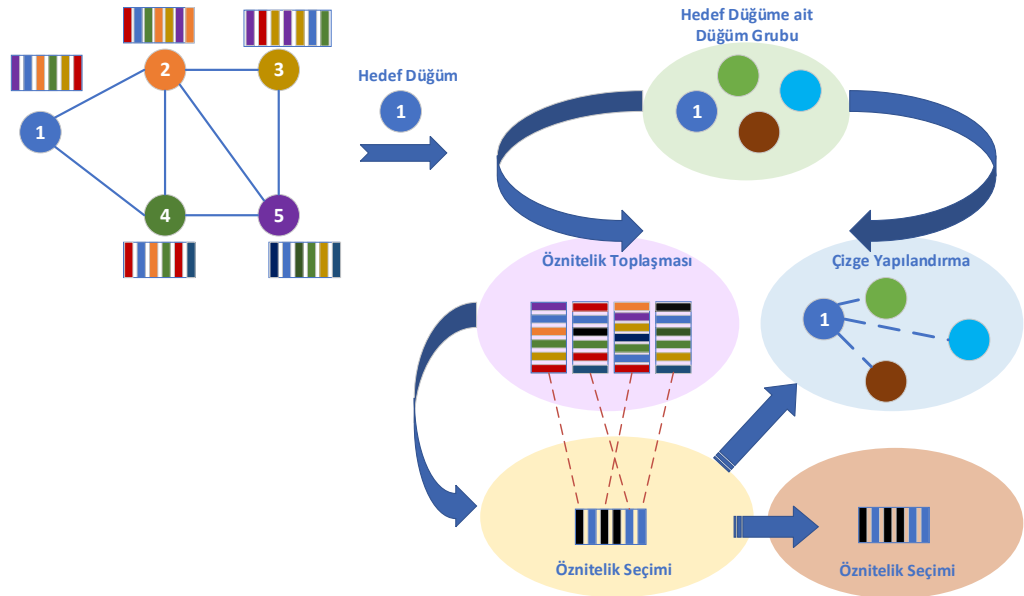
Aşağıdaki bölümlerde, öncelikle NetBFC yöntemi için kısa bir giriş ve sonrasında öznitelik oluşturma sürecine dair tanımlamalara ve yöntemin algoritmasına yer verilmektedir.

4.1. Yöntem: NetBFC

NetBFC yöntemi, bir kümeleme görevinin başarısını artırmak için aynı kümede olma ihtimali yüksek benzer nesnelerin öznitelik alanları arasındaki yakınlığı artıracak şekilde bir öznitelik oluşturma süreci gerçekleştirmektedir. Öznitelik oluşturma, öznitelik üretme ve öznitelik alt kümesi seçme süreçlerinden oluşmaktadır. Yöntem, her nesnenin öznitelik alanını yeniden yapılandırmak için nesneyi aynı kümede olma ihtimali yüksek bir grup nesne ve bu nesnelere ait öznitelik uzayı ile ilişkilendirmektedir. Diğer bir deyişle, bir nesnenin yeni öznitelik temsilini oluşturmak için yerel bir arama süreci gerçekleştirilmektedir. Bir nesne için aynı kümede olma ihtimali yüksek bir nesne grubu belirlenir ve bu grubun oluşturduğu öznitelik alanı, bu nesneye ait dolaylı öznitelikleri temsil eder. Bu dolaylı özniteliklerin nesne ile ilişkilendirilme süreci öznitelik üretme süreci olarak adlandırılmaktadır. Sonrasında, üretilen öznitelik uzayından bir öznitelik alt kümesi seçilir ve bu öznitelikler ağ nesnesinin yeni temsilini oluşturur. Şekil 4.1’de NetBFC yönteminin çalışma süreci örneklendirilmiştir. Beş ayrı ağ nesnesine ait öznitelik vektörleri ve nesneler arasındaki bağlantıları gösteren ağ yapısı verilmektedir. Yöntem, hedef bir düğüm seçerek, bu düğümü bir düğüm grubu ile ilişkilendirmektedir. Sonrasında, bu düğüm grubundaki nesnelere ait öznitelik vektörleri kullanılarak hedef düğüm için bir lokal öznitelik seçimi yapılmaktadır. Aynı anda,

düğümüne ait bağlantı yapısı ilişkilendirildiği düğüm grubu temel alınarak yapılandırılmaktadır.

Öznitelik oluşturma sürecinde, her nesne için yerel bir öznitelik üretme ve seçme süreci gerçekleştirilmektedir. Bu nedenle, öncelikle her bir nesne için aynı kümede olma ihtimali yüksek bir nesne grubu (düğüm grubu) belirlenmektedir. Nesne grubu, ağ bağlantılı veriye ait hem öznitelik alanı hem bağlantı yapısı temel alınarak belirlenmektedir. Fa- kat, aynı kümede olması beklenen nesnelerin arasında yeterli derece de bir öznitelik yakınlığı gözlemlenmeyebilir. Bu durum, ağ verisinin çoğunlukla yüksek boyutsallıkla karakterize edilmiş ve yüksek derece de seyrekliğe sahip olmasından kaynaklanmaktadır. Bazı nesneler, benzerlik hesabını zorlaştıracak şekilde çok az öznitelik ile tanım- lanmışken, bazı nesneler gereksiz ve gürültülü öznitelikler ile birlikte çok fazla sayıda özniteliğe sahip olmaktadır. Bu durum, K-means vb. algoritmaların kümeleme performansını düşürmektedir. Benzer şekilde, aynı kümede olması beklenen iki nesnenin arasında bir bağlantı, karşılık düştükleri çizgede bir kenar olmayabilir. Bu durum nedeni ile, iki nesnenin öznitelikleri arasındaki korrelasyon gözlemlenmeyebilir. Bu iki durum ile başa çıkabilmek için NetBFC yöntemi, nesneler arasındaki dolaylı olarak gözlemlenen öznitelik ve bağlantı birlikteliklerini ortaya çıkarmayı hedeflemektedir. Bu dolaylı birliktelikler, yöntemin öznitelik oluşturma sürecinin temelini oluşturmaktadır.



Şekil 4.1: NetBFC yönteminin çalışma sürecine ait gösterim.

Tez kapsamında, NetBFC ile, bir öznitelik matrisi F ve bir bağlantı matrisi L' yi kullanarak, ağ bağlantılı verinin nesnelere ait yeniden oluşturulmuş bir S boyut-sallığında $X = \{x_1, x_2, \dots, x_D\} \in [\{0, 1\}]^{N \times S}$ öznitelik matrisi elde edilmesi amaçlanmaktadır. Sonuç olarak, orijinal öznitelik matrisi F ile karşılaştırıldığında, oluşturulan yeni öznitelik matrisi X ile daha verimli bir öğrenimin elde edilmesi beklenmektedir.

4.1.1. Tanımlamalar

Ağa bağlı verilerde, düğümler ve düğüm öznitelikleri birbirleriyle ilişkili olma eğilimindedir (Hand, 2010). Bu korelasyonu açıklamak için araştırmacılar *homofili* (homophily) ve *sosyal etki* (social influence) olmak üzere iki ana eğilimi ortaya çıkarmışlardır. Homofili, benzer özelliklere sahip düğümlerin birbirleriyle bağlantı kurma eğilimi olarak (McPherson vd., 2001), sosyal etki ise düğümlerin aynı gruptaki düğümlerle benzer özellikler gösterme eğilimi olarak tanımlanmaktadır (Friedkin, 2006). Bu nedenle, ağ verilerinde bu iki etkiyi gözlemlemek için literatürde (Huang vd., 2017a; Liao vd., 2017) iki yakınlık tanımı verilmektedir. Bunlar *bağlantı yakınlığı* ve *öznitelik yakınlığı* olarak adlandırılmıştır. Adından da anlaşılacağı gibi, bağlantı yakınlığı düğümlere ait bağlantıların benzerliği tarafından belirlenen yakınlıktır. Öznitelik yakınlığı ise özniteliklerin benzerliği tarafından belirlenen yakınlıktır. Bu yakınlıklara ait tanımlamalar aşağıda verilmektedir.

- **Öznitelik Yakınlığı:** u ve v olmak üzere iki düğüme ait f_u ve f_v öznitelik kümelerinin kesişimi $(f_u \cap f_v) \neq \emptyset$ sıfırdan farklı ise u ve v düğümleri arasındaki öznitelik yakınlığı $\rho_A > 0$ ' dır.
- **Bağlantı Yakınlığı:** u ve v düğümleri arasında bir $e_{u,v}$ bağlantısı var ise u ve v düğümleri arasında bir bağlantı yakınlığı $\rho_S > 0$ vardır.

İki düğüm arasındaki bağlantı yakınlığı doğrudan veya dolaylı olarak gözlemlenebilir. Doğrudan yakınlık iki bağlı düğüm arasında bulunur ve birinci dereceden

yakınlığı gösterir. Dolaylı yakınlık, düğümler arasındaki komşuluk benzerliğini gösteren ikinci derece yakınlığa karşılık gelir.

- Dolaylı Bağlantı Yakınlığı: u ve v düğümlerine ait komşu düğümler kümesi d_u ve d_v olmak üzere, bu kümelerin kesişimi $(d_u \cap d_v) \neq \emptyset$ sıfırdan farklı ise u ve v düğümleri arasında dolaylı bir yakınlık vardır.

Bu tez çalışmasında, düğümün doğrudan ve dolaylı ilişkilerini kullanarak her bir düğüm için öznitelikleri ve çizge de karşılık düştüğü düğümün bağlantılarını yeniden düzenlemeyi amaçlıyoruz. Yeniden düzenleme sürecinde, aynı kümede olma olasılığı yüksek olan düğümler arasındaki benzerliği artıracak şekilde düğümlere ait öznitelikler yeniden oluşturulmaktadır. Bu nedenle, öncelikle aynı kümede olma olasılığı yüksek olan düğümleri belirlememiz gerekmektedir. Aynı kümede olma olasılığı yüksek olan düğümler, benzer düğümler olarak adlandırılmaktadır. Bu düğümler, öznitelik ve/veya bağlantı yakınlığı gösteren düğümlerdir. Benzerlik gösteren düğümleri tespit etmek için düğümler arasındaki öznitelik ve bağlantı yakınlıklarını temel almaktayız.

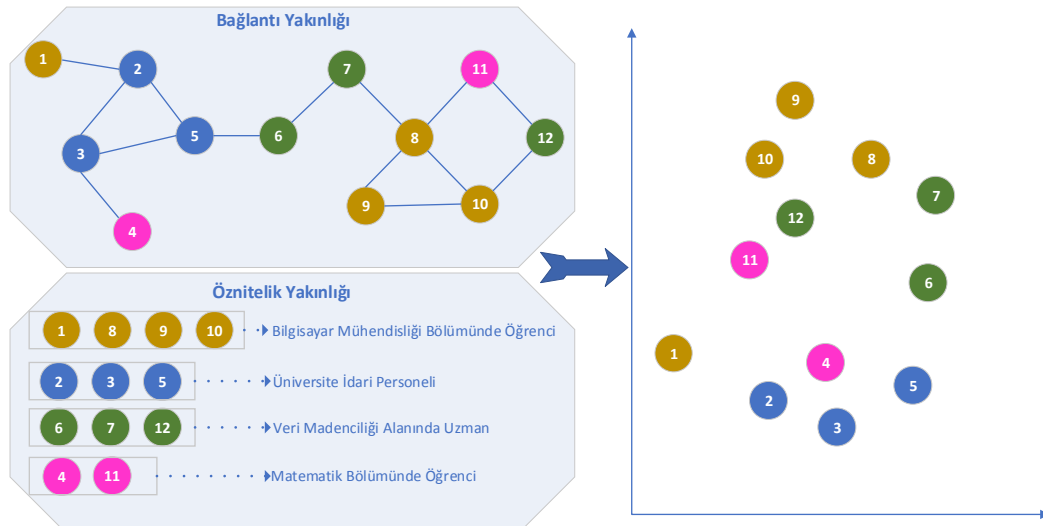
- Benzer Düğümler: Eğer u ve v düğümlerine ait öznitelik kümelerinin kesişimi $(f_u \cap f_v) \neq \emptyset$ ve/veya bu düğümlere ait komşu düğümler kümesinin kesişimi $(d_u \cap d_v) \neq \emptyset$ sıfırdan farklı ise bu düğümler benzer düğümlerdir.

İki benzer düğüm arasındaki benzerliğin kuvvetini öznitelik yakınlığına ve bağlantı yakınlığına dayalı olarak ikinci dereceden bir yakınlık değeri olarak tanımlamaktayız. Önerilen yöntemde, benzer düğümleri belirlemek için kullanılan düğümler arasındaki yakınlık aşağıda tanımlanmaktadır.

- Öznitelik-Bağlantı Yakınlığı: Öznitelik bağlantı yakınlığı ρ_{AS} öznitelik ρ_A ve bağlantı yakınlığı ρ_S 'nin birlikte oluşturduğu ikinci dereceden bir yakınlıktır. u ve v düğümleri arasındaki öznitelik bağlantı yakınlığı $\rho_{AS}(u, v) = \rho_A(u, v) * \rho_S(u, v)$ eşitliği ile temsil edilir.

Şekil 4.2'de ağ bağlantılı bir veri için bağlantı-öznitelik yakınlığına ait örnek bir

gösterim sunulmuştur. Öznitelik kümeleri benzer olan nesneler aynı renk ile örneklendirilmiştir. Bu nesnelere ait bağlantılar ağ yapısı ile gösterilmektedir. Şekilde, öznitelik-bağlantı yakınlığına göre iki boyutlu düzlemde nesnelerin arasındaki yakınlık veya uzaklık örneklendirilmiştir. Örneğin; benzer öznitelik dağılımına sahip olmasına rağmen sarı renk ile renklendirilmiş düğüm 1, sarı renkli diğer düğüm kümesinden daha uzakta konumlanmıştır. Bunun nedeni, öznitelik benzerliği gösterdiği sarı renkli diğer düğümler ile bağlantı yakınlığı göstermemesindendir.



Şekil 4.2: Öznitelik ve bağlantı yakınlığı örneği.

Sosyal etki ve homofili etkisiyle gelişen bir ağ göz önüne alındığında, aynı kümede olması muhtemel iki benzer düğümün arasında daha yüksek derecede ikinci dereceden bir yakınlık gözlemlenmesi beklenmektedir. Öznitelik-bağlantı yakınlığı temel alınarak düğümler arasındaki içeriksel ve yapısal ilişkiler daha iyi yakalanabilir. Öznitelik-bağlantı yakınlığına göre bir u düğümüne doğrudan veya dolaylı olarak farklı değerlerde yakınlık gösteren bir çok düğüm söz konusu olacaktır. Bu çalışmada, öznitelik üretme sürecinde, bir düğüm ona doğrudan veya dolaylı olarak yakınlık gösteren bir grup düğüm ile ilişkilendirilmektedir. Bu nedenle, düğüm ona en fazla benzerlik gösteren bir düğüm kümesi ile kısıtlanarak, o düğüm için aynı kümede olma ihtimali en yüksek düğümlere odaklanılmaktadır. Bir düğümün öznitelik üretme sürecinde ilişkilendirildiği *üst-k-yakınlık komşuları* olarak adlandırılan düğüm kümesi aşağıda ifade edildiği gibi belirlenmektedir.

- Üst- k -yakınlık komşuları: u düğümünün üst- k -yakınlık komşuları, u düğümü ile en yüksek öznitelik-bağlantı yakınlığına sahip k adet düğümden oluşan bir düğüm kümesinin elemanıdır.

Öznitelik üretim sürecinde, bir düğüm belirlenen üst- k -yakınlık komşuları ve bu komşulara ait öznitelikler ile ilişkilendirilir. Ağda var olan her düğüm için düğümün kendi öznitelikleri ve komşularından gelen dolaylı öznitelikleri ile düğüm ilişkilendirilerek bir öznitelik üretimi süreci gerçekleştirilmektedir. Yeni öznitelikler, düğümün doğrudan ve dolaylı olarak ilişkilendirildiği özniteliklerin toplaşması (aggregation) ile üretilmektedir. Fakat, bu toplaşma sırasında düğümün öznitelik uzayı büyük miktarda genişleyebilir, gereksiz ve gürültülü öznitelikler ile ilişkilendirilebilir. Bu nedenle, düğümün ilişkilendirildiği öznitelik sayısını kısıtlamak ve düğümü, belirlenen üst- k -yakınlık komşuları ile daha uyumlu hale getirmek için üretilen özniteliklerin bir alt kümesi seçilmektedir. Bir düğüm için toplaşma sonucunda üretilen yeni öznitelikler içerisinde aşağıda tanımlanan z -ortak öznitelikler seçilir. Seçilen öznitelik alt kümesi (z -ortak öznitelikler) bu düğümün yeniden oluşturulan öznitelikleridir.

- z -ortak öznitelikler: u düğümünün z -ortak öznitelikleri, u düğümünün üst- k -yakınlık komşuları ile en fazla ortaklık gösterdiği z adet özniteliğin oluşturduğu öznitelik kümesinin elemanıdır.

4.1.2. Öznitelik Üretme

Yöntem, her bir düğümün öznitelik alanını yeniden yapılandırmak için bir öznitelik üretme süreci gerçekleştirmektedir. Bu süreç yerel bir öznitelik sürecidir ve her düğüm üst- k -yakınlık komşuları ile ilişkilendirir. Üst- k -yakınlık komşuları, düğümün doğrudan ve dolaylı birlikteliklerini temsil eder. Öznitelik-bağlantı yakınlığı, ağ verilerindeki her bir düğüme ait düğüm grubunu bulmak için kullanılır. Düğüm çiftleri arasındaki öznitelik-bağlantı yakınlık değerlerini hesaplamak için, ilk olarak, öznitelik yakınlık matrisi ρ_A ve bağlantı yakınlığı matrisi ρ_S , Cosine benzerlik ölçüsü kullanılarak hesaplanır. u ve v düğümleri arasındaki yakınlık değeri $\rho_{AS}(u, v)$ Denklem 4.3'

de olduğu gibi elde edilir.

$$\rho_A(u, v) = \begin{cases} \frac{f_u f_v}{\|f_u\| \|f_v\|}, & \text{if } \frac{f_u f_v}{\|f_u\| \|f_v\|} > 0 \\ \gamma, & \text{diğer durumda.} \end{cases} \quad (4.1)$$

$$\rho_S(v_i, v_j) = \begin{cases} \frac{l_u l_v}{\|l_u\| \|l_v\|}, & \text{if } \frac{l_u l_v}{\|l_u\| \|l_v\|} > 0 \\ \gamma, & \text{diğer durumda.} \end{cases} \quad (4.2)$$

$$\rho_{AS} = ((\rho_A)^\alpha \bullet (\rho_S)^\beta)_N \quad (4.3)$$

γ sıfırdan farklı bir sabittir. Bu değer, her düğüm için bağlantı ve öznitelik yakınlık değerlerini muhafaza etmek için sıfırdan büyük olmalıdır. Öznitelik-bağlantı yakınlık matrisi ρ_{AS} , ρ_A ve ρ_S matrislerinin Hadamard çarpımı sonucu ile elde edilen n boyutlu kare matristir. α ve β parametreleri sırasıyla öznitelik ve bağlantı yakınlığının katkı seviyesini kontrol etmek için kullanılır. Daha çok öznitelik alanını yansıtan bir temsil üretmek için, α parametresinin değeri β parametresinin değerinden yüksek olarak seçilir. Daha yapısal bir temsil üretmek için ise β değeri yüksek olmalıdır. Parametrelere ait farklı değerler farklı sonuçlar üretilmesini sağlar.

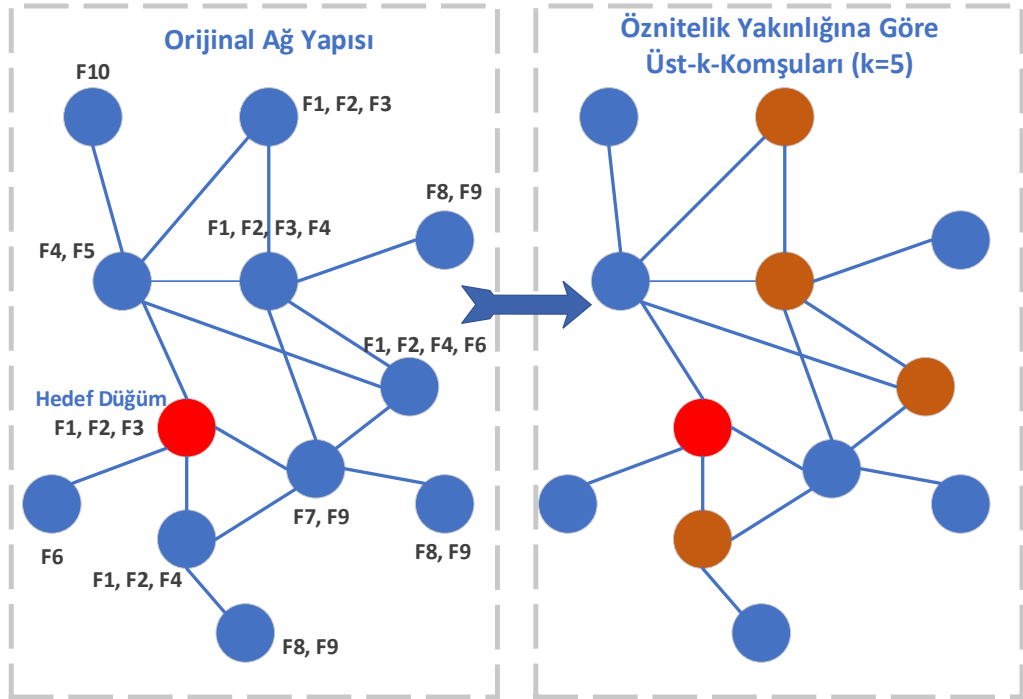
Öznitelik-bağlantı yakınlığı elde edildikten sonra, her düğüm, aynı kümede olma ihtimali yüksek üst- k -yakınlık komşuları ile ilişkilendirilir. Bu düğüm kümesi, düğüme ait doğrudan öznitelik birlikteliği ve doğrudan yada dolaylı bağlantı birlikteliklerini ifade eder. Matematiksel olarak ifade edilir ise, u düğümü için üst- k -yakınlık komşuları kp^i aşağıdaki Denklem 4.4’de verildiği şekilde belirlenir.

$$[kp_1^u, kp_2^u, \dots, kp_{k'}^u] = \max(\rho_{AS}(u, :) > \gamma^2, k'). \quad (4.4)$$

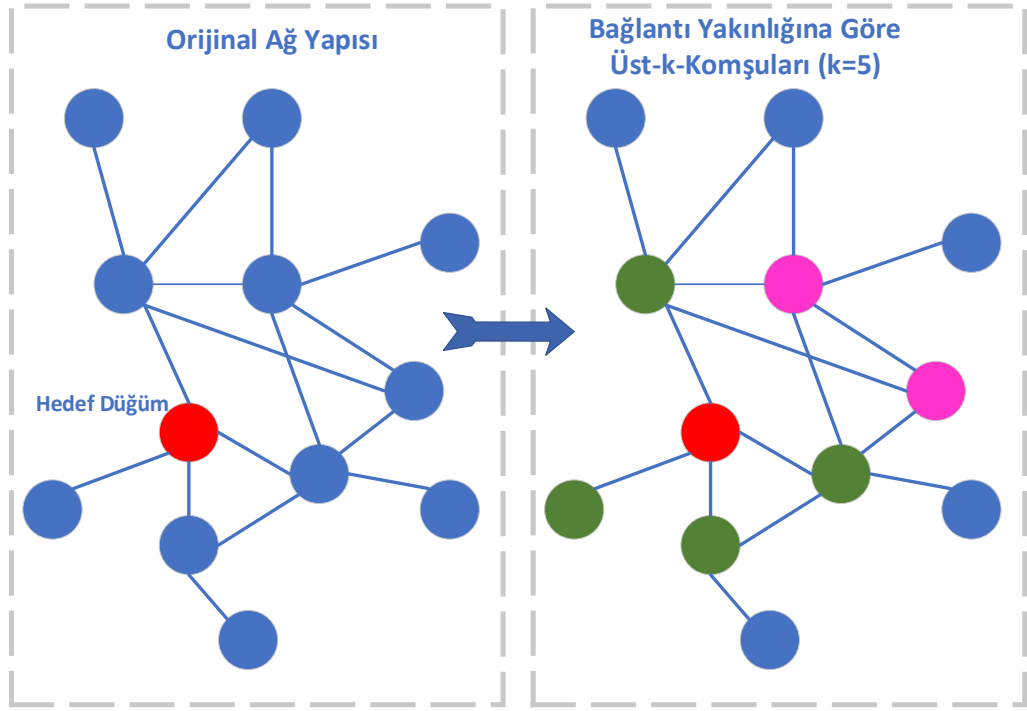
$\max(u, k)$ işlevi, u düğümünün üst- k -yakınlık komşularının (k adet düğüm) indislerini döndürür. Bazı durumlarda, bir düğümün γ^2 değerinden büyük benzerlik gösteren k adet komşusu olmayabilir. Bu nedenle, bu düğüm ancak $k' < k$ olacak şekilde k' kadar komşu ile ilişkilendirilir.

Şekil 4.3, 4.4 ve 4.5’de sırasıyla öznitelik yakınlığına, bağlantı yakınlığına ve

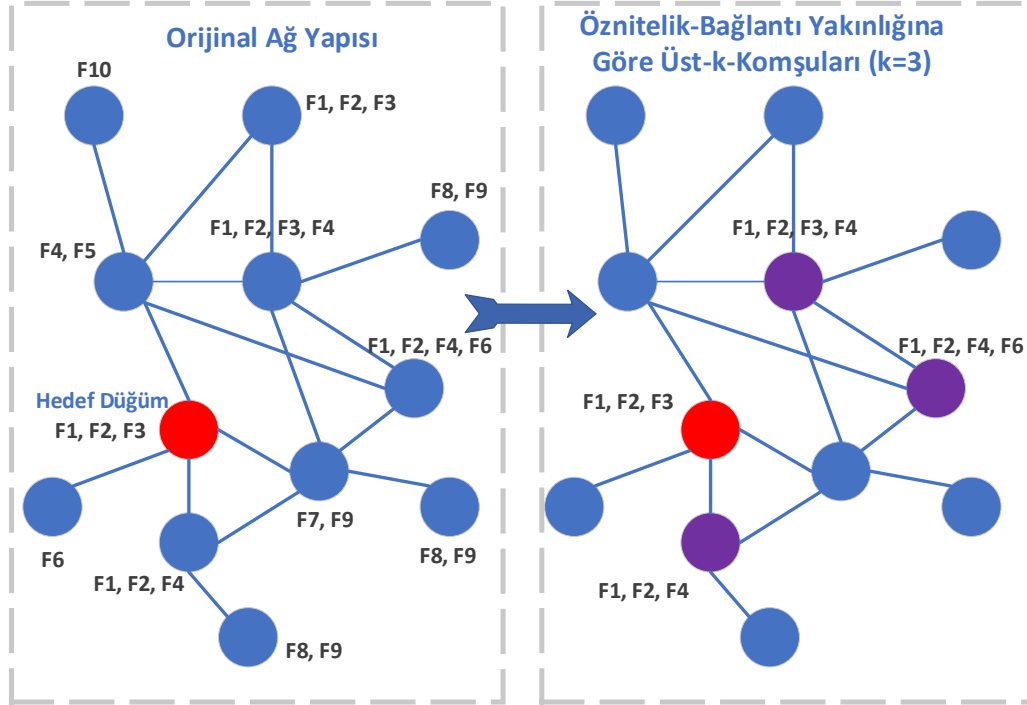
öznitelik-bağlantı yakınlığına göre hedef bir düğüm için üst- k -yakınlık komşularının nasıl belirlendiğini gösterilmektedir. Şekil 4.3’de orjinal ağ yapısı içerisinde öznitelik kümesi F_1, F_2, F_3 olan bir hedef düğüm seçildiği görülmektedir. Hedef düğüm için öznitelik kümesine en benzer öznitelik kümesine sahip 5 adet komşu seçilmesi beklenmektedir. Fakat, turuncu ile renklendirilmiş 4 adet üst- k -yakınlık komşusu hariç diğer düğümlerin öznitelik kümeleri ile hedef düğümün öznitelik kümesi hiçbir benzerlik göstermediği için (kesişimleri 0 olduğu için) sadece 4 adet komşu seçilebilmiştir. Bağlantı yakınlığına göre, aynı hedef düğüm için seçilen üst- k -yakınlık komşuları ise Şekil 4.4’de gösterilmektedir. Yeşil renk ile renklendirilmiş düğümler, hedef düğüm ile aralarında bağlantı olduğu için doğrudan bağlantılı düğümlerdir. Pembe renkli düğümler ise doğrudan bağlantılı olmayan ama ikinci dereceden bağlantı yakınlığı gösteren düğümlerdir. Önerilen yöntem NetBFC’ye göre ise üst- k -yakınlık komşularının belirlenmesi öznitelik-bağlantı yakınlığı temelinde gerçekleşmektedir. Şekil 4.5’de hem öznitelik hem bağlantı yakınlığı gösteren 3 adet düğümün hedef düğüm için üst- k -yakınlık komşusu olarak seçildiği gösterilmektedir.



Şekil 4.3: Öznitelik yakınlığına göre hedef düğüme ait üst- k -yakınlık komşularının belirlenmesi örneği.



Şekil 4.4: Bağlantı yakınlığına göre hedef düğüme ait üst-k-yakınlık komşularının belirlenmesi örneği.



Şekil 4.5: Öznitelik-bağlantı yakınlığına göre üst-k-yakınlık komşularının belirlenmesi örneği.

Öznitelik üretme süreci ise düğümün ve bu düğümün üst- k -yakınlık komşularının öznitelik vektörlerinin birleştirilmesi ile gerçekleştirilir. Bu aşama öznitelik toplama (feature aggregation) olarak adlandırılır. Bu birleştirme aşamasından önce, Denklem 4.5’de olduğu gibi düğümlerin öznitelik değerleri ikili olarak temsil edilir. Böylelikle, her bir üst- k -yakınlık komşusundan gelen özniteliklerin değerleri eşitlenmiş olur. Bu eşitlik ile, bu düğüm grubuna ait öznitelik uzayında, bir özniteliğin kaç adet düğümü temsil ettiği belirlenir. $f_u(i)$ düğüm u ’nun i . özniteliğinin değerini temsil etmek üzere Denklem 4.5’de gösterildiği şekilde hesaplanır.

$$f_u(i) = \begin{cases} 1, & f_u(i) > 0 \\ 0, & \text{diğer durumda.} \end{cases} \quad (4.5)$$

En son aşamada, her bir düğüm için üst- k -yakınlık komşularının öznitelik vektörleri toplanarak ortak bir öznitelik vektörü oluşturulur. u düğümü için i . özniteliğinin değeri Denklem 4.6’de verildiği şekilde elde edilir. f'_u vektörü u düğümünün öznitelik üretim sürecinde üretilen özniteliklerini temsil eder.

$$f'_u(i) = \sum_{j \in k p^u}^{k'} f_j(i). \quad (4.6)$$

4.1.3. Öznitelik Seçimi

Öznitelik üretim aşamasında öznitelik toplama ile u düğümü için yeni dolaylı öznitelikler elde edilmiştir. Öznitelik seçim sürecinde ise üretilen yeni öznitelik vektörü kullanılarak, bu düğümü daha iyi temsil eden bir öznitelik alt kümesi seçilmesi amaçlanır. Bu nedenle, üretilen öznitelik vektörü f'_u üzerinden z -ortak öznitelikler olarak adlandırılan bir öznitelik alt kümesi seçilir. Hangi özniteliklerin seçileceğine karar vermek için, her bir özniteliğe ait seçme olasılığı hesaplanır. u düğümü için i . özniteliğin seçilme olasılığının nasıl hesaplanacağı Denklem 4.7’de gösterilmiştir.

$$p(u, i) = \frac{f'_u(i)}{k'} \quad (4.7)$$

$p(u, i)$, u düğümü için i . özniteliğın seçılme olasılığdır. En yüksek seçılme olasılığına sahip olan z adet öznitelik u düğümü için seçilir. Düğüme ait z -ortak öznitelikler kümesi Denlem 4.8’de verildiğı şekilde elde edilir.

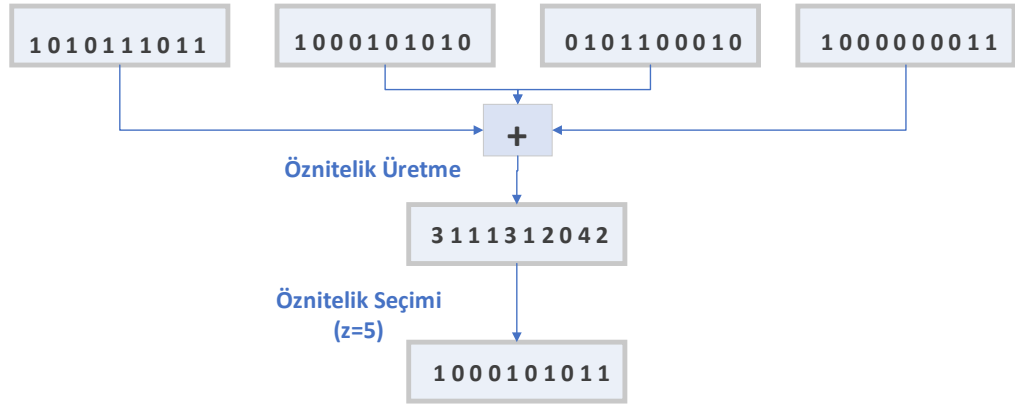
$$[zp_1^u, zp_2^u, \dots, zp_{z'}^u] = \max([p(u, 1), p(u, 2), \dots, p(u, D)] > 0, z) \quad (4.8)$$

zp^u , u düğümü için seçilen özniteliklerin kümesini belirtir. z -ortak öznitelikler kümesi, düğümün doğrudan veya dolaylı özniteliklerini içerir. Her zaman, olasılıkları sıfırdan büyük olan z adet öznitelik olmayabilir. Bu nedenle, $z' < z$ olmak üzere z' parametresi seçılme olasılığı sıfırdan büyük olan ortak özniteliklerin sayısını gösterir.

u düğümünün yeniden yapılandırılmış öznitelik vektörünü temsil eden x_u vektörü, z -ortak öznitelikler temel alınarak oluşturulur. u düğümüne ait doğrudan öznitelikler olarak bilinen orjinal öznitelikleri muhafaza edilecek şekilde, dolaylı öznitelik olarak adlandırılan z -ortak öznitelikler düğüm ile ilişkilendirilir. Eğer i özniteliğı u düğümünün z -ortak özniteliklerden biri ise vektörün bu özniteliğı $x_u(i)$ karşılık gelen değeri 1’e eşitlenir.

$$x_u(i) = \begin{cases} 1, & i \in zp^u \\ 0, & \text{diğer durumda.} \end{cases} \quad (4.9)$$

Şekil 4.6’de hedef bir düğüm için NetBFC yöntemine ait öznitelik üretimi ve seçiminin nasıl olacağı bir örnek üzerinde gösterilmiştir. Şekilde dört farklı düğümün öznitelik vektörü verilmiştir. Bu öznitelik vektörleri toplanarak hedef düğüm için tek bir öznitelik vektörü elde edilmiştir. Bu öznitelik vektörü üzerinden $z = 5$ olacak şekilde en yüksek değere sahip öznitelikler seçilerek, düğümün z -ortak öznitelikleri belirlenmektedir. En yüksek değere sahip özniteliklerin indisleri 1, 5, 7, 9, 10’ dur. Bu indislerdeki özniteliklerin değerleri 1’ eşitlenerek düğüm için nihai yapılandırılmış öznitelik vektörü elde edilmektedir.



Şekil 4.6: Hedef düğüm için öznitelik üretme ve seçme süreci örneği.

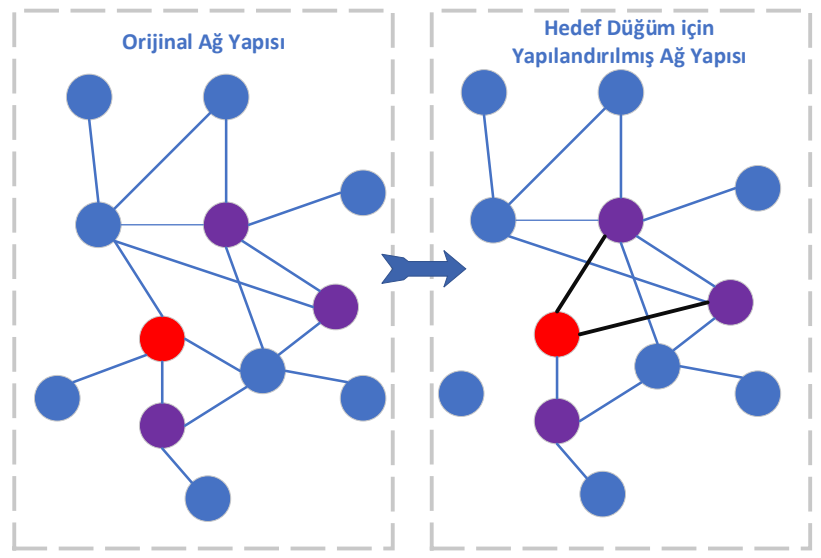
4.1.4. Çizge Yapılandırma

Her bir düğüm için üst- k -yakınlık komşuları kümesi, düğümün doğrudan veya dolaylı komşularını içerir. Bir düğümün doğrudan komşuları, düğümün karşılık düştüğü çizge de bağlantılı olduğu komşularıdır. Dolaylı komşular, çizge de düğümün direk bağlantılı olmadığı ama ikinci dereceden komşuluk benzerliği gösterdiği düğümlerdir. Bir düğüm için dolaylı komşular, düğüm ile öznitelik yakınlığı gösteriyor ise öznitelik yapılandırma sürecine olumlu katkıda bulunmaktadır. NetBFC yöntemi, düğümün öznitelik vektörünü güncellerken, aynı zamanda karşılık düştüğü çizgedeki komşularını da güncellemektedir. Böylelikle, aynı kümede olma ihtimali yüksek olarak belirlenen üst- k -yakınlık komşuları (doğrudan veya dolaylı komşuları) ile olan birliktelikleri muhafaza edilir. Bir düğüm doğrudan bir düğüme komşu değilse, fakat üst- k -yakınlık komşularından biri ise iki düğüm arasında bir bağlantı oluşturulur. Bu nedenle, düğümün bağlantı vektörü, bazı doğrudan bağlantıları koruyarak ve bazı dolaylı bağlantılar ekleyerek yeniden düzenlenir. Eğer düğüm v düğüm u 'nun üst- k -yakınlık komşularından biri ise, düğüm u 'dan v 'ye bir kenar bağlantısı eklenir. Bu durumda, bu kenara karşılık gelen yeni bağlantının değeri $l_{u,v} = 1$ olarak güncellenir.

$$l_{u,v} = \begin{cases} 1, & v \in kp^u \text{ ya da } e_{u,v} > 0 \\ 0, & \text{diğer durumda.} \end{cases} \quad (4.10)$$

Şekil 4.7'de önerilen yöntem NetBFC'nin çizge yapılandırma örneği verilmiştir.

Kırmızı ile renklendirilmiş hedef düğüm için seçilen üç adet yakınlık komşusu mor renk ile gösterilmiştir. Ağ yapılandırılması sürecinde, yakınlık komşuları ile hedef düğüm arasında bağlantı var ise korunmuş, yok ise yeni bağlantılar eklenmiştir. Orjinal ağ yapısında hedef düğüm ile bağlantılı olan düğümlerin, eğer hedef düğümün yakınlık komşusu olarak seçilmediler ise, bağlantıları da kaldırılmıştır.



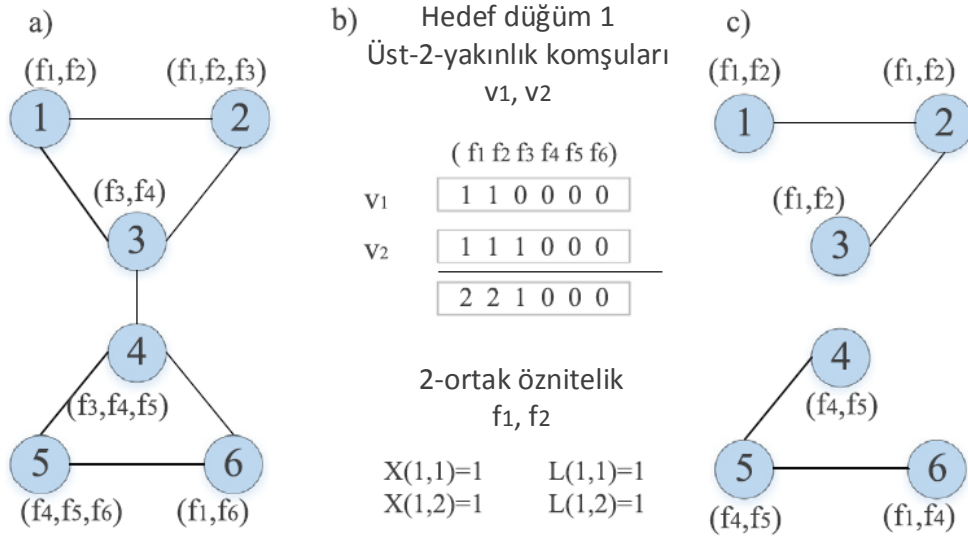
Şekil 4.7: Çizge Yapılandırma Örneği.

4.1.5. Önerilen Yöntemin Algoritması

Algoritma 1’de, önerilen yöntem NetBFC’e ait süreç ayrıntılı olarak verilmektedir. $G = V, E, F$ yapısı göz önüne alındığında, algoritma, $X \in \{0, 1\}^{N \times S}$ olan ikili öznitelik matrisini oluşturmak için k, z, α ve β parametrelerini girdi olarak alır. Başlangıçta $i = 1$ olarak ayarlanmıştır. Böylece, 1. düğümden başlanarak üst- k -yakınlık komşuları kp^1 elde edilir ve sonrasına f'_1 ’e ait yeni öznitelik vektörü üretilir. 1. düğüm için özniteliklerin seçilme olasılıkları sp^1 hesaplanır. Seçilme olasılıkları azalan sırada sıralanarak, en yüksek olasılığa sahip z adet öznitelik seçilir. Yukarıda belirtilen öznitelik üretme ve seçim adımları, ağdaki her düğüm için yerel olarak uygulanır. $Iter$ parametresini kullanarak, bu adımlar tekrarlanabilir. Bu işlemler, bağlantı yapısı sabit bir bağlantı yapısına yakınsayana kadar diğer bir değişle grafa ait bağlantı yapısı değişim göstermeyene kadar yinelenabilir. Her yinelemenin sonunda, hiçbir düğüm ile

ilişkilendirilmemiş bazı öznitelikler X matrisinden kaldırılır. Çünkü bu özniteliklerin değeri her düğüm için 0'a eşit olacaktır. Böylelikle, X matrisinin boyutsallığında azalma olacaktır.

Basit bir ağ bağlantılı veri örneği üzerinde yönteminin algoritmasının çalışma süreci Şekil 4.8'de örneklendirilmiştir. Şekil 4.8 (a)' da verilen ağ bağlantılı veri, altı düğümünden ve düğümleri temsil eden altı farklı öznitelikten oluşmaktadır. Şekil 4.8 (b), hem k hem de z değeri 2 olarak ayarlandığında bağlantı yapısının ve v_1 düğümünün özniteliklerinin nasıl yeniden yapılandırıldığını gösterir. İterasyon parametresi $Iter = 3$ olduğunda, her düğüm için elde edilen nihai öznitelikler ve yeniden oluşturulmuş bağlantı yapısı Şekil 4.8 (c)'de verilmiştir.



Şekil 4.8: NetBFC yöntemi ile öznitelik ve bağlantı oluşturma süreci örneği.

Algoritma 1 : NetBFC

Girdi: $E, F, N, k, z, iter, \alpha, \beta$

Çıktı: X

$L \leftarrow E + I, X \leftarrow F$

for $t = 1$ to $Iter$ **do**

$D \leftarrow X$ 'in boyutu,

X^t ' yi kullanarak Denklem 4.1 ile ρ_A^t ' yi hesapla,

L^t ' yi kullanarak Denklem 4.2 ile ρ_S^t ' yi hesapla,

α ve β ' yi kullanarak Denklem 4.3 ile ρ_{AS}^t ' yi hesapla,

for $i = 1$ to N **do**

k parametresini kullanarak Denklem 4.4 ile $[kp_1^i, kp_2^i, \dots, kp_{k'}^i]^t$ ' yi hesapla,

Denklem 4.6 göre f'_i vektörünü oluştur,

$l_{i,[kp_1^i, kp_2^i, \dots, kp_{k'}^i]}^{t+1} \leftarrow 1$

$l_{i,\sim[kp_1^i, kp_2^i, \dots, kp_{k'}^i]}^{t+1} \leftarrow 0$

Denklem 4.7 ile $[p(i, 1), p(i, 2), \dots, p(i, D)]^t$ ' yi hesapla,

z parametresini kullanarak Denklem 4.8 ile $[zp_1^i, zp_2^i, \dots, zp_{z'}^i]^t$ ' yi elde et,

$x_{i,[zp_1^i, zp_2^i, \dots, zp_{z'}^i]}^{t+1} \leftarrow 1$

$x_{i,\sim[zp_1^i, zp_2^i, \dots, zp_{z'}^i]}^{t+1} \leftarrow 0$

end for

for $i = 1$ to D **do**

if $\sum_{j=1}^N x_{j,i}^{t+1} == 0$ **then**

Elimine etmek için i özniteliği işaretle.

end if

end for

İşaretlenmiş öznitelikleri X^{t+1} matrisinin sütunlarından sil.

$D \leftarrow X^{t+1}$ ' in boyutu.

end for

return X

5. DENEYSEL SONUÇLAR

Bu bölümde, önerilen öznitelik oluşturma yönteminin performansı bir kümeleme algoritması kullanarak değerlendirilmekte ve literatürde var olan yöntemler ile kıyaslanmaktadır. Ayrıca, NetBFC yöntemi bir ön işleme adımı olarak literatürde var olan öznitelik seçim ve çıkarım yöntemleri ile birlikte kullanıldığında yöntemlerin performansının nasıl etkilendiği analiz edilmektedir. Önerilen yöntemin girdi olarak aldığı parametrelerin analizi yapılmakta ve son olarak yöntemin çalışma zamanı hakkında detaylı inceleme verilmektedir.

5.1. Veri Seti

Yöntemimizin başarısını ölçmek ve literatürde var olan diğer yöntemler ile kıyaslamak için literatürde sıklıkla kullanılan (Cataltepe vd., 2014; Li vd., 2016; Huang vd., 2017b) 4 farklı veri seti seçilmiştir. Bunlar, WebKB, Cora, Citeseer ve BlogCatalog verisetleridir. Bu veri setleri hakkında detaylı bilgi aşağıda verilmektedir.

- WebKB veriseti (Slattery ve Craven, 1998), çeşitli üniversitelerin bilgisayar bilimleri bölümlerinden (Cornell University, The University of Texas and The University of Wisconsin) Ocak 1997’ de CMU metin öğrenme grubunun Dünya Çapında Bilgi Bankası (Web-Kb) projesi tarafından toplanan web sayfalarını içermektedir. WebKB veri setinin tamamı, bilgisayar bilimleri bölümlerinden toplanan yedi sınıf web sayfasından oluşur. Bu web sayfaları öğrenci, fakülte, kurs, proje, bölüm, personel ve diğerleri olarak kategorize edilmiştir. Bu veri setinde, iki web sayfası arasında bir köprü varsa, sayfalar birbiriyle bağlantılıdır. Web sayfaları arasındaki bağlantı bilgileri WebKB’e ait çizge yapısını oluşturur. Bu verisetine <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/> adresinden ulaşılabilir.
- Cora veriseti (McCallum, 2017) Andrew McCallum tarafından elle gruplandırılarak oluşturulan makine öğrenme makalelerinin bibliyografik kayıtlarını (başlık, yazar, özet, referanslar) içerir. Makalelerin bibliyografik kayıtlarında geçen her

bilgi makalelerin özniteliklerini oluşturur. Birbirini alıntılaman makaleler arasında (yazar ve referans bilgileri ile) bir bağlantı oluşur. Bu bağlantılar, Cora'ya ait çizge yapısını oluşturulur. Cora verisetinde bulunan makaleler olay tabanlı, genetik algoritmalar, sinir ağları, olasılıksal modeller, kural tabanlı, teori yada takviyeli öğrenme olarak belirlenmiş yedi farklı sınıf ile kategorize edilmiştir. Cora veri setine <https://people.cs.umass.edu/mccallum/data.html> adresinden ulaşılabilir. Bu adreste, bahsedilen yedi sınıftan farklı bir çok grupta kategorize edilmiş Cora veriseti bulunmaktadır.

- Citeseer veriseti (Sinka ve Corne, 2002), daha çok bilgisayar bilimi literatürünü içeren bir bilimsel literatür dijital kütüphanesi ve arama motoru olan CiteSeer kütüphanesinden toplanmış bilimsel makalelere ait bibliyografik kayıtları içerir. Cora veri setine benzer bir veri setidir. Citeseer verisetinde makaleler AI (Artificial Intelligence), DB (Database), ML (Machine Learning), IR (Information Retrieval) ve HCI (Human Computer Interaction) olarak kategorize edilmiştir. Citeseer verisetine <http://www.cs.umd.edu/sen/lbc-proj/LBC.html> adresinden ulaşılabilir.
- BlogCatalog veriseti, BlogCatalog olarak bilinen kullanıcıların birbirleriyle etkileşimde bulunduğu ve bir arkadaşlık ağı oluşturduğu bir blog yazar topluluğundan elde edilmiştir. BlogCatalog uygulamasında kullanıcıların bloglarının kısa bir açıklaması olarak anahtar kelimeler (tags) üretmesine izin verilir. Bu anahtar kelimeler blog yazarları için öznitelikleri oluşturmaktadır. Kullanıcılar ayrıca bloglarını ön tanımlı kategoriler altında kaydederler. Bu kategoriler kullanıcılara ait sınıf bilgisini oluşturur. BlogCatalog verisetine <http://socialcomputing.asu.edu/datasets/BlogCatalog3> adresinden ulaşılabilir.

Bu verisetlerinde bulunan her nesne, söz konusu özneliliğın (kelime, etiket vb.) sözlükte bulunduğunu gösteren 0/1 değerli bir vektör ile tanımlanır. Tablo 5.1'de ayrıca tüm veri kümeleri için ortalama homofili değerleri verilmiştir. Ağdaki bir düğüm için homofili derecesi, bu düğümle aynı etikete sahip olan komşularının sayısı ile belirlenir. Veri kümesinde kullanılan her nesnenin homofili derecesi Denklem 5.1 (Cataltepe vd., 2014) kullanılarak hesaplanmıştır.

$$H(u) = \frac{1}{N(u)} \sum_{v \in N(u)} [r(u) == r(v)] \quad (5.1)$$

u düğümü için $N(u)$ düğümün komşu düğümler kümesini ve $r(u)$ düğümün sınıf etiketini temsil etmektedir. Ağa ait homofili derecesi ise ağdaki düğümlerin homofili derecelerinin ortalaması olarak hesaplanmaktadır.

Tablo 5.1: Veri setine ait detaylı bilgiler.

Özellikler	WebKB	Cora	Citeseer	BlogCatalog
Düğüm Sayısı	647	2,708	3,312	5,196
Öznitelik Sayısı	1,703	1,433	3,703	8,189
Bağlantı Sayısı	1,063	5,429	4,598	171,743
Küme Sayısı	5	7	6	6
Homofili (Ort.)	0.14	0.82	0.71	0.39

5.1. Değerlendirme Kriterleri

Önerilen öznitelik oluşturma yönteminin bilinen bir kümelenme algoritması üzerindeki etkisini analiz ederek başarısını değerlendirmek için Normalleştirilmiş Karşılıklı Bilgi (NMI) ve Doğruluk (Acc) ölçütleri kullanılacaktır. Doğruluk değerinin hesabı Denklem 5.2’de verilmektedir.

$$Acc = \frac{\sum_{i=1}^n \delta(c_i = map(p_i))}{n} \quad (5.2)$$

Denklemde, p_i örnek i için bulunmuş etiketi, c_i ise örnek i ’nin gerçek etiket bilgisini temsil etmektedir. $map(p_i)$ fonksiyonu p_i ’yi Kuhn-Munkres algoritmasına göre (Frank, 2005) bir küme ile eşleştiren permütasyon haritalama fonksiyonudur.

Diğer bir karşılaştırma ölçütü olan NMI değeri ise aşağıdaki denklemde verildiği gibi hesaplanmaktadır. Denklemde verilen C gerçek kümeleri ve elemanlarını, C' ise yöntem tarafından tahmin edilen kümeler ve elemanlarını temsil etmektedir. $H(C)$ ve $H(C')$, C ve C' ’nin entropisini ve $MI(C, C')$ C ve C' arasındaki ortak bilgiyi (mutual information) temsil eder.

$$NMI(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))} \quad (5.3)$$

5.2. Deneysel Kurulum

Önerilen yöntem, geleneksel veri üzerinde çalışan iki öznitelik seçim yöntemi ve bir öznitelik çıkarımı yöntemi ile, ağ bağlantılı veri üzerinde çalışan bir öznitelik seçimi ve üç öznitelik çıkarım yöntemi ile karşılaştırılmaktadır. Karşılaştırma için kullanılan yöntemler gözetimsiz çalışan yöntemlerdir. Bu yöntemler aşağıda kısaca açıklanmaktadır:

- Laplace Skoru (LS) (He vd., 2006): Laplace skoru, yerel manifold yapısını en iyi koruyabilen özellikleri seçen bir öznitelik seçme yöntemidir. Geleneksel veri ile çalışır.
- UDFS (Yang vd., 2011): Ayırıcı analiz ve $l_{2,1}$ norm küçültme yöntemlerini kullanarak öznitelik seçimi yapmaktadır. Laplace skorundan farklı olarak, UDFS özellikleri tek tek değil ortak olarak seçer. Geleneksel veri ile çalışır.
- NETFS (Li vd., 2016): Birleşik bir öğrenme çerçevesinde gizli temsilleri ve öznitelik ağırlıklarını öğrenerek ağ verileri üzerinde denetimsiz öznitelik seçimi yapılmasını sağlayan bir yöntemdir. Ağ bağlantılı veri üzerinde çalışır.
- NMF (Berry vd., 2007): Negatif olmayan matris çarpanları ayrıştırma yaklaşımı ile öznitelik çıkarımı yapan bir yöntemdir. Geleneksel veri ile çalışır.
- LANE (Huang vd., 2017b): Ağ verisine ait yapısal yakınlıkları ve korelasyonları modelleyerek bunları düzgün bir şekilde düşük boyutlu bir gösterime gömen bir yöntemdir. Ağ bağlantılı veri üzerinde çalışır.
- AANE (Huang vd., 2017a): Ortak öğrenme sürecini karmaşık modelleme ve optimizasyon ile bir çok alt soruna ayrıştırılarak dağıtılmış şekilde yapılmasını sağlayan ağ verisi üzerinde düşük boyutlu bir gösterim oluşturan ağ gömüleme yöntemidir. Ağ bağlantılı veri üzerinde çalışır.
- SPEC-MINDIS (De Sa, 2005): Ağın içerik ve bağlantı bilgisini 2-görüntülü heterojen öznitelik olarak değerlendirerek her bir nesne için bu iki görüşlü yapıdan

tek bir vektör gösterimi elde etmeyi sağlayan bir çok-görüşlü yöntemdir. Ağ bağlantılı veri üzerinde çalışır.

- DeepWalk (Perozzi vd., 2014): Ağın yalnızca bağlantı bilgisinden elde edilen graf üzerinden düşük boyutlu bir gösterim sağlayan bir gömme yöntemidir. Ağ topolojisi üzerinde çalışır.
- Gat2Vec (Sheikh vd., 2019): Ağın içerik ve bağlantı bilgisini bir ikili grafa bütünleştirip ortak bir temsil oluşturan ve bu temsil üzerinden düşük boyutlu bir gösterim sağlayan bir gömme yöntemidir. Ağ bağlantılı veri üzerinde çalışır.

Bu yöntemlerin kurulumları için ilgili makalelerde açıklanan parametre değerleri kullanılmıştır. LS yöntemi için benzerlik ölçüsü olarak Cosinus yöntemi kullanılmıştır. NetFS, LANE ve AANE yöntemlerinin parametreleri, 10'luk bir artış adımıyla 0.01 ile 100 değeri arasında değiştirilerek kullanılmıştır. Yineleme sayısı, LANE ve AANE için 5 olarak alınmıştır. LS, UDFS, NMF ve SPECMINDIS yöntemleri için diğer parametreler önerilen parametreleriyle çalıştırılmıştır.

Üst- k -yakınlık komşular ve z -ortak öznitelikler için k ve z parametrelerinin değerleri [30, 50, 100, 150, 200, 250, 300, 350, 400] olarak belirlenmiştir. α ve β parametreleri 0 ile 1 arasında 0.1 oranında değiştirilerek kullanılmıştır. Diğer parametre γ ise arama stratejisi kullanılarak ve [0.001, 0.01, 0.1, 1] aralığında test edilerek belirlenmiş ve bu parametrenin değeri 0.1 olarak ayarlanmıştır. Yöntemin yineleme sayısı 10'a olarak eşitlenmiştir. Fakat, bazı testler için yöntem yakınsayana kadar yineleme devam etmiştir. Bu durumda yineleme sayısı 10'dan küçük yada büyük olabilmektedir. Kümeleme algoritması için K-means yöntemi kullanılmıştır. K-means algoritması, 10 kez çalıştırılarak, her veri seti için ortalama kümeleme performansı elde edilmiş ve test sonuçları için ortalama kümeleme değerleri verilmiştir.

5.4. Performans Değerlendirmesi

Deneylerde, NetBFC ile öznitelik oluşturma sürecinde indirgenecek öznitelik uzayına ait boyutsallık değerleri [100, 200, 400, 600] olarak seçilmiştir. Yöntem, yeniden oluşturulmuş öznitelik matrisinin boyutunda istenen seviyeside azalma olana kadar

Tablo 5.2: Yöntemlere ait K-means kümelemesi Acc değerleri.

Yöntemler	WebKB						Cora						Citeseer						BlogCatalog					
	100	200	400	600	100	200	400	600	100	200	400	600	100	200	400	600	100	200	400	600	100	200	400	600
Sadece öznitelik	44.61	44.61	44.61	44.61	32.39	32.39	32.39	32.39	38.53	38.53	38.53	38.53	38.53	38.53	38.53	38.53	18.12	18.12	18.12	18.12	18.12	18.12	18.12	18.12
LS	35.47	38.67	38.51	43.14	35.41	35.79	36.89	38.21	21.35	21.37	21.68	29.07	17.98	17.99	18.20	18.78	17.98	17.99	18.20	18.78	17.98	17.99	18.20	18.78
UDFS	51.78	53.45	54.73	62.09	30.68	31.24	35.47	35.65	24.66	28.46	42.91	43.11	17.80	17.81	17.83	18.01	17.80	17.81	17.83	18.01	17.80	17.81	17.83	18.01
NETFS	39.57	41.19	46.57	49.29	38.49	38.64	41.14	42.23	38.63	40.12	42.01	42.82	29.31	28.41	26.62	26.61	29.31	28.41	26.62	26.61	29.31	28.41	26.62	26.61
NMF	36.35	36.15	38.21	41.51	33.27	36.80	36.29	35.41	39.24	43.78	47.28	46.91	17.84	18.59	18.27	17.96	17.84	18.59	18.27	17.96	17.84	18.59	18.27	17.96
LANE	45.75	48.23	44.32	41.49	41.62	35.06	37.23	38.76	41.63	37.17	35.18	38.52	44.77	48.74	42.92	54.34	44.77	48.74	42.92	54.34	44.77	48.74	42.92	54.34
AANE	44.15	44.24	46.45	46.97	39.42	37.48	40.83	42.33	39.52	43.48	47.90	45.91	44.12	43.41	42.35	43.56	44.12	43.41	42.35	43.56	44.12	43.41	42.35	43.56
SPECMINDIS	33.72	35.92	39.31	32.92	31.89	25.78	24.85	23.44	36.27	29.61	24.98	24.08	51.82	35.71	31.13	28.64	51.82	35.71	31.13	28.64	51.82	35.71	31.13	28.64
DeepWalk	30.17	31.12	32.31	35.82	55.41	54.11	56.72	56.44	41.77	44.12	43.24	43.58	38.06	39.43	36.19	38.42	38.06	39.43	36.19	38.42	38.06	39.43	36.19	38.42
Gat2Vec	33.87	35.13	38.33	40.62	58.73	60.18	59.31	59.26	57.21	57.14	58.34	58.89	59.08	58.78	60.63	61.45	59.08	58.78	60.63	61.45	59.08	58.78	60.63	61.45
NetBFC	62.75	68.31	62.53	51.91	62.58	62.24	62.26	62.94	56.78	58.62	59.59	60.12	54.26	58.96	61.19	63.74	54.26	58.96	61.19	63.74	54.26	58.96	61.19	63.74

Tablo 5.3: Yöntemlere ait K-means kümelemesi NMI değerleri.

Yöntemler	WebKB						Cora						Citeseer						BlogCatalog					
	100	200	400	600	100	200	400	600	100	200	400	600	100	200	400	600	100	200	400	600	100	200	400	600
Sadece öz nitelik	0.192	0.192	0.192	0.192	0.150	0.150	0.150	0.150	0.150	0.150	0.150	0.150	0.167	0.167	0.167	0.167	0.167	0.167	0.167	0.167	0.038	0.038	0.038	0.038
LS	0.154	0.206	0.229	0.231	0.152	0.161	0.175	0.184	0.046	0.044	0.041	0.092	0.046	0.044	0.041	0.092	0.025	0.026	0.036	0.043	0.025	0.026	0.036	0.043
UDFS	0.093	0.117	0.143	0.276	0.049	0.041	0.163	0.179	0.068	0.067	0.196	0.211	0.068	0.067	0.196	0.211	0.024	0.022	0.024	0.025	0.024	0.022	0.024	0.025
NETFS	0.176	0.236	0.261	0.271	0.205	0.242	0.259	0.185	0.239	0.234	0.255	0.201	0.239	0.234	0.255	0.201	0.212	0.187	0.187	0.178	0.212	0.187	0.187	0.178
NMF	0.042	0.031	0.026	0.027	0.128	0.181	0.176	0.157	0.159	0.210	0.239	0.235	0.159	0.210	0.239	0.235	0.025	0.028	0.026	0.034	0.025	0.028	0.026	0.034
LANE	0.027	0.025	0.017	0.019	0.261	0.158	0.178	0.145	0.242	0.177	0.165	0.171	0.242	0.177	0.165	0.171	0.335	0.407	0.384	0.341	0.335	0.407	0.384	0.341
AANE	0.193	0.161	0.204	0.251	0.216	0.226	0.248	0.252	0.222	0.228	0.271	0.249	0.222	0.228	0.271	0.249	0.289	0.264	0.250	0.261	0.289	0.264	0.250	0.261
SPECMINDIS	0.069	0.056	0.064	0.059	0.156	0.092	0.077	0.054	0.139	0.073	0.037	0.027	0.139	0.073	0.037	0.027	0.361	0.186	0.138	0.099	0.361	0.186	0.138	0.099
DeepWalk	0.048	0.099	0.141	0.169	0.402	0.387	0.422	0.442	0.191	0.201	0.207	0.216	0.191	0.201	0.207	0.216	0.219	0.198	0.194	0.261	0.219	0.198	0.194	0.261
Gat2Vec	0.049	0.055	0.089	0.069	0.465	0.453	0.412	0.443	0.377	0.355	0.341	0.342	0.377	0.355	0.341	0.342	0.421	0.398	0.441	0.439	0.421	0.398	0.441	0.439
NetBFC	0.308	0.310	0.347	0.341	0.379	0.436	0.444	0.457	0.295	0.319	0.349	0.348	0.295	0.319	0.349	0.348	0.374	0.417	0.449	0.477	0.374	0.417	0.449	0.477

yinelenmektedir. Bu yinelemeler sırasında oluşan yeni öznitelik temsilleri K-means algoritması kullanılarak test edilmiştir. Elde edilen ortalama kümeleme performansları Tablo 5.2 ve Tablo 5.3’de verilmektedir. NMI ve Acc için elde edilen yüksek değerler daha iyi bir kümeleme başarısı elde edildiğini gösterir. Diğer yöntemlerle karşılaştırıldığında, önerilen NetBFC yönteminin tutarlı olarak veri setleri için yüksek kümeleme performansı elde ettiği gözlemlenmektedir. WebKB veri seti için, UDFS yöntemi, boyut 600 olarak ayarlandığında bizim yöntemimizden daha yüksek doğruluk sağlamıştır. Bununla birlikte, aynı veri seti üzerinde yöntemimiz WebKB veri seti için daha yüksek NMI değeri elde etmiştir. NetBFC yöntemine en yakın sonuçlar Gat2Vec yöntemi ile edilmektedir. Boyut 100 iken Citeseer ve BlogCatalog verisetleri için Gat2Vec yöntemi ile daha yüksek doğruluk değerleri ve NMI değerleri elde edilmiştir. Fakat, neredeyse tüm verisetleri için yöntemimiz çoğunlukla deneyde kullanılan yöntemlerden daha yüksek Acc ve NMI değerleri sağlamaktadır. Özellikle boyutsallık azaltımı 400 ve 600 aralığında en yüksek performans değerlerine NetBFC yöntemi ulaşmaktadır. Bu durum, daha yüksek boyutsallık değerlerinde önerilen yöntemin daha iyi sonuçlar verebildiğini göstermektedir. Fakat, düşük boyutsallık değerlerinde de karşılaştırmak için kullanılan dokuz farklı yöntem ile kıyaslandığında çoğunluğundan daha yüksek performans değerleri sağladığı görülmektedir. NetBFC, deneydeki en düşük boyutsallık değeri 100 boyutsallığında bile WebKB, Cora, Citeseer ve BlogCatalog için sırasıyla % 40.66, % 93.21, % 47.36 ve % 199.45 oranında iyileştirme sağlamıştır. Bu durumda, NetBFC yönteminin, ağ bağlantılı veriler için kümeleme performansında dikkate değer bir düşüş olmadan, önemli derecede boyut azaltımı sağlayabildiği anlaşılmaktadır.

NetBFC yönteminin veri setleri üzerindeki performansını daha detaylı gözlemek için k ve z parametresi 250 değerine ve yineleme sayısı 6 değerine eşitlenerek Tablo 5.4’de doğruluk, NMI, boyut ve bunlara ait standart sapma değerleri verilmiştir. Tablo 5.4’ü temel alarak yöntemin sonuçlarına bakıldığında, WebKB ve Cora veri setlerinin doğruluk değerlerinin yineleme boyunca artış gösterdiği anlaşılmaktadır. Aynı zamanda belli bir yineleme sayısından sonra Acc ve NMI değerlerine ait standart sapma değerlerinde azalış görülmektedir. Bu durum, belli bir yinelemeden sonra yöntemin daha kararlı bir hale geldiğini göstermektedir. Bu parametre değerlerinde Citeseer ve BlogCatalog verileri için yineleme sayısı 1’e eşit iken daha yüksek Acc ve NMI değerleri elde edilmiştir. Boyut azaltımı ile birlikte Acc ve NMI değerlerinde azalma

gözlemlenmesine rağmen, elde edilen performans değerlerinin, Tablo 5.2 ve Tablo 5.3 temel alındığında, karşılaştırmak için kullanılan yöntemlerin çoğunun performans değerlerinden daha iyi olduğu anlaşılmaktadır. Elde edilen boyut azaltımı ile ilgili sonuçlara bakıldığında, NetBFC yönteminin her iterasyonda belli bir miktarda boyut azaltımı yaptığı görülmektedir. Bu boyut değerlerine göre en yüksek azalış değerleri 2. iterasyonda elde edilmiştir. Her veri setinde her yineleme sırasında farklı boyut azaltımları gerçekleştirdiği görülmektedir. Verideki nesne sayısına ve verideki öznitelik dağılımındaki değişime göre boyutsallıkta farklı miktarlarda azalma gerçekleştiği anlaşılmaktadır.

Tablo 5.4: NetBFC yöntemine ait parametreler $k=250$ ve $z=250$ iken tüm verisetleri için elde edilen kümeleme performans değerleri.

		Yineleme Sayısı					
NetBFC		1	2	3	4	5	6
WebKB	Acc	51.901	53.138	60.340	61.870	63.153	63.184
	Acc-Std	1.739	0.721	1.593	0.515	0.149	0.160
	NMI	0.329	0.308	0.281	0.258	0.269	0.269
	NMI-Std	0.004	0.002	0.012	0.005	0.001	0.001
	Boyut	566	348	315	307	306	306
Cora	Acc	53.682	51.688	54.845	57.315	60.218	60.325
	Acc-Std	4.093	2.926	2.722	4.085	1.652	2.190
	NMI	0.353	0.329	0.332	0.337	0.351	0.350
	NMI-Std	0.013	0.018	0.016	0.010	0.003	0.001
	Boyut	1144	316	249	236	234	233
Citeseer	Acc	61.066	55.556	53.237	55.383	53.964	54.671
	Acc-Std	3.578	2.106	5.263	0.597	2.269	0.519
	NMI	0.361	0.310	0.287	0.280	0.281	0.272
	NMI-Std	0.014	0.016	0.029	0.012	0.015	0.011
	Boyut	2883	562	447	426	422	422
BlogCatalog	Acc	58.878	53.435	52.044	53.668	52.252	52.130
	Acc-Std	3.953	1.354	3.644	2.660	3.768	2.148
	NMI	0.414	0.350	0.343	0.339	0.342	0.336
	NMI-Std	0.010	0.013	0.027	0.010	0.015	0.021
	Boyut	3172	891	701	661	656	654

Tüm farklı k ve z parametreleri çalıştırılarak elde edilen kümeleme performans değerlerinin ortalaması ve standart sapma değerleri Tablo 5.5’de verilmektedir. k ve z parametreleri [30, 50, 100, 150, 200, 250, 300, 350, 400] aralığında çalıştırılmış ve her

veriseti için 81 farklı öznitelik temsili elde edilmiştir. Bu temsiller üzerinde K-means algoritması çalıştırılarak kümeleme performans değerleri elde edilmiş ve kümeleme performans değerleri ortalaması ve standart sapmaları verilmiştir. Sonuçlara göre, Cite-seer ve BlogCatalog verisetleri için NetBFC yönteminin ilk iterasyonda daha yüksek sonuçlara ulaştığı gözlemlenmiştir. Cora ve WebKB verisetinde iterasyonlar arasında çok farklı bir başarı dağılımı yoktur. Standart sapma değerlerine bakıldığında, 81 farklı öznitelik temsili için çoğunluğunun benzer kümeleme başarısı sağladığı anlaşılmaktadır. Tüm farklı k ve z parametreleri çalıştırılarak istenilen boyutta elde edilen ortalama kümeleme performans değerleri ise Tablo 5.6’da verilmektedir. Bu deney sonucu için, NetBFC yöntemine boyut kısıtı konularak, istenen boyuta ulaştığında iterasyon sonlandırılmıştır. Standart sapma değerlerine bakıldığında, genel olarak parametre değerlerindeki değişime rağmen, NetBFC yöntemi ile çok farklı Acc ve NMI değerlerine ulaşılmadığı görülmektedir. Elde edilen ortalama performans değerleri de yöntemin istikrarlı ve tutarlı bir yöntem olduğunu göstermektedir.

Tablo 5.5: Tüm verisetler için farklı k ve z değerleri ile elde edilen ortalama kümeleme performans değerleri.

		Yineleme Sayısı					
NetBFC		1	2	3	4	5	6
WebKB	Acc	47.700	47.686	48.922	48.762	47.395	40.546
	Acc-Std	4.208	6.155	7.418	7.528	8.690	12.461
	NMI	0.300	0.271	0.251	0.229	0.211	0.195
	NMI-Std	0.045	0.062	0.081	0.099	0.104	0.119
Cora	Acc	55.675	54.266	55.172	55.360	55.454	55.302
	Acc-Std	4.004	3.988	3.529	3.635	3.865	4.107
	NMI	0.378	0.364	0.358	0.350	0.347	0.344
	NMI-Std	0.044	0.057	0.054	0.051	0.050	0.047
Citeseer	Acc	60.923	56.473	53.927	53.226	52.728	52.378
	Acc-Std	1.768	2.122	1.785	2.343	2.769	2.714
	NMI	0.355	0.315	0.292	0.283	0.277	0.274
	NMI-Std	0.018	0.024	0.018	0.014	0.012	0.011
BlogCatalog	Acc	62.206	55.317	52.963	52.684	53.019	53.140
	Acc-Std	5.628	5.475	3.792	2.204	1.867	1.835
	NMI	0.456	0.392	0.369	0.366	0.367	0.366
	NMI-Std	0.059	0.066	0.049	0.033	0.031	0.030

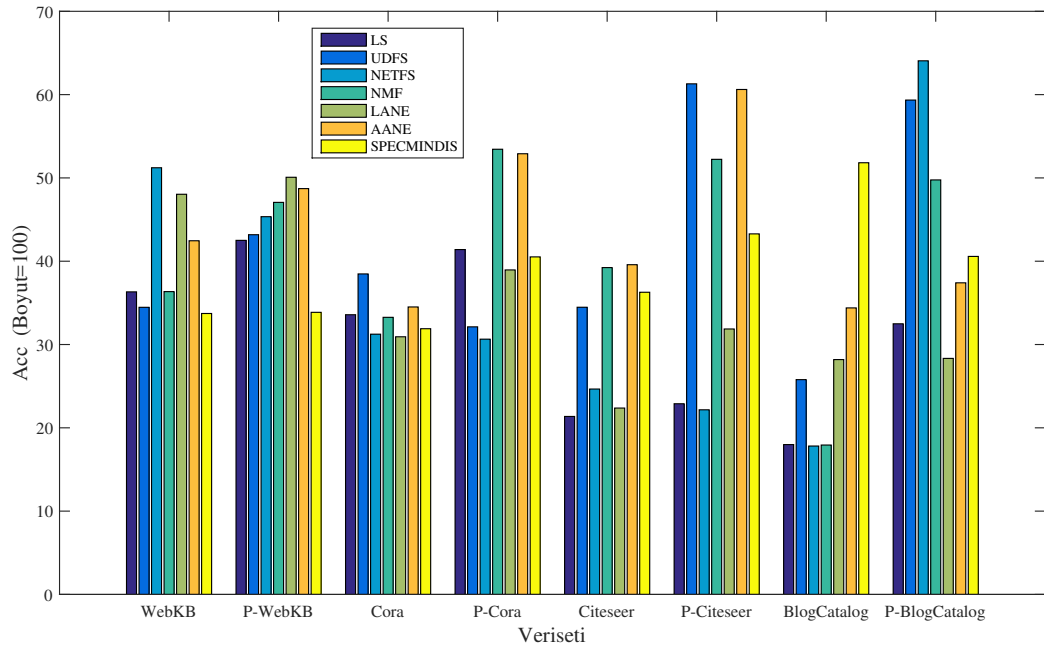
NetBFC yöntemi, diğer öznitelik seçim ve öznitelik çıkarım yöntemleri için bir

Tablo 5.6: Tüm verisetler için farklı k ve z parametreleri ile istenilen boyutta elde edilen ortalama kümeleme performans değerleri.

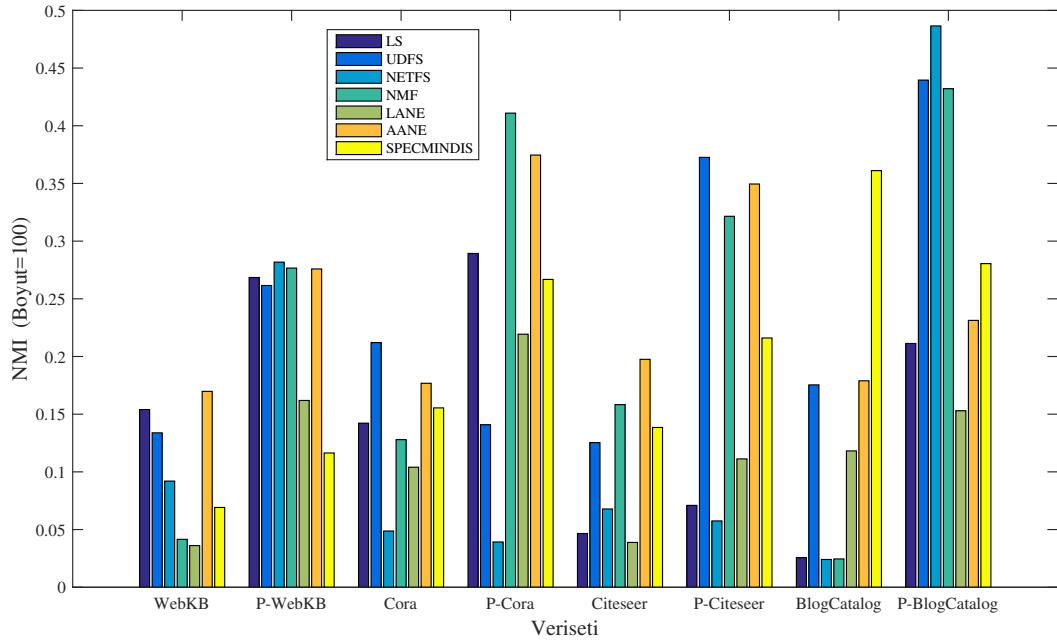
		Boyut			
NetBFC		100	200	400	600
WebKB	Acc	41.522	49.661	52.743	52.028
	Acc-Std	5.193	6.298	6.562	4.546
	NMI	0.151	0.236	0.275	0.287
	NMI-Std	0.081	0.084	0.069	0.047
Cora	Acc	56.742	57.330	54.635	55.333
	Acc-Std	3.303	2.884	3.750	3.625
	NMI	0.338	0.363	0.332	0.342
	NMI-Std	0.042	0.042	0.055	0.052
Citeseer	Acc	54.230	54.316	54.601	53.953
	Acc-Std	1.607	2.273	1.969	2.084
	NMI	0.284	0.282	0.289	0.281
	NMI-Std	0.010	0.017	0.018	0.021
BlogCatalog	Acc	52.396	54.321	52.059	53.232
	Acc-Std	1.895	2.560	2.948	3.506
	NMI	0.349	0.366	0.360	0.364
	NMI-Std	0.020	0.028	0.028	0.038

ön işleme aşaması olarak değerlendirilebilir. Bunun amaçla, veriseti üzerinde seçilen yöntemler çalıştırılmadan önce NetBFC yöntemi çalıştırılmıştır. Daha sonra elde edilen öznitelik temsilleri üzerinde diğer öznitelik seçim ve öznitelik çıkarım yöntemleri çalıştırılmıştır. NetBFC yöntemin ön işlem aşaması olarak ne kadar başarılı olduğunu ölçmek için yöntemlere ait NetBFC’li ve NetBFC’siz kümeleme performansları elde edilmiştir. NetBFC yöntemi ile ön işleme tabi tutulmuş WebKB, Cora, Citeseer ve BlogCatalog veri setlerini belirtmek için P-WebKB, P-Cora, P-Citeseer ve P-BlogCatalog etiketleri kullanılmıştır. Bu deney için $K = 100$, $z = 100$, $\gamma = 0.1$ ve $Iter = 1$ olarak ayarlanmıştır. α ve β parametrelerinin değeri 1’e eşitlenmiştir.

Şekil 5.1 ve Şekil 5.2’de, yöntemlerin NetBFC’li ve NetBFC’siz kümeleme performansları verilmektedir. NetBFC ile elde edilen öznitelik temsilleri üzerinde çalıştırılan LS, NMF, LANE ve AANE yöntemlerinin daha iyi Acc ve NMI değerleri sağladığı görülmektedir. NetBFC ile birlikte uygulanan UDFS yönteminin, Cora veri kümesi hariç, neredeyse tüm verisetleri için ön işlemsiz UDFS’ye kıyasla daha yüksek Acc ve NMI değerleri sağladığı görülmektedir. NetBFC ile birlikte çalıştırılan yöntemlerin



Şekil 5.1: NetBFC yöntemi ile ve NetBFC yöntemi olmadan çalıştırılan yöntemlerin kümeleme Acc değerleri.



Şekil 5.2: NetBFC yöntemi ile ve NetBFC yöntemi olmadan çalıştırılan yöntemlerin kümeleme NMI değerleri.

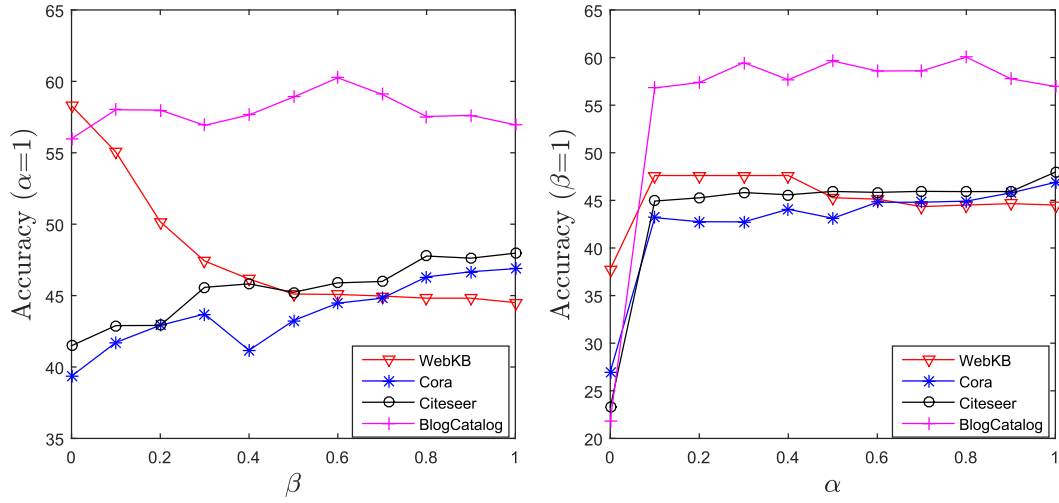
özellikle BlogCatalog veri seti için çok yüksek Acc ve NMI değerlerine ulaştığı gözlemlenmektedir. NetBFC ile ön işlemleri ile birlikte uygulanan SPECMINDIS yöntemi, Cora ve Citeseer veri setleri için ön işlemsiz SPECMINDIS' e kıyasla daha iyi sonuçlar elde etmiştir. Yöntemler birbirinden farklı olsa da, bu yöntemler NetBFC ile birlikte çalıştırıldığında neredeyse tüm yöntemlerin kümeleme performans değerlerinde önemli bir artış sağlanmıştır. Bu sonuçlara göre, NetBFC yönteminin bir ön işlem süreci olarak kullanılmaya uygun olduğu görülmektedir. Sonuç olarak, diğer yöntemlerin de başarısını artırması göz önüne alınırsa, yöntemin başarılı bir öznetelik temsili ürettiği anlaşılmaktadır.

5.5. Parametre Analizi

Önerilen yöntem NetBFC, bağlantılı veriye ait içerik ve yapısal bilginin sonuca katkısını ayarlayabilmek için sırasıyla α ve β parametrelerini kullanmaktadır. Ayrıca, üst- k -yakınlık komşularını belirlemek için girdi olarak k ve z parametrelerini kullanmaktadır. Bu bölümde, bu dört farklı parametrenin yöntem üzerindeki etkisi değerlendirilecektir.

α ve β parametrelerinin yöntemin sonuçları üzerindeki etkisini analiz etmek için öncelikle parametrelerden biri 1 değerinde sabit tutulup, diğer parametre 0.1 artış ile $[0 - 1]$ arasında değişecek şekilde ayarlanmıştır. Bu deney sırasında, k ve z parametreleri 400 değerinde sabit tutulmuştur. Şekil 5.3'de verilen test sonuçlarına göre, NetBFC yöntemi ile etkili sonuçlar elde etmek için α parametresinin sıfırdan büyük olması gerektiği görülmektedir. Yöntemin etkinliğinin daha çok β parametresine duyarlı olduğu gözlemlenmektedir. α parametresinin, homofoli derecesinin yüksek olduğu verilerde daha yüksek performans sağladığı görülmektedir. Cora ve Citeseer veri setleri için homofoli derecesinin, WebKB ve BlogCatalog verisetlerine kıyasla oldukça yüksek olduğu Tablo 5.1'de verilmektedir. Sonuçlara göre, $\alpha = 1$ ve $\beta = 1$ olduğu ayarlamalarda, Cora ve Citeseer veri setleri için en iyi doğruluk değerleri elde edilmiştir. Burada, hem içerik hem de bağlantı bilgisinin kümelemede eşit şekilde olumlu katkıda bulunduğu görülmektedir. WebKB, Cora, Citeseer ve BlogCatalog verisetleri için en yüksek doğruluk değerleri, α parametresi 1'e eşit iken, sırasıyla $\beta = 0$, $\beta = 1$, $\beta = 1$

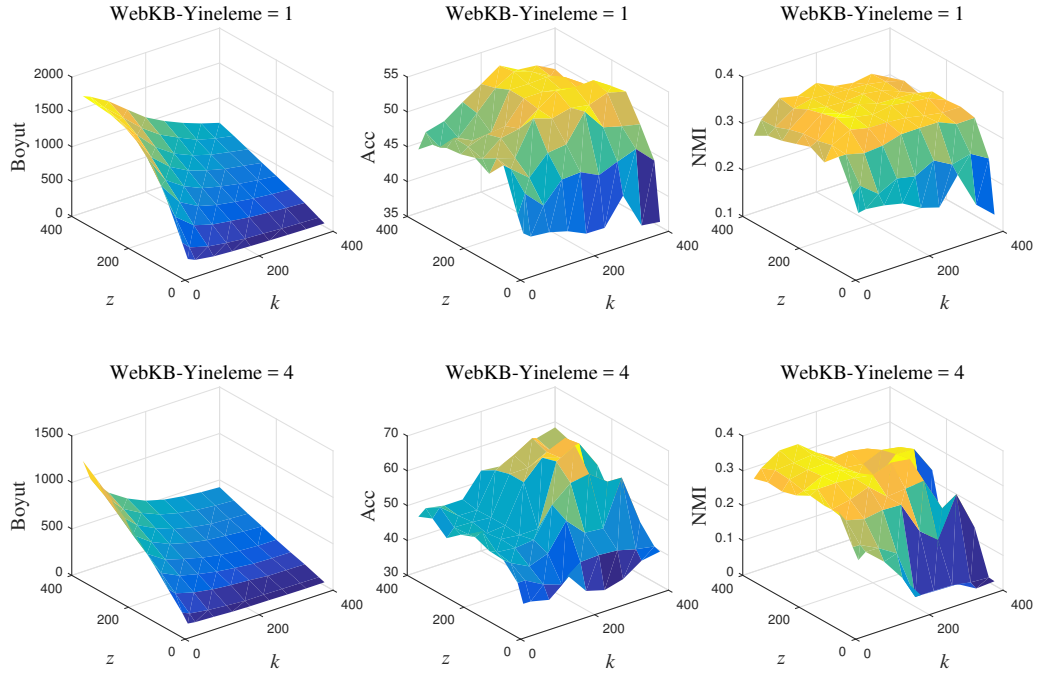
ve $\beta = 0.6$ 'da elde edilmiştir.



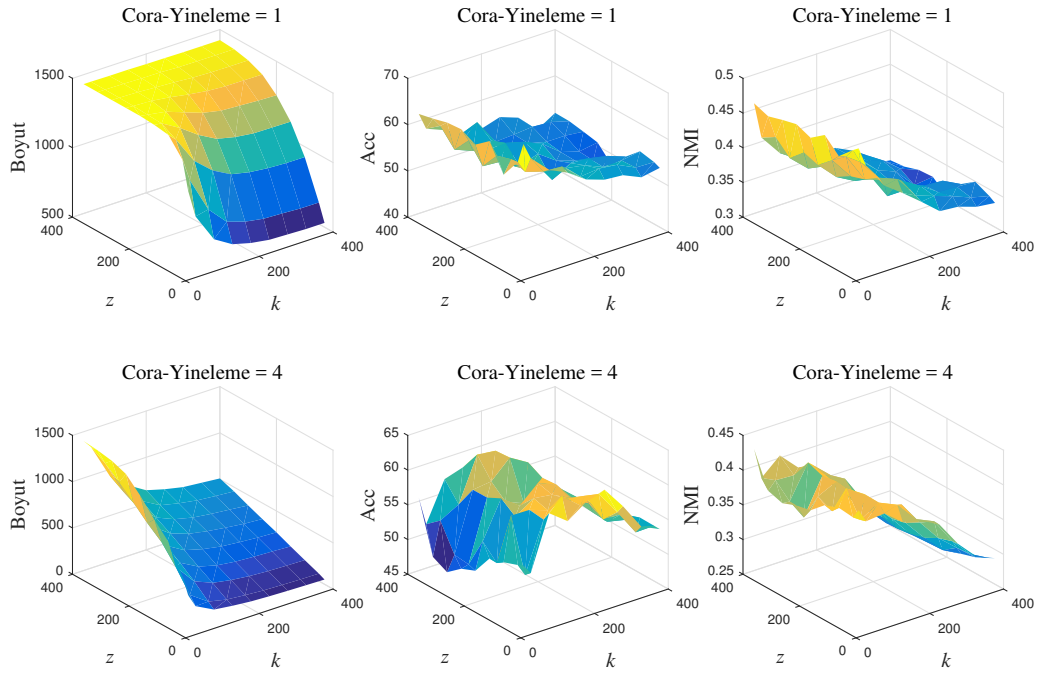
Şekil 5.3: Tüm verisetleri için farklı α ve β parametreleri ile elde edilen kümeleme Acc değeri.

k ve z parametrelerinin NetBFC yöntemi üzerindeki katkılarını ve özellikle yöntemin boyutsallığı azaltma sürecine olan etkilerini analiz etmek için önce her veriseti için en iyi performansın elde edildiği değerler belirlenmiştir. WebKB, Cora, Citeseer ve BlogCatalog verisetleri için performans değişimleri sırasıyla Şekil 5.4, 5.5, 5.6 ve 5.7' de gösterilmektedir. k değerinin yüksek ve z değerinin düşük olduğu durumlarda verilere ait boyutsallığın önemli miktarda azaltılabileceği anlaşılmaktadır. WebKB veri seti için boyut 228'e ve yineleme sayısı $iter$ 5'e eşit iken %68.18 ile en yüksek doğruluk değerine erişilmiştir. Diğer veri setleri için en yüksek doğruluk değerlerine 1. yinelemede ulaşılmıştır. En yüksek doğruluk değeri, Cora veri seti için boyut 1086'da iken %67.71, Citeseer veri seti için boyut 3694'de iken %63.92 ve BlogCatalog veri seti için boyut 7846'da iken %73.96 olarak bulunmuştur.

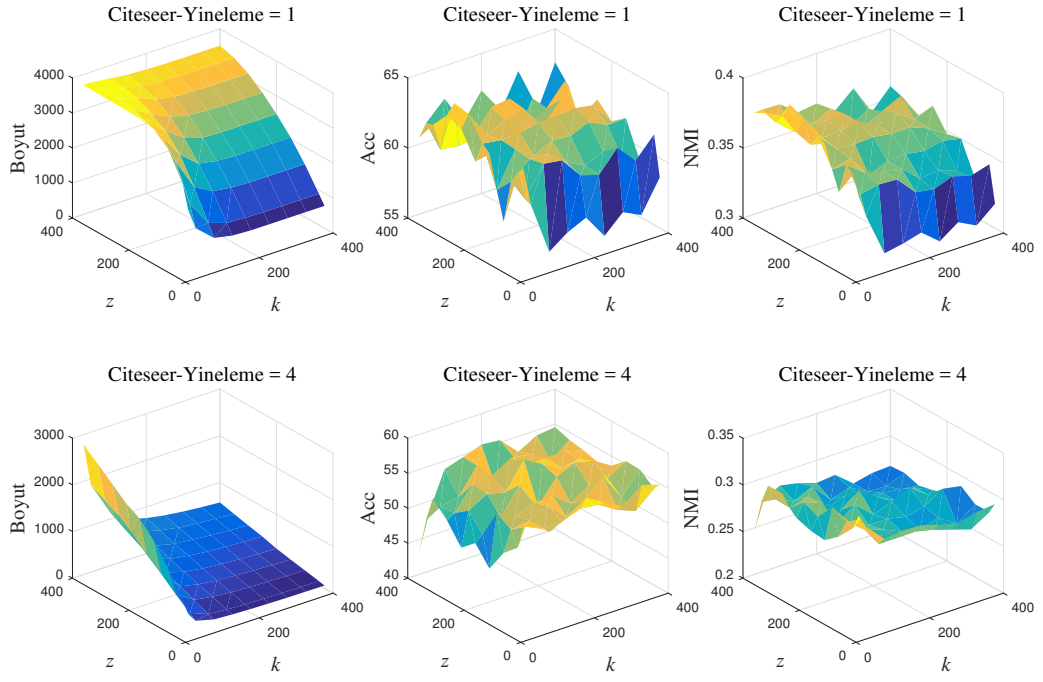
Şekil 5.8 ve 5.9'da, farklı yineleme değerleri için tüm veri kümelerinde elde edilen boyut ve doğruluk (Acc) değişimleri gösterilmektedir. NetBFC, oluşturulan ikili öznitelik matrisi yakınsayana diğer bir deyiş ile değişim göstermeyene kadar yinelenmiştir. Gözlemlenen sonuçlara göre, yöntemin 14. yinelemeden daha az yinelemede hızla yakınsadığı görülmektedir. Bununla birlikte, tüm veri kümeleri için boyutsallıktaki azalmanın, yaklaşık 4. yinelemeden sonra belirli değerlerde sabitlendiği gözlemlenmiştir. Benzer şekilde, her veri kümesinin kümeleme sonucunda elde edilen ortalama doğruluk değerlerinin belirli aralıklarda sabitlendiği görülmektedir. Sonuç olarak,



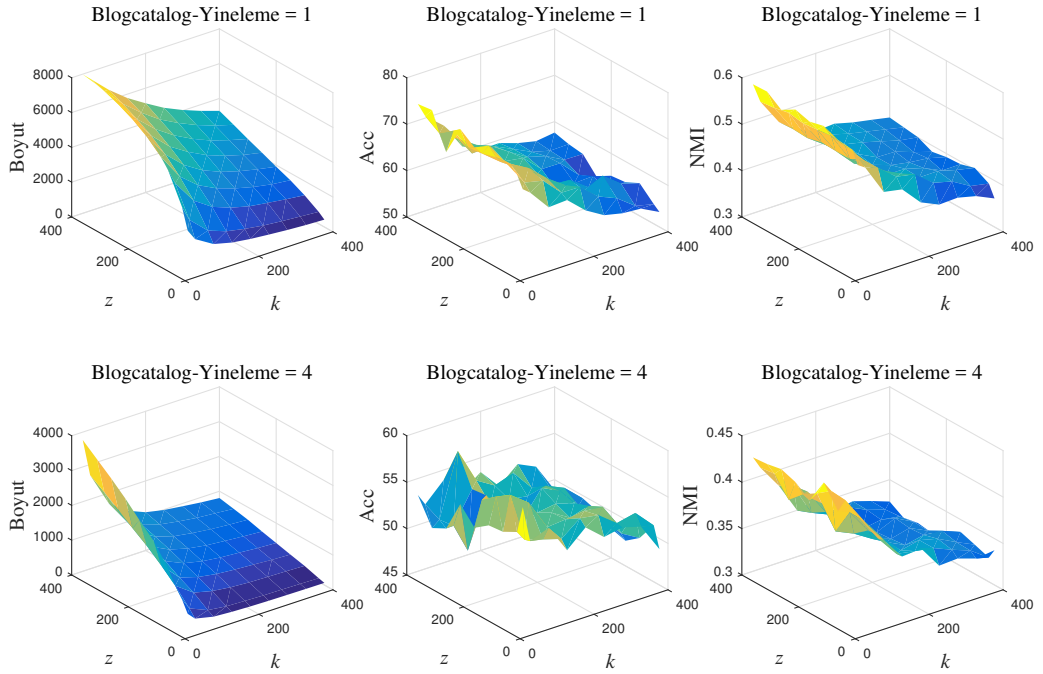
Şekil 5.4: Farklı k ve z parametre değerleri ile WebKB verisi için NetBFC yönteminin kümeleme performansı.



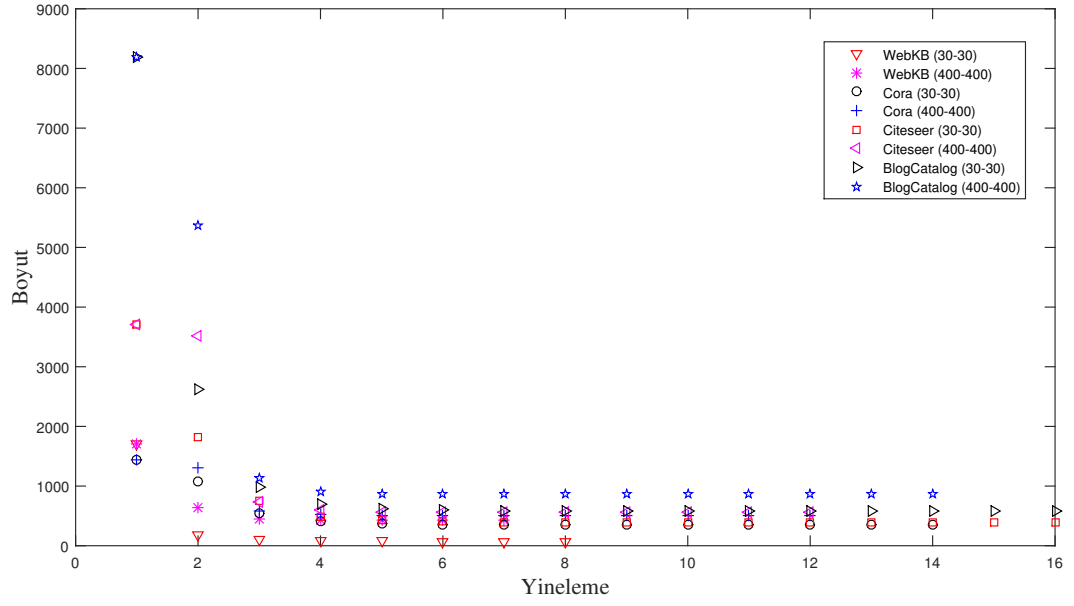
Şekil 5.5: Farklı k ve z parametre değerleri ile Cora verisi için NetBFC yönteminin kümeleme performansı.



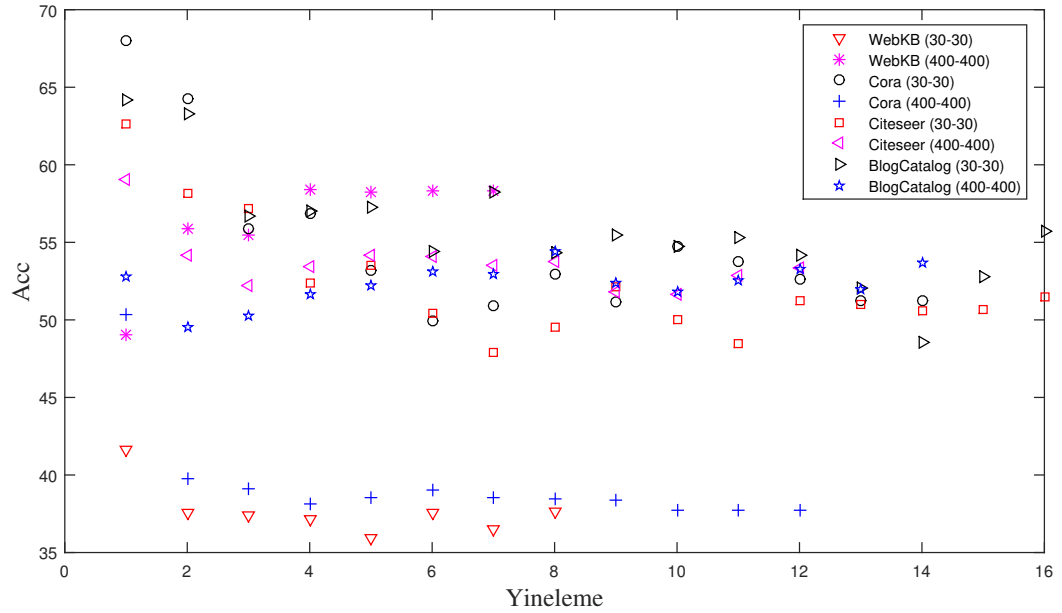
Şekil 5.6: Farklı k ve z parametre değerleri ile Citeseer verisi için NetBFC yönteminin kümeleme performansı.



Şekil 5.7: Farklı k ve z parametre değerleri ile BlogCatalog verisi için NetBFC yönteminin kümeleme performansı.



Şekil 5.8: Farklı k ve z parametre değerleri ile verisetlerinin boyutsallığında gözlenen değişimler.



Şekil 5.9: Farklı k ve z parametre değerleri ile tüm verisetlerine ait kümeleme Acc değerleri.

yöntemimizin tüm veri kümeleri için hızlı bir şekilde yakınsadığı anlaşılmaktadır.

5.6. Öznitelik Seçimi Analizi

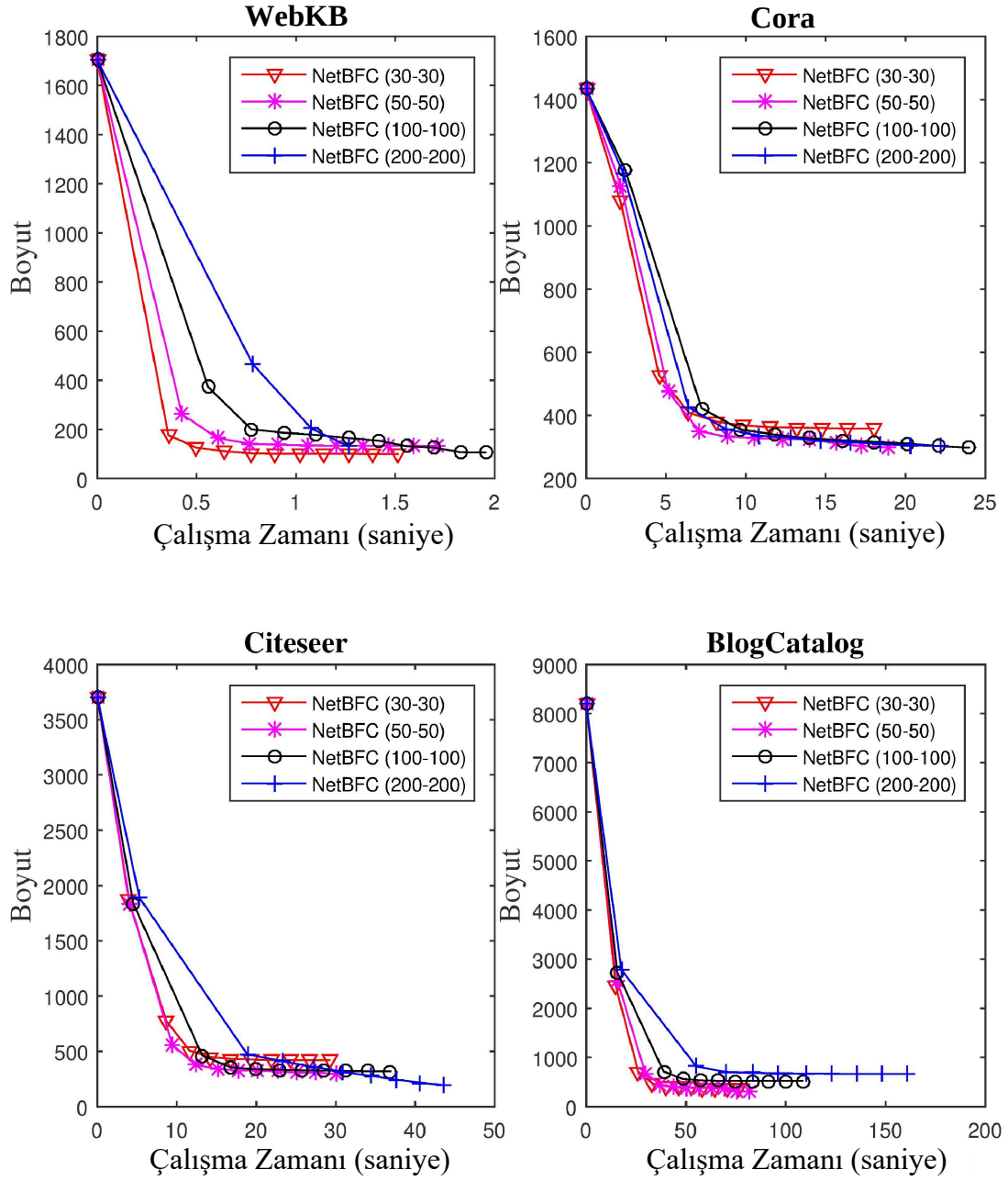
NetBFC yönteminin öznitelik yapılandırma ve seçme süreci ile nihai olarak yakınsadığı öznitelik uzayının ne kadar anlamlı olduğunu analiz etmek için Cora verisetine ait öznitelikler Tablo 5.7’de verilmektedir. Cora verisetine ait veri tabanı (DB), makine öğrenmesi (ML) ve ağ (NET) olmak üzere üç kümeye ait öznitelikler örneklendirilmiştir. Tablo 5.7’de bir küme için verilen her kelime o kümedeki nesnelerde sıklıkla gözlenen ve verilen kümeye kıyasla diğer iki küme de daha az gözlenen kelimelerden oluşmaktadır. Bir küme tüm nesneleri göz önüne alındığında bir özneliğin bu kümede kaç kere gözlemlendiği ve diğer kümelerde kaç kere gözlemlendiği hesaplanmıştır. Her bir öznitelik için bir küme puanı ve diğer kümelere olan uzaklığı elde edilmiştir. Bu değerlere göre, en yüksek küme puanı ve uzaklık değerine sahip öznitelikler Tablo 5.7’de listelenmiştir. Orjinal verisetine ait öznitelikler ve NetBFC ile seçilmiş olan öznitelikler incelendiğinde, NetBFC yöntemi öznitelik oluşturma ve seçme süreci sonucunda her bir küme ile ilişkili olarak anlamlı öznitelikler oluşturduğu görülmektedir.

Tablo 5.7: Cora verisetinin üç farklı grubuna ait orjinal ve NetBFC ile yapılandırılmış öznitelik dağılım örneği.

Orjinal Veri			Yapılandırılmış Veri		
DB	ML	NET	DB	ML	NET
non	comparison	active	data	bayesian	packet
concept	prediction	admissions	database	reasoning	multicast
neural	general	networks	logic	dynamic	traffic
genetic	based	routing	design	approach	wide
markov	low	internet	programs	algorithms	web
finite	layer	view	software	efficient	protocol
chain	design	protocol	finite	case	TCP/IP
neural	using	multicast	flow	classification	services
induction	belief	distributed	parallel	system	distributed

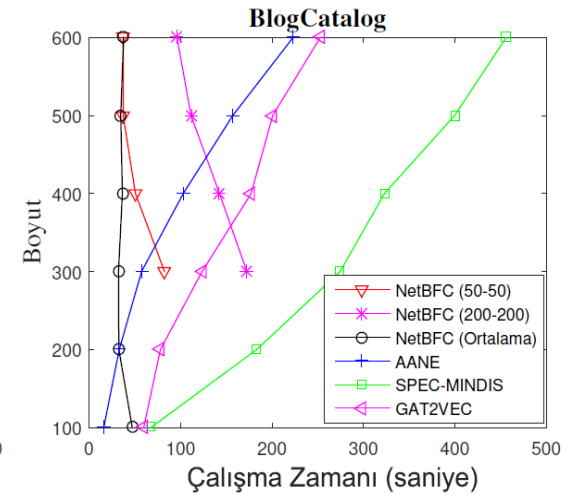
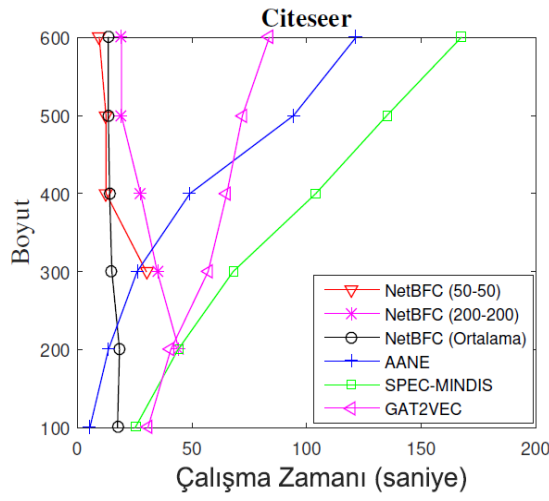
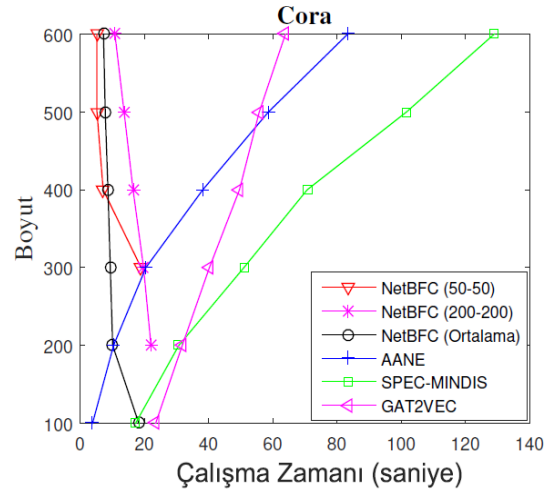
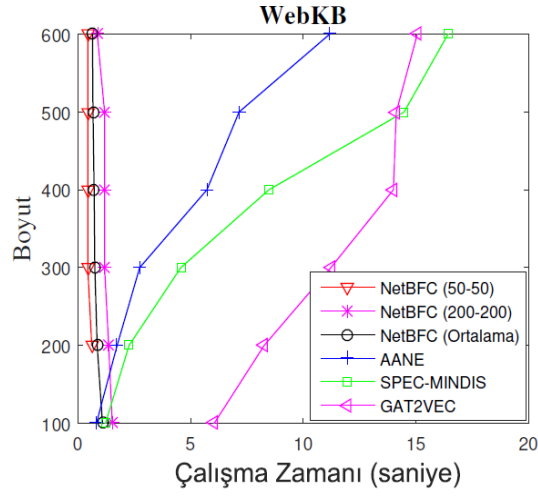
5.7. Çalışma Zamanı Analizi

NetBFC yönteminin çalışma süresini analiz etmek için Şekil 5.10 verilmektedir. Şekil 5.10'da farklı k ve z parametre değerlerine göre farklı verisetlerinde elde edilen çalışma süreleri karşılaştırılmaktadır. Yineleme sayısı her veriseti için 10 olmak üzere, her iterasyon sonunda elde edilen ulaşılan azaltılmış öznitelik boyutu ve her bir öznitelik boyutuna ne kadar sürede erişildiği gösterilmektedir. Eğer devam eden iterasyonda öznitelik uzayında yakınsama var ise iterasyon sonlandırıldığından bazı parametre ayarlamalarında 10'dan daha az iterasyon sonucu verilmektedir. NetBFC yöntemi öznitelik sayısı 1073 olan WebKB verisi için k ve z parametreleri 30'a eşit iken ilk iterasyonda % 83 boyut azaltımı sağlamış ve verinin boyutunu 177'e indirgemıştır. Bu iterasyon 0.357 saniyede gerçekleşmiştir. İlk iterasyonda en az boyut azaltımı % 56'lık boyut azaltımı ile parametre değerleri 200'e eşit iken gerçekleşmiştir. NetBFC yöntemi 3. iterasyonda şekilde verilen tüm parametre ayarlamaları ile WebKB öznitelik uzayını 200 boyutuna indirgeyebilmiştir. Parametre değerlerinin her ikisi de 50'ye ve her ikisi de 200'e eşit iken 10. iterasyonda öznitelik boyutu yaklaşık 130 olarak elde edilmiştir. Parametre değerlerinin her ikisi de 30'a ve her ikisi de 100'e eşit iken 10 iterasyon sonunda boyut 100 olarak elde edilmektedir. NetBFC, WebKB verisi için 100 boyutuna ortalama 1.6 saniye de erişmektedir. NetBFC yöntemi, Cora veriseti için ilk iterasyonda ortalama 2.2 saniyede tüm parametre ayarlamaları göz önüne alındığında yaklaşık % 23'lük boyut azaltımı sağlamaktadır. Cora verisetine ait öznitelik boyutu, 10 iterasyon sonunda tüm parametre ayarlamaları dikkate alındığında, yaklaşık ortalama 20 saniye de % 79 azaltılmıştır. Citeseer veriseti için parametre değerlerinin her ikisi de 200'e eşitken maksimum boyut azaltımı sağlamıştır. Citeseer verisetine ait öznitelik uzayının boyutu, 9. iterasyonun sonunda % 94.73 boyut azaltımı sağlamış ve bu iterasyon 43.53 saniye de gerçekleşmiştir. BlogCatalog veriseti için en yüksek boyut azaltımı parametre değerleri 50'ye eşit iken elde edilmiştir. Bu veriseti için 81.67 saniye çalışma süresi ile 10. iterasyon sonucunda % 96.28 boyut azaltımı elde edilmiştir. Sonuç olarak, NetBFC yönteminin çalışma süresi özellikle veriseti boyutuna bağlı olarak değişim göstermektedir. Farklı k ve z parametre değerleri yöntemin çalışma süresini etkilemekle beraber, veriseti boyutuna göre çalışma sürelerindeki değişim daha az fark ile gözlemlenmektedir.



Şekil 5.10: Farklı parametre ayarlamaları ile verisetleri için NetBFC yöntemine ait zaman analizi.

NetBFC yöntemi çalışma süresi açısından diğer yöntemler ile karşılaştırılmakta ve Şekil 5.11 de çalışma sürelerine ait sonuçlar verilmektedir. NetBFC yöntemi tüm verisetleri için k ve z parametrelerinin değerinin her ikisi 50'ye ve her ikisi 200'e eşitlenerek ayrı ayrı olarak çalıştırılmıştır. Ayrıca, k ve z parametrelerinden biri, sonrasında diğeri sabit tutulup [30, 50, 100, 150, 200, 250, 300, 350, 400] değerleri sırasıyla parametrelere atanmış ve yöntem tüm parametre ayarlamaları için çalıştırılmıştır. İstenilen boyutta elde edilen sonuçların ortalaması alınarak Şekil 5.11'de NetBFC (ortalama) olarak verilmiştir. NetBFC (ortalama) sonuçlarına göre, WebKB veriseti için istenilen boyut 300 ve 600 aralığında ise en az çalışma süresi NetBFC yöntemine aittir. İstenilen boyut 100 ve 200'e NetBFC yöntemi daha fazla iterasyon ile eriştiğinden dolayı tüm verisetleri göz önüne alındığında AANE yöntemi NetBFC yönteminden daha düşük çalışma süresine sahip olduğu gözlemlenmiştir. Ayrıca, NetBFC yöntemine ait k ve z parametreleri yüksek değerlere atandığında yöntemin çalışma süresinin arttığı gözlemlenmektedir. NetBFC ile diğer yöntemlerin çalışma süreleri karşılaştırıldığında, istenilen boyut 300 ve üzeri bir değer de ise neredeyse tüm yöntemlerden daha hızlı çalışmaktadır. Diğer yöntemler ile önerilen yöntemin arasındaki en belirgin fark ise NetBFC yönteminin istenilen boyut azaltımı daha düşük olduğunda çok az çalışma süresine sahip olmasıdır. Çünkü karşılaştırılan yöntemler, çoğunlukla özdeğer ve özvektör hesaplaması yaptıkları için, boyutsallık arttığında yüksek çalışma süresi gerektirmektedir. NetBFC yöntemi ise bir kaç iterasyon da istenilen daha düşük boyut azaltımını sağlamaktadır. Fakat, tüm verisetleri için öznitelik boyutsallığı 300 olarak temel alındığında ve k ve z parametre değerleri 50 olarak ayarlandığında SPECMINDIS ve GAT2VEC yöntemlerine göre önerilen yöntem daha az çalışma süresine sahiptir.



Şekil 5.11: Tüm verisetleri için yöntemlere ait zaman analizi.

6. SONUÇ VE GELECEK ÇALIŞMALAR

Bu çalışmada, bağlantı ve öznitelik bilgilerinden yararlanarak ağ bağlantılı veriler için anlamsal ve yapısal olarak etkili öznitelikler üreten denetimsiz bir ikili öznitelik oluşturma yöntemi önerilmiştir. Aynı grupta olma olasılığı yüksek olan ağ nesneleri arasındaki öznitelik yakınlıkları yeniden modellenerek, ağ bağlantılı veriler üzerinde gerçekleştirilen kümeleme görevlerinin başarısını artırmak amaçlanmıştır. Bu çalışmada, birinci dereceden ve ikinci dereceden öznitelik-bağlantı yakınlıkları temelinde, bir nesne kendisine en çok benzerlik gösteren bir düğüm kümesi ve bu nesne grubunda en sık gözlenen bir öznitelik kümesi ile ilişkilendirilmiştir. Böylelikle, nesnenin öznitelik uzayı, kendisine benzer olan nesnelerin öznitelik uzayına benzetilmektedir. Bu durumda, bir nesnenin öznitelikleri ve öznitelik sayısı (seyreklik fazla), bu nesnenin doğru kümelendirilmesi için yeterli olmadığında, önerilen algoritma NetBFC ile bu nesnenin doğru kümelenebilmesi için komşuluklarından faydalanmaktadır. Benzer şekilde, ağ bağlantılı veriye ait bağlantı yapısı da, hem öznitelik hem de bağlantı benzerliği gösteren düğüm kümelerinin arasındaki bağlantı yoğunluğunun artırılacak şekilde yeniden yapılandırılmaktadır. Belli bir yineleme sayısından sonra, öznitelik uzayının ve bağlantı yapısını temsil eden çizge yapısının yakınsadığı görülmektedir. Yöntem, hedef bir düğüm için seçilen düğüm kümesini ve bu düğüm kümesindeki baskın öznitelikleri belirlemek için parametreler kullanmaktadır. NetBFC yöntemi farklı parametre değerleri ile farklı öznitelik temsilleri elde edebilmektedir. Deneyler, dört gerçek dünya verisi üzerinde gerçekleştirilmiştir. Tüm veri kümeleri için, NetBFC yöntemi ile karşılaştırmak için kullanılan yöntemlerin çoğundan daha iyi kümeleme performansı elde edilmiştir. NetBFC, öznitelik oluşturma sürecinde yerel bir öznitelik seçme süreci ve öznitelik eliminasyonu gerçekleştirdiğinden önemli miktarda boyut azaltımı sağlayarak, boyutsallık laneti problemine çözüm üretmektedir. NetBFC belli bir yineleme sonrasında yakınsama gösterdiğinde veya istenilen boyut azaltma miktarına erişildiğinde sonlanmaktadır. Öznitelik uzayının boyutsallığında yüksek miktarda azaltma sağlandığında bile, yöntem yüksek doğruluk ve yüksek NMI değerleri sağlama eğilimindedir.

Gelecekteki çalışmalar için dört konuya odaklanılabilir. Öncelikle, yöntemin performansını artırmak için farklı öznitelik oluşturma operatörleri kullanılabilir. Var

olan gözetimsiz bir şekilde öznitelik seçimi veya öznitelik çıkarımı yapan yöntemlere ait amaç fonksiyonu, yöntem için de kullanılan yerel öznitelik üretimi ve seçimi yaklaşımı ile birlikte tekrar yorumlanabilir. İkinci olarak, NetBFC farklı parametre değerleri ile çalıştırıldığında, bir nesne için çok sayıda farklı öznitelik temsilleri elde edilmektedir. Bu temsiller kullanılarak bir derin öğrenme algoritmasının performansının nasıl değiştiği gözlemlenebilir. Üçüncü olarak, parametresiz bir sistem geliştirmek için parametrelerin değerinin gözetimsiz bir şekilde ağ verilerinin yapısından nasıl tahmin edilebileceği bir araştırma konusu olabilir. Son olarak, NetBFC yöntemi dinamik ağlara adapte olacak şekilde güncellenebilir. Yöntem, yerel bir öznitelik oluşturma süreci gerçekleştirdiğinden, dinamik ağlara uygulanabilir olduğu düşünülmektedir. Bir güncelleme ölçütü ile hızlıca ağa dahil olan bağlantılar ve öznitelikler için ağ ve öznitelik temsili güncellenebilir.

KAYNAKLAR

- Abualigah L. M., Khader A. T., Al-Betar M. A., , Alomari O. A., (2017), “Text feature selection with a robust weight scheme and dynamic dimension reduction to text document clustering”, *Expert Systems With Applications*, 84, 24–36.
- Akal Ş., (2014), “Gerçek ve model ağların karakteristik özelliklerinin karşılaştırılması”, *Journal of Marmara University Social Sciences Institute/Öneri*, 11 (41).
- Ankerst M., Breunig M. M., Kriegel H.-P., , Sander J., (1999), “OPTICS: ordering points to identify the clustering structure”, *ACM Sigmod Record*, Volume 28, 49–60.
- Backstrom L., Huttenlocher D., Kleinberg J., , Lan X., (2006), “Group formation in large social networks: membership, growth, and evolution”, *Proceedings of the 12th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, 44–54.
- Bagallo G., Haussler D., (1990), “Boolean feature discovery in empirical learning”, *Machine Learning*, 5 (1), 71–99.
- Barabási A.-L., Albert R., , Jeong H., (1999), “Mean-field theory for scale-free random networks”, *Physica A: Statistical Mechanics And Its Applications*, 272 (1-2), 173–187.
- Bellman R., (1957), “Dynamic Programming”, 2010 Edition, Versity Press, Princeton, NJ: Princeton University.
- Bennasar M., Hicks Y., , Setchi R., (2015), “Feature selection using joint mutual information maximisation”, *Expert Systems With Applications*, 42 (22), 8520–8532.
- Berman K. K., Iyer R. S., Richardson D. J., Rahurkar A. S., , Seetharaman A., (2012), “Recommendation engine for social networks”, *US Patent* 8,095,432.
- Berry M. W., Browne M., Langville A. N., Pauca V. P., , Plemmons R. J., (2007), “Algorithms and applications for approximate nonnegative matrix factorization”, *Computational Statistics & Data Analysis*, 52 (1), 155–173.
- Berry M. W., Drmac Z., , Jessup E. R., (1999), “Matrices, vector spaces, and information retrieval”, *SIAM Review*, 41 (2), 335–362.
- Bezdek J. C., Ehrlich R., , Full W., (1984), “FCM: The fuzzy c-means clustering algorithm”, *Computers & Geosciences*, 10 (2-3), 191–203.
- Bharti K. K., Singh P. K., (2015), “Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering”, *Expert Systems With Applications*, 42 (6), 3105–3114.
- Bhattacharya P., Iliofotou M., Neamtii I., , Faloutsos M., (2012), “Graph-based analysis and prediction for software evolution”, *2012 34th International Conference On Software Engineering (ICSE)*, 419–429.

- Biesiada J., Duch W., (2007), “Feature selection for high-dimensional data—a Pearson redundancy based filter”, *Computer Recognition Systems* 2, 242–249. Springer.
- Bird C., Gourley A., Devanbu P., Gertz M., , Swaminathan A., (2006), “Mining email social networks”, *Proceedings of the 2006 International Workshop On Mining Software Repositories*, 137–143.
- Bonchi F., Castillo C., Gionis A., , Jaimes A., (2011), “Social network analysis and mining for business applications”, *ACM Transactions On Intelligent Systems And Technology (TIST)*, 2 (3), 22.
- Broder A., Kumar R., Maghoul F., Raghavan P., Rajagopalan S., Stata R., Tomkins A., , Wiener J., (2000), “Graph structure in the web”, *Computer Networks*, 33 (1-6), 309–320.
- Cai D., Zhang C., , He X., (2010), “Unsupervised feature selection for multi-cluster data”, *Proceedings of the 16th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, 333–342.
- Carrer-Neto W., Hernández-Alcaraz M. L., Valencia-García R., , García-Sánchez F., (2012), “Social knowledge-based recommender system. Application to the movies domain”, *Expert Systems With Applications*, 39 (12), 10990–11000.
- Cataltepe Z., Sonmez A., , Senliol B., (2014), “Feature enrichment and selection for transductive classification on networked data”, *Pattern Recognition Letters*, 37, 41–53.
- Chang S., Han W., Tang J., Qi G.-J., Aggarwal C. C., , Huang T. S., (2015), “Heterogeneous network embedding via deep architectures”, *Proceedings of the 21th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, 119–128.
- Chung F. R., Graham F. C., (1997), “Spectral graph theory”, number 92, American Mathematical Soc.
- Dave R. N., Bhaswan K., (1992), “Adaptive fuzzy c-shells clustering and detection of ellipses”, *IEEE Transactions On Neural Networks*, 3 (5), 643–662.
- De Sa V. R., (2005), “Spectral clustering with two views”, *ICML Workshop On Learning With Multiple Views*, 20–27.
- Debnath S., Ganguly N., , Mitra P., (2008), “Feature weighting in content based recommendation system using social network analysis”, *Proceedings of the 17th International Conference On World Wide Web*, 1041–1042.
- Demmel J. W., (1997), “Applied numerical linear algebra”, Volume 56, SIAM.
- Dhillon I. S., Modha D. S., (2001), “Concept decompositions for large sparse text data using clustering”, *Machine Learning*, 42 (1-2), 143–175.
- Diesner J., Frantz T. L., , Carley K. M., (2005), “Communication networks from the Enron email corpus “It’s always about the people. Enron is no different””, *Computational & Mathematical Organization Theory*, 11 (3), 201–228.

- Ding Y., (2011), “Scientific collaboration and endorsement: Network analysis of coauthorship and citation networks”, *Journal of Informetrics*, 5 (1), 187–203.
- Domingos P., (2005), “Mining social networks for viral marketing”, *IEEE Intelligent Systems*, 20 (1), 80–82.
- Duda R. O., Hart P. E., , Stork D. G., (2012), “Pattern classification”, 2nd Edition, John Wiley & Sons.
- Dy J. G., (2008), “Unsupervised feature selection”, *Computational Methods of Feature Selection*, 19–39.
- Dy J. G., Brodley C. E., (2004), “Feature selection for unsupervised learning”, *Journal of Machine Learning Research*, 5 (Aug), 845–889.
- Erdős P., Rényi A., (1960), “On the evolution of random graphs”, *Publ. Math. Inst. Hung. Acad. Sci*, 5 (1), 17–60.
- Ester M., Kriegel H.-P., Sander J., Xu X., , others, (1996), “A density-based algorithm for discovering clusters in large spatial databases with noise.”, *Kdd*, Volume 96, 226–231.
- Euler L., (1736), “*Mechanica sive motus scientia analytice exposita... instar supplementi ad Commentar. Acad. scient. imper*”, ex typographia Academiae scientarum.
- Fiore A. T., Donath J. S., (2005), “Homophily in online dating: when do you like someone like yourself?”, *CHI’05 Extended Abstracts On Human Factors In Computing Systems*, 1371–1374.
- Fodor I. K., (2002), “A survey of dimension reduction techniques”, Technical report, Lawrence Livermore National Lab., CA (US).
- Forgy E. W., (1965), “Cluster analysis of multivariate data: efficiency versus interpretability of classifications”, *Biometrics*, 21, 768–769.
- Fowler J. H., Settle J. E., , Christakis N. A., (2011), “Correlated genotypes in friendship networks”, *Proceedings of the National Academy of Sciences*, 108 (5), 1993–1997.
- Frank A., (2005), “On Kuhn’s Hungarian method—a tribute from Hungary”, *Naval Research Logistics (NRL)*, 52 (1), 2–5.
- Friedkin N. E., (2006), “A structural theory of social influence”, Volume 13, Cambridge University Press.
- Goldberg A. V., Tarjan R. E., (1988), “A new approach to the maximum-flow problem”, *Journal of the ACM (JACM)*, 35 (4), 921–940.
- Gu Q., Han J., (2011), “Towards feature selection in network”, *Proceedings of the 20th ACM International Conference On Information And Knowledge Management*, 1175–1184.
- Guha S., Rastogi R., , Shim K., (1998), “CURE: an efficient clustering algorithm for large databases”, *ACM Sigmod Record*, Volume 27, 73–84.

- Guha S., Rastogi R., , Shim K., (2000), “ROCK: A robust clustering algorithm for categorical attributes”, *Information Systems*, 25 (5), 345–366.
- Gundecha P., Liu H., (2012), “Mining social media: a brief introduction”, *Tutorials In Operations Research*, 1 (4), 1–17.
- Guyon I., Elisseeff A., (2003), “An introduction to variable and feature selection”, *Journal of Machine Learning Research*, 3 (Mar), 1157–1182.
- Hand D. J., (2010), “Statistical Analysis of Network Data: Methods and Models by Eric D. Kolaczyk”, *International Statistical Review*, 78 (1), 135–135.
- He R., Tan T., Wang L., , Zheng W.-S., (2012), “ l_2, l_1 regularized correntropy for robust feature selection”, *Computer Vision And Pattern Recognition (CVPR), 2012 IEEE Conference On*, 2504–2511.
- He X., Cai D., , Niyogi P., (2006), “Laplacian score for feature selection”, *Advances In Neural Information Processing Systems*, 507–514.
- He Z., Xie S., Zdunek R., Zhou G., , Cichocki A., (2011), “Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering”, *IEEE Transactions On Neural Networks*, 22 (12), 2117–2131.
- Hoseini E., Mansoori E. G., (2016), “Selecting discriminative features in social media data: An unsupervised approach”, *Neurocomputing*, 205, 463–471.
- Hu L., Gao W., Zhao K., Zhang P., , Wang F., (2018), “Feature selection considering two types of feature relevancy and feature interdependency”, *Expert Systems With Applications*, 93, 423–434.
- Hu X., Tang J., Gao H., , Liu H., (2013), “Unsupervised sentiment analysis with emotional signals”, *Proceedings of the 22nd International Conference On World Wide Web*, 607–618.
- Huang X., Li J., , Hu X., (2017a), “Accelerated attributed network embedding”, *Proceedings of the 2017 SIAM International Conference On Data Mining*, 633–641.
- Huang X., Li J., , Hu X., (2017b), “Label informed attributed network embedding”, *Proceedings of the Tenth ACM International Conference On Web Search And Data Mining*, 731–739.
- Hughes G., (1968), “On the mean accuracy of statistical pattern recognizers”, *IEEE Transactions On Information Theory*, 14 (1), 55–63.
- Jain A. K., (2010), “Data clustering: 50 years beyond K-means”, *Pattern Recognition Letters*, 31 (8), 651–666.
- Jain A. K., Dubes R. C., (1988), “Algorithms for clustering data”, Volume 6, Prentice hall Englewood Cliffs.
- Kakisim A. G., Sogukpinar I., (2019), “Unsupervised binary feature construction method for networked data”, *Expert Systems With Applications*, 121, 256–265.

- Kandel D. B., (1978), "Homophily, selection, and socialization in adolescent friendships", *American Journal of Sociology*, 84 (2), 427–436.
- Kaufman L., Rousseeuw P. J., (2009), "Finding groups in data: an introduction to cluster analysis", Volume 344, John Wiley & Sons.
- Kim Y., Street W. N., , Menczer F., (2002), "Evolutionary model selection in unsupervised learning", *Intelligent Data Analysis*, 6 (6), 531–556.
- Kohavi R., John G. H., (1997), "Wrappers for feature subset selection", *Artificial Intelligence*, 97 (1-2), 273–324.
- Kossinets G., Watts D. J., (2009), "Origins of homophily in an evolving social network", *American Journal of Sociology*, 115 (2), 405–450.
- Kramer R., (1998), "Chemometric techniques for quantitative analysis", 1st Edition, CRC Press.
- Krawiec K., (2002), "Genetic programming-based construction of features for machine learning and knowledge discovery tasks", *Genetic Programming And Evolvable Machines*, 3 (4), 329–343.
- Krishnamurthy B., Wang J., (2000), "On network-aware clustering of web clients", *ACM SIGCOMM Computer Communication Review*, 30 (4), 97–110.
- Kumar A., Daumé H., (2011), "A co-training approach for multi-view spectral clustering", *Proceedings of the 28th International Conference On Machine Learning (ICML-11)*, 393–400.
- Langley P., (1994), "Selection of relevant features in machine learning", *Proceedings of the AAAI Fall Symposium On Relevance*, 1–5.
- Le T. M., Lauw H. W., (2014), "Probabilistic latent document network embedding", *Data Mining (ICDM), 2014 IEEE International Conference On*, 270–279.
- Lee D. D., Seung H. S., (2001), "Algorithms for non-negative matrix factorization", *Advances In Neural Information Processing Systems*, 556–562.
- Leighton F. T., (1979), "A graph coloring algorithm for large scheduling problems", *Journal of Research of the National Bureau of Standards*, 84 (6), 489–506.
- Lensen A., Xue B., , Zhang M., (2017), "Improving k-means clustering with genetic programming for feature construction", *Proceedings of the Genetic And Evolutionary Computation Conference Companion*, 237–238.
- Li J., Hu X., Wu L., , Liu H., (2016), "Robust unsupervised feature selection on networked data", *Proceedings of the 2016 SIAM International Conference On Data Mining*, 387–395.
- Li J., Liu H., (2017), "Challenges of feature selection for big data analytics", *IEEE Intelligent Systems*, 32 (2), 9–15.
- Li Z., Yang Y., Liu J., Zhou X., , Lu H., (2012), "Unsupervised feature selection using nonnegative spectral analysis", *Twenty-Sixth AAAI Conference On Artificial*

Intelligence.

Liao L., He X., Zhang H., , Chua T.-S., (2017), “Attributed social network embedding”, ArXiv Preprint ArXiv:1705.04969.

Lin C.-J., (2007), “Projected gradient methods for nonnegative matrix factorization”, *Neural Computation*, 19 (10), 2756–2779.

Liu J., Ji S., , Ye J., (2009), “Multi-task feature learning via efficient l_2 , l_1 -norm minimization”, *Proceedings of the Twenty-Fifth Conference On Uncertainty In Artificial Intelligence*, 339–348.

Lloyd S., (1982), “Least squares quantization in PCM”, *IEEE Transactions On Information Theory*, 28 (2), 129–137.

Logendran R., Sriskandarajah C., (1996), “Sequencing of robot activities and parts in two-machine robotic cells”, *International Journal of Production Research*, 34 (12), 3447–3463.

Lu Q., Getoor L., (2003), “Link-based classification”, *Proceedings of the 20th International Conference On Machine Learning (ICML-03)*, 496–503.

MacQueen J., (1967), “Some methods for classification and analysis of multivariate observations”, *Proceedings of the Fifth Berkeley Symposium On Mathematical Statistics And Probability*, Volume 1, 281–297.

Markovitch S., Rosenstein D., (2002), “Feature generation using general constructor functions”, *Machine Learning*, 49 (1), 59–98.

Marlow C., Naaman M., Boyd D., , Davis M., (2006), “HT06, tagging paper, taxonomy, Flickr, academic article, to read”, *Proceedings of the Seventeenth Conference On Hypertext And Hypermedia*, 31–40.

Marsden P. V., Friedkin N. E., (1993), “Network studies of social influence”, *Sociological Methods & Research*, 22 (1), 127–151.

Martínez V., Berzal F., , Cubero J.-C., (2017), “A survey of link prediction in complex networks”, *ACM Computing Surveys (CSUR)*, 49 (4), 69.

Matheus C. J., Rendell L. A., (1989), “Constructive Induction On Decision Trees.”, *IJCAI*, Volume 89, 645–650.

McCallum A., (2017), “Cora dataset”, *Inter-University Consortium For Political And Social Research (ICPSR)*.

McPherson M., Smith-Lovin L., , Cook J. M., (2001), “Birds of a feather: Homophily in social networks”, *Annual Review of Sociology*, 27 (1), 415–444.

Mikolov T., Chen K., Corrado G., , Dean J., (2013), “Efficient estimation of word representations in vector space”, ArXiv Preprint ArXiv:1301.3781.

Mitra P., Murthy C., , Pal S. K., (2002), “Unsupervised feature selection using feature similarity”, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 24 (3), 301–312.

- Mostafa M. M., (2013), “More than words: Social networks’ text mining for consumer brand sentiments”, *Expert Systems With Applications*, 40 (10), 4241–4251.
- Motoda H., Liu H., (2002), “Feature selection, extraction and construction”, *Communication of IICM (Institute of Information And Computing Machinery, Taiwan)* Vol, 5 (67-72), 2.
- Newman M. E., (2004), “Fast algorithm for detecting community structure in networks”, *Physical Review E*, 69 (6), 066133.
- Newman M. E., (2006), “Modularity and community structure in networks”, *Proceedings of the National Academy of Sciences*, 103 (23), 8577–8582.
- Nicholas F. C., Carswell I. B., (2012), “Method and system for providing network based target advertising”, *US Patent* 8,131,585.
- Nie F., Huang H., Cai X., , Ding C. H., (2010), “Efficient and robust feature selection via joint l_2, l_1 -norms minimization”, *Advances In Neural Information Processing Systems*, 1813–1821.
- Otero F. E., Silva M. M., Freitas A. A., , Nievola J. C., (2003), “Genetic programming for attribute construction in data mining”, *European Conference On Genetic Programming*, 384–393.
- Paatero P., Tapper U., (1994), “Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values”, *Environmetrics*, 5 (2), 111–126.
- Park H.-S., Jun C.-H., (2009), “A simple and fast algorithm for K-medoids clustering”, *Expert Systems With Applications*, 36 (2), 3336–3341.
- Perlich C., Provost F., (2006), “Distribution-based aggregation for relational learning with identifier attributes”, *Machine Learning*, 62 (1-2), 65–105.
- Perozzi B., Al-Rfou R., , Skiena S., (2014), “Deepwalk: Online learning of social representations”, *Proceedings of the 20th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, 701–710.
- Popescul A., Ungar L. H., (2004), “Cluster-based concept invention for statistical relational learning”, *Proceedings of the Tenth ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, 665–670.
- Qi G.-J., Aggarwal C., Tian Q., Ji H., , Huang T., (2012), “Exploring context and content links in social media: A latent space method”, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 34 (5), 850–862.
- Rezvanian A., Meybodi M. R., (2015), “Sampling social networks using shortest paths”, *Physica A: Statistical Mechanics And Its Applications*, 424, 254–268.
- Sen P., Namata G., Bilgic M., Getoor L., Galligher B., , Eliassi-Rad T., (2008), “Collective classification in network data”, *AI Magazine*, 29 (3), 93.
- Shakibian H., Charkari N. M., (2016), “Optimization problems in complex net-

works: Challenges and directions”, 2016 24th Iranian Conference On Electrical Engineering (ICEE), 682–686.

Sharan R., Shamir R., (2000), “CLICK: a clustering algorithm with applications to gene expression analysis”, Proc Int Conf Intell Syst Mol Biol, Volume 8, 16.

Shaw P. J., (2009), “Multivariate statistics for the environmental sciences”, 1st Edition, Wiley.

Sheikh N., Kefato Z., , Montresor A., (2019), “gat2vec: representation learning for attributed graphs”, Computing, 101 (3), 187–209.

Shi C., Li Y., Zhang J., Sun Y., , Philip S. Y., (2016), “A survey of heterogeneous information network analysis”, IEEE Transactions On Knowledge And Data Engineering, 29 (1), 17–37.

Sinka M. P., Corne D. W., (2002), “A large benchmark dataset for web document clustering”, Soft Computing Systems: Design, Management And Applications, 87, 881–890.

Slattery S., Craven M., (1998), “Combining statistical and relational methods for learning in hypertext domains”, International Conference On Inductive Logic Programming, 38–52.

Sondhi P., (2009), “Feature construction methods: a survey”, Sifaka. Cs. Uiuc. Edu, 69, 70–71.

Stelzl U., Worm U., Lalowski M., Haenig C., Brembeck F. H., Goehler H., Stroedicke M., Zenkner M., Schoenherr A., Koeppen S., , others, (2005), “A human protein-protein interaction network: a resource for annotating the proteome”, Cell, 122 (6), 957–968.

Strehl A., Ghosh J., (2002), “Cluster ensembles—a knowledge reuse framework for combining multiple partitions”, Journal of Machine Learning Research, 3 (Dec), 583–617.

Tang J., Liu H., (2012), “Feature selection with linked data in social media”, Proceedings of the 2012 SIAM International Conference On Data Mining, 118–128.

Tang J., Liu H., (2013), “Coselect: Feature selection with instance selection for social media data”, Proceedings of the 2013 SIAM International Conference On Data Mining, 695–703.

Tang J., Liu H., (2014), “An unsupervised feature selection framework for social media data”, IEEE Transactions On Knowledge And Data Engineering, 26 (12), 2914–2927.

Tang J., Sun J., Wang C., , Yang Z., (2009a), “Social influence analysis in large-scale networks”, Proceedings of the 15th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining, 807–816.

Tang L., Liu H., (2010), “Community detection and mining in social media”, Synthesis Lectures On Data Mining And Knowledge Discovery, 2 (1), 1–137.

- Tang W., Lu Z., , Dhillon I. S., (2009b), “Clustering with multiple graphs”, Data Mining, 2009. ICDM’09. Ninth IEEE International Conference On, 1016–1021.
- Thelwall M., (2010), “Emotion homophily in social network site messages”, First Monday, 15 (4).
- Tommasel A., Godoy D., (2018), “Short-text feature construction and selection in social media data: a survey”, Artificial Intelligence Review, 49 (3), 301–338.
- Tran B., Xue B., , Zhang M., (2017), “Using feature clustering for GP-based feature construction on high-dimensional data”, European Conference On Genetic Programming, 210–226.
- Tufekci Z., (2014), “Big questions for social media big data: Representativeness, validity and other methodological pitfalls”, Eighth International AAAI Conference On Weblogs And Social Media.
- Wang X. F., Chen G., (2003), “Complex networks: small-world, scale-free and beyond”, IEEE Circuits And Systems Magazine, 3 (1), 6–20.
- Wang Y., Wang S., Tang J., Liu H., , Li B., (2015), “Unsupervised Sentiment Analysis for Social Media Images.”, IJCAI, 2378–2379.
- Watts D. J., Strogatz S. H., (2018), “Collective dynamics of “small-world” networks. nature, 393 (6684): 440–442, 1998”, A First Comparison of Abstract Argumentation Systems, 245.
- Wei X., Xie S., , Yu P. S., (2015), “Efficient partial order preserving unsupervised feature selection on networks”, Proceedings of the 2015 SIAM International Conference On Data Mining, 82–90.
- Xu D., Tian Y., (2015), “A comprehensive survey of clustering algorithms”, Annals of Data Science, 2 (2), 165–193.
- Xu R., Wunsch D. C., (2005), “Survey of clustering algorithms”, IEEE Transactions On Neural Networks, 16 (3), 645–678.
- Yager R. R., Filev D. P., (1994), “Approximate clustering via the mountain method”, IEEE Transactions On Systems, Man, And Cybernetics, 24 (8), 1279–1284.
- Yang S., Hou C., Nie F., , Wu Y., (2012), “Unsupervised maximum margin feature selection via L 2, 1-norm minimization”, Neural Computing And Applications, 21 (7), 1791–1799.
- Yang Y., Pedersen J. O., (1997), “A comparative study on feature selection in text categorization”, ICML, Volume 97, 35.
- Yang Y., Shen H. T., Ma Z., Huang Z., , Zhou X., (2011), “L2, 1-Norm Regularized Discriminative Feature Selection for Unsupervised Learning”, Twenty-Second International Joint Conference On Artificial Intelligence.
- Zahn C. T., (1970), “Graph theoretical methods for detecting and describing gestalt

clusters”, IEEE Trans. Comput., 20 (SLAC-PUB-0672-REV), 68.

Zhang T., Ramakrishnan R., , Livny M., (1996), “BIRCH: an efficient data clustering method for very large databases”, ACM Sigmod Record, Volume 25, 103–114.

Zhao B., Kwok J., Wang F., , Zhang C., (2009), “Unsupervised maximum margin feature selection with manifold regularization”, 2009 IEEE Conference On Computer Vision And Pattern Recognition, 888–895.

Zhao J., Lu K., , He X., (2008), “Locality sensitive semi-supervised feature selection”, Neurocomputing, 71 (10-12), 1842–1849.

Zhao Z., Liu H., (2007), “Spectral feature selection for supervised and unsupervised learning”, Proceedings of the 24th International Conference On Machine Learning, 1151–1157.

Zhu S., Yu K., Chi Y., , Gong Y., (2007a), “Combining content and link for classification using matrix factorization”, Proceedings of the 30th Annual International ACM SIGIR Conference On Research And Development In Information Retrieval, 487–494.

Zhu Z., Ong Y.-S., , Dash M., (2007b), “Wrapper–filter feature selection algorithm using a memetic framework”, IEEE Transactions On Systems, Man, And Cybernetics, Part B (Cybernetics), 37 (1), 70–76.

ÖZGEÇMİŞ

Arzu Kakışım 2009 yılında Karadeniz Teknik Üniversitesi Bilgisayar Mühendisliği Bölümünden mezun oldu. 2010 yılında başladığı Gebze Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü yüksek lisans programını 2013 yılında başarıyla tamamlayarak aynı yıl doktora eğitimine Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında başladı. 2011-2019 yılları arasında Gebze Teknik Üniversitesi Bilgisayar Mühendisliği Bölümünde araştırma görevlisi olarak çalıştı. Başlıca araştırma alanları arasında ağ analizi, topluluk tespiti, graf kümelemesi, olasılık modelleri, metin madenciliği, graf madenciliği ve kötü amaçlı yazılım analizi yer almaktadır.

EKLER

Ek A: Tez Çalışması Kapsamında Yapılan Yayınlar

Kakisim A. G., Sogukpınar I. (2019), "Unsupervised binary feature construction method for networked data", Expert Systems with Applications, 121, 256-265.