

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK ve FEN BİLİMLERİ ENSTİTÜSÜ

SOSYAL AĞLARDA ETKİLEŞİM ÖRÜNTÜLERİ
KULLANILARAK BAĞLANTI TAHMİNİ

BURAK IŞIKLI
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE
2014

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK ve FEN BİLİMLERİ ENSTİTÜSÜ

SOSYAL AĞLARDA ETKİLEŞİM
ÖRÜNTÜLERİ KULLANILARAK
BAĞLANTI TAHMİNİ

BURAK IŞIKLI
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
DOÇ. DR. F. ERDOĞAN SEVİLGİN

GEBZE

2014



**GEBZE YÜKSEK
TEKNOLOJİ ENSTİTÜSÜ**

YÜKSEK LİSANS JÜRİ ONAY FORMU

GYTE Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 27.01.2014 tarih ve 2014/06 sayılı kararıyla oluşturulan jüri tarafından 18/02/2014 tarihinde tez savunma sınavı yapılan Burak IŞIKLI'nın tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Doç. Dr. Fatih Erdoğan SEVİLGİN

ÜYE

: Yrd. Doç. Dr. Tunahan ÇAKIR

ÜYE

: Doç. Dr. Mehmet GÖKTÜRK

ONAY

GYTE Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

ÖZET

Son yıllarda sosyal medyanın kullanımında hızlı artışı ile beraber sosyal medya analizi konusunda yapılan çalışmalar da önem kazanmıştır. Yaşanan birçok toplumsal ve siyasi olayın sosyal medyada yankı bulması ve artan kullanım sayısı ile reklam sektörünün sosyal medyaya yönelik yeni reklam stratejileri geliştirmesi sosyal medyanın önemini göstermektedir. Sosyal medyayı analiz yöntemlerinden biri sosyal ağ analizi ve bu kapsamda öne çıkan çalışma konularından biri bağlantı tahminidir.

Bu çalışmada *Sosyal Ağlarda Etkileşim Örüntüleri Kullanılarak Bağlantı Tahmini* yapılmıştır. Ağ tahmin algoritmaları, pratikte büyük ağ yapıları için iyi bir uygulama alanı değildir (yaklaşım yöntemleri kullanılmadan). Bunun nedeni bu yöntemlerde ya işlem süresi uzun sürmektedir ya da güçlü, pahalı süper bilgisayarlar gerekmektedir. Bu sebeple bu çalışmada klasik ağ tahmin algoritmaları dışında bir yöntem önerilmiştir. Bu yöntemde ağ yapısı düzleştirilerek özellik vektörleri şekline getirilmektedir. Ağdaki tekrarlanan örüntü vektörleri bulunarak eksik veya bilinmeyen bağlantıları tahmin etmek için klasik makine öğrenmesi algoritmalarından faydalanılmıştır. Önerilen yöntem, Twitter ağına uygulanarak yeni takip edilebilecek kullanıcılar tavsiye edilmeye çalışılmıştır.

Bu çalışmanın literatüre ana katkısı, bağlantı tahmini probleminin etiket tahmini problemine dönüştürülmesi, büyük boyuttaki ağlar için bağlantı tahmininde kullanılacak örüntüleri çıkartmada yeni bir metot kullanılması ve her bir çıkartılan örüntünün sağladığı yararın gösterilmesidir.

Anahtar Kelimeler: Sosyal ağlar, veri madenciliği, büyük veri, ağ madenciliği, ağ işleme, bağlantı tahmini.

SUMMARY

The usage of social media has been rapidly increased in recent years. Studies on social media analysis become more important due to increasing usage of social media, as well. Spreading many social and political events on social media and developing new advertising strategies for social media by advertising companies indicate the importance of social media. One approach for analyzing social media is network analysis and a popular area about network analysis is link prediction.

In this work, *Link Prediction using Annotation Patterns on Social Networks* is examined. Graph theoretical link and annotation prediction algorithms are usually not applicable (except using approximation approach) for very big network structures since processing takes a long time or requires powerful, expensive supercomputer. Thus a new method without using graph theoretical link or annotation prediction algorithms is proposed in this study. The network structure is transformed into feature vectors by flattening. Repeated sub-network patterns are found for each nodes of the network and they fed into machine learning algorithms to predict missing or unknown annotations and links. This method is applied to Twitter network for suggesting new followers.

The major contributions of this study are transformation of link prediction into annotation prediction problem, a new method for extracting features from large-scale networks, and evaluation of benefits provided by each extracted pattern.

Key Words: social networks, data mining and knowledge discovery, big data, business intelligence, link prediction, graph processing, graph mining.

TEŞEKKÜR

Başta, yüksek lisans eğitimimde ve akademik hayatımda desteğini ve yardımlarını hiçbir zaman esirgemeyip bilgisi ile bu çalışmanın oluşmasının yolunu açan danışmanım Doç. Dr. Erdoğan SEVİLGEN'e ve Zeki ERDEM'e,

Bütün çalışmam boyunca yanımda olan, değerli aileme, bilgi ve tecrübelerini benimle paylaşan değerli arkadaşım Mustafa KIRAÇ'a, Turkcell'deki çalışma arkadaşlarıma ve değerli yöneticilerime en içten teşekkürlerimi sunarım.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	iv
SUMMARY	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
SİMGELER ve KISALTMALAR DİZİNİ	xi
ŞEKİLLER DİZİNİ	x
TABLOLAR DİZİNİ	xi
1. GİRİŞ	1
2. İLGİLİ ÇALIŞMALAR	6
3. METODOLOJİ	9
3.1. Sosyal Ağ Analizi	9
3.2. Twitter	9
3.3. Ağ Madenciliği	10
3.4. Kullanılan Makine Öğrenmesi Algoritmaları	11
3.4.1. Genelleştirilmiş Lineer Model (GLM)	11
3.4.2. Destek Vektör Makineleri (SVM)	13
3.4.3. Naive Bayes (NB)	14
3.4.4. Karar Ağaçları (DT)	15
3.5. Nitelik Önemi (Attribute Importance) Algoritmaları	16
3.5.1. Minimum Tanım Uzunluğu (MDL)	16
3.6. Metotlarla İlgili Kullanılan Problem ve Tanımları	17
4. YÖNTEMLER VE ALGORİTMALAR	21
5. DENEYSEL SONUÇLAR	26
5.1. Makine Öğrenmesi Algoritmalarının Seçilmesi	27
5.2. Örüntü Türüne Göre Tahmin Karşılaştırması	32
5.3. Örüntü Karmaşıklığını Arttırmanın Tahmine Etkisi	33
5.4. Dolaylı Komşuluktan Gelen Tahmin Bilgisi	34
5.5. Örüntü Türünün Birbiri Üzerindeki Etkisi	35

5.6. Ünlü Düğümlerinin Kendi Örüntüleri Üzerindeki Nitelik Öneminin Tespit Edilmesi	36
5.7. Genelleştirilmiş Lineer Model Kullanılarak Deneylerin Tekrarlanması	37
5.8. Benzer Çalışmalar ile Başarı Karşılaştırması	42
6. SONUÇLAR	43
7. GELECEK ÇALIŞMALAR	45
KAYNAKLAR	46
ÖZGEÇMİŞ	49
EKLER	50

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler ve Açıklamalar Kisaltmalar

API	: Uygulama Programlama Arabirimi (Application programming interface)
Biobase	: Biyolojik bibliyografik Veritabanı (Biological Bibliographic Database)
CDF	: Kümülatif Dağılım Fonksiyonu (Cumulative distribution function)
DBLP	: Veri tabanı Sistemleri ve Mantıksal Programlama (Database systems and Logic Programming)
DT	: Karar Ağaçları (Decision Tree)
ETL	: Önişlem (Al-Dönüştür-Yükle veya Extract-Transform-Loading)
GLM	: Genelleştirilmiş Lineer Model (Generalized Linear Model)
GYTE	: Gebze Yüksek Teknoloji Enstitüsü
Maks.	: Maksimum
MDL	: Minimum Tanım Uzunluğu (Minimum description length)
Min.	: Minimum
NB	: Naive Bayes
ODM	: Oracle Veri Madencisi (Oracle Data Miner)
Ort.	: Ortalama
PLSQL	: Usule Ait Yapılandırılmış Sorgu Dili (Procedural Language SQL)
ROC	: Alıcı İşletim Karakteristiği (Receiver operating characteristic)
SQL	: Yapılandırılmış Sorgu Dili (Structured Query Language)
SVM	: Destek Vektör Makineleri (Support Vector Machines)
WWW	: İnternet

ŞEKİLLER DİZİNİ

<u>Sekil No:</u>	<u>Sayfa</u>
3.1: Örnek bir takip ağı.	10
3.2: Örnek bir ilgi alanı ağı yapısı.	11
3.3: Lineer regresyon örneği.	12
3.4: Destek vektör makineleri örneği.	13
3.5: Naive bayes örneği.	14
3.7: Bir Karar Ağacının (Yatay) belli bir parçası gösterilmektedir.	16
4.1: Eksik olan etiketleri tahmin etmek için kullanılan yaklaşımın metodolojisi.	21
4.2: Ünlü kullanıcılar ile düzenli kullanıcıları içeren ağ yapısı.	23
5.1: Ortalama doğruluk kullanılarak makine öğrenmesi algoritmaları karşılaştırması.	29
5.2: Lift eğrisi kullanılarak makine öğrenmesi algoritmaları karşılaştırması.	29
5.3: Etiket yerinin (düğümün kendisinin, gelen veya giden komşuları) tahmin kapasitesi üzerine etkisi.	32
5.4: Örüntü karmaşıklığının tahmin sonucu üzerine etkisi.	34
5.5: Ağ komşulukları yarıçapının tahmin sonucu üzerine etkisi.	35
5.6: Ağ bağlantı türünün birbiri üzerine etkisi.	36
5.7: GLM algoritması kullanılarak etiket yerinin tahmin üzerine etkisi.	38
5.8: GLM algoritması kullanılarak örüntü karmaşıklığının tahmin üzerine etkisi.	39
5.9: Örüntü karmaşıklığını arttırmanın tahmine etkisi deneyi için a) GLM kullanılarak lift sonucu b) DT algoritması kullanılarak lift sonucu.	39
5.10: Ağ komşulukları yarıçapının bütün örüntüler için tahmin sonucu üzerine etkisi.	40
5.11: Ağ komşulukları yarıçapının t1 örüntüleri için tahmin sonucu üzerine etkisi.	41
5.12: Dolaylı komşuluktan gelen tahmin bilgisinin a) GLM kullanılarak lift etkisi b) DT kullanılarak lift etkisi.	41

TABLolar DİZİNİ

<u>Tablo No:</u>	<u>Sayfa</u>
3.1: Twitter ağ tablosu.	18
4.1: Bu tablo her bir örüntü türünü, komşuluklarını dikkate alınarak açıklamaktadır.	22
5.1: Örüntü türü bazında örüntü sayısı.	27
5.2: Makine öğrenmesi algoritmaları için doğruluk ve lift kullanarak yapılan karşılaştırma tablosu.	28
5.3: Farklı veri setlerinde makine öğrenmesi algoritmaları için ortalama doğruluk değerleri (%).	30
5.4: Farklı veri setlerinde makine öğrenmesi algoritmaları için kümülatif lift değerleri.	30
5.5: Dengeli hedef değişkeninde makine öğrenmesi algoritmaları için ortalama doğruluk değerleri (%).	31
5.6: Dengeli hedef değişkeninde makine öğrenmesi algoritmaları için kümülatif lift değerleri.	31
5.7: Ağ bağlantı türünün birbiri üzerine etkisi tablo gösterim.	36
5.8: 10 ünlü düğümü için kendi örüntüleri üzerindeki bağlantı türünün nitelik önemi (%).	37

1. GİRİŞ

Son yıllarda sanal ortamların kullanımının artmasıyla beraber insanların günlük alışkanlıklarında birçok değişiklik meydana gelmiş ve gelmeye devam etmektedir. İnternet üzerinden alışveriş, internet bankacılığı, çevrimiçi haberleşme gibi birçok alanda eski alışkanlar değişerek internet üzerinden kullanım tercih edilmeye başlanmıştır. Gerçek dünyadan sanal dünyaya geçişin veya sanal dünyadaki ilginin artmasının temel nedeni internet olanaklarının iyileşerek günün herhangi bir anında kolay, hızlı, uygun bir şekilde kullanımının sağlanması ve bağlantılı olarak sosyal ağ uygulamaları ile sosyal medya sitelerinin hızlı çıkışıdır. Sosyal medya siteleri kendi içerisinde farklı kullanımlara olanak sağlayarak farklı kitledeki kullanıcıların da bu siteleri kullanımını sağlamıştır. Örneğin, kullanıcılar Facebook [Web 2, 2013] sitesini daha çoğunlukla ilgi alanları, fotoğraflarını veya videolarını paylaşmak için kullanırken, LinkedIn [Web 3, 2013] sitesini iş arama, meslektaşlarını ile iletişim kurma amaçları ile kullanmaktadırlar. Benzer şekilde Twitter [Web 4, 2013] ise, daha çoğunlukla fikirlerin ve yorumların paylaşıldığı bir ortam olarak kendini göstermektedir.

Kaplan ve Haenlein [Kaplan and Haenlein, 2010] sosyal medyayı, “Web 2.0 kullanan ve kullanıcı tarafından oluşturulan içeriğin oluşturulmasına ve paylaşılmasına olanak kılan bir grup İnternet tabanlı uygulama” olarak tanımlamıştır. Kullanıcının kendi içeriğini kendisinin oluşturabilmesi, eski yöntemdeki (Web 1.0) gibi herhangi bir yönetici tarafından onaylanma ihtiyacı olmaması sebebiyle daha aktif bir ortam sunması Web 2.0'ın en önemli yeniliği olarak belirtilmektedir. Bu nedenle kullanıcılar bu ortamları kullanarak daha kolay adapte olmaya, kendilerini ifade edip fikirlerini paylaşmaya başlamışlardır.

Sosyal medya ve ağ yapısının artan bu önemi, bu konudaki araştırmalarda son yıllarda artışa sebep olmuştur. Bu artışın ardındaki ana motivasyon İnternet kullanıcılarının kendileriyle ilgili oluşturduğu değerli içeriktir. Çünkü bu içerik, kullanıcılar hakkında kimlerle konuştuğu, nelerden hoşlandığı, politik görüşü, hobileri gibi pek çok konuda önemli bilgi vermektedir.

Bu konudaki ağların veri büyüklüğünden dolayı özellikle veri toplama, saklama ve işleme teknolojileri konusunda araştırma ve geliştirmeler artmıştır. Bu ağlardaki diğer çalışmalardan bazıları; bağlantı tahmini (link prediction), eğilim ve

fikir tahmini, duygu analizi (sentiment analysis) olmaktadır. Sosyal ağlar, mobil arama ağları ve WWW (Internet) bu konudaki ağlardan birkaç örneğini oluşturmaktadır.

Twitter, son zamanlarda popülerliği artan sosyal ağ (mikroblog) sitelerinden biridir. Maksimum 140 karakter uzunluğunda tweet adı verilen serbest formatta kısa mesajlar gönderebilmesini sağlayan bir servis sunmaktadır. Kullanıcılar birbirlerini takip ederek (follow) birbirlerinin attığı tweet'leri görebilmektedir. Kendi gizliliklerini genel (public) yaparak attıkları tweetlerin herkes tarafından görülebilmesini ya da gizli (private) yaparak sadece kendilerinin onayladıkları kullanıcıların görebilmesini de sağlayabilmektedir. 2003 yılındaki Twitter verilerine göre 200 milyon aktif kullanıcı günde 400 milyon tweet oluşturmaktadır [Web 5, 2013]. Bu kadar çok kullanıcının ve paylaşımın olduğu bir ortam bize gösteriyor ki kullanıcılar kendi fikirlerini ve paylaşımlarını gerçek dünyaya kıyasla çok daha hızlı yayabilmektedir. Bunun başlıca nedenlerinden biri, kullanıcıların birbirlerinden daha kolay etkilenmesidir. Son zamanlarda bazı sivil olaylarda Twitter'ın kullanıcılar üzerindeki etkisi ortaya çıkmıştır. Örneğin "Arap Baharı" [Web 8, 2013] olarak adlandırılan büyük çapta Tunus, Mısır, Libya, Suriye, Bahreyn, Cezayir, Ürdün ve Yemen'de gerçekleşen olaylarda, "Occupy Wall Street" [Web 6, 2013] olarak isimlendirilen New York'da gerçekleşen olaylarda koordinasyon ve iş birliğinin büyük ölçüde Twitter üzerinden sağlandığı görülmüştür. Son dönemde gerçekleşen "Gezi Parkı Olayı" [Web 7, 2013] Taksim, İstanbul, Türkiye'de gerçekleşen olay, büyük ölçüde Twitter'da başlayan ve bu platformun protestoların yayılmasında en fazla rolü olan toplumsal olaylar arasında yer almaktadır.

Twitter'ın etkili olduğu başka bir önemli alan ise, ürünlerin yayılmasında ve o ürünlerle ilgili yorumların paylaşılmasında sağladığı zemindir. [Web 9, 2013] Twitter ortamında yorumlar ürün yöneticileri veya firmalar tarafından çok değerlidir. Çünkü kendi ürünleri ile ilgili diğer ilgilenebilecek insanları yani hedef kitleyi bilgilendirebilecek hatta etkileyebilecek bir ortam bulma şansı elde etmişlerdir. Diğer bir deyişle bir firma ile kişi arasındaki ilişki, takip etme bağlantısından çıkarılabilir. Böylelikle kendi ürünlerini diğer insanlara yayarak satın alınmasını veya kullanılması sağlanabilir. Bu şekilde yapılan pazarlama, hedef pazarlama olarak isimlendirilir.

Bütün bu kullanımlar tweet'lerin ve sosyal ağların birçok farklı konu, olay ve kişilerin ilişkileri hakkında değerli bilgileri tuttuğunu ve analiz edilebileceğini

göstermektedir. Ancak başka bir taraftan sahte olan, yeterli bilgi sağlayamayacak olan bazı kullanıcılar da bulunmaktadır. Bu nedenle hangi ağdaki kullanıcıların seçilip işlenmesi gerektiği konusu ise zorlu, kritik ve üzerinde araştırmaları devam eden popüler bir konuyu oluşturmaktadır.

Ağ yapıları doğrudan olarak düğümlerden elde edilmesi zor olan bilgileri ortaya çıkarmak için kullanılmaktadır. Örneğin, protein etkileşim ağı içindeki hastalık ile ilişkili proteinlerin keşfi [Kırac et al., 2006], sosyal ağ kullanarak reklamlarda potansiyel müşteri hedefleme [Gupta et al., 2013], ve kimyasal molekül yapılarında benzer ağ örüntüleri araştırma gibi konular ağ tabanlı analitik metodolojileri kullanarak gerçekleştirebilir. Düğüm etiketli ağ (node-annotated network) şunlardan oluşur:

- Düğüm listesi
- Düğümler arasında ilişkiyi temsil eden bağlantılar
- Düğümlerle ilişkili özellikler

Örneğin, Facebook ağı düşünülürse: Facebook, bir düğüm etiketli sosyal ağıdır. Her düğüm kullanıcıyı, kullanıcılar arasındaki arkadaşlık aktivitesi bağlantıyı ve sevilen ya da favori edilen parçalar düğüm özelliğini oluşturmaktadır. Facebook içeriklerinin paylaşımları kullanıcıların kendisi tarafından gönüllü olarak gerçekleştirildiğinden dolayı, kullanıcıların hepsi bütün bilgilerini açıklamak istemez. Benzer şekilde başka bir örnek olarak Twitter ağı düşünülürse; Twitter da düğüm etiketli sosyal ağıdır. Her bir düğüm kullanıcıyı, kullanıcılar arasındaki takip etme ilişkisi bağlantıyı ve kullanıcının demografik bilgileri veya takip ettiği önemli kişiler düğümlerle ilişkili özellikleri oluşturmaktadır.

Bu çalışmadaki ana hipotez, bir sosyal ağdaki bir düğümün veya düğümler kümesinin topolojik ilişkilerini kullanarak bağlantı ilişkilerinin tahmin edilebilmesidir. Bir düğüm veya birden fazla düğümden oluşan düğüm kümesi için olası bağlantılar tahmin edilmeye çalışılabilir. Örneğin Twitter ağındaki Turkcell, gncrtkcell, profesyonellerkulubu isimlerindeki kullanıcılar hedef kümesi olarak seçilebilir. Bu kullanıcı veya kullanıcıların diğer kullanıcılar ile arkadaşlık ilişkileri kullanılarak bu kullanıcılar ile şu anda etkileşimi bulunmayan ancak etkileşimi olabilecek muhtemel kullanıcıların tahmin edilmeye çalışılması ana objektifi oluşturmaktadır.

Bu hipotezi doğrulamak için odaklanılan ana problem eksik veya tamamlanmamış düğüm etkileşim ve etiketlerini tamamlamaktır. Bu nedenle bir sosyal ağdaki bağlantılar ve sosyal ağın topolojisi kullanılarak eksik ya da tamamlanmamış bağlantı ve etiketlerin tahmin edilmesi problemi hedef alınmıştır. Kullanıcılar arasındaki henüz bulunmamış ancak bulunabilecek etkileşimler eksik ya da tamamlanmamış bağlantı etiketleri olarak düşünülmüştür. Böylelikle kullanıcılar için kendilerini takip etmeyen ama takip ettiğinde kendisi açısından faydalı olabilecek bağlantıları yani potansiyel etkileyici müşteriler tahmin edilmiş olacaktır. Örneğin Ahmet isimli kullanıcının X firmasını takip etmediğini varsayalım. Bu hipotez sonucunda varılmak istenen nokta, Ahmet'in kendisi ve çevresindeki arkadaşlarının bu firma ile ilişkili topolojik özelliklerine bakılarak X firmasına “Ahmet, X firmasının muhtemel takipçisi olur mu?” sorusunun cevabı verilmeye çalışılmaktadır.

Bu çalışma ile problemin çözümü yani ağ düğümlerinin bilinmeyen bağlantı veya etkileşimlerinin tahmin edilmesi için yeni bir yöntem sunulmaktadır. Bu metodolojide, ağ bağlantı tahmini problemi [Sen et al., 2008], düğüm etiket tahmini probleminin bir formuna dönüştürülerek çözülmektedir.

Bir ağdaki eksik olan düğüm etiketlerini tahmin etmek için bu çalışmada kullanılan yaklaşım şu adımlardan oluşmaktadır. İlk olarak, ağ bilgisi ön işleme (yani extract-transform-load) yapılacak ortama yüklenir. Sonra, her bir düğümdeki o düğüme ait komşuların bütün kullanılabilir örüntüleri toplanır ve gözlemlenen ağdaki etiketlenen örüntülerin toplam sayılarını temsil eden bir nümerik vektör özelliği yaratılır. Bu yöntem metin madenciliğinde kullanılan metin dokümanlarının terim frekans vektörlerine dönüştürüldüğü yöntem ile benzerdir [Lee et al., 1997]. Sonrasında ağ düğümlerine ait nümerik vektörlerden bir tablo oluşturulur ve oluşturulan bu tablo, modeldeki düğüm etiketleri arasındaki karmaşık ilişkileri açıklayabilmek için makine öğrenmesi algoritmalarına gönderilir. Son olarak, hedef özelliğin etiketleri ile olasılığı yüksek olarak ilişkilendirilen ağdaki düğümlerin etiketlerini kullanarak öğrenen modellerden yararlanılır.

Yukarıda açıklanan işlem büyük miktarda veri işleme gerektirmektedir. Bu işlem, büyük bir ağ verisinden etiket ağını oluşturma ve sayma yeteneğine bağlıdır. Birçok gerçek dünya vakasında, ağ verisi tek bir bilgisayardaki belleğe sığmayacak kadar büyüktür. Bu problemi çözmek için bir seçenek, çok pahalı süper bilgisayar kurarak örüntü sayma algoritmalarını uygulamaktır. Diğer bir seçenek ise, veriye

örnekleme yaparak küçültmektir. Ancak örnekleme yapıldığında, ağ yapısından örnek alıp bağlantısız parçalara ayırmak gerekmektedir ki bu durum bilgi kaybına neden olacaktır. Bu nedenle melez bir yaklaşım sunulmaktadır: Ağ örüntülerini disk tabanlı dağıtık işleme ortamında hesaplayarak, ağ verisi düğüm tabanlı özellik vektör setine dönüştürülür. Daha sonra, tek bir hesaplama çerçevesi üzerinde düğüm tabanlı özellik setinden örneklenmiş veri kullanılarak makine öğrenmesi algoritmaları çalıştırılır. Açıklanan hipotez ve problem kapsamında bu çalışmanın literatüre katkıları şunlardır:

- Ağ yapılarını öznitelik vektör kümelerine dönüştürecek sistematik bir yol
- Özellik vektörlerini paralel veri tabanı ortamında işlemek için ölçeklenebilir yöntem
- Bağlantı tahmini problemi için düğüm etiket tahmin probleminin bir türü olarak formüleştirilerek yeni bir çözümü

2. İLGİLİ ÇALIŞMALAR

Gerçek ağlardaki eş benzerlik (homophily) [Golub et al., 2012] ve yayılma (contagion) [Christakis et al., 2012] ile ilgili araştırmalar ağdaki birbirine bağlı düğümlerde paylaşılan özellikler kullanıldığında yararlı olduğunu göstermiştir. Sosyologlar aynı şeyi seven kişiler için eş benzerlik terimini oluşturmuştur [McPherson et al., 2001]. Kişilerin değişmez eğilimlerini açıklamak için inançlarını test etmek yerine çeşitli yollarla birbirleri ile bağlantı kurarak açıklamaya çalışmışlardır. Diğer bir deyişle bunlar, aynı şeyleri seven ya da aynı şeyleri yapmaktan hoşlanan kişilerdir. Yayılma ise özellikle etkileyici kişilerin (influencer'ların) oluşturduğu belirli bir davranışın ya da olayın yayılmasıdır. Örneğin bir kullanıcının sevdiği bir ürünü ya da haberi paylaşması ile diğer insanları etkileyerek o ürünü denemesi, haberi okuyup beğenmesi ve diğer insanların da bunu paylaşması yayılıma neden olmaktadır. Bu iki özellik, eş benzerlik ve yayılma, sosyal medyada sıklıkla görülmektedir. Bu çalışmadaki hipotez insanların etkileyici insanları takip ettiği hatta onlardan etkilendiği sonucunu temel olarak almaktadır.

Ağ yapılarının birçok farklı uygulama alanları bulunmaktadır. Örneğin, protein etkileşim ağı içindeki hastalık ile ilişkili proteinlerin keşfi [Kirac et al., 2006], sosyal ağ kullanarak reklamlarda potansiyel müşteri hedefleme [Gupta et al., 2013], kimyasal molekül yapılarında benzer ağ örüntüleri araştırma ve ağdaki bağlantı tahmini [Sen et al., 2008], biyoinformatik [Shojaie, 2013] gibi konularda pek çok araştırma ağ tabanlı analitik metodolojiler kullanılarak gerçekleştirilmiştir.

Ağ yapılarını inceleyen diğer bir araştırma, var olan düğüm özellikleri arasındaki korelasyonun ağdaki eksik veya bilinmeyen düğümlerin özelliklerini tahmin etmekte kullanılabileceğini ifade etmektedir [Zhou et al., 2009]. Bu araştırmaya benzer olarak bu tez çalışmasında gerçek ağdaki benzer özellikleri olan düğüm etiketlerindeki korelasyonu öğrenerek makine öğrenmesi algoritmalarıyla veri madenciliği modelinde kullanılmış ve eksik etiketli düğümlere uygulanmıştır.

Sosyal ağlarla ilgili çalışmalar psikologlar ve sosyal bilimciler tarafından da yürütülmektedir [Barabasi et al., 2002], [Dorogovtsev et al., 2002]. Ancak bilgisayar bilimleri alanındaki çalışmalar bağlantı tahmini problemine farklı bir açıdan bakmaktadır. Bağlantı tahmini problemi ile ilgili ilk ve genel yaklaşımda, ağ topolojisinden alınan özelliklerin denetimli öğrenmede etkin bir şekilde

kullanılabileceği ortaya ıkartıldı [Liben et al., 2004]. Bu alıřma baėlantı tahmini problemi iin birok etkin zelliėin var olduėunu gsterdi. Aė yapısındaki hangi zelliklerin tahmin doėruluėunu arttırarak daha etkin hale getireceėi grlmř oldu. Liben'in yaptıėı bu alıřmada sadece aė topolojisinden faydalanılarak oluřturulmuř bir zellik kmesi ile tahmin yapıldı. Bylece denetimli ėrenmede bu zelliklerin sistematik olarak kullanılabileceėini de gstermiř olunmuřtu. Buna ek olarak bařka bir alıřmada topolojik olmayan bazı zellikleri de (rneėin iki kiři arasındaki ortak ilgi alanları) ekleyerek deneyler yapıldı [Hasan et al., 2006]. Bu alıřmada buna benzer zelliklerin de baėlantı tahmininde doėruluėu iyileřtirdiėini ortaya ıkartılmıř oldu. Benzer bir alıřma olarak iliřkisel Markov aė algoritmaları kullanarak iyileřtirilme yapılmaya alıřıldı [Taskar et al., 2003]. Bu alıřmadaki ama, eksik olan web ve sosyal aėdaki dėmleri tahmin etmeye alıřmaktı. Bu řekildeki alıřmalar topolojik ve topolojik verilerden retilen topolojik olmayan zelliklerin tahmin etmede bařarılı bir řekilde kullanılabileceėini gstermiřtir.

Doėrudan baėlantı tahmini problemi ile ilgisi olmasa da bařka bir alıřmada kullanıcıların ortak ilgi alanlarına gre birbirleriyle iliřkili olduėu gsterilmiřtir. Doėal dil iřleme ve metin madenciliėi metotları [Lee et al., 1997] kullanarak yazılan yazıların (blog, mesajlar veya yorumlar olabilir) veya bu alıřma iin tweet'lerin benzerliėine bakılarak ortak ilgi alanları tespit edilmeye alıřılmıřtır. nk ortak ilgi alanları arttıa iliřki de aynı oranda artacaktır. Bu yntemde kullanılan metin dokmanlarının terimleri terim frekans vektrlerine dnřtrlr. Sonrasında metin madenciliėi algoritmaları ile terim frekans vektrleri kullanarak benzerlik bulunur. Yapılan bu tez alıřmasındaki yntem ile metin madenciliėinde kullanılan metin dokmanlarının terim frekans vektrlerine dnřtrldė bu yntem benzerdir. Sadece terim frekans yerine baėlantı frekansı kullanarak vektr haline getirilmiřtir.

Aė rnt analizi konusunda alt aė izomorfizm (subgraph isomorphism) kullanarak rnt tanıma, bilgi keřfi, biyoloji, dinamik aė trafiėi gibi birok alanda kullanılmaktadır [Aggarwal and Wang, 2010], [Conte et al., 2004], [Gallagher, 2006], [Shasha et al., 2002]. Ancak alt aė izomorfizm problemi NP-Tam bir problem [Garey, 1979] olduėu iin bu yntemle problemi zmek zor ve pahalıdır. Bu nedenle gerek dnyadaki veride uygulanmak zere yeni algoritmalar veya yntemler zerinde alıřmalar devam etmektedir. Yapılan bir alıřmada algoritmaları gerek dnyadaki veri byklėne uyumlu hale getirmek iin farklı dzenlemeler

yapılmış, artışı (incremental) yöntemler izlenmiştir [Fan, 2012]. Ancak bu çalışmalar ağ madenciliği algoritmalarının yapısından kaynaklanan karmaşıklıktan dolayı yavaş kalmaktadır. Bu çalışma kapsamında önerilen metot ile ağ madenciliği algoritmalarının karmaşıklığından kaçınılarak klasik öğrenme metotları kullanılabildiği için daha hızlı ve kolay işlem sonucu elde edilmesi sağlanmıştır.

Bu çalışma ile ilişkili daha birçok araştırma daha bulunmaktadır. Bu çalışmalara küçük ağlar (örneğin protein-protein etkileşimleri) için güvenilir komşulardan yararlanma [Goldberg et al., 2003], sosyal ve biyolojik ağlar için tahmin algoritmaları geliştirmeleri [Clauset et al., 2008] örnek olarak verilebilir.

3. METODOLOJİ

Bu çalışma kapsamında Twitter ağındaki eksik olan düğüm etiketlerini tahmin etmeyi amaçlanmıştır. Bu bölümde metodoloji ile ilgili kullanılan gerekli tanımları içermektedir.

3.1. Sosyal Ağ Analizi

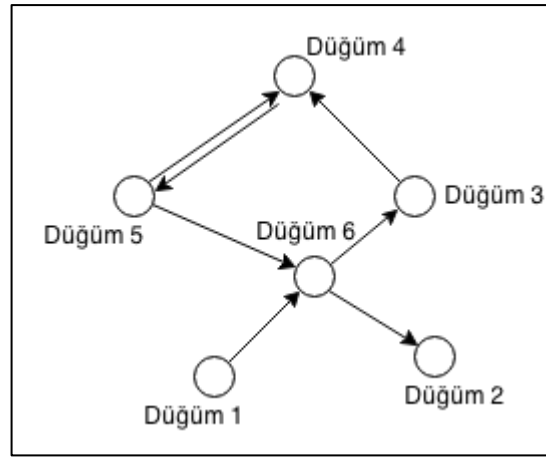
Sosyal ağ analizi, bir sosyal ağdaki düğümler kümesi arasındaki çeşitli ilişkilerin analizidir. Sosyal ağ, sosyal aktörler ve aktörler arasındaki etkileşimleri açıklayan sosyal bağlardan oluşur. Bu sosyal aktörler, arkadaşlar, akrabalar, olay içerisinde geçen kullanıcılar veya ünlü isimler olabilir. Sosyal ağları incelemek için birçok farklı teknik, yaklaşım ve perspektif kullanılmıştır. Bu tekniklerden bazıları ağdaki zayıf veya kuvvetli bağları bulmaya [Leskovec et al., 2010] çalışırken, bazıları topluluk bulmaya [Porter et al., 2009] çalışmaktadır. Sosyal ağlar, düğümler arasındaki ilişkilere göre ağırlıklandırılmış veya ağırlıklandırılmamış, yönlendirilmiş veya yönlendirilmemiş olarak tanımlanabilir.

3.2. Twitter

Twitter, kullanıcıların serbest formatta maksimum 140 karakter uzunluğunda kısa mesaj veya bu mesajlara eklenebilecek bir fotoğraf veya maksimum 6 saniye uzunluğunda video gönderebilmelerini sağlayan çevrimiçi sosyal ağ sitesi ve mikroblogtur. Bu mesajlar tweet olarak bilinmektedir. Kullanıcılar herhangi bir konu hakkında bu siteye tweet yazabilmektedir. Kullanıcılardan birbirlerini takip edenler birbirlerinin tweet'lerini görebilmektedir. Bu nedenle her bir kullanıcının takipçi ilişkisi bulunmaktadır. Kullanıcı gizlilik ayarlarından kendisini genel olarak ayarlayıp herhangi diğer bir kullanıcının onayını beklemezsizin takip etmesini ve tweet'lerini okumasını sağlayabileceği gibi gizli yaparak sadece kendisinin onay vereceği kullanıcıların takip etmesini ve tweet'lerini okuyabilmesini de sağlayabilir. İşte takip edilen bu listeye takipçi listesi olarak adlandırılmaktadır. Takip edilme ilişkisi tek yönlüdür. Çünkü kullanıcı diğer kullanıcıyı takip eder veya etmez. Daha

önce Bölüm 1’de de bahsedildiği üzere Mart 2003 yılındaki Twitter verilerine göre 200 milyon aktif kullanıcı günde 400 milyon tweet oluşturmaktadır. Bu sayıların yanında tweet yazmadan sadece tweet’leri okuyan kayıtlı olmayan kullanıcılar da bulunmaktadır. Bu sayı da ağın büyüklüğünü ortaya çıkarmaktadır.

Şekil 3.1’de küçük bir takip ağı örneklendirilmiştir. Bu ağda düğümler rakamlarla isimlendirilmiştir. Bu ağın bize anlattığı bazı bilgiler vardır. Düğüm 1, Düğüm 6’yı takip ederken; Düğüm 6, Düğüm 1’i takip etmemektedir. Diğer taraftan Düğüm 5, Düğüm 4’ü, Düğüm 4’de Düğüm 5’i karşılıklı olarak takip etmektedir.



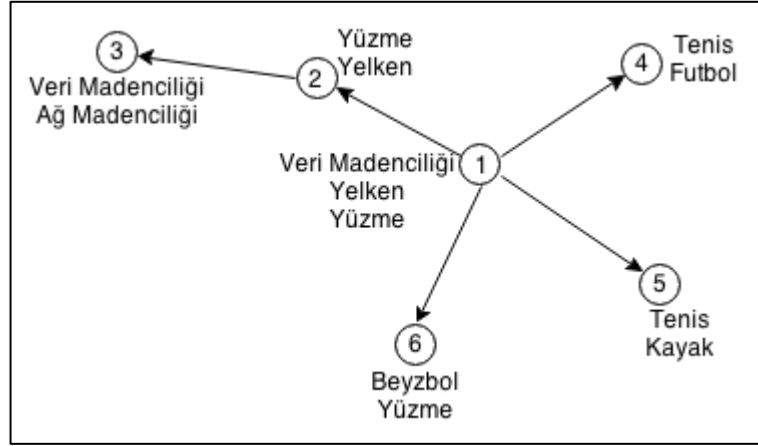
Şekil 3.1: Örnek bir takip ağı.

Kullanıcıların Twitter’daki kendi takip ettiği kişilerin gönderdiği tweet’leri yeniden gönderebilir, bu olaya retweet olarak isimlendirilmektedir. Ayrıca beğenebilir (favorite), yanıt verebilir (reply), metnin içinde kullanıcı adını “@” sembolü ile yazarak anabilir (mention).

3.3. Ağ Madenciliği

Ağ madenciliği, özellikle ağ olarak tanımlanabilecek sosyal medyada, proteinlerin etkileşimi ve keşfi [Kirac et al., 2006] gibi birçok noktada tahmin olanağı sağlamaktadır. Örneğin örüntü analizi veya anormallik tespitinde de ağ madenciliği metotları kullanılmaktadır [Kang et al., 2012]. Özellikle sosyal ağ analizinde yayılma tespitinde sıkça ağ madenciliği metotları kullanılmaktadır. Sosyal ağ analizinde kullanılan diğer örneklerden bazıları önemli insanların ve etkileyici

insanları (influencer) bulma [Anagnostopoulos et al., 2008] ve sosyal ağlarda bağlantı tahminidir [Sen et al., 2008].



Şekil 3.2: Örnek bir ilgi alanı ağı yapısı.

Şekil 3.2’de örneklendirilen ilgi alanı ağını kullanırsak, her daire bir insanı, düğümler arasındaki her iletişim bağlantısını (örneğin kimin kimi takip ettiğini veya arkadaş olduğunu) ve her düğümün bir ilgi alanını temsil edecek bir etiketi bulunmaktadır.

Düğümelerde var olmayan etiketleri düğümün kendisi ve çevresindeki düğümlerin ortak etiketlerinden faydalanarak bulmayı etiket tahmini yöntemi olarak isimlendirilmektedir. Örneğin, her düğüm numaralandırılmıştır. 1. Düğüm, 2. Ve 6. Düğüm ile etkileşim içindedir. Bu üç düğümün ortak ilgi alanı yüzme olduğu görülmektedir. Bu durumda 1. ve 2. Düğüm yelken yapmaktan hoşlandığı için ortak ilgi alanının (Yelken) da olduğu düşünülerek 6. Düğüm de bu ilgi alanından hoşlanabilir mi sorusuna cevap ararken etiket tahmini yöntemleri kullanılmaktadır.

3.4. Kullanılan Makine Öğrenmesi Algoritmaları

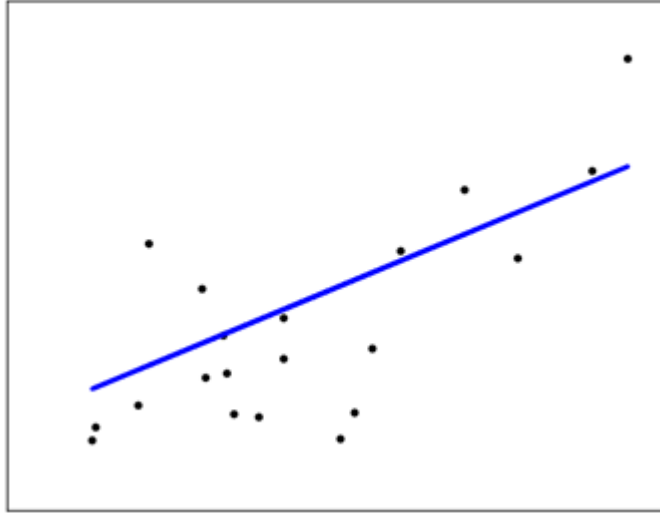
3.4.1 Genelleştirilmiş Lineer Model (GLM)

Genelleştirilmiş lineer model (GLM) lineer regresyon (linear regression) model tekniğini genişleten bir denetimli öğrenme algoritmasıdır. Eğer tahmin edilen hedef değişken ile tahmine yardımcı olan değişkenler arasında ilişki kesin bir çizgi ile

yakınsayabilirse, lineer regresyon tekniği kullanılabilir. Şekil 3.3’de lineer regresyon örneği verilmiştir. Bu örnekte belli koordinatlara dağılmış noktaları gruplamak için lineer bir model bulunmak isteniyor. Bunun için de temsilen mavi çizgiyi oluşturan regresyon yapılıyor.

$$\text{Lineer Problem formülü: } y = F(x, \theta) + e = \theta_2 x + \theta_1 + e \quad (3.1)$$

- θ_2 , çizginin eğimi, verinin bulunduğu nokta ile regresyon çizgisinin arasındaki açıdır.
- θ_1 , y sınırlarını çizen kısım, noktanın x koordinatının y koordinatını kestiği yerdir ($x = 0$).



Şekil 3.3: Lineer regresyon örneği.

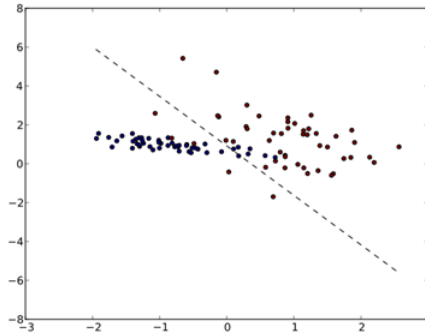
Lineer modellerin sağladığı avantaj hesaplama kolaylığı, kolay yorumlanabilir model formu ve uyum kalitesi hakkında kesin teşhis bilgilerinin hesaplanabilmesidir. Ancak bunun yanında hedef değişkenin normal dağılımlı olduğuna bakılmaksızın sabit bir varyans değeri ile diğer değişkenler kullanılarak oluşturulur. Hedef değişkenin normal dağılımlı olduğu varsayım olarak alınır. GLM ise pratikte daha etkin olarak kullanılabilmesi için bu varsayımları yumuşatır. Örneğin ikili cevaplar (evet/hayır ya da 0/1) aynı varyansa sahip değildir. Üstelik lineer modelden farklı olarak ikili model için cevap değerleri pozitif veya negatif değerleri kapsayan çok büyük alana yayılmak yerine $[0, 1]$ aralığındaki değerleri oluşturur.

GLM lineer modelin varsayım olarak gördüğü iki mekanizmayı sağlar: bağlantı fonksiyonu ve varyans fonksiyonu. Bağlantı fonksiyonu hedef değişkenini artı/eksi sosuz değerleri için belli bir aralığını oluşturur diğer bir deyişle ayırıklaştırır. Varyans fonksiyonu tahmin edilen değer fonksiyonunu ifade eder. Bu nedenle sabit olmayan varyanslı uygun cevaplar oluşturulur (örneğin iki cevap gibi).

En iyi regresyon modelleri hedef değişken ile yüksek korelasyonu olan diğer değişkenlerin oluşturduğudur ancak bu diğer değişkenlerin kendi arasında da çok küçük de olsa bir korelasyon vardır. Çoklu doğrusalılık (Multicollinearity) bunu açıklamak için kullanılan bir terimdir. Çoklu doğrusal gidermek için tepe regresyon (ridge regression) kullanılan bu teknikte yapılan problemi adreslemek için katsayıların büyüklüğüne göre bir cezalandırma yapılır. Bu cezalandırma tepe değişkeni eklenerek yapılır.

3.4.2 Destek Vektör Makineleri (SVM)

Destek vektör makineleri sınıflandırma, regresyon ve aykırı değerler algılamada kullanılan denetimli öğrenme algoritmasıdır. Avantajları; yüksek boyutlu alanlarda etkin, hatta boyut sayısı örnek sayısından fazla olduğunda dahi etkindir. Karar fonksiyonlarda (destek vektörleri denilir) eğitim noktalarının alt kümesi kullanılır bu nedenle bellek açısından da etkindir. Karar fonksiyonları için farklı çekirdek fonksiyonları (kernel functions) tanımlanabilir. Genel olarak tanımlanmış lineer, lineer olmayan (gaussian) fonksiyonları da bulunduğu gibi özel fonksiyonlar da tanımlanabilir. Dezavantajları; özelliklerinin sayısı örnek sayısından çok fazla olduğunda kötü bir performans vermesi muhtemeldir. Doğrudan olarak olasılık tahmini yapmaz 5-fold çapraz doğrulama ile pahalı bir hesaplamaya ihtiyaç duyulur.



Şekil 3.4: Destek vektör makineleri örneği.

3.4.3 Naive Bayes (NB)

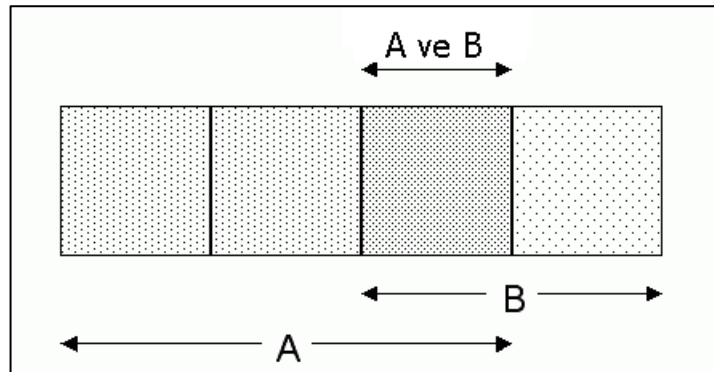
Naive Bayes şartlı olasılığını temel alan denetimli öğrenme algoritmasıdır. Bayes teoremini kullanır. Bu teorem, değişkenlerin değerlerinin ve değerlerinin kombinasyonlarının frekans değerlerini sayarak olasılık hesaplayan bir formülden oluşur. Özellikle örüntü tanımada (pattern recognition) sıklıkla kullanılır. NB hızlı, ölçeklenebilirdir ve aynı zamanda paralelleştirilebilir. Verilen bir sınıftaki y değişkeni ve x_1 'den x_n 'e kadar özellik vektörleri Bayes teoreminde şu ilişkiyi belirtir:

$$P(y | x_1, \dots, x_n) = \frac{P(y) P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)} \quad (3.3)$$

Bu formül ve teorem ile daha önceden oluşan bir olayın verilen olasılığının tekrar oluşma olasılığını bulur. Formülü daha basit ifade etmek gerekirse, B bağlı olan olay ve A önceki olayı temsil ederse, Bayes' teorimi şu şekilde ifade edilir:

$$P(B \text{ verilen } A) = \frac{P(A \text{ ve } B)}{P(A)} \quad (3.4)$$

Örneğin 22 yaşın altındaki müşterinin harcamalarını arttırmasının ihtimali tahmin edilmek istendiği düşünölsün. Bu durumda önceki koşul (A) “22 yaşın altı” ve bağlı koşul (B) “harcamaların arttırılması” olmuş olur. Şekil 3.5 bu örneğin grafiğini temsil etmektedir.



Şekil 3.5: Naive bayes örneği.

Eğer eğitim verisinde 100 müşteri varsa ve 25 müşteri 22 yaşının altında ve harcamalarını arttırıyorsa, bu durumda;

- $P(A \text{ ve } B) = \%25$

Olur. Eğer 100 müşterinin 75'i 22 yaşının altında ise;

- $P(A) = \%75$

Bayes teoremi hesaplanırsa;

- $P(B \text{ verilen } A) = \frac{25}{75} = \%33$

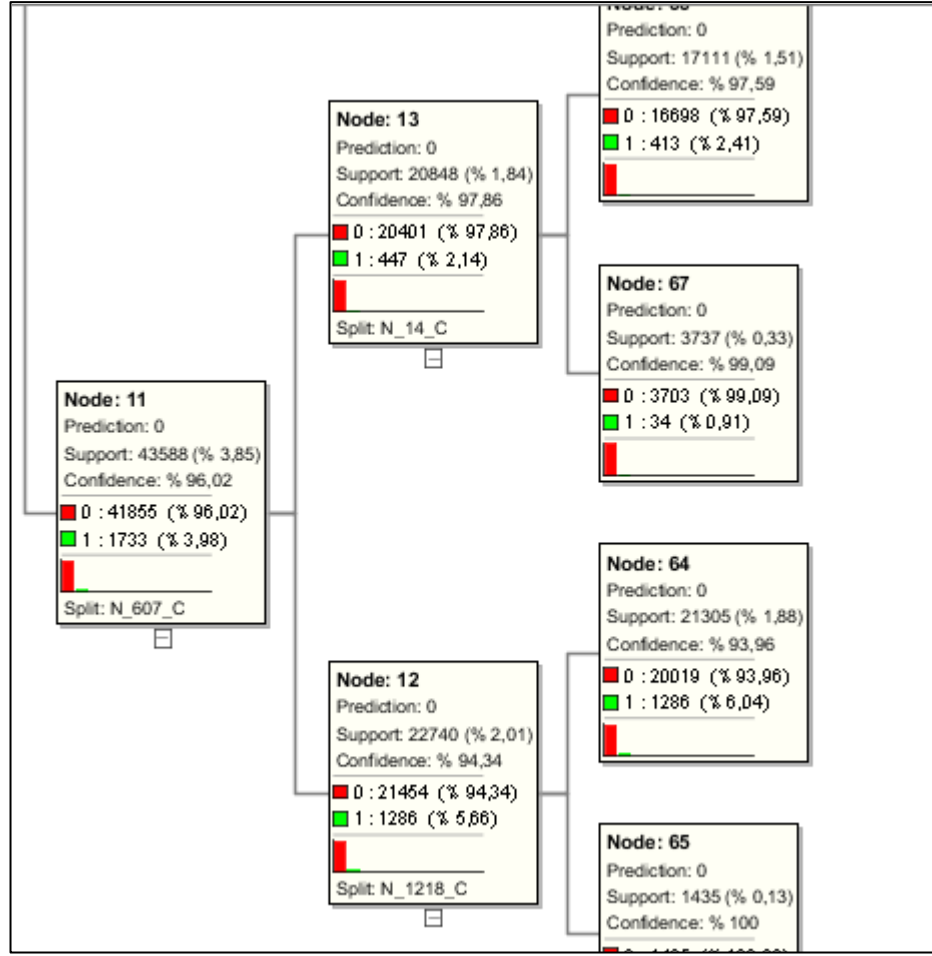
Eğer bütün koşullar beraber sağlanırsa birlikte (pairwise) olarak nitelenir. Bu örnekte koşulların %25'i birlikte. Eğer sadece önceki koşul sağlanırsa tekil (singleton) olarak nitelenir. Bu örnekte koşulların %75'i tekildir.

3.4.4 Karar Ağaçları (DT)

Karar Ağacı algoritması Naive Bayes gibi şartlı olasılığını temel alan denetimli öğrenme algoritmasıdır. Naive Bayes'den farklı olarak karar ağaçları kuralları üretir. Kurallar, kullanıcılar tarafından kolayca anlaşılabilir olacak şartlı ifadelerdir (Şekil 3.6 ve Şekil 3.7) ve veri tabanı içerisinde kayıt kümesini tanımlamak için kullanılır.

Bazı veri madenciliği uygulamalarında sadece tahmin doğruluğu önemlidir. Modelin nasıl çalıştığına bir önemi olmayabilir. Ancak diğerlerinde verilen kararların açıklanması kritik önem taşımaktadır. Örneğin başarılı bir kampanya pazarlaması için müşteri segmentasyonu belirlemede karar ağacı algoritması kullanılabilir.

Avantajları; anlaşılması kolay, hem nümerik hem de kategorik veriyi kullanabilir, beyaz kutu modeli kullanır bu da kullanımı kolaylaştırır, istatistiksel testlerle modelin geçerliliği test edilebilmesidir. Dezavantajları; çok fazla eğitildiğinde (overfitting) karmaşık ağaçlar oluşabilir, belirli ağaçlar oluşabilir bu nedenle eğitimden önce veri setini dengeli hale getirmek faydalı olabilir.



Şekil 3.7: Bir Karar Ağacının (Yatay) belli bir parçası gösterilmektedir.

Örneğin Şekil 3.7’deki ağaç kullanılarak düğüm 64 için kural oluşturulursa; “Eğer N_616_C $\leq$ 0,5 ve N_611_C $\leq$ 0,5 ve N_613_C $>$ 0,5 ve N_612_C $\leq$ 0,5 ve N_607_C $\leq$ 0,5 ve N_1218_C, "0" ise o zaman 0’dır” olarak oluşturulmuş olur.

3.5. Nitelik Önemi (Attribute Importance) Algoritmaları

3.5.1 Minimum Tanım Uzunluğu (MDL)

Minimum Tanım Uzunluğu model seçimi için teorik bilgi veren bir algoritmadır. MDL en kolay, en sıkı veri gösteriminin en iyi ve en doğru tahmin edilebilir veri olduğunu varsayar. Her bir niteliği modelin hedef değişkeni olarak düşünerek tahmin yapar. Bu tahmin modelleri karşılaştırılır ve sıraya koyulur. Ayrıca MDL gereğinden fazla öğrenmeyi engellemek için model karmaşıklığını cezalandırır.

MDL’de model seçimi problemine bir haberleşme problemi gibi davranılır. Bir verici, bir alıcı ve iletilecek bir veri vardır. Sınıflandırma modelleri için iletilecek veri bir modeldir.

Nitelik önemi veriyi iletmek için iki parçayı kullanır. İlk parça (giriş) modeli iletir. Modelin parametreleri, her tahmin değeri ile ilişkili hedef olasılıklardır. Her bir hedef j değeri için ve diğer değişkenleri k olan, her bir değer için n_i ($i=1,...,k$) veri dizisi, n_i-1 zamanındaki koşullu olasılığı $j-1$ değişkenin kombinasyonu için C_i olsun. Giriş bit büyüklüğü $\sum_k \log_2 C_i$ olarak gösterilebilir. İkinci bölüm hedef değerlerinin model kullanılarak iletilmesidir. İletim için en iyi bilinen yöntem en iyi eşleşmeye sahip hedef değişken olasılığını kodlamaktır. Bu nedenle dizideki en yüksek olasılığı en düşük iletim maliyetine sahip olur. Bu maliyet formül olarak $\sum_k \log_2 p_i$, p_i her bir satır için modeldeki tahmin edilen olasılık olur.

3.6. Metotlarla İlgili Kullanılan Problem ve Tanımları

Tanım 3.1 (Ağ): Bir yönlendirilmiş (directed) ağ $G = (N, L, F, A)$, $l \in L$; $l: n_1 \rightarrow n_2$; $n_1, n_2 \in N$ olsun. N , düğümler kümesini, L iki düğüm arasında kurulan bağlantılar kümesini, F düğüm hakkında (örneğin ilgi alanları, hobiler) bilgi içeren özellik (feature) kümesini, A özelliklerle ilişkili özellik etiketlerini (feature annotations) ($f: n \rightarrow a, a \in A; n \in N; f \in F$) tanımlar.

Örnek 3.1: Daha önceden kullanılan Şekil 3.2 ağ yapısı örneğini ele alındığı düşünölsün. Düğümler 1’den 6’ya kadar numaralandırılmış sosyal ağ profillerini temsil etmektedir ve her düğüme yüzme, yelken yapma, basketbol oynama gibi bazı özellikler atanmıştır. Düğümler, kendileri arasındaki arkadaşlığı veya ilişkiyi temsil eden bağlantılar (link) ile bağlanmıştır.

Twitter ağı iki sütunlu kullanıcı çiftleri biçimindeki bir tabloda oluşturulmuştur. Her bir kullanıcı bir düğüm f , her bir kullanıcının ilgi alanı A ifade etmektedir. Twitter takip tablosundan bir alıntı Tablo 3.1’de gösterilmiştir. Bu tabloda gösterildiği gibi her bir sütun bir Twitter kullanıcı düğümünü ve her kayıt bir yönlendirilmiş bağlantıyı temsil etmektedir. Bundan dolayı Twitter tablosu Twitter ağındaki bağlantı listesinden oluşmaktadır. Tablo 3.1, kullanıcılardan kimin kimi

takip ettiğini belirtmektedir. Bu tablo, kullanıcı numaralarından (id) oluşmaktadır. Örneğin, kullanıcı 48954673, 49233593, 27433315, 60843485, kullanıcı 43661838'ün takipçileridir. Benzer şekilde kullanıcı 27433315, kullanıcı 43661851'in takipçisidir. Twitter ağı ve daha detaylı bilgi Twitter Uygulama Arayüzü (API) kullanılarak elde edilebilir. [Web 1, 2013]. Bu tez çalışması kapsamında 2010 yılında yayınlanan makalenin Twitter Sosyal Ağ verisi kullanılmıştır [Kwak et al., 2010]. Bu veri setinde yaklaşık 40 milyon düğüm ve 1.5 milyar bağlantı vardır.

Tablo 3.1: Twitter ağ tablosu.

From Id	To Id
43661838	48954673
43661838	49233593
43661838	27433315
43661838	60843485
43661851	27433315

Problem 3.1 (Ağ Bağlantı Tahmini): Gerçek dünyada ağlara sürekli olarak yeni düğüm, bağlantı veya etiket eklendiği veya çıkartıldığı için ağ yapıları sürekli değişmektedir. Bu sebeple, bir ağ $G = (N, L, F, A)$ gerçek dünyadaki o ağın belli bir t_1 zamanındaki anlık görüntüsüdür ve ağ gelişip değişerek t_2 zaman diliminde ($t_2 > t_1$) $G' = (N', L', F', A')$ olmaktadır. Ağ bağlantı tahmini problemi yeni bir bağlantı olan $l = n_1 \rightarrow n_2; l \notin L, n_1, n_2 \in N; l \in L', n_1, n_2 \in N'$ tahmin edilmeye çalışılmaktadır.

Problem 3.2 (Ağ Etiket Tahmini): Gerçek dünyadaki ağ verisi tamamen verilen bir ağ tarafından temsil edilemeyebilir. Ağ etiketleri tamamlanmamış ve var olan etiketler veri kalitesi problemlerinden dolayı doğru olmayabilir. Gerçek dünyadaki ağ modeli $G = (N, L, F, A)$, elde edilen verideki ağ modeli G 'nin bilinen temsili $G' = (N, L, F', A')$ olsun. Ağ etiket tahmini problemi $A - A'$ da belli olasılık ve güvenle yeni etiketler bulmaya çalışma problemi olarak tanımlanabilir. Bu problem, aynı zamanda ağ etiket düzeltme problemi olarak da düşünülebilir. Ağ etiket düzeltme

problemi, var olan $A' - A$ 'daki etiketler gürültü veya hatalı olarak belli olasılık ve güvenle atanmasıdır.

Tanım 3.2 (Ağ Komşuluğu): $G = (N, L, F, A)$ ağındaki bir düğüm $n \in N$ gelen doğrudan komşuları $m \in N$, ve $\exists l_1 \in L, l_1 = m \rightarrow n$ ve giden doğrudan komşuları $o \in N$, ve $\exists l_2 \in L, l_2 = o \rightarrow n$ olsun. G ağındaki $n' \in N$ 'dan $n_z \in N$ 'ya olan doğrudan yolun uzunluğu z , bütün bağlantılardan, $l_1, l_2, \dots, l_z$ öyle ki $l_1 = n' \rightarrow n_1$, $l_2 = n_1 \rightarrow n_2, \dots$, ve $l_z = n_{z-1} \rightarrow n_z$ oluşmaktadır. Bir düğüm n 'in bütün doğrudan komşuları n üzerinden 1 doğrudan yol uzunluğu ile bağlıdır. Bir düğüm n 'in ağ komşuluğu, n ile en az bir doğrudan yol ile bağlı düğümlerden oluşur. N düğümüne ait ağdaki komşuluklarında bütün düğümler arasındaki maksimum doğrudan yolu kısıtlandığında ve düğüm n , sabit bir sayı z verildiğinde, n 'in ağ komşuluklarının yarıçapı z 'dir.

Bu tez kapsamında, Twitter kullanıcı profillerinin özelliklerinin tahmin edilmesine odaklanılmıştır. Twitter kullanıcı ilgi alanlarını bulabilmek için birçok farklı yol bulunmaktadır. Kullanıcı haber kaynaklarındaki metinler veya kullanıcı tweet'leri, ilgi alanlarını elde etmek için doğal dil işleme ve metin madenciliği kullanılarak elde edilebilir [Pennacchiotti et al., 2011]. Başka bir çalışmada, kullanıcıların fiziksel konum ve ilgi noktalarında (point of interests) olduğunu paylaşan kullanıcıların konumlarından (check in'lerinden) bulunabilir. Konumların kullanıcı ilgi alanlarını yansıttığından dolayı, konum tabanlı Twitter kullanıcı ilgi alanı oluşturmak da mümkündür [Lee et al., 2010].

Bu çalışma, ünlü insanları (celebrity) Twitter kullanıcılarının ilgi alanı olarak düşünecektir. Ünlü kullanıcılar ile düzenli kullanıcılar arasında Facebook'da kesin bir ayrım bulunmasına rağmen (örneğin Facebook'daki ayrı sayfalar, insanlar varken) Twitter'da kesin bir ayrım bulunmamaktadır. Bundan dolayı bu basit yaklaşım ile bu ayrım oluşturulmaya çalışılmıştır. Takipçi sayısı belli bir eşik değerinin üzerinde olan kullanıcılar, ünlü kullanıcılar olarak değerlendirilmiştir. Belirtilen ünlü kullanıcıların aktiviteleri düzenli kullanıcıların ilgi alanları olarak değerlendirilmiştir. Kısaca bu çalışmadaki amaç, düzenli kullanıcıların ilgi alanlarını (örneğin etiketleri) tahmin etmek veya eşdeğer olarak, normal kullanıcıların ünlü

kullanıcıları takip edip etmeyeceğini (örneğin ilgi gösterip göstermeyeceğini) tahmin etmektir (örneğin ünlü aktris, bir rock grubu veya otomobil üretici firması).

Tanım 3.3 (Ağ Bağlantı Tahmini probleminin Ağ Etiket Tahmini problemini dönüşümü): Gerçek hayattaki ağlar ölçüsüz özellikler [Kwak et al., 2010] içerir. Bu özelliklerin olduğu yerde, düğümlerden bazıları çok yüksek bağlantı sayısına sahipken diğer düğümler daha az bağlantı sayısına sahiptir. Genellikle, yüksek bağlantı sayısına sahip düğümlerin gerçek dünyada özel bir anlamı vardır (örneğin Twitter ağındaki ünlü kişiler veya markalar). Bundan dolayı, yüksek bağlantı sayısına (örneğin belli eşik değerinin üzerindeki) sahip bütün düğümler silinerek düzenli düğümler (örneğin genel kullanıcılar) kümesi elde edilebilir. Silinen düğümlerin bağlantı bilgisini kaybetmemek için, bilgiler düğüm etiketlerinde saklanır; farz edelim ki u yoğun bağlantı bilgileri olan bir ünlü düğümü, v ve w ise çok daha az bağlantı sayısı olan genel düğümler olsun. Ayrıca farz edelim ki ağ G şu bağlantıları içermektedir: $v \rightarrow u$ ve $u \rightarrow w$. Bu bağlantılar ($v \rightarrow u$ ve $u \rightarrow w$), v düğümünün bağlantısı $\{giden_U\}$ ve w düğümünün bağlantısı $\{gelen_U\}$ olarak, v ve w iki yeni etiketine dönüştürülmektedir. Bağlantı bilgisi etiket bilgisine dönüştürüldüğü için, artık bağlantı tahmini problemini etiket tahmini problemi olarak ifade edebiliriz. Bu çalışma kapsamında $\{giden_U\}$ kullanılmıştır. Başlangıçta Twitter ağ veri seti etiketlenmemiş yönlendirilmiş ağdır. Ağdaki bütün ünlü düğümleri sildikten sonra, ünlü kullanıcıların bağlantılarını düzenli kullanıcıların etiketleri olarak dönüştürülür. Bu durumda bu etiketli yönlendirilmiş ağ haline gelir.

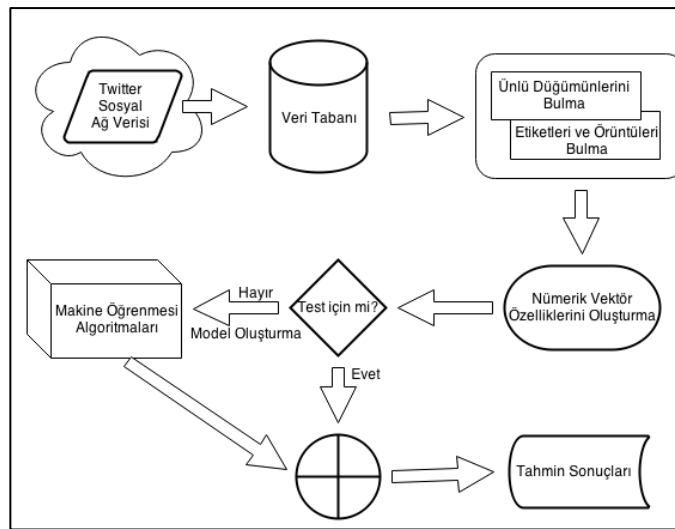
Tanım 3.4 (Ağ Örüntüsü): Verilen yönlendirilmiş etiketli ağ $G = (N, L, F, A)$, ağ örüntüsü bir alt ağ $p = (N', L', F', A')$, öyle ki $N' \subseteq N$, $L' \subseteq L$, $F' \subseteq F$, $A' \subseteq A$ ve p her $n_1, n_2 \in N'$ düğümleri arasında en az bir yönlendirilmiş yol bulunan bağlantılı bileşendir (connected component). p , G 'nin alt ağlarında çok küçüldüğünde p ile G arasındaki izomorf haritası (isomorphic mapping) birden fazla olabilir. Bu durumda p , G ağındaki tekrarlı ağ örüntüsü olarak isimlendirilir.

4. YÖNTEMLER VE ALGORİTMALAR

Bu çalışma kapsamında bir ağdaki eksik veya tamamlanmamış olan düğüm etiketleri tahmin edilmeye çalışılıyordu. Bunu yapmak için bu tez çalışmasında kullanılan yaklaşım adımları özetlenecek olursa;

- Ağ bilgisi ön işleme yapılacak ortama yani bu çalışma için veri tabanına yüklenir.
- Ünlü düğümü olarak seçilecek veya tespit edilecek düğümler bulunur.
- Bu ünlü düğümüne ait etiketler oluşturulur.
- Her bir düğümdeki o düğüme ait komşuluklarının bütün kullanılabilir örüntüleri toplanır ve ağdaki etiketlenen örüntülerin toplam sayılarını temsil eden bir tane nümerik vektör özelliği yaratılır. Ağ düğümlerine ait nümerik vektörlerden bir tablo oluşturulur.
- Oluşturulan bu tablo, modeldeki düğüm etiketleri arasındaki karmaşık ilişkileri açıklayabilmek için makine öğrenmesi algoritmalarına iletilir.
- Hedef özelliğin etiketleri ile olasılığı yüksek olarak ilişkilendirilen ağdaki düğümlerin etiketlerini kullanarak öğrenen modellerden yararlanılır.

Bu yaklaşımdaki kullanılan adımlar ve metodolojisi şekil 4.1’de gösterilmiştir.



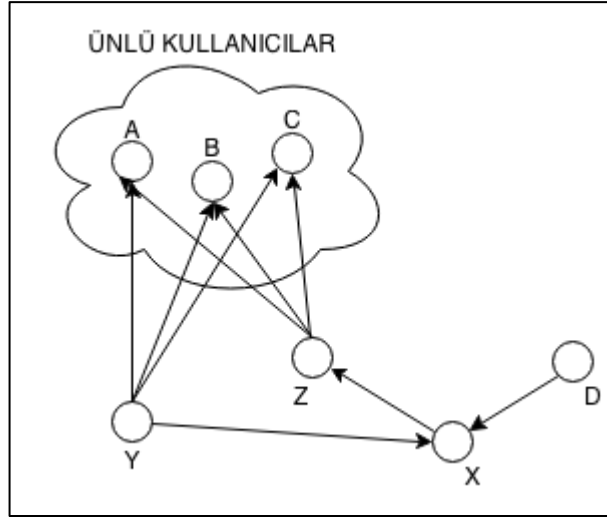
Şekil 4.1: Eksik olan etiketleri tahmin etmek için kullanılan yaklaşımın metodolojisi.

Tanım 4.1 (Ağ Örüntü Düzleştirilmesi): $G = (N, L, F, A)$ ağındaki bir ağ örüntüsü $p = (N', L', F', A')$ vektör halinde aşağıdaki gibi düzleştirilir;

Bir n düğümündeki etiket alt kümesi şu şekilde tanımlanır: $S(n) = \{s_i \mid s_i \rightarrow f, s_i \in A, f \in F, n \in N\}$. Sonrasında, n düğümündeki ve n düğümünün komşuluklarında z yarıçapında bulunan bütün diğer düğümlerdeki alt kümelerinin etiketlerinin listesi, p hesaplanır. p tekrarlı etiket alt kümelerine sahip olabilir, bunlar gruplanarak farklı etiket alt kümeleri bulunabilir ve tekrarlanma sayıları kayıt edilir. Böylece ağ örüntülerinin sayısına sahip vektörlere düzleştirilmiş ağ örüntüsü vektörü olarak adlandırılmaktadır (Tablo 4.1).

Tablo 4.1: Bu tablo her bir örüntü türünü, komşuluklarını dikkate alınarak açıklamaktadır.

Tür	Açıklama
t1	1 uzunluğunda 0 yarıçapındaki olan bağlantıların örüntüleri
i1	1 uzunluğunda 1 yarıçapı içinde olan gelen bağlantıların örüntüleri
i2	2 uzunluğunda 1 yarıçapı içinde olan gelen bağlantıların örüntüleri
o1	1 uzunluğunda 1 yarıçapı içinde olan giden bağlantıların örüntüleri
o2	2 uzunluğunda 1 yarıçapı içinde olan giden bağlantıların örüntüleri
c1	1 uzunluğunda 1 yarıçapı içinde olan gelen, giden ve kendisi dâhil tüm bağlantıların örüntüleri (örneğin t1-i1-o1)
c2	2 uzunluğunda 1 yarıçapı içinde olan gelen, giden ve kendisi dâhil tüm bağlantıların örüntüleri (örneğin t1-i2-o2)
c3	3 uzunluğunda 1 yarıçapı içinde olan gelen, giden ve kendisi dâhil tüm bağlantıların örüntüleri (örneğin t1-i3-o3)
n0	0 yarıçapı içinde olan 1 uzunluğundaki bağlantıların örüntüleri ($n0=t1$)
n1	1 yarıçapı içinde olan 1 uzunluğundaki bağlantıların örüntüleri ($c1=n1$)
n2	2 yarıçapı içinde olan 1 uzunluğundaki bağlantıların örüntüleri
n3	3 yarıçapı içinde olan 1 uzunluğundaki bağlantıların örüntüleri



Şekil 4.2: Ünlü kullanıcılar ile düzenli kullanıcıları içeren ağ yapısı.

Şekil 4.2’de her bir düğüme bir harf verilmiş ve aralarında yönlenmiş bağlantı gösterilmiştir. A, B, C ünlü kullanıcılara ait düğümleri oluşturmaktadır (mesela Turkcell, Brad Pitt, Jennifer Lopez). X düğümünü değerlendirildiği farz edilsin. X düğümüne ait gelen bağlantıları Y ve D düğümleri oluşturmaktadır. Ancak D düğümünün ünlü kullanıcılarla arasında bir bağlantı olmadığı için herhangi bir örüntüsü olmayacaktır. Y düğümü için bütün örüntüleri $\{A, B, C, AB, AC, BC, ABC\}$ oluşturmaktadır. $\{AB\}$ ve $\{BA\}$ gibi aynı örüntüler gruplanmıştır. $p_{gelen}(N', z, L')$; $N' \subseteq N, L' \subseteq L$ ve z yarıçap değerini, belirtmektedir. Bu durumda X düğümüne gelen 1 uzunluğundaki örüntüler (bu aynı zamanda il türüne karşılık gelir) $p_{gelen}(X, 1, 1) = \{A, B, C\}$, 2 uzunluğundaki örüntüler $p_{gelen}(X, 1, 2) = \{AB, AC, BC\}$ ve 3 uzunluğundaki örüntüler $p_{gelen}(X, 1, 3) = \{ABC\}$ oluşturmaktadır. Tanım 3.4 kullanılarak $p(X, 1, A, gelen) = \{Y\}$ oluşturmaktadır. X düğümüne ait giden bağlantıyı Z düğümü oluşturmaktadır ve ünlü kullanıcılarla bağlantısı olduğu görülmektedir. Y ile Z’nin ünlü kullanıcılarla aynı bağlantıları olduğu için aynı örüntülere sahip olacaktır.

Örüntü keşfi ve sayma, ağ komşuluklarındaki düğümlerinin düzleştirilerek özellik vektörlerine çevrildiği adımdır. İlk olarak Twitter takipçi ağını temsil eden [Kwak et al., 2010] veri seti kullanılarak from_id ve to_id adlarında kolonları olan bir tablo (TWITTER_GRAPH) yaratılır. Önceden tanımlanmış bir eşik değerinin üzerindeki bağlantı sayısına sahip kullanıcılar bulunarak ünlü kullanıcıları olarak tanımlanır (CELEBRITY_NODES). Eşik değeri verideki sayısal değerlere bakılarak

karar verilmiştir. Sonrasında bir önceki adımda oluşturulan sayısal veriyi sıralayarak top-k ünlü düğümleri seçilir (k tane).

Ünlü düğümü ile özellik arasındaki fark ünlü düğümleri, meşhur insanlar (artist, şarkıcı, milletvekili... vs.) veya yüksek bağlantı sayısına sahip sıradan insanlar oluşabilirken özellikler meşhur insanları içermez. Bu nedenle özellikler, ünlü düğümlerinin bir alt kümesidir denilebilir. G ağından bulunan ünlü düğümleri çıkarılarak her kullanıcının, özellik bazında, gelen, giden veya bütün komşuları farklılaşarak toplam bağlantı sayısı bulunur. Bu sayı, destek (support) olarak isimlendirilir. Örneğin 1 uzunluğundaki giden komşularının tüm düğümlerinin türü “o1” olarak işaretlenir ve her bir kullanıcının her özelliği bazında giden bağlantı sayısı bulunur. Bulunan bu bilginin anlamı, her bir kullanıcı için 1 uzunluğundaki örüntüleri ve sayılarını oluşturulmuştur. 1 uzunluğundaki örüntüleri elde ettikten sonra bu bilgileri kullanarak her bir kullanıcı için 2 uzunluğundaki örüntü sayıları elde edilmektedir. 2 uzunluğundaki örüntüler şu şekilde ifade edilmektedir. 1 uzunluğundaki iki farklı örüntü $f_1, f_2 \in F$ ve 2 uzunluğundaki örüntüler alt kümesi $F' \subseteq F$ olsun. Her bir 1 uzunluğundaki örüntü değeri kullanılarak 2 uzunluğundaki örüntü kümesi $F' = f_1 + ' - ' + f_2$ permutasyonu ile elde edilmektedir. Şekil 4.1'deki ağ yapısını ele alırsak, 2 uzunluğundaki örüntü kümesi şu şekilde oluşturulacaktır: $F' = \{(A - B), (B - C), (A - C)\}$ olacaktır.

Bu tez çalışmasındaki işlemler Yapılandırılmış Sorgulama Dili (SQL) ile programlanmıştır. SQL, ağ yapısını düzleştirerek özellik vektörüne çevirirken geçerli ve ölçeklenebilir bir ortam sağlamıştır. Oluşturulan belli transformasyondan sonra, Twitter ağını her kaydın ağ düğümünü, o düğüme ait düzleştirilmiş özellik vektörünü ve o düğümün ağdaki bilinen etiketlerini temsil edecek şekilde kayıt kümesine dönüştürülmüştür. Bu çizelge halindeki temsil, tahmin etmek için ağ madenciliği algoritmalarından farklı olarak birçok klasik makine öğrenmesi algoritmasını kullanılabilmesini sağlamıştır.

Bu çalışmada kullanılabilecek birçok farklı güçlü sınıflandırma makine öğrenmesi algoritması bulunmaktadır. Performansları kıyaslanabilir olmasına rağmen, içlerinden biri diğerlerinden daha iyi çalışır. Bu çalışmada 4 farklı sınıflandırma algoritması düşünülmüştür. Destek Vektör Makineleri (SVM) [Milenova et al., 2005], Naïve Bayes (NB), Genelleştirilmiş Lineer Model (GLM), Karar Ağaçları (DT). Bu algoritmalar doğruluk (accuracy) ve lift değerlerine bakılarak karşılaştırılmıştır. Bütün algoritmalar Oracle Data Miner (ODM)

kütüphanesinde bulunmaktadır. Varsayılan ODM parametreleri iyi çalıştığından dolayı herhangi bir değişiklik yapılmadan kullanılmıştır. Bütün algoritmalar için, k-fold çapraz doğrulama (cross validation) kullanılmıştır.

5. DENEYSEL SONUÇLAR

Bu çalışmada sosyal ağlarda düğüm etkileşimleri üzerine birçok pratik deneysel çalışmalar yapılmıştır. Bu çalışmalarda en önemli göze çarpan sorun, deney ortamlarında karşılaşılan uzun süren işleme sürecidir. Bu nedenle yapılan hatalarda tekrar başa dönüp çalıştırmak veya farklı deneyler yapmak uzun süren bir işlem haline gelmiştir.

Bütün deneyler Oracle Exadata V2 makinesinde 64 bit Red Hat Linux 2.4 işletim sistemi ve Oracle 11g veri tabanı kullanılarak gerçekleştirilmiştir. İşlem Yapılandırılmış Sorgulama Dili (SQL) ve Usule Ait Yapılandırılmış Sorgu Dili (PLSQL) kullanılarak programlanmıştır.

Twitter sosyal ağ verisi [Kwak et al., 2010] kaynağından alınmıştır. Deneyin bir parçası olarak, 90000'den fazla bağlantısı olan düğümler ünlü düğümleri olarak değerlendirilmiştir. 90000 ile 100000 arasında bağlantı sayısı olan düğümler özellik olarak seçilmiştir. Bu şekilde olan 34 düğüm bulunmaktadır. Bu özellikleri kullanarak örüntü bağlantıları düğüm, özellik ve bağlantı olan örüntü sayısını içeren bir örüntü bağlantı tablosunda saklanmıştır. Ancak, sayıca çok fazla farklı örüntü sayısını içeren tabloyu işlemek zordur. Bu nedenle, sadece 10000'den fazla bağlantısı olan örüntüler seçilmiştir ve toplam 1224 örüntü kullanılmıştır (Tablo 5.1). Örüntüler numaralandırılmış ve her bir tür için destek sayısı (bağlantı sayısı) hakkında örüntü bilgileri olan arama (lookup) tablosu yaratılmıştır. Bu tablo kullanılarak her bir örüntü için özet başka bir tablo oluşturulmuştur. Bu çalışmanın amacının düğümlerin var olup olmadığını tahmin etmek olmasından dolayı, düğümler hedef oldukları zaman gerçek destek değerini kullanmak yerine ikili (boolean) destek (eğer destek 0'dan büyükse 1, değilse 0) kullanılmıştır. Böylece, makine öğrenmesi algoritmalarında kullanılacak ağ düğümlerindeki veri, sayısal vektörler olarak biçimlendirilmiştir. Bu işlem, bu çalışmanın önışlem (Extract-Transform-Loading) sürecidir.

Tablo 5.1: Örüntü türü bazında örüntü sayısı.

Örüntü Türü	Örüntü Sayısı
t1	34
i1	34
o1	34
i2	561
o2	561

Tahmin edilen etiketlerin kendileri arasındaki önemini bulmak için nitelik önemi (attribute importance) bulunmuştur. Böylece hangi etiketin diğerlerine göre daha fazla işe yaradığı ortaya çıkartılmıştır. Nitelik önemi için Minimum Tanım Uzunluğu (MDL) algoritması kullanılmıştır.

5.1. Makine Öğrenmesi Algoritmalarının Seçilmesi

Öncelikle deneylerde hangi makine öğrenmesi algoritmasının kullanılacağına karar vermek üzere deneyler yapılmıştır. Deneyleri yapabilmek için ağ düğümlerinden sayısal vektör halinde etiket (veya sayısı) oluşturma yoluyla veri seti oluşturulmuştur. Örneğin, {A, B, C, D} bütün veri setinde bulunan 4 etiket olduğunu ve düğüm v'ye {A, C} atandığını varsayalım. Bu etiketlerden v için {A:1, B:0, C:1, D:0} olan özellik vektörü oluşturulur. Sonrasında her bir etiket için, etiketleri hedef seçmek yoluyla öğrenme seti oluşturulur ve giriş setinden hedef değer çıkartılır. Örneğin, v düğümünün A etiketi olup olmadığını anlayarak tahmin edebilmek için, A çıkarılarak onun özellik vektörü {B:0, C:1, D:0} olarak dönüştürülür. Sonrasında, ağdaki her etiket için veri seti oluşturulur ve 10-fold çapraz doğrulama metodu ile test edilerek tahmin doğruluğu ölçülür.

Doğruluk, tahmin sonucunda bulunan değerle testte bulunan değer aynı olup olmadığına göre yapılan bir testtir. Toplam doğru ve yanlış tahmin sayısına göre yüzde bir sonuç elde edilir. Doğru sayısı ne kadar çoksa tahmin değeri o kadar yükselecektir. Lift değeri ise, bütün veri setine göre yapılan tahminin durumu anlaşılmasına çalışılır. Rastgele yapılan bir tahminle, algoritmadaki modelle oluşturulan tahmin arasındaki eğriye bakılarak ne kadar iyi bir tahmin elde edildiği anlaşılmasına çalışılır. Yani modelin sağladığı faydanın değerlendirilmesinde

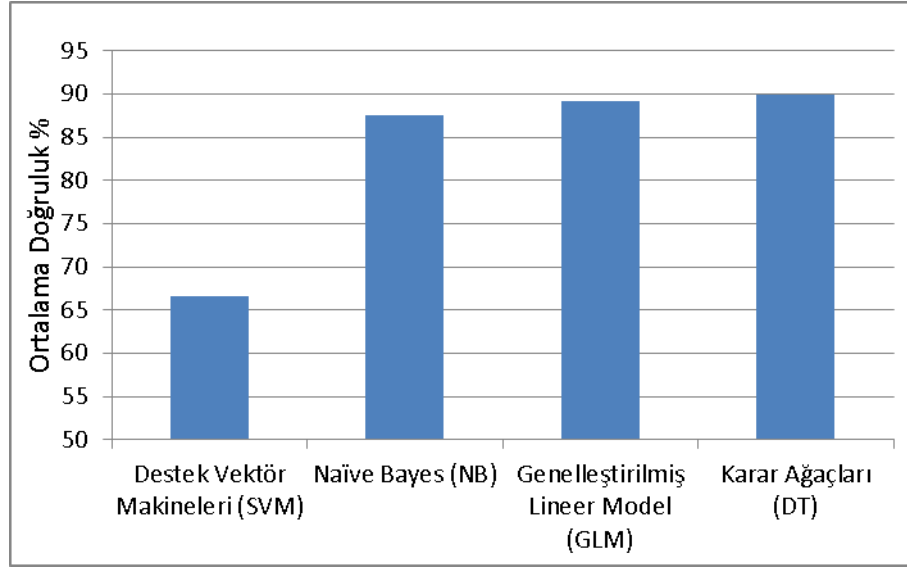
kullanılır. Lift kullanılması nereden tüm modelde yapılan tahminin kolayca yorumlanabilmesini sağlamaktır Lift eğrisi ROC eğrisine benzetilebilir. Quantile denilen rastgele bir değişkenin kümülatif dağılım fonksiyonu (CDF) kullanılarak düzenli aralıklarına ayrılan eğri üzerinden yorumlanır. Her quantile bazında eşik orana bakılarak modelin tüm değerler için karşılaştırılması yapılmış olur. Örneğin A ve B şeklinde iki giriş değişkeni ve bu giriş değişkenlerinin 0 ve 1 şeklinde hedef değişkeni olduğu düşünülürse A giriş değişkeni için 0 olma durumu için lifti şu şekilde formülize edilebilir:

$$Lift(A | 0) = \frac{P(A'nın 0 olma olasılığı)}{P(0 olma olasılığı)} = \frac{P(0 | A)}{P(0)} = \frac{P(A \cap 0)}{P(A) P(0)} \quad (5.1)$$

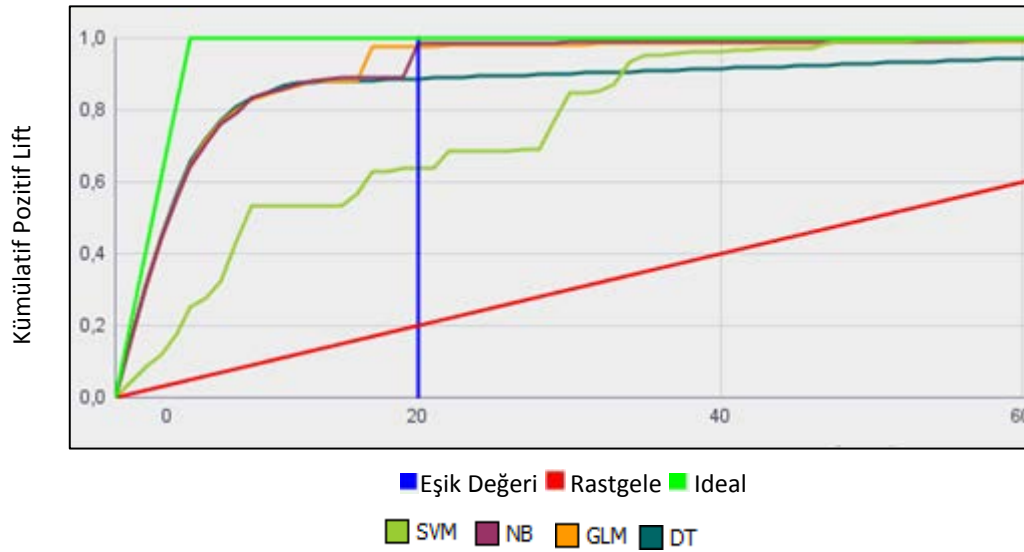
İlk önce düğümlerin kendisindeki örüntüleri, “t1” örüntü bağlantılarının oluşturduğu veri seti kullanılarak deneylere başlanır. SVM, NB, GLM ve DT algoritmaları için bu veri setindeki seçilen herhangi bazı örüntüler için tahmin edilmeye çalışılmıştır. Elde edilen tahminler toplam ortalama bazında hesaplanarak doğruluk metrikleri olan doğruluk (accuracy) ve lift değeri hesaplanır. Bu deneyin sonuçları tablo 5.2, şekil 5.1 ve 5.2’de verilmiştir.

Tablo 5.2: Makine öğrenmesi algoritmaları için doğruluk ve lift kullanarak yapılan karşılaştırma tablosu.

<i>Algoritma</i>	<i>Ortalama Doğruluk %</i>	<i>Ortalama Pozitif Doğruluk %</i>	<i>Ortalama Negatif Doğruluk %</i>	<i>Kümülatif Lift</i>
Destek Vektör Makineleri (SVM)	66.68	99.89	7.66	4.06
Naïve Bayes (NB)	87.49	98.81	51.44	4.92
Genelleştirilmiş Lineer Model (GLM)	89.18	99.07	45.43	4.88
Karar Ağaçları (DT)	89.91	99.26	38.39	4.43



Şekil 5.1: Ortalama doğruluk kullanılarak makine öğrenmesi algoritmaları karşılaştırması.



Şekil 5.2: Lift eğrisi kullanılarak makine öğrenmesi algoritmaları karşılaştırması.

Genelleştirilmiş Lineer Model'in (GLM) ve karar ağaçları algoritmasının diğer iki algoritmaya (Destek Vektör Makineleri ve Naïve Bayes) göre daha yüksek ortalama doğrulukları olduğu görüldü. Buna ek olarak, GLM için daha iyi bir lift eğrisi olması nedeniyle yani daha hızlı bir şekilde iyi bir sonuca eriştiği görülmüştür.

Sonraki deneylerde veri seti değiştirilerek "t1-i1" ve "t1-o1" veri setleri kullanılmış ve düğümün kendisine ait, t1'e ait örüntüler tahmin edilerek hesaplama

yapılmıştır. Yani düğümlerin kendisindeki örüntülerin yanında gelen bağlantılara ait örüntüler veya giden bağlantılara ait örüntüler eklenerek benzer yöntemle test edilmiştir.

Tablo 5.3: Farklı veri setlerinde makine öğrenmesi algoritmaları için ortalama doğruluk değerleri (%).

<i>Algoritma</i>	<i>t1</i>	<i>t1-i1</i>	<i>t1-o1</i>	<i>c1</i>	<i>c2</i>
Destek Vektör Makineleri (SVM)	66.68	49.57	65.67	56.88	56.89
Naïve Bayes (NB)	87.49	64.50	72.17	73.77	55.24
Genelleştirilmiş Lineer Model (GLM)	89.18	66.33	75.99	50	50
Karar Ağaçları (DT)	89.91	73.84	85.70	82.26	76.45

Tablo 5.4: Farklı veri setlerinde makine öğrenmesi algoritmaları için kümülatif lift değerleri.

<i>Algoritma</i>	<i>t1</i>	<i>t1-i1</i>	<i>t1-o1</i>	<i>c1</i>	<i>c2</i>
Destek Vektör Makineleri (SVM)	4.06	1.0034	1.0049	1.05	1.05
Naïve Bayes (NB)	4.92	1.0027	1.0042	1.04	1.03
Genelleştirilmiş Lineer Model (GLM)	4.88	1.0031	1.0048	1	1
Karar Ağaçları (DT)	4.43	1.004	1.0054	1.05	1.05

Tablo 5.3 ve 5.4 değerlere bakarak genel olarak karar ağaçları algoritmasının genelleştirilmiş lineer modelden daha iyi bir doğruluk ve lift değeri verdiği görülebiliyor. Ancak tablo 5.5 dikkat çekici bir nokta t1 haricindeki veri setlerinde lift değerinin 1'e yakınsadığı görülmektedir. Bu nedenle hedef değişkenindeki 0 veya 1 değerinin dengeli (balanced) olmamasından kaynaklanabileceği düşüncesi ile farklı bir deney daha yapılmıştır. Örneğin c1 örüntü bağlantılarının olduğu veri setinde hedef değişkene ait 0'ların sayısı 1805082 iken 1'lerin sayısı 83619'dur. Bu nedenle

bir örnekleme yapılarak hedef değişkenin 0'ların toplam sayısı ile 1'lerin toplam sayısını birbirine orantılı yapılmıştır (örneğin 0'lar 1'lerin 2 katıdır).

Tablo 5.5: Dengeli hedef değişkeninde makine öğrenmesi algoritmaları için ortalama doğruluk değerleri (%).

<i>Algoritma</i>	<i>c1</i> <i>Eski</i>	<i>c1</i> <i>Yeni</i>	<i>c2</i> <i>Eski</i>	<i>c2</i> <i>Yeni</i>
Destek Vektör Makineleri (SVM)	56.88	73.70	56.89	73.53
Naïve Bayes (NB)	73.77	74.11	55.24	55.31
Genelleştirilmiş Lineer Model (GLM)	50	50	50	50
Karar Ağaçları (DT)	82.26	82.03	76.45	76.48

Tablo 5.6: Dengeli hedef değişkeninde makine öğrenmesi algoritmaları için kümülatif lift değerleri.

<i>Algoritma</i>	<i>c1</i> <i>Eski</i>	<i>c1</i> <i>Yeni</i>	<i>c2</i> <i>Eski</i>	<i>c2</i> <i>Yeni</i>
Destek Vektör Makineleri (SVM)	1.05	1.31	1.05	1.30
Naïve Bayes (NB)	1.04	1.25	1.03	1.191
Genelleştirilmiş Lineer Model (GLM)	1	1	1	1
Karar Ağaçları (DT)	1.05	1.32	1.05	1.32

Deney sonucunda oluşturulan Tablo 5.5 ve 5.6 bakıldığında destek vektör makineleri algoritmasında hem doğruluk oranında hem de lift değerinde ciddi bir artışa neden olduğu görülmektedir. Ancak her ne kadar hissedilebilir bir artış olsa da karar ağaçları algoritması daha iyi sonuç verdiği görülmektedir. Bu nedenle karar ağaçları algoritması kullanılmasına karar verilmiştir.

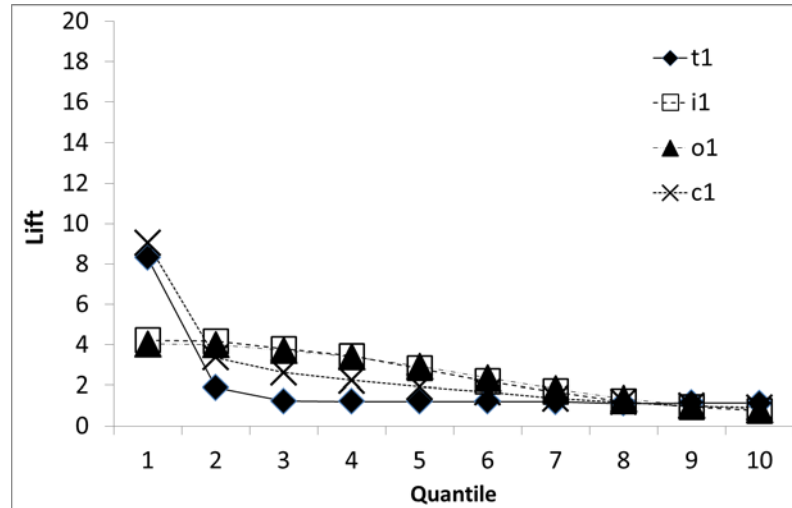
Ayrıca Tablo 5.6 görüldüğü gibi veri seti dengelenmesine ve değişmesine rağmen GLM algoritması lift değerinin sürekli olarak 1 olduğu görülmektedir. Bunun

sebebi GLM algoritmasının bütün veri setine aynı değerin geleceğini düşünerek model oluşturmasıdır.

Bir önceki deneylerde yapılan GLM algoritmasının başarısının yüksek olmasının nedeni özellik vektörü arttıkça GLM algoritmasının başarısızlığının arttığı ve veriyi tahmin edecek fonksiyonu oluşturmak yani lineerleştirmek için başarısız sonuçlara neden olabilecek bir model oluşturması olarak yorumlanabilir. Bu nedenle de veri setini değiştirip dengelememize rağmen karar ağaçları algoritması daha başarılı olmuştur. Bunun bir diğer sonucu olarak GLM algoritmasının lift değerinin 1 kalmasına sebep olmuştur.

5.2. Örüntü Türüne Göre Tahmin Karşılaştırması

Bu deneyde, bir düğümün bilinen etiketleri (vektör $t1$ denilirse), gelen bağlantıları üzerinden bağlı komşularının etiketleri (vektör $i1$ denilirse), giden bağlantıları üzerinden bağlı komşularının etiketleri (vektör $o1$ denilirse) için ne kadar tahmin sonucunu iyileştirdiği gözlemlenmek istenmiştir. Sonuçta 4 ayrı veri seti oluşturulmuş ve tahmin performansları karşılaştırılmıştır: $t1$, $i1$, $o1$ ve $c1 = t1 + i1 + o1$, '+' birleştirme operatörünü ifade etmektedir.



Şekil 5.3: Etiket yerinin (düğümün kendisinin, gelen veya giden komşuları) tahmin kapasitesi üzerine etkisi.

Gelen bağlantıların giden bağlantılara göre daha fazla tahmin bilgisi sağladığı görülmüştür. İlk quantile’da, $t1$, $i1$, $o1$ (örneğin veri kümesi $c1$) arttırmada $c1$, $i1$ ’e

göre sadece %53; c_1 , o_1 'e göre %55 iyileşme sağladığı hesaplanmıştır. Buna ek olarak komşuluk bilgisinin yani c_1 'in t_1 'e oranla tahmin performansını %8 arttırmıştır. Bu sonucun anlamı, Twitter'da kullanıcıların gelen, giden ve kendi bilgileri beraber kullanıldığında sadece kendi bilgileri kullanılarak yapılan tahmine göre daha iyi bir bağlantı tahmini yapılabileceğini göstermiştir.

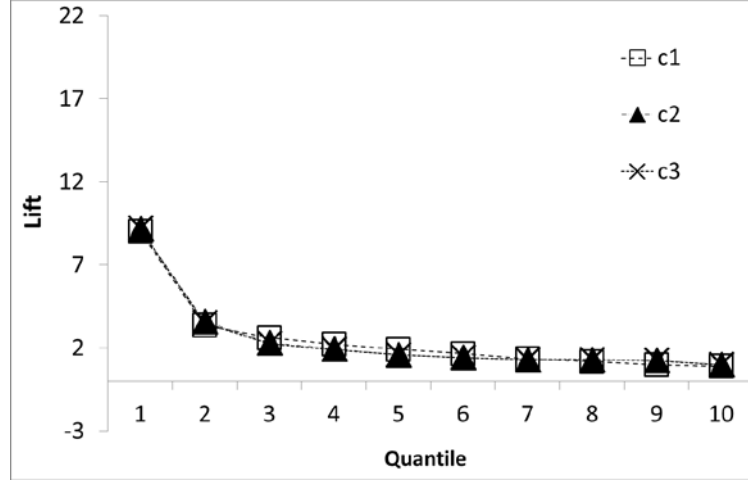
5.3. Örüntü Karmaşıklığını Arttırmanın Tahmine Etkisi

Bu deneyde, doğrudan komşuluk düğümlerindeki (örneğin düğümün kendisinden gelen ve giden bağlantıları) etiketlerinden özellikler oluşturuldu. Doğrudan komşulukların etiket bilgileri 3 farklı vektöre ayrılmıştır: t_1 , i_1 , o_1 . Örüntü uzunluklarının tahmin kapasitesindeki etkisini hesaplayabilmek için farklı öğrenme setleri dikkate alınmıştır: $c_1 = t_1 + i_1 + o_1$, $c_2 = c_1 + t_2 + i_2 + o_2$, $c_3 = c_2 + t_3 + i_3 + o_3$.

Aynı uzunluktaki örüntülere ait bütün özellik vektörleri beraber oluşturulmuştur. Öğrenme setindeki t_1 , t_2 ve t_3 ait vektörlerden hedef etiketi içeren örüntüler çıkarılarak sadece t_1 'e ait örüntüler tahmin edilmeye çalışılmış ve bu değerler üzerinden doğruluk oranı ortalaması alınmıştır. Örneğin, eğer A etiketi tahmin edilmeye çalışılıyorsa ve t_2 düğümünün vektörü {AB, BC, AC} örüntülerine sahipse t_2 'den {AB, AC} örüntüleri çıkarılır. Çünkü bu tür örüntüler A varlığına işaret etmektedir.

c_1 , c_2 , c_3 tahminine ait veri setleri Şekil 5.4'de gösterilmiştir. 2 uzunluğunda örüntülere ait özellik vektörleri eklendiğinde %1,5 iyileşme sağlanmıştır ve 3 uzunluğunda örüntülere ait özellik vektörleri eklendiğinde (İlk quantile'daki lift değerlerine göre) %1 iyileşme sağlanmıştır.

Sonuç olarak, özellik vektörlerinde daha karmaşık örüntüler tahmin kapasitesini arttırmaktadır. Ancak karmaşıklık artarken bu tür iyileşmeler daha az önemli olur. Twitter'da kullanıcıların seçilen farklı kullanıcılarla etkileşimi bulunarak bağlantı tahmini yapıldığında bağlantı tahmini sonucunu iyileştirdiği görülmüştür. Ancak seçilen kullanıcı sayısı ile karmaşıklık arttığı zaman iyileşmenin daha az karmaşıklıkla yapılan iyileşmelere göre azaldığı gözlemlenmiştir.

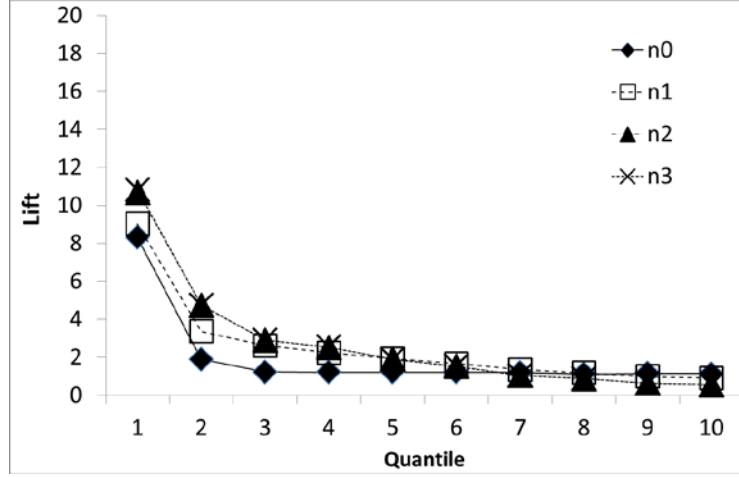


Şekil 5.4: Örüntü karmaşıklığının tahmin sonucu üzerine etkisi.

5.4. Dolaylı Komşuluktan Gelen Tahmin Bilgisi

Bu deneyde, daha büyük yarıçaptaki ağ komşuluklarından gelen veriyi eklemenin tahmin kapasitesine iyileştirme yapıp yapmayacağı incelenmiştir. İlk olarak sadece düğüm v 'ye ait (n_0 komşuluğunda) etiketleri kullanarak özellik vektörü oluşturuldu. Sonrasında doğrudan v 'ye ait komşularından (hem gelen hem de giden komşularına sahip n_1 komşulukları) gelen ek etiketlerle test edildi. Daha sonra n_2 ve n_3 yarıçapları arttırılmıştır. Ağ komşuluğu etiketlerini tek bir vektöre dönüştürmek için, her yarıçaptaki özellik vektörleri birleştirilmiştir. Örneğin, n_3 anlamı, düğüm v 'nin kendi etiketleri (örneğin n_0), v 'nin doğrudan komşularına ait vektörlerinin etiketleri ile (örneğin r_1 olarak isimlendirilirse), v 'nin 2 ve 3 uzaklığındaki dolaylı komşuları ile birleştirilir (r_2 ve r_3 olarak isimlendirilirse): $n_i = n_0 + r_1 + \dots + r_i$

Ek olarak, bu deney için 1 uzunluğundaki örüntü etiketleri kullanılmıştır. Bu deneyde, bir düğümün etiketleri ve doğrudan komşularına ait etiketler tahmin konusunda en fazla bilgiyi vermektedir. 2 ve daha büyük yarıçaplar hesaplama çabasına değer tahmin kapasitesinde ek bilgi getirmemektedir (Şekil 5.5).



Şekil 5.5: Ağ komşulukları yarıçapının tahmin sonucu üzerine etkisi.

İlk quantile'daki n_3 'ün n_2 'ye göre lift iyileşmesi %1.1 ve n_2 'nin n_1 'e göre %15,5'dir. İkinci quantile için, n_3 'ün n_2 'ye göre iyileşmesi %1, n_2 'nin n_1 'e göre %28,8'dir. Bu deney, Twitter'da arkadaşlarının arkadaşları (n_2) bilgileri kullanılarak yapılan tahminin arkadaşlarının (n_1) bilgileri kullanılarak yapılan tahminden daha başarılı olduğu sonucunu vermiştir. Ancak arkadaşlarının arkadaşlarının arkadaşları (n_3) gibi kullanıcının kendisinden uzaklaştıkça karmaşıklıkla dolaylı işlem zamanı artıyor ve işlem zamanına göre kazanılan başarı artışı değer sağlanamadığı görülmüştür.

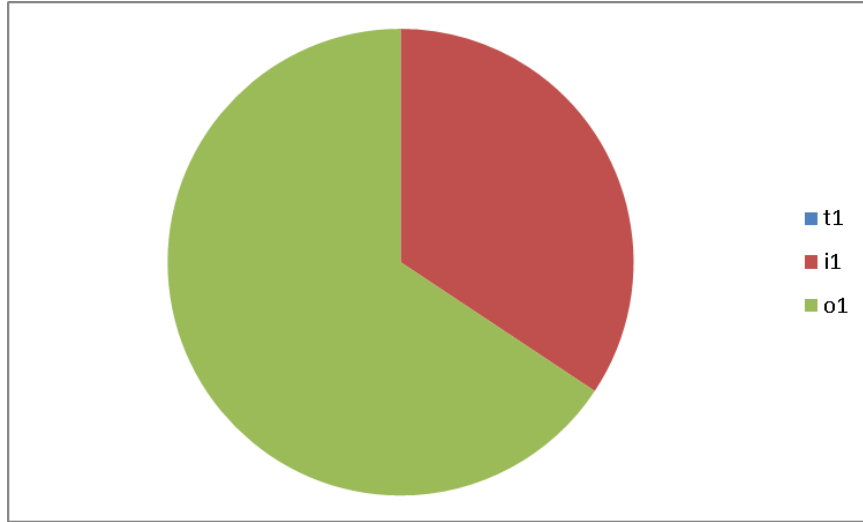
5.5. Örüntü Türünün Birbiri Üzerindeki Etkisi

Bu deneyde, ünlü düğümlerinin farklı bağlantı türleri için birbiri üzerine etkisi araştırılmıştır. Yani örüntü türlerinden hangilerinin en değerli bilgiyi sağladığı bulunmaya çalışılmıştır. Öncelikle her bir düğümün bütün bağlantı türleri için etiketler oluşturulmuştur. Sonrasında, her bir etiket için, etiketleri hedef değer olarak seçilip tahmin esnasında giriş setinden hedef değerlerden biri çıkartılarak diğer değerlere göre nitelik önemi bulunur.

34 farklı ünlü düğümünde 3 farklı örüntü türü (örneğin t_1 , i_1 , o_1) için bu değerler hesaplanmıştır. Bütün veri seti için 1 uzunluktaki örüntülerden en çok giden düğümlere ait örüntülerin (o_1) oluşturduğu örüntülerin önemli olduğu görülmüştür. 67 farklı giden bağlantı düğümüne (o_1) ait etiket, 35 farklı gelen bağlantı düğümüne (i_1) ait etiket önemli görülmüştür (Tablo 5.7, Şekil 5.6).

Tablo 5.7: Ağ bağlantı türünün birbiri üzerine etkisi tablo gösterim.

Örüntü Türü	Etki Oranı
t1	%0
i1	%34
o1	%66



Şekil 5.6: Ağ bağlantı türünün birbiri üzerine etkisi.

5.6. Ünlü Düğümlerinin Kendi Örüntüleri Üzerindeki Nitelik Öneminin Tespit Edilmesi

Bu deneyde, tek bir ünlü düğümü için bağlantı türlerinin (i1, o1, i2, o2) tahmin kabiliyeti üzerine etkisini incelenmektedir. MDL algoritması kullanılmaktadır. İlk olarak tek bir ünlü düğümü seçilir. Sonrasında seçilen bu ünlü düğümüne ait bütün özellik vektörleri tahmin edilecek veri seti üzerinde bulunur. Bir sonraki adım olarak her bir etiket için, etiketleri hedef seçmek yoluyla ve giriş setinden bulunan bütün özellik vektörleri ve hedef değer çıkartılarak nitelik önemi bulunur (Tablo 5.8).

Tablo 5.8: 10 ünlü düğümü için kendi örüntüleri üzerindeki bağlantı türünün nitelik önemi (%).

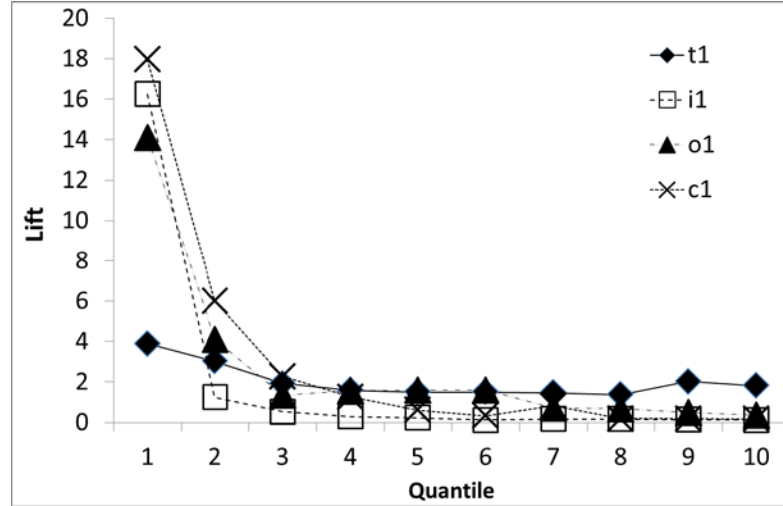
Ünlü Düğümü	<i>t1</i> Nitelik Önemi (%)	<i>i1</i> Nitelik Önemi (%)	<i>i2</i> Nitelik Önemi (%)	<i>o1</i> Nitelik Önemi (%)	<i>o2</i> Nitelik Önemi (%)	Ort.	Maks.	Min.
804455	0,0211	0,0923	0,0908	0,1422	0,1435	0,09798	0,1435	0,0211
16150075	0,021	0,0891	0,0875	0,1387	0,1404	0,09534	0,1404	0,021
5656402	0,02	0,089	0,087	0,1387	0,1398	0,0949	0,1398	0,02
16899075	0,0195	0,0835	0,0795	0,1314	0,1303	0,08884	0,1314	0,0195
15770446	0,0183	0,0891	0,0882	0,1369	0,139	0,0943	0,139	0,0183
8071902	0,0179	0,0916	0,0887	0,1378	0,1381	0,09482	0,1381	0,0179
18238282	0,0177	0,0911	0,0916	0,1398	0,1431	0,09666	0,1431	0,0177
14430060	0,017	0,0897	0,0903	0,1366	0,1397	0,09466	0,1397	0,017
12589972	0,0158	0,0874	0,0889	0,1304	0,134	0,0913	0,134	0,0158
23748699	0,004	0,0063	0,0127	0	0	0,0046	0,0127	0

Yapılan bu deneyde, 1 uzunluğunda gelen ve giden (sırasıyla i1, o1) örüntüler ile 2 uzunluğunda gelen ve giden örüntüler (i2, o2) nitelik önemi bulunup ortalaması karşılaştırılmıştır. Tablo 5.8 kullanıldığında görülüyor ki düğümlerin neredeyse tamamına yakını düğümün kendisi için en değerli bilgiyi giden bağlantıların oluşturduğu örüntüler (o2) oluşturmaktadır. Sadece tek bir düğümde i2 en önemli oranı vermiştir. Bu sonuç mantıklıdır çünkü kişiler etrafındaki arkadaşlardan gelen bilgiyi en fazla değerlendirir.

5.7. Genelleştirilmiş Lineer Model Kullanılarak Deneylerin Tekrarlanması

5.2, 5.3 ve 5.4 bölümlerinde açıklanan “örüntü türüne göre tahmin karşılaştırması”, “örüntü karmaşıklığını arttırmanın tahmine etkisi”, “dolaylı komşuluktan gelen tahmin bilgisi” deneyleri, genelleştirilmiş lineer model algoritması kullanılarak tekrarlanmıştır.

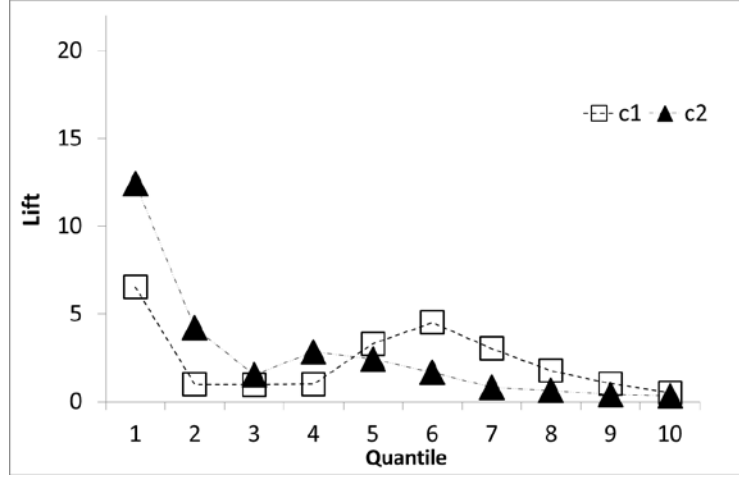
“Örüntü türüne göre tahmin karşılaştırması” deneyinde karar ağaçları algoritmasındaki gibi gelen bağlantıların giden bağlantılara göre daha fazla tahmin bilgisi sağladığı görülmüştür. İlk quantile’da, t1, i1, o1 (örneğin veri kümesi c1) arttırmada c1, i1’e göre sadece %11; c1, o1’e göre %27 iyileşme sağladığı hesaplanmıştır. Buna ek olarak komşuluk bilgisinin yani c1’in t1’e oranla tahmin performansını %362 arttırmıştır.



Şekil 5.7: GLM algoritması kullanılarak etiket yerinin tahmin üzerine etkisi.

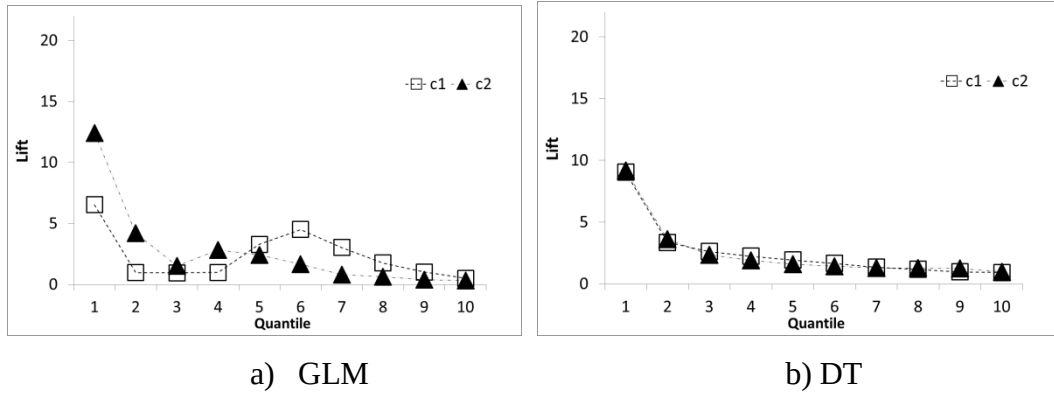
Bu deneyin sonucu olarak GLM algoritması ile de benzer sonuçlar elde edildiği görülmektedir. GLM algoritmasının karar ağaçları algoritmasından farklı olarak c1’in diğer örüntü türlerine (t1, i1, o1) oranla tahmin performansı daha fazla arttırmıştır. Bunun nedeni t1, i1, o1 örüntü türlerinin tahmin performansının GLM algoritmasında karar ağaçları algoritmasına göre daha düşüktür. Bu nedenle daha fazla iyileşme oluşmuştur.

“Örüntü karmaşıklığını arttırmanın tahmine etkisi” deneyinde c1, c2, c3 tahminine ait veri setleri şekil 5.8’de gösterilmiştir. 2 uzunluğunda örüntülere ait özellik vektörleri eklendiğinde %8 iyileşme sağlanmıştır ve 3 uzunluğunda örüntülere ait özellik vektörleri eklendiğinde (İlk quantile’daki lift değerlerine göre) %3.5 iyileşme sağlanmıştır.



Şekil 5.8: GLM algoritması kullanılarak örüntü karmaşıklığının tahmin üzerine etkisi.

DT algoritmasından farklı olarak GLM algoritmasında, 2 uzunluğundaki örüntüler kullanılarak yapılan c2 veri seti deneyinde %47 bir iyileşme elde edilmiştir. Bunun nedeni DT algoritması, c1 veri setinde GLM algoritmasına göre çok daha iyi bir doğruluk sonucu belirlediği için c2 veri kümesinde sağlanabilecek iyileşme çok limitli kalmıştır. Ancak aslında tahmin değeri zaten yüksek olması nedeniyle DT algoritması hem c1 veri kümesinde hem de c2 veri kümesinde 10 quantile içerisinde ilk quantile için %9.2 gibi yüksek bir sonuç çıkartmıştır.

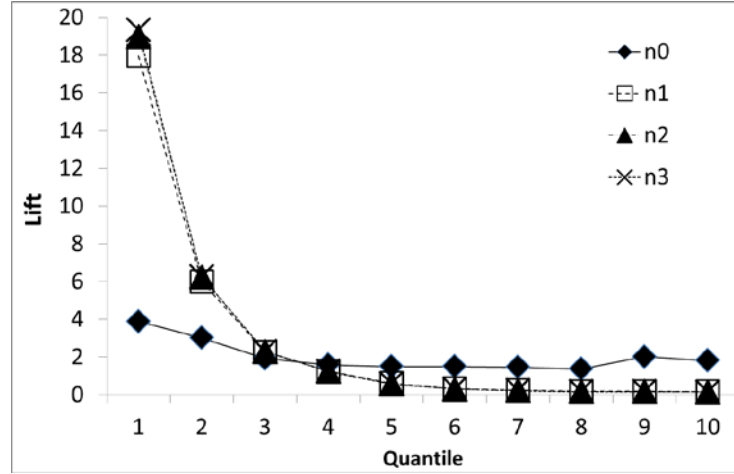


Şekil 5.9: Örüntü karmaşıklığını arttırmanın tahmine etkisi deneyi için a) GLM kullanılarak lift sonucu b) DT algoritması kullanılarak lift sonucu.

Şekil 5.9’de GLM ve DT algoritmalarının lift değerlerinin grafiksel karşılaştırılması yapılmıştır. DT algoritması GLM algoritmasına göre daha düzenli bir eğri izlediği görülmektedir. Bu da DT algoritmasının bütün veri içerisinde daha

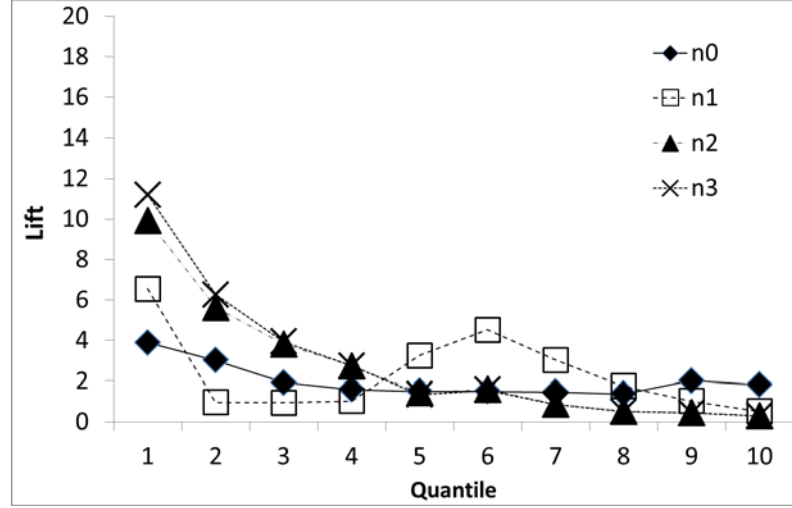
başarılı olduğunu göstermektedir. İlk quantile bakıldığı takdirde c1 veri kümesine ait lift değerinin DT algoritmasında çok daha iyileştiği görülmektedir.

“Dolaylı komşuluktan gelen tahmin bilgisi” deneyinde bütün örüntüler için tahmin edildiğinde ilk quantile’daki n_3 ’ün n_2 ’ye göre lift iyileşmesi %1.7 ve n_2 ’nin n_1 ’e göre %5.8’dir. İkinci quantile için, n_3 ’ün n_2 ’ye göre iyileşmesi %1, n_2 ’nin n_1 ’e göre %1.04’dür.

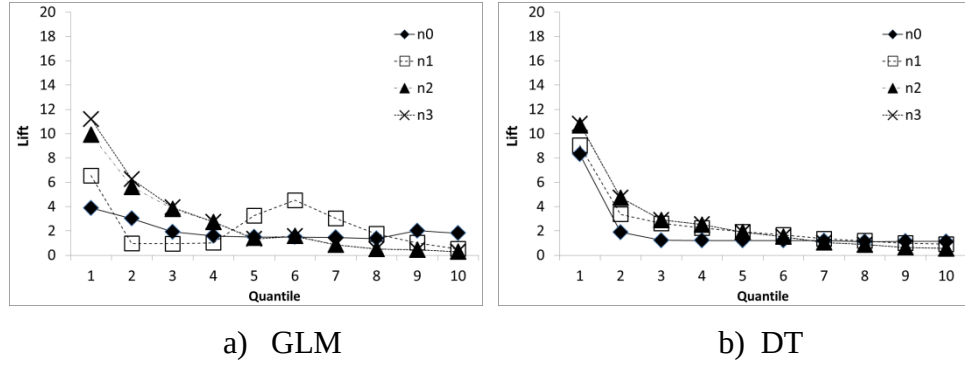


Şekil 5.10: Ağ komşulukları yarıçapının bütün örüntüler için tahmin sonucu üzerine etkisi.

Bu deneyi GLM algoritmasını kullanarak sadece t1 örüntülerini tahmin edecek şekilde yeniden yaparsak sonuç Şekil 5.11’deki gibi çıkmaktadır. Karar ağaçlarındaki gibi yarıçap arttıkça tahmin sonucunda iyileşme görülmektedir. Ancak karar ağaçları algoritmasından farklı olarak genel dağılımın düzenli olmadığı görülmektedir. Bunun anlamı tüm veri içerisinde tahmin sonucu anlamında iyi olmadığı anlaşılmaktadır (Şekil 5.12)



Şekil 5.11: Ağ komşulukları yarıçapının t1 örüntüleri için tahmin sonucu üzerine etkisi.



Şekil 5.12: Dolaylı komşuluktan gelen tahmin bilgisinin a) GLM kullanılarak lift etkisi b) DT kullanılarak lift etkisi.

DT algoritmasının avantajlı yanı bölüm 3.4.4 bahsedildiği gibi kurallar serisi oluşturması ve bu kuralları daha sonrasında görüntüleyebilmesidir. Örneğin 15770446 numaralı ünlü düğümü için c1 veri kümesinde oluşturulan modeldeki karar ağacının bir yaprağını inceleyelim. Kullanıcı numarası ile Twitter API'dan sorgulanarak kullanıcı adına ulaşırsa MikeKlingler kullanıcı adı ile “Online Business School” kurucusu olduğu bulunabilir. Bu ünlü düğümü takip etmek için oluşturulan kural şöyle olacaktır:

Eğer test edilen düğüm (15770446-@MikeKlingler) ile 16150075 numaralı (@MichaelMillman, pazarlama strateji uzmanı) düğüm üzerinden giden toplam bağlantı örüntü (o1) sayısı $> 10,5$ ve 14430060 numaralı (@shawnrobinson, SolutionStream.com iş geliştirme uzmanı) düğüm üzerinden giden toplam bağlantı

örüntü (o1) sayısı > 317,5 ve 16150075 numaralı (@MichaelMillman, pazarlama strateji uzmanı) düğümle toplam bağlantı örüntü sayısı $\leq 0,5$ ise 1 olarak yani test edilen düğümle ünlü düğümü arasında bağlantı kurulabilir olarak tahmin edilir. Bu yaprağın doğru etme tahmin başarı oranı %86.87'dir.

5.8. Benzer Çalışmalar ile Başarı Karşılaştırması

Benzer çalışmalarla sonuç karşılaştırılması yapıldığında [Hasan et al., 2006] çalışmasında başarı oranı DBLP veri kümesinde %83.13, Biobase veri kümesinde %90.56 çıkmıştır. Bu tez çalışmasında tahmin başarı oranı, 1 yarıçapı içerisindeki 0 uzunluğundaki örüntüler (t1) kullanılarak %89.91, 1 yarıçapı içerisindeki 1 uzunluğun gelen, giden ve kendisi dâhil örüntüler (c1) kullanıldığında %82.26, 2 yarıçapı içerisindeki 1 uzunluğun gelen, giden ve kendisi dahil örüntüler (c2) kullanıldığında %76.45 çıkmıştır.

[Taskar et al., 2003] yaptığı çalışmada %4.9 doğruluk iyileştirmesi ortaya çıkartmıştır. Bu tez çalışmasında sadece 1 yarıçapı içerisindeki 0 uzunluğundaki örüntüler (t1) kullanıldığında %4.43, 1 yarıçapı içerisindeki 1 uzunluğun gelen, giden ve kendisi dâhil örüntüler (c1) kullanıldığında %1.05, 2 yarıçapı içerisindeki 1 uzunluğun gelen, giden ve kendisi dâhil örüntüler (c2) kullanıldığında %1.05 doğruluk iyileştirmesi çıkmıştır.

Sonuçlara bakıldığında doğruluk iyileştirmesi ve ortalama başarı oranı olarak benzer diğer çalışmalarla benzer sonuçlar ortaya çıkarttığı görülmektedir.

6. SONUÇLAR

Bu çalışmada, ağ bağlantılarını ağ etiketlerine dönüştürmek ve eksik bağlantı ve etiketleri tahmin etmek için yeni bir metot sunulmuştur. Sosyal ağ yapısı olarak Twitter veri kümesini hem literatürde kabul görmüş kullanıma açık olması hem de veri büyüklüğü olarak gerçeğe yakın düğüm ve bağlantı sayısına sahip olması nedeniyle tercih edilmiştir. Bu tez çalışmasının ana katkılarından biri ağ yapılarını vektör kümelerine dönüştürerek ölçeklenebilir ortamlar için işlenebilir hale getirmektedir. Yapılan deneyler sonucunda ağ yapıları başarıyla vektör kümelerine dönüştürülebilmektedir.

Diğer bir katkısı özellik vektörlerini paralel veri tabanı ortamında işlemek için ölçeklenebilir bir yöntem sunulmasıdır. Bu yöntemin de başarı ile gerçekleştirildiği ortaya çıkmıştır. Ancak veri tabanının kendi getirdiği kısıtlardan dolayı bazı noktalarda müdahale yapmak gerekmiştir. Bu kısıtlardan bazıları veri tabanındaki bir tabloda bulunabilecek maksimum kolon sayısı 1000, maksimum karakter sayısı 4000 oluşturmaktadır.

Son katkı olarak bağlantı tahmin problemini etiket tahmini problemine dönüştürülmüştür. Daha önceden tartıştığımız gibi ağ yapılarını vektör setlerine dönüştürmek için ağ bağlantı tahmini probleminden ağ etiket problemine dönüştürülmesi gerekiyordu. Düşünüldüğü gibi ağ etiket problemine dönüştürülerek 1 yarıçapındaki 1, 2, 3 uzunluğundaki gelen, giden ve o düğümün bağlantılarına ait örüntüler bulunmuştur.

Genel olarak yapılan deneylerden sosyal bir ağda eksik veya tamamlanmamış bağlantıların farklı sınıflandırma algoritmaları kullanılarak bu metot ile uzunluk arttıkça tahmin kapasitesinin arttığı görülmüştür. Ancak işlem karmaşıklığına oranla değer bir artış sağlamadığı da fark edilmiştir. Basit bir yaklaşım olan arkadaşımın arkadaşını bulma yöntemine göre daha başarılı olduğu da görülmüştür. Ölçeklenebilir yapısı sayesinde işlemleri hızlı bir şekilde tamamlayabilmektedir.

Deneylerden ilkinde görüldüğü gelen ve giden düğümleri özellik değişkenleri olarak alındığında sınıflandırmada daha başarılı sonuçlar elde edilmiştir. Bunun nedeni özellikle Twitter'da kullanıcıların diğer kullanıcılardan sıklıkla etkilenmesidir. Twitter'da belki arkadaşımız olmayan bir kişinin paylaşımını kullanıcı, arkadaşısı vasıtası ile görüp retweet'lediği takdirde asıl gönderen kullanıcıyı

tanınmasa bile onunla bir ortak noktası olduğunu göstermektedir. Böylelikle kullanıcının çevresinin de önemli bilgiler içerdiği görülmüş olur.

Diğer deneylerde birden fazla kullanıcının ortak noktaları bulunarak yani sınıflandırma yapılarak daha fazla bilgi olup olmadığına bakılmıştır. Yani iki farklı kullanıcının çevresindeki özellikler beraber araştırılarak aynı ünlü kullanıcıları sevdiği görülüyor ise diğer ünlü kullanıcıları da sevebilir düşüncesine cevap aranmaya çalışılmış ve bu konuda deney yapılmıştır. Deney sonucu bize gösterdi ki daha farklı uzunluktaki örüntüler arttıkça gerçekten daha iyi bir sınıflandırma yapılabilir ve daha başarılı bir şekilde ünlü düğümlerini takip etme olasılığı bulunuyordur.

Kullanıcıların takip ettiklerini takip edilmesine bakıldığında ise küçük de olsa bir katkı sağladığı görülmüştür. Ancak ne kadar kişinin kendisinden takip edilen kişiye doğru yarıçap arttıkça o kadar uygulama anlamında karmaşıklık arttığı için yapılan işlemin alınan kazanca oranla değerli olmadığı görülmüştür.

Gelen, giden bağlantıları ile düğümün kendisine ait bağlantılarına ait özellik etiketleri örüntü bulma konusunda karşılaştırıldığında giden düğümlere ait bağlantıların daha fazla bilgi içerdiği görülmüştür. Bu da Twitter’da takip edilen kullanıcıların bize daha fazla bilgi verdiğini gösteriyor. Bu deney sonucunda da etkilenme durumuna bakıldığında gerçekten takip edilen kullanıcılardan daha fazla etkilenilmektedir.

Bu tez çalışması sonucunda Twitter’da topolojik ilişkiler kullanılarak belirlenen ünlü kullanıcılar için takipçi tavsiye edildi. Bu ünlü kullanıcılar firmaların Twitter’daki hesapları olarak düşünülebilir. Böylelikle firmalar bütün kullanıcıları takip etmeye çalışmak yerine takip edilecekleri tahmin edilen kullanıcıları hedefleyerek takip etmesi sağlanabilir.

7. GELECEK ÇALIŞMALAR

Bu çalışma kapsamında önerdiğimiz metot birden fazla adımdan oluşmaktadır. Daha iyi tahmin sonuçları bulabilmek için gelecek çalışmalarda bu adımlarda iyileştirme yapılabilir. Bir iyileştirme ünlü düğümlerinin seçimiyle ilgili olabilir. Bu çalışma kapsamında ünlü düğümlerinin seçiminde sadece bağlantı sayısının belli bir eşik değerinden fazla olmasına bakılmıştır. Ancak bu ünlü düğümlerinin kim olduğuna dair Twitter'daki profil bilgilerine bakılmamış ve değiştirilmesi veya alınmaması gereken bir ünlü düğümü varsa bu yönde bir değişikliğe gidilmesi gelecek çalışma olarak bırakılmıştır.

Bu tezdeki çalışmaların bir benzeri Facebook verisi kullanılarak da tekrarlanabilir. Facebook'dan toplanacak veri ile sosyal ağ oluşturularak aynı yöntemler bu veri ile denenerek sonuçlar karşılaştırılabilir.

Deney aşamalarında sabit yarıçaptaki farklı uzunluktaki düğüm etiketleri için örüntü bulunmaya çalışılmıştır. Yarıçapı değiştirildiğinde aynı örüntülerin tahmin performansı ve farklı karmaşık örüntülerin ölçeklenebilir bir yapıda bulunması gelecek çalışmalarda araştırılmalıdır.

Daha önce bahsedildiği gibi bu çalışmanın bir benzeri metin madenciliği çalışmaları yapılarak da bulunabilir. Aynı çalışmayı metin madenciliği çalışmaları ile yapmaya çalışarak sonuçlar karşılaştırılabilir. Böylece hangi durumlarda metin madenciliği kullanılması gerektiği hangi durumlarda ağ madenciliği veya topoloji kullanımına gidilmesi gerektiği bulunabilir.

KAYNAKLAR

Aggarwal C. C., Wang H., (2010), “Managing and Mining Graph Data”, 1st Edition Springer.

Anagnostopoulos A., Kumar R., Mahdian M., (2008), “Influence and Correlation in Social Networks”, Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 7-15, Las Vegas, Nevada, USA, 24-27 August.

Barabasi A. L., Jeong H., Neda Z., Schubert A., Vicsek T., (2002), “Evolution of the Social Network for Scientific Collaboration”, PHYSICA A, 311, 590-614.

Christakis N. A., Fowler J. H., (2012), “Social contagion theory: examining dynamic social networks and human behavior”, Statistics in Medicine”, 32 (4), 556-577.

Clauset A., Moore C., Newman M. E. J., (2008), Nature 453, 1st Edition, Nature.

Conte D., Foggia P., Sansone C., Vento M., (2004), “Thirty years of graph matching in pattern recognition”, International Journal of Pattern Recognition and Artificial Intelligence, 18 (3), 265-298.

Dorogovtsev S. N., Mendes J. F., (2002), “Evolution of Networks”, Advances in Physics 1, 1-67.

Fan W., (2012), “Graph Pattern Matching Revised for Social Network Analysis”, Proceedings of the 15th International Conference on Database Theory, 8-21, Berlin, Germany, 27-30 March.

Gallagher B., (2006), “Matching structure and semantics: A survey on graph-based pattern matching”, The fall symposium on capturing and using patterns for evidence detection AAAI, 1, 45-53.

Goldberg D.S., Roth F.P., (2003), “Assessing experimentally derived interactions in a small world”, Proceedings of National Academy Sciences USA, 100(8), 4372-4376.

Golub B., Jackson M. O., (2012), “How Homophily Affects the Speed of Learning and Best-Response Dynamics”, Quarterly Journal of Economics, 127 (3), 1287-1338.

Gupta P., Goel A., Lin J., Sharma A., Wang D., Zadeh R., (2013), “WTF: The Who to Follow Service at Twitter”, Proceedings of the 22nd international conference on World Wide Web, 505-514, Rio de Janeiro, Brazil, 13-17 May.

Hasan M. A., Chaoji V., Salem S., Zaki M., (2006), “Link Prediction using Supervised Learning”, The proceedings of the Fourth Workshop on Link Analysis, Counterterrorism and Security, Bethesda, Maryland, 20-22 April.

Kang U., Faloutsos C., (2012) “Big Graph Mining: Algorithms and Discoveries”, ACM SIGKDD Explorations Newsletter, 14 (2), 29-36.

Kaplan A., Haenlein M. M., (2010), “Users of the World, Unite! The Challenges and Opportunities of Social Media”, Business Horizons, 53 (1), 59-68.

Kirac M., Özsoyoglu G., Yang J., (2006), “Annotating proteins by mining protein interaction networks”, Bioinformatics, 22 (14), 260-270.

Kwak H., Lee C., Park H., Moon S., (2010) “What is Twitter, a Social Network or a News Media?”, Proceedings of the 19th International World Wide Web (WWW) Conference, 591-600, Raleigh, NC, USA, 26-30 April.

Lee D.L., (1997), “Document ranking and the vector-space model”, Software IEEE, 14 (2), 67-75.

Lee R., Sumiya K., (2010), “Measuring Geographical Regularities of Crowd Behaviors for Twitter-based Geo-social Event Detection”, Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, 1-10, San Jose, CA, USA, 2-3 November.

Leskovec J., Huttenlocher D., Kleinberg J., (2010), “Predicting Positive and Negative Links in Online Social Networks”, Proceeding 19th International World Wide Web Conference, 641-650, Raleigh, North Carolina, USA, 26-30 April.

Liben-Nowell D., Kleinberg J., (2003), “The Link Prediction Problem for Social Networks”, J. American Society for Information Science and Technology, 58 (7), 1019-1031.

McPherson M., Smith-Lovin L., Cook J. M., (2001), “Birds of a Feather: Homophily in Social Networks”, Annual Review of Sociology, 27, 415–444.

Milenova, B.L., Yarmus J. S., Campos M. M., (2005), “SVM in Oracle Database 10g: Removing the Barriers to Widespread Adoption of Support Vector Machines”, Proceedings of the 31st international conference on Very large data bases, 1152-1163, Trondheim, Norway, 30 August-2 September.

Pennacchiotti M., Popescu A. M., (2011), “A Machine Learning Approach to Twitter User Classification”, Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 281-288, Barcelona, Catalonia, Spain, 17-21 July.

Porter M. A., Onnela J.-P., Mucha P. J., (2009), “Communities in Networks”, Notices of the AMS, 56 (9), 1082-1097.

Sen P., Namata G., Bilgic M., Getoor L., (2008), “Gallagher B., Eliassi-Rad T.: Collective Classification in Network Data”, AI Magazine 29 (3), 93-106.

Shasha D., Wang J. T. L., Giugno R., (2002), “Algorithmics and applications of tree and graph searching”, Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 39-52, Madison, Wisconsin, ABD, 3-6 June.

Shojaie A., (2013), “Link Prediction in Biological Networks using Multi-Mode Exponential Random Graph Models”, Proceedings of the 13th KDD Workshop on Learning and Mining with Graphs, Chicago, Illinois, ABD, 11-12 August

Taskar B., Wong M.-F., Abbeel P., Koller D., (2003), “Link prediction in relational data”, Proceeding of Neural Information Processing Systems, 1, 659–666.

Web 1, (2013), <https://dev.twitter.com/api>, (Erişim Tarihi: 01/12/2013).

Web 2, (2013), <http://www.facebook.com>, (Erişim Tarihi: 01/10/2013).

Web 3, (2013), <http://www.linkedin.com>, (Erişim Tarihi: 01/10/2013).

Web 4, (2013), <http://www.twitter.com>, (Erişim Tarihi: 01/10/2013).

Web 5, (2013), <https://blog.twitter.com/2013/celebrating-twitter7>, (Erişim Tarihi: 01/12/2013).

Web 6, (2013), https://en.wikipedia.org/wiki/Occupy_Wall_Street, (Erişim Tarihi: 01/12/2013).

Web 7, (2013), http://en.wikipedia.org/wiki/2013_protests_in_Turkey, (Erişim Tarihi: 01/12/2013).

Web 8, (2013), https://en.wikipedia.org/wiki/Arab_Spring, (Erişim Tarihi: 01/12/2013).

Web 9, (2013), <https://blog.twitter.com/2013/more-powerful-tools-for-small-business-and-self-service-advertisers>, (Erişim Tarihi: 01/12/2013).

Yan X., Han J., (2003), “gSpan: graph-based substructure pattern mining”, Proceeding 2002 of International Conference on Data Mining Conference, 721-724, Maebashi City, Japan, 9-12 December.

Zhou T., Lu L., Zhang Y.-C., (2009), “Predicting missing links via local information”, The European Physical Journal B. 71 (4), 623-630.

ÖZGEÇMİŞ

Burak IŞIKLI 1987 yılında Kırşehir’de doğdu. 2005 yılında başladığı Dumlupınar Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü’nü 2009 yılında başarıyla tamamladı. 2010 yılında Bilgisayar Mühendisliğinden çift ana dal olarak mezun oldu. Aynı yıl yüksek lisans eğitimine Gebze Yüksek Teknoloji Enstitüsü (GYTE), Mühendislik ve Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalında başladı. 2009 yılından bu yana Turkcell Teknoloji’de İş Zekâsı Sistemleri (BIS) bölümünde veri tabanı, veri işleme ve veri madenciliği konularında geliştirici olarak çalışmaktadır.

EKLER

Ek A: Tez Çalışması Kapsamında Yapılan Yayınlar

Isikli B., Sevilgen F. E., Kirac M., (2014), “Link and Annotation Prediction Using Topology and Feature Structure in Large Scale Social Networks”, Beyond Databases Architectures and Structures, Ustron, Poland, 27-30 May.