İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ÇİZGE VE İÇERİK VERİLERİNDE KOLEKTİF SINIFLANDIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Özge ATASEVEN

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Eylül 2017

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ÇİZGE VE İÇERİK VERİLERİNDE KOLEKTİF SINIFLANDIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Özge ATASEVEN
(504131540)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Yrd. Doç. Dr. Yusuf YASLAN

Eylül 2017

İTÜ, Fen Bilimleri Enstitüsü'nün 504131540 numaralı Yüksek Lisans Öğrencisi Özge ATASEVEN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "ÇİZGE VE İÇERİK VERİLERİNDE KOLEKTİF SINIFLANDIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Yrd. Doç. Dr. Yusuf Yaslan** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Yrd. Doç. Dr. Serap Kırbız Şimşek** Yıldız Teknik Üniversitesi	.....
	Yrd. Doç. Dr. Tolga Ovatman İstanbul Teknik Üniversitesi	.....
	Pro[illegible] Adı[illegible] ADI [illegible] Teknik Üniver[illegible]	.....

Teslim Tarihi : 13 Eylül 2017
Savunma Tarihi : 20 Eylül 2017

Sevgili aileme ve bana destek olan tüm arkadaşlarıma ,

ÖNSÖZ

Danışmanım Sayın Yrd. Doç. Dr. Yusuf Yaslan'a yüksek lisans egitimim boyunca çalışmalarıma vermiş olduğu destek nedeniyle teşekkürü bir borç bilirim. Kendisinin yapmış olduğu yönlendirmeler ve bana sağlamış oldugu fırsatlar bu çalışmanın yapılabilmesini mümkün kılmıştır. Tezin son halini almasındaki katkılarından dolayı Sayın Yrd. Doç. Dr. Serap Kırbız Şimşek'e de teşekkür ederim. Ayrıca bütün eğitim hayatım boyunca bana olan maddi manevi desteklerini bir an olsun esirgememiş aileme ve arkadaşlarıma teşekkür ederim.

Eylül 2017

Özge ATASEVEN
(Bilgisayar Mühendisi)

İÇİNDEKİLER

Sayfa

ÖNSÖZ ..... **vii**
İÇİNDEKİLER ..... **ix**
KISALTMALAR ..... **xi**
SEMBOLLER ..... **xiii**
ÇİZELGE LİSTESİ ..... **xv**
ŞEKİL LİSTESİ ..... **xvii**
ÖZET ..... **xix**
SUMMARY ..... **xxi**
1. GİRİŞ…. **1**
2. İNCELENEN YÖNTEMLER **3**
2.1 Grup Keşfi **3**
2.2 Örüntü Eşleme **4**
2.3 Topluluk Keşfi **4**
3. SINIFLANDIRMA YÖNTEMLERİ **7**
3.1 İçerik Tabanlı Sınıflandırma (Content-Only Classification) **7**
3.2 Bağlantı Tabanlı Sınıflandırma **7**
3.3 Kolektif Sınıflandırma **8**
4. SINIFLANDIRICILAR **11**
4.1 KNN Sınıflandırıcı (En Yakın k Komşu Sınıflandırıcısı) **11**
4.1.1 İçeriğe göre benzerlik ölçüsü 11
4.1.2 Ağdaki konumlarına göre benzerlik ölçüsü 11
4.2 Karar Ağacı (Decision Tree) Sınıflandırıcısı **13**
4.3 Naive Bayes Sınıflandırıcı **13**
4.3.1 İçerik tabanlı Naive Bayes sınıflandırma 13
4.3.2 Bağlantı tabanlı Naive Bayes sınıflandırma 16
4.4 Logistic Regression Sınıflandırıcısı **19**
5.KOLEKTİF SINIFLANDIRMA ALGORİTMALARI **21**
5.1 İteratif Sınıflandırma Algoritması (ICA) **21**
5.2 Global Sınıflandırma Algoritmaları **23**
5.3 Ağ Verilerinde Kolektif Sınıflandırma **28**
6.TARTIŞMA VE DENEYSEL SONUÇLAR **31**
6.1 Kullanılan Veri Kümeleri **31**
6.2 Uygun Eğitim ve Test Kümelerinin Oluşturulması **32**
6.3 Deneysel Sonuçlar **35**
6.4 Sınıflandırıcıların Birleştirilmesi ve Doğruluk İyileştirmesi **38**
6.5 Sonuçlar ve Tartışmalar **41**
KAYNAKLAR **43**

KISALTMALAR

SNA : Social Network Analysis
ICA : Iterative Classification Algorithm
NB : Naive Bayes
LBP : Loopy belief propagation
MF : Mean-Field Relaxation Labeling
CO : Content-only clasification
LR : Logistic regression
CC : Collective classification
ID : Identification (bir varlığı tanımlayan ona özel birimi)
KNN : K-Nearest Neighbour (K-En yakın komşuluk)
MRF : Markov Random Field (Markov Rasgele Alanlar)
ML-LBP : Multi-label Loopy Belief Propagation
ML-MF : Multi-label Mean-Field Relaxation Labeling
CO-KNN : Yalnızca içeriğe bağlı KNN sınıflandırıcısı (Content-Only KNN)
CO-NB : Yalnızca içeriğe bağlı Naïve Bayes sınıflandırıcısı
MV : Çoğunluk Seçimi (Majority Voting-MV)
WMV : Ağırlıklı Çoğunluk Seçimi (Weighted Majority Voting-WMV)

SEMBOLLER

N	**:** Komşu düğüm (Çevre fonksiyonu)
Y	**:** Etiketsiz düğüm
X	**:** Etiketli düğüm
G	**:** Çizge (graph)
V	**:** Düğüm (vertex)
C	**:** Etiket sınıfı
P	**:** Doğruluk değeri (accuracy)
E	**:** Kenar (edge)
Ψ	**:** Klik potansiyeli
Φ	**:** Genel klik potansiyeli
a	**:** Birleştirme işleminden geçmiş test vektörü
$\boldsymbol{V_i}$	**:** i'inci döküman verisi
$\boldsymbol{W_i}$	**:** İncelenen etiketli ya da etiketsiz bir düğüm
$\boldsymbol{F_i}$	**:** Nesnenin sahip olduğu öznitelik vektörü (feature)
T	**:** LBP algoritması klik potansiyellerini tutan tablo
$\boldsymbol{S_i}$	**:** Çoklu-etiketli veri kümesindeki herhangi bir etiket kümesi
$\boldsymbol{\beta}_i$	**:** Regresyon katsayısı
$\boldsymbol{O}$	**:** ICA geçici etiket listesi

ÇİZELGE LİSTESİ

Sayfa

Çizelge 4.1 : Naïve Bayes örnek veri kümesi **14**
Çizelge 4.2 : F_1, F_2, F_3, F_4, F_5 kelimeleri için Naïve Bayes öğrenme değerleri. **15**
Çizelge 4.3 : Makalelerin birbirine yaptıkları atıf sayıları **16**
Çizelge 6.1 : CORA veri kümesinde doğruluk değerleri **35**
Çizelge 6.2 : CiteSeer veri kümesinde doğruluk değerleri **36**
Çizelge 6.3 : Terrorist-Rel veri kümesinde doğruluk değerleri **37**
Çizelge 6.4 : Sınıflandırıcıların Birleştirilmesi **39**

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : En kısa yola dayalı topluluk keşfi **5**
Şekil 3.1 : Sınıflandırıcılar ve sınıflandırma algoritmalarının hiyerarşisi **8**
Şekil 3.2 : Kolektif Sınıflandırma algoritmalarının etiketsiz düğümü kullanması **10**
Şekil 4.1 : Kırmızı düğüm için k adet en yakın komşu düğümler.. ..... **12**
Şekil 4.2 : Naïve Bayes'in çizge incelemesi..... **18**
Şekil 4.3 : Logistic Regression **20**
Şekil 5.1 : İteratif sınıflandırma algoritması (ICA) **22**
Şekil 5.2 : ICA sözde kodu. **23**
Şekil 5.3 : Global sınıflandırma algoritması. **25**
Şekil 5.4 : LBP algoritması sözdekodu **27**
Şekil 5.5 : Çizge veri kümesinde atıf ilişkisi **29**
Şekil 6.1 : Uygun test verisinin oluşturulması. **33**
Şekil 6.2 : Etiketsiz düğümler arasındaki döngülerin engellenmesi..... **34**
Şekil 6.3 : CORA veri kümesinde doğruluk değerleri..... **36**
Şekil 6.4 : CiteSeer veri kümesi doğruluk değerleri..... **37**
Şekil 6.5 : Terrorist-Rel veri kümesinde doğruluk değerleri..... **38**
Şekil 6.6 : MV İşlemine Göre Sınıflandırıcı Birleştirme..... **39**
Şekil 6.7 : WMV İşlemine Göre Sınıflandırıcı Birleştirme..... **40**
Şekil 6.8 : Sınıflandırıcılar birleştirildikten sonraki doğruluk değerleri..... **40**

ÇİZGE VE İÇERİK VERİLERİNDE KOLEKTİF SINIFLANDIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI

ÖZET

Nesnelerin kategorilerine, bağlantılarına, içeriklerine göre sınıflandırılmasına finans, sosyal bilimler, iletişim, bilgisayarda görü ve biyoloji gibi birçok araştırma alanında ihtiyaç duyulur. Özellikle son yıllarda bu çalışma alanlarında kullanılan ağ verilerinin içerikleri ve birbirleriyle olan ilişkilerini çözmeye çalışmak günden güne önem kazanmaktadır. Sınıflandırma problemlerinde, nesnelerin sadece öznitelikleri kullanılarak yapılan sınıflandırma işlemleri, metin sınıflandırma gibi uygulama alanlarında İçerik-Tabanlı sınıflandırma olarak adlandırılmaktadır.

Geleneksel sınıflandırma algoritmaları yalnızca bağımsız içerik özelliklerine bakarak sınıflandırma işlemini gerçekleştirirken Kolektif Sınıflandırma algoritmaları hem bağımsız nesneleri (içerik verileri) hem de bağımlı nesneleri (çizge verilerini) kullanarak sınıflandırma işlemini gerçekleştirir. Bu algoritmalar doküman, görüntü, ilişki tipi sınıflandırma ya da farklı özelliklerdeki düğümlerin tespiti gibi işlemlerde sınıflandırıcının performasını arttırmak için nesneler arasındaki çizge verilerini kullanırlar. Kolektif sınıflandırma üzerine yapılan çalışmalar, ağ verilerinin bilgilerini de sınıflandırıcılara dahil ettiği için yalnızca içerik verileri kullanan sınıflandırma algoritmalarına göre daha yüksek doğruluk sonuçları elde eder.

Ağ verisinin yapısı çizge şeklindedir ve içeriğindeki her bir satır iki düğüm arasındaki ilişkileri temsil eder. Bu ilişkiler dökümanlar arası atıf, kişiler arası akrabalık, görüntüler arası benzerlik olabilir. Bu ilişki bilgilerinin sınıflandırıcılara dahil edilmesi nesneler arası benzerliklerin daha ön plana çıkmasını sağlar.

Sınıflandırma işlemini çizge verileri arasında gerçekleştiren algoritmalardan biri Iterative Classification Algorithm (ICA)'dir. ICA algoritması sınıflandırmayı gerçekleştirirken etiketi belli olmayan düğümün komşularının etiket bilgilerini kullanır. Bu yüzden bu algoritma yerel (lokal) sınıflandırma algoritması olarak adlandırılır. ICA algoritması etiketsiz bir düğümün etiketini belirlerken incelediği komşu düğümlerin de etiketi belli değilse özyinelemeli bir şekilde etiketsiz komşulara geçici değerler atar.

İçerik verisi ile ağ verisini birlikte kullanabilmek için yerel sınıflandırıcı olan Naïve Bayes ve K-En Yakın Komşuluk (KNN) algoritmaları ICA algoritması ile birleştirilerek Naïve Bayes-ICA, Naïve Bayes-KNN şeklinde kullanılır. Her iki algoritmada ele alınan etiketsiz düğümler için Naïve Bayes ve KNN içerik verisini sınıflandırırken, ICA algoritması ağ verisini sınıflandırır.

Yerel sınıflandırıcılara alternatif olarak Loopy Belief Propagation (LBP) ve Mean Field Relaxation Labeling (MF) gibi global sınıflandırıcılar da bulunmaktadır. Bu algoritmaların global olarak adlandırılma sebebi, ağ verilerinde sınıflandırma yaparken global bir amaç fonksiyonu (global objective function) kullanması ve

Markov rasgele alanlar ile ağın belli parçalarından sınıflandırılacak düğümleri elde etmesidir.

Özetle bu çalışmada, yerel ve global sınıflandırıcılar içerik ve bağlantı tabanlı veri kümelerine uygulanmış, sınıflandırıcıların doğruluk değerleri karşılaştırılmış ve bu sınıflandırıcıların birleştirilmesiyle daha yüksek başarım elde edilebilmesi amaçlanmıştır.

A COMPARISON OF COLLECTIVE CLASSIFICATION TECHNIQUES ON NETWORK AND CONTENT DATA

SUMMARY

The classification of objects according to categories, links or topics on network data is an important research area such as finance, social science, communication, computer vision and biology. Solving problems within network data by using each node's relation with the others, where for each node its features and relations with other nodes are available, become more common and important research topic in the last decade. If only attributes in the objects are used for classification, the classification is called Content-Only classification.

Collective classification algorithms use both independent objects (content dataset) and interconnect objects (graph dataset). These algorithms use link structure among the instances in order to improve the classification performance. This iterative algorithm which uses network dataset is called Iterative Classification Algorithm (ICA). The format of the network structure is graph data type and it includes the cites relationship between objects. Moreover, content dataset includes word and label information for document classification tasks. To use content data with this graph data we have combined the local classifiers as Naive Bayes-ICA and K-Nearest Neigbors (KNN)-ICA for "Collective Classification". These two algorithms provide local classification because of using neighborhood of unknown labeled object that needs to be classified.

ICA is one of the most successful local classification algorithms that uses local classifiers. It also has different variations such as Iterative Collective Classification (ICC) which is a generalization of ICA that use case-based classifier and Bayesian classifier.

Collective classification techniques aim to improve the classification performance of linked data by utilizing unknown nodes in the network that are classified by using known nodes and network structure. They can be grouped into local and global classification algorithms and use both link and content data for classifying the objects which are shown as nodes in the graph. The class labels can be both estimated by using content only Naïve Bayes classifier and collective classifier as well.

K-Nearest Neighbors (KNN) is another classifier that is used to label an unlabeled object. The KNN algorithm measures the closeness of the objects by looking at the words and citations of known and unknown labels in the network. Then, it looks at the neighbors in a particular proximity of the objects and assign the value of the label with the highest number of labels to the object. The similarities between unlabeled node and its neighbor labeled nodes are computed using the content information and weighted using the degree of the labeled nodes. Thus a node which is a higher degree neighbor of an unlabeled node will have small similarity value.

As an alternative to local classifiers, there are also global classifiers that use Markov Random Fields (MRF) as Loopy Belief Propagation (LBP) and Mean Field Relaxation

Labeling (MF) algorithms. We focus on both single-label and multi-label dataset classification by using these global algorithms and compare the approximation results for these two type of datasets.

Global classification algorithms use pairwise MRF to assign the class label for an unlabeled node by defining a global objective function. The algorithm uses clique potential of nodes which are found through marginal probability distribution of observed nodes. However, it is not an easy task to achieve numerically. In order to calculate one marginal probability, it is required to sum an exponentially large number of variables. Therefore, two approximate inference algorithms LF and MF are used to compute marginal probabilities. Both of the algorithms embrace the idea of avoiding the complexity in calculating marginal probability distribution. They use trial distribution rather than directly using pairwise MRF probability distribution. We begin with making trial distribution suitable for MRF distribution. Afterwards, it is easy to get marginal probabilities from trial distribution.

Each of the random variables in MRF sends a message to unlabeled node which is a neighbor of unlabeled node. This message consists of sum of the neighbor clique potentials and includes the probability of the label information. The LBP algorithm is applied on pairwise MRF.

Another approximate inference algorithm that can be applied to pairwise MRF is MF. As in the LBP algorithm MF algorithm computes marginal probabilities for each messages and it repeats the process until all marginal probabilities are stabilized. When the process is completed, the calculated marginal probabilities are given as a belief result. We also rearrange the MF for adapting the multi-label classification named as Multi-Label Mean Field Relaxation Labeling (ML-MF). For each class label we do an MF and considered all class labels together during performance evaluation.

Global classification algorithms aim to optimize a global objective function. As a global classification algorithm LBP is successful in different areas such as fraud detection on online retailers or classification in social network. However, it does not guarantee convergence due to increased cycles in the heterogeneous networks.

In this research, we use two different types of datasets namely single-labeled and multi-labeled datasets. CORA and Citeseer are the single-labeled document classification datasets where document topics are assigned as a label for each document. CORA is a bibliographic dataset with 2708 documents, 1433 words, 5429 links and 7 labels which are Rule Learning, Neural Network, Theory, Reinforcement Learning, Probabilistic Method, Genetic Algorithm. CiteSeer is also one of the most commonly analyzed bibliographic dataset with 3312 documents, 4732 links and 5 labels which are Agents, IR, DB, AI, HCI, ML.

Whereas as a multi-labeled dataset Terrorist-Rel has 4 different relation information (family, accomplice, contact, congregate) for each person which are treated as class labels. Multi-label classification resembles multi-class classification, except classes are not mutually exclusive. In multi-label dataset an instance can be assigned simultaneously into multiple classes. This dataset has the relationship between terrorists and represents each event of a terrorist as a feature. These events are used to obtain the relation between two terrorist nodes such as family or contact relationship. In our experiment two terrorist node could belong to same organization and same family. The terrorists are connected by a total number of 851 relation nodes with event information and it has 8592 links between each other. There are 4 sub-datasets that represents the relationship. If two people are in the same family (e.g. father-son,

husband-wife) they have family relation. If two people are members of the same terrorist organization, they have accomplice relation. If two people have contacted each other (e.g email, phone), they have contact relation. If two people use the same facility (e.g. go to the same training camp), they have congregate relation.

All of datasets (CORA, Citeseer, Terrorist-Rel) have graph dataset and these relational datasets have binary class labels. Thus, the relation between two terrorists is represented as a node. If the label of this node is family, contact, non-accomplice, noncongregate this means that these two terrorists are in the same family and contact each other but they have never been in the same terrorist organization and never used the same facility. In the experimental results, the test datasets are obtained using the snowball sampling (SS) method by selecting initial node and expanding the other nodes iteratively.

In this study, single-labeled linked data classification problem has been evaluated using ICA-KNN, ICA-Naïve Bayes, LBP, MF algorithms on bibliographic dataset. Then we have extended our experiments on terrorism relation multi-labeled linked dataset by using Multi-Label Loopy Belief Propagation (ML-LBP), Multi-Label Mean Field Relaxation Labeling (ML-MF) global classification algorithms.

To sum up, in this thesis we consider both single and multi-labeled linked data classification problem using local and global classification algorithms. Initially, single-labeled linked data classification problem is evaluated using ICA-KNN, ICA-Naïve Bayes, LBP and MF algorithms on bibliographic datasets. Then we extend our experiments on terrorism relation multilabeled linked dataset by using ML-LBP, ML-MF global classification algorithms. The experimental results show that for single-labeled linked data the best classification accuracy is obtained by MF global classification algorithm. For multi-labeled data both ML-LBP and ML-MF algorithms perform similarly.

1.GİRİŞ

Günümüzde, birçok makine öğrenmesi probleminde içerik bilgisine ek olarak nesneler arasındaki ilişki (bağlantı) bilgisi de kolaylıkla elde edilebilmektedir. Bu tür problemlerde sadece içerik bilgisini kullanan sınıflandırıcılar yerine bağlantı bilgisini de kullanan sınıflandırıcılar önem kazanmıştır. Bu doğrultuda kullanılan Kolektif Sınıflandırma (Collective Classification) algoritmaları hem bağlantı hem de içerik tabanlı sınıflandırıcıları bünyesinde barındırmaktadır. Sıradan sınıflandırıcılar nesnelerin etiketini belirlemek için yalnızca içerik bilgisini kullanırlar ancak Kolektif Sınıflandırma yöntemleri çizge bilgisini de sınıflandırma işlemine katarak sınıflandırıcının performansını arttırmayı hedefler.

Kolektif Sınıflandırma problemleri için literatürde önerilen ve başarıyla çalışan yöntemlerden biri ICA'dır [1]. ICA komşuluk bilgisini özyinelemeli olarak kullanarak etiketsiz verinin etiketini belirler. Bu işlemi yaparken sadece yakın komşuluk bilgisini kullandığı için lokal bir yöntemdir. Lokal yöntemlere alternatif olarak literatürde de Mean Field Relaxation Labeling (MF) ve Loopy Belief Propagation (LBP) gibi global yöntemler de önerilmiştir [1]. Bu yöntemler Markov Rastgele alanlar ile ağ üzerinden aldığı belirli ağ parçalarının bilgilerini kullanarak etiket tespiti gerçekleştirir.

Global sınıflandırma algoritmaları bir global nesne fonksiyonunu en iyilemeyi amaçlar. LBP algoritması global sınıflandırma algoritması olarak çevrimiçi sahtekarlık algılama, sosyal ağ sınıflandırması gibi konularda rahatlıkla kullanılabilir fakat Amazon gibi heterojen ağlarda çevrimiçi öneri verebilmesi için ZooBP geliştirilmiştir [2]. Bu algoritma LBP'den farklı olarak farklı türdeki düğümlerde (örneğin ürünler, kullanıcılar ve satıcılar gibi) sınıflandırma yapılabilmesini sağlar [2]. LBP dışında kullanılan diğer global sınıflandırma algoritması MF ise sonsal olasılığı arttırdığından dolayı görüntü algılamada, stereo eşleme, özyinelemeli segmentasyon gibi alanlarda uygulanabilir [3].

Yapılan çalışmada bağlantı ve içerik tabanlı Kolektif Sınıflandırma problemi hem lokal hem de global sınıflandırıcılar kullanılarak incelenmiştir. Çalışmada ele alınan tüm veri kümeleri; CORA, Citeseer ve Terrorist-Rel veri kümeleridir. CORA ve Citeseer veri kümeleri bibliyografik veri kümeleridir ve içeriğinde makaleler, makalelerin

kelimelerinin varlık durumları ve makalelerin birbirine yaptığı atıflar bulunmaktadır [1]. Terrorist-Rel veri kümesi ise terörist olduğu düşünülen kişileri içeren düğümler ve terörist düğümler arasındaki ilişkilerden oluşmaktadır [4].

Bu tez çalışmasında lokal yöntemler için ICA, global yöntemlerde de Mean Field Relaxation Labeling (MF) ve Loopy Belief Propagation (LBP) algoritmaları kullanılmıştır. Bu yöntemlerden ICA algoritması KNN ve Naive Bayes sınıflandırıcıları ile birleştirilerek ICA-Naive Bayes, ICA-KNN algoritmaları elde edilmiştir. Tüm bu algoritmalar tekli-etiketli olan CORA ve CiteSeer bibliyografik veri kümelerine uygulanmıştır. Böylece bu veri kümelerinde bulunan makalelerin konusu belli olmayanlarının konusunun tespit edilmesi sağlanmıştır. İncelemede ele alınan her bir makale için sadece konu bilgisi araştırıldığından bu veri kümeleri tekli-etiketli veri kümeleri olarak adlandırılır.

Çalışmada kullanılan diğer veri kümesi Terrorist-Rel ilişki tiplerini tutan bir veri kümesidir. Bu veri kümesi iki kişi arasındaki ilişkinin nasıl bir ilişki olduğunu dört farklı açıdan incelediği için (akraba-haberleşme-organizasyon-egitim) çoklu-etiketli veri kümesi olarak adlandırılır [4]. Global yöntemlerden türetilerek oluşturulan Multi-Label Loopy Belief Propagation (ML-LBP), Multi-Label Mean Field Relaxation Labeling (ML-MF) algoritmaları da çoklu-etiketli veri kümesi olan Terrorist-Rel üzerinde uygulanmıştır. Böylece ML-MF, ML-LBP algoritmalarıyla iki terörist düğüm arasındaki ilişkinin niteliği saptanmaya çalışılmıştır.

Son aşamada çizge (graph) veri tipi üzerinde klik potansiyellerinin nasıl elde edildiği ve Markov rasgele alanlar (MRF) konuları açıklanmıştır. Tartışma ve Sonuç kısmında ise programın çıktıları çizelgeler üzerinde gösterilerek algoritmaların doğruluk değerleri karşılaştırılmıştır. Hangi yöntemin hangi durumlarda daha iyi sonuç verdiği açıklanmıştır.

2.İNCELENEN YÖNTEMLER

Ağ ve içerik yapılı verilerin incelenmesi makine öğrenmesi için zengin bir araştırma alanı olmasıyla birlikte zorlukları da beraberinde getirmektedir. Bu zorlukların üstesinden gelebilecek etkin yöntemler üzerinde çalışmalar yapılmaktadır. Kolektif Sınıflandırma yöntemi de bu çalışmalardan sadece biridir. Kolektif Sınıflandırma seçilmeden önce bu yöntemlerin belli başlı olanları araştırılmıştır. Kolektif Sınıflandırma seçimi aşamasında incelenen yöntemler şunlardır:

- Grup Keşfi
- Örüntü Eşleme
- Topluluk Keşfi

2.1 Grup Keşfi

Grup keşfi gerçekleştirilirken çizge veri kümeleri kullanılarak sosyal ağ analizi yapılır ve incenen ağ belli başlı kriterlere göre alt gruplara ayrılarak kümeler elde edilir. Örneğin önceden yapılmış bir çalışma olan Geetor ve Diehl'in çalışma prensibi bir ağdaki aynı karakteristiğe sahip alt ağları kümelemektir. Alt ağlar bir dereceye kadar genel ağ içindeki ilginç ve ortak modelleri ortaya çıkarabilir [5].

Diğer taraftan sosyal ağ analizi, birbirine bağlı olan alt grupları bulurken birbirleriyle olan ilişkileri güçlü, direk, yoğun, sık veya olumlu olan alt grupları inceler. Çizge eşleme yöntemi de grup keşfi için tavsiye edilir. Grup keşfi yöntemi olarak kullanılan Kojak Group Finder uygulaması önce olası grupların yerini belirler ve bu grupları bilgi tabanlı sorgulayarak genişletir. Sonrasında ağda incelediği herhangi bir varlığın ilişkilerini kullanarak onu ilgili gruplara ekler [6].

Grup keşfinde alt küme oluştururken kümeler içinde tekrar kümeleme yapmaya izin verilmelidir. Çünkü varlıklar birden fazla gruba bağlı olabilir. Ancak çoğu algoritma bunu desteklemez. Bir bağlantıdaki verilerin aynı gruptan geldiğini düşünürsek, varlıkları gruplamak yerine bağlantılarını gruplayabiliriz. Böylece bağlantılar kümelendiğinde ona bağlı olan varlıklar da kümelenmiş olur. Bu sayede bağlantı bilgisi kullanarak alt grup

tespitinin daha kesin yapılabilmesi sağlanmış olur. Bu işlemi gerçekleştirmek için bağlantıları kümelere bölebilen ve uygun tanımlamış bir mesafe ölçümü yapabilen bir algoritmaya ihtiyaç duyulur [7].

2.2 Örüntü Eşleme

Örüntü eşleme büyük bir modelden küçük bir alt modelin aranarak elde edilmesidir. Örüntü eşlemede en bilindik yöntemler, kaba kuvvet tam örüntü eşleme ve Knuth-Morris-Pratt örüntü eşleme algoritmalarıdır [8]. Kaba kuvvet tam örüntü eşleme algoritmasına örnek verilecek olursa; arama yapılacak metnin her karakteri için aranacak metnin ilk karakterinin eşleşip eşleşmediği kontrol edilir. Eğer ilk karakterde eşleşme sağlandıysa arama yapılan metnin bir sonraki karakterine bakılır. Bir sonraki karakter için eşleşme sağlanmadıysa; arama yapılan metinde ele alınan karakterden başlanıp, aranacak metnin ilk karakteri için baştan karşılaştırma yapılır. Bu işlem tam eşleşme sağlanana kadar veya arama yapılan metin bitene kadar devam eder.

Namasteozge→ içinde arama yapılacak metin ozge→aranacak metin

Namasteozge
ozge
ozge
ozge

Şeklinde arama yapılır.

Knuth-Morris-Pratt örüntü eşleme algoritması, aranacak metnin bir kurala uygun şekilde düzenlenerek her seferinde en baştan arama yapmayı gerektirmeyecek bir arama yapılmasını öngörür [9]. İçinde aradığımız veride “aaabaabaaab” gibi a karakterinin tekrarlandığı bir veri olursa, ilk “a” karakterinden sonra hiç “b” karakteri gelmediği için buna göre bir kural oluşturulur. Böylece arama yaparken en başa dönmeden arama işlemini hızlı bir şekilde gerçekleştirir.

2.3 Topluluk Keşfi

Bir topluluk, birbirlerine atıfta bulunan makaleler içinden, bağlantılı web sayfalarından yani birbirleriyle ilişki içerisinde olan düğümlerin incelenmesi ile çıkarılabilir. Topluluklar bir ağın tamamının özeti olarak düşünülebilir. Bu da bizim ağı görselleştirmemizi ve anlamamızı kolaylaştırır. Bazen topluluklar, bireysel bilgilerin yayınlanmasına gerek kalmadan bu gruptaki düğümlerin özelliklerini ortaya koyabilirler.

Çizge verilerinde topluluk keşfi için kullanılan yöntemler aşağıdaki gibidir: [10]

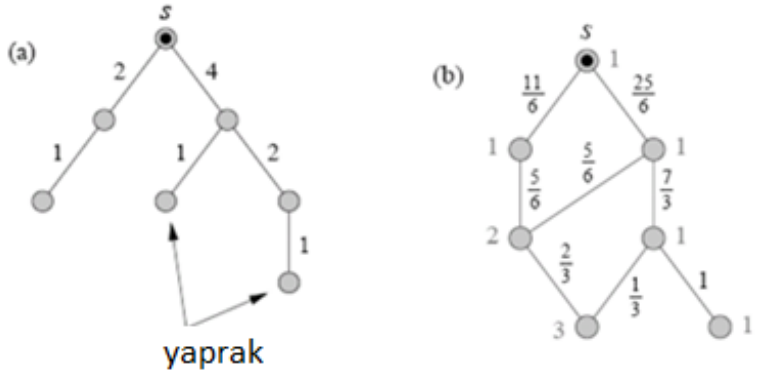
- Kenar aralığı (Edge betweeness):Tüm düğüm çiftleri arasındaki en kısa yolları bulup her bir kenardan kaç tane geçiş olduğunu sayar.
- Rastgele gidiş aralığı(Random walk betweeness).
- Current-flow betweeness.

Şekil 2.1'de bulunan örnek iki ağ üzerinden "En kısa yola (Shortest path betweenness) dayalı" topluluk keşfi yöntemi incelenmektedir. Şekil 2.1 a bir çocuk düğümün yalnızca bir ebeyni olduğu, Şekil 2.1 b ise bir çocuk düğümün birden fazla ebeveyne sahip olduğu bir ağ örneğidir. Her iki ağda da "S" tepe düğüm olmak üzere tüm düğümlerin ağırlıkları hesaplanacaktır. Şekil 2.1 a'yı inceleyecek olursak; yaprak düğümlerden başlayıp en üstteki "S" kaynak düğümüne doğru tüm düğümlerin benzerlikleri hesaplanır.

- Çocuk düğümlerin kenarları 1 ağırlığı ile başlamıştır.
- En alttaki yapraklardan inceleme başlanır ve yukarı seviyeye çıkıldıkça her düğüm için +1 ağırlık değeri eklenir. (bottom-up)
- Çocuk düğümlerin ağırlığı bir üst düğüme eklenir.

Birden fazla en kısa yol mevcutsa Şekil 2.1 b'yi incelemek gerekir. Bunun için önce hiçbir düğüm ve kenar için bir sayı atanmadığı düşünülür. İlk yapılacak işlem, "S" üst düğümünden başlayarak düğümlere ağırlıklar atamaktır.

- Önce kaynaktan herbir tepeye giden yolların sayısı hesaplanır.
- Yol sayıları için uygun ağırlık ataması yapılır.
- Yolların toplamı erişilebilir tepelerin sayısına eşitlenir.
- "S" başlangıç düğümü 1 ağırlığını alır. "S" düğümünün altındaki iki düğümün 1 adet ebeveyni olduğu için onlar da 1 ağırlığını alır.



Şekil 2.1: En kısa yola dayalı topluluk keşfi

- 2 numaralı düğümü oluşturan düğümün 2 adet ebeveyni vardır o yüzden bu düğüm 2 ağırlığını alır.
- Aynı şekilde 3 ağırlığını alan düğüm de 2 ağırlığını alan ve 1 ağırlığını alan 2 adet ebeveyne sahip olduğu için 3 ağırlığını almıştır.
- Düğümlere yapılan ağırlık atamasından sonra bu sefer yukarı çıkarak kenarların ağırlıkları hesaplanır. Yukarı çıkarken üzerinden geçilen her düğüm için çocuk düğümlerinin ağırlıkları toplanır. Elde edilen değer, üzerinden geçilen düğümün ebeveyn sayısına bölünür. Böylece üzerinden geçilen düğümün kenar ağırlıkları hesaplanmış olur.

Sınıflandırma işlemlerinde kullanılacak veri kümeleri hem içerik hem bağlantı bilgisinden oluştuğu için incelenen yöntemlerin bu veri kümeleri üzerinde kullanılması zordur. Bu veri kümelerine uygulanacak en uygun yöntem olarak Kolektif Sınıflandırma (CC) algoritmaları tercih edilmiştir. CC algoritmaları sayesinde hem yerel hem global sınıflandırma işlemleri başarıyla gerçekleştirilebilmektedir.

3.SINIFLANDIRMA YÖNTEMLERİ

3.1 İçerik Tabanlı Sınıflandırma (Content-Only Classification)

İçerik tabanlı sınıflandırmada nesnenin gözlemlenen özelliklerine bakılarak gözlemlenemeyen özelliklerin (etiket) kestirimleri yapılır. Tekli-etiketli veri kümelerinde ele alınan nesne makaleler olduğu için gözlemlenen özelliklerden öznitelik olarak kelimeler kullanılacaktır.

Araştırmanın ilk aşamasında sadece makale konusu belirleneceği için tekli-etiket sınıflandırması yapılır. Sonrasındaki çalışmada ise terörist düğümler arasında birden fazla ilişki tipi saptanacağı için çoklu-etiket sınıflandırması yapılacaktır. Tekli-etiket sınıflandırmasında nasıl içerik verileri makaledeki kelimeler ise çoklu-etiket sınıflandırmasında içerik verileri ilişki tipleridir.

3.2 Bağlantı Tabanlı Sınıflandırma

Bağlantı tabanlı sınıflandırma, çizge veri kümelerinden faydalanarak yapılır. Çizge veri kümesi makalelerin atıf ilişkilerini, bir şehrin birbiri ile bağlantılı ilçelerini, bir sosyal ağdaki bireylerin birbirleri ile olan arkadaşlıklarını gösterebilir. Çizge incelemesinde varlıklar birer nokta (vertex), varlıklar arasındaki ilişkiler de kenar (edge) olarak G(V, E) ile ifade edilir. Bağlantı tabanlı sınıflandırmalarda Naive Bayes, KNN gibi sınıflandırma yöntemleri çizge veri kümelerine uygulanır.

Bağlantı tabanlı sınıflandırma problemi Facebook üzerinden bir örnekle açıklanacak olursa; Terörist olduğu bilinen Tom adında bir kullanıcı ile Facebook'ta arkadaşlık kurduğu iki kişi incelensin. Alice, Tom ile aynı yerde çalışsın ama yakın arkadaş olmasın ve sosyal ağdaki ilişkilerinin tipi "iş arkadaşı" olsun. Diğer yandan Tom, sosyal ağda Bob ile "akraba" bağlantı tipi ile bağlı olsun. Tom'un bağlantıları incelenirken çizgede verilen bu bilgiler için «akraba» ilişki tipi «iş arkadaşı» ilişki tipinden daha fazla yakınlık puanına sahip olmalıdır. Bu sebeple bağlantı tabanlı sınıflandırmaların en önemli kriteri bağlantı tiplerinin önceliklerini belirlemektir [11].

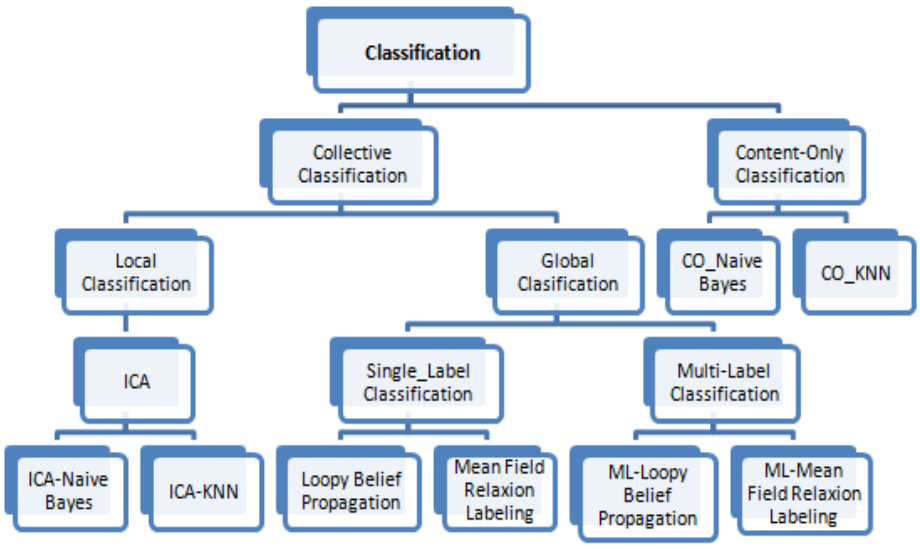
Tekli-etiketli veri kümesi olan makale veri kümesinde hem atıf yapan hem de atıf yapılan iki makale için atıf ilişkisi çift yönlü olduğundan çizge veri kümesi yönsüzdür.

3.3 Kolektif Sınıflandırma

Kolektif Sınıflandırma (CC), hem içerik tabanlı hem de bağlantı tabanlı algoritmaların kullanılmasıyla gerçekleştirilir. Bu işlemde CC algoritmalarının kullandığı sınıflandırıcılar Naive Bayes (NB), K-Neares Neighbour (KNN), Logistic Regression ve Karar Ağaçları (Decision Tree) 'dir [1].

Yapılan çalışmada, bu sınıflandırıcılardan KNN ve NB sınıflandırıcıları ICA ile birleştirilmek üzere gerçeklenmiştir. Bu sınıflandırıcılar yerel sınıflandırma yapan ICA algoritması ile çizge veri kümesinden ele alınan düğümlerin komşuluklarını kullanır.

Global sınıflandırma algoritmaları ise bu sınıflandırıcıların hiçbirine ihtiyaç duymaz, onun yerine Markov rasgele alanlar (Markov Random Fields-MRF) ile çizge verisi içinden rasgele seçtiği düğümlerin olası etiket değerlerinin marjinal olasılıklarını kullanır. Şekil 3.1'de çalışmada kullanılan tüm algoritmalar verilmiştir. Kolektif Sınıflandırma algoritmalarının lokal ve global olarak ayrıldığı, sadece içeriğe bağlı sınıflandırıcı olarak KNN ve Naive Bayes algoritmalarının kullanıldığı görülmektedir. Global yöntemlerde de tekli etkiketli ve çoklu-etiketli diye iki farklı tipte veri kümesi sınıflandırılmıştır.



Şekil 3.1 : Sınıflandırıcılar ve sınıflandırma algoritmalarının hiyerarşisi

Kolektif Sınıflandırma algoritmaları etiketsiz düğümü 3 farklı bakış açısıyla inceler. Bu aşamalar sırasıyla Şekil 3.2'de, her aşamada incelenen düğümün etki alanları kırmızı olacak şekilde gösterilmiştir. W_1, W_2 etiketsiz düğüm, W_3, W_4 etiketi belli düğümler olsun; F_1, F_2, F_3, F_4, F_5 bu düğümlerin sahip olduğu örnek içerikler olsun ve W_1 düğümünün etiketi tahmin edilmeye çalışılsın. Kolektif Sınıflandırma algoritmaları W_1 düğümünün etiketini bulmaya çalışırken çizge ve içerik bilgilerini kullanarak şunları inceler [1]:

1. Nesnenin gözlemlenen özellikleri ile nesnenin etiketi arasındaki ilişki;

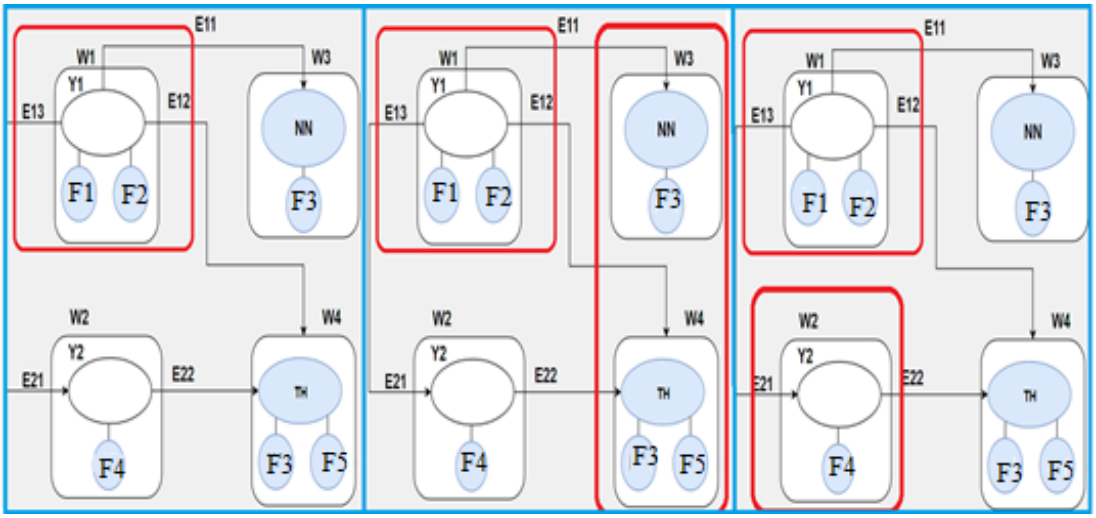
   W_1 düğümünün etiketini bulmak için öncelikle gözlemlenen özelliklerine bakılır (burada W_1'in sahip olduğu özellikler F_1 ve F_2 'dir). KNN ya da Naive Bayes sınıflandırıcıları Content-Only (sadece içerik) bu aşamada kullanılır.

2. Nesnenin komşu düğümlerinin gözlemlenen özellikleri ile nesnenin etiketi arasındaki ilişki;

   W_1 düğümünün komşularında gözlemlenebilen (etiket değeri belli olan) düğümler W_3 ve W_4'tür. Bu durumda W_1 düğümünün etiketi belirlenirken, etiketi belli olan komşularının da etiketleri dikkate alınır.

3. Nesnenin komşuluklarındaki nesnelerin gözlenemeyen etiketleri ve nesnenin etiketi arasındaki ilişki;

   Bu aşamada W_1 etiketsiz düğümünün, etiketsiz komşuları incelenir. Burada örnek etiketsiz komşu W_2 düğümüdür. W_2 için olası etiket tahminlemesi yapılırken özyinelemeli lokal sınıflandırma algoritması ICA ya da global sınıflandırmalarda mesaj geçişleri kullanılır. Her üç durum Kolektif Sınıflandırma'nın bağlantı tabanlı verilere nasıl yaklaştığını gösterir. Kolektif Sınıflandırma bağlantı tabanlı verilerde komşuluk matrisleri oluşturarak çalışır [12].



Şekil 3.2 : Kolektif sınıflandırma algoritmalarının etiketsiz düğümü kullanması.

4. SINIFLANDIRICILAR

4.1 KNN Sınıflandırıcı (En Yakın k Komşu Sınıflandırıcısı)

KNN etiketi bilinmeyen (unobserved) bir düğüme etiket atamak için kullandığımız bir sınıflandırıcı algoritmadır [13]. Yapılan çalışmada bu algoritma ICA algoritmasıyla birlikte kullanılmaktadır. KNN algoritması, ortamdaki etiketi bilinen ve etiketi bilinmeyen düğümlerin içerik ve atıf bilgilerini kullanıp bu düğümlerin birbirine olan yakınlıklarını ölçer. Daha sonra etiketi belli olmayan bir düğüme, bu düğümün belirli bir yakınlıktaki komşularının etiket bilgilerine bakarak etiket ataması gerçekleştirir. KNN çoklu-etiketli terörist düğümlerinin tespitinde de kullanılabilir [13].

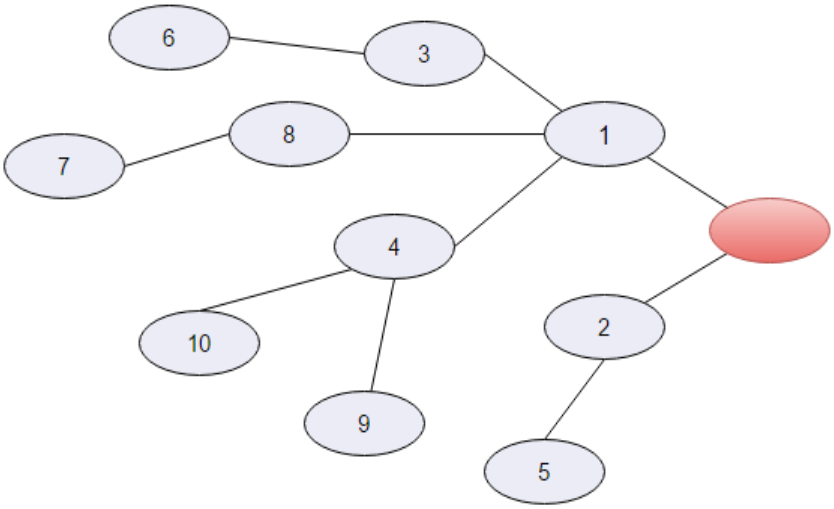
4.1.1 İçeriğe göre benzerlik ölçüsü

Veri kümesi üzerinde çalışırken düğümlerin birbirine yakınlıkları sayısal olarak ifade edilir. Bu işlemi yaparken düğümlerin içeriklerine ve ağdaki konumlarına ayrı ayrı bakılır ve bir benzerlik ölçüsü belirlenir. KNN benzerlik değerleri, içeriğe göre benzerlikler ve ağdaki konumlarına göre benzerliklerin birleştirilmesiyle hesaplanır [14].

4.1.2 Ağdaki konumlarına göre benzerlik ölçüsü

Bir düğüm eğer bir konu hakkında başka bir düğüme atıf yapıyorsa bu düğümler birbirleriyle doğrudan ilişkilidir. Bu önermeyi kullanmak için ağdaki konumlarına göre benzerlik ölçülerine ihtiyaç duyulur. Ağdaki konumlarına göre düğümler birinci dereceden yakın, ikinci dereceden yakın, üçüncü dereceden yakın,… i'inci dereceden yakın olarak sınıflandırılır. Ve yakınlık puanları $P_y = a^i \quad | \; 0 < a < 1$ şeklinde hesaplanır. Burada P_y , Y düğümünün herhangi bir etiket değeri için sahip olduğu yakınlık puanı, a yakınlık katsayısı, i yakınlık derecesini göstermektedir.

Şekil 4.1'de makale veri kümesinin atıf ilişkisi çizge ile gösterilmiş olsun. Kırmızı düğüm etiketsiz bir makaleyi gösterir. Bu makalenin birinci dereceden yakınları, atıflari olan 1 ve 2 numaralı makalelerdir. İkinci dereceden yakınları birinci dereceden yakınlarının atıfları olan 3, 8, 4 ve 5 numaralı makalelerdir. Üçüncü dereceden yakınları ikinci dereceden yakınlarının atıfları olan 6, 7, 10 ve 9 numaralı makalelerdir.



Şekil 4.1 : Kırmızı düğüm için k adet en yakın komşu düğümler.

Düğümleri yakınlıklarına göre grupladıktan sonra yakınlık dereceleri denklem 4.1 ile bulunur, "a" yakınlık sabitidir (denklemde örnek olarak 0.9 verilmiştir). P_{y1} birinci derecedeki düğümlerin kırmızı düğüme olan yakınlık değerini gösterir. P_{y1}'i bulmak için kırmızı düğüme birinci dereceden yakın olan 1 ve 2 numaralı düğümlerin yakınlık puanları olan P_{m1} ve P_{m2}'in birinci dereceden kuvveti alınır. Aynı şekilde ikinci derecedeki yakınlık değeri P_{y2} ile gösterilir. P_{y2}'yi bulmak için de ikinci derece komşu olan 3, 4, 5, 8 düğümlerinin yakınlık puanlarının ikinci dereceden kuvveti alınır. Son olarak üçüncü derecede bulunan 6, 7, 9, 10 düğümlerinin yakınlık puanlarının üçüncü dereceden kuvveti alınarak her birinin yakınlık derecesi P_{y3} bulunur. Bu işlem belli bir sayıdaki komşuluğu elde edene kadar alt dallara inecek şekilde devam eder.

Veri kümeleri üzerinde birden fazla veri test edilirken her bir verinin k adet komşusuna bakılacaksa, öncelikle çizgedeki konumlarına göre en yakın birinci derece komşular bulunur. Eğer araştırılacak k adet komşu tamamlanamamışsa daha sonra çizgenin ikinci seviyesine inilip ikinci derece komşular da alınır. Bu şekilde k adet komşu elde edilene kadar alt seviyelere inilir. Her inilen seviyede 4.1 denklemindeki gibi uzaklıkla ters orantılı olacak şekilde kuvvet alma işlemi yapılarak benzerlik oranları hesaplanır.

$$
\begin{aligned}
& a = 0.9 \Rightarrow P_{y1} = P_{m1} = P_{m2} = 0.9^1 = 0.9 \\
& \Rightarrow P_{y2} = P_{m3} = P_{m8} = P_{m4} = P_{m5} = 0.9^2 = 0.81 \\
& \Rightarrow P_{y3} = P_{m6} = P_{m7} = P_{m9} = P_{m10} = 0.9^3 = 0.73
\end{aligned}
\tag{4.1}
$$

4.2 Karar Ağacı (Decision Tree) Sınıflandırıcısı

Veri kümelerinden elde edilen eğitim verilerinin öznitelikleri kullanılarak karar ağacı ile kökten yaprağa doğru inilerek birtakım anlaşılabilir kurallar yazılır [15]. Karar Ağacı bir eğitim kümesi, sınıf öznitelikleri ve çeşitli özelliklere sahip sınıflandırılmış örnekleri kullanır. Karar ağacının her adımında en doğru niteliği seçip en iyi ayırmayı yapabilmek için istatiksel yöntemlerden biri olan bilgi kazanımı (information gain) kullanılır [15].

Bilgi kazanımı yöntemi kullanılarak karar ağacındaki en iyi nitelik seçildikten sonra bu seçme kriterine göre bütün örnekler iki ya da daha fazla alt kümeye ayrılır. Bu bölme, karar ağacındaki her bir bölünmüş küme için dallanan tek bir düğümle temsil edilir. Örnekler, kullanılan özellik tarafından tekrar bölünemezse, ilgili özellik ait olduğu özellik kümesinden kaldırılır. Bu durum bir alt kümede yalnızca tek bir örnek bırakılana veya başka bir durdurma ölçütü ile karşılaşıncaya kadar her alt grup için defalarca tekrarlanır, böylece kalan örnekler için birer yaprak düğüm oluşur.

Karar ağacında aşırı öğrenme, veri kümesine ait detay bilgiler barındıran gereğinden fazla karmaşık bir karar ağacının oluşmasıdır. Sadece düğüm ve yapraklardan ibaret bir karar ağacı oluşturulduktan sonra ağaç üzerinde fazla özelleşmiş nitelikler varsa aşırı öğrenme durumu söz konusu olabilir. Bu nitelikler öğrenme kümesinde kesin kararlar vermemizi engelleyebilir. Ağaçta aşırı öğrenme durumu gerçekleşirse budama ve erken sona erdirme işlemi ile ağaçta bir eşik değeri (dal seviyesi) belirleyerek karar ağacı oluştuktan sonra ağacı küçültme işlemine gidilebilir. ID3 algoritması, C4.5 algoritması gibi algoritmalar karar ağacı öğrenmesinde kullanılmaktadır. Karar ağaçları ile ilgili daha detaylı bilgiler [16] çalışmasında bulunmaktadır.

4.3 Naive Bayes Sınıflandırıcı

Naive Bayes yöntemi hem içerik tabanlı hem bağlantı tabanlı sınıflandırma yapar.

4.3.1 İçerik tabanlı Naive Bayes sınıflandırma

Naive Bayes'in çalışma yapısı makale veri kümesi üzerinde incelenecek olursa; Denklem 4.2'de F_n kümesi kelimeler kümesidir. C ise etiketler kümesidir. Naive Bayes sınıflandırıcısı ise temelini şartlı olasılıktan alır. P(A|B) şeklinde verilen bir P olasılığında " B bilindiğinde A'nın olma durumu nedir?" sorunu incelenir.

$$P(C|F_1,\dots F_n) = \frac{P(C)P(F_1,,\dots F_n|C)}{P(F_1,,\dots F_n)} \quad (4.2)$$

C=[C_1, C_2,.. C_7] etiket sınıfı (machine learning, neural network vs.), F=[F_1, F_2,…. F_{1400}] ise kelime sınıfını (kelime1, kelime2, vs..) temsil eder (Örnek CORA veri kümesi üzerinden verilmiştir. Bu veri kümesinde 7 sınıf 1400 içerik bulunmaktadır). P($C_1|F_1$,.. F_n), F kelime kümesi biliniyorsa bu kelimelere sahip bir makalenin C_1 etiketinde olma olasılığını gösterir. Bu olasılığın bulunması için öncelikle her bir etiketin C_1 olduğu bilindiğinde F_n kelimesinin geçme olasılığı P($F_n|C = C_1$) bulunmalıdır. Öznitelikler birbirlerinden bağımsız olarak kabul edildiğinde, herhangi bir makale için C= C_1 etiketi incelenecek olursa;

$$P(C = C_1|F_1,\dots F_n) \cong \frac{P(C=C_1)P(F_1|C=C_1)\,P(F_2|C=C_1)\dots\dots\,P(F_n|C=C_1)}{P(F_1,,\dots F_n)} \quad (4.3)$$

Denklem 4.3 için parametreleri açıklayacak olursak:

P($C = C_1|F_1,\dots F_n$): Kelimeleri bilinen bir makalenin C_1 etiketinden olma olasılığı,

P($F_1|C = C_1$) : C_1 etiketli olduğu bilinen makalelerde F_1 kelimesinin geçme olasılığı,

P($F_2|C = C_1$) : C_1 etiketli olduğu bilinen makalelerde F_2 kelimesinin geçme olasılığı,

P($F_n|C = C_1$) : C_1 etiketli olduğu bilinen makalelerde F_n kelimesinin geçme olasılığı,

P($C = C_1$) :Tüm veri kümesinde C_1 etiketinin görülme olasılığı.

V_1, V_2, V_3, V_4, V_5 bir veri kümesi içinde bulunan ve etiketi bilinen herhangi 5 makale olsun ve bu 5 makale üzerinde Naive Bayes uygulanıp etiketi bilinmeyen bir V_6 makalesinin etiketi tahmin edilmeye çalışılsın;

Çizelge 4.1 : Naïve Bayes örnek veri kümesi.

Makale\Kelime	F_1	F_2	F_3	F_4	F_5	Etiket
V_1	1	0	1	0	0	C_1
V_2	0	1	0	1	1	C_2
V_3	0	0	1	1	0	C_2
V_4	0	1	1	0	1	C_2
V_5	1	0	0	0	0	C_1

F_1, F_2, F_3, F_4, F_5 incelenen kelimeler olmak üzere; bir kelimenin makalede geçme durumu 1, geçmeme durumu 0 ikilik değeri ile ifade edilsin. C_1 ve C_2, C veri kümesindeki herhangi iki etiket olsun, örnek V_1 makalesi için öğrenme vektörü V_1= [1 0 1 0 0] şeklindedir. Bu, V_1 makalesinin 1. ve 3. kelimeleri içerdiğini, 2., 4. ve 5. kelimeleri içermediğini gösterir. Çizelge 4.2'de hesaplanmış her değer, Çizelge 4.1'deki örnek veri kümesinden yola çıkarak oluşturulmuş Naive Bayes'in öğrenme değerleridir.

Çizelge 4.2 : F_1, F_2, F_3, F_4, F_5 kelimeleri için Naïve Bayes öğrenme değerleri.

Etiket\Kelime	F_1		F_2		F_3		F_4		F_5	
	0	1	0	1	0	1	0	1	0	1
C_1	0	1	1	0	$\frac{1}{2}$	$\frac{1}{2}$	1	0	1	0
C_2	1	0	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$

Çizelge 4.2'yi incelersek; Eğer etiketin C_2 olduğu biliniyor ise F_2'nin makalede geçme olasılığı 2/3 ,geçmeme olasılığı 1/3 tür. Eğer etiketin C_1 olduğu biliniyor ise F_2'nin makalede geçme olasılığı 0, geçmeme olasılığı ise 1'dir. Etiketi bilinmeyen yeni bir V_6 makalesi V_6 =[0, 1 ,1, 1, 1] şeklinde verilsin, V_6 test verisi incelenirse :

- C_1 etiketine sahip olma olasılığı;

P (C $= C_1|F_1$=0, F_2=1, F_3=1, F_4=1, F_5=1) = 0 × 0 × (1/2) × 0 × 0

= 0 sonucunu verir.

- C_2 etiketine sahip olma olasılığı;

P (C $= C_2|F_1$=0, F_2=1, F_3=1, F_4=1, F_5=1) = 1 × (2/3) × (2/3) × (2/3) × (2/3)

= 0.296 sonucunu verir.

İncelenen V_6 makalesinin C_2 olma olasılığı, C_1 olma olasılığından daha büyük oldugu için bu test verisine hemen C_2 etiketlidir denilemez. Öğrenme değerlerinden çıkan 0'ları sadece olasılık değerleriyle çarpmak doğru değildir. 0 çıkan olasılıklar sonuçları bozar. Bu durumu engellemek için "Laplacian Correction" düzeltmesine gidilir. Laplacian yöntemi ile C_1 etiketine sahip olma olasılığı tekrardan hesaplanarak sıfır sonucunu veren örneklerin olasılığı sıfıra yakınsar hale getirir [17].

- Laplacian yöntemi ile C_1 etiketine sahip olma olasılığı;

P (C $= C_1|F_1$=0, F_2=1, F_3=1, F_4=1, F_5=1) = (1/3) × (1/3) × (1/2) × (1/3) ×(1/3)

= 0,005 sonucunu verir.

4.3.2 Bağlantı tabanlı Naive Bayes sınıflandırma

Veri kümelerinin birbirlerine yaptıkları atıflar da etiket analizinde kullanılır. İçerik üzerinde çalışan Naive Bayes sınıflandırıcısı bu atıflar ile çalışmaya uygun değildir. Bu yüzden bağlantı durumları için ayrı bir Naive Bayes sınıflandırıcı oluşturulması gerekir. Bu sınıflandırıcı her bir etiketteki nesneyle bağlantı kurma olayını ayrı bir koşul olarak kabul eder. Bağlantı tabanlı Naive Bayes sınıflandırıcı örnek bir veri kümesi üzerinde incelenirse, veri kümesinde C=[C_1, C_2,.. C_7] şeklinde 7 ayrı etiket olduğu düşünülsün. Öncelikle bu etiketlerdeki makalelere yapılan atıflar için etiket bazında atıf sayısı hesaplanır. Makalelere yapılan atıfların etiket bazında sayısı farklı birer durumdur ve bu durumlar için etiketsiz düğümlere her bir etiketin atanma olasılığı hesaplanmalıdır. Örnek bir hesaplama Çizelge 4.3'te verilmiştir.

Çizelge 4.3 : Makalelerin birbirine yaptıkları atıf sayıları.

C_1 etiketindeki makale atıfları						C_5 etiketindeki makale atıfları					
i \j	1	2	3	4	+	i \j	1	2	3	4	+
C_1	190	-	50	30	270	C_1	10	5	2	1	18
C_2	4	-	1	1	6	C_2	5	3	2	1	11
C_3	15	-	1	1	17	C_3	20	4	4	1	9
C_4	23	-	5	1	29	C_4	15	3	3	1	22
C_5	18	-	3	4	25	C_5	45	30	15	5	95
C_6	10	-	1	1	12	C_6	30	2	2	1	35
C_7	10	-	4	1	15	C_7	10	4	2	1	17
+	270	-	65	39	374	+	135	51	30	11	227

Çizelge 4.3'de i etiket değerini, j makale sayısını gösterir. Ayrıca çizelgede C_1 ve C_5 etiketli makalelere 1, 3, 4 sayıda atıf yapan makalelerin sayılarının etiketlere göre dağılımı gösterilmiştir. C_1etiketli makalelere 1 adet atıf yapan 190 tane C_1 etiketli makale vardır. 3 tane C_1'e atıf yapmış toplamda 50 adet C_1 etiketli makale vardır.

Olasılıklar 4.4 denklemi kullanılarak i'inci etiketin j adet makaleye yaptığı $A[i,j]$ atıf sayısının, i'inci etiketin tüm j adet toplam makale sayısına $\sum_{j=1}^{n} A[i,j]$ oranlanması ile hesaplanır. Böylece C_i etiketinde olduğu bilinen bir makalenin j tane C_i etiketli makaleye atıf yapma olasılığı $p(i,j)$ bulunur.

$$p(i,j) = \frac{A[i,j]}{\sum_{j=1}^{n} A[i,j]} \quad (4.4)$$

Makalelere yapılan atıf sayılarının tamamı eğitim kümesinden çıkarılamayabilir. Eğitim kümesinde C_1'e 2 atıf yapan hiçbir makale bulunmamaktadır. Bundan dolayı çizelgede C_1'e atıf yapma durumu boş gösterilmiştir. Ancak öğrenme matrisinde C_1'e 2 atıf yapma durumu yok diye bu durumun olasılığını sıfır kabul etmek doğru değildir. Çünkü C_1'e 3 atıf yapılmasıyla alakalı bir olasılık değeri varsa, bu C_1'e 2 atıf yapma olasılığını da içerir. Bunun bir sonucu olarak C_1'e 2 atıf yapma olasılığı C_1'e 1 atıf yapma olasılığıyla 3 atıf yapma olasılığı arasında bir değere sahiptir. Böyle durumdaki ara değerler 4.5 denklemindeki Lagrange interpolasyonu ile bulunur.

4.5 denklemi kullanılarak Çizelge 4.3'teki C_1'e 2 atıf yapma olasılıkları bulunmaya çalışılırsa; x_0, C_1'e 1 atıf yapan makale sayısını; x, C_1'e 2 atıf yapan makale sayısını; x_1, C_1'e 3 atıf yapan makale sayısını gösterirken; $f(x_0)$, C_1'e 1 atıf yapma olasılığını; $f(x_1)$, C_1'e 3 atıf yapma olasılığını göstermektedir. Tüm bu bilgilerden $f(x)$ bilinmeyeni (C_1'e 2 atıf yapma olasılığı) bulunmaya çalışılır.

$$f(x) = \frac{x-x_1}{x_0-x_1}.f(x_0) + \frac{x-x_0}{x_1-x_0}.f(x_1) \quad (4.5)$$

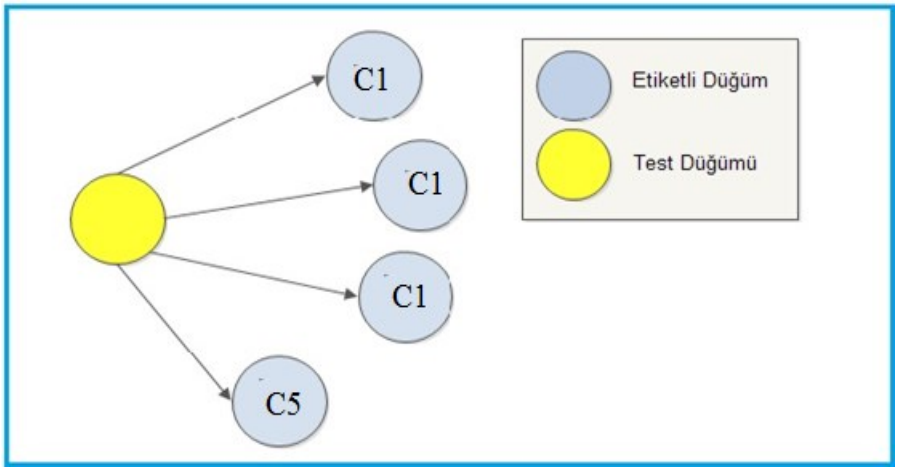
Şekil 4.2 üzerinde Naive Bayes sınıflandırıcının çalışması açıklanacak olursa; Etiketsiz sarı düğümün 3 adet C_1'e ve 1 adet C_5'e atıf yaptığı görülür. Çizelge 4.3'te i düğüm sayısı, j atıf sayısını gösterdiğinden dolayı etiketsiz test düğümü için bilenen durum X= (i=C_1, j=3; i=C_5, j=1) şeklinde ifade edilir. Çizelge 4.3'te P_1, C_1'e atıf yapma; P_2, C_5'e atıf yapma olayını göstersin. Böylece etiketsiz sarı düğümü bulmak için Çizelge 4.3 öğrenme kümesi olarak kullanıldığında sarı düğümün her bir C=[C_1, C_2,.. C_7] etiketinden olma olasılığı aşağıdaki gibi hesaplanır;

Test düğümünün C_1etiketinde olma olasılığı;

$$P(C = C_1|\, X) = P_1(C = C_1) \times P(C_{1_{3_{Atıf}}} | C = C_1) \times P_2(C = C_1) \times P(C_{5_{1_{Atıf}}} | C = C_1)$$

$$= \left(\frac{270}{374}\right) \times \left(\frac{50}{270}\right) \times \left(\frac{18}{227}\right) \times \left(\frac{10}{18}\right)$$

=0.00588



Şekil 4.2 : Naïve Bayes'in çizge incelemesi.

Test düğümünün C_2 etiketinde olma olasılığı;

$$P(C = C_2|\, X) = P_1(C = C_2) \times P(C_{1_{Atıf}}|C = C_2) \times P_2(C = C_2) \times P(C_{5_{1_{Atıf}}}|\, C = C_2\,)$$
$$= \left(\frac{6}{374}\right) \times \left(\frac{1}{6}\right) \times \left(\frac{11}{227}\right) \times \left(\frac{5}{11}\right)$$

=0.00005889

Test düğümünün C_3 etiketinde olma olasılığı;

$$P(C = C_3|\, X) = P_1(C = C_3) \times P(C_{1_{Atıf}}|\, C = C_3) \times P_2(C = C_3) \times P(C_{5_{1_{Atıf}}}|C = C_3)$$
$$= \left(\frac{17}{374}\right) \times \left(\frac{1}{16}\right) \times \left(\frac{29}{227}\right) \times \left(\frac{20}{29}\right)$$

=0.0000193

Test düğümünün C_4 etiketinde olma olasılığı;

$$P(C = C_4|\, X) = P_1(C = C_4) \times P(C_{1_{Atıf}}|C = C_4) \times P_2(C = C_4) \times P(C_{5_{1_{Atıf}}}|C = C_4)$$
$$= (\frac{29}{374}) \times \left(\frac{5}{29}\right) \times (\frac{22}{227}) \times \left(\frac{15}{22}\right)$$

=0.0000883

Test düğümünün C_5 etiketinde olma olasılığı;

$$P(C = C_5|\, X) = P_1(C = C_5) \times P(C_{1_{Atıf}}|C = C_4) \times P_2(C = C_5) \times P(C_{5_{1_{Atıf}}}|C = C_4)$$
$$= (\frac{25}{374}) \times \left(\frac{3}{25}\right) \times (\frac{95}{227}) \times \left(\frac{45}{95}\right)$$

=0.00159

Test düğümünün C_6 etiketinde olma olasılığı;

$$P(C = C_6|\, X) = P_1(C = C_6) \times P(C_{1_{3_{Atıf}}}|C = C_4) \times P_2(C = C_6) \times P(C_{5_{1_{Atıf}}}|C = C_4)$$
$$= (\frac{12}{374}) \times \left(\frac{1}{12}\right) \times (\frac{35}{227}) \times \left(\frac{30}{35}\right)$$

=0.00003533

Test düğümünün C_7 etiketinde olma olasılığı;

$$P(C = C_7|\, X) = P_1(C = C_7) \times P(C_{1_{3_{Atıf}}}|\, C = C_4) \times P_2(C = C_7) \times P(C_{5_{1_{Atıf}}}|C = C_7)$$
$$= (\frac{15}{374}) \times \left(\frac{4}{15}\right) \times (\frac{17}{227}) \times \left(\frac{10}{17}\right)$$

=0.0000471 şeklinde bulunur.

En yüksek olasılık $P(C = C_1|\, X)$ =0.00588 olduğu için makaleye C_1 etiketi atanır.

4.4 Logistic Regression Sınıflandırıcısı

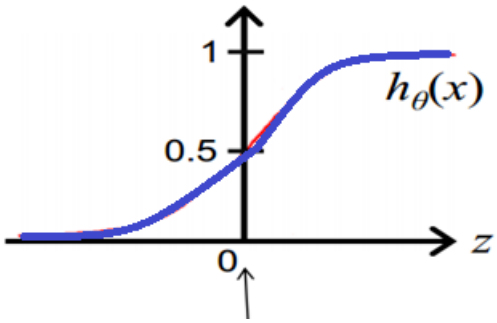
Logistic Regression, bir veri kümesinde bulunan etiketler ve içerikleri kullanılarak aralarındaki istatistiksel ilişkiyi matematiksel yollarla ifade edebilmeyi sağlar ayrıca Logistic Regression yönteminin amacı bağımlı etiket değişkeni ile bağımsız içerik değişkenleri arasındaki ilişkiyi en iyi uyuma sahip olacak şekilde tanımlamaktır.

Bağımlı değişkenler diğer değişkenler tarafından etkilenen değişkenlerdir. Veri kümelerinde bağımlı değişkenlere karşılık olarak etiketler gelir. Bağımsız değişkenler bağımlı değişkenleri etkiler. Kullanılan veri kümelerinde içeriklerin ve bağlantıların değişimi etiketlerin olasılıklarını değiştireceğinden, etiketler; içerik ve bağlantı bilgilerine bağımlıdır. 4.6 denkleminde x değeri aracılığı ile denetlenebilen eğim sayesinde eşik değeri için daha esnek bir çözüm sunulur.

$$H(x) = \frac{1}{1+e^{-x}} \tag{4.6}$$

4.6 denkleminin doğrusu Şekil 4.3'de verilmiştir. 4.7 denkleminde Logistic Regression 0 ile 1 arasında tanımlı bir sigmoid işlevinden gectiği için çıkardığı değerler olasılık olarak kullanılabilir.

$$\lim_{x\to\infty} H(x) = 1 \qquad \lim_{x\to-\infty} H(x) = 0 \tag{4.7}$$



Şekil 4.3: Logistic Regression

x değişkeninin tanımı aşağıdaki şekilde ifade edilir ve bu değer "logit" olarak adlandırılır.

$$x = \beta_0 + \beta_1 F_1 + \beta_2 F_2 + \beta_3 F_3 + \ldots\ldots\ldots + \beta_n F_n \qquad (4.8)$$

4.8 denklemindeki β_0 katsayısı engelleyici olarak adlandırılırken and β_1, β_2, β_3,.. β_n katsayıları sırası ile F_1, F_2 , $F_3 \ldots F_n$ özniteliklerinin regresyon katsayıları olarak adlandırılır. Engelleme katsayısı β_0, bütün katsayılar sıfır oldugu durumda logitin alacagı değerdir. Her bir regresyon katsayısı o özniteliğin logit degerine olan katkısını göstermektedir. Pozitif bir katsayı o özniteliğin olasılık değerini arttırdığı, negatif bir katsayı ise etkilediği özniteliğin olasılık değerini düşürdüğü anlamına gelir. Yüksek değerli bir katsayı özniteliğin çıktıya etkinin büyük olduğunu, sıfıra yakın düşük katsayı ise özniteliğin çıktıya etkisinin düşük olduğunu gösterir.

Logistik Regression'da birden fazla bağımsız değişkenin bağımlı değişkene etkileri bulunur, daha sonra bu değerler simetrik hale getirilerek 4.8'deki logit işlevini oluşturulur. Böylece belirtilen bağımsız değişkenlerin olduğu durumda bağımlı değişkenin hangisi olacağının tahmini yapılır. Bu bağımlı değişkenin belirlenmesi işleminden dolayı Logistic Regression bir sınıflandırma işlemidir.

5.KOLEKTİF SINIFLANDIRMA ALGORİTMALARI

5.1 İteratif Sınıflandırma Algoritması (ICA)

ICA lokal (yerel) sınıflandırma algoritması; Naive Bayes, KNN gibi sınıflandırıcılarla birlikte kullanılarak birden çok etiketsiz verinin aynı anda sınıflandırılmasını sağlar. ICA, her bir etiketsiz veri için etiket atama işlemini sırayla yapar. Ancak etiket ataması yaparken etiket değlerlerini rastgele seçer ve bu seçimde takip ettiği sıra sınıflandırmanın sonucunu etkiler. Etiketsiz bir düğüme yapılan ilk atamada, o düğüm artık test verisi olmaktan çıkar ve bir eğitim verisi olur. Böylece sınıflandırılan veriler tekrar eğitim için kullanılarak diğer etiketsiz verilerin sınıflandırılması için kullanılır.

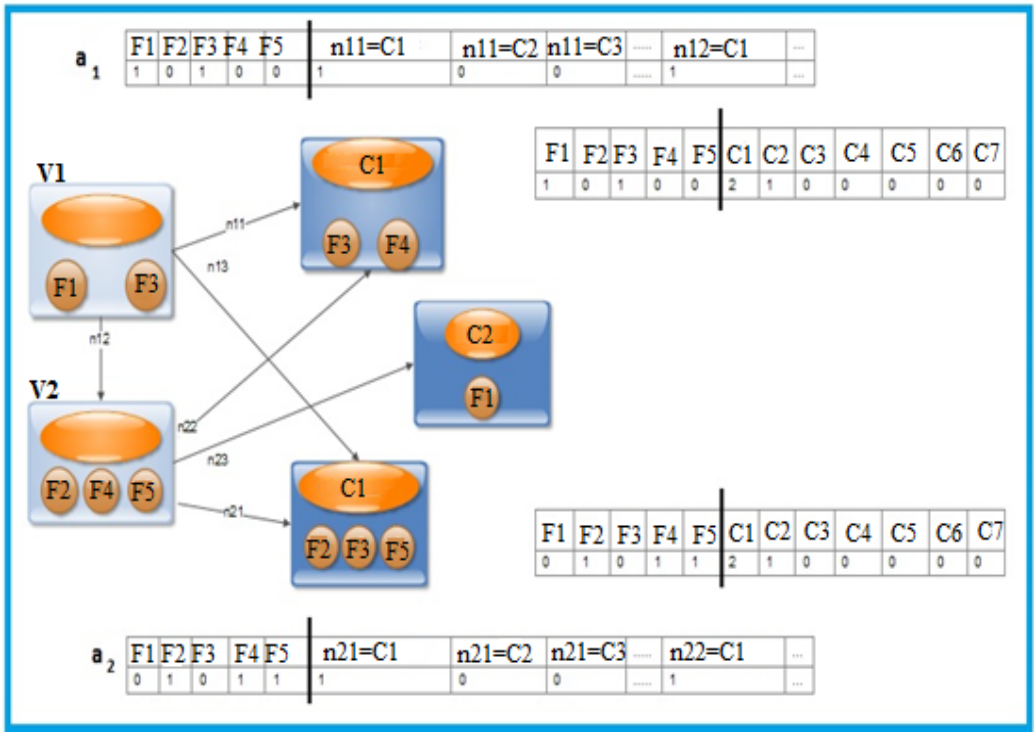
ICA bir düğümün sadece komşuluklarını inceleyerek etiket tahminleri yapar. Bunu yaparken Naive Bayes ya da KNN sınıflandırıcıları ile işbirliği içinde çalışır. Bu sınıflandırıcılardan herhangi biri etiketi bilinmeyen bir düğüme etiket atarken, ICA algoritması bu düğümün komşuluklarına bakarak sınıflandırıcıdan gelen olasılık değerlerini de kullanıp ilgili düğüm için yeni bir etiket çıkarımı yapar. Bu işlemi yaparken özyinelemeli bir yöntem kullanır.

Şekil 5.1'de ICA'nın örnek bir makale sınıflandırma probleminde nasıl çalıştığı gösterilmektedir. İncelenen etiketler C=[C_1, C_2,.. C_7] olmak üzere 7 tanedir. Her makalenin F_1, F_2, F_3, F_4, F_5 olmak üzere 5 tane kelimeyi içerip içermediğine bakılır. Şekil 5.1'de bulunan a_1 ve a_2 vektörleri sırasıyla V_1, V_2 makalelerinin kelimelerini ve bağlantı bilgilerini göstermektedir. Kelime makalede geçiyorsa 1, geçmiyorsa 0 olarak gösterilmiştir. Vektörlerin devamında ise V_1, V_2 makalelerinin bağlantı bilgileri bulunur. Örnek verilecek olursa a_1 vektörü, etiketsiz V_1 makalesine ait hem kelime hem de $\mathrm{n}_{11}, \mathrm{n}_{12}, \mathrm{n}_{13}$ bağlantı (atıf) bilgilerini tutar. V_1 makalesi n_{11} bağlantısını C_1 etiketli bir makale ile kurduğu için, a_1 vektörüne C_1 etiketi için 1 puan eklenir. V_1'in tüm komşuluklarıyla olan bağlantıları bu şekilde hesaplanır.

Bağlantı vektörlerinin uzunlukları herbir makale için farklıdır, çünkü her makale eşit sayıda atıf yapmaz. Makale vektörlerinin eşit uzunlukta olması için sayım birleştirmesi (count aggregation) yapılır. Böylece bir makalenin toplam atıf sayısı, (makale hangi

etikete kaç atıf yaptıysa hesaplanarak) etiket sayısı kadar uzunluklu bir vektörde tutulur. Sonuç olarak a_1, a_2… gibi makale vektörleri elde edilir ve bu makale vektörleri sınıflandırıcılara gönderilmek üzere hazırlanır.

Şekil 5.1'de V_1, V_2 etiketsiz makalelerinin her ikisine de bootstrapping (önyükleme) yöntemi ile rastgele C_1 etiketinin atandığı düşünülürse; V_1 makalesinin n_{12} bağlantısı için C_1 etiketine puan verilir. Böylece ilk iterasyonda V_1 makalesi için a_1 vektörünün n_{12} alanına 1 yazılır. ICA, makale vektörlerine etiketsiz komşulardan gelen bu yeni etiket değerlerini de ekleyerek sınıflandırma yapılması için elde edilen a_i vektörlerinin son halini f yerel sınıflandırıcısına verir. ICA her iterasyonda V_1 makalesinin etiketsiz komşularına farklı etiketler verip yeni oluşturduğu a_1 değerlerini sınıflandırıcılara gönderir ve sınıflandırıcı her vektör için yeni etiket çıktıları üretir. Bir önceki işlemden çıkan etiketler, bir sonraki işlemden çıkacak etiletlerle karşılaştırılmak üzere bir listede tutulur. İki liste birbiriyle aynı sonuçları verene kadar ICA sınıflandırma işlemini devam ettirir. Bu işlem özyinelemeli bir şekilde etiketler stabil hale gelene kadar tüm düğümler için yapılır. Listedeki etiketler bir daha değişmemek üzere sabit kaldığında etiket atama işlemi durur ve sınıflandırıcılardan gelen son etiketler gerçeğe en yakın değerler olarak belirlenir.



Şekil 5.1: İteratif Sınıflandırma Algoritması (ICA)

Şekil 5.2'de ICA'nın sözde kodu verilmiştir. Öncelikle etiketi belli olmayan Y_i makalelerine ön yükleme (bootstrapping, ön etiket belirleme) yöntemiyle kesin olmayan birer C_i etiket değeri atanır.

```
1: for tüm γi ∈ Y yap ön yükleme (bootstrapping)
2:   Her bir ai vektörü için Ni komşulukları bul
3:   γi ← f(ai) //f() sınıflandırıcı KNN ya da Naive Bayes
4: bit for
5: tekrarla
6:    O vektör listesini etiketsiz Y düğümü için hesapla
7:    for her bir Yi ∈ O
8:       yeni ai vektörlerini hesapla Ni komşuluklar ile
9:       γi ← f(ai)
10:   bit for
11: until tüm etikerler stabil hale gelinceye dek
```

Şekil 5.2: ICA sözdekodu

Etiket atama işleminden sonra Naive Bayes ya da KNN sınıflandırıcısı tarafından makalelerin olası etiket değerlerini bulmak için hesaplama yapılırken, ICA algoritması da bu sınıflandırıcılardan gelen olası etiket değerlerinin stabil olup olmadıklarını kontrol eder. Kısacası yerel sınıflandırıcılar etiket ataması yaparken bu etiket değerlerini ICA'ya verir. ICA da etiketsiz düğümlerin olası etiketlerinin değişimini kontrol ederek etiket atamasının durmasına veya devam etmesine karar verir.

5.2 Global Sınıflandırma Algoritmaları

Kolektif Sınıflandırma yapmak için alternatif yaklaşımlardan biri de global nesnesel (global objective function) fonsiyon tanımlamaktır. Bu fonksiyonun tanımı Markov Rastgele Alanlar (MRF- Markov Random Fields) ile sağlanır.

MRF, etiketli ve etiketsiz düğümlerden oluşan ve bu düğümler arasındaki kenarların yönsüz olduğu çizgelerdir. MRF'lerde düğümler V ile, düğümler arasındaki kenarlar da E ile gösterilir. "Markov rastgele alan çifti" bir çizgeden rastgele alt çizgelerin seçilerek üzerinde inceleme yapılmasını sağlar (pairwise MRF olarak da söylenir). İkili (pairwise) MRF'de her düğüm rastgele bir değişken ve her kenar bu rastgele değişkenlerin arasındaki bağlantılardır. Bu düğümler ve bağlantılar G=(V,E) çizge yapısını oluşturur. V iki türde de rastgele değişkenlere karşı gelmektedir. Burada gözlemlenemeyen düğüm Y ve gözlemlenebilen düğüm X olarak gösterilir. MRF'lerde etiketsiz düğümlere etiket atamak için "klik potansiyeli" (clique potantial) denilen değerler kullanılır. Klik

potansiyeli: etiketsiz düğümlerin komşusu olan düğümlerle ve içeriğiyle arasındaki ilişkilere göre bu düğümlere etiket kümesindeki etiketlerin atanma olasılıklarını gösteren yakınlık ölçüsü birimidir.

Ψ Simgesi "klik potansiyeli" kümesini gösterir. 3 tip **Ψ** (klik potansiyeli) vardır.

1-her etiketsiz düğümün kendi özellikleriyle olan **Ψ** (klik potansiyeli)

2-her etiketsiz düğümün etiketi belli komşularıyla **Ψ** (klik potansiyeli)

3-her etiketsiz düğümün etiketi belli olmayan komşularıyla arasındaki **Ψ**(klik potansiyeli)

Etiketsiz düğümler arasındaki ilişki için klik potansiyeli bulunurken, etiketsiz düğümlerin her bir etikette olma durumuna göre klik potansiyelleri hesaplanır. İki etiketsiz düğüm arasındaki (i,j) klik potansiyeli $\Psi_{i,j}$ şeklinde gösterilir. Burada i incelenecek etiketsiz düğümü gösterirken, j değeri i'nin bir niteliğini, etiketli komşusunu ya da etiketsiz komşusunu göstermektedir.

Çizgedeki etiketi belli komşular X, etiketi bilinmeyen düğümler de Y ile gösterilir. Buna göre Y_i'ye atanacak etiket değeri C_i olsun, Y_i'nin etiketsiz komşusu Y_j'ye atanacak etiket değeri ise C_j ile gösterilsin. Gözlemlenemeyen etiketsiz bir Y_i düğümü için ilk iki tipteki **Ψ** klik potansiyeli kullanılarak (Y_i'nin kendi özellikleriyle ya da etiketi belli X_j komşusuyla arasındaki klik potansiyeli) 5.1 denklemindeki Φ_i hesaplanır. Bu da Y_i etiketsiz düğümüne ait genel klik potansiyelini gösterir.

$$\Phi_i(C_i) = \Psi(C_i) \prod_{(Y_i, X_j) \in E} \Psi_{ij}(C_i) \qquad (5.1)$$

5.1.denklemine göre bir Y_i etiketsiz düğümün olası her C_i etiket değeri için tüm komşulukları arasında oluşturduğu ağ, ikili Markov rastgele alanlar olarak ifade edilir (pairwise MRF). Tüm gözlemlenemeyen Y_i düğümlerine etiket atanması, komşuluklarıyla olan ikili MRF olasılık dağılımıyla sağlanır.

$$P(C \mid x) = \frac{1}{Z(x)} \prod_{Y_i \in y} \Phi_i(C_i) \prod_{(Y_i, Y_j) \in E} \Psi_{ij}(C_i, C_j) \qquad (5.2)$$

Denklem 5.2'de ise $P(C|x)$ değeri yani herhangi bir Y_i düğümünün C etiketinde olma olasılığı, Y_i düğümünün $\Phi_i(C_i)$ genel klik potansiyeli ile Y_j komşuluklarıyla arasındaki **Ψ** klik potansiyellerinin çarpılmasıyla elde edilir. Bu olasılığı bulabilmek için 5.1 denkleminden elde edilen Y_i düğümünün genel klik potansiyeli $\Phi_i(C_i)$ ile, Y_i'nin etiketsiz komşuları arasındaki klik potansiyelleri Ψ_{ij} çarpılır. Burada kullanılan Z

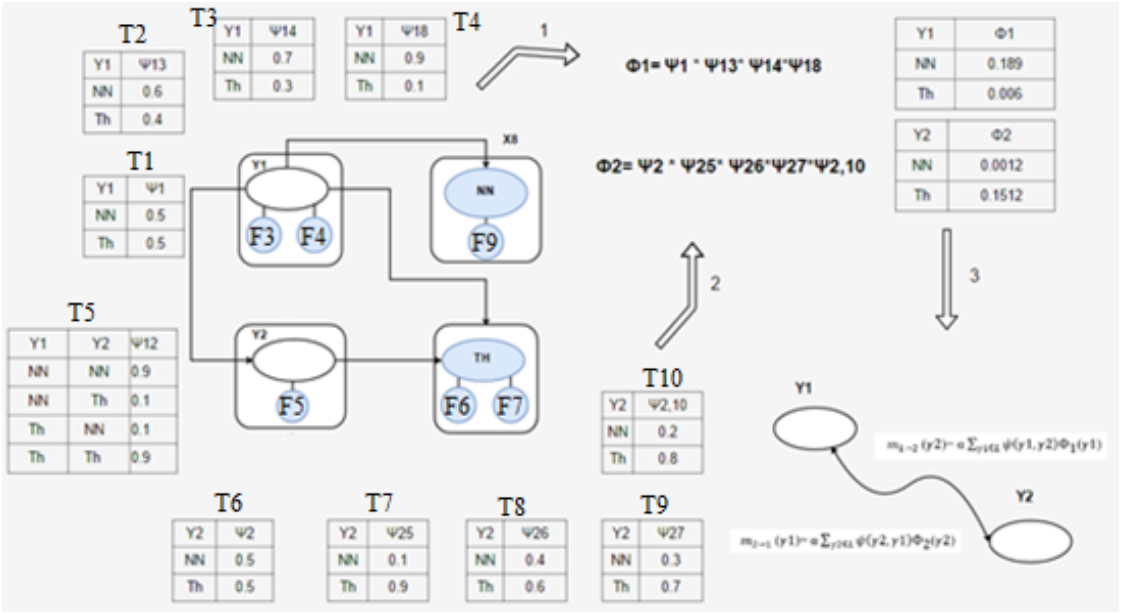
katsayısı 0 ile 1 arasında bir olasılık değeri bulunması için kullanılan bir normalizasyon sabitidir. Denklem 5.3'teki gibi de gösterilebilir.

$$Z(x)=\sum_{Y} \prod_{Y_i \in y} \Phi_i(C_i) \prod_{(Y_i,Y_j) \in E} \Psi ij(C_i, C_j) \quad (5.3)$$

Şekil 5.3'te LBP örneği verilmiştir. Y_1 ve Y_2 etiketsiz iki düğümü; X_8, X_{10} etiketi belli düğümleri; F_3, F_4, F_5, F_6, F_7, F_9 bu düğümlere ait özellikleri (features) göstermektedir. X_8 düğümünün etiketi NN, X_{10} düğümünün etiketi ise Th olarak verilmiştir. $\boldsymbol{\Psi}_1$ ve $\boldsymbol{\Psi}_2$ ise Y_1 ve Y_2 için kendileriyle olan klik potansiyellerini göstermektedir. Etiketsiz her düğümün kendi özellikleriyle ve başka düğümlerle arasındaki her bağı için birer tane $\boldsymbol{\Psi}$ (klik potansiyel) tanımlanır. Örneğin $\boldsymbol{\Psi}_{1,3}$, Y_1 düğümüyle F_3 özelliği arasındaki klik potansiyelini gösterir. $\boldsymbol{\Psi}_{1,4}$, Y_1 düğümüyle F_4 özelliği arasındaki klik potansiyelini gösterir. $\boldsymbol{\Psi}_{18}$, Y_1 düğümüyle X_8 düğümü arasındaki klik potansiyelini gösterir. Y_1 düğümünün bağ kurduğu tüm etiketi bilinen düğümlerle ve kendi özellikleriyle arasındaki klik potansiyeli bulunduktan sonra $\boldsymbol{\Phi}_1$ genel klik potansiyeli denklem 5.4'teki gibi hesaplanır. Kısacası Şekil 5.3'e göre verilen bir Y_i etiketsiz değişkeni için, tüm etiketi belli değişkenlere giden kenarlar alınır ve bu kenarlarla ilgili klik potansiyelleri, Y_i'nin kendi klik potansiyeli ile çarpılır. Y_1 ve Y_2 için; Φ_1, Φ_2 genel klik potansiyeli aşağıdaki gibi bulunur:

$$\Phi_1(C_i) = \boldsymbol{\Psi}_1 \times \boldsymbol{\Psi}_{13} \times \boldsymbol{\Psi}_{14} \times \boldsymbol{\Psi}_{18} \quad (5.4)$$

$$\Phi_2(C_i) = \boldsymbol{\Psi}_2 \times \boldsymbol{\Psi}_{25} \times \boldsymbol{\Psi}_{26} \times \boldsymbol{\Psi}_{27} \times \boldsymbol{\Psi}_{2,10} \quad (5.5)$$



Şekil 5.3: Global sınıflandırma algoritması

Gözlemlenemeyen Y_i düğümüne en iyi etiket atanmasını gerçekleştirmek için hesaplanan klik potansiyelleri, Y_i düğümüyle ilişkili olasılıklardan elde edilir. T_1 tablosundaki klik potansiyeli Y_1 düğümünün kendisiyle olan klik potansiyelidir ve ağ içinde toplamda 2 adet etiket bulunduğu için her bir etiketin ağdaki olasılığı 0.5 olarak belirlenmiştir. T_2 ve T_3 tabloları etiketsiz Y_1 düğümünün kendi özellikleriyle arasındaki klik potansiyelidir. Burada da T_2 tablosu örnek verilecek olursa, F_3 kelimesinin geçtiği makalelerin %60 'ının NN etiketli olduğunu görürüz.

Çalışmada gerçekleştirilen LBP, MF yaklaşık sonuç çıkarma algoritmalarının her ikisi de marjinal olasılık dağılımının hesaplanmasındaki sayısal karmaşıklıktan kaçınma yaklaşımını benimsemiştir. Doğrudan ikili MRF ile alakalı olasılık dağılımını kullanmak yerine her ikisi de denklem 5.3'teki deneme (trial) dağılımını kullanır. Sonrasında marjinal olasılıklar deneme dağılımından çıkarılır.

MRF içerisindeki her rastgele değişken Y_i komşusu olan Y_j'ye bir mesaj gönderir $m_{i\to j}$. $m_{i\to j}$ mesajı Y_i düğümünün Y_j'ye ağdaki herbir etiket değeri için gönderdiği olasılık değerlerini içerir. Bu Y_j'nin etiketinin ne olması gerektiğini gösterir. MRF içerisindeki rastgele değişkenlerdeki mesaj geçiş algoritması LBP (loopy belief propagation) dir.

LBP'yi diğer mesaj protokollerinden ayıran en önemli özelliği şudur; bir $m_{i\to j}$ mesajı hesaplanırken LBP daha önceki iterasyonda Y_j'nin Y_i ye gönderdiği $m_{i\to j}$ mesajını dikkate almaz. Bunun sebebi Y_j nin hesapladığı mesajın kendine dönmesini engellemektir. İkili MRF (G, **Ψ**) üzerinde uygulanmış LBP algoritması 5.6 denklemiyle gösterilebilir.

$$m_{i\to j}\,(C_j) = \alpha \sum_{C_i \in L} \psi(C_i, C_j)\Phi_i(C_i) \qquad (5.6)$$

5.6 denkleminde i'den j'ye gönderilen bir mesajın olası tüm etiket değerleri için genel klik potansiyellerinin toplamı bulunur. İşlem sırası da şu şekildedir; Öncelikle mesajın geldiği i düğümünün her etiket değeri için genel klik potansiyelleri bulunur. Daha sonra bu potansiyeller toplanır. En son bir α katsayısı ile normalizasyon işlemi yapılır.

$$\prod_{Y_k \in N_i \cap y \setminus Y_j} m_{k\to i}\,(C_i)\,, \forall\, C_i \in C \qquad (5.7)$$

5.7 denkleminde Y_j düğümü için; incelenen etiketsiz düğüm Y_j dışındaki bütün etiketsiz düğümlerin etiketli düğümler ile aralarındaki mesaj geçişleri bulunur. Algoritma her Y_i

(gözlemlemeyen, etiketsiz) düğümü için, komşusu olan tüm etiketsiz düğümlere bağlantı mesajı gönderir.

$$b_{\mathrm{i}}(C_{\mathrm{i}}) = \alpha\, \Phi_{\mathrm{i}}(C_{\mathrm{i}}) \prod_{Y_{\mathrm{j}} \in N_{\mathrm{i}} \cap y} m_{j \to i}(C_{\mathrm{i}}) \,, \forall\, C_{\mathrm{i}} \in C \qquad (5.8)$$

5.8 denkleminde ise her etiket değeri için bütün mesaj geçiş değerleri genel klik potansiyelleri ile çarpılarak inanç (belief) $b_{\mathrm{i}}(C_{\mathrm{i}})$ değeri bulunur. Buradaki α her mesaj ve her marjinal olasılık toplamlarının 1'e eşit olacağını garanti eden bir normalizasyon sabitidir. Böylece Y_{i} etiketsiz (unobserved) düğümüne C_{i} etiketinin atanma olasılığı 5.8 denklemindeki $b_{\mathrm{i}}(C_{\mathrm{i}})$'nin hesaplanmasıyla bulunur. Şekil 5.4'de ise tüm denklemlerin birleştirildiği LBP algoritmasının sözdekodu verilmiştir.

$$\sum_{C_{\mathrm{i}}} m_{i \to j}(\, C_{\mathrm{i}}) = 1 \; ve \; \sum_{C_{\mathrm{i}}} b_{\mathrm{i}}(C_{\mathrm{i}}) = 1 \qquad (5.9)$$

1: **for each** $S_{\mathrm{t}} \in$ S her bir çoklu-etiket veri kümesi için
2: **for each** $(Y_{\mathrm{i}}\text{-}Y_{\mathrm{j}}) \in$ Y pairwise $\in$ Y
3: **for each** $C_{\mathrm{i}} \in$ C **do**
4: $m_{i \to j}\,(C_{\mathrm{j}})$=1
5: **end for**
6: **end for**
7: **repeat**
8: **for each** $(Y_{\mathrm{i}}\text{-}Y_{\mathrm{j}}) \in$ Y
9: **for each** $C_{\mathrm{j}} \in$ C **do**
10: $m_{i \to j}(C_{\mathrm{j}})$=α $\sum_{\gamma_{\mathrm{i}} \in L} \psi(C_{\mathrm{i}}, C_{\mathrm{j}}) \Phi_i(C_{\mathrm{i}})$
$\prod_{Y_k \in N_{\mathrm{j}} \cap y \setminus Y_{\mathrm{j}}} m_{k \to i}\,(C_{\mathrm{i}}), \forall\, \gamma_{\mathrm{j}} \in L$
11: **end for**
12: **end for**
13: **until** all $m_{i \to j}\,(C_{\mathrm{j}})$ stabilizasyon sağlanana kadar
14: **for each** $Y_{\mathrm{i}} \in$ Y
15: **for each** $C_{\mathrm{i}} \in$ C
16: $b_{\mathrm{i}}(C_{\mathrm{i}}) =$ α $\Phi_{\mathrm{i}}(C_{\mathrm{i}}) \prod_{Y_{\mathrm{j}} \in N_{\mathrm{i}} \cap y} m_{j \to i}\,(C_{\mathrm{i}})$
17: **end for**
18: **end for**
19: **end for**

Şekil 5.4: LBP algoritması sözdekodu

$$b_{\mathrm{i}}(C_{\mathrm{i}}) = \alpha\, \Phi_{\mathrm{i}}(C_{\mathrm{i}}) \prod_{Y_{\mathrm{j}} \in N_{\mathrm{i}} \cap y} \; \prod_{C_{\mathrm{i}} \in \mathrm{C}} \; \psi_{\mathrm{ij}}{}^{b_{\mathrm{j}}(\mathrm{j})}(C_{\mathrm{i}}, C_{\mathrm{j}}) \,, \forall\, C_{\mathrm{i}} \in C \qquad (5.10)$$

Denklem 5.10'da, MF algoritması için $b_{\mathrm{i}}(C_{\mathrm{i}})$ belief değerinin nasıl hesaplandığı görülmektedir. MF algoritmasını LBP algoritmasından ayıran fark, $b_{\mathrm{j}}(C_{\mathrm{j}})$ komşu düğümün belief değerinin MF algoritmasında kuvvet olarak kullanılmasıdır.

5.3 Ağ Verilerinde Kolektif Sınıflandırma

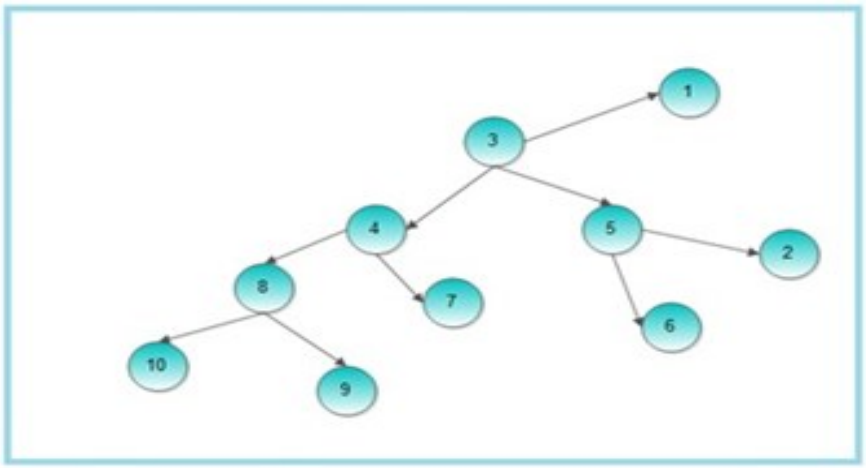
Ağ verisi nesneler arası bağlantı bilgilerini tutan bir veri kümesidir. Bu nesneler kişi, makale, web sayfası olabilir. Nesnelerin birbirleriyle ilişkilerini gösteren veriler; sosyal ağ analizleri, makale analizleri ve buna benzer bir çok işlemde etkin bir biçimde kullanılmaktadır.

Ağ yapısındaki verilerde her bir nokta (node), bağlantılarla (egde) bir diğer noktaya bağlıdır. Bu bağlantılar bir sosyal ağda kişiler arasındaki arkadaşlığı gösteririken bu çalışmada makaleler arası atıfları ve terörist düğümler arası ilişkileri gösterecektir.

Web sayfaları, sosyal ağ, iletişim ağları, biyolojik ağlar (protein etkileşimi) gibi ağ verileri üzerinde son zamanda en çok yoğunlaşılan konulardan biri de düğümleri sınıflandırarak klasik makine öğrenmesi tekniklerini genişletmektir. Bağlantı (network) ağları, fiziksel sistemlerde tanımlı ağlar ve sosyal ağların günden güne daha da önemi artmaktadır. Bu ağlardaki düğümlerin birbirlerini nasıl etkiledikleri ayrı bir araştırma alanıdır [18]. Internet sitelerinin birbirleriyle olan bağlantıları ya da bir makalenin atıfta bulunduğu diğer makalelerle arasındaki konu yakınlığı gibi birçok senaryonun çözümü için en önemli adım ağdaki düğümlerde sınıflandırma ya da etiketlemeler yapmaktır.

Bağlantı tipi sınıflandırmalarda Naïve Bayes gibi sınıflandırıcılar bağlantılar üzerinde uygulanır. Sosyal bir ağda terörist düğümleri olduğu bilinen düğümlerin bağlantı tiplerini ve bağlantı yapılarını kullanarak, diğer bağlantılardaki teröristler ya da terörist olmayanlar saptanabilir. Yapılan çalışmada çizge tipi veri kümelerini incelemek için matrislerden yararlanılmıştır. Bu veri kümelerine örnek olarak, birbirine atıf yapan 10 adet makalenin Şekil 5.5 üzerinde çizgesi ve matris formu gösterilmiştir. Daha sonra bu çizgenin kenarlarından yola çıkarak, bir makalenin (düğümün) komşularına bakılır. Çizgedeki bir makale için tüm makalelere bakıp, diğer tüm makalelerle ilişkide olma durumu 1, olmama durumu da 0 ile ifade edilecek şekilde 10x10'luk bir A çizge matrisi oluşturulur. Daha sonraki işlemler (metrik hesapları, öğrenme verisi bulma vs.) oluşturulan çizge matrisi üzerinden yapılır.

Örneğin çizge üzerinde yapılacak işlem makale atıf analizinde birbirine en yakın makaleleri bulmak olsun; Bu durumda bir düğümün herhangi bir düğüme olan yakınlık derecesi, bu iki düğümün diğer düğümlere olan ortak komşuluklarının toplanması ile bulunur. Bu da A matrisinin, kendisinin transpozu ile çarpılmasıyla gerçekleşir. Bulunan N komşuluk matrisindeki herhangi bir $N_{i,j}$= 1 için i,j düğümlerinin bir adet ortak komşusu



KOMŞULUK MATRİSİ

	1	2	3	4	5	6	7	8	9	10
1	0	0	1	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	1	0	0	1	1	0	0	0	0	0
4	0	0	1	0	0	0	1	1	0	0
5	0	1	1	0	0	1	0	0	0	0
6	0	0	0	0	1	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0
8	0	0	0	1	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	1	0	0	0

Şekil 5.5: Çizge veri kümesinde atıf ilişkisi

var demektir. Bu da N komşuluk matrisindeki iki makalenin yakınlık derecesini gösterilir [19].

Günümüzde ağ verilerini en iyi temsil eden yapı sosyal ağlardır. Sosyal ağlarda sınıflandırma çeşitli yollarla başarılı bir şekilde yapılmaktadır. Bu yollardan en kolayı; niteliği (etiketi) bilinen düğümünleri veri olarak kullanmak ve sınıflandırılan düğümlerin kendilerine ait niteliklerini baz alan bir (Naive Bayes gibi) sınıflandırıcı kullanmaktır.

Sosyal ağlarda kullanılan başka bir yöntem ise "bağlantı tabanlı sınıflandırma" (relational classifier) yöntemidir. Bu sınıflandırma algoritmaları, tipik bir ağdaki türdeş varlıkları ya da belirli bir düğüm kümesindeki sosyal ağ verilerini bir çizge olarak modeller. Herhangi bir veri kümesinde belirli kısıtlara dayanarak düğümlerin arasına kenarlar (edge) eklenir. Örneğin bir Facebook verisinde, düğümler arasında var olan bir arkadaşlık bağlantısına dayanarak, düğümler arasına arkadaşlık ilişkisi anlamına gelen kenarlar eklenir. Bu şekilde oluşturulan çizge verisine sınıflandırıcılar uygulanır.

6.TARTIŞMA VE DENEYSEL SONUÇLAR

6.1 Kullanılan Veri Kümeleri

Araştırmada tekli-etiketli (single-labeled) ve çoklu-etiketli (multi-labeled) olarak iki farklı veri kümesi kullanılmıştır. CORA ve Citeseer, makale sınıflandırmasında kullanılan tekli-etiketli veri kümeleridir. Terorist-Rel veri kümesi çoklu-etiketli veri kümesi olup, sınıf etiketleri olarak değerlendirilen her kişi için 4 farklı bağ bilgisi (aile, suç ortağı, iletişim, topluluk) içerir. Çoklu-etiketli sınıflandırmada, her etiket sınıfı birbirinden bağımsız değerlendirilir [20]. Çok etiketli veri kümesinde, eşzamanlı olarak birden çok etiketsiz nesneye birden fazla etiket ataması yapılabilir [21]. Örneğin iki terörist düğümün aynı organizasyona ve aynı aileye ait olduğu eşzamanlı saptanır. Çalışmada kullanılan tüm veri kümeleri aşağıdaki gibi detaylandırılabilir:

CORA: 2708 makale, 1433 kelime, 5429 bağlantı ve 7 etiketin bulunduğu tekli-etiketli bibliyografik veri kümesidir. Etiket isimleri; Rule Learning, Neural Network, Theory, Reinforcement Learning, Probabilistic Method, Genetic Algorithm şeklindedir.

CiteSeer: 3312 makale, 1400 kelime, 4732 bağlantı ve 5 etiketin bulunduğu bir veri kümesidir. Etiket isimleri; Agents, IR, DB, AI, HCI, ML şeklindedir.

Terrorist-Rel: Bu veri kümesi terörist düğümleri ve bu düğümler arasındaki ilişkileri içerir. Veri kümesindeki her bir bağlantı bilgisi (iki terörist düğüm arasındaki ilişki) bir özellik olarak kullanılır. Veri kümesinde 851 adet düğüm, 8592 adet bağlantı ve 4 adet alt veri kümesi bulunmaktadır.Veri kümeleri şu şekildedir [13]:

- Aile: Birbiriyle ilişkili iki terörist düğüm aynı aileye mensup ise iki düğüm arasında aile ilişkisi vardır. (baba-oğul, eş, soyad)
- Suç Ortağı: Birbiriyle ilişkili iki terörist aynı tipten suça karıştıysa suç ortağı ilişkisi vardır.(bomba, kaçırma, NBCR (nükleer-biyoloji-kimya-radyoloji), silah)
- Haberleşme: Birbiriyle ilişkili iki terörist herhangi bir yolla birbiriyle haberleştiyse aralarında haberleşme ilişkisi vardır. (e-mail, telefon)

- Toplantı: Birbiriyle ilişkili iki terörist daha önce aynı toplantıya katıldıysa toplantı ilişkisi vardır. (bulundukları kamp bilgileri ve etkinlikler)

Bu veri kümesinde tüm ilişkiler ikilik sistemde temsil edilir. Böylece iki terörist düğüm arasındaki ilişki bir düğüm olarak gösterilir. Veri kümesinde ele alınan örnek bir düğümün etiketleri "aile - haberleşme - suç ortağı değil - toplantı değil (family, contact, non-accomplice, non-congregate)" ise bu iki düğüm aynı aileye mensup, birbirleriyle haberleşiyor ancak aynı terörist organizasyonunda hiç bulunmamışlar ve aynı suçta hiç birlikte isimleri geçmemiş sonucuna varılır.

Deneysel sonuçlarda ağ verileri için kartopu örneklemesi snowball sampling (SS) yönteminden yararlanılarak örnek test veri kümesi özyinelemeli bir şekilde komşu düğümleri kullanarak oluşturulmuştur. Kartopu örneklemesinde sahip olunan bağlantılar ile arttırılan ilişkiler sayesinde bir anda daha fazla bağlantılar kurulabilmekte, bu sayede bir anda aşırı derecede büyüyen ilişkisel bilgiler toplanabilmektedir. Bu durum küçük bir kartopunun yuvarlanırken yerdeki karları etrafında toplayarak boyut olarak bir anda büyümesine benzetilir [22].

6.2 Uygun Eğitim ve Test Kümelerinin Oluşturulması

Çizge verileri üzerindeki etiketsiz düğümler test edilmeden önce uygun test verilerinin oluşturulması gerekir. Çizge araştırmasında kullanılan veri çeşitleri test verisi ve eğitim verisidir.

Eğitim kümesi: Etiketi belli olan çıkarım yapılacak veri kümesidir.

Test kümesi: Gerçekleştirdiğimiz çalışmada test kümesi, etiketi belli olmayan makaleler ya da terörist düğümleridir. Yapılan çalışmanın amacı ise bu verilerin etiketlerinin tahmin edilmesidir.

Test verisi oluşturmak için uygun veri desenlerinin elde edilmesi gerekmektedir. Kolektif sınıflandırmanın çizge verileri üzerindeki bir düğümü üç değişik açıdan incelediği 3.3 numaralı konu başlığında anlatılmıştı. Böylece Şekil 6.1'e göre bir test verisi için her üç değişik olayı da içerisinde barındıracak şekilde komşuluklarından yeni test verileri seçilmelidir.

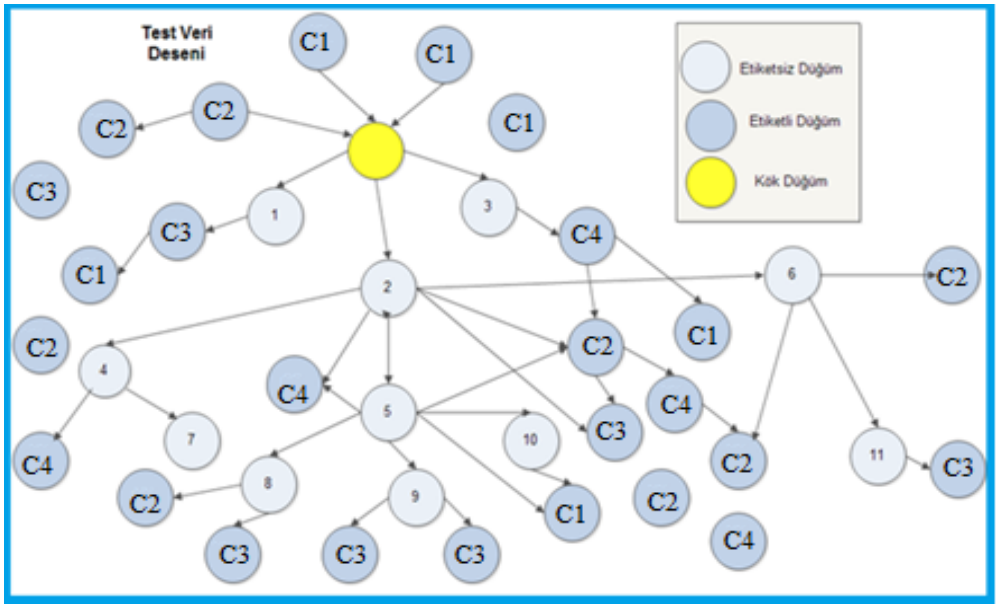
İncelenen algoritmaların genelinde etiketsiz düğümlerin komşuluğu probleminin çözümüne dair yöntemler vardır. Bu yüzden, test veri kümesi çıkarılırken birbirleriyle

komşu makalelerin etiketsiz düğüm olarak belirlenmesi gerekir. Komşu etiketler alınmasında izlenen yöntem şu şekildedir;

Veri kümesinden seçilen rastgele bir düğümün komşularının yarıdan fazlası etiketli bırakılacak şekilde seçimler yapılır. Bir test düğümü için komşularının hangisinin etiketsiz düğüm yapılacağı belirlenirken, $komşu\_sayısı/2$ kadar düğüm test verisi olarak alınır. Alınan düğümlerin komşu sayısının çok olmasına özen gösterilir. Bu sayede yaprak düğümlerin test verisi olarak seçilmesi önlenmiş olur. Aksi halde yaprak düğümlere ulaşınca komşu sayısı sadece 1 olacak ve istenen sayıda etiketsiz düğüm oluşturulamayacaktır.

Sürekli aynı etiketteki komşulukların seçilmemesine dikkat edilir. Çünkü düğümler arası mesaj geçişlerinin daha doğru hesaplanabilmesi için komşu düğümlerden olabildiğince veri kümesindeki tüm etiket çeşitlerini içerecek şekilde test verisi seçilir.

Şekil 6.1'de C_1, C_2, C_3, C_4 sınıfına ait makalelerin bulunduğu çizge veri deseni örneği görülmektedir. Veri deseninde bulunan koyu renkli düğümler etiketi belli düğümler iken açık renkli düğümler etiketsiz test düğümleridir. Test düğümleri belirlenmeden önce tüm veri deseni koyu renkli (etiketi belli) düğümlerden oluştuğu kabul edilir. Veri deseni üzerinde test düğümleri belirlenirken sarı ile gösterilmiş örnek bir düğüm için komşuları arasında en çok komşusu olandan en az komşusu olana doğru bir sıralama yapılır ve bunların $komsusayısı/2$ kadarı etiketsiz test düğüm olarak seçilir. Bir sonraki aşamada 2 numaralı düğümün 7 tane komşu düğümü (sarı, 4, 5, 6, C_2, C_3, C_4) olduğu için en çok

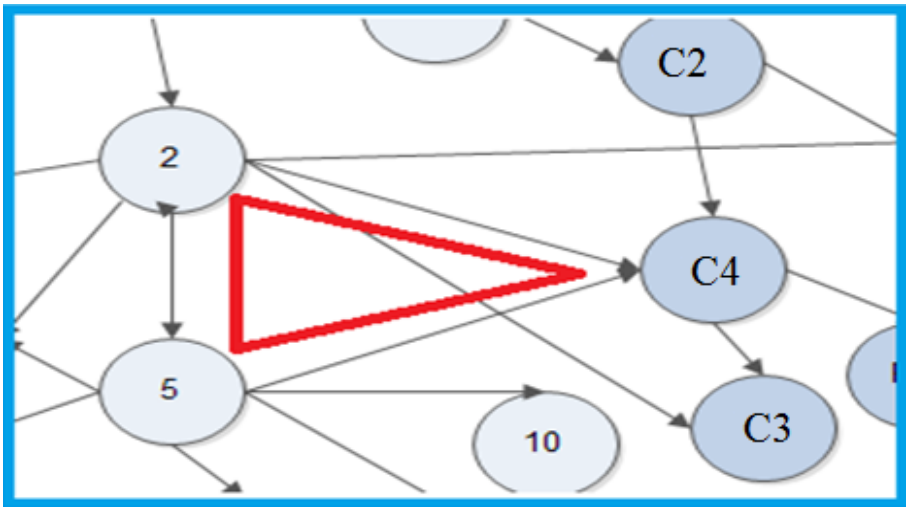


Şekil 6.1: Uygun test verisinin oluşturulması.

komşuya sahip ilk 3 komşu düğümü (4, 5, 6 düğümleri) test düğümü olarak belirlenir. Aynı işlemler geri kalan tüm düğümler için yapılır. En çok komşusu olan düğümler test düğümü olarak alındıktan sonra bir sonraki çok sayıda komşuya sahip düğümlere geçilir.

Etiketsiz düğümler seçilirken, komşu düğümlerin bir döngü (loop) oluşturmayacak şekilde seçilmesi gerekir. Eğer döngü oluştururlarsa, mesaj geçişleri sürekli o döngü içerisinde kalır. Bu durumda mesaj geçiş algoritmaları test edilemez. Şekil 6.2'de Şekil 6.1'den alınmış bir parça olarak gösterilen 2. ve 5. numaralı etiketsiz düğümler ve C_4 etiketli örnek bir düğüm bulunmaktadır. C_4 düğümünün test düğümü olarak alınması 2., 5. ve C_4 etiketli düğüm arasında bir döngü oluşturacağından C_4 etiketli düğüm test verisine dahil edilmemelidir.

Bunun dışında veri kümesinde makaleler etiketlerine göre özdeş dağılmamış olabilirler. Nitekim CORA veri kümesinde etiketlerin sayıca oranları birbirinden farklıdır. CORA'da örnek olarak NN (Neural Network) etiketinden 818 makale varken Rule Learning (RL)'den 180 makale vardır. Bu yüzden RL etiketine göre NN etiketinden daha fazla sayıda veri test verisi olarak alınmalıdır. Böylece veri kümesindeki etiket dağılım oranı bozulmamış olur. CORA veri kümesinin tamamında 2700 civarı makale vardır ve veri kümesindeki NN yoğunluğu yaklaşık %30 civarındadır. Oluşturulacak test veri kümesinde de bu oran korunacak şekilde test verisi seçme işlemi gerçekleştirilir.



Şekil 6.2 : Etiketsiz düğümler arasındaki döngülerin engellenmesi

6.3 Deneysel Sonuçlar

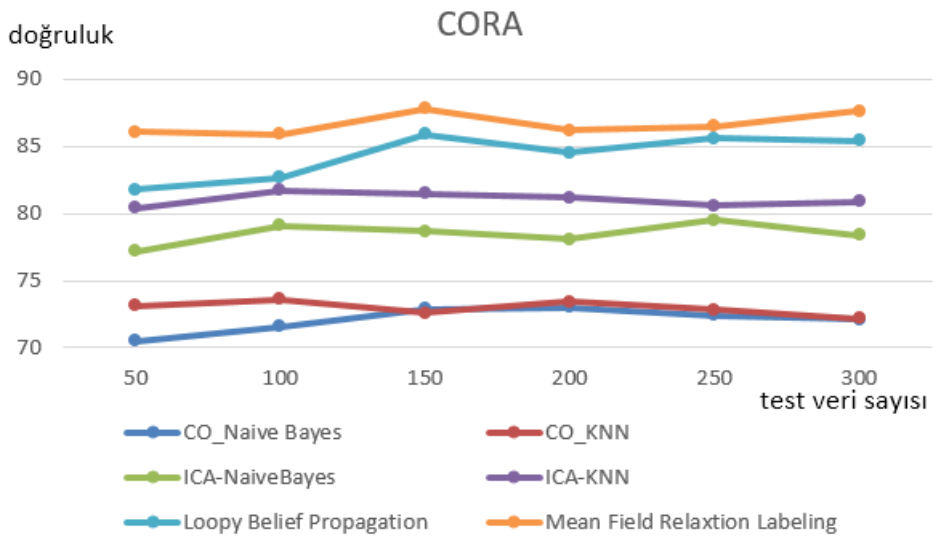
Araştırmada kullanılan her algoritma, karşılaştırılmak üzere CORA, CiteSeer, Terrorist-Rel adlı hazır veri kümeleri üzerinde uygulanmıştır. Bu veri kümelerinden CORA'da 2700, CiteSeer'de 3300 civarı makale; Terrorist-Rel'de ise 851'e yakın terörist düğüm bulunmaktadır. Test aşamasında seçilebilen maksimum test verisi 300'e kadar çıkabilmektedir. Bu da veri kümesinin içinde yaklaşık %9-%11 civarında verinin etiketsiz hale getirilmesine neden olur.

Çizelge 6.1'de 10 kat çapraz geçerleme ile elde edilen 50-300 test verisi arasında farklı test verileri için bütün algoritmaların doğruluk değerleri verilmiştir. Aynı şekilde Çizelge 6.2'de CiteSeer veri kümesinin doğruluk değerleri verilmiştir. Her iki çizelgeye göre bibliyografik veri kümelerinde global sınıflandırma algoritmalarının diğer algoritmalara göre daha iyi sonuçlar verdiği görülür. Daha sonraki yüksek doğruluk oranları yerel sınıflandırma algoritmalarına aittir. Son olarak da doğruluk oranının yanlızca içeriğe dayalı sınıflandırma algoritmalarında en az olduğu görülür. Yanlızca içeriğe dayalı sınıflandırma algoritmaların daha düşük doğruluk oranı çıkarmasının sebebi algoritmaların sadece içerik verilerini kullanıyor olmasıdır.

Çizelge 6.1 : CORA veri kümesinde doğruluk değerleri.

Algoritma\test veri sayısı		50	100	150	200	250	300
CO	CO-NB	70.5	71.6	72.9	73.0	72.4	72.1
	CO-KNN	70.1	72.6	73.5	73.4	72.8	72.2
Lokal	ICA-NB	77.2	77.1	78.7	78.1	77.5	77.4
	ICA-KNN	80.4	81.7	81.5	81.2	80.6	80.9
Global	LBP	81.8	82.7	**85.9**	84.5	85.6	85.4
	MF	86.1	85.9	**87.8**	86.2	86.5	87.6

Yerel sınıflandırma algoritmalarının, yanlızca içeriğe dayalı sınıflandırma algoritmalarından daha iyi sonuç vermesinin sebebi; yerel sınıflandırma algoritmalarının çizge verilerini, yani makalelerin birbirlerine yaptıkları atıfları da kullanıyor olmasıdır. Yerel sınıflandırma algoritmaları bu işlemi yaparken test edilen düğümlerin sadece komşularını inceler, o yüzden global sınıflandırıcılar kadar iyi sonuç vermezler. Lokal sınıflandırma algoritmalarında sadece komşu düğümler incelenirken, global sınıflandırma algoritmaları çizge veri kümeleri üzerinde MRF (Markov Rasgele Alanlar) ile ağın farklı noktalarından elde ettiği parçaları sınıflandırmada kullanır. Bu sebeple



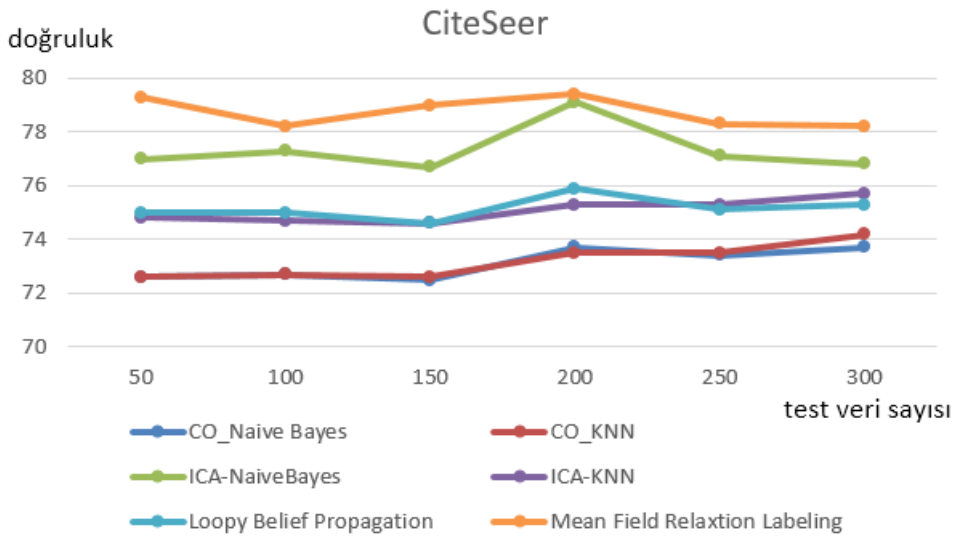
Şekil 6.3 :CORA veri kümesinde doğruluk değerleri.

global sınıflandırma algoritmalarının inceleme alanı daha geniş olduğu için daha iyi sonuçlar verir. Şekil 6.3'te CORA veri kümesinin doğruluk değerleri solda, test veri sayısı ile algoritma isimleri alt kısımda gösterilmiştir.

- CORA veri kümesinde yanlızca içeriğe dayalı sınıflandırmada en yüksek doğruluk oranı Content-Only KNN (CO-KNN)'ye aittir. CiteSeer'da ise en yüksek doğruluk CO-Naive Bayes'e aittir.
- Local Classification algoritmalarına bakılacak olursa CORA veri kümesi için en iyi sonucu ICA-KNN vermiştir. CiteSeer veri kümesinde ise ICA-Naive Bayes vermiştir.
- Global Sınıflandırma algoritmalarında en iyi sonucu her iki veri kümesi için de Mean Field Relaxation Labeling algoritması vermiştir.

Çizelge 6.2: CiteSeer veri kümesinde doğruluk değerleri.

Algoritma\test veri sayısı		50	100	150	200	250	300
CO	CO-NB	72.6	72.7	72.5	73.7	73.4	73.7
	CO-KNN	70.4	72.1	72.6	73.5	73.5	74.2
Lokal	ICA-NB	77.0	77.3	76.7	76.1	77.1	76.8
	ICA-KNN	74.8	74.7	74.6	75.3	75.3	75.7
Global	LBP	75.0	75.0	74.6	**78.9**	75.1	75.3
	MF	79.3	78.2	79.0	**79.4**	78.3	78.2



Şekil 6.4 : CiteSeer veri kümesi doğruluk değerleri.

Grafiklere göre en iyi sonucu global sınıflandırma algoritmalarından MF verirken, en düşük sonucu sadece içerik tabanlı algoritmalar vermiştir. Şekil 6.4'te CiteSeer veri kümesinin doğruluk değerleri solda, test veri sayısı ile algoritma isimleri ise alt kısımda gösterilmiştir

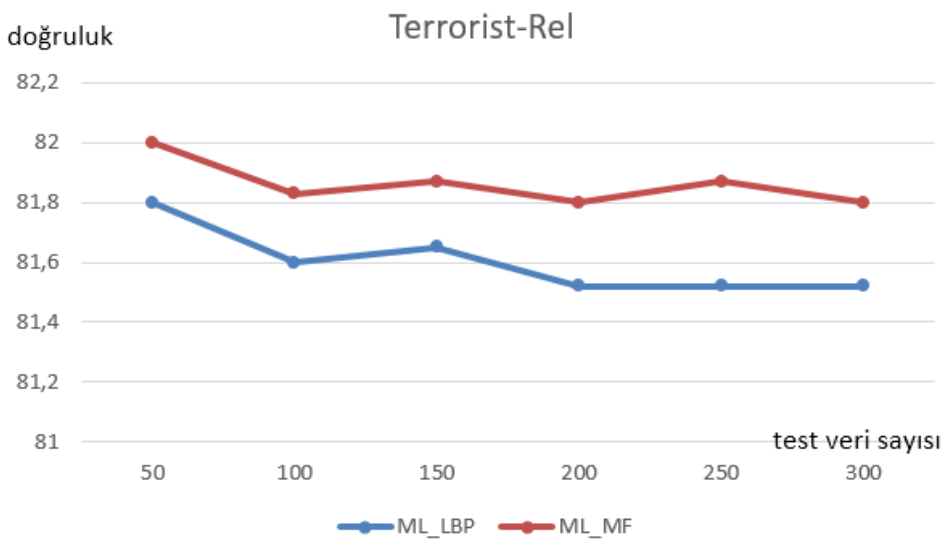
Çizelge 6.3'de çoklu-etiketli veri kümesi olan Terrorist-Rel 'in doğruluk değerleri verilmiştir. Yine aynı şekilde 50-300 arasında test verisi oluşturulmuştur. Çoklu-etiketli sınıflandırma yapılabilmesi için uygulanan ML-MF ve ML-LBP algoritmaları karşılaştırılmıştır.

Şekil 6.5'te Terrorist-Rel veri kümesinin doğruluk değerleri solda, test veri sayısı ile algoritma isimleri alt kısımda gösterilmiştir.

- ML-LBP ve ML-MF doğruluk oranları birbirine yakın çıkmıştır.
- ML-MF algoritması test kümelerinde ML-LBP algoritmasından daha iyi sonuçlar vermiştir.

Çizelge 6.3: Terorrist-Rel veri kümesinde doğruluk değerleri.

Algoritma\test veri sayısı	50	100	150	200	250	300
ML-LBP	81.8	81.6	81.7	81.4	81.5	81.5
ML-MF	**82.0**	81.9	81.9	81.7	81.8	81.8



Şekil 6.5 : Terrorist-Rel veri kümesi doğruluk değerleri.

- Hem tekli-etiketli veri kümeleri, hem çoklu-etiketli veri kümeleri için MF algoritması LBP algoritmasından daha iyi sonuçlar vermiştir.
- Tekli-etiket sınıflandırmalarında kullanılan CORA veri kümesi CiteSeer veri kümesine göre daha iyi doğruluk sonucu vermiştir. Bunun sebebi ise CORA veri kümesindeki bağlantı bilgisinin CiteSeer'den daha fazla olmasıdır [23].

6.4 Sınıflandırıcıların Birleştirilmesi ve Doğruluk İyileştirmesi

Araştırmada tekli-etiketli veri kümelerinde kullanılan birbirinden faklı sınıflandırıcılar birleştirilerek doğruluk değelerinin arttırılması amaçlanmıştır. ICA-Naive Bayes, ICA-KNN, Mean Field Relaxation Labeling algoritmaları birleştirilerek k-katlı çarpaz doğrulama ile sınanan test verileri toplam etiket puanına ya da oranına bakılarak iki farklı yönden incelenmiştir.

Sınıflandırıcı birleştirmede kullanılan "Çoğunluklu Seçim (Majority Voting-MV)" ile her 3 algoritma sonucunda etiketsiz bir düğüme atanacak olası etiket değeri hangi etiket için fazlaysa, etiketsiz düğüme o etiket değeri atanır. Bir etiketsiz düğüm için her 3 algoritma da farklı etiket değeri çıkarırsa, bu durumda doğruluk değeri en büyük olan Mean Field Relaxation (MF) algoritmasının sonucu etiketsiz düğüme etiket değeri olarak atanır.

Sınıflandırıcı birleştirmede kullanılan diğer bir işlem ise "Ağırlıklı Çoğunluk Seçimi (Weighted Majority Voting-WMV)"'dir. Bu yöntem ile herbir algoritmanın tüm etiket değerleri için verdiği olasılık sonuçları çarpılır. Çarpım sonucu en yüksek çıkan etiket

değeri etiketsiz düğüme atanır. Çizelge 6.4'te sınıflandırıcıların birleştirilmesi sonucu oluşan doğruluk değerleri verilmiştir. Bu değerlerden çıkarılan sonuçlar aşağıdaki gibidir:

- Sınıflandırıcı birleştirme yöntemleri, birden fazla sınıflandırıcının bir etiketsiz düğüm için ortak kararını kullandığından doğruluk değerlerini arttırmıştır.
- Doğruluk değerleri "WMV" işleminde "MV" işleminden çok az farkla daha iyi sonuçlar vermiştir.
- CORA veri kümesindeki doğruluk değerleri sınıflandırıcılar birleştirildikten sonra yine CiteSeer'den yüksek çıkmıştır.

Çizelge 6.4: Sınıflandırıcıların birleştirilmesi.

Veri Kümesi	İşlem	50	100	150	200	250	300
Cora	MV	85.3	84.5	85.4	86.3	84.3	84.8
	WMV	87.3	87.7	87.3	87.5	**87.9**	87.4
CiteSeer	VM	82.3	81.8	80.6	78.9	79.2	79.4
	WMV	**84.1**	82.6	81.8	81.2	80.6	80.9

Şekil 6.6'da MV işleminde her bir Y_i etiketsiz düğümü için birleştirmede kullanılacak her bir P_t sınıflandırıcısı tarafından birer etiket değeri atanır. Sınıflandırıcılar hangi C_j etiket sonucunu verdiyse C etiket listesinde o etiket değeri için puan verilir. En çok puana sahip etiket değeri, etiketsiz test düğümüne atanır.

1: **for each** $Y_i \in$ Y her bir etiketsiz düğüm için
2: **for each** $P_t \in$ P her bir sınıflandırıcı için
  i'inci etiketsiz düğümün P_t sınıflandırıcısı ile olası
  C_k etiketini bul
3: **for each** $C_j \in$ C tüm etiket değerleri için C_k hangi etikete karşılık geliyorsa bir puan arttır
  if $(C_k == C_j)$
    $++ (C_j)$ // Majority Voting (MV)
4: **end for**
5: **end for**
6: $C = max(C_j)$ // ele alınan test düğümü için en çok hangi etikete oy verildiyse onun etiketi değeri atanır.
7: **end for**

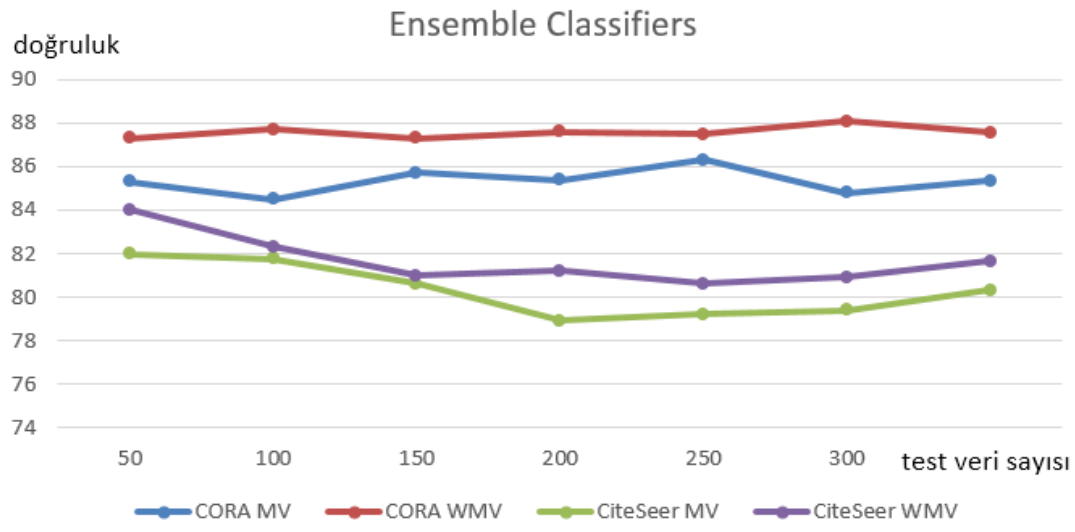
Şekil 6.6 : MV işlemine göre sınıflandırıcı birleştirme

Şekil 6.7'de WMV işleminde her bir Y_i etiketsiz düğümüne birleştirmede kullanılacak bir P_t sınıflandırıcısı tarafından herbir etiket değeri için bir olasılık değeri atanır. Sonrasında her bir etiket değer için tüm sınıflandırıcılardan elde edilen olasılık değerleri çarpılır.

1: **for each** $Y_i \in$ Y her bir etiketsiz düğüm için
2: **for each** $L_t \in$ L her bir sınıflandırıcı için
3: i'inci etiketsiz düğümün L_t sınıflandırıcısı ile olası C_k etiketini bul
2: **for each** $C_j \in$ C // tüm etiket değerleri için C_k hangi etikete //karşılık geliyorsa onun olasılık değerini L_t sınıflandırıcısının // K_{L_t} doğruluk katsayısı ile çarp
if $(C_k == C_j)$
$\mathrm{P}(C_j) = \mathrm{P}(C_j) \times K_{C_t}$ // voting *WMV*
5: **end for**
6: **end for**
4: $C = max(C_j)$ //etiketsiz test düğümüne en çok hangi etiket için oy verildiyse onun etiket değeri atanır.
6: **end for**

Şekil 6.7 : WMV işlemine göre sınıflandırıcı birleştirme.

En yüksek çarpım değerine sahip etiket değeri etiketsiz test düğümüne atanır. Şekil 6.8'de sınıflandırıcı birleştirilmesi sonucu oluşan doğruluk değerleri grafiksel, Çizelge 6.4'te ise birleştirme algoritmalarının doğruluk değerleri rakamsal olarak gösterilmiştir.



Şekil 6.8 : Sınıflandırıcılar birleştirildikten sonraki doğruluk değerleri

6.5 Sonuçlar ve Tartışmalar

Bu çalışma kapsamında yerel ve global kolektif sınıflandırma yöntemleri incelenmiş ve tekli ve çoklu-etiket içeren üç farklı türdeki veri kümesi üzerinde sınıflandırma başarımları karşılaştırılmıştır. Sonrasında bu yöntemlerin performanslarının ve doğruluk değerlerinin arttırılabilmesi için sınıflandırıcı birleştirme yöntemleri veri kümelerine uygulanmıştır.

Yanlızca içeriğe dayalı sınıflandırmada doğruluk değerleri %70 oranındayken çizge veri kümeleri de sınıflandırıcılara dahil edildiğinde doğruluk değeri %80-%85 aralığında artış göstermiştir. Kolektif sınıflandırma algoritmaları birleştirildiğinde ise doğruluk değerlerinin %85 ve üzerine çıkabildiği gözlemlenmiştir. Böylece veri kümelerine uygulanan sınıflandırıcı birleştirme algoritmalarının doğruluk değerlerini arttırdığı saptanmıştır.

Hem CiteSeer hem de CORA veri kümelerinde global sınıflandırıcılar, lokal sınıflandırıcılardan daha iyi sonuçlar vermiştir. Bunun en önemli sebebi global sınıflandırıcıların ikili MRF'den faydalanarak düğümler arası mesaj geçişlerini sınıflandırmada kullanabilmesidir.

Literatürde varolan tekli-etiketli veri kümelerine uygulanan kolektif sınıflandırma algoritmalarında doğruluk değerlerinin arttırılmasının yanı sıra çoklu-etiketli veri kümesi olan Terrorist-Rel veri kümesine de kolektif sınıflandırma algoritmaları uygulanmış ve %82 oranında seyreden doğruluk değerleri ile başarılı sonuçlar elde edilmiştir.

KAYNAKLAR

[1] **Prithviraj, S., Galileo, N., Mustafa, B., Lise, G., Brian, G., and Tina E.** (2008) Collective Classification in Network Data, AI Mag., vol. 29, pp. 93–106.

[2] **Eswaran, D., Gunnemann, S., Faloutsos, C.** (2017) ZooBP Belief Propagation for Heteronegeneous Networks, Proceedings of the VLDB Endormewment 10.5.

[3] **Saito, M., Okatani, T., Deguchi, K.** (2012) Application of the Mean Field Methods to MRF Optimization in Computer Vision, Tohoku University, Japan.

[4] **Zahoa, B.** (2006) Event classification and relationship labeling in affiliation networks. In Proceedings of the Workshop on Statistical Network Analysis, (SNA) at the 23rd International Conference on Machine Learning (ICML).

[5] **Lise, G., Christopher, P.** (2005) Link Mining: A Survey, University of Maryland, College Park, MD.

[6] **Jafar, A., Hans, C., Eric, M., Andre, V.** (2004) The KOJAK Group Finder: Connecting the Dots via Integrated Knowledge-Based and Statistical Reasoning, USC Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292.

[7] **Coffman, T, Marcus, S.** (2004) Pattern Classification in Social Network Analysis: A Case Study, Proc. IEEE Aerospace Conf., March 6-13,Big Sky, MT.

[8] **Wan, J., Moy, M., Darr, T.,Coffman, T., Snyder, J., Hollinger, M.,Thomason, B.**, (2006) Key Elements of an Evidence Detection System, AAAI Fall Symposium on Capturing and Using Patterns for Evidence Detection, pp. 62-67, Oct 13-15, 2006.

[9] **Ager, M., Danvy, O., Rohde, H.** (2003) Fast Partial Evaluation of Pattern Matching in Strings, Department of Computer Science University of Aarhus, April, Denmark Copenhagen, pp 9-10.

[10] **Lei, T.** (2016) Community Detection in Social Networks, The University of Texas at Dallas, December, pp 18-21.

[11] **Raymond, H., Murat, K., Bhavani, T.** (2009) Social Network Classification Incorporating Link Type Values The Proceedings of IEEE International Conference on Intelligence and Security Informatics (ISI 2009), pp. 19-24, June 8-11 2009, Dallas, Texas.

[12] **Kajdanowicz, T., Kazienko, P., Janczak, M.** (2013) Collective Classification Techniques : an Experimental Study*,New Trends Databases …*, pp. 1–10.

[13] **Faryal, G., Haider, B., Usman, Q.** (2014) Terrorist Group Prediction Using Data Classification,Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition, 17-19 November, Kuala Lumpur, Malaysia,

[14] **Cataltepe, Z., Sonmez, A., Basoglu, K., Erzan, A**. (2011) Collective Classification Using Heterogeneous Classifiers, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

[15] **Jicy, G., Sandhya, N., Suja, G.** (2014) Classification Problem in Text Mining, International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163 Volume 1 Issue 8 September, Bangalore.

[16] **Karsten, W., Johannes, F., Anette, F.** (2014) Anette F. A Machine Learning Approach for Coreference Resolution , 12 December, Darmstadt Germany.

[17] **K. Ming Leung** (2007) Finance and Risk Engineering Naive Bayesian Classifier, Polytechnic University Department of Computer Science, 28 November, Milan Italy.

[18] **Jun, K., Yutaka, M., Hikaru, Y., Mitsuru, I.** (2007) Generating Social Network Features for Link-based Classification, European Conference on Principles of Data Mining and Knowledge Discovery pp 127-139.

[19] **Kırcı, A., Boyacı, O.** (2011) Sosyal Ağların Web Madenciliği Teknikleri ile Analizi ve Ortak Atıf Analizi ile Analizi ile Benzerlik Tahimini, İnönü Üniversitesi.

[20] **Finley, T., Joachims, T.** (2007) Parameter Learning for Loopy Markov Random Fields with Structural Support Vector Machines Proceedings of the ICML Workshop on Constraint Optimization and Structured Output Spaces, Oregon State University, 24 June, USA.

[21] **Kong, X., Shi, X., Philip Y** (2011) Multi-Label Classification Collective, Proceedings of the 2011 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 28-30 April.

[22] **Senliol, B., Cataltepe, Z.** (2010) Kolektif Sınıflandırma Yöntemleri için Öznitelik ve Düğüm Seçimi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

[23] **Ataseven, O., Yaslan, Y.** (2017) A Comparison of Collective Classification Techniques on Network Data, Information Systems and Technologies (CISTI), 2017 12th Iberian Conference, Portugal, 21-24 June.

ÖZGEÇMİŞ

Ad-Soyad **: ÖZGE ATASEVEN**

Doğum Tarihi ve Yeri **: 01.03.1989**

E-posta **:** atasevenoz@itu.edu.tr

ÖĞRENİM DURUMU:

- **Lisans** **:** 2012, İstanbul Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Yükseklisans** **:** İstanbul Teknik Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2007 Mehmet Niyazi Altuğ Anadolu Lisesi Okul Birinciliği.
- 2012 İstanbul Üniversitesi Yüksek Onur Derecesi Mezuniyet.
- 2012 Garanti Teknoloji Yazılım Mühendisi.
- 2014 Anadolu Hayat Emeklilik BT Uzmanı.

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN YAYINLAR, SUNUMLAR :

- **O. Ataseven, Y. Yaslan,** 2017. A Comparison of Collective Classification Techniques on Network Data, Information Systems and Technologies (CISTI), 2017 12th Iberian Conference, Portugal.