

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

ÇİZGE TABANLI ÖRNEKLERDE
KULLANICI DESTEKLİ
VERİ MADENCİLİĞİ

BURCU YILMAZ ŞENSOY
DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMANI
YRD. DOÇ. DR. MEHMET GÖKTÜRK

GEBZE

2010

ÖZET

TEZİN BAŞLIĞI : ÇİZGE TABANLI ÖRNEKLERDE KULLANICI DESTEKLİ VERİ MADENCİLİĞİ

YAZAR ADI : BURCU YILMAZ ŞENSOY

Çizge veri kümelerinin kullanımının artmasıyla, çizgelerde veri ve örüntü madenciliği alanına araştırmacıların ilgisi artmaktadır. Özellikle de, çizgelerde sıklıkla tekrarlayan örüntülerin bulunması uygulama alanlarının genişliği ve çeşitliliği sebebiyle oldukça ilgi toplamaktadır. Bu tez çalışmasında, söz konusu alanda yapılmış çalışmalar incelenmiş ve eksikliklerinden bahsedilmiştir. Bu eksikliklerin bir kısmını giderecek ve daha başarılı sonuçlar elde edilmesine olanak sağlayacak çizge veri kümelerinde kullanıcı destekli örüntü madenciliği için yeni bir yaklaşım önerilmiştir.

Önerilen yaklaşımın temeli çizge veri yapıları ile ifade edilen veri kümelerinin özellik uzayına aktarılmasına dayanmaktadır. Bu sayede benzer örüntüleri bulmak amacıyla veri kümesi üzerinde çeşitli veri madenciliği yöntemlerinin uygulanmasına olanak sağlanmaktadır. Ayrıca önerilen yaklaşım, otomatik parametre seçimindeki dezavantajları görsel destekli veri madenciliği yöntemlerinin avantajlarını kullanarak azaltmaktadır. Ara adımlarda elde edilen bilgileri grafikler ve istatistiksel bilgiler olarak uzmanlara sunmaktadır. Böylece uzmanların veri kümesi ve ara adımlardaki sonuçlar hakkında bilgi sahibi olması sağlanmakta, gerekli yerlerde parametre seçiminde veya otomatik hesaplanan parametrelerde ince ayar yapılmasına olanak tanınmaktadır.

Önerilen yaklaşım ile çizge veri kümelerinin, üç boyutlu özellik uzayına izdüşümü alınır. Bir çizgedeki her bir kenar ve kenarın iki ucundaki düğümler özellik uzayında bir nokta olarak gösterilir. Aynı sınıfa ait tüm çizgeler noktalar şeklinde özellik uzayına dönüştürüldüğünde, benzer özellikte tekrar eden yapılar nokta küme gruplarını

oluşturur. Tekrarlayan örüntüler, kümeleme yöntemleri ile çıkarılır. Çok geniş veri kümeleri ile çalışıldığında tüm çizgelerin aynı özellik uzayına aktarılması nokta kümelerinin elde edilmesini zorlaştırmaktadır. Özellik uzayındaki tekrar eden alt çizgeleri ortaya çıkarmak için örüntüler haricindeki diğer kenarlar histogram tabanlı görsel veri madenciliği yöntemleri ile filtrelenir. Tüm bu işlemler boyunca uzmanlar gerekli gördüğü yerlerde (ör: filtreleme, kümeleme) grafikleri inceleyerek gerektiğinde parametre değerlerinde ince ayar yapabilirler.

Bu tez çalışmasında önerilen yaklaşım, gerçek bir veri kümesinde ve sentetik veri kümelerinde ayrıntılı olarak test edilmiştir. Her veri kümesinde iki sınıf vardır, her sınıf az miktarda gürültü ile birlikte aynı özellikteki örüntülere sahip çizgeleri içermektedir. Detaylı testlerden elde edilen sonuçlara göre, gürültüsüz veri kümelerinde önerilen yöntemin, örüntüleri %100 başarılı olarak bulduğu görülmüştür. Gürültülü veri kümelerinde ise ortalama başarı %95 olarak gerçekleşmiştir. Literatürdeki diğer yöntemler gürültüsüz ortamda başarılı bir şekilde örüntüleri bulurken, gürültülü veri kümelerinde örüntüleri çoğunlukla bulamamışlardır.

SUMMARY

**THESIS TITLE : USER ASSISTED DATA MINING IN
GRAPH BASED DATASETS**

AUTHOR NAME : BURCU YILMAZ ŞENSOY

With the increasing use of graph datasets, pattern and data mining in graphs are attracting more attention of the researchers. Especially, finding frequently repeating patterns in graphs is increasingly drawing more attention, because of its wide and versatile application areas. In this dissertation, previous studies on frequent pattern mining in graphs are analyzed and their deficiencies are discussed. To overcome some of these deficiencies and to allow more successful results in graph data sets, a user-assisted pattern mining method is proposed.

The proposed approach is based on the transformation of graph data structures into a feature space. In this way, various data mining methods can be applied to the data to find similar patterns. Proposed approach eliminates the disadvantages of the automatic parameter selection methods using the advantages of visual data mining methods. The information obtained through intermediary steps is represented to the experts as graphs and statistical data. Hence, the experts are informed about the dataset and steps of the process; moreover, the selection or fine-tuning of computed parameters by the experts are permitted.

With the proposed approach, graph representations of structural data are projected into a 3D feature space. Each graph piece, which is a triple composed of an edge and two nodes at each end, is represented as a point in the feature space. Once we represent each graph in the dataset as a set of points in the feature space, using a clustering algorithm, we determine clusters of points that have similar characteristics. However, discovering the clusters is non-trivial because of the noisy points that

represent infrequent bonds. At this stage, noise is filtered using a histogram-based visual data mining method, so that the clusters can be discovered more clearly by various clustering algorithms. Once the clusters are discovered, frequent sub-structures are computed. During all these steps, a domain expert can intervene to guide the system or fine-tune the parameters if necessary (e.g., during filtering, clustering and so on).

The approach proposed in the dissertation is tested in detail with a real-life dataset and various synthetic datasets. Datasets may include samples belonging to two classes, where each class contains graphs with some common patterns and properties with some noise. According to detailed analysis of our test results, the proposed approach can find frequent patterns with 100% success when noise is not included to the datasets. The average success in the noisy datasets is 95%. While other methods in the literature find the patterns successfully only when noise is not included to the datasets, they couldn't successfully find patterns in the noisy data sets.

TEŞEKKÜR

Tez çalışmam boyunca gösterdiği her türlü desteği, teşviki ve engin sabrı için danışman hocam Yrd. Doç. Dr. Mehmet Göktürk'e içtenliğimle teşekkür ediyorum.

Tez izleme komitesinde bulunmayı kabul edip tezime verdikleri desteklerden, bana zaman ayırıp tezimin ilerlemesinde katkıda bulundukları için Dr. Yusuf Sinan AKGÜL, Dr. Ertuğrul SAATÇI'ya şükranlarımı sunarım.

Tez savunma komitesinde bulunmayı kabul edip çalışmalarına verdikleri desteklerden dolayı Dr. İbrahim SOĞUKPINAR'a, Dr. Erdoğan SEVİLGİN'e teşekkürü borç bilirim.

Gebze Yüksek Teknoloji Enstitüsünde ve İstanbul Kültür Üniversitesinde yardımlarını esirgemeyen bütün hocalarıma ve araştırma görevlisi arkadaşlarıma teşekkür ediyorum.

Tez çalışmama verdiği emekler için Dr. Natali DİMOĞLU'ya teşekkür ederim.

Yoğun akademik çalışmaları arasında zaman ayırarak doktora ve tez süresince desteğini esirgemeyen Dr. Eşref ADALI'ya şükranlarımı sunarım.

Eşim Dr. Murat ŞENSOY'a, babam Erdoğan YILMAZ'a, annem SEVİM YILMAZ'a ve kardeşim Av. Merve YILMAZ'a destekleri ve anlayışları için teşekkür ediyorum.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZET	i
SUMMARY	iii
TEŞEKKÜR	v
İÇİNDEKİLER DİZİNİ	vi
SİMGELER VE KISALTMALAR DİZİNİ	ix
ŞEKİLLER DİZİNİ	x
ÇİZELGELER	xiv
1 GİRİŞ	1
2 LİTERATÜR ANALİZİ	6
2.1 Çizge Tabanlı Örüntü Madenciliği	6
2.2 Çizge Veri Kümesinden Tekrarlayan Altçizgelerin Çıkarılması	7
2.2.1 Problemin Tanımı	7
2.2.2 Yöntemler	9
2.2.3 Uygulama Alanları	17
2.2.3.1 Yapı Aktivite İlişkileri (SAR)	18
2.2.3.2 Bilinen QSAR uygulamaları	20
2.2.3.3 Genel Yapısal Yaklaşım Uygulamaları	21
2.3 Tek Bir Çizgede Tekrarlayan Altçizge Madenciliği Problemi	22
2.3.1 Problem Tanımı	22
2.3.2 Yöntemler ve Uygulama Alanları	26
2.3.2.1 Prosedürler ile ilgili Özet Çıkarma	26
2.3.2.2 Sosyal Ağlar, Biyolojik Ağlar ve Web Uygulamaları	26
2.4 Görsel Veri Madenciliği	32
2.5 Videolardaki Nesne Eşleme	33
2.6 Hiyerarşik Kümeleme Algoritmaları	35
2.7 Benzerlik Ölçüsü	37
2.8 Çizgelerin Düğümleri Arasında Benzerlik Ölçümü	38
3 PROBLEMİN TANIMI VE HİPOTEZ	40

3.1 Çizge Tabanlı Veri Kümelerinde Tekrarlayan Altçizgelerin Çıkarılması	40
3.2 Çizgede İzomorfizm, Altçizge İzomorfizm ve Destek Değeri Kavramı	44
3.3 Tekrarlayan Benzer Altçizge Madenciliği	46
3.4 Kullanıcı Desteğinin Örüntü Çıkarma Sistemlerindeki Önemi	49
3.5 Hipotez	51
4 YÖNTEM	52
4.1 Veri Kümesi	52
4.2 Çizge Veri Yapısından 3B Özellik Uzayına Dönüşüm	53
4.3 Örüntü Kümelerinin Belirlenmesi	55
4.4 Örüntü Kümelerinden Örüntülerin Çıkarılması	59
4.5 Filtrelemenin Gerekliliği	59
4.6 Görsel Veri Madenciliği Yöntemi	64
4.6.1 Kesişen Yatay Ve Dikey Doğrular Yöntemi	65
4.6.2 Görsel Veri Madenciliğinde Gri Ölçek Gölgeleme Yöntemi (GSS)	66
4.6.3 HSV Ölçek Renklendirme	70
4.7 Veri Ardışık Düzenegi Yapısı	71
4.8 Hibrit Yöntem	75
4.9 Kullanıcının Rolü	76
5 DEĞERLENDİRME	78
5.1 Yöntemlerin Gerçek Hayat Veri Kümeleri Kullanılarak Karşılaştırılması	78
5.2 Sentetik Veri Kümesi Oluşturulması	80
5.3 Yöntemin Sentetik Veri Kümelerinde Test Sonuçları	82
5.3.1 Kullanılan Parametreler	83
5.3.2 Literatürdeki Yöntemler ve Benzerlik Kriteri	93
5.3.3 Gürültünün Etkisi	95
5.3.4 Alfa Benzerlik Eşik Değeri ve Destek Değeri	102
6 SONUÇLAR	105

7 GELECEK ÇALIŞMALAR	107
7.1 Çizgelerin Sınıflandırılmasında Benzer Alt Çizgelerin Özellik Olarak Kullanımı	107
7.2 Örüntü Haritalarından Filtreleme Eşik Değerinin Çıkarılması	110
KAYNAKLAR	115
ÖZGEÇMİŞ	122

SİMGELER VE KISALTMALAR DİZİNİ

Kde	: Yoğunluk tahmin fonksiyonu
ÇK-TAÇ	: Çizge Kümesinden Tekrarlayan Alt çizgelerin Çıkarılması
Ç-TAÇ	: Çizgeden Tekrarlayan Alt çizgelerin Çıkarılması
GSS	: Görsel Veri Madenciliğinde Gri Ölçek Gölgeleme Yöntemi
HSVSV	: HSV Ölçek Renklendirme Yöntemi
IBE	: İnsan bilgisayar etkileşimi
HSV	: Hue Saturation Value
KED	: Kümeleme Eşik Değeri
FED	: Filtreleme Eşik Değeri
EDED	: En düşük eşik değeri
EYED	: En yüksek eşik değeri

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1 Örnek bir çizge veri kümesi.	8
2.2 Şekil 1 deki örnek veri kümesinin alt çizgelerinin kafes yapısı ile	11
2.3 Cylin bileşiğinin alt bileşikleri [Fischer and Meinl, 2004].	11
2.4 (a) çizgesi ve çizgenin birbiri ile izomorfik üç DFS ağaç örneği (b-d).	14
2.5 Örnek bir DFS ağaç yapısı [Yan and Han, 2002].	14
2.6 $\{a, b, c, d, e\}$ kümesinin elemanları ve Yapısal Daraltma Yöntemi ile oluşturulmuş bir ağaç yapısı örneği [Borgelt and Berthold, 2002].	16
2.7 Karşılaştırılmalı Alan Analizi örneği.	21
2.8 Biyomolekülün içine ilacın indirgenme örneği.	21
2.9 Örnek bir ARG çizgesi ve elde edilen örüntü [Coffman et al., 2004].	23
2.10 Örnek bir çizge ve elde edilen örüntü [Kuramochi and Karypis, 2005].	24
2.11 Bir derleyici kodu ve kodun veri akışı.	26
2.12 Terörist ağları [Mukherjee and Holder, 2004].	27
2.13 Tavsiye ağından elde edilen çeşitli alt çizgeler [Leskovec et al., 2006].	29
2.14 WEB veri kümesinden elde edilen örüntü örnekleri [Kuramochi and Karypis, 2005].	30
2.15 Düzenli (a) ve düzensiz (b) ağ yapıları [Hinneburg et al., 1999].	33
2.16 a) Bir boyutlu renk gösterimi, b) bir boyutlu histogram yoğunluk gösterimi. c) iki boyutlu renk histogramı gösterimi.	33
2.17 İngilizcedeki “likely” kelimesinden elde edilen yakınlık çizgesi [Blondel et al, 2004].	39
3.1 N düğümlü bir çizge örneği.	41
3.2 G_p ve G_q çizgeleri için alt çizge izomorfizm örneği.	45
3.3 G_p ve G_q çizgeleri için alt çizge benzerlik örneği.	47
4.1 Çizgeden özellik uzayına dönüşüm.	54

4.2	Özellik uzayında görülen örnek kümeler.	56
4.3	Sadece tek bir özelliğe ait filtrelenmiş verinin 3 boyutlu koordinat sisteminde gösterilmesi.	58
4.4	Filtrelenmiş veriden elde edilen aday örüntü kümeleri.	58
4.5	Özellik uzayında görülen örnek kümeler.	60
4.6	Çizge veri kümesindeki çizgelerin işlenmemiş olarak özellik uzayındaki görüntüsü (Her çizgesi ortalama 40 düğümlü iki sınıfa ait toplam 100 çizgeden oluşan bir veri kümesinden elde edilmiştir).	61
4.7	Dört çizgeli bir veri kümesi örneği ve elde edilen kümeler (Durum 1).	62
4.8	Filtrelenmemiş veri ve elde edilmiş kümeler.	63
4.9	Filtrelenmiş veri ve elde edilmiş kümeler.	63
4.10	Tüm aktif moleküller için a) kesişen yatay ve dikey doğrular yöntemi gösterimi b) ETMC matrislerinin köşegen olmayan değerler için molekül yüzdeleri gösteren histogram c) ETMC matrislerinin köşegen değerleri için molekül yüzdeleri gösteren histogram.	66
4.11	İlk tekniği kullanarak çizge resimlerinin oluşturulması.	68
4.12	İkinci tekniği kullanarak resimlerin oluşturulması.	68
4.13	Dört komşu hücre.	69
4.14	İlk tekniği kullanarak yakınlık matrisinden iki boyutlu resimlere kadar uygulanan aşamalar.	69
4.15	Aynı sınıfa ait çizge tabanlı 10 örneğin örüntü yoğunluk haritası.	70
4.16	HSVSV yöntemi kullanılarak elde edilmiş bir örüntü haritası.	72
4.17	Tezde önerilen yöntemin veri ardışık düzeneği yapısı.	73
5.1	Standard sapmanın (σ) gauss olasılık dağılımına etkisi; (a) $\sigma = 1$, (b) $\sigma = 3$.	82
5.2	Delta değerlerinin seçiminin zamana (sn.) etkisi ($\Delta_x = \text{Delta}$ ve $\Delta_y = 4 \times \text{Delta}$ için).	84
5.3	Çeşitli filtreleme eşik değerlerinin başarıya etkisi. (a): Filtreleme eşik değerinin P ve R değerlerine etkisi. (b): Filtreleme eşik değerinin değişimi	

için doğru pozitif oranının yanlış pozitif oranına göre değişim grafiği.	85
5.4 Filtreleme eşik değerinin zamana (sn.) göre değişimi.	86
5.5 Kümeleme eşik değerinin başarıya etkisi. (a): Kümeleme eşik değerinin P ve R değerlerine etkisi. (b): Kümeleme eşik değerinin değişimi için doğru pozitif oranının yanlış pozitif oranına göre değişim grafiği.	88
5.6, En düşük eşik değeri ve en yüksek eşik değerinin değişiminin başarıya etkisi. (a): EDED ve EYED nin P ve R değerlerine etkisi. (b): EDED ve EYED değişimi için doğru pozitif oranının yanlış pozitif oranına göre değişim grafiği.	89
5.7 Elde edilen veri kümesindeki kümeler (Kümelere ait istatistiksel bilgiler çizelge 5.2 de listelenmektedir.).	90
5.8 Gürültü _m parametresinin değişimine göre en yüksek P-R değişimi grafiği.	92
5.9 Gürültü _m parametresinin değişimine göre en düşük P-R değişimi grafiği.	92
5.10 Gürültü _m parametresinin değişimine göre ortalama P-R değişimi grafiği.	93
5.11 Gürültü _m parametresinin değişimine göre P ve R deki standart sapma.	93
5.12 Benzer iki çizge örneği.	94
5.13 Subdue yaklaşımında kenar ve düğümlere gürültü _o parametresinin başarıya etkisi.	96
5.14 Subdue yönteminin çeşitli gürültü _o değerleri için doğru pozitif oranının yanlış pozitif oranına değişim grafiği.	97
5.15 100 çizgeli bir veri kümesinde çeşitli gürültü _o değerlerinin zamana (sn.) etki grafiği.	98
5.16 Önerilen yöntem için veri kümesindeki çizge sayısındaki artışın gürültü _o değerleri 0 ve 1 olduğunda zamana (sn.) etkisi.	99
5.17 Subdue için veri kümesindeki çizge sayısındaki artışın gürültü _o değerleri 0 ve 1 olduğunda zamana etkisi.	100
5.18 100 çizgeden oluşan gürültüsüz bir veri kümesinde FFSM, Mofa, Gaston, gSpan, önerilen yöntem ve Subdue nun ortalama harcadığı zaman	

(sn).	100
5.19 Gauss dağılımı ile rastgele oluşturulmuş gürültü _m parametresinin değişiminin P değerine etkisi (gürültü _o =1 için).	101
5.20 Gauss dağılımı ile rastgele oluşturulmuş gürültü _m parametresinin değişiminin R değerine etkisi (gürültü _o =1 için).	102
5.21 Önerilen yöntem ile Subdue nun başarısının doğru pozitif oranının yanlış pozitif oranına değişim grafiği.	102
5.22 Önerilen yöntem için alfa değerinin değişiminin destek değerine etkisi.	103
5.23 Önerilen yöntem için alfa değerinin değişiminin başarıya etkisi.	104
7.1 Tek bir sınıfa ait örnekler için bir örüntü haritası örneği.	111
7.2 Tek bir sınıfa ait örnekler içeren örüntü haritasındaki renklerin 1 boyutlu nokta dağılımı olarak gösterilmesi.	112
7.3 Tek bir sınıfa ait örnekler içeren örüntü haritasındaki renklerin 1 boyutlu histogram ile gösterilmesi.	112
7.4 Tek bir sınıfa ait örnekler içeren örüntü haritasındaki renklerin 1 boyutlu histogram ile gösterilmesi.	113
7.5 Tek bir sınıfa ait örnekler içeren örüntü haritasındaki renklerin yoğunluğunun kde fonksiyonu ile gösterilmesi.	113
7.6 Tek bir sınıfa ait örnekler içeren örüntü haritasındaki renklerin yoğunluğunun kde fonksiyonu ile gösterilmesi.	114

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1 Sıklıkla tekrarlayan alt çizgelerin bazıları ve onların destek yüzdesi değeri.	8
2.2 Şekil b-d deki DFS çizge kodları [Yan and Han, 2002].	14
2.3 ABD de ilacın hazırlanması için gereken harcamaların yıllara göre dağılımı milyon \$ cinsinden görülmektedir.	19
2.4 Kümeleme yöntemlerinin zaman karmaşıklığı ve yüksek boyutlu veri kümeleri açısından değerlendirilmesi [Xu and Wunsch, 2005]. (D: Veri kümesinin boyut sayısı, N: Veri kümesindeki eleman sayısı, K: Küme sayısı)	36
5.1 Tezde önerilen yöntem, SUBDUE, FSG yöntemlerinin R ve P değerlerine göre performansı.	80
5.2 Şekil 5.6 deki küme adaylarına ait istatistiksel bilgiler	91
5.3 Önerilen yöntemin ara adımlarındaki en kötü durum zaman karmaşıklığı.	97
7.1 Filtrelenmemiş ve önerilen görsel veri madenciliği yöntemleri ile filtrelenmiş Anti-tüberküloz veri kümesinin sınıflandırma yöntemlerinin performansına göre karşılaştırılması.	110

1 GİRİŞ

Miktarı ve özellikleri sürekli artış gösteren verilerin varlığının, birçok özel uygulamada yapısal veri yapılarının kullanımını daha önemli hale getirdiği görülmektedir. Özellikle sosyal ağlar, semantik web uygulamaları, bilgisayar ağ yapıları uygulamaları, tıbbi ve kimya uygulamaları (ör: DNA ve moleküller) gibi alanlarda çizgeler veriyi modellemek için güçlü veri yapıları olarak kullanılmaktadır [Chakrabarti and Faloutsos, 2006]. Bununla birlikte son yıllarda çizgeler ile ifade edilen veri yapılarından önemli bilgilerin çıkarılması ile ilgili çok sayıda çalışma yapıldığı görülmektedir. Bu alanda yapılan çalışmaların artması çizge tabanlı veri madenciliğinin popüleritesinin giderek arttığını ortaya koymaktadır.

Bu tez çalışmasında çizge veri kümelerinden anlamlı örüntülerin çıkarılmasına olanak sağlayan bir yaklaşım önerilmektedir. Literatürde örüntü çoğunlukla çizgelerdeki tekrarlayan alt çizgeler olarak ifade edilmiştir. Örüntüleri bulmak amacıyla çizge veri kümelerinden tekrarlayan alt çizge çıkarma yöntemleri kullanılmaktadır. Bu amaçla tez çalışmasında iki yakın problem (çizge veri kümelerinden tekrarlayan alt çizgelerin çıkarılması problemi ve çok sayıda düğüm ve kenardan oluşmuş tek bir çizgeden tekrarlayan alt çizgelerin çıkarılması problemi) bahsedilecek ve literatürdeki bu problemler için önerilen yaklaşımlar incelenecektir.

İlk problem literatürde çok sayıda çizge tabanlı örneklerden oluşan veri kümesinden tekrarlayan alt çizgelerin çıkarılması problemi olarak tanımlanmaktadır. Çoğunlukla DNA da bulunan amino asit ağları ve moleküller gibi çizge veri kümeleri ile ifade edilen biyolojik ağlarda önemli örüntüleri bulma, benzeri alanlarda konu edilmektedir. Bu problem için önerilen yöntemde, bir veri kümesinde bulunan çizgeler ile ifade edilen örneklerin (ör: DNA, molekül) çoğunda görülen ortak tekrarlayan örüntülerin elde edilmesi amaçlanır. Ortak tekrarlayan örüntüler, veri kümesini oluşturan çizge örneklerinin ortak özelliği hakkında uzmanlara önemli bilgiler vermektedir. Örüntüler çizge madenciliğinde alt çizgeler ile ifade edilir. Özellikle bu tarz tıbbi

alıřmalarda, elde edilen rntlerin daha sonra ilaların tasarımında ya da DNA ile yapılan alıřmalarda kullanılacağı gz nnde bulundurulduğunda, rntlerin doėru olarak ıkarılmasının olduka nemli olduėu ortaya ıkar.

Tek bir izgeden tekrarlayan rntlerin ıkarılması olarak bilinen diėer problemin zm yaklařımında ise ok sayıda dėm ve kenardan oluřan tek bir izge ele alınır ve dėm sayısı 120.000 dėme varabilen tek bir izgeden rntler ıkarılması amalanır. Literatrdeki yntemler bu problemde rntleri alt izgeler ile gstermektedir. Bu problem genelde sosyal aė veya bilgisayar aėları uygulamalarında insan ve grupların davranıřlarının modellenmesinde kullanılmaktadır. İnsan veya nesneler izgelerin dėmleri, aralarındaki iliřkiler ise kenarlar ile ifade edecek biimde tm topluluk geniř tek bir izge ile ifade edilir. Bu tr bir rnekten tek bir bireyin davranıřı herhangi bir anlam ifade etmezken, toplumun oėunda grlen belirli davranıř biimi o topluluk hakkında arařtırmacılara bilgi verecektir. Bu noktada izgedeki tekrarlayan rntlerin toplumun davranıřlarını gsterdiėi ortaya ıkmaktadır.

İlgili gemiř arařtırmalarda izge tabanlı rnt madenciliėi zerine eřitli yntemlerin geliřtirildiėi grlmektedir Subdue, FSG, g-Span, Apriori, FFSM, Gaston, Mofa, ILP bu yntemler arasında en ok bilinenleridir [Cook and Holder, 1994; Kuramochi and Karypis, 2001 ;Yan and Han, 2002; Agrawal et al, 1993 ;Huan et al; Nijssen and Kok, 2004; Borgelt and Berthold, 2002; Stenberg and Muggleton, 2003]. nerilen bu yntemler ile bařarılı sonular elde edilse de, bazı aılardan bunların yetersiz kalabildiėi de grlmřtr.

Literatrdeki bu yntemler izge teorisindeki izge izomorfizm ve alt izge izomorfizm problemlerine yaklařımlar getirmektedirler. Bu iki problem yksek zaman karmařıklıėına sahiptir [Yang G, 2006]. zellikle ok fazla kenar ieren yoėun izgelerden oluřan geniř veri kmeleriyle alıřıldığında bu problemler daha da belirginleřmektedir. Geliřtirilen yaklařımlara raėmen, bu problemler iin gerekli zm bulunamamıřtır.

Gözlemlenen en temel eksikliklerden biri klasik yöntemlerin probleme dar bir açıdan bakmasıdır. Bu yöntemler tekrarlayan tamamen aynı örüntüleri bulmaya çalışmaktadır. Bu sebeple kullandıkları veri tabanlarındaki çizgelerin kenar ve düğümlerindeki etiket değerleri ayırık değerlerden oluşmaktadır. Ayırık değerler kavramı aşağıda bir örnekle açıklanacaktır. Tamamen ağırlıklı (kenar ve düğümleri etiketlenmiş) bir çizge ile ifade edilmiş bir moleküldeki bağların özellikleri çizgenin kenar değerlerini, atomların özellikleri ise çizgenin düğüm noktalarındaki değerleri gösterebilir. Ayırık değerlerden oluşan bir çizgede kenar değerleri bağın tipini (ör: Tekli, ikili, üçlü bağ), düğüm değerleri ise atomları (ör: Karbon, Hidrojen) göstermektedir. Bu tarz çizge tabanlı molekül örneklerinde ortak olarak geçen bir alt çizge elde edilmeye çalışılırken tüm çizgelerde ya da çizgelerin çoğunda tekrar eden aynı alt çizgenin bulunması beklenir. Şu ana kadar incelediğimiz tüm çalışmalara dayanarak, literatürdeki yöntemlerin test için kullandığı gerçek hayat veri kümelerinde ve yapay veri kümelerinde ayırık veri kullanılmaktadır, çünkü önerilen klasik yöntemler tamamen aynı tekrarlayan alt çizgeyi bulma problemini çözmemektedir. Fakat kenar ve düğümler için ayırık veri kullanmak ve bu şekilde oluşmuş veri kümelerinden tamamen aynı tekrarlayan örüntüleri çıkarma problemi bazı durumlarda yetersiz kalabilir.

Bir moleküler veri kümesindeki düğüm ve kenarları devamlı değerler içeren bir çizge için, kenar değerleri bağın belli bir özellik için ölçüm değerlerini, düğüm değerleri ise atomların bir özelliği için ölçülmüş değerlerini içeren bir çizge ile örnek verilebilir. Ölçüm değerleri ortam şartlarına göre değişebilir ve veri kümeleri gürültü içerebilir. Bu sebeple birçok gerçek hayat uygulamasında bir örüntü, çizge veri tabanlarındaki farklı çizgelerde ufak farklılıklarla tekrar edebilir. Eğer bu farklar çok az ise bu farklı ama çok benzer örüntüler de birbirinin tamamen aynı örüntüler kadar önemli olabilir. Şu ana kadar incelenen literatürdeki yöntemler ufak farklarla tekrar eden örüntüleri bulamamakta ya da eksik bulmaktadır [Yan and Han, 2002; Kuramochi and Karypis, 2001; Borgelt and Berthold, 2002; Nijssen and Kok, 2004]. Bu sebeple örüntü madenciliğinde tekrarlayan benzer örüntüleri de bulabilecek, örüntüler arasındaki ufak değişikliklere toleranslı yaklaşımların geliştirilmesi gerekmektedir.

Gözlemlenen diğer bir eksiklik ise literatürdeki yöntemlerin kapalı bir kutu gibi çalışmasıdır [Yan and Han, 2002; Kuramochi and Karypis, 2001; Borgelt and Berthold, 2002; Nijssen and Kok, 2004]. Parametreler verilip sonuç alınmaktadır. Uzmanlar işlemi yarıda kesip ara adımlarda bilgilerini çözüme katamamakta ve ara adımlarda bilgi alamamaktadır. Bu iki probleme sebep olmaktadır. İlki elde edilen sonuçların ancak işlemin sonucunda kontrol edilebilmesi, ikincisi ise işlemin herhangi aşamasındaki bir hatanın işlem bitmeden uzmanlar tarafından düzeltilememesidir. Çok sayıda parametre girdi olarak verildiğinde, parametrelerden biri için yanlış bir değer seçimi sonucun başarısız görünmesine sebep olacaktır. Bu sebeple tamamen otomatik bir yöntem önermek yerine, yarı otomatik olarak da kullanılabilecek bir yöntemin gerekliliği araştırılmalıdır.

Çok sayıda çizge içeren veri kümelerinden tekrarlayan örüntüleri bulma problemi çizge teorisinde tek sınıflı bir problemidir. Fakat birden fazla sayıda sınıfa ait örnekler içeren veri kümelerinde (ör. Bir hastalığa karşı aktif, yarı aktif ve aktif olmayan moleküller), belli bir sınıfa ait örneklerin çoğunda görülen ve diğer sınıflara ait örneklerde nadiren görülen tekrarlayan alt çizgelerin bulunması gerekmektedir. Çok sınıflı problemi bir örnekle açıklamak gerekirse A ve B iki ayrık sınıf olsun. A sınıfına ait örnekler belli bir özelliğe sahip olsun, B sınıfına ait örnekler ise o özelliği içermesin. Problemin tanımını yeniden yapmak gerekirse A sınıfına ait çizge örneklerinde bulunan ama B sınıfındaki çizge örneklerinde ise bulunmayan veya çok az geçen ortak tekrarlayan örüntülerin bulunmasıdır. Önerilen yöntemin bu gereksinime cevap verebilmesi gerekmektedir. Tek bir çizgede tekrarlayan örüntüleri bulma problemi ise tek sınıflı bir problemidir ve literatürde çok sınıflı uygulamaları yoktur.

Tezin ikinci bölümünde çizgelerde örüntü madenciliğinin iki alt probleminin tanımı yapılmış, literatürde problemin çözümü için getirilmiş yaklaşımlardan, uygulama alanlarından bahsedilmiştir. Bölüm 3 te tekrarlayan benzer altçizge madenciliği, tekrarlayan altçizge madenciliği yöntemin dayanarak anlatılmıştır. Bölüm 4 te tezde önerilen yaklaşım ayrıntılı olarak anlatılmıştır. Bölüm 5 te tezde önerilen yaklaşımın çeşitli sentetik veri kümelerinde ve bir gerçek hayat veri kümesinde elde edilen sonuçlar

ve sonuçların diğ er y ontemlerle karřılařtırılmasından bahsedilmiř ve sonuçlar yorumlanmıřtır. B olum 6 da tezde vardığımız sonuçlar ve gelecek  alıřmalar  zetlenmiřtir. B olum 7 de gelecek  alıřmalara ıřık tutması a ısından yapılan  n  alıřmalardan bahsedilmiřtir. B olum 7.1 de ise elde edilen  r nt lerin yeni  izgelerin sınıflandırılmasında kullanımı ile ilgili yaptığımız bir  n  alıřma ve sonuçlarından bahsedilmiřtir. Son olarak b olum 7.2 de  r nt  haritalarındaki eřik değeri otomatik olarak bulunması i in alternatif y ontemler anlatılmıřtır.

2 LİTERATÜR ANALİZİ

Bu bölümde literatürdeki çizge tabanlı örüntü madenciliği alanındaki bir kaç problemten bahsedilip, konu ile ilgili iki problem tanımlanacaktır. Problemlerin çözümü için önerilmiş yaklaşımlardan ve uygulama alanlarından bahsedilecektir.

2.1 Çizge Tabanlı Örüntü Madenciliği

Özellikle son yıllarda çizge tabanlı örüntü madenciliğine ilgi gittikçe artmaktadır. Bunun sebeplerinden bir tanesi de özellikle sosyal ağların (ör. facebook), internetin kullanımının artması ve tıbbi çalışmaların bilgisayar ortamına aktarılması ile birlikte yapısal veri yapılarına ihtiyacın artması ve araştırmacıların bu gibi veri kümelerinden önemli bilgiler elde etme istediğidir [Jamali and Abolhassan, 2006]. Çizgeler, veri madenciliğinin birçok alanında güçlü veri yapıları olarak kullanılmaktadırlar. Çizge tabanlı örüntü madenciliği de veri madenciliğinin bir alt dalıdır. Çizge tabanlı örüntü madenciliğindeki birçok uygulama alanında araştırmacılar aşağıdaki problemler gibi bir çok probleme yoğunlaşarak, yaklaşımlar getirmektedirler. Problemler benzer gibi görünse de birbirinden farklı çözümleri gerektirmektedir. Önerdiğimiz yöntem bunlardan sadece ilk iki probleme yakındır.

- Çok sayıda küçük çizge kümelerinden sıklıkla tekrarlayan alt çizgelerin bulunması
- Tek bir büyük çizgeden sıklıkla tekrarlayan alt çizgelerin bulunması
- Çizgedeki belirli iki düğüm arasında anlamlı yolların (ör. en çok akışın olduğu) çıkarılması [Tong and Faloutsos, 2006]
- Çizgedeki düğümlerin çoğu ile bağlantılı alt-çizgeyi bulmak [Ramakrishnan et al, 2005]
- Bir topluluğun ya da nesnelerin farklı zaman aralıklarındaki etkileşimlerinden elde edilen ve arka arkaya gelen çizgelerden oluşan veri kümesinden (“temporal

network”), bu topluluğun gelecekte hangi zaman aralıklarında nasıl davranış göstereceğini bulmak [Lahiri and Berger, 2007].

Bölüm 2.1 de *çizge veri kümelerinde tekrarlayan alt çizgelerin çıkarılması problemi* ve bölüm 2.2 de *tek bir çizgede tekrarlayan alt çizgelerin çıkarılması problemi* ayrıntılı olarak tanımlanacak ve farkları üzerinde durulacaktır. Çizgeler ile ilgili tanımlar ise bölüm 3 te yapılacaktır.

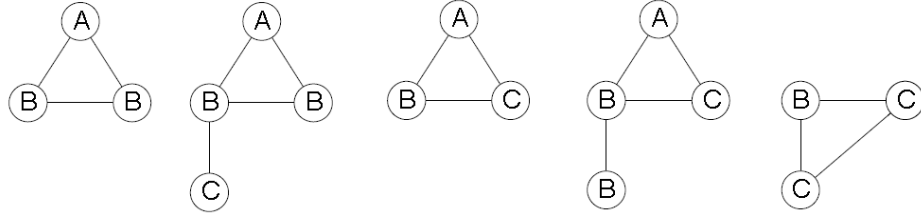
2.2 Çizge Veri Kümesinden Tekrarlayan Alt çizgelerin Çıkarılması

2.2.1 Problemin Tanımı

İlk problem *çizge veri kümesinden tekrarlayan alt çizgelerin çıkarılması problemi* (ÇK-TAÇ) olarak bilinmektedir [Kuramochi and Karypis, 2005]. ÇK-TAÇ problemi çizge tabanlı örneklerden oluşan veri kümelerinde sıklıkla tekrarlayan alt çizgelerin çıkarılmasına dayanır. Veri kümesindeki çizgelerde sıklıkla tekrarlayan alt çizgeler *örüntü* olarak alınır ve veri kümesinin karakteristik özellikleri hakkında bilgi verirler. Bu problemde, bölüm 2.3 de bahsedilecek *tek bir çizgede tekrarlayan alt çizgelerin çıkarılması probleminden* farklı olarak az sayıda düğümlü seyrek çizgelerden oluşan çok sayıda çizge içeren veri kümeleri kullanılır (Bkz. şekil 2.1). Veri kümesindeki her çizge aynı ya da farklı sayıda düğüm ve kenar içerebilir. Düğüm sayısı uygulamadan uygulamaya değişirken genelde 10 ile 50 arasında olduğu gözlemlenmektedir.

Çizge izomorfizm ve alt çizge izomorfizm probleminin karmaşıklığından dolayı düğüm sayısındaki artış işlem süresini çok fazla arttıracığından bu uygulamalar az sayıda düğümlü seyrek çizgeler için uygundur [Kuramochi and Karypis, 2005]. ÇK-TAÇ problemini, *tek bir çizgede tekrarlayan alt çizgelerin çıkarılması probleminden* ayıran bir diğer özellik ise örüntünün sıklığının nasıl hesaplandığıdır. ÇK-TAÇ probleminde

literatürde *destek değeri* (“support value”) olarak tanımlanan örüntünün sıklığı, diğer bir deyişle örüntünün çizge veri kümesindeki çizgelerde geçme oranıdır. Çizelge 2.1 de şekil 2.1 deki veri kümesinden elde edilen alt çizgelerin destek değerleri görülmektedir. Bir örüntünün tek bir çizgede kaç defa geçtiği dikkate alınmaz.



Şekil 2.1, Örnek bir çizge veri kümesi

Çizelge 2.1, Sıklıkla tekrarlayan alt çizgelerin bazıları ve onların destek yüzdesi değeri.

Örüntü adayı Alt çizgeler	Yapısı	Destek değeri
CS ¹		4/5 {G ¹ , G ² , G ³ , G ⁴ }
CS ²		3/5 {G ¹ , G ² , G ⁴ }
CS ³		2/5 {G ¹ , G ² }
CS ⁴		2/5 {G ² , G ⁴ }
CS ⁵		1/5 {G ² }

Bölüm 3.1 de problemin tek sınıflı ve çok sınıflı çizge veri kümeleri için tanımları ayrıntılı olarak yapılacaktır. Çok sınıflı veri kümelerinde birden fazla sınıfa ait çizge grupları bulunmaktadır. Her sınıf kendi içinde benzer özellikler ve benzer örüntüler içerirler. Bu örüntüler o sınıfın karakteristik özellikleridirler. Farklı sınıflardaki çizgelerin özellikleri birbirinden farklıdır. Fakat farklı sınıflara ait çizgelerde ortak tekrarlayan alt çizgeler olabilir. Bu alt çizgeler sınıflara ait karakteristik özellikleri göstermezler. Örüntüleri bulurken bu alt çizgeleri örüntülerden ayırmak gerekmektedir.

2.2.2 Yöntemler

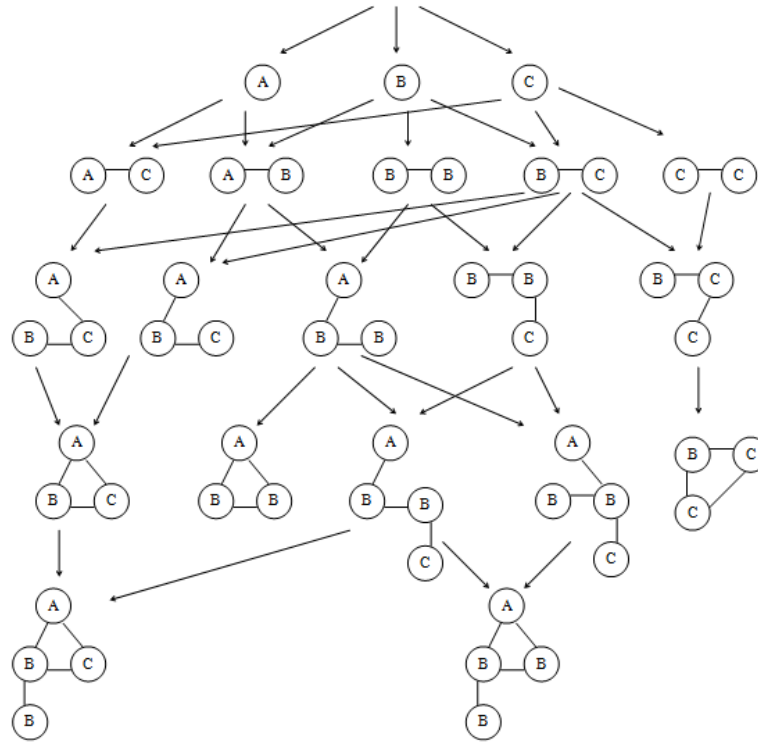
Çizge tabanlı veri madenciliği yöntemleri, çizge tabanlı çok sayıda çizge içeren veri kümelerinden, önemli örüntüleri çıkarmayı amaçlarlar. Bu yöntemleri birçok kategoriye ayırmak mümkündür. Çalışmalarımıza ve veri madenciliği yöntemlerine en yakın olanlar matematiksel çizge tabanlı yöntemler, greedy arama tabanlı yaklaşımlar ve tümevarımsal mantık programlama yöntemleridir [Washio and Motoda, 2003]. Bu bölümde bu alandaki önerilmiş yaklaşımlardan, avantajları ve yetersizliklerinden bahsedilecektir.

Literatürde çizgelerde veriyi tutmak için bilinen iki yöntem, yakınlık matrisleri (adjacency matrix) ve yakınlık listeleri (adjacency lists) yaygınca kullanılmaktadır. Fakat bu gösterimler çizge madenciliği için uygun olmayabilir. Bir örnek vermek gerekirse bir çizgenin yakınlık matrisi ve çizgenin düğümlerinin farklı permütasyonlarından oluşan yakınlık matrisleri aynı çizgeye aittir. Fakat iki yakınlık matrisi aynı elemanları farklı sırada ihtiva ederler. Bu sebeple, yakınlık matrisleri çizge izomorfizm probleminin çözümünde arama uzayını arttırdığı için uygun bir gösterim değildir [Washio and Motoda, 2003]. Bu yönteme alternatif olarak makine öğrenmesi, veri madenciliği ve istatistiksel yöntemlerde sıkça kullanılan kanonik tanımlama (canonical label) gösterim kullanılmaktadır. Değişik şekillerde kanonik tanımlamaların oluşturulması mümkündür. g-Span yöntemindeki DFS dizilimi kanonik tanımlamaya bir örnektir ve g-Span yöntemi anlatılırken bahsedilecektir [Yan and Han, 2002].

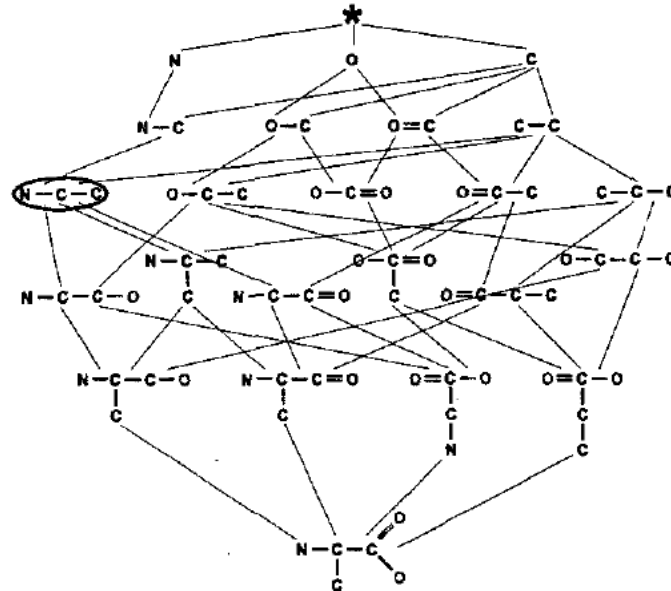
ÇK-TAÇ işlemi literatürdeki iki zor problemin çözülmesini gerektirmektedir. Bunlar çizge izomorfizm ve alt çizge izomorfizm problemleridir. Çizge izomorfizm iki çizgenin izomorfik olup olmadığını inceler. Şu ana kadar zaman karmaşıklığının P ya da NP tam olup olmadığı kanıtlanamamıştır. Diğer bir problem olan alt çizge izomorfizm ise bir çizgenin diğer çizgenin alt çizgesi olup olmadığını inceler [Yang, 2006]. Bu problem ise NP tam zaman karmaşıklığına sahiptir [Yang, 2006]. Literatürdeki yöntemler bu yüksek zaman karmaşıklığını çözmek ve arama uzayını azaltmak için bazı limitler kullanırlar.

Literatürdeki alt çizge arama yöntemleri genel olarak veri kümesindeki tüm alt çizgelerden oluşmuş bir kafes yapısında arama yapmaktadır [Fischer and Meinl, 2004]. Şekil 2.2 de, şekil 2.1 deki örnek veri kümesinden elde edilen kafes yapısı görülmektedir. Kafes yapısındaki her seviyedeki çizge, üst seviyelerdeki çizgelerin birleşiminden oluşmaktadır. Örnek vermek gerekirse, kafes yapısında ilk seviyede düğümler, ikinci seviyede bu düğümlerin olası birleşimlerinden oluşan çizgeler (iki kenar ve bir düğümden oluşan çizgeler) görülmektedir. Üçüncü seviyede ise ikinci ve birinci seviyedeki çizgelerin olası birleşimlerinden oluşan çizgeler sıralanmaktadır ve kafes yapısı bu şekilde oluşturulmaktadır. Oklar ise çizgeyi, önceki seviyedeki çizgelerin hangilerinin birleştirildiğini göstermektedir. Şekil 2.3 de tek bir molekülden oluşturulan örnek bir kafes yapısı görülmektedir [Fischer and Meinl, 2004]. Kafes yapısının oluşturulmasında iki adım kullanılmaktadır:

- **Aday üretme:** Aday üretme ilk adımı küçük çizgeler birleştirilerek yeni çizgelerin oluşturulmasına dayanır. Literatürdeki çeşitli yaklaşımlar bu adımda iki farklı yöntem kullanmaktadır.
 - o İlk yöntemde ortak alt çizgelere sahip küçük çizgeleri birleştirerek ağaç yapısı oluşturulur. Bu yöntemin kullanım alanı olarak Apriori tabanlı yaklaşımlar örnek verilebilir [Agrawal et al, 1993]. Bu yöntemde hangi çizgelerin birleştirileceğini belirlemek için ortak alt çizgelere sahip çizgelerin belirlenmesi gerekmektedir.



Şekil 2.2, Şekil 1 deki örnek veri kümesinin alt çizgilerinin kafes yapısı ile gösterilmesi.



Şekil 2.3, Cylin bileşiğinin alt bileşikleri [Fischer and Meinl, 2004].

- o İkinci yöntemde ise kafes yapısının her seviyesindeki çizgeler, bir alt seviyedeki çizgelere birer kenar eklenerek oluşturulur. Bu yönteme Mofa, g-span yaklaşımları örnek verilebilir [Borgelt and Berthold, 2002; Yan and Han, 2002]. Bu yöntem ise kafes yapısının çok fazla genişlemesine diğer bir deyişle gereksiz çizgelerin oluşmasına sebep olmaktadır. Burada kastedilen gereksiz çizgeler veri kümesindeki çizgelerde bulunmayan çizgelerdir. Daha sonra bu çizgelerin destek değerlerinin hesaplanması gerekeceğinden işlem süresinin artmasına sebep olacaktır.
- **Destek değerinin belirlenmesi:** Destek değeri elde edilen yeni çizgenin veri kümesinde geçme oranıdır (Veri kümesindeki çizgelerin kaçında geçtiğini gösterir.). Bir alt çizgenin tüm çizgelerde olup olmadığının belirlenmesi NP tam bir problemdir. Bu probleme literatürde getirilen en bilinen yaklaşımlar aşağıda sıralanmıştır:
 - o Bu problem için önerilmiş ilk yaklaşımda iki ana alt çizge birleştirilirken elde edilen çocuk çizgenin destek değerini hesaplamak için sadece çocuk çizgeye ait ana alt çizgelerin veri kümesinde geçtiği çizgelerde destek değerine bakılır. Bu sayede arama uzayı daraltılmış olur. Bu yaklaşımın dezavantajı alt çizgenin geçtiği tüm çizgelerin hafızada tutulmasının gerektirmesidir. G-Span yöntemi bu yöntemi kullanmaktadır.
 - o İkinci yaklaşımda gömülü listeler (embedding lists) kullanılmaktadır. Gömülü listeler, alt çizgenin kenar ve düğümleri ile çizgenin (onu içeren çizge) içindeki ona karşılık gelen kenar ve düğümler arasında bir eşleme yapar. Bu sayede alt çizgenin veri kümesindeki kaç çizgede geçtiğini hesaplamak için, sadece yeni eklenen kenar ve düğümleri aramak yeterli olacaktır. Bu yaklaşımda da ilk yaklaşımda ki gibi büyük bir hafıza gereksinimi vardır. MOFA, FFSM, GASTON yöntemleri bu yaklaşımı kullanmaktadırlar.
 - o Kafesi daraltmaya yönelik olan üçüncü yaklaşım olan Apriori yönteminde (antimonocity constraint method) ise iki ana alt çizgenin destek değeri, eşik değerinden düşük ise onların birleşiminden oluşan

çizge kafes yapısına eklenmez. Bu sayede kafes budanmış olur. Bu yaklaşım ilk kez Apriori yönteminde kullanılmıştır [Agrawal et al, 1993].

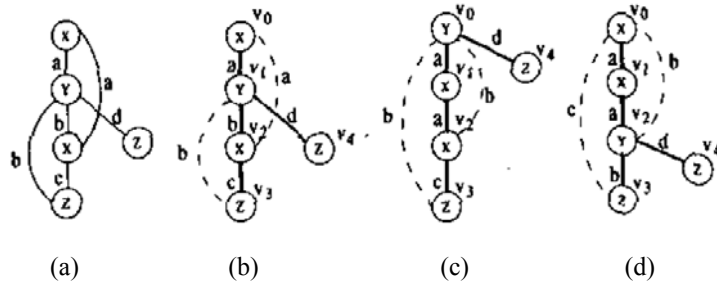
Bu iki adımı sonlu zamanda ve daha az kaynakla çözmek için çeşitli yöntemler geliştirilmiştir. Bu bölümün devamında yukarıda örnek verilen ve literatürde sıklıkla geçen yöntemlerden bahsedilecektir.

Apriori tabanlı benzer ortak parçaları çıkarma: Apriori tabanlı benzer ortak parçaları çıkarma yöntemi sepet analizindeki Apriori algoritmasına dayanmaktadır [Agrawal et al, 1993]. Bu yöntemde veri kümesinden elde edilen aday alt parçalar şekil 2.2 dekine benzer bir ağ yapısı oluşturur. Eğer aday alt çizge belirli bir destek ve güven parametrelerini sağlamıyorsa, aday alt çizge listesinden çıkarılır. Bu sayede arama uzayı budanmış olur. Son olarak bu aday alt çizgeler tüm çizge örneklerinde aranır. Bu yöntemin dezavantajı enine arama yapması ve bu sebeple çok fazla hafıza gerektirmesidir.

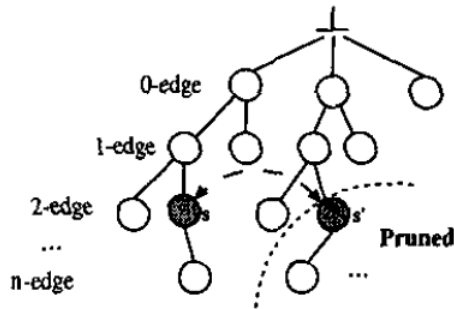
g-Span: g-span yöntemi tekrarlayan alt çizgeleri bulmak için DFS sözlüksel sıra yöntemini kullanır [Yan and Han, 2002]. Şekil 2.4 (b-d), şekil 2.4 (a) deki çizgenin farklı görünüşlerdeki izomorfik DFS ağaç yapısını göstermektedir. Çizelge 2.2 de ise farklı kenarları kök yani başlangıç kenar olarak seçerek oluşturulmuş DFS dizilimi görülmektedir. İki çizge ancak onların en küçük DFS dizilimleri aynı ise izomorfiktir. Dolayısıyla g-Span yöntemi tekrarlayan alt çizgeleri bulma problemini DFS kodunda birbiriyle ilişkili kısımları çıkarma problemine dönüştürür. Bunun için ardışık örüntü bulma yöntemlerini (DFS Kod ağacı) kullanır. DFS Kod ağacında her düğüm DFS kodlarını içerir. $\alpha = (a_0, a_1, \dots, a_m)$ üst düğüm $\beta = (a_0, a_1, \dots, a_m, b)$ ise onun bir alt düğümü olacak şekilde DFS kodları bir ağaç yapısı oluştururlar. Ağaç yapısındaki n-inci seviyede (n-1) düğümlü çizgeler bulunmaktadır. Derinlemesine arama (Depth first search) ile tüm en küçük DFS kodları görülmektedir. Şekil 2.5 de örnek bir DFS ağaç yapısı görülmektedir. Burada s ve s' düğümleri aynı çizgenin farklı DFS kodlarını göstermektedir. s' en küçük DFS kodu olmadığı için ağaç yapısından silinir.

Çizelge 2.2, Şekil b-d deki DFS çizge kodları [Yan and Han, 2002].

Kenar	Şekil (b)	Şekil (c)	Şekil (d)
0	$(0, 1, X, a, Y)$	$(0, 1, Y, a, X)$	$(0, 1, X, a, X)$
1	$(1, 2, Y, b, X)$	$(1, 2, X, a, X)$	$(1, 2, X, a, Y)$
2	$(2, 0, X, a, X)$	$(2, 0, X, b, Y)$	$(2, 0, Y, b, X)$
3	$(2, 3, X, c, Z)$	$(2, 3, X, c, Z)$	$(2, 3, Y, b, Z)$
4	$(3, 1, Z, b, Y)$	$(3, 0, Z, b, Y)$	$(3, 0, Z, c, X)$
5	$(1, 4, Y, d, Z)$	$(0, 4, Y, d, Z)$	$(2, 4, Y, d, Z)$



Şekil 2.4, (a) çizgesi ve çizgenin birbiri ile izomorfik üç DFS ağaç örneği (b-d).



Şekil 2.5, Örnek bir DFS ağaç yapısı [Yan and Han, 2002].

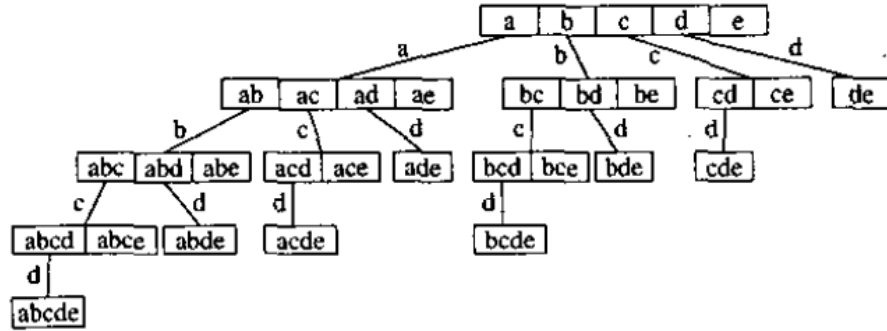
FSG: FSG yaklaşımı da kanonik çizge gösterimlerini kullanır [Kuramochi and Karypis, 2001]. Her alt çizge için yakınlık matrisinin kenar ve düğümlerinin tüm kombinasyonları oluşturulup tek bir kanonik tanım oluşturulmaya çalışılır. Bu yaklaşım da şekil 2.2 dekine benzer bir ağ yapısı oluşturur. Fakat alt çizgeler yerine kanonik gösterimler bulunmaktadır. Arama uzayını budamak için çizgeyi alt parçalara bölüp alt parçalarda düğümlerin permütasyonlarını hesaplayan düğüm invariant tekniğini kullanır. Bu yöntemde Apriori gibi aday alt çizge üretme ve destek değerine göre arama uzayını budama yöntemini uygular. FSG yöntemi tek sınıflı uygulamalar için geliştirilmiştir.

ILP: Tümevarımsal mantık programlama yöntemlerinde her sınıfa ait gözlemler, geçmiş bilgi ve sonuçlar birinci dereceden mantık ile ifade edilir [Stenberg and Muggleton, 2003]. Veriden mantık tabanlı kurallar çıkararak, bir sınıfa ait veri kümesinde çok fazla bulunan ama diğer sınıflarda az miktarda bulunacak özellikler çıkarır. Bu yöntem birden fazla sınıfa ait veri kümeleri için uygundur. Bu yöntemin dezavantajı ise nümerik işlemleri desteklememesidir.

Mofa Yöntemi: Temel olarak sepet analizinde ilişki kuralları çıkarma ve sık elemanları bulma tekniklerine dayanmaktadır. Mofa yaklaşımı moleküler bağlı örüntüleri çıkarmak için elde edilmiştir fakat farklı uygulamalara da uygulanabilmektedir [Borgelt and Berthold, 2002]. Yapılan uygulamada çizgelerdeki düğümler atomların yük değerlerini ya da aromatik bir halkanın parçası olup olmadığını gösterirken çizgenin kenar değerleri ise bağların tiplerini (tekli, çiftli, üçlü ya da aromatik) göstermektedir. Mofa da sık alt çizgeler bir ağaç yapısında tutulur. Ağaç yapısındaki her düğümde bir sayaç ile o düğümdeki alt çizgenin veri kümesindeki çizgelerin kaçında olduğu bilgisi (destek değeri) tutulmaktadır. Ağaç arama için derinlemesine arama (*Depth First Search*) yöntemi kullanılmıştır. (Market sepet analizini kullanan diğer bir yaklaşım olan Apriori algoritmasında ise çok fazla hafıza gerektiren diğer bir arama yöntemi enine arama (*Breadth First Search*) kullanılmıştır.)

Arama uzayını daraltmak için iki tür daraltma yöntemi kullanılmıştır. *Destek Değerine Dayalı Daraltma* yönteminde destek değeri, destek eşik değerinden az olan alt çizgeler ağaç yapısından elenir. Örüntüde olması istenen en fazla düğüm sayısı kullanıcı tarafından belirlenmiştir (*Boyut Tabanlı Daraltma*). İkinci yöntem ise *Yapısal Daraltmadır*. Bu yöntemi bir örnekle açıklayalım. Şekil 2.6 de bir ağaç yapısı örneği görülmektedir. (Basit olması açısından çizgeler yerine {a,b,c,d,e} elemanlarından oluşan bir ağaç yapısı ile gösterelim.) Görüldüğü gibi ağaç yapısı aynı elemanları içermemektedir. Örnek vermek gerekirse ağacın bir düğümünde {a,c} kümesini bulunduruyor ise {c,a} kümesinin bulunmaması gerekir (*Yapısal Daraltma*). Bunun için önce {a,b,c,d,e} kümesindeki elemanları sıralanmıştır. Daha sonra ağaçtaki her seviyeyi oluştururken üst seviyedeki kümenin sağ tarafına o elemandan sonra gelen elemanlar

eklenerek oluşmuş alt düğümler oluşturulur. Aşağıdaki örnekte eğer $\{a, c\}$ düğümüne bu kümenin elemanlarından sıralamada sonra gelen d ve e elemanlarını sağ tarafa eklenir. Bu sayede $\{a, c\}$ kümesinin alt düğümleri $\{a, c, d\}$ ve $\{a, c, e\}$ olacaktır. Şekil 2.6 da bu şekilde oluşturulmuş bir ağaç yapısı görülmektedir.



Şekil 2.6, $\{a, b, c, d, e\}$ kümesinin elemanları ve Yapısal Daraltma Yöntemi ile oluşturulmuş bir ağaç yapısı örneği [Borgelt and Berthold, 2002].

Gaston: Gaston yöntemi da Apriori özelliğine (antimonocity constraint method) dayanır [Nijssen and Kok, 2004].

Tanım 2.1 Apriori özelliği: D bir veri kümesi, $G_1, G_2 \in D$ olmak üzere iki çizge olsun. Freq fonksiyonu eşitlik 2.11 deki gibi tanımlansın.

$$\text{freq}(G, D) = \#\{G' \in D \mid G \subseteq G'\} \quad (2.11)$$

Bu durumda $G_1 \subseteq G_2$ ise $\text{freq}(G_1, D) \geq \text{freq}(G_2, D)$ sağlanır. Bu özelliğe dayanarak küçük (ana) çizgeler tekrarlayan çizgeler değilse onları içeren daha büyük çizgeler de tekrarlayan alt çizgeler olmayacaktır. Tek bir çizgeden başlayarak derinlemesine oluşturma (depth-first) algoritması ile bağlı çizgeler (alt çizgeler) oluşturulur. Gaston, izomorfik çizgelerin oluşmasını önlemek için çizge kodunu kullanır. Çizge kodu çizgeye eklenen düğüm ve kenarların sırasını gösteren bir karakter dizisidir.

Subdue: Problemin zaman karmaşıklığını çözmek için greedy arama yöntemini kullanan Subdue yöntemi en bilinen yöntemlerdendir [Cook and Holder, 1994]. Subdue yöntemi tekrar eden parçaları sıkıştırarak, benzer parçaları bulma yoluna gider. Verinin

hiyerarşik yapısını oluşturur. Önce tek bir düğümden başlayarak, her iterasyonda greedy mantığı ile o an için en uygun başka bir kenar ekleyerek alt parçayı büyütür. Greedy yönteminin mantığında olan genel ideal sonuçlar bulunamaması durumu yüzünden, tüm alt parçacıkları bulamayabilmektedir.

2.2.3 Uygulama Alanları

Literatürdeki bu problem için geliştirilmiş yaklaşımlar çoğunlukla protein, DNA ve farmakoloji gibi alanlardaki problemleri çözmek için kullanılmaktadır. Gen çalışmalarında, proteinlerin birbirleri ile etkileşimleri incelenir. Bu amaçla veri madenciliği uzmanları protein veri kümelerinde birincil, ikincil ve üçüncül protein yapılarından yapısal örüntülerin çıkarılması ile ilgili çalışmaktadırlar [Han et al, 2007]. Farmakoloji alanında ise moleküller; düğümler atomları, kenarlar ise atomlar arasındaki bağları ifade edecek şekilde üç boyutlu çizgeler ile ifade edilirler. İlaç dizaynında, uzmanlar *Yapı Aktivite İlişkileri (SAR)* olarak bilinen yöntemle dayanarak bileşiklerin yapıları ve aktiviteleri (ör. Koku ya da bir hastalığa olan aktivitesi) arasındaki ilişki ile ilgilenmektedir. Bu yöntemlerden bazıları *Nitelik Aktivite İlişkileri (QSAR)* olarak bilinmekte ve molekül veri tabanlarından belli bir özellikteki ortak önemli örüntüleri bulmaya çalışmaktadır. Örüntüler atom ve bağların birleşmesi ile oluşan molekül parçalarıdır. SAR uygulamalarında aktif bir örüntü belli bir özelliğe (ör: hastalık) sebep olan molekül parçalarına denir. Bu örüntüler daha sonra uzmanlar tarafında yeni ilaçların dizaynında kullanılmaktadır. Yapı aktivite ilişkileri çalışmaları olarak adlandırılan yöntemlerin bir kısmı temelde çizge tabanlı örneklerden tekrarlayan alt çizgeleri bulmaya çalışırlar [Fischer and Meinl, 2004]. Yöntemlerin başarısı, yöntemin uygunluğuna bağlı olduğu kadar seçilen veri kümesinde kullanılan atom ve bağların özelliklerine de bağlıdır [Feri et al., 2007]. Yapı aktivite ilişkileri ile ilgili yapılan çalışmalar bölüm 2.2.3.1 de kısaca bahsedilecektir. Bu problemin tek sınıflı [Kuramochi and Karypis, 2001] ve birden fazla sınıflı uygulamaları [Cook and Holder, 1994] literatürde mevcuttur (ör. Aktif ve aktif olmayan moleküller).

Bu problemin moleküler veri kümelerinde uygulanması bazı zorlukları da beraberinde getirmektedir. Bazı bileşikler molekül yapılarında ufak değişiklikler olmasına karşı kimyasal reaksiyonlarda aynı davranışı sergilerler. Örnek vermek gerekirse karbon halkasından oluşan aktif bir parça bazı bileşiklerde nitrojenle yer değiştirip aynı aktiviteyi sergileyebilir. Gerçek hayat veri kümelerinde bu tarz durumlardan dolayı örüntüler bulunamayabilir.

ÇKTAÇ, örüntü madenciliğinde tek çizge problemine göre daha çok çalışılmış ve daha ilgi çekmiştir. Bunun en önemli sebebi uygulama alanının sadece biyolojik veride kısıtlı kalmasına rağmen, elde edilen örüntülerin ilaç dizaynında veya DNA'nın keşfedilmesinde kullanılması ve kimyacılar için önemli bilgiler vermesidir [Han et al., 2007]. Bu sebeple elde edilen örüntülerin doğruluğunun hayati derecede önemli olması araştırmacıların ilgisini çekmektedir.

2.2.3.1 Yapı Aktivite İlişkileri (SAR)

Yapı-biyolojik aktivite problemi, ilaç dizaynı, yiyeceklerdeki katkı maddeleri, parfümler gibi günlük hayat uygulamalarında görülmektedir. Kimya, ilaç sanayi, biyoloji vb. gibi bilimlerde en genel problemleri çözmek ve deneysel veriden en fazla miktarda bilgiyi elde edebilmek için biyologların, kimyacıların, farmakologların, istatistikçilerin ve yazılım mühendislerinin ortak çalışması gerekmektedir. Özellikle kimya, ilaç sanayi, biyoloji gibi bilimlerdeki deneysel verilerin çok boyutlu ve karışık olması problemin çözümünü zorlaştırmaktadır.

İnsanların günlük hayatta kullanabilecekleri yeni maddelerin sentezi, modern fizik, kimya ve malzeme bilimlerinde hala en önemli problemlerinden bir tanesidir [Leach, 2001; Veselovsky and Ivanov, 2003]. Günümüzde ilacın sentezlenmesinden kullanımına kadar gelişimi için harcamalar gün geçtikçe artmaktadır [Good et al, 2000]. Çizelge 2.3'de senelere göre ilaç hazırlanması için yapılan harcamalar görülmektedir.

Çizelge 2.3, ABD de ilacın hazırlanması için gereken harcamaların yıllara göre dağılımı milyon \$ cinsinden görülmektedir.

Yıl	1962	1970	1979	1986	1996	2004
Harcama	4	25	50	100	350	600

Son yıllarda yapılan bu çalışmalar, “yapı aktivite ilişkisi” (structure-activity relationship, SAR) ve “nicel yapı-aktivite ilişkisi” (quantitative SAR, QSAR) başlığı altında yayınlanmıştır. Günümüzde, moleküler modelleme tekniklerini kullanarak, moleküllerin üç boyutlu yapılarını ve bunlara eşlik eden spesifik fizikokimyasal niteliklerini tanımlamak artık olası duruma gelmiştir. Bu teknikler, moleküler grafik çizimlerinden yararlanılarak, üç boyutlu moleküler özelliklerinin tanımlanmasını sağlamaktadır.

Geliştirilen bu yöntemlerin neredeyse tamamı, yüksek teknoloji gerektirmesi ve araştırma-geliştirme maliyetlerinin yüksek olması nedeniyle gelişmiş ülkeler tarafından geliştirilmektedir. Bu problemi çözmek için daha az maliyetli ve verimli yöntemler geliştirilmesi gereklidir. Son zamanlarda sadece benzer yapıda bileşikler inceleyen veya elektronik yapının rolünü ihmal eden QSAR yöntemlerinin eksikliklerini gidermek amacı ile Elektron-Topolojik QSAR (ET-QSAR) yöntemi geliştirilmiştir [Dimoglo, 1985; Shvets, 1993; Bersuker and Dimoglo, 1991; Shvets and Dimoglo, 1999]. ET-QSAR yönteminde; farklı yapıdaki bileşikler inceleyebilmekte, aynı zamanda hesaplamalar da bileşiklerin elektronik özellikleri de dikkate alınmaktadır. ETM yöntemi moleküllere ait elektronik ve topolojik bilgiler gibi aktivitenin belirlenmesi için gerekli ve önemli özellikleri içerdiğinden güçlü bir yöntemdir. ETM deneysel aktiviteleri belli olan bileşik serilerine uygulanabilmektedir ve biyolojik aktivitenin açıklanmasında oldukça etkili bir yöntemdir. Bu yöntemin prognoz sonuçları %90-98 arasındadır.

İlaç dizaynındaki en önemli problemlerden biri “Yapısal Aktivite” ilişkileri problemi (SAR) olarak bilinen bileşiklerin aktivitesinin tahmin edilmesi ve aktiviteyi belirleyen özelliklerin çıkarılması problemidir. Bu problem “yapısal-özellik”

probleminin bir alt parçasıdır. SAR problemini çözmek için iki türlü yaklaşım kullanılmaktadır. Bunlar:

1) **Receptor tabanlı ilaç dizaynında** (direct method) uygun bir bio-reseptörün yapısı biliniyorsa (ligant-reseptör) etkileşimi incelenir.

2) **Doğrudan olmayan yaklaşım** ise iki gruptan oluşur [Fujita et al, 1984].

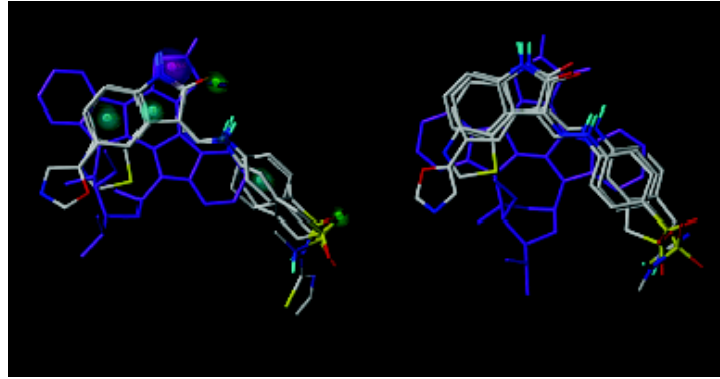
- **Nicel Yaklaşımlar (Quantitative methods):** Molekülün aktivitesi ile yapısı arasında bir denklem olarak ifade edilir. Moleküler özellikler, katsayılar ve sabitler olacak şekilde; $A = F(Y)$ eşitliği sağlanır. (Y=Yapı, A=Aktivite)
- **Nitel (Yapısal) Yaklaşımlar (Qualitative or structural methods):** SAR probleminde kullanılan Nitel yaklaşımlar, aktif bileşiklerin moleküler yapısındaki belli bir alt yapının aktiviteye sebep olduğu hipotezine (**anahtar-kilit hipotezi**) dayanır.

2.2.3.2 QSAR uygulamaları

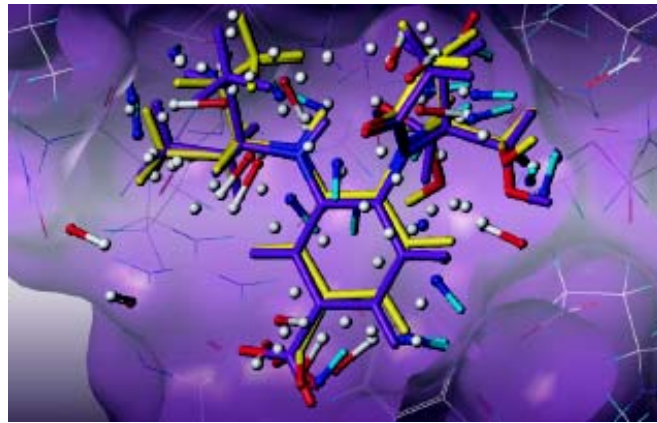
1) ComFA – Karşılaştırmalı Alan Analizi (Comparative Molecular Field Analysis):

3 boyutlu moleküler yüzeylerin topolojik sıralanışlarındaki çeşitli moleküllerin çıkışan ortak noktalarının bulunması prensibine dayanır (Bkz. şekil 2.7). ComFA moleküllerin yapıları ve özellikleri ile alakalı istatistiksel ve grafiksel bir model inşa eder [Kubinyi, 1997]. Bu modeller daha sonra yeni bileşiklerin aktivitesinin ve özelliklerinin tahmin edilmesinde kullanılır. Çok çeşitli yapısal tanımlayıcılar (structural descriptors) ele alınır. Bir molekülü tanımlamak için bir dizi tanımlayıcı kullanılır. Yapısal tanımlayıcılar olarak molekülün parçaları kullanılır. (Bu tanımlamanın seviyesi atom seviyesinde değildir.)

2) **Biyomolekülün içine ilacın indirgenmesi (Molecular Docking):** Bir aktif molekülün bir reseptör ile reaksiyonu sırasında zaman içindeki tüm değişiklikleri modellenir (Bkz. şekil 2.8).



Şekil 2.7, Karşılaştırmalı Alan Analizi örneği.



Şekil 2.8, Biyomolekülün içine ilacın indirgenme örneği.

Bu uygulamalar da moleküllerin etkileşimini ve yakınlığını göstermek için büyük ölçüde görüntülemeyi kullansa da, aynı anda sadece bir ya da çok az molekül üzerinde çalışabilir. Aynı anda çok fazla molekül üzerinde işlem yapılması imkânsızdır. Hesaplamaları zaman harcayıcıdır, haftalar sürebilir. Limitli uygulamaları vardır. Bulunan bir ilaç için test aşamasında kullanılabilir.

2.2.3.3 Genel Yapısal Yaklaşım Uygulamaları

QSAR uygulamalarının çoğunluğu, desen tanıma teorisini (Pattern Recognition Theories, PRT) kullanır. PRT kullanılarak, bazı sınıflandırma kurallarıyla inceleme altındaki moleküllerin vasıtasıyla aktif ve aktif olmayan bileşikler sınıflandırılır. PRT yaklaşımlarında en önemli olan, molekülleri tanımlama yöntemidir. Molekülün

aktifliğini gösteren özellikler bakımından molekülün tanımı ne kadar iyi yapılırsa, yapıyı tanıyarak molekülün aktif ve aktif olmayan olarak sınıflandırılması o kadar başarılı olur. Bu yüzden QSAR yöntemleri, molekülün tarif edilmesinde farklı yaklaşımlar kullanır.

Yapısal yaklaşım yöntemlerinde öncelikle uygun bir tanımlama dili ve karar kuralları sınıfı seçilir. Buna göre aktif bileşiklerde ortak olarak bulunan aktif özelliklere **farmakoforlar** denir. Aynı işlem inaktif moleküller için de tekrarlanır. Sadece inaktif moleküllerde bulunan ortak parçalara **anti-farmakoforlar** denir. Aktif ve aktif olmayan moleküllerdeki ortak parçalar farmakofor ya da anti-farmakofor olarak alınmaz. Çünkü aktif bileşiklerde bulunan farmakofor eğer aktif olmayan bileşiklerde de ortak olarak bulunuyorsa bu özellikler aktiviteyi veren özellikler değildirler. Bulunan farmakoforlar yeni bileşiklerin aktif ya da aktif olmayan olarak sınıflandırılmasında kullanılır.

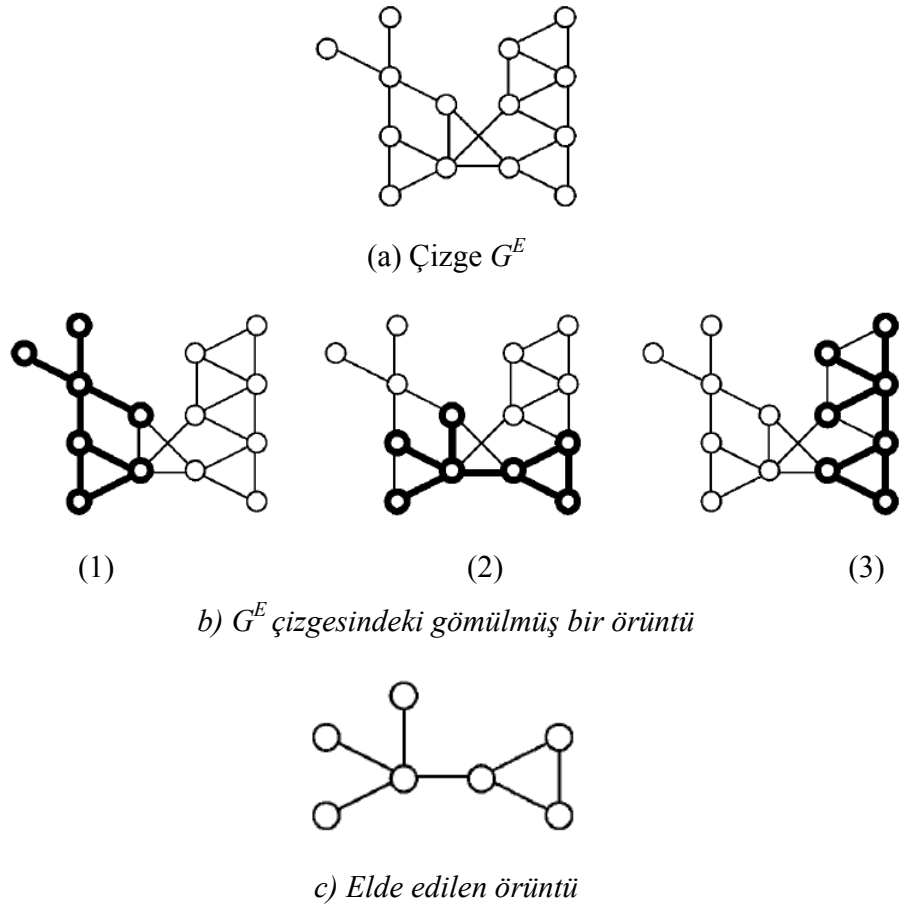
2.3 Tek Bir Çizgede Tekrarlayan Alt çizge Madenciliği Problemi

2.3.1 Problem Tanımı

ÇK-TAÇ yöntemleri çoğunlukla çok sayıda çizgeden oluşan veri kümelerini kullanmaktadır. Fakat gerçek hayattaki çizge tabanlı veri madenciliği uygulamalarının birçoğunda tek bir çizge ile ifade edilen yapısal veriler kullanılmaktadır (Örneğin çeşitli sosyal ağ uygulamaları) [Kuramochi and Karypis, 2005., Bringmann and Nijssen, 2008, Mukherjee and Holder 2004]. Bu tarz uygulamalarda nesneler, (örneğin: aktörler, gruplar ve olaylar) ve bunlar arasındaki çeşitli ilişkiler veya özellikler tek bir çizge veri yapısı ile ifade edilir. Çok sayıda düğüm ve kenardan oluşan bir çizge ile ifade edilen veri yapısından kurallar çıkarmak, mantık yürütmek ve ağ yapısındaki kişilerin ya da grupların davranışlarını modellemek günümüzde uzmanların ilgisini çekmektedir [Coffman et al., 2004].

Örnek vermek gerekirse her düğümün bir insanı temsil ettiği 30000 düğümden oluşan gerçek bir ağ yapısını düşünelim. Ağ yapısındaki insanların davranışlarını modellemek gerektiğinde bunları ayrı ayrı incelemek zaman alacaktır. Bu tarz bir

Tek çizge probleminde örüntünün sıklığı *Destek değeri** (“support value”) ya da *Destek yüzdesi değeri** olarak da geçmektedir. Bu değer *Destek yüzdesi* (Bkz. Bölüm 2.2.1) değerinden biraz daha farklı bir şekilde hesaplanır. Bir örüntünün G^* çizgesinde bulunma sıklığı, örüntünün G^* çizgesinde kaç kere geçtiğini göstermektedir. Elde edilen alt çizgeler ayrık ya da kesişiyor olabilir. Bu sebeple sıklığın hesaplanmasında iki yöntem kullanılmaktadır. İlkinde çizgeden elde edilen iki farklı alt-çizge eğer kenarlarından en az biri farklıysa farklı kabul edilir (şekil 2.10 (b-1) ve şekil 2.10 (b-2)) ve bu şarta göre örüntünün tekrarlanma sıklığı hesaplanır [Kuramochi and Karypis, 2005]. İkinci yöntemde ise iki alt çizge eğer tüm kenarları farklı ise farklı çizgeler olarak kabul edilir (şekil 2.10 (b-1) ve şekil 2.10 (b-3)). Yöntemlerin seçimi veri kümesine ve isteğe bağlıdır. Eğer örüntüler tamamen ayrık değil ise belli bir miktar çakışmaya tolerans tanınması gerekebilmektedir.



Şekil 2.10, Örnek bir çizge ve elde edilen örüntü [Kuramochi and Karypis, 2005].

Tanımlar 2.2:

- Etkileşim Çizgesi tek bir çizgeden oluşur ve $G^*=(V^*,E^*)$ şeklinde gösterilir.
- *Örüntü adayı* * CS^* , $G^{CS}=(V^{CS^*},E^{CS^*})$ şeklinde tanımlanır ve G^* nin bir alt çizgesidir. $V^{CS^*} \subseteq V^*$ ve $E^{CS^*} \subseteq E^*$ yi sağlar.
- *Örüntü adayının*, G^* çizgesinde bulunma oranı *Destek yüzdesi* $^*(CS^*)$ iki yöntem ile hesaplanır. (Yöntemlerin seçimi isteğe veya veri kümesine bağlıdır.)
 1. **Çakışmayan örüntüler:** Birbiri ile çakışmayan çizgeler örüntü olacak şekilde G^* çizgesinde tekrarlayan örüntü sayısının bulunması.
 2. **Çakışan örüntüler için:** İki örüntü karşılaştırılırken en az k adet kenarı farklı ise iki örüntüyü farklı örüntüler olarak kabul ederek G^* çizgesinde tekrarlayan örüntü sayısının bulunması. K değeri kullanıcı tarafından isteğe bağlı olarak veya veri kümesine bağlı olarak seçilmektedir.
- *Destek yüzdesi* * değeri için eşik değeri kullanıcı tarafından belirlenen *destek eşik yüzdesi* * değeridir. Eğer bir örüntü adayının *destek yüzdesi değeri* * , *destek eşik yüzdesi* * den büyük ise *örüntü adayı* * CS^* , *örüntü* * olarak tanımlanır ve FS^* olarak gösterilir. *Örüntü* * , k_1 kenarlı ve k_2 düğümden oluşan bir alt çizgedir.

$$M_{\text{Örüntü}^*}^E(i, j) = \begin{cases} l(v_i), i = j \\ l(e_{i,j}), i \neq j \end{cases} \quad (2.12)$$

şeklinde gösterilir. Örüntü aşağıdaki özellikleri sağlar.

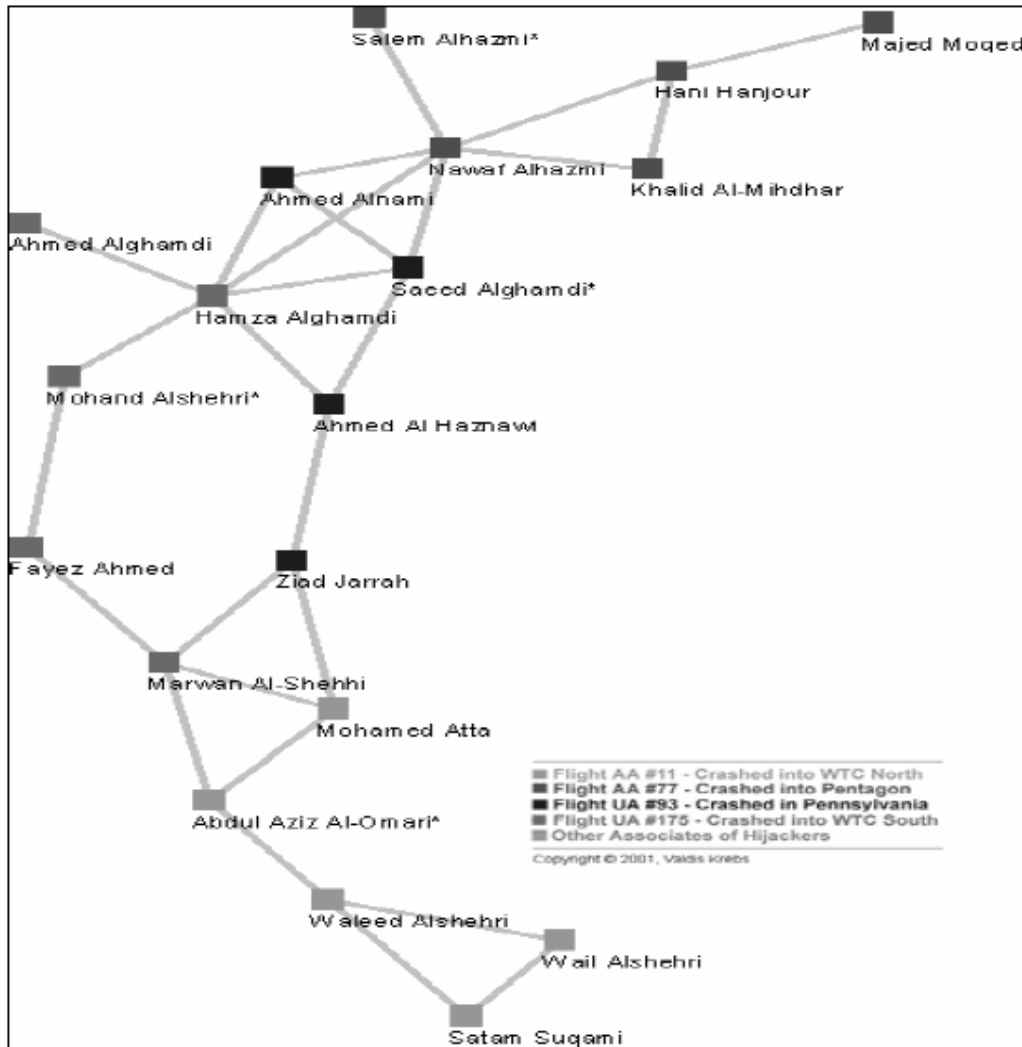
$$\begin{aligned} V(FS^E) &\subseteq V(CS^E) \subset V(G^E) \\ E(FS^E) &\subseteq E(CS^E) \subset E(G^E) \end{aligned} \quad (2.13)$$

Tanım 2.3: Problemin Tanımı

Düğümleri ve kenarları ağırlıklı bir çizge G^E verilsin. *Destek yüzdesi* ^{E} , *destekesikyüzdesi* ^{E} değerinden büyük olan alt çizgeleri bulunması.

ağı sağlanmaktadır. Bu sayede örgütün başındaki kişi ile eylemi uygulayacak olan kişi doğrudan temas etmeyip, örgütteki önemli kişilerin yakalanma ihtimali azaltılmış olmaktadır.

Bu tarz uygulamalarda gizlilik esas olduğu için klasik veri madenciliği uygulamalarından farklıdır. Bu bilgiler ışığında çeşitli veri madenciliği yöntemleri ile eylem öncesindeki teröristlerin davranışlarından eylem hakkında bilgiler, terörist grupları hakkındaki bilgiler elde edilmeye çalışılmaktadır.



Şekil 2.12, Terörist ağları [Mukherjee and Holder, 2004].

Bunun için çeşitli yöntemler uygulanmaktadır [Phua et al, 2005]. Çizge tabanlı veri madenciliği yöntemleri ise bu yöntemlerden bir tanesidir [Noble and Cook, 2003]. Çizge tabanlı örneklerden benzer alt yapı çıkarma yöntemlerinden SUBDUE, tek bir çizgede tekrarlayan alt çizgeleri bulma problemine uyarlanarak bu alanda kullanılmıştır [Mukherjee and Holder, 2004]. SUBDUE yöntemi veriyi sıkıştırarak tekrarlayan alt parçaları bulmaktadır. [Mukherjee and Holder, 2004] de terörist ağlarında tekrarlayanlar alt parçaların önemli bilgiler içermediği öne sürmüştür. İyi bir şekilde sıkıştırılmış bir ağ yapısının önemli nitelikler göstereceğinden SUBDUE yöntemini kullanmıştır.

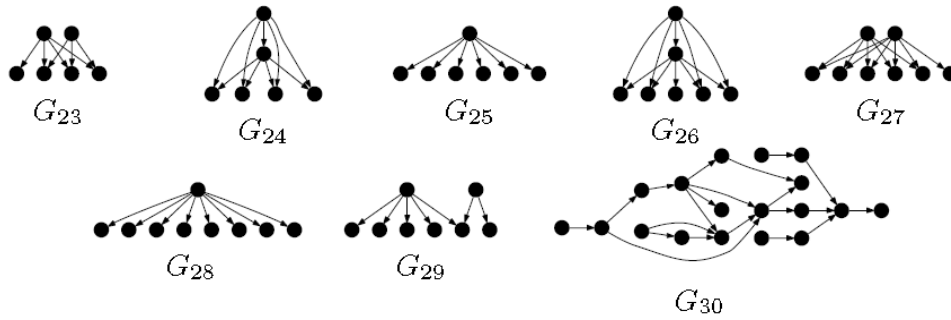
2.3.2.2.2 Tavsiye Ağları:

Internet sayfalarından film kiralarken ya da Amazonda bir kitap satın alırken genelde görülen tavsiye ürünler son zamanlarda yaygınlaşmaktadır. Bu konu birçok alanda özellikle tavsiye sistemleri (recommendation systems), güven sistemleri (güven ağları), işbirliği ağları (collaboration systems) gibi birçok alanda çalışılmaktadır. Son zamanlarda gelişen bu yeni çalışma alanı ile veri madenciliği yöntemleri ile ağ yapısına ait özellikler çıkarılmaya çalışılmaktadır. Aşağıda bu alanda yapılmış örüntü madenciliği ile ilgili bir çalışmadan bahsedilecektir.

Yapılan bu çalışmada internet üzerinden ürünler satan bir perakendeciden elde edilen veriler kullanılmıştır [Leskovec et al., 2006]. Bu siteden kitap, DVD, video ya da müzik ürünü satın alan müşteriler e-posta ile istediği kişilere tavsiye etmektedir. Düğümler müşterileri, yönlü kenarlar ise v ve w düğümleri arasındaki, v nin w ye tavsiye ettiği ürün p yi ve zaman t yi tutmaktadır. Tavsiye ağında 15,646,121 tavsiye yapılmıştır. 3,943,084 müşteri 711 gün içinde 542,719 ürün satın almıştır. Her ürün tipi için ayrı bir ağ yapısı oluşturulmuştur. Örnek vermek gerekirse aynı müşterilerin DVD için yaptığı tavsiyeler DVD ağında, kitaplar için yaptığı tavsiyeler kitap ağında tutulmaktadır. Eğer bir müşteriden hiç tavsiye gitmemişse, o ağ yapısı için o düğüm ağ dışında bırakılmaktadır. Veri kümesinde kitap ağının çok fazla düğümden oluştuğu ama

ortalama iki kitap tavsiye edildiği, DVD tavsiye ağında ise daha az düğüm olduğu ama ortalama 10 DVD tavsiye edildiği gözlemlenmiştir.

Sonuç olarak elde edilen sık alt çizgelere göre ağ yapısı hakkında bilgi edinilmeye çalışılmıştır. Şekil 2.13 de veri kümesinden elde edilen sıklıkla tekrarlayan alt çizge sınıfları görülmektedir. G_{23} ve G_{27} aynı arkadaş grubundaki alt çizgelerinde aynı kişilere tavsiyede bulunmalarına rağmen birbirlerine tavsiye de bulunmadıkları gözlemlenmiştir. G_{24} ve G_{26} alt çizgelerinde bir düğüm diğerine tavsiye ederken, iki düğümden de diğer düğümlere tavsiyeler gitmiştir. G_{25} , G_{28} , G_{29} de ise tek bir düğümden dallanma görülmektedir. Fakat sadece ilk düğümden dallanma görülmektedir. Elde edilen örüntüler müşteriler hakkında bilgiler vermektedir.



Şekil 2.13, Tavsiye ağından elde edilen çeşitli alt çizgeler [Leskovec et al., 2006].

2.3.2.2.3 WEB, Sosyal Ağlar ve Biyolojik Ağlar İle İlgili Diğer Uygulamalar

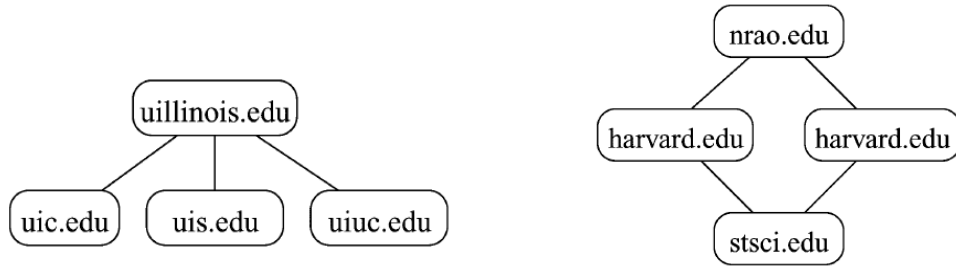
Çizge veri kümelerinde tekrarlayan alt çizge madenciliği problemi ile ilgili birçok konferans ve dergi makalesi literatürde mevcuttur. Tek çizge alanında ise çok sayıda konferans makalesi yayınlanmasına rağmen az sayıda dergi yayını bulunmaktadır. Basılan dergi makalelerden biri olan [Kuramochi and Karypis, 2005] de farklı alanlardaki gerçek veri kümelerinde önerdikleri yöntemlerini uygulamışlardır. Aşağıda önce veri kümelerinden ve daha sonra da yöntemden bahsedilecektir.

1. Web Veri Kümesi:

Yapılan bir çalışmada 2002 Google programlama yarışmasından elde edilen internet veri kümesi kullanılmıştır. Orijinal veri kümesi “edu” uzantılı internet sayfaları

ve onlar arasındaki bağlantıları içermektedir. Kolaylık olması açısından düğümler arasındaki kenarlarda yön bilgisi içermeyen bir çizge yapısı kullanılmıştır. Bu veri yapısındaki kenarlar, çeşitli edu uzantılı sayfalar ile çeşitli alt sayfalar arasındaki bağlantıları göstermektedir. Aynı web sayfalarının alt sayfaları arasındaki bağlantılar veri kümesinden çıkarılmıştır. Elde edilen veri kümesinden, önerilen yöntem ile tekrarlayan alt çizgeler elde edilmiştir.

Makalede elde edilen alt çizgeler belirli kurumlar arasındaki sıkça tekrarlayan çapraz bağlantı yapılarını göstermektedir. Diğer bir deyişle bu yapılar ortak işbirliği yapan üniversiteleri, disiplinler arası takımları ve belirli bir konudaki toplulukları göstermektedir. Şekil 2.14 de veri kümesinden elde edilen iki örüntü görünmektedir. (a) kısmında yıldız topolojili bir örüntü görülmektedir. Illinois üniversitesinin internet sayfasından üniversitenin üç farklı yerdeki kampüslerinin (Chicago, Urbana-Champaign ve Springfield) web sayfalarına bağlantı görülmektedir. (b) kısmındaki örüntüde ise Harvard Üniversitesi ile Ulusal Radyo Astronomi Gözlem-evi (nrao.edu) ve Uzak Teleskop Bilim Enstitüsü arasındaki bağlantıları göstermektedir.



(a) Illinois Üniversitesi sistemi

(b) Astronomi ile alakalı web sayfaları

Şekil 2.14, WEB veri kümesinden elde edilen örüntü örnekleri [Kuramochi and Karypis, 2005].

2. Alıntı Veri Kümesi:

İkinci veri kümesi olarak KDD Kupa 2004 yarışmasında kullanılan alıntı veri kümesi kullanılmıştır. Çizge ile ifade edilen alıntı sosyal ağ yapısında her düğüm bir makaleyi ve her kenar iki makale arasında alıntı olduğunu göstermektedir. Yöntemde yönsüz çizge kullanıldığı için alıntının yönü ele alınmamıştır. Orijinal veri kümesinde

düğümelerde anlamlı veri olmadığından makalelerin özetleri dikkate alınarak konularına göre makaleler kümelenmiş (50 küme) ve her konuya bir numara atanmıştır. Diğer bir deyişle çizgenin düğüm değerleri makalelerin konularını göstermektedir. Kenar değerlerine ise iki makalenin yayınlanma yıllarının farkı atanmıştır. Bu veri kümesinde farklı alanlardaki yayınlardaki alıntılanma modellenmiştir.

3. Amino asit Bağlanma Haritası:

Diğer bir veri kümesi olarak Protein Veri Bankasından elde edilen biyolojik ağ yapısı kullanılmıştır. Çizgeler ile ifade edilen amino asit ağ yapısında amino asitler düğümleri ve iki amino asit birbiri ile ardışık sırada ise veya üç boyutlu protein yapısında bağlı ise, iki düğüm arasındaki kenarı göstermektedir.

4. Kimyasal Bileşik Veri kümesi:

Daha önce çizge veri kümelerinde tekrarlayan alt çizge madenciliği probleminde bahsedilen veri kümesine benzer olarak çizgedeki düğümlerin atomların tipini, kenarlar ise bağların tipini gösterdiği biyolojik ağ yapısında 223644 bileşik birleştirilerek tek bir çizge ile ifade edilen bir ağ yapısı oluşturulmuştur.

Yöntem:

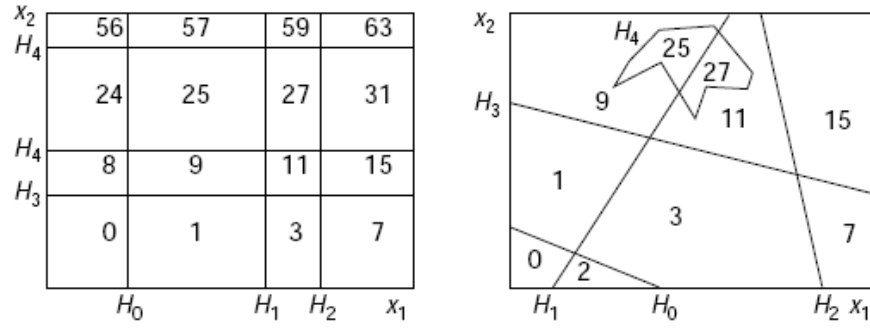
Bu çalışmada iki yöntem HSIGRAM ve VSIGRAM geliştirilmiştir [Kuramochi and Karypis, 2005]. Bu yöntemler daha önceden çizge veri kümelerinde tekrarlayan alt çizge madenciliği problemini çözmek için önerilen bazı yöntemlere dayanmaktadır. İki yöntemde de sıklıkla tekrarlayan alt çizgeleri tutan bir kafes (ağaç) yapısı kullanılmıştır. Kafes yapısında k-ıncı seviyede k düğümlü alt çizgeler bulunmaktadır. İki yöntemin birbirinden farkı ise kafes yapısındaki düğümleri oluşturma şeklidir. HSIGRAM yatay bir yöntem (breadth-first algorithm) kullanırken, VSIGRAM ise dikey olarak (depth-first algorithm) düğümlerdeki alt çizgeleri belirlemektedir. HSIGRAM yönteminde kafes yapısında k+1 inci seviye oluşturulurken k. seviyedeki k düğümlü alt çizgeler ikişerli gruplar halinde birleştirilerek oluşturulur (*alt çizge yaratma*). Daha sonra elde edilen tüm alt çizgelerin tekrarlanma sıklığı hesaplanır. Eğer bir aday alt çizge eşik değeri den

küçükse o alt çizge kafes yapısına konmaz. Bu işlem hiçbir alt çizge yaratılamayınca kadar devam eder.

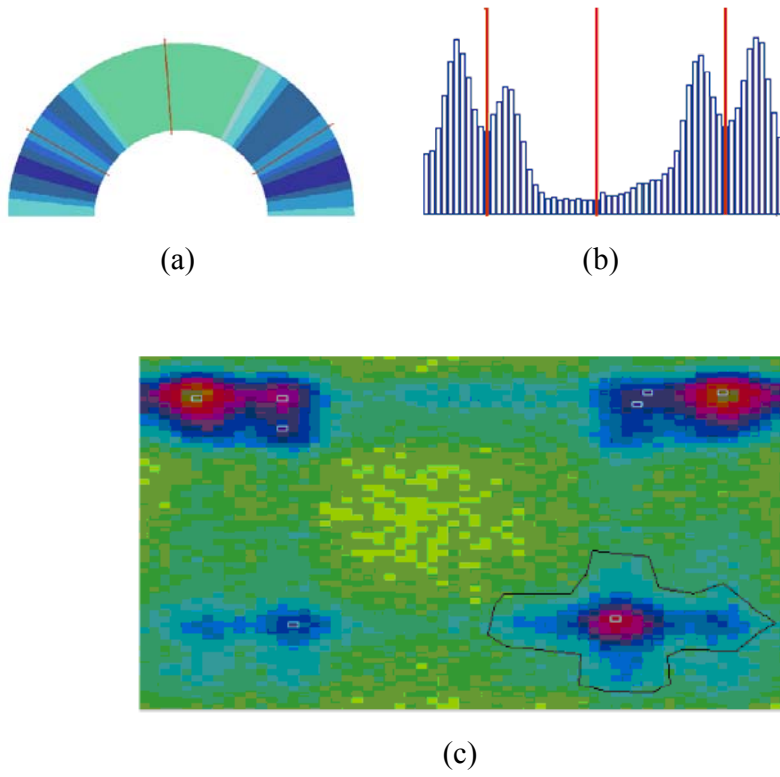
2.4 Görsel Veri Madenciliği

Literatürde görsel veri madenciliğine dayanan birçok yöntem geliştirilmiştir [Oliveira and Levkowitz, 2003]. Bu bölümde görsel veri madenciliğinin ve özellik uzayı kavramının kullanıldığı ve temelde önerdiğimiz yönteme yakın olan bir yöntemden bahsedilecektir. HD-Eye görsel veri madenciliği yönteminde OptiGrid yöntemine dayanarak iteratif biçimde verinin en iyi şekilde kümelerine ayırabilecek izdüşümleri aranır ve ayıraçlar belirlenir [Hinneburg et al., 1999]. Kümelerin düzgün bir şekilde ayrılabilmesi için verinin tek bir izdüşümünü bulmak yerine, veri iteratif olarak alt parçalara bölünür. Her iterasyonda en iyi izdüşümü bulunarak alt kümeler hesaplanır. Her iterasyonda veriyi alt parçalara ayıracak ayıraçları belirlemek için alt parçalardaki en düşük yoğunluk noktalarına bakılır. Yoğunluk noktalarını belirleyebilmek için nokta yoğunluk fonksiyonu olarak çekirdek yoğunluk tahmini yöntemi (kernel density estimation) kullanılmıştır.

Kümelerin bilinmeyen karakterlerinden dolayı, kümeleme yöntemlerinde parametrelerin belirlenmesi zordur. HD-Eye yöntemi, OptiGrid yöntemini geliştirerek ve çeşitli renkli grafiksel gösterimler kullanarak ayıraçlar belirler (Bkz. şekil 2.16). Elde edilen ayıraçlardan çok boyutlu düzensiz ağ yapısı (irregular grid) oluşturulur (Bkz. şekil 2.15). Verideki tüm noktalar bu ağ yapısına düşürülür. Kümeleri bulabilmek için, noktaların fazla düştüğü ağ hücreleri belirlenir. Bir eşik değeri kullanarak veri filtrelenir ve kümeler çıkarılır. Bu sayede kümelerin şeklinden bağımsız olarak, nokta kümeleri elde edilir.



Şekil 2.15, Düzenli (a) ve düzensiz (b) ağ yapıları [Hinneburg et al., 1999].



Şekil 2.16, a) Bir boyutlu renk gösterimi, b) bir boyutlu histogram yoğunluk gösterimi. c) iki boyutlu renk histogramı gösterimi

2.5 Videolardaki Nesne Eşleme

Bu bölümde çizgeler ile ifade edilebilen yapısal veri kümelerinin kullanıldığı bir problemten ve önerilen bir yöntemden bahsedilecektir [Sivic and Zisserman; 2003]. Videolardaki Nesne Eşleme Problemi resimlerden oluşan bir veri kümesinde aynı

nesneleri bulmayı amaçlamaktadır. Bu problemin zorlukları bakış açısındaki farklılıklar, aydınlatmadaki değişiklikler, bazı parçaların bazı kısımlarının görünmemesidir.

Bu yaklaşımda her nesne örtüşen bölgelerin birleşmesi ile oluşturulur. Her bölge için ise, o bölgenin görüntüsünden elde edilen özellik vektörleri oluşturulur. Özellik vektörleri, vektör nicemlemesi yöntemi ile kümelere ayrılırlar. Bir nesnenin tanınması (sınıflandırılması) için özellik vektörlerine en yakın komşuluk algoritması uygulanır. Diğer bir deyişle yeni bir film karesi geldiğinde film karesindeki tanımlayıcılar en yakın kümeye atanırlar

Videoda nesne eşleme probleminde bahsedilen zorluklar veya problemler bölüm 4.5 de bahsedilen gürültü kavramına örnek teşkil etmektedir. Bu tarz bir uygulamada bir nesne çoğunlukla birebir olarak bulunamayacaktır. Aranılan nesne videoda bu problemlerden dolayı ufak farklılıklar ile tekrarlayabilmektedir. Bu sebeple önerilen yöntemin bu tarz ufak değişikliklere toleranslı olması gerekmektedir.

Videolardaki Nesne Eşleme Problemine getirilen bu yaklaşım önerilen yöntemle temelde çok benzerdir. Bu yaklaşımda önce veri kümesi özellik uzayına dönüştürülmekte, daha sonra veri kümesi kümelere ayrılmaktadır (vektör nicemlemesi) ve yeni gelen çizge sınıflandırılmaktadır (en yakın komşuluk algoritması). Önerilen yöntemde ise veri kümesi özellik uzayına dönüştürülmekte daha sonra hiyerarşik bir kümeleme yöntemi ile kümelere ayrılmakta ve son olarak şablon çizge kullanılarak görüntü yeniden çıkarılmaktadır (1 en yakın komşuluk algoritması). Bu basamaklarda kullanılan yöntemler farklı da olsa önerilen yöntem ile bu yaklaşım benzerlik göstermektedir. Makalede önerilen yaklaşımla tezde önerilen yaklaşımdaki bir diğer benzerlik ise nesnelerden elde edilen özellikleri özellik uzayına aktardıktan sonra görüntü karelerindeki nesnenin konumu gibi bilgileri kaybetmesidir.

2.6 Hiyerarşik Kümeleme Algoritmaları

Önerdiğimiz yaklaşımda hiyerarşik bir kümeleme yöntemi kullanılmıştır. Bu sebeple bu bölümde, hiyerarşik kümelemeden ve en bilinen hiyerarşik kümeleme yöntemlerinden bahsedilecektir. Hiyerarşik kümeleme algoritmaları veriyi çoğunlukla ikili ağaç yapısında ya da dendrogram yapısında tutmaktadırlar [Larose, 2005]. Hiyerarşik kümeleme algoritmaları yığılma (agglomerative) algoritmaları ve bölücü (divisive) algoritmalar olmak üzere ikiye ayrılırlar.

i) Bölücü algoritmalar: Yığılma algoritmalarının tersi şekilde çalışır. En başta tüm veri kümesi tek kümeye aitken kümeler bölünerek son kümeler elde edilir. Bölücü kümeleme algoritmalarının işlem zamanı uzundur.

ii) Yığılma algoritmaları: Her kümede tek bir nesne olacak şekilde veri kümesini kümelere böler. Daha sonra kümeler birleştirilerek son kümeler elde edilir. Yığılma algoritmalarında genel algoritması aşağıdaki gibi özetlenebilir: (N örnekli bir veri kümesi için)

1- N küme ile başlanılır. N küme için yakınlık matrisi hesaplanır.

2- Kümeler arasında en kısa uzaklık aranır. En yakın iki küme birleştirilir.

$$D(C_i, C_j) = \underset{\substack{1 \leq m, l \leq N \\ m \neq l}}{\text{en küçük}} D(C_m, C_l) \quad (2.14)$$

($D(*,*)$ bir yakınlık fonksiyonudur. Yakınlık fonksiyonlarından bölüm 2.6 da bahsedilecektir.)

3- Yeni küme ile diğer kümeler arasında uzaklık hesaplanarak yakınlık matrisi güncelleştirilir.

4- Tüm benzer özellikteki örnekler aynı kümeye düşene kadar 2 ve 3 üncü adımlar tekrar edilir.

Yığılma algoritmalarından en bilinenleri tek bağlantı (en yakın komşuluk algoritması) ve bütün bağlantı teknikleridir. En yakın komşuluk algoritmasında iki küme arasındaki uzaklığı belirlerken iki farklı kümedeki en yakın iki nesne arasındaki uzaklık

hesaplanır. Bütün bağlantı tekniğinde ise iki küme arasındaki uzaklığı hesaplamak için iki farklı küme deki en uzak iki nesnenin uzaklığı esas alınır. Ortalama bağlantı tekniği [Xu and Wunsch, 2005] ise yayılma algoritmalarından biridir ve bölüm 4.3 de anlatılacaktır.

Hiyerarşik kümeleme yöntemi esnek ve parametrik olmayan bir yöntemdir. Verinin karakteristiğini diğer yöntemlere göre daha az varsayar ve daha az ön bilgi gerektirir [Murtagh, 1983]. Bu sebeple literatürde sıkça kullanılmaktadırlar. Hiyerarşik kümeleme yöntemleri gürültüye ve aykırı değerlere duyarlıdır ve küre biçiminde kümeler çıkarmaya elverişlidir. Ayrıca HK yöntemleri en az $O(n^2)$ zaman karmaşıklığına sahiptir. Literatürdeki bazı diğer kümeleme yöntemlerine göre zaman karmaşıklığı yüksek sayılabilir (çizelge 2.4).

Çizelge 2.4., Kümeleme yöntemlerinin zaman karmaşıklığı ve yüksek boyutlu veri kümeleri açısından değerlendirilmesi [Xu and Wunsch, 2005]. (D: Veri kümesinin boyut sayısı, N: Veri kümesindeki eleman sayısı, K: Küme sayısı)

Kümeleme Algoritması	Zaman ve Hafıza Karmaşıklığı
K-ortalama	$O(NKd)$ (zaman) $O(N+K)$ (hafıza)
Bulanık c-ortalama	$O(N)$ 'e yakın
Hiyerarşik Kümeleme	$O(N^2)$ (zaman) $O(N^2)$ (hafıza)
CLARA	$O(K(40+K)^2+K(N-K))$ (zaman)
CLARANS	Tüm prformansta ikinci dereceden
BIRCH	$O(N)$ (zaman)
DBSCAN	$O(N\log N)$ (zaman)
CURE	$O(N_{sample}^2 \log N_{sample})$ (zaman) $O(N_{sample})$ (hafıza)
DalgaKüme	$O(N)$ (zaman)
DENCLUE	$O(N\log N)$ (zaman)
FC	$O(N)$ (zaman)
CLIQUE	Nesnelerin sayısı ile lineer, boyutların sayısı ile ikinci dereceden
OpriGrid	$O(Nd)$ ile $O(Nd\log N)$ arasında
ORCLUS	$O(K_0^3 + K_0Nd + K_0^2d^3)$ (zaman) $O(K_0d^2)$ (hafıza)

2.7 Benzerlik Ölçüsü

Bir çift nesne arasındaki benzerlik ya da farklılığı ölçmek için çeşitli benzerlik ölçüleri kullanılmaktadır. Kümeleme yöntemlerinde ise yakınlığı ölçmek için uzaklık olarak adlandırılan benzerlik ölçüleri kullanılır. [Backer and Jain,1981] da belirtildiği gibi

“Küme analizinde bir nesne gurubu birden fazla, az ya da çok homojen alt gruplara subjectif benzerlik ölçüsü kullanarak ayrılırlar (Ör. İlgi çekici kümeler yaratılabilecek şekilde seçilir). Öyle ki bir alt gruptaki nesneler arasındaki benzerlik, değişik alt gruplardaki nesnelerden büyük olmalıdır.”

Bir veri nesnesi bir özellik ile ifade edilir. Özellikler sayısal olabilir, nitelik belirtebilir, ikili (0 ya da 1) olabilir. Özelliğin cinsi benzerlik ölçüsünün seçiminde büyük rol oynar. N örüntü içeren bir veri kümesinde benzerlik *yakınlık matrisi* kullanarak belirlenir. Yakınlık matrisi $N \times N$ lik bir simetrik matristir.

Aşağıda literatürde sıklıkla kullanılan bazı benzerlik ölçülerinden örnekler verilecektir.

- Minkowski Uzaklığı:

$$D_{ij} = \left(\sum_{l=1}^d |x_{il} - x_{jl}|^{1/n} \right)^n \quad (2.15)$$

- Öklid Uzaklığı: Minkowski uzaklığının $n=2$ durumu için özelleşmiş şeklidir. En sık kullanılan uzaklık fonksiyonudur.

$$D_{ij} = \left(\sum_{l=1}^d |x_{il} - x_{jl}|^{1/2} \right)^2 \quad (2.16)$$

- Şehir Blok Uzaklığı:

$$D_{ij} = \sum_{l=1}^d |x_{il} - x_{jl}| \quad (2.17)$$

- Sup Uzaklığı:

$$D_{ij} = \max_{1 \leq l \leq d} |x_{il} - x_{jl}| \quad (2.18)$$

- Mahalanobis Uzaklığı:

$$D_{ij} = (x_i - x_j)^T S^{-1} (x_i - x_j) \quad (2.19)$$

S= Grup içinde kovaryans matrisi

- Pearson Bağıntısı:

$$D_{ij} = (1 - r_{ij}) / 2 \quad r_{ij} = \frac{\sum_{l=1}^d (x_{il} - \bar{x}_i)(x_{jl} - \bar{x}_j)}{\sqrt{\sum_{l=1}^d (x_{il} - \bar{x}_i)^2 \sum_{l=1}^d (x_{jl} - \bar{x}_j)^2}} \quad (2.20)$$

- Nokta Simetri uzaklığı: x_i nesnesi ile x_r referans noktası arasındaki uzaklığı hesaplar.

$$D_{ir} = \min_{j=1, \dots, N \text{ and } j \neq i} \frac{\|(x_i - x_r) + (x_j - x_r)\|}{\|(x_i - x_r)\| + \|(x_j - x_r)\|} \quad (2.21)$$

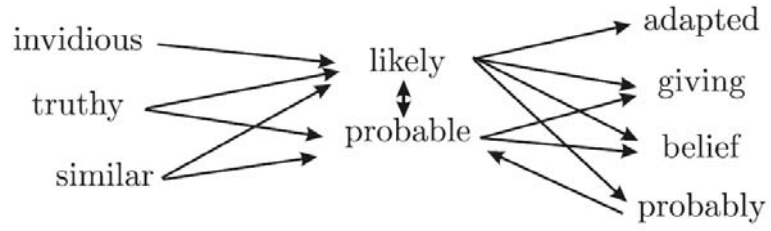
- Kosinüs Benzerliği:

$$S_{ij} = \cos \alpha = \frac{x_i^T x_j}{\|x_i\| \|x_j\|} \quad (2.22)$$

2.8 Çizgelerin Dügümleri Arasında Benzerlik Ölçümü

[Blondel et al, 2004] de çizgelerin düğümleri arasındaki benzerliğin bulunması problemine yaklaşım getirilmiştir ve *benzer kelimelerin çıkarılması* ve *internette arama* uygulama alanlarına uygulanmıştır. Benzer kelimelerin otomatik çıkarılması uygulaması, “benzer kelimelerin tanımlarında ortak birçok kelime vardır” varsayımına dayanır. Örnek olarak alınan Webster sözlüğünden bir sözlük çizgesi G elde edilir. Sözlükteki her kelime G çizgesinin düğümünü oluşturur. Eğer v kelimesi u kelimesinin tanımında var ise G sözlük çizgesinde u ve v düğümleri arasında bir kenar çizilir. Daha sonra sorgu kelimesine göre bu kelimenin tanımında olan kelimeleri ve bu kelimeyi tanımında kullanan kelimeleri içeren bir yakınlık çizgesi G_w çıkarılır. G_w çizgesi G çizgesinin bir alt çizgesidir.

Şekil 2.17 de İngilizcedeki “likely” sorgu kelimesinden elde edilen yakınlık matrisi görülmektedir. Yakınlık matrisinde bu kelimenin tanımında olan kelimeler ve bu kelimeyi tanımında kullanan kelimeler görülmektedir. G_w çizgesi elde edildikten sonra tüm düğümlerin benzerlik puanı hesaplanır. En yüksek merkezi puana sahip kelimeler sorgu kelimesine eş anlamlı olmaya aday kelimelerdir.



Şekil 2.17, İngilizcedeki “likely” kelimesinden elde edilen yakınlık çizgesi [Blondel et al, 2004].

Benzerlik puanı aşağıdaki şekilde hesaplanır. G_A ve G_B iki yönlü çizge olsunlar. n_A , A çizgesinin düğüm sayısı; n_B ise B çizgesinin düğüm sayısı olsun. İki çizgenin benzer olup olmadığını anlamak için $n_B \times n_A$ lik bir benzerlik matrisi S oluşturulur. Bu matrisin değerleri s_{ij} ; G_A nın düğümü j nin, G_B nin düğümü i ye olan benzerliğini diğer bir deyişle benzerlik puanını göstermektedir. S benzerlik matrisini oluşturmak için eşitlik 2.23 yakınsayına kadar tekrarlanır ve S yakınlık matrisi elde edilir. S_0 matrisinin değerleri 1 dir.

$$S_{k+1} = B S_k A^T + B^T S_k A \quad (2.23)$$

3 PROBLEMİN TANIMI VE HİPOTEZ

Tez çalışmasında ÇK-TAÇ problemine bir yaklaşım getirilmiştir. Çizge madenciliği yöntemlerinde örüntü, bağlantılı bir alt çizge olarak tanımlanmaktadır. Bu problemi çözmek için literatürde tekrarlayan alt çizgelerin çıkarılması yöntemleri uygulanmaktadır (Bkz. Bölüm 2.2.2). Bu sebeple tekrarlayan alt çizgelerin çıkarılması ile ilgili problem bu bölümde ayrıntılı olarak tanımlanacaktır. Bölüm 3.1 de çizgeler ile ilgili gerekli tanımlar yapıldıktan sonra ÇK-TAÇ problemi ayrıntılı olarak tanımlanacaktır. Bölüm 3.2 de çizge izomorfizm ve alt çizge izomorfizm ile ilgili tanımlamalara yer verilecektir. Bölüm 3.3 de çizgelerde benzerlik tanımlanacaktır. Son olarak da bölüm 3.4 te tezin dayandığı hipotezden bahsedilecektir.

3.1 Çizge Tabanlı Veri Kümelerinde Tekrarlayan Alt çizgelerin Çıkarılması

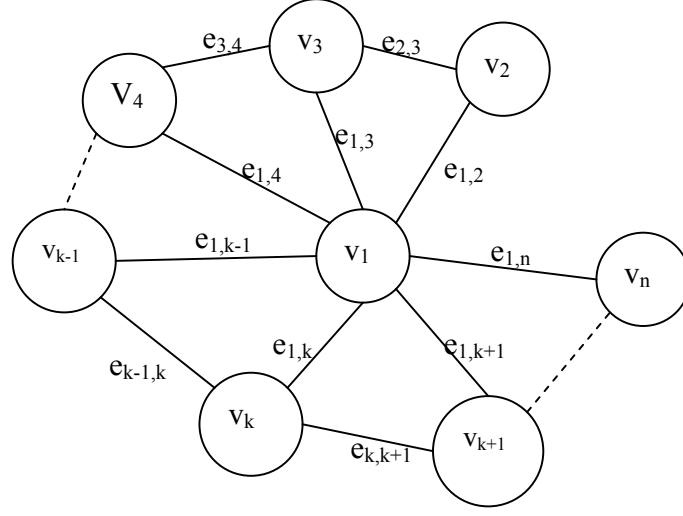
Tanım 3.1: Çizgeler ile ilgili özellikler ve tanımlar

Bir çizge $G=(V, E)$, düğümler ve düğümlerin birleştiren kenarlardan oluşur. V , G çizgesine ait düğümler kümesidir ve $E \subset V \times V$ (veya $E \subset |V|^2$) ise G çizgesinin kenarlar kümesidir (Bkz. şekil 3.1).

- G çizgesindeki düğümlerin sayısı $|V|$ ve kenarların sayısı ise $|E|$ olarak gösterilir.
- G çizgesindeki iki düğüm $u, v \in V$ birbiri ile bir kenar $e \in E$ ile bağlı ise $e = (u, v)$ şeklinde gösterilir. Bu durumda u, v düğümleri bitişik düğümlerdir denir.

Çizgelerin kenar ve düğümleri bilgi içerebilir. Eğer bu bilgi isim ya da sayı ise çizgeye *etiketlenmiş çizge* denir. *Ağırlıklı çizgelerde* kenar ya da düğümlerinde sayısal ağırlık değeri içerirler. Kenarları ağırlıklı bir $G=(V, E, w_e)$ çizgesinin ağırlık fonksiyonu $w_e : E \rightarrow R$ şeklinde gösterilir. $w(u, v)$, (u, v) kenarının ağırlık değeridir. Düğümleri

ağırlıklı $G=(V, E, w_v)$ çizgesinin ağırlık fonksiyonu $w_v : V \rightarrow R$ şeklinde tanımlanır.



Şekil 3.1, N düğümlü bir çizge örneği.

- Ağırlık matrisi M_G

$$M_G(i, j) = \begin{cases} l(v_i), & i = j \\ l(e_{i,j}), & i \neq j \end{cases} \quad (3.1)$$

şeklinde tanımlanır ($1 \leq i, j \in Z^+ \leq |V|$).

$$M(G) = \begin{bmatrix} l(v_1) & l(e_{1,2}) & l(e_{1,3}) & \dots & l(e_{1,i}) & \dots & l(e_{1,n}) \\ & l(v_2) & l(e_{2,3}) & \dots & l(e_{2,i}) & \dots & l(e_{2,n}) \\ & & l(v_3) & \dots & l(e_{3,i}) & \dots & l(e_{3,n}) \\ & & & \dots & \dots & \dots & \dots \\ & & & & l(v_i) & \dots & l(e_{i,n}) \\ & & & & & \dots & \dots \\ & & & & & & l(v_n) \end{bmatrix} \quad (3.2)$$

Şimdi tek bir sınıfa ait çizgelerden oluşan bir veri kümesinde problemi tanımlayalım. Veri kümesindeki tüm çizgeler aynı niteliktedirler.

Tanım 3.2: Özellikler ve Tanımlar (Tek sınıfa ait veri kümeleri için)

Problemi basitleştirmek için önce tek bir sınıfa ait örnekler için problemi tanımlayalım.

- GDS veri kümesi g adet çizge tabanlı örnekten oluşsun ($GDS = \{ G_1, G_2, \dots, G_i, \dots, G_g \}$). G_i , i -nci çizgeyi ifade etmektedir. GDS veri kümesindeki çizgelerin düğüm sayıları aynı olmak zorunda değildir.
- C bir sınıf olmak üzere $G \subseteq GDS$ ve $G \in C$ olsun. Diğer bir deyişle GDS deki tüm örnekler aynı sınıfa (C sınıfına) ait olsun.
- *Örüntü adayı alt çizge kümesi* CS sembolü ile gösterilir. CS , GDS veri kümesini oluşturan çizgelerden en az birinin alt çizgesidir ve

$$\begin{aligned} E(CS) &\subseteq E(G) \\ V(CS) &\subseteq V(G) \end{aligned} \quad (3.3)$$

sağlar. (G ise GDS veri kümesindeki çizgeleri temsil etmektedir.)

- *Örüntü adayının*, GDS deki çizgelerde bulunma oranı *destek yüzdesi*(CS) aşağıdaki şekilde tanımlanır.

$$Destek\ yüzdesi(X) = \frac{\sum_{G \in G^{DS}} f(X, G)}{|GDS|} \quad (3.4)$$

Denklemden $f(x, G)$ fonksiyonu x bir alt çizge olmak üzere aşağıdaki gibi tanımlanmıştır.

$$f(x, G) = \begin{cases} 1, & x \subseteq G \\ 0, & \text{diğer} \end{cases} \quad (3.5)$$

- *Bir örüntü adayının destek yüzdesi*, *destek yüzdesi*(CS) olarak gösterilir. Eşik değeri ise *destek eşik yüzdesi* olarak adlandırılır. Eğer örüntünün destek yüzdesi değeri, destek eşik yüzdesi değerinden büyük ise örüntü adayı *örüntü* olarak tanımlanır ve FS sembolü ile gösterilir.
- *Örüntü* bir alt çizgedir ve M_{Oruntu}^{DS} ağırlık matrisi ile gösterilir.

$$M_{Oruntu}^{DS}(i, j) = \begin{cases} w(v_i), & i = j \\ w(v_i, v_j), & i \neq j \end{cases} \quad (3.6)$$

- *Örüntü kümesi*, *örüntü adayı* kümesinin alt kümesidir. Bu sebeple örüntüdeki kenar kümesi, aday örüntüdeki kenar kümesinin alt kümesidir. Aynı şekilde örüntüdeki düğüm kümesi, aday örüntüdeki düğüm kümesinin alt kümesidir.

$$\begin{aligned} V(\text{Örüntü}) &\subseteq V(\text{Örüntü}^{aday}) \\ E(\text{Örüntü}) &\subseteq E(\text{Örüntü}^{aday}) \end{aligned} \quad (3.7)$$

Tanım 3.3: Tek sınıfa ait veri kümeleri için problemin tanımı

Yukarıda tanımlanan GDS bir veri kümesi olsun. Örüntü olarak adlandırılan ve *destek yüzdesi değeri*, *destek eşik yüzdesi* değerinden büyük ya da eşit olan örüntüleri bulunuz. (Örüntü, bağlantılı bir alt çizge olarak tanımlanmaktadır.)

Şimdi çok sınıflı çizgelerden oluşan bir veri kümesinde problemi tanımlayalım. K sınıflı bir veri kümesinde, çizgelerin her biri K adet çizge küme grubundan (sınıf) sadece birine aittir. Her sınıf kendi içinde benzer niteliklere sahip çizgeler içerir. Farklı sınıflara ait iki çizgenin nitelikleri birbirinden farklıdır. Sınıf kavramı aşağıdaki örnekte daha ayrıntılı olarak açıklanacaktır. Bir grup hukuk öğrencisi ve aralarındaki etkileşimleri gösteren sosyal ağ, bir çizge ile gösterilsin. Bu şekilde her üniversite için bir çizge oluşacak şekilde çizgeler grubu oluşturulsun. Aynı şekilde her üniversite için mühendislik öğrencilerinden oluşan ayrı bir sosyal ağ grubu oluşturulsun. Bu iki öğrenci grubundan oluşturulmuş veri kümesi iki sınıflı bir veri kümesine örnek gösterilebilir. Bu örnekte her sınıf kendi içinde benzer özellikler göstermekte ve diğer sınıfın özelliklerinden çoğunlukla farklıdır. Eğer iki sınıfa ait çizgelerde ortak benzer özellikler var ise (örneğin öğrenci olma durumu gibi) bu özellikler sınıfları ayırt edici özellikler olmadığı için dikkate alınmazlar.

Tanım 3.4: Probleme ait özellikler ve tanımlar (Birden fazla sınıfa ait veri kümeleri için)

Problemin ikinci halinde GDS çizge veri kümesi K adet farklı sınıfa ait çizgelerden oluşsun.

- k-ıncı sınıf C_k olacak şekilde $(1 \leq k \leq K), \forall G \in GDS$ için G çizgesi C_k ıncı sınıflardan sadece birine aittir.
- Bir örüntü adayının C_i inci sınıfa ait destek yüzdesi değeri,

$$\text{destek yüzdesi}_{C_i}(X) = \frac{\sum_{(G \in GDS \& G \in C_i)} f(X, G)}{|GDS_{G \in C_i}|} \quad (3.8)$$

fonksiyonu kullanılarak hesaplanır. ($f(x, G)$ daha önce tanımlanmıştır.)

- Örüntü adayının, K adet sınıf için destek yüzdesi değerleri ayrı olarak hesaplanır, en yüksek destek yüzdesi değerini veren C_k sınıfı örüntü adayının ait olduğu sınıftır.
- *Örüntü adayının* ait olduğu sınıfın C_k olduğunu varsayalım. *Örüntü adayının destek yüzdesi*, $\text{destekyüzdesi}_{C_k}(CS)$ olarak gösterilir. C_k sınıfı için eşik değeri ise *destek eşik yüzdesi* olarak tanımlanır. Eğer

$$\text{destek yüzdesi}_{C_k}(CS) \geq \text{destek eşik yüzdesi}_{C_k} \quad (3.9)$$

eşitliği sağlanıyor ise örüntü adayı CS , *örüntü* olarak tanımlanır ve FS sembolü ile gösterilir.

Tanım 3.5: Birden fazla sınıfa ait veri kümeleri için problemin tanımı

$1 \leq k \leq K$ olmak üzere C_k sınıfına ait destek yüzdesi ($\text{destek yüzdesi}_{C_k}$), *destek eşik yüzdesi* ($\text{destek eşik yüzdesi}_{C_k}$) değerinden büyük ya da eşit olan örüntüleri bulunuz. (Örüntü, bağlantılı bir alt çizge olarak tanımlanmaktadır.)

3.2 Çizgede İzomorfizm, Alt çizge İzomorfizm ve Destek Değeri Kavramı

Tanım 3.6:

Çizgelerin Birebir Eşleşmesi Problemi (Çizge izomorfizm):

$G_M = (V_M, E_M)$ ve $G_D = (V_D, E_D)$ çizgeleri $|V_M| = |V_D|$ olacak şekilde iki çizge olsun. Çizgelerin birebir eşleme problemi, aşağıdaki şartları sağlayan bire bir örten $f: V_D \rightarrow V_M$ fonksiyonunu bulmayı amaçlar.

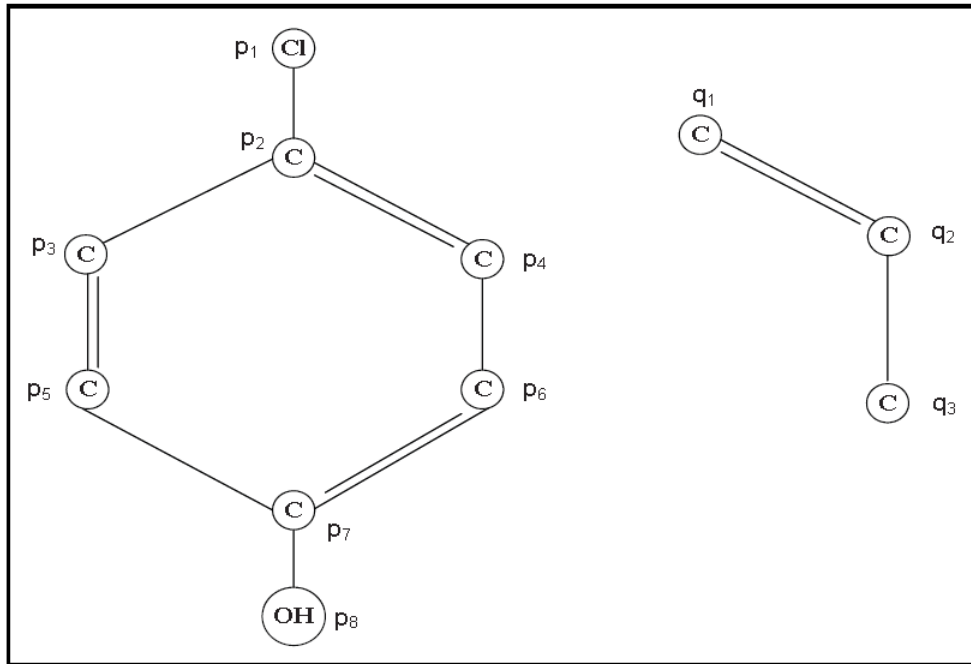
- 1) $\forall u \in V_D$ ise $f(u) \in V_M$
- 2) $\forall (u, v) \in E_D$ ise $(f(u), f(v)) \in E_M$

Eğer böyle bir f fonksiyonu tanımlanabiliyorsa bu probleme *çizge izomorfizm* denir.

Tanım 3.7:

Alt Çizge İzomorfizm: $G=(V, E)$ ve $G_2=(V_2, E_2)$ çizgeleri verilsin ($V_2 \subseteq V$ ve $E_2 \subseteq E$). Eğer $G = (V, E)$ çizgesinin $G_2=(V_2, E_2)$ çizgesi ile izomorfik bir alt çizgesi bulunabiliyor ise bu problem *alt çizge izomorfizm* (veya alt çizge eşleme problemi) olarak adlandırılır.

Alt çizge izomorfizm şekil 3.2 de bir örnek ile açıklanacaktır. Örnekte sırasıyla iki tane moleküler çizge G_p ve G_q görülmektedir. Çizgelerde kenarlar moleküllerdeki tekli ya da çiftli bağları, düğümler ise atomların tiplerini ifade etmektedir. G_q çizgesi G_p nin bir alt alt çizgesi (ör: $\{p_2, p_4, p_6\}$, $\{p_7, p_6, p_4\}, \dots$) ile izomorfiktir. Bu sebeple G_p çizgesi ile G_q çizgesi alt çizge izomorfiktir.



Şekil 3.2, G_p ve G_q çizgeleri için alt çizge izomorfizm örneği.

Tanım 3.8:

Çizgelerden oluşan bir veritabanı DB verilsin ve eşik değeri ise $\sigma(0 < \sigma \leq 1)$ olsun. Eğer bir G çizgesi için $destek\ degeri_G \geq destek\ esik\ degeri$ şartı sağlanıyor ise, G

çizgesi DB de *tekrarlayan bir alt çizgedir* denir. Destek değeri_G, G çizgesinin destek değeridir ve DB deki G alt çizgesini içeren çizgelerin tüm çizgelere oranı olarak tanımlanır.

$$destek\ değeri_G = \frac{|\{G' \in DB \mid G \subseteq G'\}|}{|DB|} \quad (3.10)$$

Bu tanımların ışığında tekrarlayan *alt çizge madenciliği problemi*, DB veri tabanındaki destek eşik değerini sağlayan tüm tekrarlayan alt çizgeleri bulmayı amaçlar.

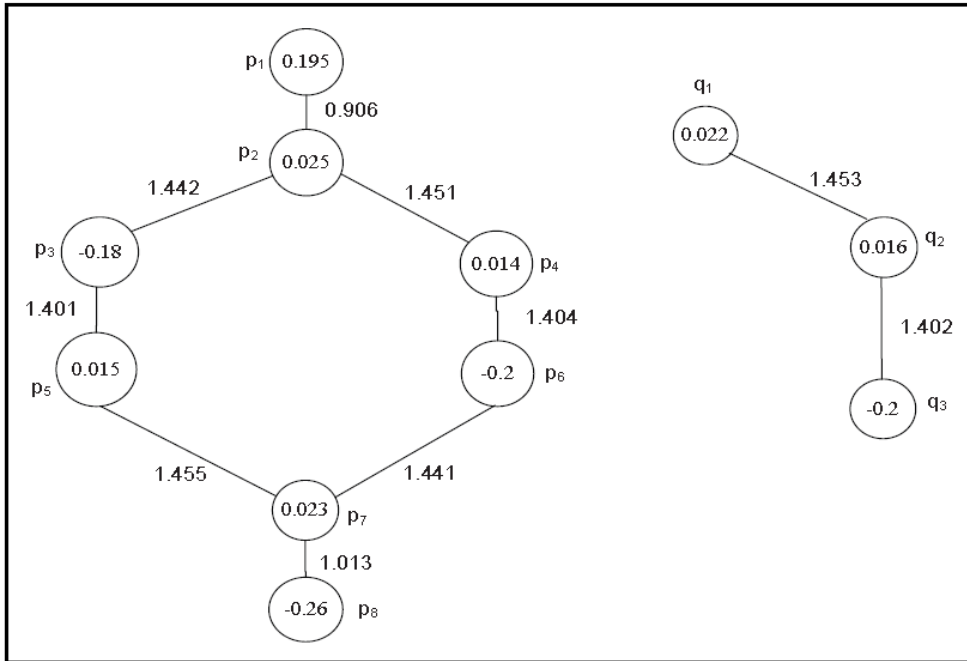
3.3 Tekrarlayan Benzer Alt Çizge Madenciliği

Tekrarlayan Benzer Alt Çizge Madenciliği probleminin birçok gerçek hayat uygulaması vardır. Moleküler veri madenciliği, ilaç dizaynı ve sosyal ağ uygulamaları en çok çalışılmış olanlarıdır. Birçok gerçek hayat veri kümesinde bir örüntü, çizgelerin çoğunda ufak farklar ile tekrarlayabilir. Eğer aradaki farklar önemsiz derecede ufaksa, birbirinden ufak farklarla benzeyen alt çizgeler hala uzmanların ilgisini çekiyor olabilir. Bu konuya görüntü karelerinde nesne eşleme problemi örnek gösterilebilir. Görüntülerden çizge elde etme ve bir nesnenin geçtiği film karelerini bulmak için çizge eşleme uygulamaları literatürde sıkça geçmektedir [Sand and Teller, 2004]. Bu tarz uygulamalarda bakış açısındaki farklılıklar, aydınlatmadaki değişiklikler, bazı nesnelerin bazı kısımlarının görünmemesi gibi sebeplerden aranan nesne görüntüde çoğunlukla ufak farklılıklarla tekrarlanır, bazen ise bir kısmı görülmeyebilir. Fakat tekrarlayan alt çizge madenciliği algoritmaları ufak değişikliklere toleranslı olmadıkları için bu benzer alt çizgeleri bulmak için kullanılamazlar. Şekil 3.2 de bahsedilen moleküler çizge örneğinde kenarlar ve düğümlerin ağırlık değerleri ayrık değerlerdir. Bu örnekte düğüm ve kenarların değerlerinin ayrık değerlerden oluşması sebebiyle tekrarlayan alt çizgeler kolayca görülmektedir.

Fakat uzmanlar ayrık değerlerin dışında düğüm ve kenarların özellik değerlerini de ağırlık değeri olarak kullanabilirler. Bunun yanında özellik değerleri kullanılarak elde

edilen veri kümelerinde düğüm ve kenarların ağırlıkları reel sayılardan oluşmuş özellik değerleri olabilir (devamlı değerler). Böyle veri kümelerinde sadece tamamen aynı alt çizgelere bakarak örüntüleri belirlemek eksik ve yanlış sonuçlar verebilir. Benzer örüntüler araştırmacılara önemli ipuçları verecektir.

Moleküler çizge örneğinde atom ve bağların özellikleri ortam şartlarına bağlı olarak değişebilir (ör: sıcaklık, konum vs.). Bu sebeplerden bir molekülün değişik ortam koşullarındaki özelliklerinden elde edilmiş moleküler çizgeler tamamen aynı olmayabilir. Şekil 3.3 deki örnekte şekil 3.2 deki moleküler çizgenin farklı özellikler için bir örneği görülmektedir (ör: atomların elektronik özellikleri, bağların wiberg indeksi). Şekil 3.3 de G'_p ve G'_q çizgelerinin karşılaştırdığımızda birebir aynı tekrarlayan örüntüyü göremeyiz. Diğer bir yandan dikkatlice G'_p ve G'_q çizgeleri incelenirse G'_p çizgesinin bazı alt çizgelerinin G'_q çizgesi ile tamamen aynı olmadığı ama çok benzer olduğu görülecektir (Ör: $\{p'_7, p'_5, p'_3\} \approx \{q'_1, q'_2, q'_3\}$).



Şekil 3.3, G_p ve G_q çizgeleri için alt çizge benzerlik örneği.

Bu bölümde tekrarlayan benzer alt çizge madenciliği probleminin gerçek hayat uygulamalarında ve birçok uygulama alanındaki tekrarlayan tamamen aynı alt çizgeleri bulma kadar önemli olduğundan bahsedildi. Bahsedilen sebeplere dayanarak tekrarlayan alt çizge madenciliği problemi genişletilip yeni bir problem tanımlanacaktır. Bu amaçla Tanım 3.9 te iki çizgenin benzerliğinden bahsedilecektir.

Tanım 3.9: Hata toleranssız benzerlik

Birbiri ile *izomorfik* $G=(V, E, w_e, w_v)$ ve $G'=(V', E', w'_e, w'_v)$ çizgeleri aşağıdaki şartları sağlıyor ise G çizgesi ve G' çizgesi ile *benzerdir* denir.

$$\begin{aligned} 1. \quad \forall u \in V : f_{sim_v}(w_v(u), w'_v(f(u))) &\geq \alpha_v, \\ 2. \quad \forall (u, v) \in E : f_{sim_e}(w_e(u, v), w'_e(f(u), f(v))) &\geq \alpha_e, \end{aligned} \tag{3.11}$$

f_{sim_v} ve f_{sim_e} fonksiyonları sırasıyla düğümlerin ve kenarların ağırlık değerlerinin benzerliğini hesaplar. Ağırlık değerleri veri kümesine bağlı olarak reel sayı, doğal sayı ve tam sayı olabilir. Tez çalışmasında f_{sim_v} ve f_{sim_e} fonksiyonları aynı fonksiyonlar olarak kullanılmıştır ($f_{sim_v} = f_{sim_e} = f_{sim}$). f_{sim} fonksiyonu aşağıdaki özellikleri sağlar.

1. $f_{sim} : \mathcal{R} \times \mathcal{R} \rightarrow [0,1]$
2. Eğer $x=y$ ise $f_{sim}(x,y)=1$ dir.
3. $f_{sim}(x,y)=0$ ise x ve y hiç benzemiyor olarak tanımlamaktayız.

Benzerlik fonksiyonu olarak bölüm 2.7 deki benzerlik ölçülerinden biri kullanılabilir. α_v ve α_e değerleri ise sırasıyla düğümün ağırlık değerinin ve kenarın ağırlık değerinin benzer olup olmadığını belirten birer eşik değeridir.

Tanım 3.10: Hata toleranslı benzerlik (Bir çizgenin bir alt çizge ile benzerliği)

$G=(V, E, w_e, w_v)$ ve $G'=(V', E', w'_e, w'_v)$ çizgeleri *alt çizge izomorfik* olsunlar ($V' \subseteq V$ ve $E' \subseteq E$). Ancak aşağıdaki şartlar sağlanır ise G çizgesi, G' çizgesi hata toleranslı benzerdir denir.

$$1. \quad \forall u \in V' : f_{sim_v}(w'_v(u), w(f(u))) \geq \alpha_v, \tag{3.12}$$

$$2. \forall (u, v) \in E' : f_{sim_E}(w'(u, v), w(f(u), f(v))) \geq \alpha_e,$$

Yukarıdaki tanımlamalara dayanarak Tanım 3.11 de tekrarlayan benzer alt çizge kavramı tanımlanacaktır. Eğer bir alt çizge G , bir çizge veri kümesindeki anlamlı sayıdaki çizgelerin benzer alt çizgesi ise G alt çizgesi *tekrarlayan benzer bir alt çizgedir* denir. Tekrarlayan benzer alt çizge problemi DB çizge veri kümesindeki tekrarlayan benzer alt çizgeleri α benzerlik parametresine göre bulmayı amaçlar. Eğer $\alpha=0$ ise tekrarlayan benzer alt çizge madenciliği problemi tekrarlayan alt çizge madenciliği problemine dönüşür.

Tanım 3.11:

Bir çizge veritabanı DB verilsin ve eşik değerleri ise $(0 < \sigma, \alpha \leq 1)$ olsun. Eğer bir G çizgesi $destek\ degeri_G^\alpha \geq \sigma$ şartını sağlıyor ise, G çizgesi DB de *tekrarlayan benzer bir örüntüdür* denir. $destek\ degeri_G^\alpha$, G çizgesinin destek değeridir ve α benzerlik eşik değerine göre, G alt çizgesine benzer alt çizgeleri içeren DB veritabanındaki çizgelerin DB deki çizge sayısına oranı olarak hesaplanır. σ ise kullanıcı tarafından seçilen destek eşik değeridir. Örüntü bağlantılı bir alt çizge olarak tanımlanmaktadır.

$$destek\ degeri_G^\alpha = \frac{\left| \left\{ G' \in DB \mid G \subseteq^\alpha G' \right\} \right|}{|DB|} \quad (3.13)$$

3.4 Kullanıcı Desteğinin Örüntü Çıkarma Sistemlerindeki Önemi

Yapısal veri kümelerinde tekrarlayan önemli altyapıların çıkarılması; verinin yorumlanmasını ve özetlenmesini sağlar. Diğer bir yandan uzman olduğu bir alanda çalışan kullanıcı daha önce tanımlanan tipte ya da o alana özgü olan altyapıları arar [Fayyad et al., 1996]. Uzmanlar çoğunlukla belli özellikteki yararlı, o alan için anlamlı alt yapılara öncelik verirler. Uzmanın o alandaki bilgisi bu altyapıların elde edilmesinde kolaylık sağlar. Uzmanın uzmanlık bilgisini işleme katmak; istenmeyen, anlamlı

olmayan alt çizgelerin elde edilmesini engeller. Günümüzde bir çok sistemde kullanıcının veri alanındaki uzmanlığı kullanılmaktadır. FIREFLY¹ isimli müzik tavsiye etmeni (agent) kullanıcının çeşitli şarkılar için fikrini sorduktan sonra, kullanıcının beğeneceği şarkılarını önermektedir. Bu tarz örnekler kullanıcı destekli uzman sistemler alanında sıkça görülmektedir [Fayyad et al., 1996].

Elde edilen örüntülerin uygunluğu değerlendirilirken çeşitli ölçüler esas alınır. Kesinlik, yararlılık gibi kesin ölçülerin yanında anlaşılabilirlik, orijinallik gibi kişiye özgü kavramlar da ölçü olarak kullanılırlar. Bunlara ek olarak diğer bir önemli kavram ise ilginçliktir [Silberschatz and Tuzhilin, 1995]. İlginçlik kavramı doğruluk, orijinallik, yararlılık ve basitlik gibi ölçülerin birleşimi olarak tanımlanmaktadır. Bir örüntü bir uzman için çok ilginçken, örüntüyü inceleyen diğer bir uzman örüntüyü kayda değer bulmayabilir. Bu sebeple bu gibi subjektif görünen ölçüler sadece veri kümesine ya da kurallara bağlı değil aynı zamanda veriyi inceleyen uzmana da bağlıdır. Ancak pek çok uzmanın birlikte kayda değer buldukları örüntülere, kullanıcı desteği ile uzman görüşünün değerli bir katkı sağlayacağı açıktır.

Kullanıcı desteğinin örüntü madenciliği uygulama alanında bir gereksinim olmasının en önemli sebebi ise parametre seçiminin önemidir. Parametrelerin seçimi uygulama alanına yoğunlukla bağlıdır. Moleküler örüntü madenciliği uygulama alanı hataya daha az toleranslı olduğu için *benzerlik eşik değerinin* o veri kümesi için özel olarak hassasiyetle seçilmesi gerekmektedir. Bu derece önemli bir uygulama alanında bu parametrenin veri kümesi üzerinde uzman biri tarafından seçilmesi gerekir. Bir sosyal ağ uygulaması ise hataya daha toleranslı olabilir. Benzerlik eşik değerinin yanında, benzer sebeplerden *destek değeri* de uygulamadan uygulamaya değişecek parametrelerdir.

Bu sebeplerden tıbbi alanlar gibi bazı uygulama alanları için detaylı alan bilgisi ve uzmanlığı gerekmektedir. Bu tarz alanlarda uzmanın bu parametreleri doğru seçebilmesi için görsel veri madenciliği yöntemlerine ve istatistiksel bilgilere ihtiyaç

¹ (www.firefly.com).

duymaktadır. Tezde önerilen yaklaşım bu gereksinimleri karşılamak için kullanıcı desteğini kullanmaktadır. Ayrıca sosyal ağlar gibi uygulama alanlarında çeşitli grafikler ve istatistik bilgiler sayesinde kullanıcının veri kümesi üzerinde uzmanlaşması sağlanmakta ve otomatik sistemlere göre daha şeffaf bir yöntem önerilmektedir.

3.5 Hipotez

Görsel veri madenciliği son yıllarda önem kazanmaktadır. Bunun en önemli sebeplerinden biri görsel grafikler sayesinde otomatik yöntemlerin tam olarak hesaplanamayan parametre değerlerini kullanıcının seçmesine olanak sağlamasıdır. Çok sayıda çizgeden oluşan veri kümelerinde tekrarlayan örüntülerin çıkarılması problemini çözmek için literatürde önerilmiş yöntemler otomatik çalışmaktadırlar. Tezde istatistiksel verileri ve görsel grafikleri katarak uzmanlara ara adımlarda değerli bilgiler vermek ve gerekli yerlerde bu bilgileri kullanarak parametrelerde ince ayar yapılmasına olanak verecek bir yaklaşım önermeyi amaçladık. Bu sebeple görsel veri madenciliği ve kullanıcı desteğini kullanarak desteklediğimiz bir örüntü madenciliği yöntemi önerdik. Bu tezi oluştururken ortaya koyduğumuz hipotezler ise aşağıda verilmektedir.

Hipotez 1) Grafiksel gösterimler, karmaşık veri kümelerinin kullanıldığı çizge veri kümelerinde tekrarlayan alt çizge madenciliği probleminin çözümünde verimli sonuçlar alınabilmektedir.

Hipotez 2) Çizge veri kümelerinin özellik uzayına dönüştürülmesi örüntülere ait özelliklerin çıkarılmasında geleneksel yöntemlere göre daha verimli sonuçlar vermektedir.

4 YÖNTEM

Bu bölümde tezde önerilen yaklaşımdan bahsedilecektir.

4.1 Veri Kümesi

Çizgeleri ifade etmek için kullanılan en yaygın gösterim şekillerinden biri yakınlık matrisleridir. Tezde önerilen yaklaşımda her biri yakınlık matrisleri ile ifade edilen çizgelerden oluşan bir veri kümesi kullanılmıştır. n düğümlü bir $G=(V, E, L_V, L_E, \ell)$ çizgesini tanımlamak için kullanılan veri yapısı aşağıdaki gibi tanımlanmıştır. ℓ fonksiyonu kenar ve düğümlerin ağırlık (etiket) değerlerini verecek şekilde tanımlanmaktadır. Matrisin elemanları M_{ij} aşağıdaki şekilde tanımlanır ($v_i \in V$, $e_{ij}=(v_i, v_j) \in E$, $1 \leq i, j \leq n$ ve $i, j \in Z^+$).

$$M_{i,j} = \begin{cases} \ell(e_{i,j}), & \text{eger } (i \neq j) \& (i < j) \\ \ell(v_i), & \text{eger } i = j \\ 0, & \text{eger } i. \text{ ve } j. \text{ dugum ara sin da kenar yok ise} \end{cases} \quad (4.1)$$

Yönsüz çizgeler ile uğraşıldığından matris simetrik olacaktır, bu sebeple M_{ij} üçgen matris olarak tanımlanmıştır. Köşegendeki elemanlar düğümlerin ağırlıklarını, köşegende bulunmayan elemanlar ise kenarların ağırlıklarını tutmaktadır. Yakınlık matrislerine bir örnek olarak molekülleri tanımlamak için kullanılan veri yapılarından ETMC matrisleri verilebilir [Dimoglo, 1985; Shvets, 1993; Bersuker and Dimoglo, 1991; Shvets and Dimoglo, 1999].

Tezde önerilen yaklaşım, çizge tabanlı veri kümelerini özellik uzayına dönüştürmektedir. Bunun için önce tek bir çizgenin özellik uzayına dönüştürülmesi gösterilecektir. İlk olarak her kenarı üç boyutlu bir vektör ile ifade edilecek şekilde çizge kenar parçalarına bölünür. Her vektör ise kenar ağırlığının ve kenarın her bir ucundaki düğümlerin ağırlıklarından oluşmaktadır ve $[M_{ij}, \text{en küçük}(M_{ii}, M_{jj}), \text{en büyük}(M_{ii}, M_{jj})]$, ya da $[\ell(e_{i,j}), \text{en küçük}(\ell(v_i), \ell(v_j)), \text{en büyük}(\ell(v_i), \ell(v_j))]$ şeklinde gösterilmektedir.

4.2 Çizge Veri Yapısından 3B Özellik Uzayına Dönüşüm

Benzer örüntüleri bulma problemini çözmek için tezde önerilen yaklaşımda çizgeyi üç boyutlu özellik uzayına dönüştürme yoluna gidildi [Yılmaz and Gokturk, 2009a; Yılmaz and Gokturk, 2009b]. Bilindiği kadarıyla bu dönüşüm şu ana kadar önerilen çizge tabanlı tekrarlayan örüntü madenciliği yaklaşımlarında kullanılmamıştır. Bu dönüşüm çizgelerin özelliklerini özellik uzayına aktararak, benzer örüntülerin bulunabilmesi için veri madenciliği ve kümeleme yöntemlerinin uygulanabilmesine olanak sağlamaktadır. Üç boyutlu özellik uzayındaki her boyut kenar ve kenarın iki ucundaki düğümlerin ağırlıklarını göstermektedir. Bölüm 4.1 de bahsedilen vektörlerin her biri özellik uzayında bir nokta ile ifade edilirler. Bu sebeple her nokta çizgenin bir kenarına ait özellikleri içermektedir. Çizgenin kenarları tanım 4.1 te çizgenin en küçük parçası olarak tanımlanmıştır.

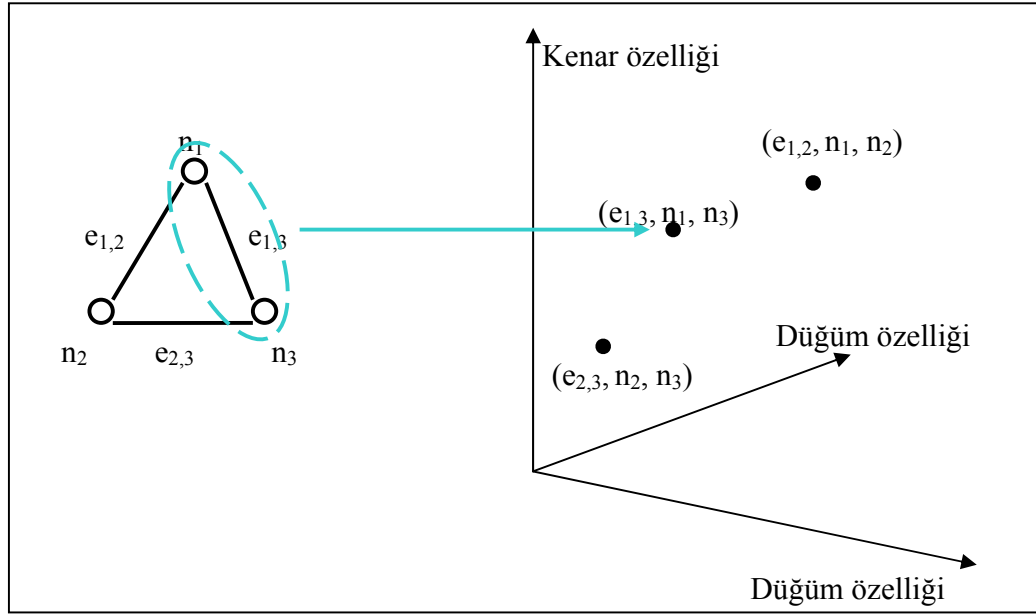
Tanım 4.1:

Bir çizgenin *en küçük parçası* ($Sp_{i,j}$); çizgenin kenarı (yakınlık matrisinin i. satırı ve j. sütunundaki M_{ij} değeri) ile onun iki ucundaki düğümlerini (*en küçük*(M_{ii}, M_{jj}) ve *en büyük*(M_{ii}, M_{jj})) içerir ve $Sp_{i,j}=[M_{ij}, \text{en küçük}(M_{ii}, M_{jj}), \text{en büyük}(M_{ii}, M_{jj}), i, j, C]$ şeklinde gösterilir. Çizgenin kenar sayısı kadar en küçük parçası vardır.

Bir çizgenin her bir en küçük parçası $Sp_{i,j}$ için tek bir nokta ($a_{ij}, \text{en küçük}(a_{ii}, a_{jj}), \text{en büyük}(a_{ii}, a_{jj})$) oluşturulur. C_i sınıfına ait örüntüler bulunmak isteniyorsa, veri kümesindeki C_i sınıfına ait tüm çizgeler özellik uzayına aktarılır. Bunun için C_i sınıfına ait her bir çizgedeki tüm $Sp_{i,j}$ ler özellik uzayına dönüştürülür. Şekil 4.1 de $n_1 > n_2 > n_3$ olacak şekilde düğümleri $N=\{n_1, n_2, n_3\}$ ve kenarları $E=\{e_{1,2}, e_{2,3}, e_{1,3}\}$ olan $G=(N,E)$ çizgesi ve bu çizgeden özellik uzayına dönüşüm görülmektedir.

Tek sınıflı uygulamalarda ise tüm çizgeler aktarılır. Çizgelerin özellik uzayına aktarılması bölüm 2.3.1 de tanımlanan çok sayıda düğüm ve kenardan oluşan tek bir çizgede tekrarlayan örüntüleri bulma problemini çözmeye de olanak sağlar. Bu problemde tek bir çizgedeki tüm kenarlar özellik uzayına aktarılır (Bu problem tek

sınıflı bir problem olduğunu hatırlatalım). Bu problem tez sonrası çalışmalarda incelenecektir.



Şekil 4.1, Çizgeden özellik uzayına dönüşüm.

Çizgelerden özellik uzayına dönüşümün dezavantajı ise geri dönülebilir bir dönüşüm olmamasıdır. Dönüşüm çizgeye ait bilgilerin (Örneğin düğümlerin birbirleri ile olan ilişkileri) kaybedilmesine sebep olmaktadır. Tezde getirilen yaklaşım çizge veri kümelerinden tekrarlayan alt çizge çıkarma yöntemi değildir. Literatürdeki diğer yaklaşımlar gibi alt çizge aramamaktadırlar. Tezde önerilen yaklaşımda tekrarlayan örüntüler ile ilgili özellikleri çıkarmayı amaçlamaktayız. Bu sebeple bu problemi çözmek için bir veri madenciliği yöntemi önerilmiştir. Bölüm 4.3 ün sonunda ve bölüm 4.4 de elde edilen örüntülere ait özelliklerden örüntüleri çıkarmaya yönelik bir yaklaşım önerilmiştir. Çizgelerden özellik uzayına dönüşümün avantajı ise yakın özelliklere sahip kenarları göstermede uygun bir gösterim sağlamasıdır. Birbirine yakın özellikteki kenarlar (en küçük parçalar) küme grupları oluşturur. Bu küme grupları kümeleme algoritmaları ile elde edilirler. Kümeleme algoritmaları birbirine yakın özelliklerdeki örüntü adaylarının parçalarını elde etmeye olanak vermektedir. Her boyut farklı bir özelliği ifade ettiği için kümeleme algoritması uygulanmadan önce veri normalize

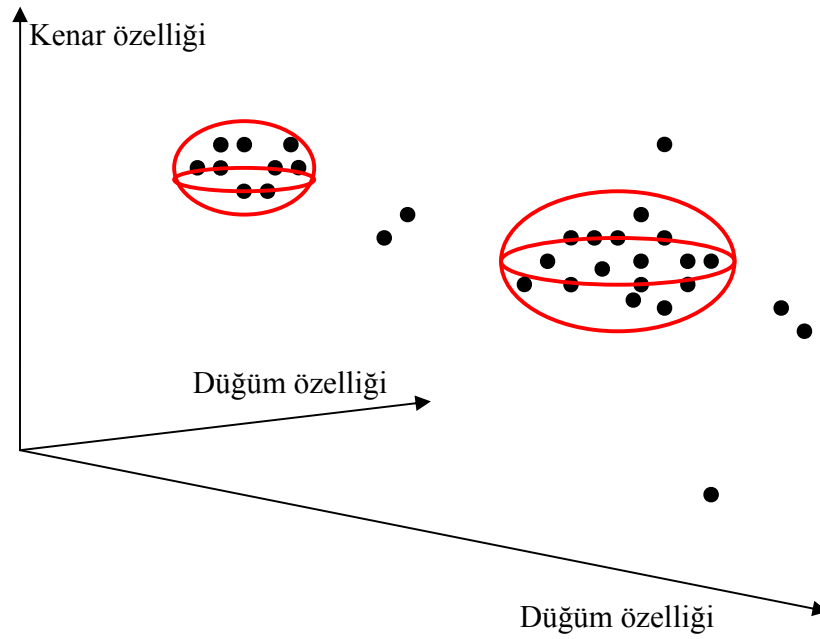
edilmiştir. Normalize etme işleminde Bölüm 4.3 de kullanılan kümeleme yönteminden bahsedilecektir.

4.3 Örüntü Kümelerinin Belirlenmesi

Özellik uzayında C_i sınıfına ait çizgelerin kenarlarını gösteren noktaların bir kısmı nokta grupları oluşturur. Nokta grupları *aday örüntü kümeleri* olarak adlandırılır (Bkz. şekil 4.2). Aday örüntü kümeleri, örüntü adaylarının kenar parçalarını içermektedir. Tezde nokta gruplarını elde etmek için hiyerarşik kümeleme yöntemlerinden, *ortalama bağlantı kümeleme yöntemi (average linkage clustering)* kullanılmıştır [Xu and Wunsch, 2005]. Bu yöntemin seçilmesinin sebebi kümeleri oluştururken diğer bazı yöntemlerden farklı olarak kümeler arasındaki en yakın ya da en uzak noktalara bakmak yerine kümelerin ağırlık noktası dikkate alınarak kümeler oluşturulur (ör. En yakın komşuluk algoritması). Kümeleme yönteminde kümeleri bulmak için veri filtreleneceğinden bu yöntemler uygun olmayacaktır. Bölüm 4.5 ve bölüm 4.6 da filtrelemenin sebepleri ve önerilen yöntemden bahsedilecektir. Ortalama bağlantı kümeleme yönteminde önce veri kümesindeki her nokta ayrı kümeler olarak alınır. Daha sonra küme ağırlık merkezleri arasındaki mesafe belli bir eşik değerinden küçük ise kümeler iteratif olarak birleştirilir. Ortalama bağ kümeleme yönteminde eşik değeri kullanıcı tarafından veri kümesine bağlı olarak seçilmektedir. Eşik değeri büyük tutulursa kümeler daha büyük, eşik değeri daha küçük seçilirse kümeler daha küçük elde edilir. Kullanıcı elde edilen kümelere bakarak kümeleme parametresinin uygunluğuna karar vermektedir. Bu işlem birleştirilecek küme kalmayıncaya kadar sürer. Elde edilen kümelerdeki noktaların benzerliğini belirlemek için benzerlik ölçüsü olarak Öklid uzaklığı alınmıştır. Benzerlik ölçüsü f_{sim} fonksiyonuna ve benzerlik eşik değeri α ya bağlıdır.

Elde edilen küme grupları C_i sınıfına ait (istenen özellikteki) çizgelerin *en küçük parçaları* olan kenarları içermektedir. Örüntüler sadece C_i sınıfına ait örneklerin çoğunda tekrarlanmalı, diğer sınıflarda bulunmamalı ya da çok azında geçmelidir. Bu

sebeple küme gruplarının içine düşen tüm sınıflara ait çizge parçalarının ait olduğu çizgeler belirlenir. Daha sonra her sınıf için ayrı olarak o sınıfa ait çizge sayısının tüm çizgelere oranı yüzde cinsinden hesaplanır. Eğer bir aday örüntü kümesinde C_i sınıfına ait çizge yüzdesi yüksekse ve diğer sınıflara ait çizge yüzdesi düşükse, bu aday örüntü kümesi C_i sınıfı için *örüntü kümesidir*. Diğer durumda örüntü kümesi C_i sınıfına ait değildir denir.



Şekil 4.2, Özellik uzayında görülen örnek kümeler.

Örnek olarak bir küme grubunda; C_1 sınıfına ait çizgelerin %10 u, C_2 sınıfına ait çizgelerin %25 i ve C_3 sınıfına ait çizgelerin %80 i bulunmakta ise bu küme grubu C_3 sınıfına ait bir örüntü kümesi gibi görünmektedir. Bu sebeple iki eşik değeri: *en yüksek eşik değeri* ve *en düşük eşik değeri* kullanıcı tarafından seçilir. Bu değerler kümelerden elde edilen istatistiksel bilgilere göre belirlenmektedir. C_i sınıfına ait örüntüler bulunmak istendiğinde *en düşük eşik yüzdesinden* fazla C_i sınıfına ait çizge içeren ve *en yüksek eşik yüzdesinden* az diğer sınıflara ait çizge içeren örüntü kümeleri adayları *örüntü kümeleri* olarak seçilir.

Daha önce bahsedildiği gibi çizgelerden özellik uzayına dönüşümün geri dönülebilir bir dönüşüm değildir. Bu sebeple aday örüntü kümesine düşen tüm sınıflara ait çizge yüzdelerini bulmak için aşağıdaki şekilde bir yol izlenmiştir. Bunun için bölüm 4.1 deki prosedür uygulanarak tüm sınıflara ait çizgelerden vektörler elde edilir. Daha sonra diğer sınıflara ait çizgeler de özellik uzayına aktarılır. Ve çizgelerin noktalarının aday örüntü kümelerine düşüp düşmediğini dikkate alarak aday örüntü kümelerine düşen çizgelerin her sınıf için ayrı olarak yüzdesi hesaplanır. Diğer sınıflara ait çizgelerin aday örüntü kümelerine düşüp düşmediğini anlamak için aşağıdaki yöntem uygulanmıştır.

Örüntü kümelerinden her biri eşitlik 4.2 deki şekilde tanımlanan elipsoitler ile modellenir. (Elipsoidin merkezi (x_m, y_m, z_m) dir.)

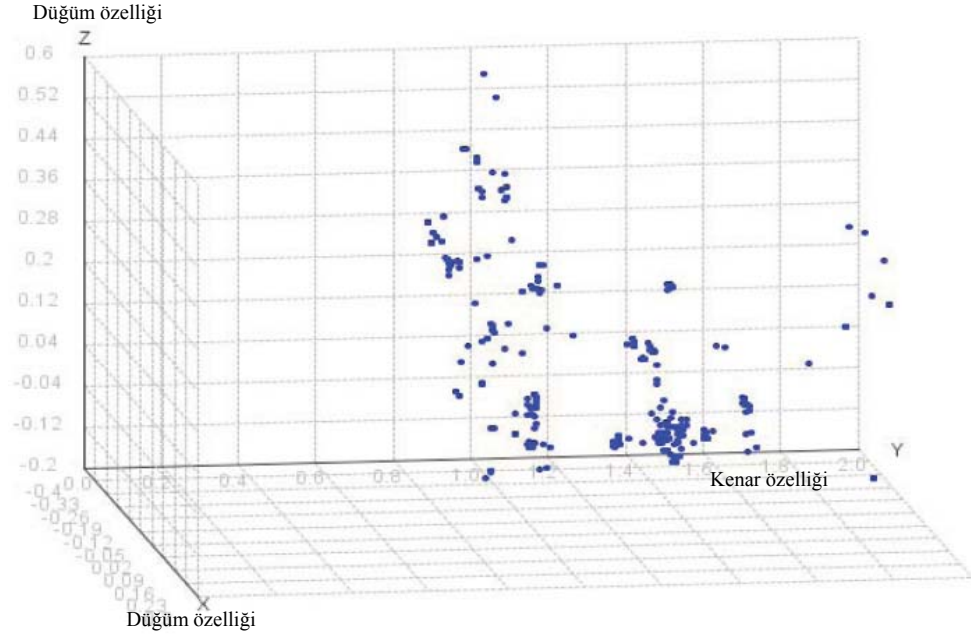
$$\frac{(x - x_m)^2}{a} + \frac{(y - y_m)^2}{b} + \frac{(z - z_m)^2}{c} = 1 \quad (4.2)$$

Elipsoitleri oturtmak için [Todd and Yıldırım, 2007] de bahsedilen elips oturtma yöntemi kullanılmıştır. Bu sayede a, b ve c parametreleri belirlenmiş ve diğer sınıflara ait çizgelerin noktalarının aday örüntü kümelerine düşüp düşmediğini anlamak için elipsoid denklemi kullanılmıştır.

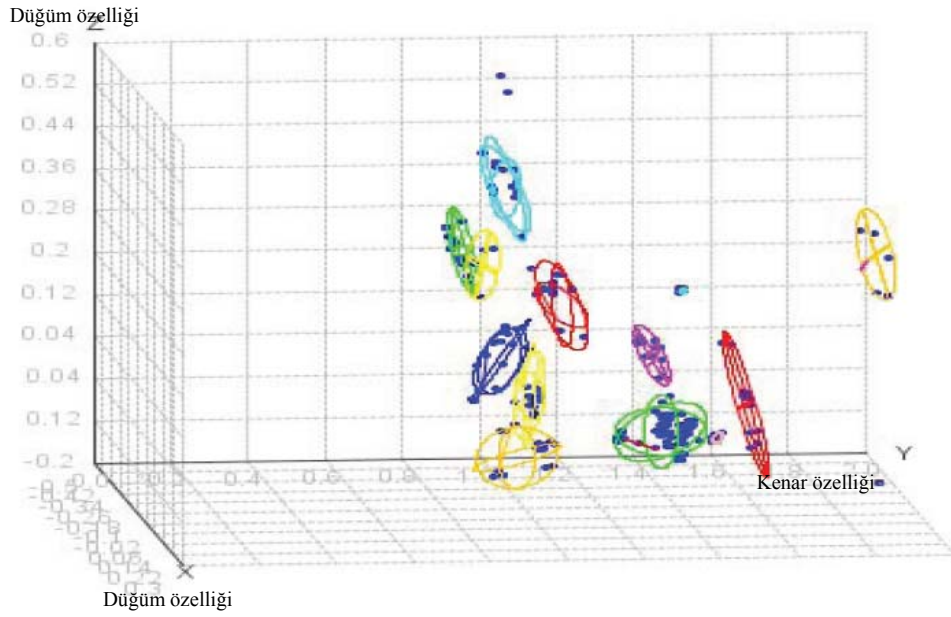
Bir örnekle açıklayalım. Nokta (x,y,z) bir çizgenin kenarlarından birini temsil eden nokta olsun. Noktanın özellik uzayındaki aday örüntü kümesine düşüp düşmediğini anlamak için denklem 4.3 ü sağlayıp sağlamadığına bakılır. Bu şekilde her sınıf için aday örüntü kümelerine düşen çizge yüzdeleri hesaplanır.

$$\frac{(x - x_m)^2}{a} + \frac{(y - y_m)^2}{b} + \frac{(z - z_m)^2}{c} \leq 1 \quad (4.3)$$

Şekil 4.3 de 20 çizgeden oluşan sentetik bir veri kümesinden elde edilen özellik uzayı ve şekil 4.4 te elde edilen örüntü kümeleri görülmektedir. Veri kümesi bölüm 4.5 te bahsedilecek sebeplerden filtrelenmiştir.



Şekil 4.3, Sadece tek bir özelliğe ait filtrenilmiş verinin 3 boyutlu koordinat sisteminde gösterilmesi.



Şekil 4.4, Filtrenilmiş veriden elde edilen aday örüntü kümeleri.

4.4 Örüntü Kümelerinden Örüntülerin Çıkarılması

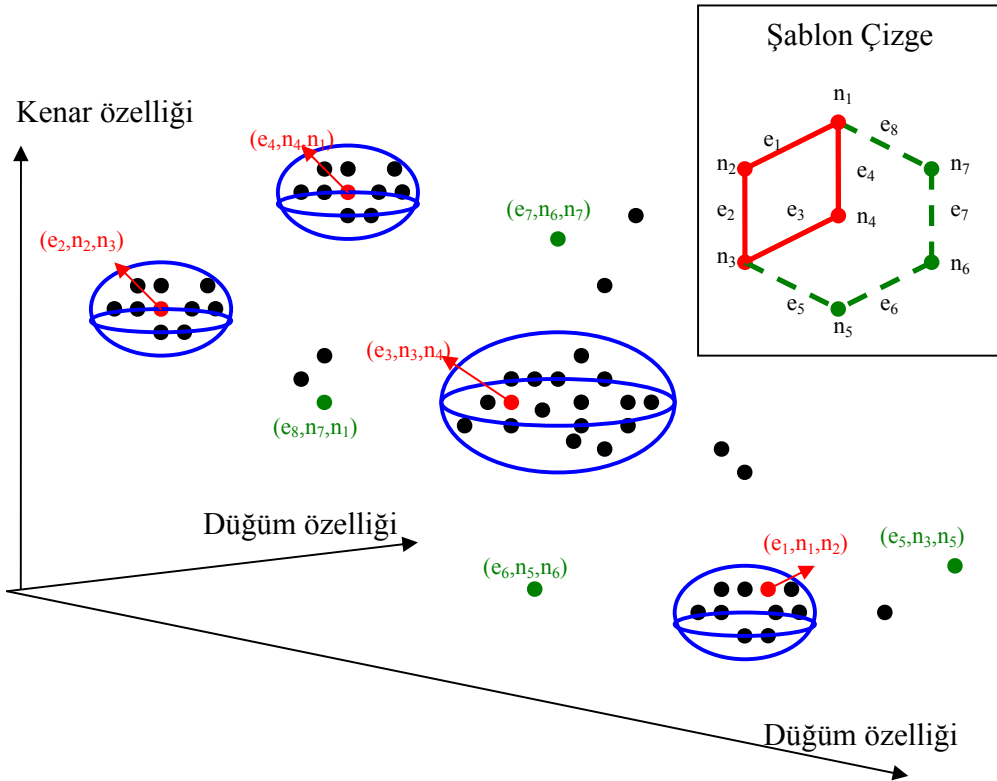
Son olarak elde edilen örüntü kümelerinde örüntülerin çıkarılması gerekmektedir. Çizgelerden özellik uzayına yapılan dönüşüm tek taraflı olduğundan, özellik uzayında elde edilen örüntü kümelerinden bir geri dönüşüm ile örüntü kümeleri çizge olarak elde edilemez. Bunun için veri kümesindeki istenen sınıfa ait çizgelerden biri şablon çizge olarak kullanılır. Kullanıcı, ait olduğu sınıfın özelliklerini en çok üzerinde taşıyan çizgeyi şablon çizge olarak seçer. Şablon çizgenin seçimi sırasında veri kümesindeki çizgelerin özellikleri önemli rol oynar. Dolayısıyla şablon çizgenin seçimi veri kümesi üzerinde uzman kullanıcılar tarafından yapılmalıdır. Şablon çizge kavramı çizge tabanlı alt çizge madenciliği literatüründe [Deshpande et al., 2005] tarafından kullanılmıştır. Örüntüyü çizge olarak elde edebilmek için şablon çizgenin kenarlarını özellik uzayına dönüştürür. Şablon çizgenin en küçük parçalarının (kenarlarının) noktaları elipsoit şekline modellenen örüntü kümelerinin içinde kalıyorsa şablon çizgenin o kenarları örüntünün kenarıdır denir. Şablon çizgedeki kenarlarından bir tanesine ait nokta (x_b, y_b, z_b) şeklinde tanımlanmaktadır ($x_b = e_{ij}$, $y_b = en\ kucuk(v_i, v_j)$, $z_b = en\ büyük(v_i, v_j)$). Eğer bağ örüntü kümesinin içinde ise eşitlik 4.4 ü sağlaması gerekmektedir.

İşlem sonunda birden fazla çakışmayan tekrarlayan örüntüleri elde etmek mümkündür. Şekil 4.5 deki örnekte bir şablon çizge örneği ve özellik uzayı görülmektedir. Kırmızı noktalar özellik uzayına aktarıldığında örüntü kümelerinin içinde kalırken yeşil noktalar ise kümelerin dışında kalmıştır. Şablon çizgedeki kırmızı kenarlar $\{(e_2, n_2, n_3), (e_1, n_1, n_2), (e_4, n_1, n_4), (e_3, n_3, n_4)\}$ örüntünün parçaları, yeşil kesikli kenarlar ise çizgenin parçası olmasına rağmen örüntüye ait değildir. Noktaların örüntü kümelerinin içine düşüp düşmediğini anlamak için eşitlik 4.4 i sağlayıp sağlamadığına bakılır.

4.5 Filtrelemenin Gerekliliği

Özellik uzayında kümeleme yönteminin uygulanması bazı durumlar da başarıyı olumsuz etkileyebilmektedir. Başarıyı etkileyecek faktörler veri kümesinin çok fazla

çizgeden oluşması ya da çizgelerin yoğun (fazla kenar içermesi) olmasıdır. Literatürdeki diğer yöntemler de tekrarlayan alt çizgeleri bulma problemine getirdikleri yaklaşımlarda bu faktörlerden etkilenmektedir. Bu sebeple arama uzayını daraltmak için bölüm 2.2.2 te bahsedilen çeşitli teknikleri uygulayarak yöntemlerini geliştirmişlerdir. Başarıyı etkileyen bu faktörler örüntülerin veri kümesinde gizlenmesine sebep olmaktadır.

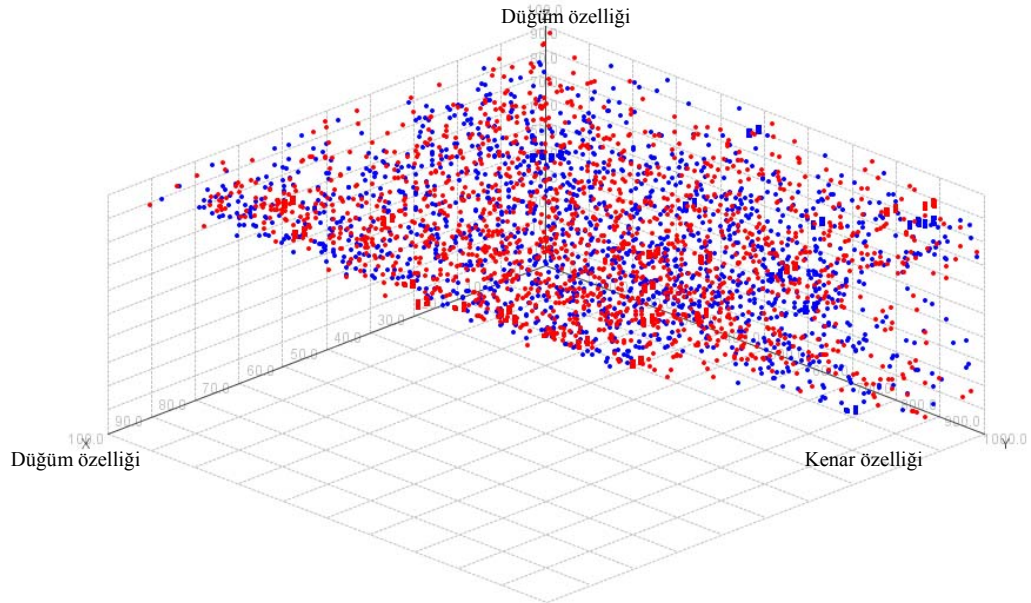


Şekil 4.5, Özellik uzayında görülen örnek kümeler.

Bu tezde örüntülerin çevresinde örüntü gibi görünen ama örüntünün parçası olmayan çizge parçalarını gürültü olarak adlandırılmaktadır. Gürültü olarak adlandırılan bu çizge parçaları, o alanda çizge yüzdesi düşük olan çizgelerin parçalarıdır. Eğer belli bir alanda nokta yoğunluğu fazla olmasına rağmen çizge yüzdesi düşük ise diğer bir deyişle bu noktalar az sayıda çizgeye ait ise o noktaların örüntünün parçası olmazlar. Veri kümesi özellik uzayına aktarıldığında gürültüden dolayı özellik uzayından kümeleme yöntemi ile elde edilen kümeler doğru çıkmayabileceğinden bu probleme tezde bir yaklaşım getirdik. Şekil 4.6 da toplam 100 çizgeden oluşan iki sınıflı bir veri kümesinin özellik uzayına dönüştürülmüş hali görülmektedir.

Şekildeki örnek veri kümesinden kümelerin en doğru şekilde elde edilmesinde karşılaşılan zorluklar aşağıda sıralanmıştır:

1. Veri kümesindeki çizge sayısının fazla olması ya da çizgelerin yoğun olması işlem zamanını oldukça fazla arttırmaktadır. (İşlem zamanındaki artış değerlendirme bölümünde ayrıntılı olarak örneklerle gösterilecektir.)
2. Kümeler insan gözüyle bile çok açık değildir.
3. Kümelerin nerden başladığı ve nerede bittiği belli olmamaktadır.
4. Kümeleme algoritması için uygun parametrelerin ya da veri kümesine uygun modelin belirlenmesi gerekmektedir.



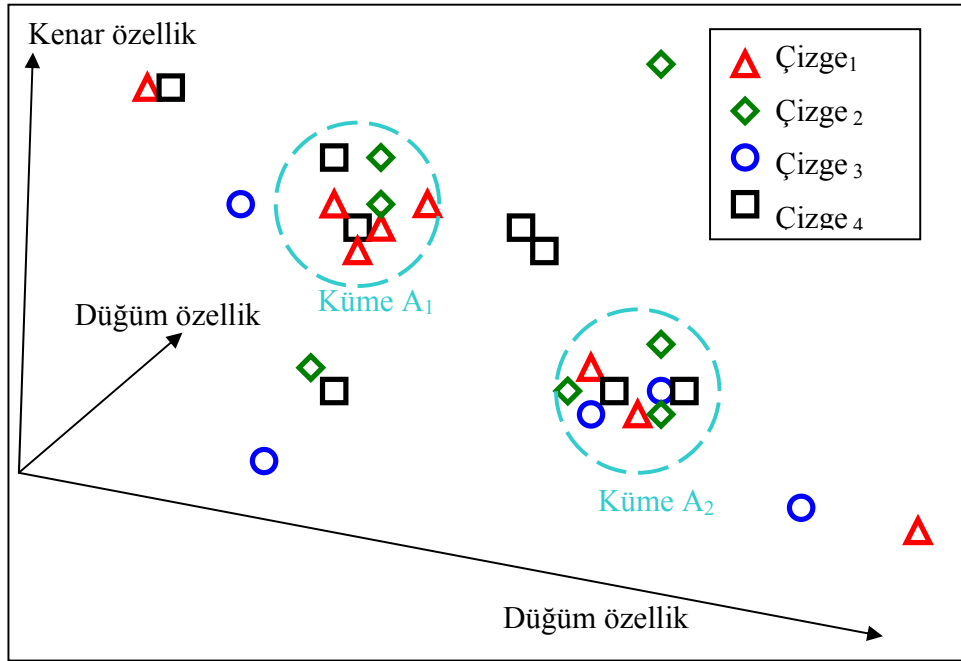
Şekil 4.6, Çizge veri kümesindeki çizgelerin işlenmemiş olarak özellik uzayındaki görüntüsü (Her çizgesi ortalama 40 düğümlü iki sınıfa ait toplam 100 çizgeden oluşan bir veri kümesinden elde edilmiştir).

Üçüncü madde aşağıda örneklerle açıklanacaktır.

Durum 1:

Şekil 4.7 de üç boyutlu özellik uzayında tek bir sınıfa ait dört çizge örneği içeren bir veri kümesi ve elde edilen kümeler görülmektedir. Elde edilen iki küme den Küme

A_1 in destek yüzde değeri %75 dir. Bu küme G_1 , G_2 ve G_4 çizgelerine ait en küçük parçaları içermektedir. Küme A_2 nin ise destek yüzdesi %100 dür. Bu küme ise tüm çizgelere (G_1 , G_2 , G_3 , G_4) ye ait elemanları içermektedir. Veri kümesi fazla gürültülü olmadığı için kümeler birbirinden kolayca ayrılmaktadır.

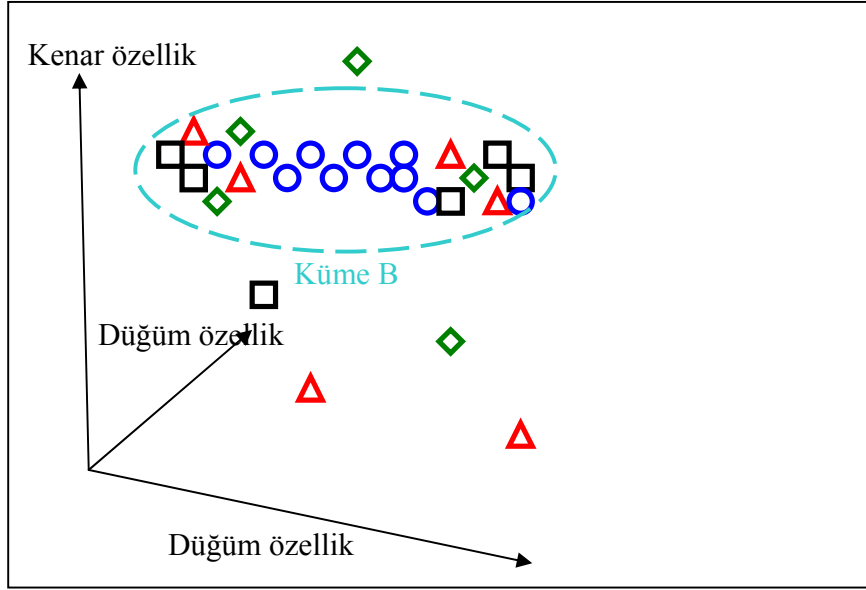


Şekil 4.7, Dört çizgili bir veri kümesi örneği ve elde edilen kümeler (Durum 1).

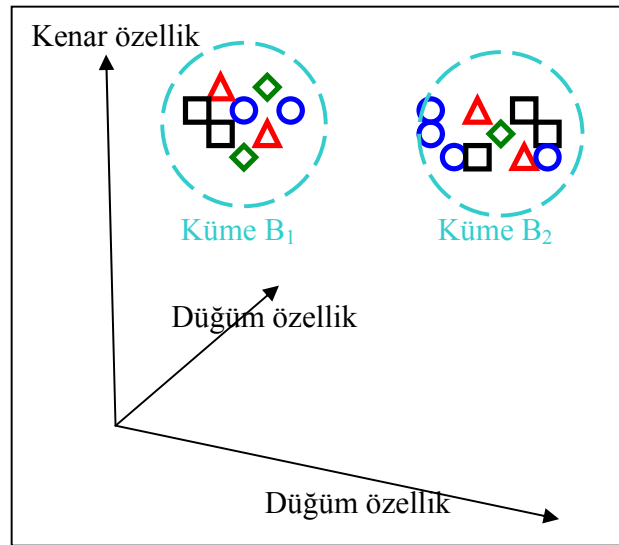
Durum 2:

Şekil 4.8 de verilen veri kümesi ise şekil 4.7 dekinden biraz daha farklıdır. Veri kümesinden elde edilmesi gereken iki kümenin arasındaki gürültü, iki kümenin tek bir küme gibi görünmesine sebep olmaktadır. Bu ise aradaki gürültünün örüntünün birer parçası gibi görünmesine sebep olmaktadır.

Aslında ortadaki mavi dairelerin olduğu kısımda tek bir çizgenin çok sayıda kenarını içermesi sonucu oluşmuştur. Bu alanda diğer çizgelere ait kenarlar yoktur. Ve bu alanda ki çizge yüzdesi %25 tir. Bu sebeple iki küme arasındaki bu alanı gürültü olarak adlandırmaktayız. Şekil 4.9 da, veriden gürültü filtrelendikten sonraki elde edilen kümeler görülmektedir.



Şekil 4.8, Filtrenmemiş veri ve elde edilmiş kümeler.



Şekil 4.9, Filtrenilmiş veri ve elde edilmiş kümeler.

4.6 Görsel Veri Madenciliği Yöntemi

Bölüm 4.5 de bahsedilen sebeplerden, özellik uzayından doğrudan kümelerin çıkarılması bazı veri kümeleri için başarısız sonuçlar verebilir. Bu problemi gidermek için görsel bir veri madenciliği yöntemi önerildi.

Literatürde veri madenciliğinde önerilen yöntemlerin çoğunda ön parametre değerleri yöntem başlamadan önce verilmektedir. Doğru parametre seçimindeki zorluklar, otomatik parametre bulma yöntemlerinin yaklaşık değerler hesaplaması sebebi ile tezde görsel bir filtreleme yöntemi önerildi. Bu sayede, kullanıcının, otomatik olarak elde edilen grafikleri ve istatistiksel bilgileri incelenerek parametreleri seçmesi ya da parametrelerde ince ayar yapmasına imkân verildi. Parametreler belirlendikten sonra önerilen yaklaşım otomatik olarak veriyi filtrelemektedir. Önerilen yaklaşım, özellik uzayındaki gürültülü veriyi filtreleyerek örüntü kümelerinin belirginleşmesini sağlamayı amaçlamaktadır.

Bunun için özellik uzayına aktarılmadan önce veri, görsel veri madenciliği yöntemleri ile filtrelenip ondan sonra özellik uzayına dönüştürülmüştür. Diğer bir deyişle filtreleme işlemi bölüm 4.2 de bahsedilen özellik uzayına dönüştürmeden önce yapılmaktadır. İki boyutta verinin görüntülenmesi için yakınlık matrisinden çıkarılan her bir vektör $[e_{ij}, \text{en küçük}(v_i, v_j), \text{en büyük}(v_i, v_j)]$ den $(e_{ij}, \text{en küçük}(v_i, v_j))$ ve $(e_{ij}, \text{en büyük}(v_i, v_j))$ nokta çifti elde edilir. İlk nokta, $(a_{ij}, \text{en küçük}(a_{ii}, a_{jj}))$, kenarın ağırlığı ve iki ucundaki düğümlerin en küçük olanının ağırlığını ifade etmektedir. İkinci nokta, $(a_{ij}, \text{en büyük}(a_{ii}, a_{jj}))$, ise kenarın ağırlığı ve iki ucundaki düğümlerin en büyük olanının ağırlığıdır.

Önerilen görüntüleme yöntemlerinden gri ölçek gölgeleme yönteminin ayrıntıları 4.6.2 de bahsedilecektir, önerilen diğer görsel veri madenciliği yöntemi HSV ölçek renklendirme yöntemi 4.6.3 de ayrıntılı olarak anlatılmıştır. *Kesişen yatay ve dikey doğrular yöntemi* bu iki yöntemin temelini oluşturduğundan bölüm 4.6.1 de bahsedilecektir.

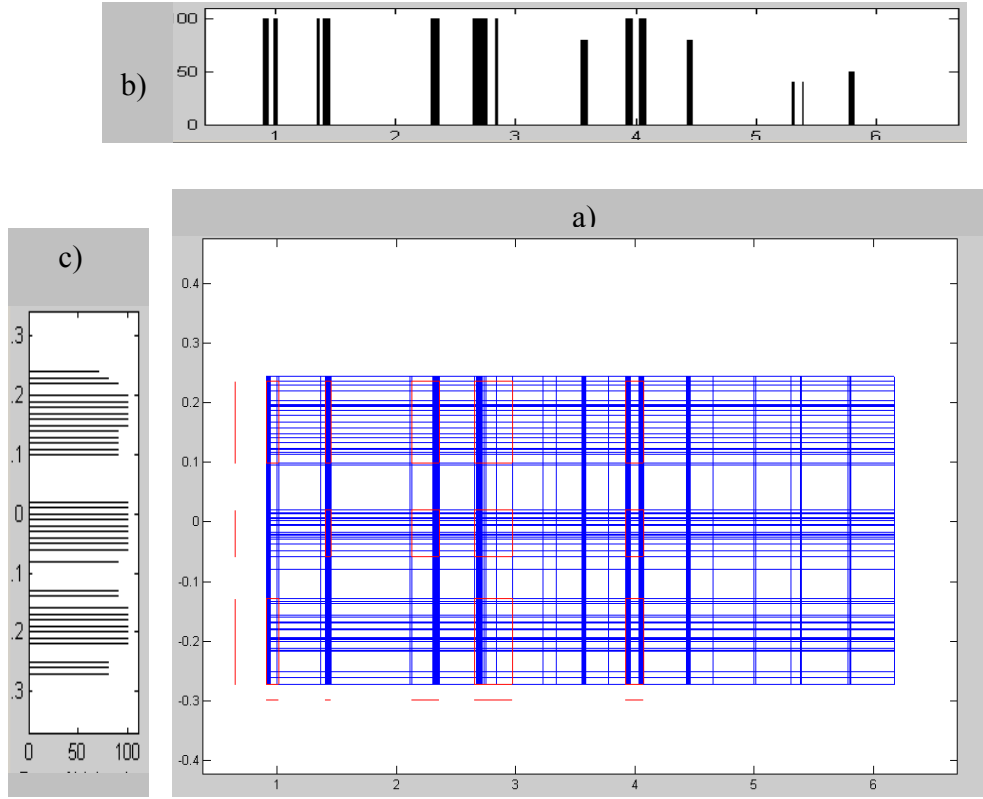
4.6.1 Kesişen Yatay Ve Dikey Doğrular Yöntemi

Kesişen yatay ve dikey doğrular yöntemi tezde önerilen iki görsel veri madenciliği yönteminin temelini oluşturmaktadır. Çizgelerin kenarlarından iki boyutlu özellik uzayına dönüşüm ile elde edilen $(e_{ij}, \text{en küçük } (v_i, v_j))$ ve $(e_{ij}, \text{en büyük } (v_i, v_j))$ nokta çiftleri iki boyutlu özellik uzayına aktarılır. Bu uzayda X ekseninde çizgelerin kenarlarının ağırlıkları $(e_{ij}, i \neq j)$, Y ekseninde ise düğümlerinin ağırlıkları bulunur. Çizgenin her kenarı için kenarın özelliklerini ifade eden nokta çiftlerinin X ve Y eksenlerine izdüşümleri bulunur. Bu şekilde veri kümesindeki tüm çizgeler iki boyutlu özellik uzayına aktarılır.

$(e_{ij}, \text{en küçük } (v_i, v_j))$ ve $(e_{ij}, \text{en büyük } (v_i, v_j))$ noktalarının X ve Y eksenleri üzerindeki izdüşümleri üzerinde uygun bir 1 boyutlu kümeleme yöntemi ile örüntü kümeleri bulunur. Diğer bir deyişle iki boyutlu uzayda kümeleri doğrudan bulmak yerine önce her boyutta bir boyutlu kümeleme uygulanır. Daha sonra X ve Y ekseninde bulunan kümelerin kesişimi olan yerlerdeki kümelere düşen çizge yüzdesi analiz edilir. Bu yöntemde kümeleri bulma bir boyutta yapıldığı için işlem zamanını büyük ölçüde düşürmektedir.

Şekil 4.10 da iki sınıflı moleküllere (örneğin bir hastalık için aktif ve aktif olmayan moleküller) ait özelliklerden oluşmuş çizge tabanlı bir veri kümesinin iki boyutlu özellik uzayına dönüşümü görülmektedir. Şekillerde her boyut için molekül yoğunlukları histogramlar ile gösterilmiştir. Bu gösterimde $(e_{ij}, \text{en küçük } (v_i, v_j))$ ve $(e_{ij}, \text{en büyük } (v_i, v_j))$ nokta çiftlerinin, X ve Y ekseninde izdüşümü mavi doğru parçaları ile gösterilmektedir. Şekil 4.10 da özellik uzayında aktif moleküllere ait örüntülerin her eksenindeki izdüşümlerinin düştüğü aralıklar bulunmuş ve kırmızı doğru parçaları ile gösterilmiştir. Bu doğru parçalarının kesişimi olan kırmızı dikdörtgen ile gösterilen alanlar ise örüntülerin düştüğü aralıkları temsil etmektedir. Ayrıca her eksen için

molekül yüzdesini gösteren histogramlar da görülmektedir. Aktif olmayan moleküller için aynı şekilde grafik oluşturulur.



Şekil 4.10, Tüm aktif moleküller için a) kesişen yatay ve dikey doğrular yöntemi gösterimi b) ETMC matrislerinin köşegen olmayan değerler için molekül yüzdeslerini gösteren histogram c) ETMC matrislerinin köşegen değerleri için molekül yüzdeslerini gösteren histogram.

4.6.2 Görsel Veri Madenciliğinde Gri Ölçek Gölgeleme Yöntemi (GSS)

Filtreleme için önerilen görsel veri madenciliğine dayanan yöntem iki boyutlu özellik uzayındaki veri kümesini histogramlar ile gösterme yöntemine dayanmaktadır. Yöntem veriyi renk histogramları ile göstererek karakteristik özelliklerini 0-255 gri renk ölçeğinde görüntülemektedir. İnsan gözünün algılayacağı gri renk tonu kısıtlı olduğu için 0-255 gri renk ölçeği tercih edilmiştir. Bunun için örüntüler hakkında bilgiler veren örüntü haritaları oluşturmaktadır. Örüntü haritalarındaki gri tonu o alana düşen çizgelerin yüzdesini göstermektedir. Eğer bir alan siyaha yakın ise veri kümesindeki

çizgelerin çoğunun o alanda bir parçası vardır demektir. Bu sayede o alanın X ve Y eksenine düşen izdüşüm aralıkları veri kümesi hakkında bilgi vermektedir. Aşağıda önerdiğimiz “gri ölçek gölgeleme yöntemi” anlatılmıştır.

İlk olarak X-Y koordinat eksenini düzgün aralıklara bölünür. Ve bu aralıkların kesişiminden oluşan bir ağ yapısı oluşturulur. Bunun için X eksenini Δ_x uzunluğundaki aralıklara ve Y eksenini ise Δ_y uzunluğundaki aralıklara bölünür. Δ_x ve Δ_y uzunlukları kullanıcı tarafından seçilir. X eksenini Δ_x uzunluğundaki aralıklara bölündüğünden, Δ_x değeri çizgedeki kenar değerlerinin en büyüğü ile çizgedeki kenar değerlerinin en küçüğü arasındaki farktan çok daha küçük olmalıdır. Aynı şekilde Δ_y değeri de düğümlerin en küçük değeri ve en düğümlerin en büyük değeri dikkate alınarak seçilir. Ayrıca Δ_x ve Δ_y değerleri veri kümesindeki çizge sayısına ve çizgelerdeki kenar sayısına bağlıdır. Çok sayıda çizge içeren ya da kenar sayısı fazla olan yoğun veri kümelerinde Δ_x ve Δ_y değerleri küçük seçilirken, daha az sayıda çizge içeren ya da kenar sayısı az olan seyrek veri kümelerinde Δ_x ve Δ_y değerleri büyük seçilir.

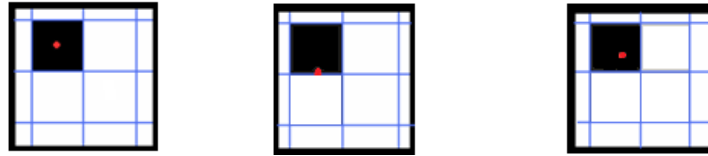
Bu ağ yapısı her çizge için ayrı olarak oluşturulur. Literatürde bölüm 2.4 te düzensiz ağ yapısından bahsedilmiştir. Bu tarz bir yapı işlem zamanını arttıracığından tezde düzensiz bir ağ yapısı tercih edilmedi. Daha sonra her çizge için ayrı olarak, çizgenin yakınlık matrisinden çıkarılan noktaların ağ yapısında düştükleri aralıklar boyanır (Bkz. şekil 4.12, şekil 4.15).

Tonlama işleminde iki teknik uygulanmaktadır [Shvets and Yılmaz, 2005; Yılmaz et al., 2008]. İlk teknikte noktanın düştüğü aralıkların kesiştiği hücre siyaha boyanır (Bkz. şekil 4.11). Bu teknik verinin çözünürlüğünü düşürdüğü için veri kaybına sebep olmaktadır. Bu eksikliği gidermek için ikinci bir tonlama tekniği önerilmiştir. İkinci yöntemde noktanın hücredeki konumuna göre hücrenin tonlaması belirlenir (Bkz. şekil 4.12). Eğer noktanın konumu hücrenin merkezinde ise hücre siyah ve komşu hücreler ise beyaz tonlandırılır. Eğer nokta merkez ile sınır arasında ise interpolasyon tekniği ile noktanın konumuna göre bir ton belirlenir ve noktanın düştüğü hücre ve onun üç komşuluğu tonlandırılır. Şekil 4.13 de dört komşu hücre görülmektedir. Bu hücrelerin

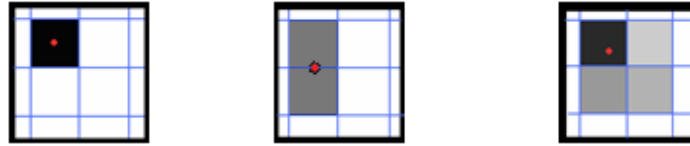
merkez koordinatları $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$ dir. Eğer nokta (x_1, y_1) de olsaydı, o hücre siyah (0) tonlandırılır ve diğer üç komşu hücre beyaz (255) tonlandırılır. Eğer (x_a, y_a) noktasının düştüğü hücrenin tonu hesaplanmak istenirse, bulunduğu hücrenin tonu $T_{(x_a, y_a)}$ interpolasyon ile hesaplanır (Eşitlik 4.4). Komşu hücreler ise aynı şekilde interpolasyon ile hesaplanır. İkinci teknik sayesinde ilk yöntemde düşürülen çözünürlüğün yarattığı veri kaybını azaltmış olmaktadır. Bahsedilen ikinci teknik keskinliği yumuşatma yöntemine (antialiasing) dayanmaktadır. [Baker, 2003]

$$\frac{(x_1 - x_a)}{(x_1 - x_3)} = \frac{0 - T_{x_a}}{0 - 255} \quad \frac{(y_1 - y_a)}{(y_1 - y_3)} = \frac{0 - T_{y_b}}{0 - 255} \quad (4.4)$$

$$T_{(x_a, y_a)} = T_{x_a}^2 + T_{y_b}^2$$



Şekil 4.11, İlk tekniği kullanarak çizge resimlerinin oluşturulması.

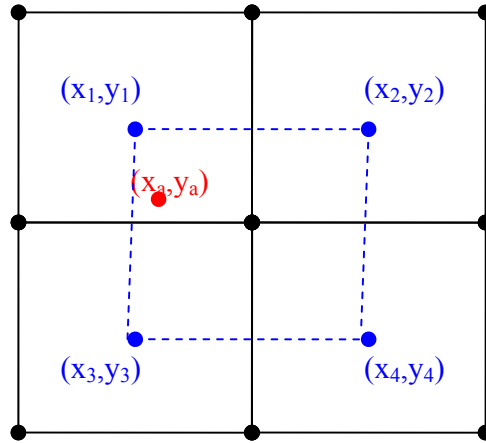


Şekil 4.12, İkinci tekniği kullanarak resimlerin oluşturulması.

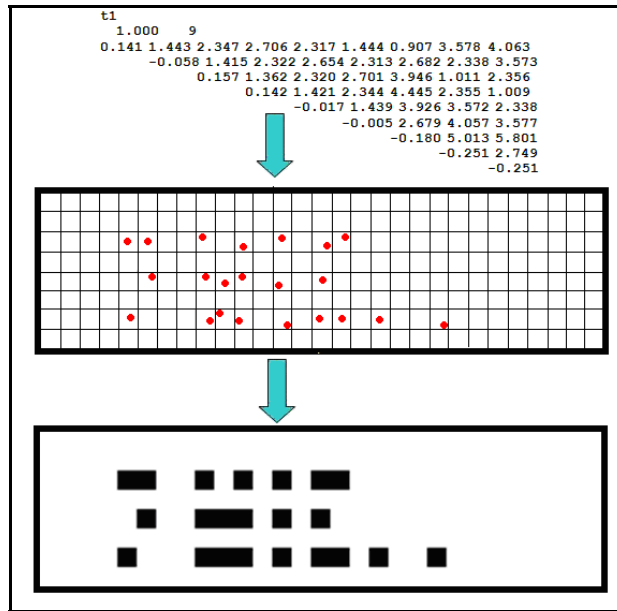
Örüntü yoğunluk haritalarında da bölüm 4.6.2 de bahsedilen ağ yapısı kullanılır. T^k , yukarıda bahsettiğimiz belli bir sınıfın k . çizgesinin tonlarından oluşan resim olsun. Örüntü yoğunluk haritasındaki hücrelerin renkleri $R_{i,j}$ ($i=X$ eksenindeki aralık numarası, $j=Y$ eksenindeki aralık numarası) ise farklı olarak hücelere düşen örnek yüzdesini $R_{i,j}$

$$=100. \frac{\sum_{k=1}^{Toplam\ çizge\ sayisi} T_{i,j}^k}{Toplam\ çizge\ sayisi} \quad \text{şeklinde hesaplayarak gösterir. Bu yöntemin temeli}$$

histogram mantığına dayanır. Örnek yüzdesini gösterirken %100 olan kısımlar için siyah, yüzde sıfır olan kısımlar için beyaz 8 bit gri sakala kullanılır (Bkz. şekil 4.15).



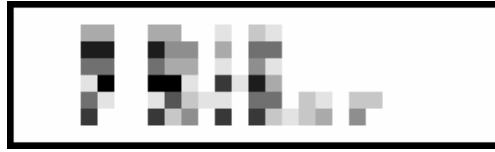
Şekil 4.13, Dört komşu hücre.



Şekil 4.14, İlk tekniği kullanarak yakınlık matrisinden iki boyutlu resimlere kadar uygulanan aşamalar.

Yoğunluk haritalarındaki koyu renk kısımlar benzer alt çizgeleri göstermektedir. Açık renkli kısımlara düşen çizge parçaları ise gürültüyü gösterir. Daha önce bahsedildiği gibi önemsiz ve sık tekrarlanmayan noktalar gürültü olarak nitelendirilmektedir. Gürültülü kısımlar filtrelenerek, tekrar eden alt parçacıkların bulunması işleminde kullanılacak kümeleme yöntemi için uygun bir veri kümesi oluşturulur. Bu sayede önerilen görsel veri madenciliği yöntemleri ve filtreleme işlemleri; birbirine geçmiş, sınırlarının nerede başladığı ve nerede bittiği belli olmayan

gürültülü çizge veri kümelerinde veriyi filtreler ve kümeleme yöntemleri ile kümelerin daha kolayca ayrılabilmesini sağlar. Filtreleme için kullanıcıdan alınan eşik değeri kullanılabilir. Ya da bölüm 7.2 de gelecek çalışmalar için bahsedilecek olan *çekirdek yoğunluk tahmin fonksiyonu* uygulanarak eşik değeri otomatik seçilebilir. Örüntü haritaları oluşturulurken her sınıfa ait çizgeler için ayrı olarak örüntü haritası oluşturulması gerekmektedir. Bu sayede her sınıfa ait örüntüler görülebilir.



Şekil 4.15, Aynı sınıfa ait çizge tabanlı 10 örneğin örüntü yoğunluk haritası.

Literatürdeki çalışmalarda genelde tek bir özellik üzerinde durulmaktadır. Fakat yöntemin başarısı aynı zamanda veri kümesini oluşturmak için kullanılan özelliğe de bağlıdır. Önerilen yoğunluk haritaları yöntemi veri kümesinde kullanılan özelliğin seçiminde uzmanlara fikir vermektedir. Bunun için iki özellik için birer örüntü yoğunluk haritası oluşturulur. Daha sonra her yoğunluk haritasında örüntüler incelenir. Bu şekilde bir özelliğin ortaya çıkarmadığı benzer altyapılar diğer özellikte belirgin hale gelebilir.

4.6.3 HSV Ölçek Renklendirme

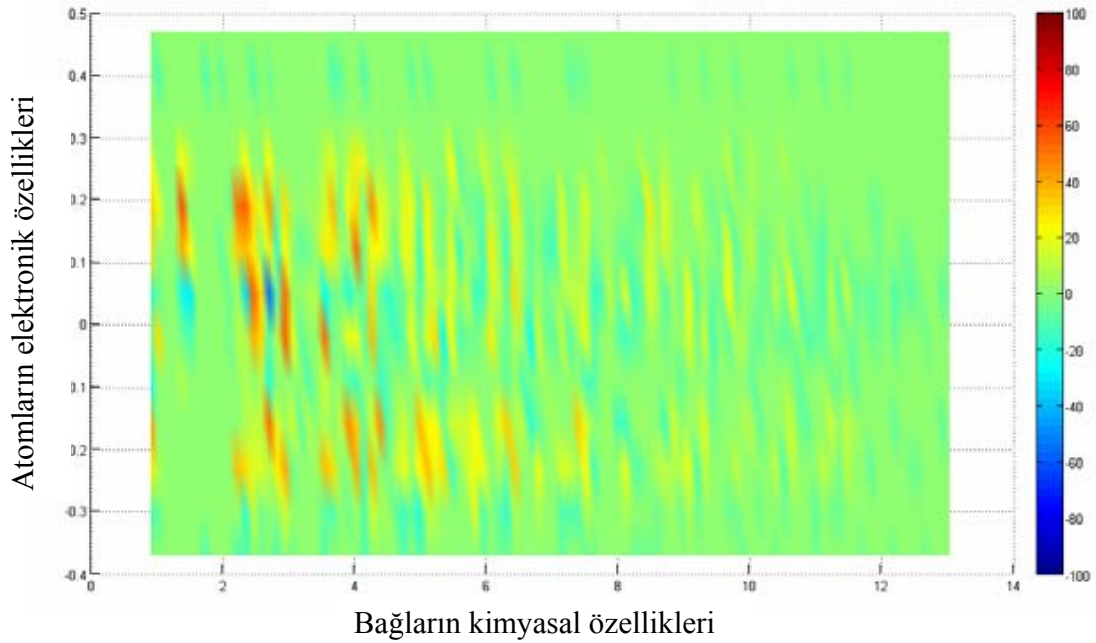
Gri ölçek renklendirme yöntemi ile elde edilen örüntü haritalarındaki çizge yüzdesi düşük kısımları filtreleyerek örüntüler hakkında bilgi sahibi olundu. Filtreleme eşik değeri otomatik olarak ya da kullanıcı tarafından belirlenmektedir. Eşik değerini belirlemek için kullanıcı ekranda gördüğü gri tonlu örüntü haritalarını kullanır. Bu sebeple kullanıcının görüşü ve uzmanlığı önemlidir. Bazı durumlarda iki ton birbirinden çok farklı bile olsa aynı gibi görülebilmektedir. İnsan gri tonlarını renklere göre daha zor ayırmaktadır. Bu bölümde gri ölçek cinsinden tonlar yerine HUE cinsinden renkleri kullanan, önerdiğimiz görsel veri madenciliği yöntemi olan HSV Ölçek Renklendirme yönteminden (HSVS) bahsedilecektir [Yılmaz et al., 2008, Baker, 2003]. (HSV: Hue Saturation Value)

GSS yönteminde olduğu gibi HSVSV yöntemi histogram tabanlı bir yoğunluk tahmini kullanmaktadır. Bunun için çizgelerin ETMC matrislerinden elde edilen iki boyutlu nokta çiftleri $(a_{ij}, \text{en küçük}(a_{ii}, a_{jj}))$ ve $(a_{ij}, \text{en büyük}(a_{ii}, a_{jj}))$ çıkarılır ve bu noktalar iki boyutlu ağ yapısına düşürülür. Ağ yapısındaki aralık mesafeleri Δ_x ve Δ_y olsun. Örüntü haritasındaki ağ yapısındaki dikdörtgenleri $R_{I,J}$ ile gösterilsin ($0 < I \leq N$ ve $0 < J \leq M$). Örüntü haritasındaki her $R_{I,J}$ dikdörtgeninin rengini belirlemek için dikdörtgenin merkezinin rengi belirlenir. Bu amaçla basit bir oylama algoritması kullanılır. Bu algoritmada aynı sınıfa ait her çizge eğer herhangi bir parçası $R_{I,J}$ ye düşüyorsa $R_{I,J}$ ye oy verir. $R_{I,J}$ deki çizge yüzdesi $P_{I,J}$ olarak tanımlansın. $P_{I,J}$ yüzde değeri, $R_{I,J}$ dikdörtgeninin rengini belirlemekte kullanılacaktır. Dikdörtgenlerin merkezlerinin rengi belirlendikten sonra bu renkler iki boyutlu HUE dağılımı kullanılarak örüntü haritasındaki renk dağılımının belirlenmesinde kullanılır. Bir örnekle açıklayalım. $R_{I,J}$ dikdörtgenindeki rengi oluştururken etrafındaki 8 dikdörtgenin merkezlerindeki rengi ve $R_{I,J}$ nin rengi arasında interpolasyon ile renk dağılımını bulunur. HUE değerleri 8 bit olduğundan 0 ile 255 arasında değerler alırlar. Bu şekilde her sınıfa ait çizgeler için ayrı örüntü haritası oluşturulur. Aynı şekilde diğer sınıflar için de örüntü haritaları oluşturulur. Şekil 4.16 de moleküllerden oluşmuş bir veri kümesinden elde edilen örüntüler görülmektedir. Grafikte kırmızı renkler örüntüleri göstermektedir.

4.7 Veri Ardışık Düzenegi Yapısı

Şekil 4.17 de K sınıfa ait örnekler içeren veri kümesi için tezde önerilen yaklaşımın veri akış diyagramı görülmektedir. Önerilen yaklaşım sayesinde, kullanıcı grafikler ve istatistiksel bilgileri işlemin çeşitli ara aşamalarında gerektiğinde sisteme önemli bilgiler vermektedir ve bilgi almaktadır. İşlemin başında kullanıcı uzmanlığını kullanarak başlangıç parametrelerini sisteme verir. Bu parametreler kullanılarak sistem otomatik olarak örüntü haritaları olarak adlandırılan K adet histogram tabanlı görsel resimleri oluşturur. Örüntü madenciliği başlamadan önce örüntü haritaları kullanıcıya

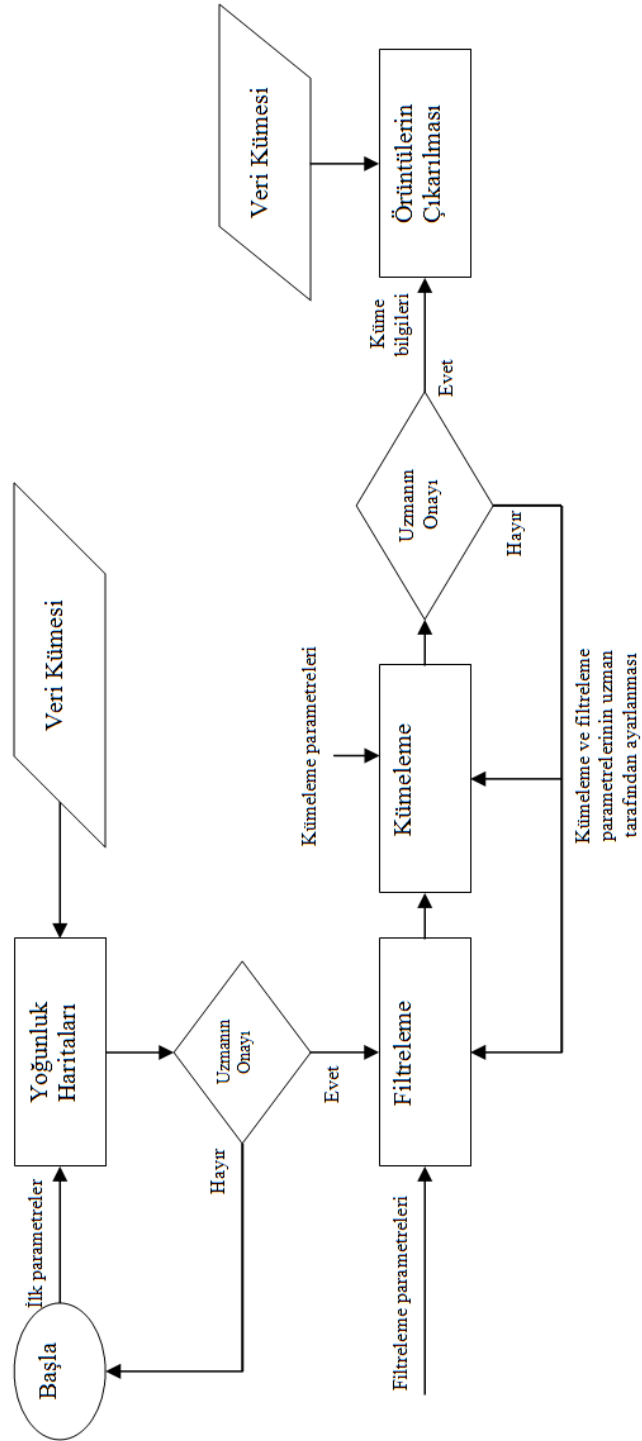
gösterilir. Bu haritalar kullanıcıya farklı sınıflara ait örüntüler hakkında ön bilgi verir ve veri kümesini filtreleme olanağı sağlar. Eğer örüntüler belirgin değilse kullanıcı parametreleri değiştirir. Tez sonrası gelecek çalışmalara fikir vermek amacı ile bölüm 7.2 de filtreleme eşik değerinin otomatik olarak hesaplanması ile ilgili yaptığımız ön çalışmadan bahsedilecektir. Eşik değeri otomatik hesaplanırsa da önerilen yöntem kullanıcının bu değerde ince ayar yapabilmesine olanak sağlar. Eğer elde edilen örüntü haritaları ve gözlemlenen örüntüler kullanıcı tarafından yeterli bulunursa az sayıda tekrar eden (çizge yüzdesi düşük olan alanlardaki kenarlar) önemsiz kenarlar veri kümesinden otomatik olarak filtrelenir. Filtreleme işleminden sonra elde edilen parçalar örüntüler hakkında fikir verir.



Şekil 4.16, HSVSV yöntemi kullanılarak elde edilmiş bir örüntü haritası.

Daha sonra önerilen yöntem filtrelenmiş veriyi üç boyutlu özellik uzayına (*düğüm-kenar-düğüm uzayı*) dönüştürür. Bu uzaydaki her nokta örüntülerin en küçük parçası olan kenarları ve kenarın iki ucundaki düğümlerin özelliklerini temsil eder. Filtrelemeden sonra geri kalan noktalar küme grupları oluşturur. Bu küme grupları sıkça tekrarlayan örüntü adaylarıdır. Filtreleme sayesinde gürültü olarak nitelendirdiğimiz

önemsiz ve sık tekrarlanmayan noktalar geçici olarak silindiği için, örüntüler belirginleşir ve küme grupları uygun bir kümeleme algoritması ile kolayca elde edilir.



Şekil 4.17, Tezde önerilen yöntemin veri ardışık düzeneği yapısı.

İlk aşamada kümeleme parametresi kullanıcı tarafından sisteme verilir. Parametrelerden elde edilen kümeler uygun ise kullanıcı işleme devam eder. Eğer parametreler nokta gruplarını yanlış bir şekilde kümelemiş ise parametreler değiştirilerek uygun bir parametre seçilir. Tez sonrası gelecek çalışmalarda bu parametrenin otomatik hesaplanabilmesi için yöntemler incelenecektir. Bu parametreler kullanılarak, kümeler elde edilir. Ve elde edilen kümelerden elde edilen örüntülere ait istatistiksel bilgiler (kümede bulunan farklı sınıflara ait çizgelerin yüzdeleri) kullanıcıya gösterilir. Kullanıcı, her küme grubunda her sınıf için olması gereken en fazla ya da en az çizge yüzdesini sisteme girer. Bu değer uygulamadan uygulamaya geçişebileceği gibi, veri kümesinden veri kümesine de geçişebilir. Aday kümelerdeki farklı sınıflara ait örneklerin oranı dikkate alınarak nihai örüntüler çıkarılır. Kullanıcının onayı ile elde edilen kümelerden elde edilen örüntü bilgileri ve filtrelenmemiş veri ile örüntü elde etme işlemi başlar. Örüntü çıkarma işleminde filtrelenmemiş veri kullanılarak filtreleme işleminden kaynaklanan veri kaybı engellenmiş olur.

Veri ardışık düzeneği yapısı sayesinde kullanıcının uzmanlığını kullanarak, kullanıcı tarafından seçilen ya da otomatik olarak elde edilen değerlerin doğruluğunu ara adımlarda grafikleri ve istatistiksel verileri inceleyerek kontrol edebilmesine ve gerekli gördüğü değerleri değiştirmesine imkân vermektedir. Eğer ara adımda kullanıcı grafikleri inceleyerek belli bir parametre için hesaplanan değerleri başarılı bulmazsa geri dönüp parametreleri değiştirebilmektedir. Tez sonrası gelecek çalışmalarda parametrelerin otomatik hesaplanması üzerinde çalışılacaktır. Bu durumda kullanıcı otomatik olarak hesaplanan değerlerin bazılarında uzmanlığından emin değilse otomatik olarak hesaplanan ara değerleri doğrudan onaylayarak kullanabilecektir. Bu durumda sistem en başta verilecek birkaç gerekli parametre dışında (ör: Delta değerleri) otomatik olarak da çalıştırılabilecektir. Bu durumda otomatik çalışan diğer yöntemlerin dezavantajlarını kendinde barındırmış olacaktır.

4.8 Hibrit Yöntem

Önerilen yaklaşım bazı durumlarda örüntünün tamamını bulamamakta, bazen ise örüntüyü tamamen içeren daha büyük bir çizge (elde ettiği örüntü fazladan kenarlar içermektedir.) bulmaktadır. Örüntüyü fazla bulmasının sebepleri:

- Kümeleme yönteminde kümeleri birleştirmek için kullanılan eşik değerinin büyük seçilerek kümelerin olması gerekenden fazla büyük olması ve bu sebeple önerdiğimiz yaklaşımın örüntülere ek kenarlar eklenmesi;
- C_i sınıfına ait örüntüleri bulmak için aday örüntü kümelerinde olması gereken C_i sınıfına ait en az çizge sayısı değeri; minimum_eşik_değerinin olması gerekenden düşük seçilmesi ve diğer sınıflara ait olması gereken en fazla çizge sayısı değeri; maksimum_eşik_değerlerinin olması gerekenden büyük seçilmesi. Bu sebeple örüntü kümelerinin içine daha seyrek tekrarlayan kenarların girmesidir.

Bu sebeple tez çalışmasına ek olarak tezde önerilen yaklaşım ile literatürdeki yaklaşımları birleştirerek parametrelerdeki ufak değişikliklere toleranslı olarak örüntüleri bulan bir hibrit yöntem geliştirdi.

Hibrit yöntem, temelde tezde önerilen yaklaşımdan elde ettiği filtrelenmiş ve işlenmiş veriyi literatürdeki diğer bir yaklaşıma (ör. gSpan, Gaston, Mofa, FFSM) verir. Bunun için önce veri kümesine tezde önerilen yaklaşımı uygular ve aday küme gruplarını çıkarır. Daha sonra literatürdeki yaklaşımların benzer örüntüleri bulabilmesi için yeni bir veri kümesi oluşturur. Bunun için küme gruplarının dışındaki kenarlar filtrelenir geriye kalan kenarlardan yeni bir veri kümesi oluşturulur. Benzer örüntüleri bulabilmek için örüntü kümelerine düşen kenarların hepsi o küme grubunun ağırlık merkezi değerini alacak şekilde değiştirilir. Buradaki amaç benzer örüntüler içeren eski veri kümesinden tamamen aynı örüntüler içeren yeni bir veri kümesi oluşturmaktır. Bu işlemin uygulanma sebebi literatürdeki gSpan, Gaston, Mofa ve FFSM gibi yaklaşımların benzer örüntüleri bulamamasıdır. Bu sayede yeni gürültüsüz bir veri kümesi elde edilir. Bu veri kümesi literatürdeki diğer yaklaşımlara verilir ve örüntüler elde edilir. Bu şekilde literatürdeki dört yöntem için ayrı hibrit yöntemler

oluşturulabilmektedir. Hibrit yöntemlerin performansı, önerilen yaklaşımın performansına bağlı olduğu kadar diğer yaklaşımların (ör. gSpan, Gaston, Mofa, FFSM) performansına da bağlıdır. Bu sebeple uygun bir ikinci yaklaşımın seçilmesi gerekmektedir. Hibrit yöntem, önerilen yöntemin kullanımındaki yanlış parametre seçiminden daha az etkilenmektedir.

4.9 Kullanıcının Rolü

Literatürde, tekrarlayan örüntüleri bulma probleminde daha önce önerilmiş yaklaşımların çoğu çizge tabanlı veri madenciliği yöntemleri kullanılmaktadır [Chakrabarti and Faloutsos, 2006]. Bu yöntemler kapalı bir kutu gibi çalışırlar, başta veri kümesi verilir ve sonuç elde edilir. Bu yöntemler birebir aynı örüntüleri aradığından parametreye bağlı değildir. Fakat önerilen yöntem benzer örüntülerin bulunmasına olanak sağladığından benzerliği bulabilmek için çeşitli eşik değerlerine gereksinim duymaktadır. Veri madenciliğinde çok fazla parametreye bağlı yöntemler bazı problemlere sebep olmaktadır [Keogh et al., 2004]. Birçok yöntemin çok sayıda parametreye bağlı olduğu düşünüldüğünde başarılı sonuçlar elde etmek için tüm parametreler için uygun değerlerin belirlenmesi gerekmektedir. Birçok yöntemde mümkün olduğunca parametre kullanımını azaltmaya ve parametrelerin otomatik olarak hesaplanmasına yönelik bir eğilim vardır. Kullanıcının hata yapabileceği bunun da yanlış sonuçlara sebep olabileceği iddiasından yola çıkarak mümkün olduğunca kullanıcıyı devre dışı bırakma ve parametre seçimi dâhil olmak üzere işlem otomatikleştirilmiştir [Lee and Ong, 1996].

Otomatik parametre hesaplayan sistemlerde parametreler yaklaşık olarak hesaplanır. Bazı durumlarda hesaplanan parametrenin olması gerekenden biraz farklı çıkması sonucun beklenenden çok farklı çıkmasına sebep olabilmektedir. Ayrıca özellikle tıp ve biyoloji gibi alanlarında yapılan çalışmalarda konunun uzmanlarının uzmanlığını göz ardı etmek başarıyı düşürecektir.

Bu yöntemlerde uzmanlar işleme ara verip, o anki ara sonuçları ve durumu dikkate alarak bilgilerini çözüme katamamaktadırlar. Bu iki probleme sebep olmaktadır. Bulunan sonuçlar işlemin sonucunda uzmanlar tarafından ayrıntılı olarak incelenmesini gerektirmektedir. Ve erken aşamadaki ortaya çıkan bir yanlışlık (ör: yanlış parametre seçimi ya da ara sonuçların istendiği gibi çıkmaması) ancak işlem bittikten sonra fark edilmektedir. Ara adımlarda düzeltilebilmesine olanak vermemektedir. Çok sayıda parametrenin girdi olarak verildiği yöntemlerde hangi parametrenin sonuca olumlu etkisi olduğu hangilerinin ise olumsuz etkilediğini anlamak zordur. Bu amaçla tezde önerilen görsel veri madenciliği yöntemi ve veri ardışık düzeneği yapısı sayesinde ara adımlarda kullanıcının bilgisini ve uzmanlığını çözüme katmaktayız.

5 DEĞERLENDİRME

Tezde önerilen yaklaşım, gerçek hayat uygulaması ve sentetik veri kümelerinde test edildi. Tüm hesaplama işlemleri Windows XP işletim sisteminde teknik hesaplama programı olarak Java Netbeans 6.5 ile 1.6 Ghz Intel Pentium Core II Duo, 1 Gb Ram özelliklere sahip bir bilgisayarda yapılmıştır.

5.1 Yöntemlerin Gerçek Hayat Veri Kümeleri Kullanılarak Karşılaştırılması

Bu bölümde anti-tüberküloz veri kümesi kullanılarak önerilen yöntem, SUBDUE ve FSG yaklaşımları karşılaştırılıp, elde edilen sonuçlardan bahsedilecektir. Çalışmaları test etmek için moleküler örüntü çıkarma uygulamalarında kullanılan bir gerçek hayat veri kümesi kullanıldı [Elçin et al. 2004]. Veri kümesi (anti-tüberküloz), 33 molekülden (13 aktif ve 20 aktif olmayan) oluşmaktadır.

SUBDUE yaklaşımı, yeni versiyonunda iki sınıfa ait veri kümeleri (aktif ve aktif olmayan moleküller) için uygundur. Fakat FSG yaklaşımı tek sınıfa ait örneklerde başarılı sonuçlar verirken, sınıf bilgisi kullanılmadığı için çok sınıflı örneklerde yetersiz kalmaktadır. Bu sebeple FSG için deneylerde sadece aktif örneklerden elde edilen veri kümesi kullanıldı. Güvenilirliği arttırmak için deneylerde SUBDUE² ve FSG³ yaklaşımlarının orijinal programları kullanıldı. Karşılaştırmalar için Anti-Tüberküloz veri kümelerinin aktif örüntü grupları olan S_T^* kullanıldı. Bu örüntüler kapsamlı kimyasal deneysel analizler kullanılarak elde edilmiştir [Elçin et al. 2004]. Önerilen yöntem (P), SUBDUE ve FSG yaklaşımlarını karşılaştırmak için bu yöntemlerden örüntüler elde edildi. Daha sonra bu örüntülerin her birinden elde edilen S_T^P , S_T^{SUBDUE} , S_T^{FSG} , doğru örüntü S_T^* ile karşılaştırıldı.

² <http://ailab.wsu.edu/subdue>

³ <http://glaros.dtc.umn.edu/gkhome/pafi/overview>

Yöntemlerin karşılaştırılması için iki ölçü *Geri getirim* değeri (“recall”) ve *duyarlılık* değeri (“precision”) ($0 \leq P \leq 1, 0 \leq R \leq 1$) kullanıldı. X yönteminin S_T^* örüntüsünde doğru olarak elde ettiği bağ oranı *geri getirim* veya R olarak tanımlanmaktadır. S_T^X ise seçilen bir X yönteminin ($X=\{P, SUBDUE, FSG\}$) örüntüsünü göstermektedir.

$$R(X) = \frac{\sum_{b \in S_A^*} f(b, S_T^X)}{|S_T^*|} \quad (5.1)$$

Denklemden b bir bağ olmak üzere $f(b, s)$ fonksiyonu aşağıdaki gibi tanımlanmıştır.

$$f(b, S) = \begin{cases} 1, & b \in S \\ 0, & \text{diğer} \end{cases} \quad (5.2)$$

Eğer *geri getirim* değeri yüksekse X yöntemi S_T^* örüntüsünün bağlarının çoğunu doğru olarak bulabiliyor demektir. Sadece *geri getirim* değerini kullanmak yöntemin başarısını ölçmekte tek başına yeterli bir ölçüt olamamaktadır. Çünkü yöntem X, elde ettiği olmaması gereken fazlalık bağları kıyaslamada kullanmamaktadır. Bu sebeple X yönteminin elde ettiği P değeri $S_T^X \equiv S_T^*$ veya $S_T^X \in S_T^*$ durumlarında da aynı olacaktır.

İlk durumda X yönteminden elde edilen örüntü S_T^X , doğru örüntü S_T^* yi elde etmektedir. İkinci durumda ise X yöntemi S_T^* örüntüsünü içermesine rağmen fazladan bağlar da içermektedir. Bu sebeple ikinci bir ölçü olan *duyarlılık* (P değeri) kullanıldı. Aşağıda *duyarlılık* değerinin denklemi tanımlanmıştır. Eğer S_T^X deki tüm bağlar S_T^* de ise, yönteminin duyarlılık değeri yüksektir. Eğer R ve P değerlerinin ikisi de 1 ise, X yöntemi tarafından elde edilen örüntü, S_T^* ye eşittir.

$$P(X) = \frac{\sum_{b \in S_A^X} f(b, S_A^*)}{|S_A^X|} \quad (5.3)$$

Deneylerde elde edilen sonuçlar çizelge 5.1 de sunulmuştur. Çizelgede görüldüğü gibi tezde önerilen yöntem en önemli örüntüyü büyük bir oranda doğru olarak bulmaktadır. Çünkü R ve P değerleri 1 yakın sonuçlar vermiştir. SUBDUE yönteminin

R ve P değerleri 0.8 ve 0.75 dir. FSG yönteminin performansı ise düşük çıkmıştır. FSG nin R ve P değerleri ise 0.40 ve 0.67 dir. FSG ve SUBDUE yöntemlerin başarısının düşük olmasının asıl sebebi devamlı değerler içeren bir veri kümesi kullanılmasından kaynaklanmaktadır (Bkz. bölüm 3.3). FSG nin SUBDUE ya göre beklenen performans düşüşünün sebebi ise FSG yönteminin örüntüleri belirlerken sınıf bilgisini (ör. Aktif ya da aktif olmayan) kullanmamasıdır. FSG aktif molekülleri kullanarak örüntüler elde etmektedir. Fakat elde ettiği örüntüler aktif örüntüler olmayabilir. Çünkü bu örüntüler aktif olmayan örüntülerde de bulunabilmektedir. Eğer bir örüntü hem aktif hem aktif olmayan moleküllerde bulunuyor ise bu örüntü örüntü değildir. Çünkü molekülün aktivitesine ait bir bilgi vermez. SUBDUE ve önerilen yöntem sınıf bilgisini çözümde kullanmaktadır. Bu sebeple elde edilen örüntüler aktif örüntülerde sıkça görülürken, aktif olmayan örüntülerde nadiren görülmektedir. Bu sebeple SUBDUE nün ve önerilen yöntemin performansı daha başarılı olduğu görülmektedir. Bu sonuçlar aktif örüntülerin bulunmasında aktif olmayan moleküllerin önemli rolünü göstermektedir. Buradan aktif örüntülerin çıkarılmasında aktif ve aktif olmayan örüntülerin ikisini de kullanan yöntemler önerilmesi gerektiği sonucu çıkmaktadır.

Çizelge 5.1., Tezde önerilen yöntem, SUBDUE, FSG yöntemlerinin R ve P değerlerine göre performansı.

	Önerilen Yöntem	SUBDUE	FSG
R	0.95	0.80	0.40
P	0.97	0.75	0.67

5.2 Sentetik Veri Kümesi Oluşturulması

Gerçek hayat veri kümelerinin bulunmasındaki zorluk önerilen yöntemin sentetik veri kümesinde test edilmesini gerekli kılmıştır. Bunun için sentetik veri kümesi üretme programları incelenmiştir⁴. Önerilen yöntemi, çeşitli açılardan (ör: çizge sayısı, gürültü vb.) diğer yöntemler ile karşılaştırılmasına olanak sağlayacak sentetik bir çizge veri kümesi oluşturulmuştur. Testlerde kolaylık olması açısından iki sınıflı çizgelerden

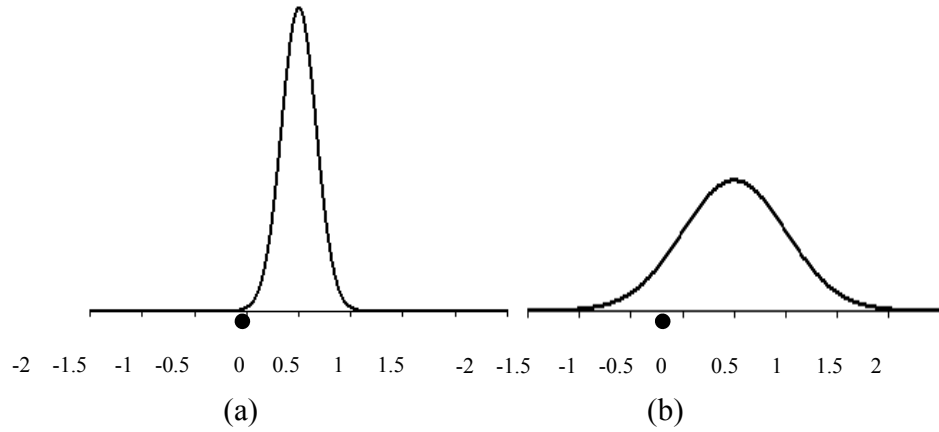
⁴ <http://ailab.wsu.edu/subdue>

oluşan veri kümeleri kullanılmıştır. Fakat önerilen yöntem ikiden fazla sayıda sınıfa ait çizge içeren veri kümelerinde de uygulanabilmektedir. Elde edilen veri kümelerindeki çizgelerde düğüm ve kenarların ağırlık değerleri reel sayılardır. Düğüm ve kenar değerleri sırası ile $[0, \text{Max}_d]$ ve $[0, \text{Max}_k]$ aralıklarında düzgün dağılıma göre rastgele olarak seçilir. Değerler belirlendikten sonra her bir düğüm ve kenara bir tanımlama numarası verilir. Düğümler arasındaki benzerliği hesaplamak için düğümlerin ağırlık değerleri arasındaki Öklid uzaklığı kullanılır. Kenarlar arasındaki benzerlikte aynı şekilde hesaplanır. Diğer bir deyişle iki kenar birbiri ile benzer ise onların ağırlık değerleri tek boyutlu uzayda birbirine yakındır.

İki sınıfa (C_a ve C_i) ait çizgeler içeren veri kümesi oluşturulmadan önce, üç tip örüntü oluşturulur. Bunlar sadece ilk sınıfa (C_a) ait örneklerde bulunan örüntüler (S_a), sadece ikinci sınıfa (C_i) ait çizgelerde bulunan örüntüler (S_i) ve iki sınıfın ikisinde de ortak geçen örüntülerdir (S_{ia}). İlk sınıfa ait çizgelerin her biri S_a ve S_{ai} ye ait alt çizgeleri içerir. Aynı şekilde C_i sınıfına ait çizgeler bu sefer S_i ve S_{ci} örüntülerinden oluşturulur. Böylece iki sınıfa ait çizgelerden oluşan bir veri kümesi oluşturulmuş olur. Örüntüler çizgelere eklenmeden önce gürültü olasılık değeri (gürültü_o) adı verilen ve bir kenar veya düğüme gürültü eklenip eklenmeyeceğini gösteren parametre değeri kullanılarak değiştirilir. Eğer $\text{gürültü}_o = 0$ ise S_a ve S_{ai} örüntüleri veri kümesindeki çizgelere aynen eklenir. Eğer $\text{gürültü}_o > 0$ ise gürültü_o olasılık değerine göre örüntünün kenar ve düğümlerine gürültü eklenir ya da eklenmez. gürültü_o değeri kenar için ayrı, düğüm için ayrı bir değer seçilebilir. Kenar veya düğümlere gürültü düzgün ya da Gauss dağılımları kullanılarak eklenebilir. Düğüm ve kenarlara düzgün dağılıma göre gürültü eklenirken aşağıdaki prosedür izlenir.

Düğüm değeri x olsun, bu durumda gürültü eklenmiş yeni düğüm değeri $[x-D_d, x+D_d]$ aralığında düzgün dağılıma göre rastgele olarak seçilir. Benzer şekilde, kenar değeri y olsun, gürültü eklenmiş yeni kenar değeri $[y-D_k, y+D_k]$ aralığında düzgün dağılıma göre rastgele olarak seçilir.

Gauss dağılımına göre gürültü eklenirken, her bir düğüm veya kenar değerinin üzerine belirli bir standard sapmaya göre bir gauss eğrisi oturtulur. Daha sonra bu gauss eğrisi kullanılarak, gauss olasılık dağılımına göre rastgele bir değer seçilir. Seçilen değer yeni gürültülü düğüm veya kenar değeri olarak kabul edilir. Bu işlemde standard sapma önemli bir rol oynamaktadır. Nitekim şekil 5.1 de gösterildiği gibi gauss olasılık dağılımı standard sapma tarafından belirlenmektedir.



Şekil 5.1, Standard sapmanın (σ) gauss olasılık dağılımına etkisi; (a) $\sigma = 1$, (b) $\sigma = 3$.

Standard sapmayı belirlemek için gürültü_m parametresi kullanılmaktadır. Bu parametre φ olduğunda, düğüm değerlerine gürültü eklerken kullanılan standard sapma $\varphi \times 3 \times D_d$; kenar değerleri için kullanılan standard sapma ise $\varphi \times 3 \times D_k$ olmaktadır. Bu durumda, gürültü_m büyüdükçe gürültü eklenmiş düğüm veya kenar değerlerinin orjinal değerlerden uzaklaşma olasılığı artmaktadır.

5.3 Yöntemin Sentetik Veri Kümelerinde Test Sonuçları

Önerilen yöntem her biri iki sınıfa ait çizgelerden oluşan ve çeşitli oranlarda gürültü eklenmiş sentetik veri kümeleri ile test edildi. Oluşturulan veri kümeleri dört çekirdek değerleri için test edilip sonuçların ortalamasını alarak başarımın farklı rastgele değerler içinde tutarlı olduğu gösterildi. Testlerde düzgün dağılıma sahip bir rastgele sayı üreticisi ile gauss dağılımı kullanılan bir rastgele sayı üreticisi kullanılmıştır. Veri

kümesindeki her çizge ortalama olarak 25 düğüm ve 35 kenardan oluşmaktadır. Testlerde dört farklı büyüklükte veri kümeleri ile çalışıldı. Veri kümelerinin her birinde eşit oranda iki sınıfa ait çizgeler olacak şekilde 20 çizge (10+10), 50 çizge (25+25), 100 çizge (50+50) ve 200 çizge (100+100) ile yaklaşım test edildi. Bu veri kümeleri oluşturulurken $Max_d = 10000$, $Max_k = 100000$, $D_d = 83$, $D_k = 125$ parametre değerleri kullanıldı. Her bir veri kümesinde iki sınıf için örüntü bulunmaktadır. Ayrıca problemi zorlaştırmak için iki sınıfa ait çizgelerde ortak bulunan tekrarlayan alt çizgeler eklendi. Önerilen yöntemin örüntülerden bu alt çizgeleri ayırması gerekmektedir. Örüntüler, ait olduğu sınıfa ait çizgelerde, yapısında değişik oranlarda farklılıklar ile tekrarlandı. Bu değişiklikler gürültü olarak değerlendirilmektedir. Gürültünün eklenmesi sayesinde birebir aynı olmayan ancak benzer örüntüler sentetik olarak oluşturulmaktadır. Gürültü için kullanılan parametrelerin dışında kullanıcıdan alınan çeşitli parametreler ve seçimler önerilen yaklaşımda kullanılmaktadır. Bu parametreler ayrıca sonuçlar ile birlikte belirtilmiştir.

5.3.1 Kullanılan Parametreler

Bu bölümde çeşitli parametre değerleri için elde edilen başarı değerleri gösterilecektir. Bu sayede kullanıcı tarafından seçilen parametre değerinin doğruluğu tartışılacaktır. (Bölüm 5.3.1 de bahsedilen testlerde aksi belirtilmedikçe düzgün dağılım ile gürültü eklenmiştir ve $gürültü_0 = 1$ alınmıştır.)

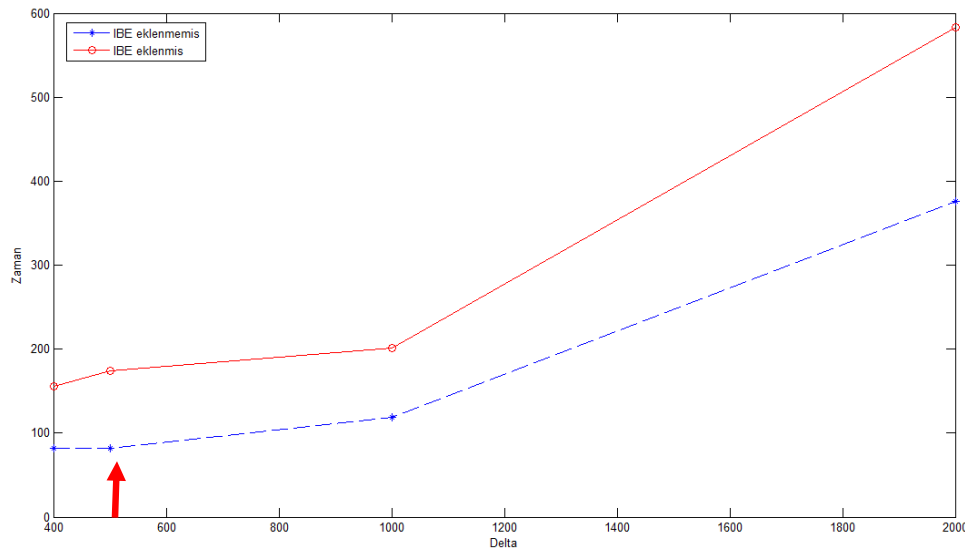
Yoğunluk Haritaları Ve Görsel Veri Madenciliğinde Kullanılan Parametreler:

Delta Değerleri:

Δ_x : Aktivite haritaları oluşturulurken, X eksenini Δ_x uzunluğundaki aralıklara bölünmektedir.

Δ_y : Aktivite haritaları oluşturulurken, Y eksenini Δ_y uzunluğundaki aralıklara bölünmektedir.

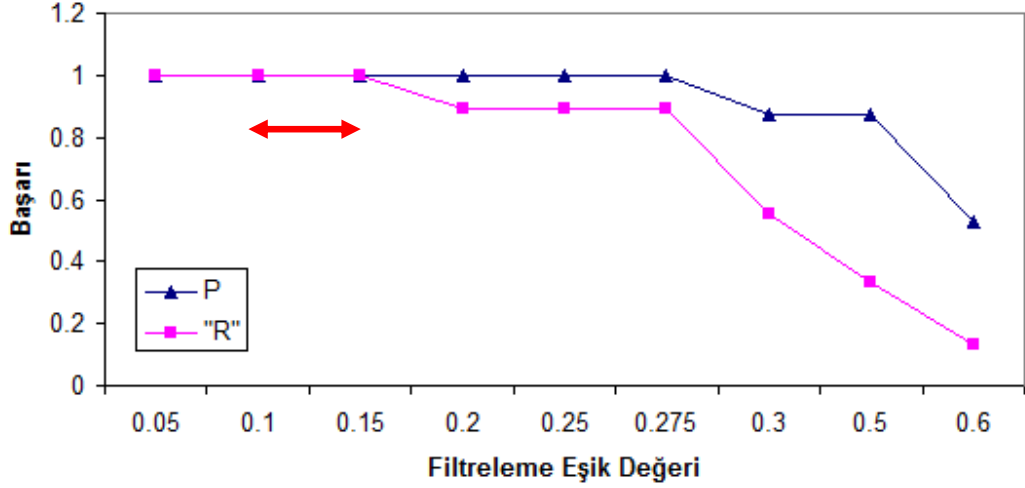
Delta değerinin seçimi filtrelenecek veri miktarını etkilemektedir. Delta değerinin seçiminin doğrudan başarıya etkisi yoktur. Fakat dolaylı yoldan *kümeleme eşik değerini* etkilemektedir. Delta değerinin küçük seçilmesi daha çok verinin filtrelenmesine imkân verdiğinden işlem zamanını büyük oranda azaltmaktadır (Bkz. şekil 5.2). Ayrıca delta değerinin küçük seçilmesi kümeleme eşik değerinin daha büyük seçilmesini gerektirmektedir. Eğer delta değeri büyük seçilirse daha az veri filtrelenecektir Bu ise işlem zamanını büyük ölçüde arttıracaktır. Böyle bir durumda elde edilen filtrelenmiş veri kümesinde çok sayıda sadece gürültüden oluşmuş kümeler oluşacaktır. Bu durumda kümeleme algoritmasının kümeleri bulabilmesi için *kümeleme eşik değerinin* daha küçük bir değer seçilmesi gerekecektir. Bölüm 5 teki testlerin tümünde $\Delta_x=500$ ve $\Delta_y=2000$ olarak seçilmiştir. Bu parametreler seçilirken örüntü haritalarında örüntülerin gözle görünür şekilde ayrılmasına olanak verecek delta değerleri seçilmiştir. Ayrıca bu parametreleri düşük seçilerek örüntü haritalarındaki çözünürlüğü yüksek tutmak amaçlanmıştır.



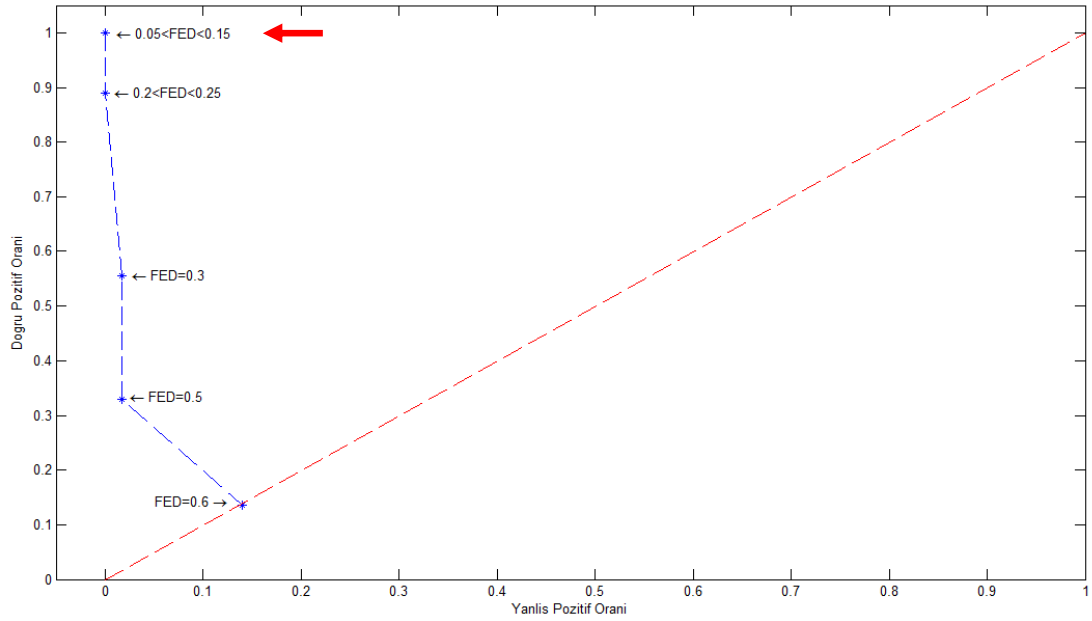
Şekil 5.2, Delta değerlerinin seçiminin zamana (sn.) etkisi ($\Delta_x = \text{Delta}$ ve

$\Delta_y = 4 \times \text{Delta}$ için)

Filtreleme eşik değeri (FED): Yoğunluk haritalarında veri kümesinin filtrelendiği için kullanılan eşik değeridir. Bu değerin çok yüksek seçilmesi daha çok verinin filtrelenmesine sebep olacağından başarıyı düşürecektir (Bkz. şekil 5.3).



(a)



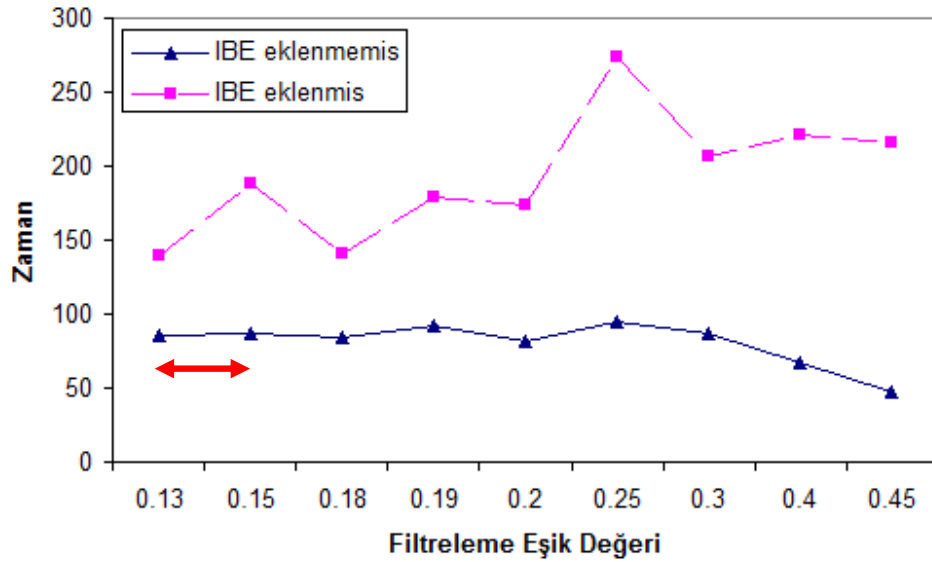
(b)

Şekil 5.3, Çeşitli filtreleme eşik değerlerinin başarıya etkisi. (a): Filtreleme eşik değerinin P ve R değerlerine etkisi. (b): Filtreleme eşik değerinin değişimi için doğru pozitif oranının yanlış pozitif oranına göre değişim grafiği.

Filtreleme eşik değerinin çok düşük seçilmesi ise daha az verinin filtrelenmesine sebep olacaktır. Bu durum başarıyı etkilememektedir. Fakat kümeleme algoritması için veri miktarı artacağından filtreleme eşik değerinin düşük seçilmesi işlem zamanının artmasına sebep olmaktadır (Bkz. şekil 5.4). Testlere filtreleme eşik değeri 0.1 ile 0.15 arasında seçilmiştir.

Sınıf sayısı: Veri kümesindeki sınıf sayısıdır. Testlerde iki sınıflı veri kümeleri kullanılmıştır.

Görsel veri madenciliği yönteminin seçimi: Görsel veri madenciliği için tezde üç yöntem önerilmiştir. Bunlar GSS yönteminde iki gölgelendirme yöntemi ve HSV ölçek yöntemidir. Testlerde görüldüğü üzere bu yöntemlerin seçimi başarıyı etkilememektedir. Ama yöntemlerin seçimi *filtreleme eşik değeri* parametresinin seçimini etkileyeceği için kullanıcının seçimine bırakılmıştır.



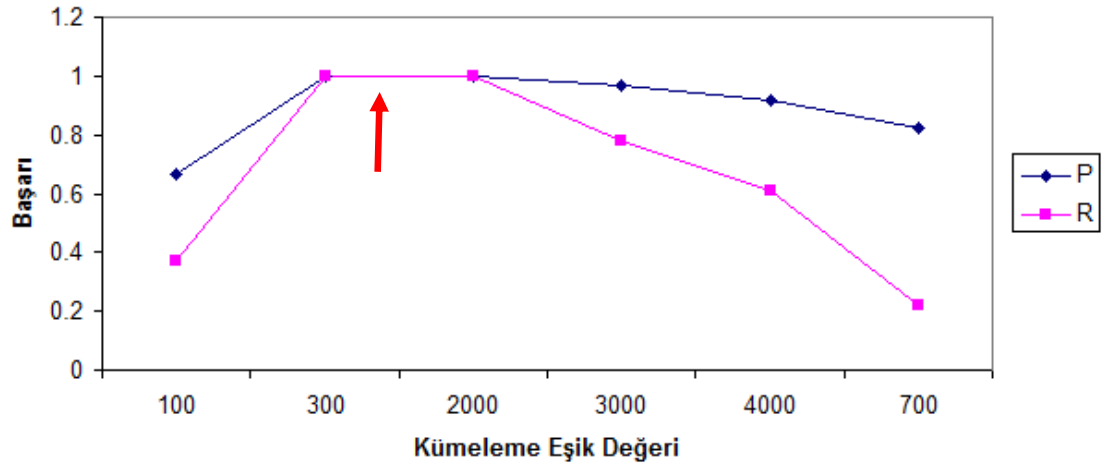
Şekil 5.4, Filtreleme eşik değerinin zamana (sn.) göre değişimi

Kümeleme Yöntemi:

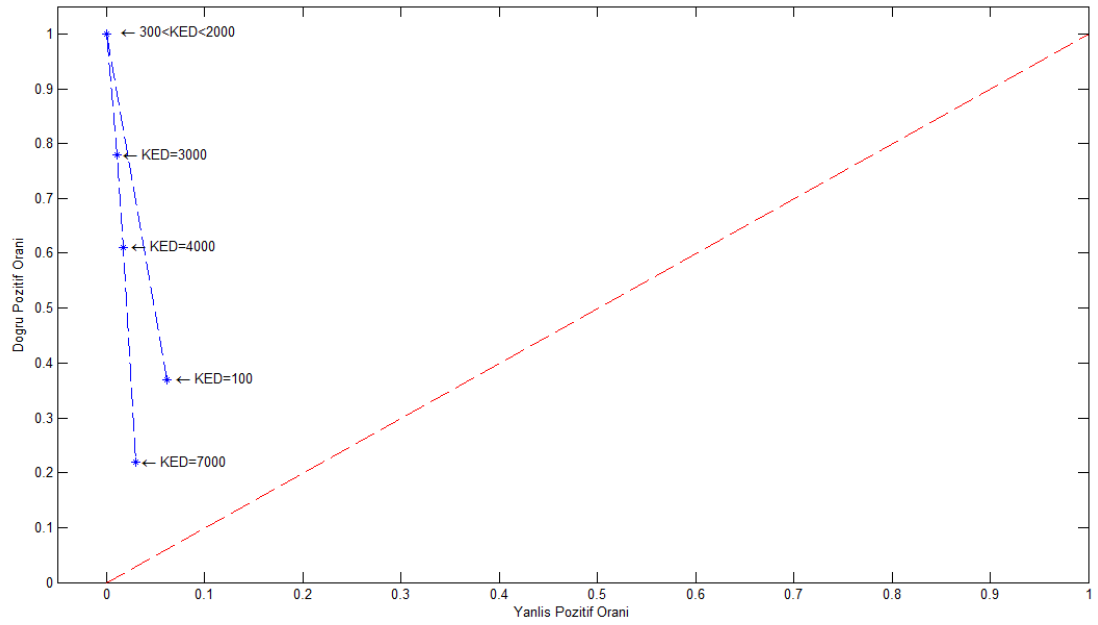
Kümeleme eşik değeri (KED): Küme ağırlık merkezleri arasındaki mesafe kümeleme eşik değerinden küçük ise kümeler iteratif olarak birleştirilir. Kümeleme eşik değeri, filtreleme eşik değeri ve delta değerlerinin seçiminden doğrudan etkilenmektedir. Farklı kümeleme eşik değerleri için P ve R değerleri şekil 5.5 de görülmektedir. Testlerde kümeleme eşik değeri olarak 500 seçilmiştir.

En düşük eşik değeri (EDED): Örüntü adaylarının örüntü olması için bu eşik değerinden daha fazla C_i sınıfına ait çizge içermesi gerekmektedir. En düşük eşik değerinin, en yüksek değer olan % 100 seçilmesi bazı örüntülerin bulunamamasına sebep olabilmektedir. Bunun sebebi örüntülerin veri kümesinde her çizgede ufak değişiklikler ile bulunması, bu sebeple oluşturulan sentetik veri kümelerinin bu özelliği taklit etmesidir. Bazı kenar ve düğüm değerlerine eklenen gürültü çok yüksek olduğundan testlerde minimum eşik değeri %75 ile %90 arasındaki değerler seçilmiş ve başarılı sonuçlar alınmıştır. Minimum eşik değerinin daha düşük olduğu değerler için, destek değeri daha düşük olan örüntüler ortaya çıkacağı için başarı düşmektedir.

En yüksek eşik değeri (EYED): Örüntü adaylarının örüntü olması için bu eşik değerinden az diğer sınıflara ait çizge içermesi gerekmektedir. Bu değer büyük seçilmesi iki sınıfa ait tekrarlayan örüntülerin çıkmasına sebep olacaktır (Bkz. şekil 5.6). Bu durum ise daha önce bahsedilen hukuk ve mühendislik öğrencilerine ait bir veri kümesinde elde edilmek istenen hukuk öğrencilerine ait ayırt edici örüntülerin yanında öğrenci olma durumunu gösteren (mühendislik öğrencisi sınıfında da ortak bulunan) örüntülerinde bulunmasına sebep olacaktır (Bkz. bölüm 3.1). Bu istenmeyen durumu önlemek için diğer bir deyişle sadece hukuk öğrencilerine ait ayırt edici örüntüleri bulabilmek için *en yüksek eşik değerinin* düşük seçilmesi gerekmektedir. Testlerde bu değer %10 ile %30 arasında seçilmektedir.

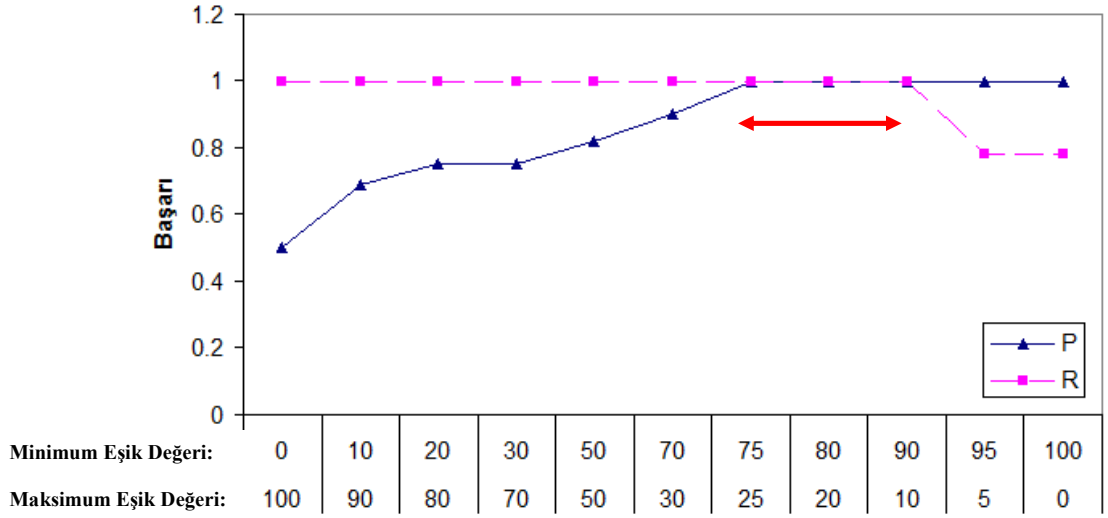


(a)

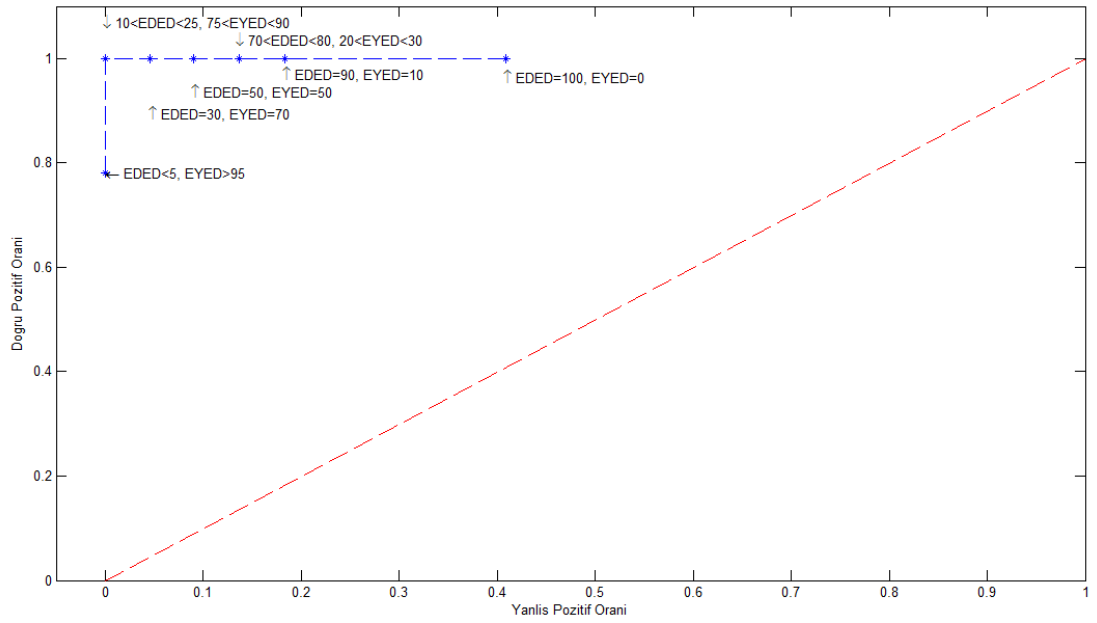


(b)

Şekil 5.5, Kümeleme eşik değerinin başarıya etkisi. (a): Kümeleme eşik değerinin P ve R değerlerine etkisi. (b): Kümeleme eşik değerinin değişimi için doğru pozitif oranının yanlış pozitif oranına göre değişim grafiği.



(a)



(b)

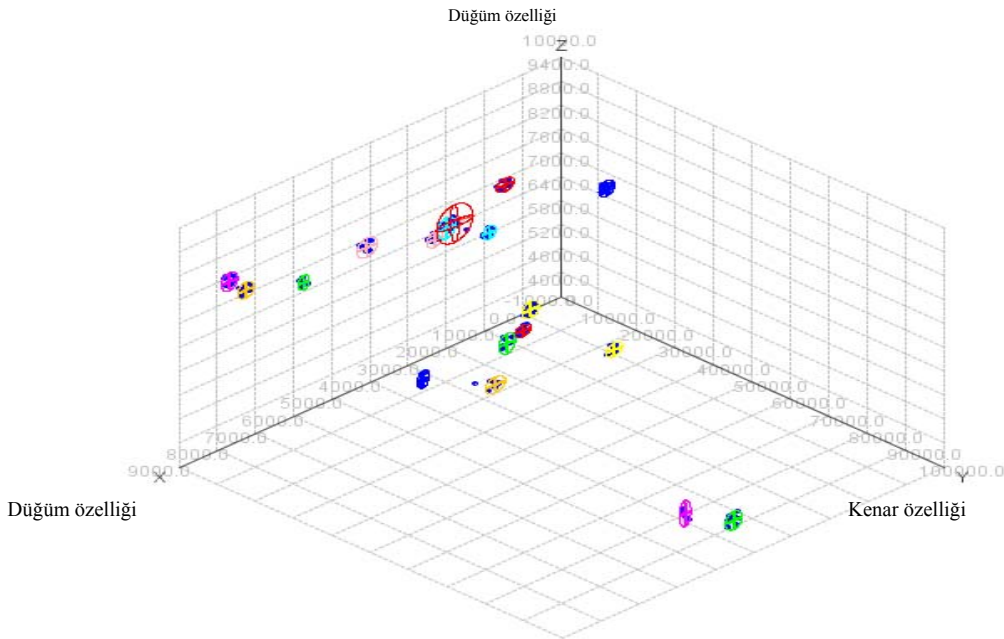
Şekil 5.6, En düşük eşik değeri ve en yüksek eşik değerinin değişiminin başarıya etkisi. (a): EDED ve EYED nin P ve R değerlerine etkisi. (b): EDED ve EYED değişimi için doğru pozitif oranının yanlış pozitif oranına göre değişim grafiği.

Bir örüntü kümesi adayının *en düşük eşik değerinden* fazla C_1 sınıfına ait çizge içeriyor ve *en yüksek eşik değerinden* az diğer sınıflara ait çizge içeriyor ise bu şartları sağlayan tüm örüntü kümelerinin toplamı örüntü olarak alınırlar. Şekil 5.7 de örüntü

kümesi adayları ve çizelge 5.2 de ise şekil 5.7 deki örüntü kümesi adaylarının C_1 ve C_2 sınıflarına ait olan kenar yüzdeleri görülmektedir. En düşük eşik değeri ve en yüksek eşik değeri çizelge 5.2 deki istatistiksel bilgiler kullanılarak seçilirler.

Örüntünün Çıkarılmasında Kullanılan Parametreler:

Şablon çizgenin belirlenmesi: Şablon çizge seçilirken C_1 sınıfına ait çizgelerden C_1 sınıfının özelliklerini en belirgin bir şekilde barındıran çizgelerden biri şablon çizge olarak seçilir. Şablon çizgenin seçimi elde edilen örüntüyü etkileyeceği için bir uzmanın çeşitli şablon çizgelerden elde edilen örüntüleri inceleyerek kesin örüntüyü belirlemesi gerekmektedir. Şekil 5.8 - 5.10 da çeşitli gürültü_m değerleri ile farklı şablon çizgeler kullanılarak elde edilen örüntüler için P ve R değerleri gösterilmektedir. Testlerde Gauss eğrisi kullanılarak rastgele eklenmiş farklı gürültü değerleri kullanılmıştır (Testlerin tümünde gürültü₀=1 alınmıştır).



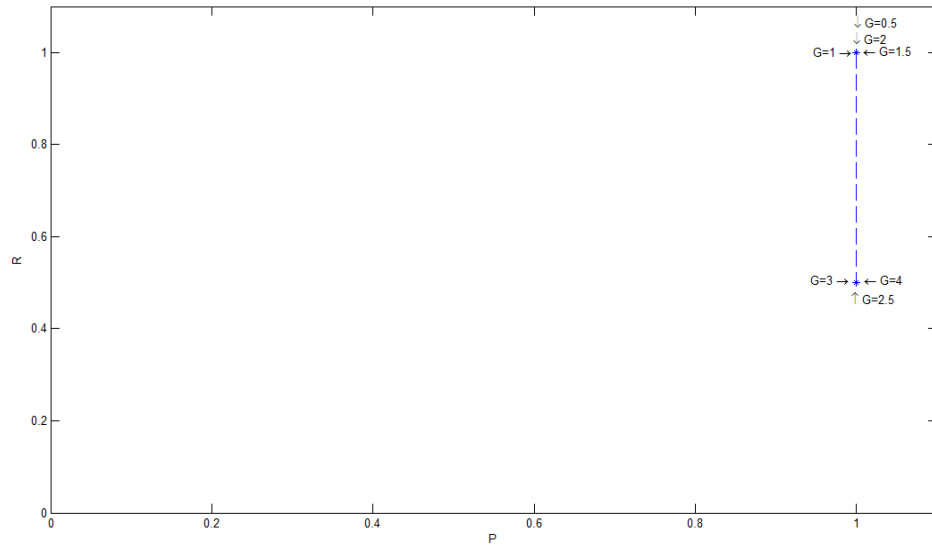
Şekil 5.7, Elde edilen veri kümesindeki kümeler (Kümelere ait istatistiksel bilgiler çizelge 5.2 de listelenmektedir.)

Çizelge 5.2, Şekil 5.6 deki küme adaylarına ait istatistiksel bilgiler

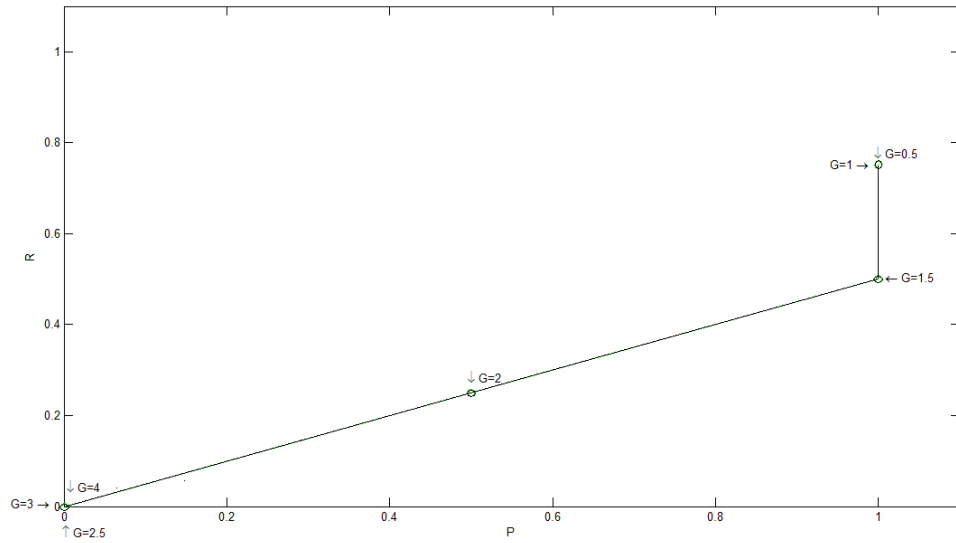
	C ₁ sınıfı	C ₂ sınıfı		C ₁ sınıfı	C ₂ sınıfı
Küme 1	100	0	Küme 11	100	100
Küme 2	100	0	Küme 12	100	90
Küme 3	100	0	Küme 13	100	100
Küme 4	100	0	Küme 14	100	40
Küme 5	90	0	Küme 15	100	90
Küme 6	90	0	Küme 16	100	90
Küme 7	100	0	Küme 17	100	60
Küme 8	100	0	Küme 18	100	30
Küme 9	100	0	Küme 19	10	0
Küme 10	100	100			

Her gürültü değeri için C₁ sınıfına ait çizgelerin her biri şablon çizge olarak ayrı ayrı alındığında elde edilen en yüksek, en düşük ve ortalama P ve R değerleri şekillerde görülmektedir. Görüldüğü gibi gürültü_m 2 ye kadar olan değerler için önerilen yöntem şablon çizgelerden en az biri en yüksek başarı değeri olan P=1, R=1 değerlerini bulmaktadır (Bkz. şekil 5.8). Gürültü_m 2.5 tan büyük olduğunda ise şablon çizgelerden elde edilen en yüksek başarı değeri P=1 ve R=0.5 dir. Şablon çizgelerden en düşük P ve R değerlerini veren şablon çizgeden elde edilen P ve R değerleri şekil 5.9 gösterilmiştir. Yöntem, gürültü_m 0.5 ile 1.5 arasında olduğunda ise en düşük P=1, gürültü 0.5 ile 1 arasında en düşük R= 0.75 değerini vermektedir. Gürültü_m arttırıldığında en düşük P ve R değerlerinin düştüğü görülmektedir. En düşük P değeri gürültü_m 2.5 olduğuna 0 olmaktadır.

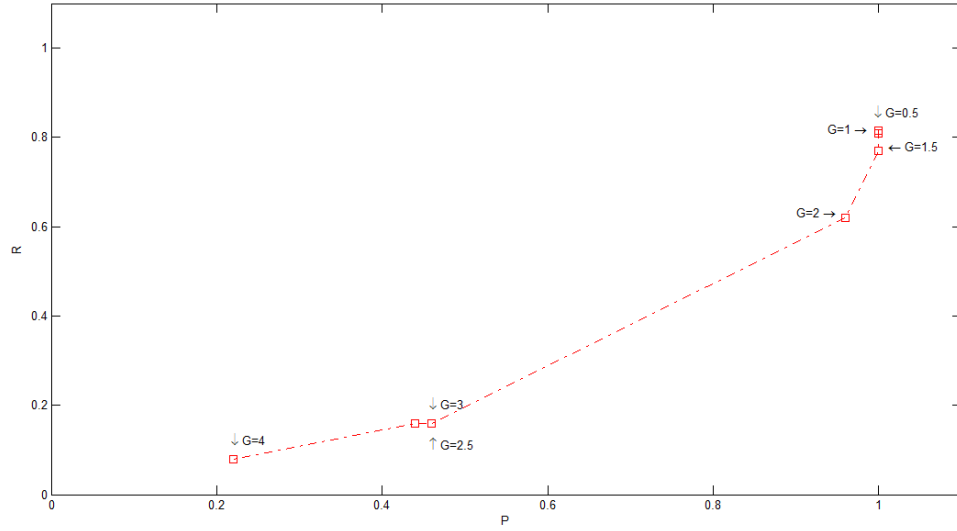
Şekil 5.10 da tüm şablon çizgelerin her birinden elde edilen P ve R değerlerinin ortalaması gösterilmektedir. Ortalama P ve R değerleri dikkate alınarak gürültü 1.5 değerine kadar önerilen yöntem başarılı sonuçlar vermektedir (P=1, R=0.77). Şekil 5.11 de şablon çizgelerden elde edilen örüntülerden için hesaplanan P ve R değerlerinin standart sapması görülmektedir. Şekil 5.8-5.11 deki sonuçlar dikkate alınarak gürültü_m 0.5 ile 1.5 değerleri arasında başarılı sonuçlar elde edilmektedir. Gürültü_m in 1.5 dan yüksek olduğu değerlerde şablon çizgenin yanlış seçimi başarıyı düşürmektedir.



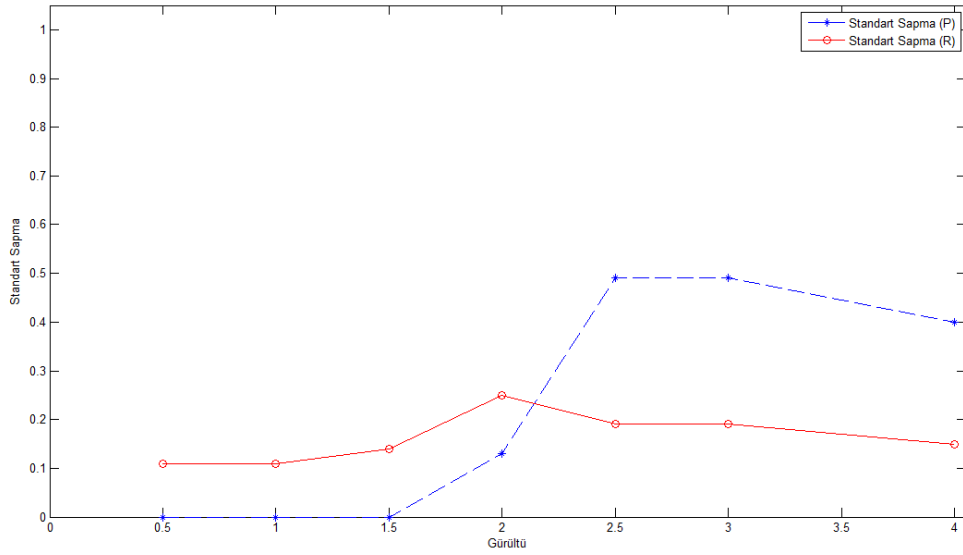
Şekil 5.8, Gürültü_m parametresinin değişimine göre en yüksek P-R değişimi grafiği



Şekil 5.9, Gürültü_m parametresinin değişimine göre en düşük P-R değişimi grafiği



Şekil 5.10, Gürültü_m parametresinin değişimine göre ortalama P-R değişimi grafiği



Şekil 5.11, Gürültü_m parametresinin değişimine göre P ve R deki standart sapma

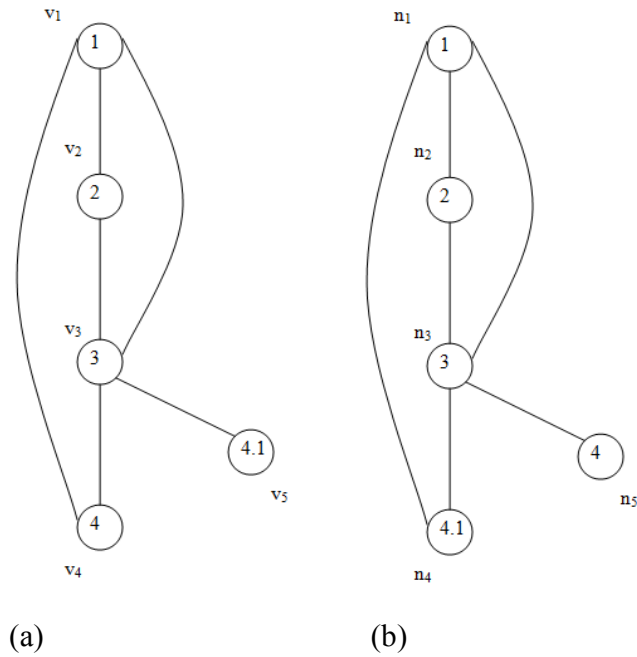
5.3.2 Literatürdeki Yöntemler ve Benzerlik Kriteri

Literatürdeki yöntemlerin, benzerlik kriterinin eklenebilmesi için uygunluğu incelenmiştir. Literatürdeki g-Span, FSG ve Gaston yöntemleri benzerlik kavramının eklenebilmesi için uygun olmadığı görülmüştür. Bunun sebebi g-Span yönteminde örnek verilerek açıklanacaktır. g-Span yöntemi tekrarlayan alt çizgeleri bulma problemini DFS

kodunda birbiriyle ilişkili kısımları çıkarma problemine dönüştürür. Bunun için her çizgeyi DFS kodlarına dönüştürür. DFS kodları, çizgelerden derinlemesine arama yöntemi kullanılarak elde edilen dizilerdir. Daha sonra DFS kodlarından örüntüleri çıkarmak için diziden örüntü bulma yöntemlerini kullanır. Bunun için DFS ağaçları kullanılmaktadır. Şekil 5.12 de iki benzer çizge örneği görülmektedir. Şekil 5.12 (a) daki çizgeden derinlemesine arama yöntemi kullanılarak elde edilen dizi $\{(v_1, v_2), (v_2, v_3), (v_3, v_1), (v_3, v_4), (v_4, v_1), (v_3, v_5)\}$ dir. Şekil 5.12 (b) deki çizgeden aynı şekilde elde edilen dizi $\{(n_1, n_2), (n_2, n_3), (n_3, n_1), (n_3, n_5), (n_3, n_4), (n_4, n_1)\}$ dir.

Tanım 5.1:

G ve G' çizgeleri verilsin. Bu iki çizgenin en küçük DFS kodları birbirine eşit ise bu iki çizge izomorfiktir [Yan and Han, 2002].



Şekil 5.12, Benzer iki çizge örneği

Şekil 5.12 deki (a) ve (b) çizgeleri arasında bir izomorfizm görülmesine rağmen tanım 5.1 e dayanarak benzer bu iki çizgenin DFS kodları farklı olduğu için izomorfik olmayacaktır. Literatürdeki FSG ve Gaston gibi diğer yöntemlerde aynı sebepten benzer örüntüleri bulamamaktadırlar.

Sadece Subdue yaklaşımı benzer örüntüleri bulabilmeye olanak tanımaktadır. Bölüm 2.2.2 de Subdue yaklaşımı anlatılmıştır. Subdue yaklaşımı, literatürdeki diğer yaklaşımlardan tamamıyla farklı olarak veriyi sıkıştırma yoluna gitmektedir. Subdue yaklaşımı orijinal halinde tam olmayan eşlemeye izin vermektedir. Diğer bir deyişle bir örüntü veri kümesindeki çizgelerde bazı kenarları eksik olarak tekrarlandığında subdue bu örüntüyü bulabilmektedir. Tezde önerilen yaklaşımı kıyaslayabilmek için Subdue yaklaşımını değiştirilip benzer örüntüleri bulabilmesi sağlanmıştır.

5.3.3 Gürültünün Etkisi

Literatürde başarının yorumlanmasında çeşitli kriterler kullanılmaktadır. Daha önce bahsedilen P ve R kriterlerinin yanında doğru pozitif ve yanlış pozitif oranları sıkça kullanılmaktadır. Bu bölümde doğru pozitif ve yanlış pozitif oranlarından kısaca bahsedilecek daha sonra gürültünün tezde önerilen yöntem ve Subdue yaklaşımına etkisi bu kriterler ile yorumlanacaktır.

Önerilen yaklaşımın başarısı çeşitli parametrelere bağlıdır. Bu parametre değerinin değişmesi ile doğru pozitif ve yanlış pozitif oranları değişmektedir. Parametrenin ne kadar iyi ya da maliyetli sonuçlar doğuracağını belirlemek için doğru pozitif ve yanlış pozitif oranları kullanılmaktadır.

$$\text{Dogru Pozitif Oranı} = \frac{|Dogru Pozitif|}{|Dogru Pozitif| + |YanlisNegatif|} \quad (5.3)$$

$$\text{Yanlis Pozitif Oranı} = \frac{|Yanlis Pozitif|}{|Yanlis Pozitif| + |Dogru Negatif|} \quad (5.4)$$

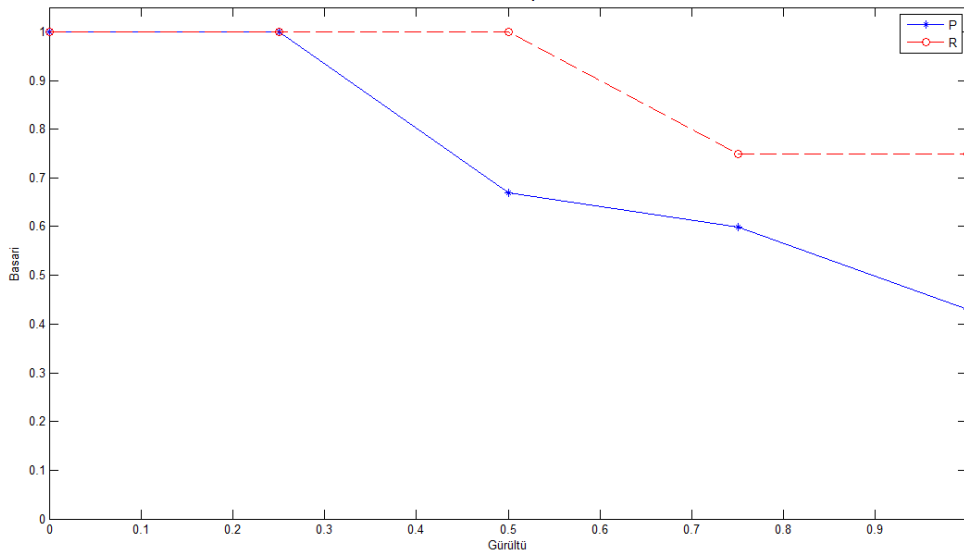
Doğru Negatif (True Negative): Doğru olarak tahmin edilen örüntüye ait olmayan kenar sayısıdır.

Yanlış Negatif (False Negative): Yanlış tahmin edilen örüntüye ait olan kenar sayısıdır.

Doğru Pozitif (True Positive): Doğru tahmin edilen örüntüye ait olan kenar sayısıdır.

Yanlış Pozitif (False Positive): Yanlış tahmin edilen örüntüye ait olmayan kenar sayısıdır.

Önerilen yöntemin başarısını göstermek için çeşitli kriterler kullanılacaktır. Bunlar *gürültü* ($gürültü_o$ ve $gürültü_m$), *zaman*, *başarı*, *çizge sayısı*dir. Testlere çeşitli $gürültü_o$ değerlerinde düzgün dağılımlı gürültü eklendiğinde tezde önerilen yöntem %100 başarı ile örüntüleri bulmaktadır. Şekil 5.13 ve şekil 5.14 de görüldüğü gibi Subdue ise 0.25 gürültü ekleme oranına ($gürültü_o$) kadar örüntüleri başarı ile bulurken gürültü arttırıldıkça başarısı düşmüştür.

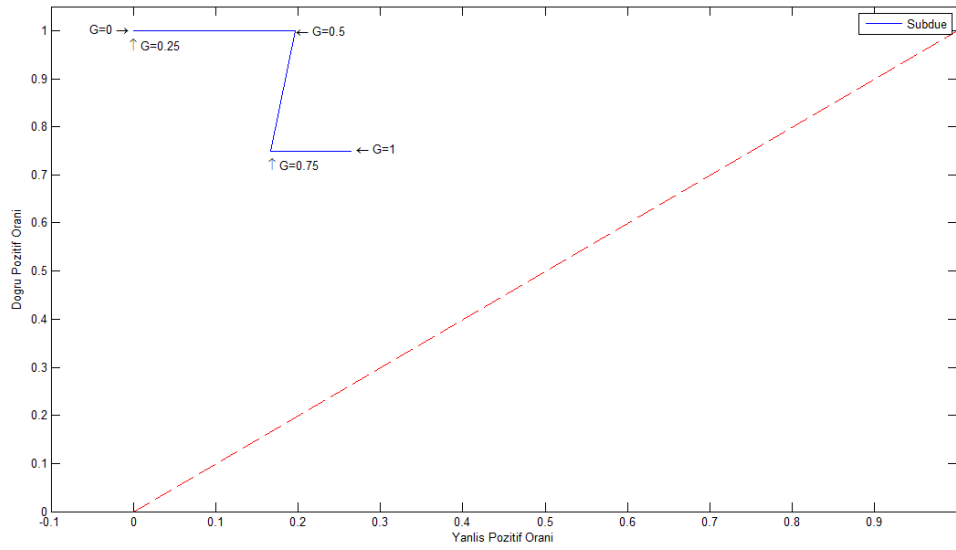


Şekil 5.13, Subdue yaklaşımında kenar ve düğümlere $gürültü_o$ parametresinin başarıya etkisi.

Tüm kenar ve düğümlere gürültü eklendiğinde ($gürültü_o=1$) başarı P %43'e düştüğü görülmekte, bu durum ise subdue yaklaşımının bulduğu örüntünün orjinal örüntüyü içeren daha büyük bir alt çizge bulunduğunu göstermektedir. Subdue $gürültü_o=1$ olduğunda orjinal örüntünün %75'ini bulabilmektedir.

Tez çalışmasında önerilen yaklaşımda çizge eşleme problemini çözmeye yönelik bir yöntem ya da bir yaklaşım getirilmemiştir. Tezde önerilen yaklaşım veri madenciliğine dayanan bir yaklaşımdır. Ve temelde özellik çıkarma yöntemi olarak düşünülebilir. Tezde önerilen yaklaşım ard arda devam eden dört alt adımdan

oluşmaktadır: yoğunluk haritalarının oluşturulması, filtreleme, kümeleme ve örüntülerin çıkarılması. Bu alt adımların zaman karmaşıklığı çizelge 5.3 de listelenmiştir. En yüksek zaman karmaşıklığı, ortalama bağlantı yönteminin kullanıldığı kümeleme adımında görülmektedir. Toplamda önerilen yaklaşımın zaman karmaşıklığı $O(n^3 \times m^6)$ olarak görülmektedir. Burada n veri kümesindeki çizge sayısını ve m ise veri kümesindeki en fazla sayıda düğüm içeren çizgedeki düğüm sayısını göstermektedir.

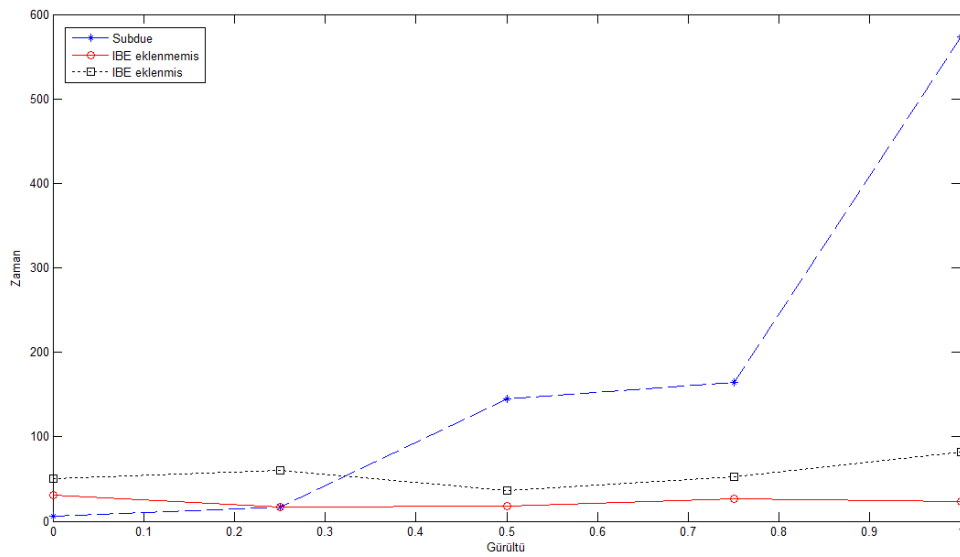


Şekil 5.14, Subdue yönteminin çeşitli gürültü, G değerleri için doğru pozitif oranının yanlış pozitif oranına değişim grafiği

Çizelge 5.3, Önerilen yöntemin ara adımlarındaki en kötü durum zaman karmaşıklığı.

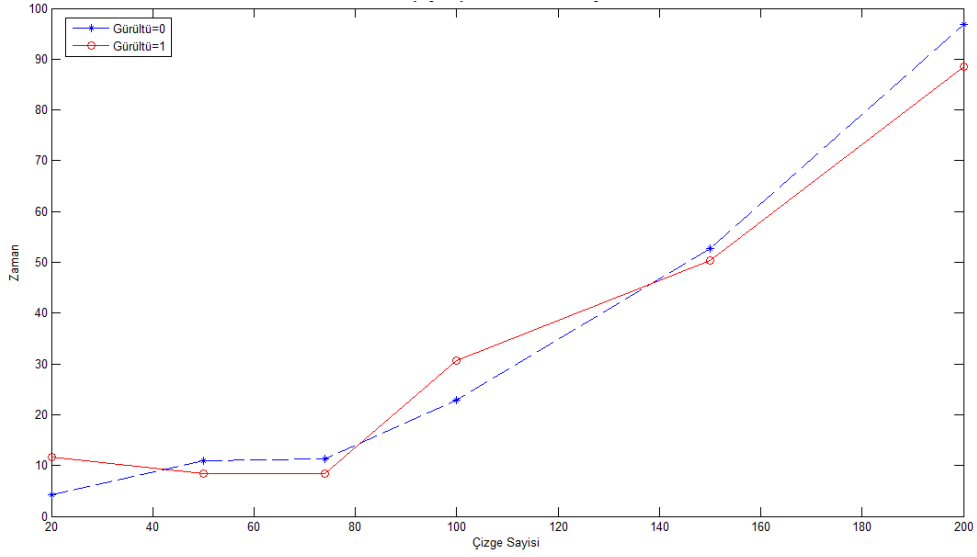
	Zaman karmaşıklığı
Yoğunluk haritalarının oluşturulması	$O(n \times m^2)$
Filtreleme	$O(n \times m^2)$
Kümeleme	$O(n^3 \times m^6)$
Örüntünün çıkarılması	$O(n \times m^4)$
Toplam	$O(n^3 \times m^6)$

Şekil 5.15 de gürültünün zamana göre değişim grafiği görülmektedir. Bu deneyde 100 çizgeden oluşan veri kümeleri kullanılmıştır. Görüldüğü gibi Subdue yönteminde gürültü arttıkça işlem zamanı hızlı bir şekilde artmaktadır. Önerilen yöntemde gürültünün artmasının işlem zamanına belirgin bir etkisi yoktur. Şekil 5.16 da veri kümesindeki çizge sayısındaki artışın işlem zamanına etkisi görülmektedir. Beklendiği gibi çizge sayısı arttırıldığında işlem zamanı da artmaktadır. Gürültünün miktarının çizge sayısı arttırıldığında da işlem zamanına belirgin şekilde etki etmediği görülmüştür.



Şekil 5.15, 100 çizgili bir veri kümesinde çeşitli gürültü₀ değerlerinin zamana (sn.) etki grafiği.

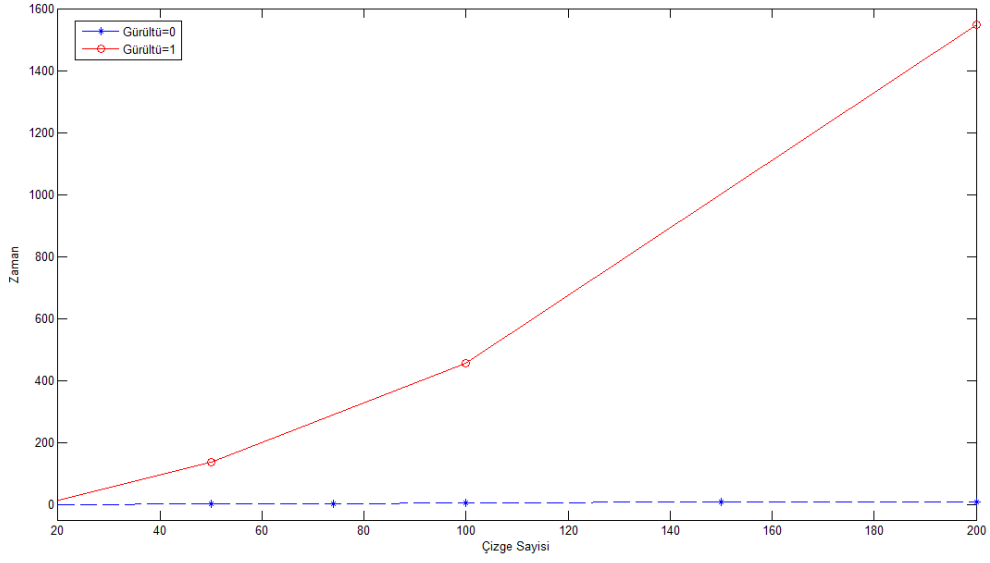
Subdue yaklaşımında ise gürültü₀=0 olduğunda 200 çizgili bir veri kümesini ortalama 6.77 sn gibi kısa bir sürede bulurken, gürültü eklendiğinde 200 çizgili veri kümesi için ortalama süre 1549 saniyeyi bulmaktadır (Bkz. şekil 5.17). Önerilen yöntem ise gürültü₀=0 ve gürültü₀=1 olduğunda sırasıyla 89 ve 97 saniyede örüntüleri bulmaktadır. Görüldüğü gibi Subdue gürültüsüz ortamda önerilen yöntemden çok daha hızlı bir şekilde örüntüleri bulurken gürültü eklendiğinde önerilen yöntemle göre çok daha yavaş çalışmaktadır.



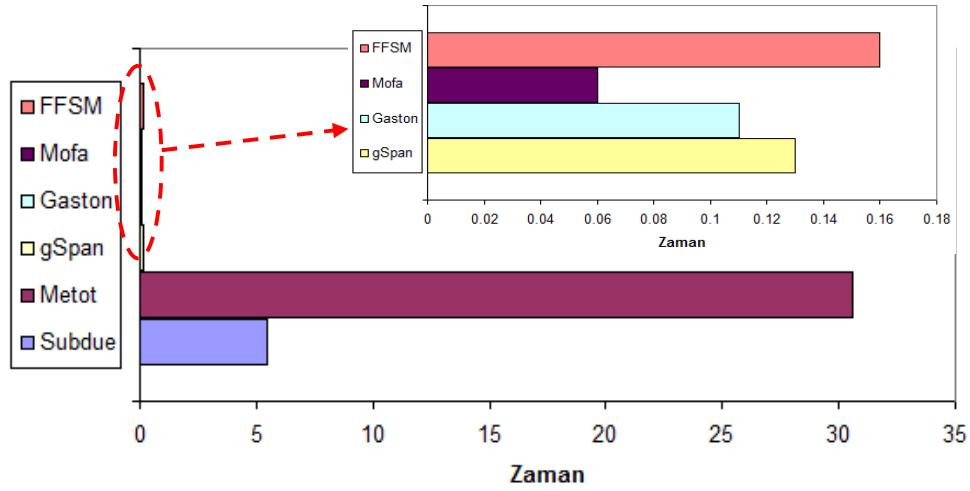
Şekil 5.16, Önerilen yöntem için veri kümesindeki çizge sayısındaki artışın gürültü, değerleri 0 ve 1 olduğunda zamana (sn.) etkisi

Literatürdeki yaklaşımların işlem zamanlarını gürültüsüz ortamda karşılaştırmak için 100 çizgeden oluşan veri kümesinde, 5 düğümlü bir örüntü gürültü₀=0 olacak şekilde gömülmüştür. Subdue, önerilen yöntem, FFSM, Mofa, Gaston, gSpan yaklaşımları başarıyla (P=1, R=1) buldukları gözlemlenmiştir. Şekil 5.18 de bu yaklaşımların işlem zamanları görülmektedir. Önerilen yöntem en çok zaman harcayarak, 31 saniyede örüntüyü bulmuştur. İkinci sırada ise Subdue 5.5 saniye harcamaktadır. Diğer yöntemler ise 1 sn den kısa bir sürede işlemi başarı ile tamamlamışlardır.

Test sonuçlarında gürültü₀ arttırıldığında önerilen yaklaşımın örüntüleri başarı ile bulabildiği görüldü. Gürültü miktarının yöntemin başarısını nasıl etkileyeceğinin daha iyi anlaşılması için gürültü Gauss dağılımı kullanılarak çeşitli oranlarda arttırıldı.



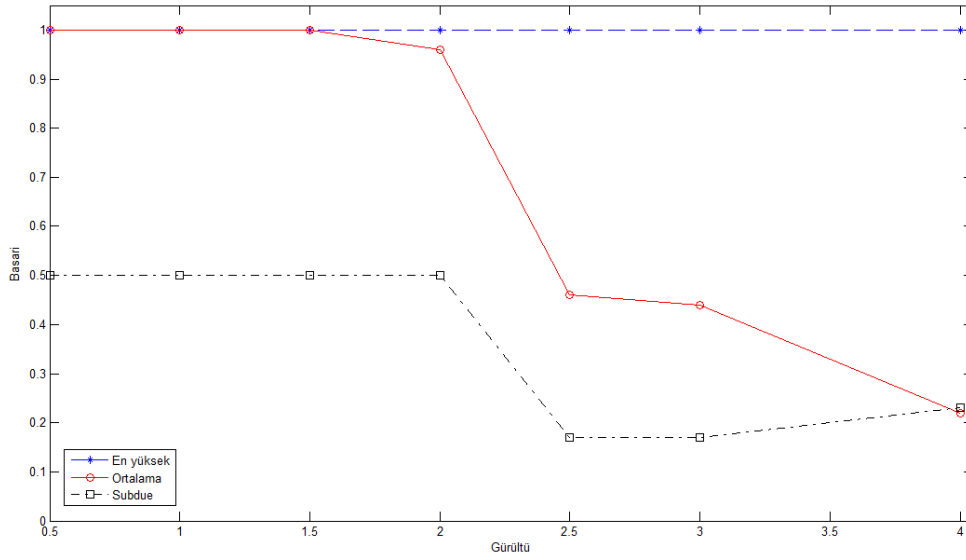
Şekil 5.17, Subdue için veri kümesindeki çizge sayısındaki artışın gürültü değerleri 0 ve 1 olduğunda zamana etkisi



Şekil 5.18, 100 çizgeden oluşan gürültüsüz bir veri kümesinde FFSM, Mofa, Gaston, gSpan, önerilen yöntem ve Subdue'nun ortalama harcadığı zaman (sn)

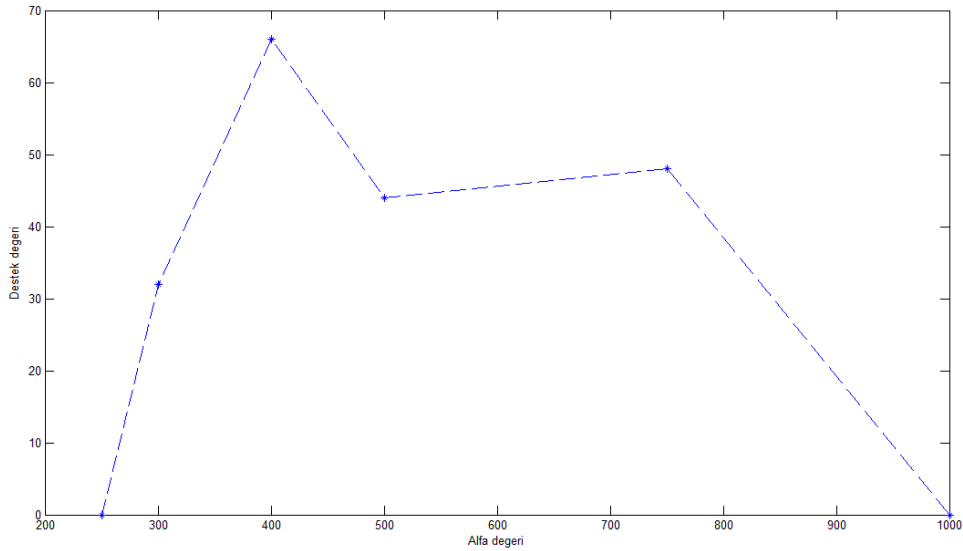
Gürültü eklenirken kullanılan Gauss eğrilerindeki standart sapma değerleri gürültü_m parametresi yardımıyla değiştirilerek gürültü artırılıp başarı test edildi. Şekil 5.19 ve şekil 5.20 de görüldüğü gibi önerilen yaklaşım uygun bir şablon çizgenin seçimi ile gürültü_m 0.5 ile 2 değeri arasında örüntüyü başarı ile bulmuştur (P=1, R=1). Subdue ise gürültüsüz ortamda örüntüyü başarı ile bulurken gürültü arttırıldığında gürültü_m 0.5

ile 2 değeri arasında başarısı düşmüştür ($P=0.5$ ve $R=0.7$). Önerilen yöntemde farklı şablon çizgelerin seçiminin başarıyı nasıl etkilediğini görmek için ortalama P ve R değerleri de şekilde görülmektedir. Gürültü_m 0.5 ile 1.5 değerleri arasında önerilen yöntemden elde edilen ortalama P ve R değerleri Subdue dan daha başarılı sonuçlar verdiği görülmektedir. Şekil 5.21 de eğriler incelenirse düşük gürültülü veri kümelerinde Subdue nun başarılı sonuçlar verdiği görülmektedir ($0.5 < \text{gürültü}_m < 2$). Fakat önerilen yöntemin başarısı Subdue yaklaşımı ile karşılaştırılırsa önerilen yöntemin gürültüden daha az etkilendiği görülmektedir.

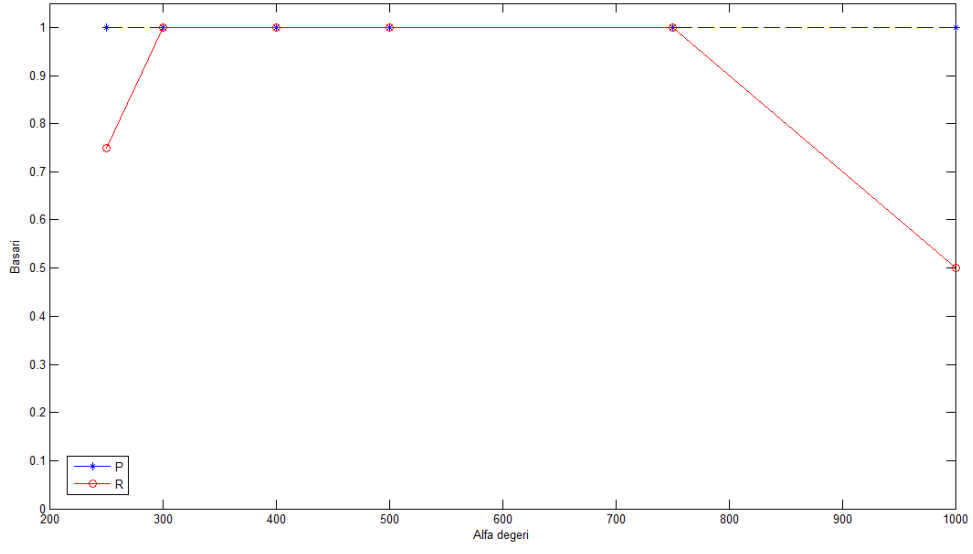


Şekil 5.19, Gauss dağılımı ile rastgele oluşturulmuş gürültü_m nin değişiminin P değerine etkisi (gürültü₀=1 için)

orjinal örüntüyü bulabilmişse bulduğu örüntünün destek değeri gösterilecek, eğer örüntünün tamamı bulunamamış ise $P < 1$ veya $R < 1$ elde edilmiş ise destek değeri 0 olarak alınacaktır. Buradaki amaç farklı alfa değerleri için orjinal örüntünün veri kümesindeki frekansını hesaplamaktır. Önerilen yaklaşımda benzerlik eşik değeri α , kümeleme eşik değeri parametresidir. Şekil 5.22 de çeşitli α değerleri için destek değerleri görülmektedir. Şekil 5.23 de ise aynı alfa değerlerinden elde edilen örüntülerin başarı değerleri P ve R görülmektedir. Şekil 5.23 de görüldüğü gibi alfa benzerlik parametresi 300 ile 750 arasında önerilen yaklaşım orjinal örüntüyü bulmuştur. Yöntem bu benzerlik değerleri arasında elde edilen örüntünün çizgelerin en az %32 sinden geçtiğini bulmuştur. Benzerlik parametresi 400 seçildiğinde ise elde edilen örüntü veri kümesindeki çizgelerin % 66 sında geçtiği bulunmaktadır. Üretilen sentetik veri kümesinde örüntü çizgelerin tümünde bulunmaktadır. Fakat gürültüden dolayı bu değer %66 olarak görülmektedir. Benzerlik eşik değerinin çok düşük seçilmesi gürültünün çok olduğu durumlarda örüntüyü bulamamasına sebep olmaktadır. Benzerlik değerinin çok büyük seçilmesi ise tüm kenar ve düğümlerin birbirine benzer olarak almasına sebep olmakta ve örüntü kümelerine diğer sınıflara ait çizgelerinde girmesine sebep olmaktadır.



Şekil 5.22, Önerilen yöntem için alfa değerinin değişiminin destek değerine etkisi



Şekil 5.23, Önerilen yöntem için alfa değerinin değişiminin başarıya etkisi

6 SONUÇLAR

Çizge veri yapılarının birçok alanda kullanımının artması, çizge tabanlı veri madenciliği yöntemlerinin önemini son senelerde arttırmaktadır. Literatürde tekrarlayan örüntü madenciliği problemi için birçok yaklaşım önerilmiştir. Fakat bu yaklaşımlar örüntülerin veri kümesinde ufak değişikliklerle tekrarlandığı ya da gürültülü veri kümeleri için uygun değildir.

Tez çalışmasında, sadece tekrarlayan örüntü madenciliği problemine yeni bir yaklaşım getirilmedi, bunun ötesinde tekrarlayan benzer örüntü madenciliği olarak adlandırılan ve tekrarlayan alt çizge madenciliği problemini kapsayan daha önce çalışılmamış bir probleme de bir yaklaşım sunulmuş olundu. Önerilen yaklaşım literatürdeki diğer yaklaşımlardan farklı olarak çizgeleri özellik uzayına dönüştürerek örüntüleri bulma yoluna gitmiştir. Benzer örüntüleri, veri madenciliği, görsel veri madenciliği ve kümeleme yöntemlerinin avantajlarını kullanarak bulmaktadır. Ayrıca çok sınıflı uygulamalarda da başarı ile uygulanabilmektedir.

Önerilen yaklaşımın getirdiği bir avantaj da görsel grafikleri ve istatistiksel verileri kullanarak ara adımlarda kullanıcıya bilgi vererek ve gerekli yerlerde parametrelerde ince ayar yapmasına olanak vererek, kullanıcının uzmanlığını yöntemeye katmaya olanak vermesidir. Özellikle ilaç dizaynı gibi uygulamalarda uzmanların bilgisi hayati derecede önemlidir.

Tekrarlayan benzer örüntüleri bulmak için geliştirilen yaklaşımın performansı gerçek bir veri kümesinde ve çeşitli sentetik veri kümelerinde test edildi. Önerilen yaklaşım gürültüsüz ortamda tüm tekrarlayan örüntüleri ve tekrarlayan benzer örüntüleri %100 doğru bir şekilde bulmaktadır. Diğer bir yandan gürültülü ortamda ise örüntüleri %95 e yakın bir ortalama ile bulmaktadır.

Önerilen özellik uzayına dönüşüm, çizge tabanlı veri kümelerini kullanarak yeni çizgelerin sınıflandırılmasına olanak vermektedir. Bu konu tezin kapsamı dışında

olmasına rağmen, elde edilen örüntülerin ileriki adımlarda veri kümelerinde bu örüntüleri içeren çizgelerin taranmasını gerektirecektir. Bu sebeple önerilen yöntemin sınıflandırmaya da imkân vermesi gerektiği düşünülmektedir. Kullanılan dönüşümün gelecek çalışmalara ışık tutması açısından bu konu ile ilgili ön çalışma yapılmış ve bölüm 7.1 de bahsedilmiştir. Bu konu tez sonrası çalışmalarda ayrıntılı olarak incelenecektir.

Geliştirilen yöntemi 3 boyutlu uzayda kenar ve düğümlerinde sadece tek özellik için çalışabilecek şekilde geliştirilmiştir. Fakat özellik uzayında çalışıldığından yöntem çok boyutlu uzayda çalışmaya olanak verecek yapıdadır. Tezden sonraki gelecek çalışmalarda yöntemin düğüm ve kenarlarda birden fazla özellik içeren çizge tabanlı veri kümelerinde de çalışabilmesine olanak verecek iyileştirmeler üzerinde çalışılacaktır.

Son olarak tez çalışmasının esas amacı tekrarlayan benzer örüntü madenciliği problemine kullanıcı destekli bir görsel veri madenciliği yöntemi ile bir yaklaşım getirmek olduğu için önerilen yöntem henüz hız açısından optimize edilmemiştir. Bu sebeple tez sonrası çalışmalarda yöntemi hız açısından optimize etmeyi amaçlamaktayız.

7 GELECEK ÇALIŞMALAR

Sonuçlar kısmında bahsedilen gelecek çalışmalara ek olarak bu bölümde tez sırasında gelecek çalışmalara ışık tutması açısından yapılan ön çalışmalardan bahsedilecektir.

7.1 Çizgelerin Sınıflandırılmasında Benzer Alt Çizgelerin Özellik Olarak Kullanımı

Benzer alt parçaları bulmak için önerilen özellik uzayına dönüşüm, yeni çizge tabanlı örneklerin sınıflandırılmasına olanak sağlayacak şekilde uyarlanabilmektedir. Bunun için elde edilen benzer örüntüler kullanılarak yeni çizgenin örüntüye benzer bir örüntü içerip içermediğine bakılır. Yeni çizgelerin sınıflandırılması tezin amacı dışındadır. Fakat kullanılan özellik uzayı kavramının yeni çizgelerin sınıflandırılmasında da kullanılabileceğini göstermek ve gelecek çalışmalara ışık tutması amacıyla bu bölümde bu konuya giriş yapma gereği duyulmuştur.

Bir alt çizgeyi içeren çizgeleri hızlı bir şekilde geniş veritabanlarında arama önemli bir çalışma alanıdır. Literatürde en yaygın uygulama alanı biyolojide geçmektedir. Çizge madenciliği (çizge arama) yöntemleri ile geniş molekül veritabanlarında belli bir hastalığı iyileştirebilecek (diğer bir deyişle aktif örüntüleri içeren) moleküller taranır. Genellemek gerekirse çizge tabanlı verilerin sınıflandırılması iki gruba ayrılır. İlk grupta çizgenin tamamından çıkarılan karakteristik özellik vektörlerine klasik sınıflandırma yöntemleri uygulanır [An and Wang, 2001]. Diğer grupta ise elde edilen tekrarlayan alt parçalar kullanılarak ya da bu parçalardan özellikler çıkarılarak sınıflandırma yapılır [Deshpande et al., 2005]. Bu çalışmada ikinci grupta bahsedilen sınıflandırma yöntemi kullanılmıştır.

Klasik çizge arama yöntemleri ile geniş moleküler veritabanlarında belli örüntüleri (alt çizge) içeren moleküllerin bulunması alt çizge izomorfizm probleminin zaman karmaşıklığından dolayı zordur. Bu sebeple çizge arama yöntemleri yerine makine öğrenmesi yöntemlerine dayanan yeni bir yöntem kullanıldı. Uygulanan yöntemin temeli [Deshpande et al., 2005] ya dayanmaktadır. Makine öğrenmesi yöntemlerinden veri kümesindeki her örneğin aynı boyutta vektörlerden oluşması gerekmektedir. Fakat her biri farklı sayıda düğüm içeren çizge veri kümelerinde, her çizgeyi tek bir vektör ile ifade etmek zordur. Bu sebeple önce her sınıf için aday örüntü kümeleri çıkarılır (Bölüm 4.3). Daha sonra bu aday örüntü kümeleri özellik olarak kullanılır. Bir örnekle açıklamak gerekirse, iki sınıflı bir veri kümesinde birinci sınıfa ait aday örüntü sayısı n_1 ve ikinci sınıfa ait aday örüntü sayısı ise n_2 olsun. Veri kümesindeki her çizge için $n_1 + n_2$ boyutlu özellik vektörleri oluşturulur. Bu vektörde her boyut bir örüntü adayını göstermektedir. Eğer bir çizgede bir örüntü adayı var ise, vektörde o örüntü adayına karşı gelen boyut, çizgenin noktalarından o örüntü adayı kümesinin merkezine en yakın olanının Öklid uzaklık değerini alır. Eğer noktalardan hiç biri örüntü adayı kümesine düşmüyor ise (diğer bir deyişle örüntü çizgenin bir parçası değil ise) özellik vektöründe o örüntü adayının bulunduğu boyut sıfır olur.

Problem çok sayıda sınıflı durumlar için aşağıda genellenecektir. Daha önceki bölümlerde bahsedilen ortak yapıların (alt çizge) çıkarılması yöntemleri ile her sınıf C_i için ortak alt çizgeler çıkarıldıktan sonra, bu alt çizgelerden özellik çıkarılır. n_i her sınıfa ait toplam aday alt parça sayısı olsun. Elde edilen örüntü alt parçalarını kullanmak yerine tüm aday alt parçaların kullanılmasının sebebi, en önemli alt parçaların seçimini sınıflandırma yöntemine bırakmaktır. Her örnek için $\sum_{i=1}^{sınıfsayı} n_i$ boyutlu özellik vektörü oluşturulur. Her boyut, o örneğin benzer aday alt çizgelerini temsil eden küme gruplarının merkezine yakınlığını ifade eder. Eğer uzaklık değeri belli bir eşik değerinden büyükse sıfır alınır.

Bu sayede her çizge için aynı boyutta vektörlerden oluşmuş ve etiketlenmiş (her çizgenin sınıf bilgisi de eklenmiş) bir veri kümesi oluşturulur. Daha sonra bu veri

kümesine literatürde sıkça kullanılan sınıflandırma yöntemlerinden *karar ağaçları* (*Decision trees*), *lineer diskriminant analizi*, *k-en yakın komşuluk algoritması*, *destek vektör makinaları* (*support vector machine*) sınıflandırma yöntemleri uygulanır [Alpaydın E., 2001]. Son olarak veri kümesinde her seferinde bir örnek dışarıda bırakılarak, tek örnekli bir test kümesi ve geri kalan örnekler öğrenme kümesi alınarak (leave one out cross validation) çeşitli sınıflandırma yöntemleri için performans test edildi. Önerilen yöntemin performansının görülebilmesi için filtreleme yapılmadan orijinal veri kümesi üzerine, önerilen benzer alt çizge çıkarma yöntemini uygulanarak karşılaştırıldı.

Çizelge 7.1 de görsel veri madenciliği kullanılarak ($\Delta_x=0.12$ ve $\Delta_y=0.07$) filtrelenmiş ve filtrelenmemiş veri kümesi kullanılarak elde edilen sınıflandırma başarıları ve işlem süresi görülmektedir. Elde edilen sonuçlara göre her iki veri kümesi aynı başarı performansına sahiptir. Gözlemlerimize göre en iyi başarıyı seyrek veri kümelerinde başarılı sonuçlar veren support vector machine yöntemi ve karar ağaçları yöntemi vermiştir. Bahsedilen diğer sınıflandırma yöntemlerinin sınıflandırma başarıları %95 in üzerindedir. LDA ise filtrelenmemiş veri kümelerinde sınıflandıramamıştır. Bunun sebebi ise LDA, C sınıflı bir veri kümesinde özellik uzayını (C-1) boyuta düşürmesidir [Alpaydın, 2001]. Çalıştığımız veri kümesi iki sınıflı olduğundan LDA özellik uzayını tek sınıfa düşürmektedir. Filtrelenmemiş veri kümesi çakışan birçok alt parça ve gürültü içerdiğinden, filtrelenmemiş veri kümelerinde sınıflandırma için 1 boyutlu uzay yeterli değildir. Ayrıca veri filtrelendiğinde LDA ile başarılı bir şekilde sınıflandırma yapılabilmektedir.

Bu sonuçların gösterdiğine göre filtrelenmiş veri, orijinal veri ile aynı sonuçlar vermektedir. Ve filtreleme işlemi önemli ve gerekli özelliklerin filtrelenmesine sebep olmamaktadır. Verinin filtrelenmesi ile sınıflandırma işleminin işlem süresi 2.7 kat artmaktadır. Baştan sona kadar toplam işlem süresi, görsel veri madenciliği için gereken işlem süresi ve dört sınıflandırma işleminin toplam işlem süresi de dahil 14 dakikadır. Görsel veri madenciliğinin işlem süresi ise aynı veri kümesi ve aynı Δ parametreleri için

2 dakikadır. Bu zamanın büyük kısmını sınıflandırma yöntemleri almaktadır. Yapılan çalışma [Yılmaz et al., 2008] da ayrıntılı olarak bahsedilmiştir.

Çizelge 7.1, Filtrelenmemiş ve önerilen görsel veri madenciliği yöntemleri ile filtrelenmiş Anti-tüberküloz veri kümesinin sınıflandırma yöntemlerinin performansına göre karşılaştırılması.

	Filtrelenmemiş Veri		Filtrelenmiş Veri (Önerilen Yöntem Uygulanmış)	
	Aktif Moleküller	Aktif Olmayan Moleküller	Aktif Moleküller	Aktif Olmayan Moleküller
Uygulanan Sınıflandırma Yöntemleri	Başarı (%)	Başarı (%)	Başarı (%)	Başarı (%)
Decision Tree	100	100	100	100
Linear Discriminant Analysis	0	100	92	95
1-Nearest Neighbor	100	100	100	95
Support Vector Machines	100	100	100	100
Ortalama İşlem Zamanı	39 dk. 19 sn.		14 dk. 29 sn	

7.2 Örüntü Haritalarından Filtreleme Eşik Değerinin Çıkarılması

Histogram tabanlı örüntü haritalarında filtreleme eşik değerini kullanıcı belirlemektedir (Bkz. şekil 7.1). Kullanıcının işini kolaylaştırmak için filtreleme eşik değerini nokta yoğunluk fonksiyonları ile ortalama bir değer belirlenebilir. Gelecek çalışmalara ışık tutması açısından bu konu ile ilgili yapılan araştırma aşağıda sunulmuştur.

Yoğunluk fonksiyonları parametrik ve parametrik olmayan olmak üzere ikiye ayrılmaktadır [Alpaydın, 2001]. Eğer verinin belli bir fonksiyonel yapısı var ise diğer bir deyişle en az hata ile veri kümesi bilinen fonksiyonlar (ör: lineer bir doğru, Gauss eğrisi) ve uygun parametreler (ör: μ, σ) ile modellenilebiliyor ise sadece parametreleri tutarak veri modellenilebilir. Fakat birçok örnekte olduğu gibi verinin nokta yoğunluk dağılımının yapısı bilinmiyor ise parametrik olmayan yoğunluk fonksiyonları kullanılmaktadır. Bunun için yoğunluk fonksiyonu olarak literatürde sıkça kullanılan çekirdek yoğunluk tahmin fonksiyonunu kde (kernel density estimation) ve histogramlar kullanılabilir. Kde fonksiyonu bir çekirdek fonksiyonunu veri üzerinde kaydırarak o alandaki yoğunluk değerini hesaplar. Tez çalışmasında kullanılacak çekirdek fonksiyonu

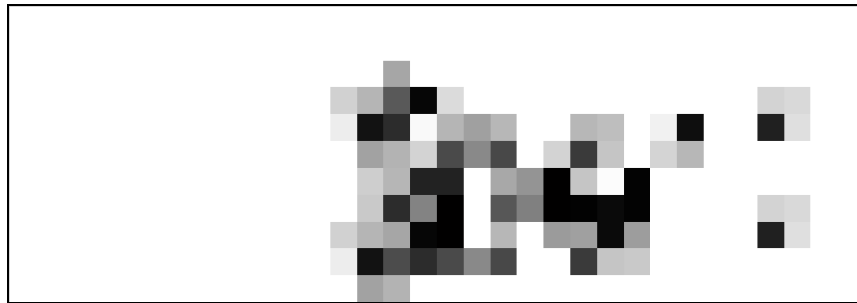
olarak Gauss fonksiyonunu önerilmektedir. Yoğunluk fonksiyonunun denklemi aşağıda görülmektedir.

$$F(x) = \frac{1}{n} \sum_{i=1}^n \frac{K(x - x(i))}{h} \quad (4.7)$$

Ve çekirdek fonksiyonunun denklemi aşağıda tanımlanmıştır

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right) \quad (4.8)$$

Şekil 7.2 de örüntü haritasındaki 255-0 (Beyaz - Siyah) arasındaki gri renk tonlarının karşı geldiği 0-100 arası yüzde değerlerinin bir boyutlu olarak nokta dağılımı görülmektedir. (Histogram tabanlı örüntü haritalarındaki renk tonu o aralığa düşen örnek çizge yüzdesini göstermektedir.) Haritadaki beyaz tonların çok yoğun olması diğer renklerin yoğunluklarının çok az görünmesine sebep olduğundan beyaz rengi renk çizelgesinden kaldırılır (Şekil 7.4).



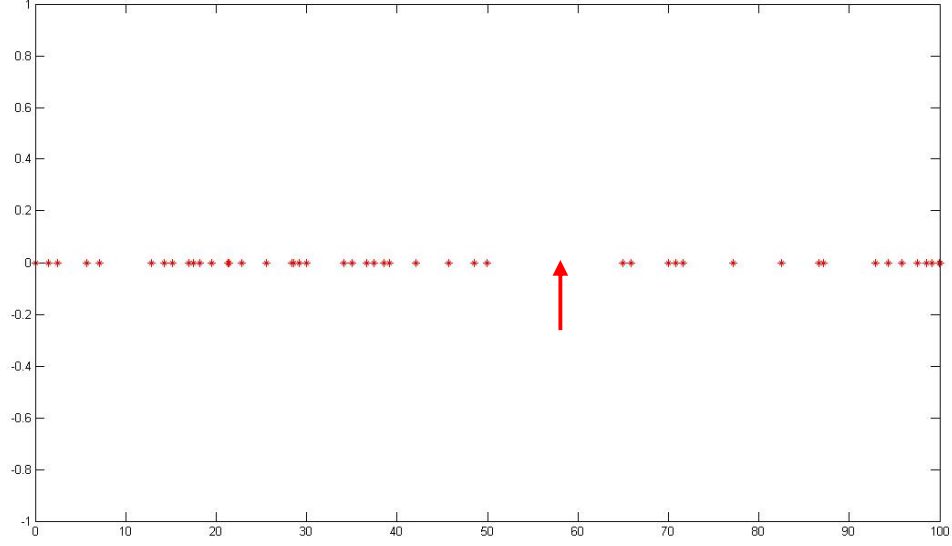
Şekil 7.1, Tek bir sınıfa ait örnekler için bir örüntü haritası örneği.

Şekil 7.3 ve şekil 7.4 da renklerin histogram gösterimine dayanan yoğunluk dağılımı görülmektedir. Şekil 7.5 ve şekil 7.6 da ise çekirdek tabanlı yoğunluk tahmin fonksiyonu kullanılarak nokta yoğunluğu bir fonksiyon ile modellenmiştir ($h=8.4578$). Fonksiyonun genel en küçük değeri ile bölgesel en küçük değeri filtreleme eşik değeri olmaya adaydır. Fonksiyonun bölgesel olarak en düşük (local minimum) değeri x^* aşağıdaki eşitsizliği sağlar.

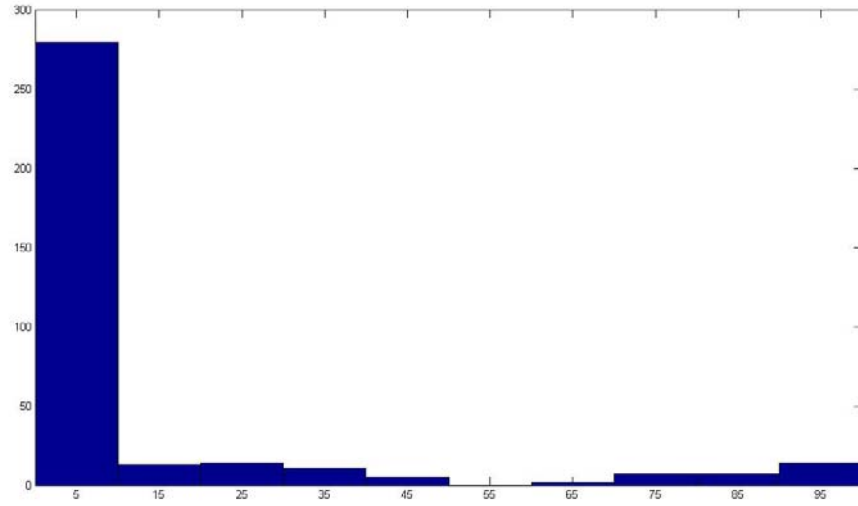
$$|x - x^*| < \varepsilon \text{ olduğunda } f(x^*) \leq f(x) \quad (4.9)$$

Fonksiyonun genel en küçük değeri ise aşağıdaki şekildedir.

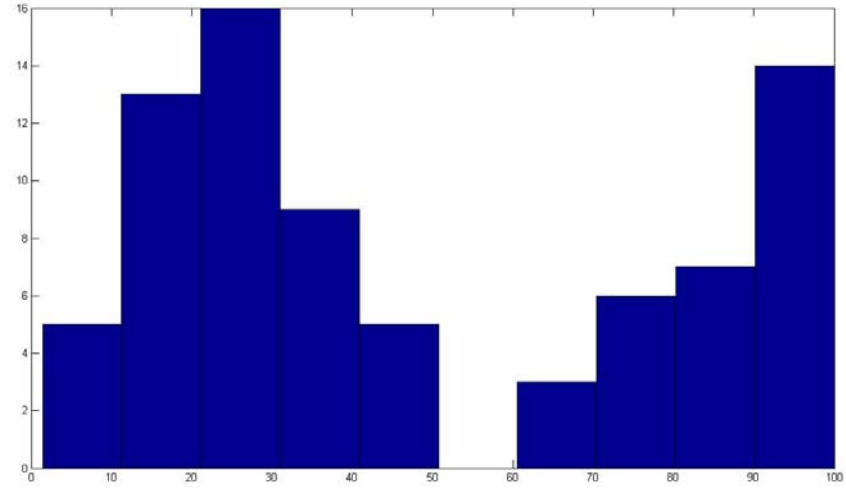
$$\forall x, \exists x^*, f(x^*) \leq f(x) \quad (4.10)$$



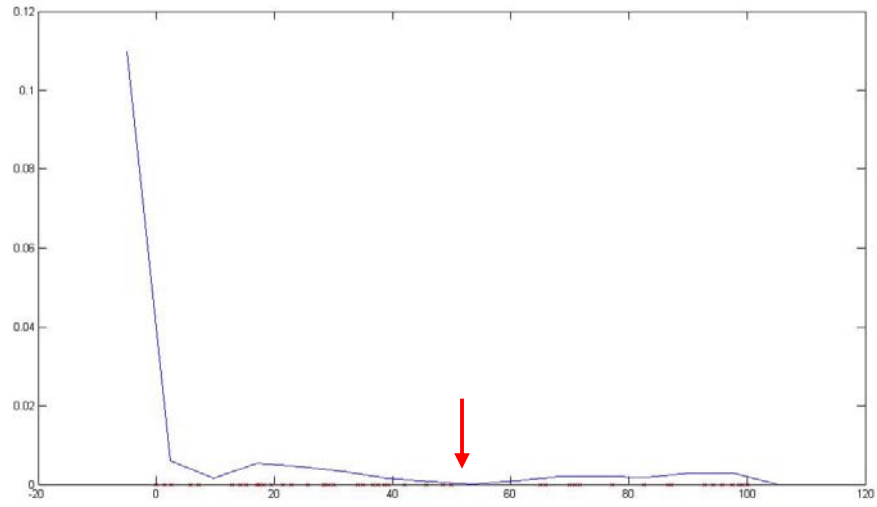
Şekil 7.2, Tek bir sınıfa ait örnekler içeren örüntü haritasındaki renklerin 1 boyutlu nokta dağılımı olarak gösterilmesi.



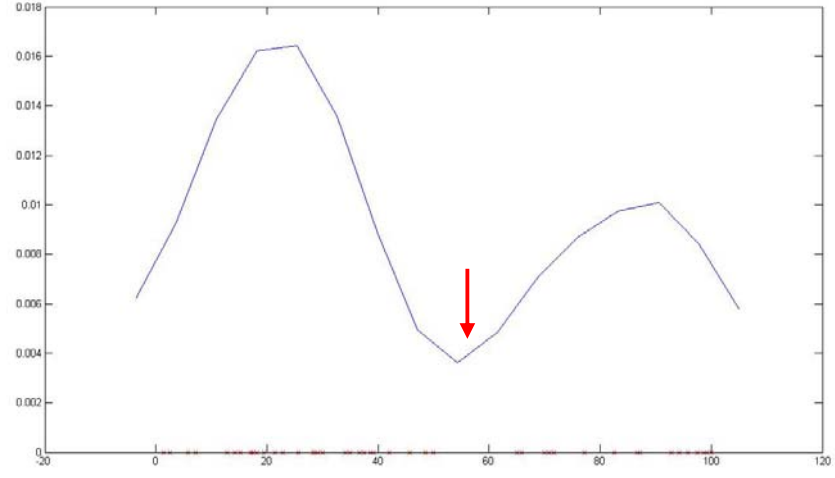
Şekil 7.3, Tek bir sınıfa ait örnekler içeren örüntü haritasındaki renklerin 1 boyutlu histogram ile gösterilmesi.



Şekil 7.4, Tek bir sınıfa ait örnekler içeren örüntü haritasındaki renklerin 1 boyutlu histogram ile gösterilmesi.



Şekil 7.5, Tek bir sınıfa ait örnekler içeren örüntü haritasındaki renklerin yoğunluğunun kde fonksiyonu ile gösterilmesi.



Şekil 7.6, Tek bir sınıfa ait örnekler içeren örüntü haritasındaki renklerin yoğunluğunun kde fonksiyonu ile gösterilmesi.

KAYNAKLAR DİZİNİ

Alpaydın, E. (2001). Introduction to Machine Learning. 1th ed., MIT Press.

An, A., Wang, Y. (2001), “Comparison of Classification Methods for Screening Potential Compounds”, *Proceedings of the IEEE Conference on Data Mining*, 11.

Agrawal, R., Imielinski, T., Swami, A. N. (1993), “Mining association rules between sets of items in large databases”. *Proceedings of ACM SIGMOD Management of Data*, Vol. 22(2), 207–216.

Backer, E., Jain, A. (1981), “A Clustering Performance Measure Based On Fuzzy Set Decomposition”. *IEEE Trans. Pattern Anal. Mach. Intell.* Vol(1). PAMI-3, 66–75.

Baker, H. (2003). Computer Graphics with OpenGL. Prentice Hall, Third Edition.

Bengoetxea E. (2002), “Inexact Graph Matching Using Estimation of Distribution Algorithms”, *Ecole Nationale Sup'erieure des T'el'ecomunications* (PhD thesis), December 2002.

Bersuker, I.B., Dimoglo, A.S. (1991), “The Electron- Topological Approach to the QSAR Problem”, *Reviews in Computational Chemistry*. VCH, Vol. 2, 423-460.

Blondel, V. D., Gajardo, A., Heymans, M., P. Senellart, Dooren, P. V. (2004), “A Measure Of Similarity Between Graph Vertices”, *Applications to Synonym Extraction and Web Searching*, Vol. 46(4), 647–666.

Borgelt, C., Berthold, M.R. (2002), "Mining Molecular Fragments: Finding Relevant Substructures of Molecules". *Proceedings of the Conference on Data Mining, ICDM*, 51–58.

Bringmann, B., Nijssen, S. (2008), "What is Frequent in a Single Graph?", *Lecture Notes in Computer Science*, Vol. 5012, 858-863.

Chakrabarti, D., Faloutsos, C. (2006), "Graph Mining: Laws, Generators, and Algorithms", *ACM Computing Surveys*, Vol. 38(1).

Coffman, T., Greenblatt, S., Marcus, S. (2004), "Graph-based technologies for intelligence analysis", *Communications of the ACM*, Vol 47(3), 45-47.

Cook, J., Holder, L. (1994), "Substructure discovery using minimum description length and background knowledge", *J. Artificial Intel. Research*, Vol. 1, 231-255.

Kubinyi H. (1997), "QSAR and 3D QSAR in Drug Design Part 1: Methodology", *Drug Discovery Today*, Vol. 2(11), 457-467.

Deshpande, M., Kuramochi, M., Wale, N., Karypis, G. (2005), "Frequent Substructure-Based Approaches for Classifying Chemical Compounds", *IEEE Transactions on Knowledge and Data Engineering*, Vol.17(8), 35.

Dimoglo, A. S. (1985), "Compositional Approach to Electronic Structure Description of Chemical Compounds, Oriented on Computer Analysis of Structure - Activity Relationships", *Chem. Pharm. Zh.*, Vol. 19, 438-444

Elçin, E., Rollas, O. S., Kandemirli, F., Shvets, N., Dimoglo, A. S. (2004), "1,3,4-Thiadiazole Derivatives. Synthesis, Structure Elucidation, and Structure-Antituberculosis Activity Relationship Investigation", *J. Med. Chem.*, Vol. 47, 6760-6767.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. (1996), "From Data Mining to Knowledge Discovery: An Overview". *Advances in knowledge discovery and data mining*, 1-34.

Ferri, J. F., Villanueva, W. D., Castro, M. J. (2007), "A Comparative Study Using Different Topological Representations in Pattern Recognition Based Drug Activity Characterization", *Proceedings of the IEEE Conference on System, Man and Cybernetics*, 96-101.

Fischer, I., Meinl, T. (2004), "Graph Based Molecular Data Mining – An Overview", *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, 4578-4582.

Fujita, T., et al, (1984), "The role of QSAR in Drug Design", Academic Press.

Good, A., Mason, J., Pickett, S. (2000), "Pharmacophore Pattern Application in Virtual Screening, Library Design and QSAR", *Virtual Screening for Bioactive Molecules*, Wiley-VCH Verlag: Weinheim, 131-159.

Han, J., Cheng, H., Xin, D., Yan, X. (2007), "Frequent pattern mining: current status and future directions", *Data Mining and Knowledge Discovery*. Vol. 15(1), 55-86.

Hinneburg, A., Keim, D. A., Wawryniuk, M. (1999), "HD-Eye: Visual Mining of High-Dimensional Data", *IEEE Computer Graphics*, Vol. 19(5), 22-31.

Holder L.B., Cook D.J. (1993), "Discovery of Inexact Concepts from Structural Data", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5(6), 992-994.

Huan, J., Wang, W., Prins, J. (2003), "Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism", *Proceeding of the IEEE Conferance on Data Mining*. 549–552.

Jamali, M., Abolhassan, H. (2006), "Different Aspects of Social Network Analysis", *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, 66-72.

Keogh, E., Lonardi, S., Ratanamahatana, C. A. (2004), "Towards Parameter-Free Data Mining", *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 206-215.

Kuramochi, M., Karypis, G. (2001), "Frequent subgraph discovery", *Proceedings of the IEEE Conference on Data Mining*, 313-320.

Kuramochi, M., Karypis, G. (2005), "Finding Frequent Patterns in a Large Sparse Graph", *Data Mining and Knowledge Discovery*, Vol. 11(3), 243-271.

Lahiri, M., Berger-Wolf, T. Y. (2007), "Structure Prediction in Temporal Networks using Frequent Subgraphs", *IEEE Symposium on Computational Intelligence and Data Mining*, CIDM, 35-42.

Larose, D. T. (2005), *Discovering Knowledge in Data*. WILEY.

Leach, A. R. (2001), *Molecular Modelling. Principles and Applications*. 2nd Ed. Harlow: Prentice Hall.

Lee, H. Y., Ong, K. L. (1996), "Visualization Support for Data Mining", *IEEE Expert: Intelligent Systems and Their Applications*, Vol. 11(5), 69-75.

Leskovec, J., Singh, A., Kleinberg, J. (2006), "Patterns of influence in a recommendation network". *Pacific-Asia Conference on Knowledge Discovery and Data Mining, Lecture Notes in Computer Science*, 380-389.

Mukherjee, M., Holder, L. B. (2004), “Graph-based Data Mining on Social Networks”, *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 22-25.

Murtagh, F. (1983), “Survey Of Recent Advances In Hierarchical Clustering Algorithms”, *The Computer Journal*, Vol:26(4), 354- 359.

Nijssen, S., Kok, J. N. (2004), “Frequent Graph Mining and its Application to Molecular Databases”. *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, 4571 – 4577.

Noble, C. C., Cook, D.J. (2003), “Graph-Based Anomaly Detection”, *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, 631-636.

Oliveira, M. C. F, Levkowitz, H. (2003), “From Visual Data Exploration to Visual Data Mining: A Survey”, *IEEE Transactions on Visualization and Computer Graphics*. Vol. 9(3).

Phua, C., Lee, V., Smith, K., Gayler, R. (2005), “A Comprehensive Survey of Data Mining-based Fraud Detection Research”, *Artificial Intelligence Review*.

Ramakrishnan, C., Milnor, W. H., Perry, M., Sheth, A. P. (2005), “Discovering Informative Connection Subgraphs in Multirelational Graphs”, *ACM SIGKDD Explorations Newsletter*, Vol. 7(2), 56-63.

Xu, R., Wunsch, D. (2005), “Survey of Clustering Algorithms”, *IEEE Transactions on Neural Networks*, Vol.16(3).

Sand P. And Teller S. (2004), “Video Matching”, *ACM Transactions on Graphics*, Vol. 23(3), 592-599.

Shvets, N., Yılmaz, B. (2005), “Pattern Recognition in Drug Design Through Image Processing”, *Proceedings of the IEEE Conference on Computer as a Tool*, Vol. 2, 1413 – 1416.

Shvets, N. M. (1993), “Applied Program System for the Prognosis of Biological Activity of Chemical Compounds: Development and Use”, *Computer Journal of Moldova*, Vol. 1, 101-110.

Shvets, N. M., Dimoglo, A. S. (1999), “The Electron-topological method (ETM): Its further development and use in the problems of SAR study”, *Molecular modeling and prediction of bioactivity*, 418-429.

Silberschatz, A., Tuzhilin, A. (1995), “On Subjective Measures of Interestingness in Knowledge Discovery”, *Proceeding of the Knowledge Discovery of Data Mining*, 275-281.

Stenberg, M. J. E., Muggleton, S. H. (2003), “Structure Activity Relationships (SAR) and Pharmacophore Discovery Using Inductive Logic Programming (ILP)”, *QSAR Comb. Sci.*, 22.

Sivic J., Zisserman A. (2003), “Video Google: A Text Retrieval Approach to Object Matching in Videos”, *Proceedings of the IEEE conference on Computer Vision*, Vol. 2.

Todd, M. J., Yıldırım, E. A. (2007), “On Khachiyan’s Algorithm For The Computation Of Minimum Volume Enclosing Ellipsoids”, *Discrete Applied Mathematics*, Vol. 155, 1731–1744.

Tong, H., Faloutsos, C. (2006), “Center-Piece Subgraphs: Problem Definition and Fast Solutions”, *Proceedings of the ACM SIGKDD*, 404-413.

Veselovsky, A. V., Ivanov, A. S. (2003), “Strategy of Computer-Aided Drug Design”. *Curr Drug Targets Infect Disord.*, Vol. 3, 33-40.

Washio, T., Motoda, H. (2003), “State of the Art of Graph based Data Mining”, *ACM SIGKDD Explorations Newsletter*, Vol. 5(1), 59-68.

Worlein, M., Meinl, T., Fischer, I., Philippsen, M. (2005), “A Quantitative Comparison of the Subgraph Miners MoFa, gSpan, FFSM, and Gaston”, *Lecture Notes in Artificial Intelligence*, Vol. 3721, 392-403.

Yan, X., Han, J. (2002), “Graph-Based Substructure Pattern Mining”, *Proceedings of IEEE Conference on Data Mining, ICDM*, 721–723.

Yang, G. (2006), “Computational Aspects of Mining Maximal Frequent Patterns”, *Theoretical Computer Science*, Vol. 362(1-3), 63-85.

Yilmaz, B., Gokturk, M. (2009a), “Discovering Similar Frequent Fragments in Drug Design: A Clustering-Based Approach”, *Transactions on Mass-Data Analysis of Images and Signals*, Vol: 1(1), 48-61.

Yilmaz, B., Gokturk, M. (2009b), “Interactive Data Mining for Molecular Graphs”, *Journal of Automated Methods & Management in Chemistry*.

Yilmaz, B., Gokturk, M., Svets, N. (2008), “User Assisted Substructure Extraction in Molecular Data Mining”. *Proceedings of Mass Data Analysis of Signals and Image in Medicine, Biotechnology, Chemistry, and Food Industries, Lecture Notes in Computer Science*, Vol. 5108, 12-26.

ÖZGEÇMİŞ

1980 yılında İstanbul’da doğdu. İlk, orta ve lise tahsilini İstanbul’da tamamladı. 2002 yılında İstanbul Üniversitesi Matematik bölümünden mezun oldu. 2004 yılında İstanbul Kültür Üniversitesi Matematik Bilgisayar bölümünde yüksek lisansını tamamladı.

2004-2008 tarihleri arasında İstanbul Kültür Üniversitesi Bilgisayar Mühendisliği bölümünde araştırma görevlisi olarak çalışmıştır.