T.C.
MALTEPE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

VEKİL SUNUCU VERİSİ ÜZERİNDE VERİ MADENCİLİĞİ İLE KULLANICI SORGULARI KÜMELEMESİ

MUSTAFA KORAY AYTEKİN

Yüksek Lisans Tezi

Tez Danışmanı
Yrd. Doç. Dr. Turgay Tugay Bilgin

İSTANBUL – 2012

T.C.
MALTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

VEKİL SUNUCU VERİSİ ÜZERİNDE VERİ MADENCİLİĞİ İLE KULLANICI SORGULARI KÜMELEMESİ

YÜKSEK LİSANS TEZİ

MUSTAFA KORAY AYTEKİN

Tez Danışmanı
Yrd. Doç. Dr. Turgay Tugay Bilgin

İSTANBUL – 2012

ÖZET

Yüksek Lisans Tezi, Vekil Sunucu Verisi Üzerinde Veri Madenciliği ile Kullanıcı Sorguları Kümelemesi , Maltepe Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.

Bu tez çalışmasında Maltepe Üniversitesi vekil sunucusu üzerinden derlenen günlük dosyası önişlemeye tabi tutularak bölümleme tabanlı ve graf tabanlı kümeleme algoritmaları ile kümelenmiş ve sonuçlar CLUSION adlı yöntem ile görselleştirilmiştir. Kullanıcıların arama motorlarında yaptıkları sorgular, günlük dosyasından önişleme ile elde edilmiş ve kümeleme amacı ile kullanılmıştır.

Toplam 5 bölümden oluşan tezin birinci bölümünde genel kavramlardan bahsedilmiştir. İkinci bölümde veri madenciliğinin genel tanımından, güncel sorunlarından, WWW ile olan ilişkisinden, üçüncü bölümde web madenciliği bileşenleri ve alt süreçlerinden, web madenciliğinde kullanılabilecek veri ve web madenciliğinin kullanım alanlarından bahsedilmiştir. Dördüncü bölümde benzerlik ölçümleri ile k-means ve graf tabanlı kümeleme algortimaları ele alınmıştır. Beşinci bölümde uygulamanın geliştirilme aşamaları, kullanılan araçlar ve geliştirme ortamı başlığı altında Zemberek doğal dil işleme kütüphanesi ve graf tabanlı kümeleme yöntemleri içeren Strehl küme analizi kütüphanesi anlatılmıştır. Bu bölümde ayrıca çalışmada kullanılan veri kümesi ve önişleme süreci açıklanmış, elde edilen sonuçlar irdelenmiştir.

Bu tez 2012 yılında tamamlanmıştır ve 97 sayfadan oluşmaktadır.

Anahtar Kelimeler: Web Kullanım Madenciliği, Veri Madenciliği, Kullanıcı Kümeleme, Vekil Sunucu Günlükleri, Arama Günlükleri Sorgusu

ABSTRACT

Master Thesis, Mining Proxy Log Data for Clustering User Queries, T.C. Maltepe University, Institute of Natural Sciences, Department of Computer Engineering.

In this master thesis, log files from web proxy server of Maltepe University have been preprocessed and clustered using partitioning and graph-based clustering algorithms. Results have been illustrated by using CLUSION algorithm. Queries performed by users on search engines have been compiled by processing proxy log files and are used for clustering.

General concepts about data mining have been presented in first section of the thesis which actually has 5 sections. In second section general definition of Data Mining has been given with contemporary problems in the field. In this section also the points which WWW and Data Mining have in common are mentioned. In third section web mining components and sub processes, data that can be used in web mining and usage of web mining in industry have been discussed. In fourth section similarity measures, k-means and graph based clustering have been presented to be basis in the following section. In fifth section which is the last one, the phases of the application have been discussed while Zemberek NLP library and Strehl cluster analysis library have been presented under tools and environments heading. In this section also the data set which has been used in the study and preprocessing task has been discussed and the results are investigated.

This thesis has been completed in 2012 and consists of 97 pages.

Keywords: Web Usage Mining, Data Mining, User Clustering, Proxy Logs, Query Search Logs

TEŞEKKÜR

Tez konusunu seçmemde beni yönlendiren, tez süreci boyunca destek ve yardımlarını esirgemeyen, değerli bilgilerinden istifade ettiğim danışman hocam Yrd. Doç. Dr. Turgay Tugay BİLGİN'e, tez sürecinde bana gösterdiği olağanüstü anlayış ve yardım için eşim Tuğba AYTEKİN'e, maddi ve manevi desteğini benden hiçbir zaman esirgemeyen çok değerli aileme ve çalışmalarım sırasında emeği geçen herkese teşekkürlerimi sunarım.

İÇİNDEKİLER

KISALTMALAR

Kısaltma	İngilizcesi	Türkçesi
BIRCH	Balanced Iterative Reducing and Clustering Using Hierarchies	Dengelenmiş Tekrarlı İndirgeme ve Hierarşiler Kullanarak Kümeleme
CLARA	Clustering LARge Applications	Büyük Veritabanlarında Kümeleme
DBSCAN	Density-Based Spatial Clustering of Applications with Noise	Gürültülü Veritabanlarında Yoğunluk Tabanlı Kümeleme
NLP	Natural Language Processing	Doğal Dil İşleme
OPOSSUM	Optimal Partitioning of Sparse Similarities Using Metis	Seyrek Benzerliklerin Metis ile En Uygun Bölümlenmesi

ŞEKİLLER

DENKLEMLER

ÇİZELGELER

1. GİRİŞ

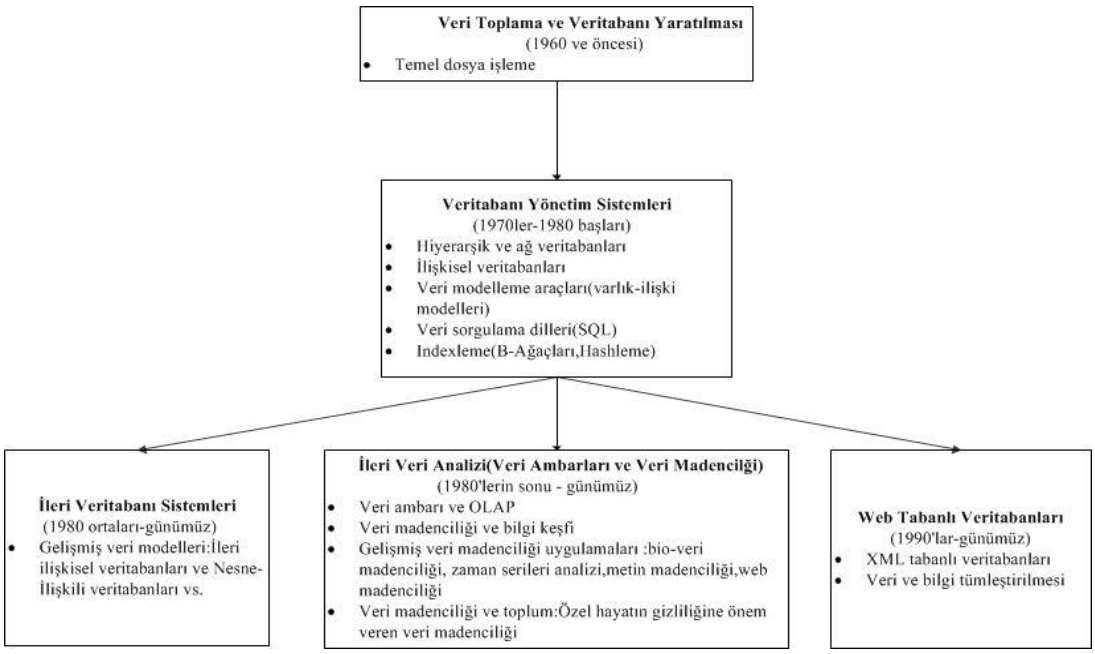
Bu tez çalışmasında graf tabanlı ve bölümlemeli kümeleme algoritmaları kullanılarak vekil sunucu günlüklerinden elde edilen kullanıcı sorguları kümelenmiştir. Veri madenciliğinin alt çalışma alanlarından olan web madenciliğinde kullanıcı davranışlarını analiz etmek yaygın uygulama alanlarından birisidir. Kullanıcı davranışları açısından ele alındığında genelde gezinme izleri analiz edilmektedir. Bu çalışma sonucunda benzer sorgular üreten kullanıcıların oluşturduğu kümelerin bulunması mümkün olacaktır.

Makine öğrenmesi alanında kümeleme algoritmaları gözetimsiz öğrenme teknikleri olarak bilinir ve veri yığınlarını, veri hakkında ön bilgiye sahip olmadan birbirine benzeyen nesnelerden oluşan kümelere ayırırlar. Veri nesnelerinin özelliklerine göre yapılan bu işlem özellik sayısı arttığında işlem zamanı olarak da artar. Bu çalışmada özellik uzayının daraltılarak benzerlik uzayında kümeleme yapan çalışması ile Prof. Dr. Alexander Strehl'in çalışması temel alınmıştır. Vekil sunucu günlüklerinde önişleme yapılarak kullanıcı sorguları metinsel olarak parçalanmış ve ortaya çıkartılan IP-Terim matrisi Strehl kütüphanesindeki fonksiyonlar aracılığı ile benzerlik matrisine dönüştürülmüştür. Kümeleme algoritmaları olan K-Means ve yine Strehl'in önerdiği graf tabanlı OPOSSUM, çeşitli benzerlik ölçüleri ile oluşturulan matris üzerinde çalıştırılmıştır. Benzerlik ölçüsü olarak Öklid metriği ve Jaccard gibi literatürde çokça geçen ölçüler dışında metin verileri üzerinde etkinliği bilinen Kosinüs Uzaklığı ve Pearson Korelasyon Katsayısı kullanılmıştır.

2. VERİ MADENCİLİĞİ

Çok büyük miktarlarda veri depolarının mevcut olması ve bu verinin kullanışlı bilgi haline dönüştürülme ihtiyacının kaçınılmaz olmasından dolayı son yıllarda bilgi teknolojileri ve bir bütün olarak veri madenciliği toplumda büyük ilgi uyandırmaktadır. Dönüştürülerek elde edilen bu bilginin kullanım alanları market analiz araştırmaları, sahtekârlık tespiti ve müşteri koruma uygulamalarından üretim kontrolü ve bilimsel keşif uygulamalarına kadar çeşitlilik gösterebilmektedir.

Veri madenciliği, bilgi teknolojisinin doğal evriminin bir sonucu olarak görülebilir. Veritabanı sistemleri endüstrisi, veri toplama ve veritabanı oluşturulması, veri yönetimi (veri depolama ve erişim ile veritabanı üzerindeki işlemlerin yönetimi) ve ileri veri analizi (veri ambarı ve veri madenciliği) fonksiyonlarının geliştirilmesinde evrimsel bir çizgi gözlemlemiştir [1].



Şekil 2.1 Veritabanı Sistemlerinin Evrimi [1]

Geçtiğimiz yüzyılın sonlarına doğru bilgisayar donanımı ve elektronik sektöründe meydana gelen düzenli teknolojik gelişme, ucuzlayan güçlü bilgisayar altyapısına erişimi kolaylaştırmıştır. Bu ucuz ve güçlü donanım aynı zamanda daha fazla veri depolama ve daha hızlı veri işleme anlamına gelmiştir. Gelişen bu teknoloji, bilgi işlem endüstrisine de ivme kazandırarak veri analizi, hızlı bilgi erişimi ve bilgi hareketlerinin yönetimi süreçlerini büyük miktarlarda veri depolayabilen donanım ürünleri ile destekleyerek daha fazla erişilebilir ve kullanılabilir olmasını sağlamıştır.

Yer sıkıntısı çekilmeden depolanabilen bol miktardaki veri, veri açısından bereketli ama güçlü veri analizi araçlarına olan ihtiyaç nedeniyle süreç içinde bilgi açısından verimsiz bir durum olarak değerlendirilmeye başlanmıştır. Bu verilerin depolandığı alanlar seyrek ziyaret edilen veri mezarlıkları haline gelmeye başlamış ve fark edilmiştir ki çok sayıdaki büyük veri alanlarında depolanan bu devasa miktardaki veriyi güçlü analiz araçları olmadan sadece insani yeteneklerimizle gerektiği gibi değerlendirmemiz mümkün olmamaktadır. Dolayısıyla ilgili kurumların vereceği önemli kararlar bu verilere dayanarak ortaya çıkarılabilecek değerli bilgiler ışığında değil bu verileri güçlü araçlarla değerlendirip sonuçlar üretecek insanların sezgisel yaklaşımları ile verilmiştir. Eğer bir de verinin biriktirildiği ortam el ile giriş yapılan veri bankası türü ortamlardansa, izlenilen bu yöntem nedeniyle taraflı bilgi oluşumu ve hataya açıklık nedeniyle aşırı miktarda zaman ve para kaybına neden olabilmiştir. Yine donanım ve yazılım altyapısının geliştiğinden bahsedilen bu süreçte veri artık daha çok sensörler ve izleme sistemleri (kredi kartı işlemleri, telefon sistemleri vs.) tarafından üretilmeye başlanmıştır [2].

Yukarıda bahsedilenler ışığında, veri madenciliği, depolanan bu büyük miktardaki verilerin bilgiye dönüştürülme sürecinde önemli yer tutar. Veri madenciliği araçları veri analizi yaparak yığınlar halindeki veri içinde bulunan örüntüleri ortaya çıkartıp veride işlenmeden önce açık olarak görünmeyen bilgiyi ortaya çıkartır. Veri ve bilgi arasındaki dönüşüme olan ihtiyacın büyümesi veri mezarlıklarını kullanışlı bilgi kaynakları haline getirecek veri madenciliği araçlarının sistematik bir şekilde geliştirilmesi ihtiyacını doğurmaktadır [1].

2.1. Veri Madenciliğinin Tanımı

Gartner Group isimli araştırma şirketine göre [3], "Veri madenciliği depolarda saklanan büyük miktarlardaki veriyi örüntü tanıma teknikleri, istatistiki ve matematiksel yöntemlerle ayıklayarak, anlamlı ve yeni korelasyonlar, örüntüler keşfetme sürecidir". Bir başka tanım ise şu şekildedir : "Veri madenciliği makine öğrenmesi, örüntü tanıma, istatistik, veritabanı yönetimi ve görselleştirme gibi disiplinlerden alınan teknikleri bir araya getirerek büyük veritabanlarından bilgi çıkarımı problemini hedefleyen disiplinler arası bir alandır." [4].

Birçok insan, veri madenciliği terimini yine popüler olan bir başka terim olan Veriden Bilgi Keşfi veya VBK ile eşanlamlı olarak kullanmaktadır. Bazıları ise veri madenciliğini bilgi keşif sürecinin önemli bir adımı olarak değerlendirmektedir. Bilgi keşif süreci aşağıda gösterildiği gibi birbirini izleyen adımlardan oluşmaktadır.

1. Veri Temizleme: Veri içindeki tutarsızlıkların ve gürültünün giderilmesidir.

2. Veri Entegrasyonu: Birden fazla veri kaynağı olması durumunda yapılır.

3. Veri Seçimi: Analiz işlemleri için gerekli ve uygun olan verinin veritabanından alınmasıdır.

4. Veri Dönüştürme: Verinin özetleme veya derleme işlemlerine tabi tutularak madenciliğe uygun hale getirilmesidir.

5. Veri Madenciliği: Veri örüntülerini ortaya çıkarmak için akıllı yöntemlerin uygulandığı önemli bir süreçtir.

6. Örüntü Değerlendirilmesi: Bilgiyi temsil eden ilginç örüntülerin özel ölçümlere dayanarak belirlenme işlemidir.

7. Bilgi Sunumu: Ortaya çıkarılan bilginin görselleştirme ve bilgi sunum yöntemleri kullanılarak kullanıcıya gösterilmesi adımıdır [1].

Bu süreç içinde 1. ve 4. adımlar arası veri önişleme olarak adlandırılmaktadır ve üzerinde madencilik yapılacak verinin temizlenmesi ve işleme hazır hale getirilmesi için gereken adımları içermektedir. Veri madenciliği adımı kullanıcı ile etkileşimli bir şekilde gerçekleştirilebilir. Bu adımdan sonraki örüntülerin değerlendirilmesi aşamasında ilgi alanı dışındaki örüntüler belirli ölçümlerle ayıklanıp önemli olanlar bir sonraki aşamada kullanıcıya değişik yollarla gösterilebilir. Yukarıdaki akışta da görülebileceği gibi veri madenciliği bilgi keşif sürecinin bir adımıdır. Ancak çok büyük veri kümelerinde standart yöntemlerle görülemeyecek bilgi ve örüntüleri ortaya çıkardığı için önemli bir adımdır. Veri madenciliği küçük veri kümeleri ile ilgilenmez.

2.2 Veri Madenciliği ile İlgili Sorunlar

Veri madenciliğine ait temel sorunlar, yöntem ve kullanıcı kaynaklı, performans kaynaklı ve farklı veri tipleri kaynaklı sorunlar başlıkları altında incelenebilir.

2.2.1 Yöntem ve Kullanıcı Kaynaklı Sorunlar

Madencilik sonucu elde edilmek istenen bilgi, çeşitli detay seviyelerinde bilgi elde etmek üzere veri madenciliği yapabilme yeteneği, alan bilgisini kullanabilme yeteneği, özel amaçlı veri madenciliği ve bilgi görselleştirme ile ilgili sorunlar bu kapsamda değerlendirilebilir.

- *Veri tabanlarında farklı çeşitte bilgi için madencilik yapmak*: Farklı kullanıcılar farklı tipte bilgilerle ilgilenebileceğinden, veri madenciliği, birliktelik ve korelasyon analizi, sınıflandırma, öngörü, kümeleme, ayrık değer analizi ve

gelişim analizi (eğilim ve benzerlik analizini de içerir) gibi geniş bir yelpazede veri analizi ve bilgi keşif yöntemleri içermelidir. Bu yöntemler aynı veritabanını farklı şekillerde kullanabilir ve pek çok veri madenciliği tekniğinin geliştirilmesine ihtiyaç duyabilirler.

- *Farklı soyutlama seviyelerinde etkileşimli veri madenciliği*: Bir veritabanında tam olarak ne keşfedileceğinin bilinmesi zor olduğundan veri madenciliği süreci etkileşimli yani insan müdahalesine izin verebiliyor olmalıdır. Büyük miktarlarda veri barındıran kaynaklarla uğraşıldığından kullanıcıya süreç içerisinde örneklemler üzerinde çalışma olanağı sağlanmalı, örüntüleri bulmaya odaklanan kullanıcıya elde edilen sonuçlara göre isteklerini düzenleyip gerekirse önişleme aşamasına dönüp eldeki verileri iyileştirmesine olanak tanınmalıdır. Bunun için ara sonuçların görselleştirilmesi ve kullanıcıya daha verimli etkileşim yöntemleri sağlanması gerekebilir. Böylece kullanıcı veri madenciliği sistemi ile etkileşim içine girebilip veriyi ve keşfedilen örüntüleri farklı detay seviyelerinde değerlendirebilecektir.

- *Veri Madenciliği sorgulama dilleri ve özel amaçlı veri madenciliği*: SQL gibi ilişkisel sorgulama dilleri kullanıcıların veri elde edebilmesi için özel amaçlı sorgular tasarlamasına olanak sağlar. Benzer şekilde kullanıcılara veri analizi için gerekli veri setlerini, üzerinde çalışılan konunun kapsamını, madencilik yapılacak bilgi türlerini ve keşfedilen örüntülerde bulunması zorunlu şartlar ve kısıtları belirtmelerini kolaylaştırarak özel amaçlı veri madenciliği işlemlerini tanımlayabilecekleri yüksek seviyeli veri madenciliği dillerinin geliştirilmesi gerekmektedir. Kullanıcı isteklerinin veri madenciliği problemine dönüştürülme ihtiyacı bulunmaktadır ve bunun için yüksek seviyeli bir dil geliştirilmelidir [5].

- *Veri madenciliği sonuçlarının sunumu ve görselleştirilmesi*: Veri madenciliğinin verimli olabilmesi için insanın veri keşif sürecine dâhil edilmesi ve insanın esneklik, yaratıcılık ve genel bilgisinin bilgisayarın depolama kapasitesi ve işlem gücüyle birleştirilmesi gerekir [2]. İnsanların görsel değerlendirme yeteneklerini büyük veri kümeleri üzerinde kullanabilmesi, ortaya çıkan sonuçların insanlar tarafından kolayca anlaşılabilmesi ve doğrudan kullanılabilmesi için keşfedilen bilgi yüksek seviyeli diller, görsel betimlemeler veya diğer tanımlayıcı şekillerde ifade edilmelidir. Bu, sistemin, ağaç yapıları, tablolar, kurallar, graflar, çizelgeler, matris ve eğriler gibi açıklayıcı bilgi betimleme tekniklerine sahip olmasını gerektirir.

- *Gürültülü ve eksik veri ile başa çıkmak*: Bir veritabanında saklanan veri gürültü, istisnai durumlar veya eksik veri nesneleri içerebilir. Veri madenciliği sırasında bu nesneler, oluşturulan bilgi modelinin veriye olmaması gereken şekilde uygunluk göstermesine neden olarak süreci yanıltabilir. Sonuç olarak keşfedilen örüntülerin doğruluğu zayıf hale gelebilir. Gürültü ile baş eden veri temizleme ve veri analizi yöntemlerinin yanı sıra istisnai durumların analiz ve keşfi için ayrık değerleri (outliers) ortaya çıkartan yöntemler de gerekebilir.

- *Örüntü değerlendirmesi*: Bir veri madenciliği sistemi binlerce örüntü ortaya çıkartabilir. Keşfedilen bu örüntülerin çoğu ya genel bilgiyi temsil ettiklerinden ya da herhangi bir yenilik içermediklerinden kullanıcıya ilginç gelmeyebilir. Verilen bir kullanıcı sınıfının, kullanıcı beklentileri ve inançlarına dayalı öznel örüntü değerleme kıstaslarından dolayı, keşfedilen örüntülerin ilginçliğini değerlendirmekle ilgili tekniklerin geliştirilmesine ait zorluklar halen bulunmaktadır. Bir kullanıcı ilginç olan tüm kurallar veya örüntüleri keşfetmek isteyecektir. Bir örüntü keşfinin ne kadar ilginç olduğuna dair konulacak ölçüye "ilginçlik" denir ve yeni veri ile test edildiğinde belirli bir "kesinlik" içinde "geçerlilik" gibi nicel ve nesnel

değerler temelinde değerlendirilebileceği gibi örüntünün "anlaşılabilirliği", örüntünün "yeniliği" veya "kullanışlılığı" gibi öznel açılardan da değerlendirilebilir [6].

2.2.2 Performans Kaynaklı Sorunlar

Bunlar veri madenciliği algoritmalarının etkinlik, ölçeklendirilebilirlik ve paralelleştirilebilirliği ile ilgili sorunları içerir.

- *Veri madenciliği algoritmalarının verimlilik ve ölçeklendirilebilirliği*: Veritabanlarındaki büyük miktardaki veriden verimli şekilde bilgi çıkartabilmek için veri madenciliği algoritmaları etkin ve ölçeklendirilebilir olmalıdır. Diğer bir deyişle bir veri madenciliği algoritmasının büyük veritabanlarında çalışma zamanı tahmin edilebilir ve kabul edilebilir olmalıdır. Birçok uygulama var olan bilgi keşif araçlarının analiz edebildiğinden daha fazla veriyi üretmekte veya edinmekte ve bu da analiz edilmeden sürekli arşivlenmeye neden olmaktadır. Dahası çok büyük miktarlardaki veri kümelerini analiz etmek bazen mevcut bilgisayarların işlem gücünü geçebilmektedir [7].

- *Paralel, dağıtık ve artımlı madencilik algoritmaları*: Çoğu veritabanının büyük boyutlu olması, verinin geniş bir dağılıma sahip olması ve bazı veri madenciliği yöntemlerinin hesaplama karmaşıklığı, paralel ve dağıtık veri madenciliği algoritmalarını teşvik eden en önemli etkenlerdir. Bu tip algoritmalar veriyi bölümlere ayırarak paralel olarak işler. Bu bölümlerden elde edilen sonuçlar daha sonra birleştirilir. Dahası bazı veri madenciliği süreçlerinin yüksek maliyeti, veritabanına yapılan güncellemeleri kullanarak tüm veritabanını baştan ele almak zorunda bırakmayan artımlı veri madenciliği algoritmalarına olan ihtiyacı ortaya çıkarmaktadır.

2.2.3 Farklı Veritabanı Tiplerinden Kaynaklanan Sorunlar

- *Karmaşık ve ilişkisel tipte veri ile başa çıkmak*: Büyük miktarlarda veri denilince ilk akla gelen ilişkisel veritabanları ve veri ambarlarıdır. Her ne kadar geniş çapta kullanıldıklarından bu tip veriler için verimli ve etkin veri madenciliği sistemleri geliştirmek önemli ise de büyük miktarlarda hiper-metin, görsel ve ses içerikli veri (çoklu ortam), uzamsal, zamansal (temporal) verileri içeren veri kaynakları da bulunmaktadır. Bu karmaşık veri nesneleri de en az klasik ilişkisel veritabanları ve veri ambarları kadar veri madenciliği ilgi alanına girmektedir. Verinin bu çeşitliliği ve veri madenciliğinin farklı hedefleri göz önüne alındığında bir sistemin tüm veri tipleri üzerinde madencilik yapmasını beklemek çok gerçekçi değildir. Belirli tipte veri üzerinde madencilik yapabilmek için belirli ve özellikli veri madenciliği sistemleri oluşturulmalıdır. Bu yüzden farklı veri tipleri için farklı veri madenciliği sistemlerinin kullanılması beklenilmelidir.

- *Heterojen veritabanlarında ve küresel bilgi sistemlerinde veri madenciliği*: Yerel ve geniş alan bilgisayar ağları (Internet gibi) çok fazla bilgi kaynağını birbirine bağlayarak devasa boyutlarda dağıtık ve türdeş olmayan (heterojen) veritabanları oluştururlar. Bu heterojen ve dağıtık veri kaynaklarının yapılanmış, yarı yapılanmış veya yapılanmamış gibi farklı yapılanma seviyelerinde olması, farklı semantik yapılara sahip verilerden oluşmaları veri madenciliğine zorluklar çıkartmaktadır. Veri madenciliği, birden çok türdeş olmayan veri kaynağı üzerinde bulunan ve basit sorgulama sistemleri ile keşfedilmesi kolay olmayan yüksek seviyeli veri düzenliliklerini ortaya çıkarmaya yardımcı olabilir.

2.3 Veri Madenciliği ve WWW

Dünya Çapında Ağ (World Wide Web - WWW) geçen yüzyılın son on yılında hayatımıza girmesiyle birlikte günümüzde hemen herkesin günlük etkileşim aracı ve bilgi kaynağı haline gelmiştir. Etkileşim sonucu veri üretmekte ve kişisel olarak ürettiğimizden daha çok veriye ulaşmaktayız. Artık WWW, eğitim, kamu yönetimi, e-ticaret, finansal işlemler, haberler, sosyal medya vs. gibi birçok bilgi hizmeti barındıran büyük, aşırı dağıtık küresel bir bilgi kaynağı haline gelmiştir. Kısaca Web olarak adlandırılan bu devasa yapı içerdiği ve her saniye etkileşim sonucu oluşan veri miktarı ile veri madenciliği için zengin ve dinamik bir kaynaktır.

WWW, içlerinde günlük ziyaretçi sayısı on milyonlarla ifade edilen web siteleri başta olmak üzere değişik bilgi servisleri barındırır. Hakkında bilgi sahibi olmak için herhangi bir konu hakkında arama yapan kullanıcılar bütün bu web siteleri arasında hiper-bağlantı adı verilen bağlantılar aracılığı ile gezinirler. Bu gezinme işlemi başlı başına bir veri kaynağını besleyebilecek miktarda veri üretir. Bu açıdan web, veri madenciliği açısından bol miktarda olanak barındırmaktadır. Kullanıcıların bu gezintileri sırasında ortaya koydukları erişim örüntülerini anlayarak sık ziyaret edilen/erişilen veri nesneleri (web sayfaları, indirilebilir dosyalar vs.) arasındaki bağlantıyı verimli hale getirmek, reklam yerleştirme, sayfalar arası veya sayfa içi tasarım gibi pazarlama kararlarında yardımcı olmak ve müşterileri sınıflandırarak kişiselleştirme vs. gibi alanlarda yardımcı olur. Web madenciliğinin bu tip dağıtık ortamlarda kullanıcı erişim örüntülerini yakalamaya çalışan alt sınıfına Web Kullanım Madenciliği denmektedir.

Bununla birlikte aşağıdaki açıklamalarda de görülebileceği gibi, Web, etkin bilgi ve kaynak keşfi açısından büyük zorluklar da taşımaktadır.

- *Web veri madenciliği uygulamaları açısından devasa büyüklüktedir.* Web'in boyutları binlerce terabayt mertebesindedir ve halen hızlı bir şekilde büyümektedir. Internet trafiği son 5 yılda 20 kat artmıştır, web üzerinde 170

milyondan fazla web sunucusu olduğu tahmin edilmekte ve 2015 yılında 5 milyar kişinin internete bağlantısının olacağı düşünülmektedir [8]. Web'in kişisel kullanım ile büyümesinin yanı sıra faydalarını ve kolaylıklarını fark eden küçük büyük birçok organizasyon, kuruluş ve hatta kamu kuruluşları sahip oldukları veri kaynaklarının tamamını veya bir kısmını web ortamına taşımaktadır. Bununla birlikte internette bulunan bu veriyi tek bir bütün olarak birleştirmek ve ele almak (bir veri ambarı gibi) mümkün görünmemektedir. Bu yüzden doğası gereği dağıtık ve heterojen kalmak zorundadır.

- *Web sayfaları geleneksel metinden daha karmaşıktır. Web* sayfaları kitap ve diğer metin tabanlı geleneksel dokümanlara göre hem yapısal hem de içerik olarak daha farklıdır. Web sayfaları yapısal olarak daha kuralcı, içerik olarak daha çeşitlidir. Web sayfaları içeriklerinin düzenlenmesi anlamında daha kuralcı olmakla birlikte bir kütüphanedeki kitaplar gibi belirli kıstaslara göre düzenlenmemiş ve indise sahip değildir. Bu özelliği ile web aranan bilginin çok zor bulunabileceği bir kütüphane gibidir. Ayrıca web sayfaları belirli kurallara göre tanımlanıyor olsa da istisnai olarak bozuk yapılanmış olabilir veya önceden tanımlanmış bir şema ya da desene sahip olmayabilirler. Bu da geleneksel metin analizinden farklı olarak bilgisayarların bu tip web sayfalarındaki anlamsallığı çözmelerini zorlaştırıp sistematik veri edinme ve web madenciliği açısından sorunlu hale getirmektedir

- *Web çok dinamik bir bilgi kaynağıdır.* Web içerdiği veri anlamında yalnızca genişlememekte, kapsadığı bu verinin bir kısmı sürekli güncellenmektedir. Haberler, borsa bilgileri, reklamlar, hava durumu, alışveriş dünyasına ilişkin bilgiler en bilinenleridir. Değişen bu içeriğe ek olarak eklenen/yok olan sayfa ve sitelerin bağlantı bilgileri ve erişim kayıtları da sıklıkla değişmekte ve güncellenmektedir.

- *Web'deki bilginin sadece küçük bir bölümü gerçekten gerekli veya yararlıdır*. Web'deki bilginin %99'u, Web kullanıcılarının %99'u için gereksiz olduğu söylenir [1]. Bunun rakamsal doğruluğu kesin olmasa da çoğumuz için

web ihtiyacımız olanın dışında ve fazlasıyla veri içermektedir. Burada kullanıcı için anlamlı ve kaliteli bilginin devasa web içeriğinden nasıl çekip çıkartılacağı problemi bulunmaktadır. Web sitelerinin ya da daha özel olarak web sayfalarının arkasındaki kapsamı anlamaksızın yapılan kelime tabanlı arama servisleri aslında kullanıcılara sınırlı oranda yardımcı olabilmektedir. Hepimizin hemen her gün karşılaştığı şekilde yapılan bir arama ile arama motorları bize ilgilendiğimiz konu ile ilgili ilişkisiz veri bağlantıları da getirmektedir. Gelenlerin bir kısmı da aradığımızla zayıf ilişki içindedir. Veri madenciliği bu gibi durumlarda web arama hizmetlerine yardımcı olabilecek potansiyele sahip görünmektedir. Örneğin web sayfaları arasındaki bağlantılardan yararlanarak güvenilir web sayfası analizi, sahip oldukları önem, etki ve konulara göre derecelendirme yapılabilir.

Web ortamında yapılan veri madenciliği, verimli web analizi ve veri madenciliği yöntemlerinin geliştirilmesi için çaba harcamaktadır. Genel olarak bakıldığında veri madenciliği, internetteki dağıtık veri hakkında bilgi sahibi olmamızda, farklı web sayfaları, kullanıcıları ve web tabanlı hareketlerin birbirleri olan ilişkilerini anlamamızda bize yardımcı olabilecek bir disiplin olarak görünmektedir.

3. WEB MADENCİLİĞİ

Web, bir bütün olarak tamamıyla kontrol dışı çok çeşitli veri yapılarının oluşturduğu dev bir veri kaynağıdır. Dolayısıyla büyüklük, çok çeşitlilik, değişkenlik ve ölçeklenebilirlik gibi sorunları barındırır. Bu özelliklerinden dolayı devasa bir veri havuzunda yüzüyor olmamıza rağmen sınırlı ve zor elde edilebilir bilgi ile karşı karşıya gibiyizdir. Bu büyük miktardaki kullanıma hazır veri, veri madenciliği çalışmaları için aslında çok bereketli bir alandır.

Web madenciliği genel olarak kullanışlı veya işe yarar bilginin WWW üzerindeki veri yığınları içinde keşfi ve analizi olarak tanımlanabilir. Web madenciliğinde kullanılacak bu veri ise sunuculardan, istemcilerden, vekil sunuculardan veya kurumsal veritabanlarından toplanabilir. Verinin tipi, toplandığı bu kaynaklara ve içeriğine göre (metin, ses, görüntü vs.) değişkenlik taşır. Bahsedilen verilerdeki bu değişkenlik de üzerlerinde çalışacak web madenciliği uygulamalarının belirli bir yelpazede değişmesine neden olur. Web üzerindeki verilerin sahip oldukları bazı özellikler şu şekilde özetlenebilir [9]:

- etiketsizdir (tanımlayıcı bilgisi yoktur),
- dağıtıktır,
- çok çeşitlidir,
- yapısal farklılıklar gösterir,
- zamana göre değişkendir,
- çok boyutludur.

Veri madenciliğinde başlıca üç çeşit çalışma alanı bulunmaktadır: Veri madenciliği, Web madenciliği ve Metin madenciliği [10]. Üzerinde madencilik yapılan veri yapısal olarak değerlendirilirse, yapılanmamıştan yapılanmışa doğru bir gelişim gösterir. Veri madenciliği daha çok veritabanları ve veri ambarları gibi düzgün yapılanmış veri üzerinde çalışırken metin madenciliği, verinin kaynağının bütünü değil de alt parçaları olarak değerlendirildiğinde yapılanmamış, belirli kurallarla şekillendirilemeyen metin verisi üzerinde çalışır. Web madenciliği ise bunların arasında yarı yapılanmış veri üzerinde çalışır diyebiliriz. Web madenciliği bu açıdan bakıldığında hem veri madenciliğinin hem de metin madenciliğinin kendi disiplinlerine özgü yaklaşımlardan yararlanabilir.

WWW, günümüzde en yaygın bilgi paylaşma ve bilgi edinme ortamıdır. Tüm dünya üzerindeki kullanımından dolayı internetteki veri o kadar hızlı büyümekte, çeşitlenmekte ve güncellenmektedir ki bu veriyi kullanmak isteyenler aşağıdaki sorunlarla karşı karşıya kalmaktadırlar [10].

1. *İhtiyaç Duyulan Bilgiyi Bulma* – İnsanlar web üzerinde arama yaparken ya gezinirler (browsing) ya da çeşitli arama servislerini (arama motorları) kullanırlar. Bununla birlikte kullanıcının gezinti ile arama yapacağı veri uzayı sınırlıdır. Arama servislerinin ise döndürdükleri sonuçların çoğunun gereksiz ve aranılan bilgi ile karşılaştırıldığında düşük duyarlılık gibi sorunları bulunmaktadır.

2. *Web'de bulunan veriden bilgi yaratabilmek* – Özünde bu problem yukarıdakinin alt problemidir. Yukarıdaki problem sorgu ile tetiklenen belirli bir kıstasa göre arama yapıp elde etmeye dayalı bir süreçtir. Yeni bilgi yaratabilme problemi ise bir yığın web verisine sahip olduğunuzu ve bu veriden anlamlı, yeni ve ilginç bilgi çıkartıp kullanmak istediğinizi varsayan, veri tarafından tetiklenen bir süreçtir.

3. *Verinin kişiselleştirilmesi* - İnsanlar web ile etkileşim içine girdiklerinde tercih ettikleri içerik ve sunum açısından farklılaşırlar.

4. *Tüketiciler veya bireysel kullanıcılar hakkında bilgi edinmek* – Buradaki problem müşteri veya kullanıcının ne yaptığı ve ne istediğinin uygulayıcılar tarafından bilinmek istenmesidir. Kullanıcı davranışını öğrenme ihtiyacı ise web sitesinin tasarım ve çalışmasının, kullanıcı tercih ve kullanım şekilleri için optimize edilme gerekliliğinden kaynaklanmaktadır. Bu pazarlama ve site yönetimini kolaylaştırma açısından önemlidir. Son aşaması bireysel kullanıcı için yapılan kişiselleştirme yani tek bir kullanıcı için bile sitenin uyarlanmasıdır.

Web madenciliği terimini ilk kullanan 1996 yılında Etzioni'dir [13]. Etzioni, web üzerindeki verinin yeteri kadar yapılanmış olduğu hipotezini ortaya atmakla başlar ve web madenciliğinin alt çalışma süreçlerini belirler. Oren Etzioni'ye göre Web Madenciliği, WWW üzerindeki doküman ve servislerden otomatik olarak bilgi çıkartmak ve keşfetmek için veri madenciliği tekniklerinin kullanımıdır. Kosala ve Blockeel [11] ve Qingyu Zhang ve Richard s. Segall [12] Web madenciliğinin aşağıdaki alt çalışma süreçlerine bölünmesini önerir:

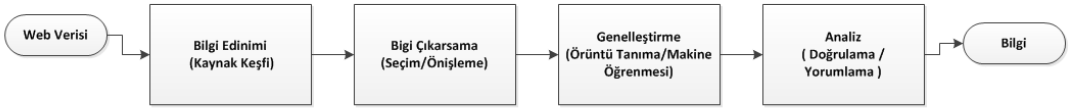
- *Kaynak Keşfi*: Web üzerindeki alışılmadık doküman ve servislerin bulunmasıdır.

- *Veri seçimi ve Önişleme*: Yeni keşfedilen Web kaynaklarından belirli verilerin otomatik olarak çıkartılıp önişlemeye tabi tutulmasıdır.

- *Genelleştirme*: Tekil web sitelerindeki ve birden çok web sitesindeki genel ya da ortak örüntülerin ortaya çıkartılmasıdır.

- *Analiz:* Ortaya çıkan örüntülerin doğrulanması ve yorumlanmasıdır.

- *Görselleştirme*: Etkileşimli bir analizin sonuçlarını görsel ve daha kolay anlaşılabilir şekilde sunmak.



Şekil 3.1 Web Madenciliği Alt Süreçleri [9].

3.1 Veri Madenciliği Bileşenleri ve Alt Süreçleri

Etzioni'ye göre web madenciliği Şekil 3.1'de de görülebildiği gibi dört süreçten oluşur [13]. Her süreç aşağıda kapsadıkları araç ve yöntemlerle birlikte tanımlanmıştır.

3.1.1 Bilgi Edinimi (Kaynak Keşfi)

Kaynak keşfi veya BE, ilgisiz dokümanların sayısını en azda tutmaya çalışarak konuyla ilgili olanların otomatik elde edilmesi ile uğraşır. BE süreci esas olarak doküman temsili, indekslenmesi ve doküman aranması işlemlerini kapsar.

Bir indeks basit olarak aranılan bilginin nerede olduğunu işaretçiler şeklinde tutan terimler kümesidir. Sürekli büyüyen dahası sürekli güncellenen web ortamında bu şekilde bir yapıyı hem oluşturmak hem de güncel tutmak zordur. Çeşitli indeksleme yöntemleri olmakla birlikte en çok bilinen ve kullanılanı arama motorları için yazılmış web robotu olarak adlandırılan indeksleyicilerdir. Bu indeksleyiciler

Google, AltaVista, Bing gibi arama motorları için web üzerinde milyonlarca veri kaynağını sürekli tarayıp dokümanlar üzerindeki kelimelerin indekslerini saklarlar.

3.1.2 Bilgi Seçme/Çıkartma ve Önişleme

Bilgi edinimi sürecini izleyen bu adımdaki zorluk, insan etkileşimi olmadan ihtiyaç duyulan bilgiyi, dokümanın anlamsal içeriğini oluşturan bölümlerini tespit ederek ortaya çıkartabilmektir. Bunun için yine özelleşmiş yazılımlardan yararlanılır. Bu yazılımlardan bir türünün çalışma yöntemi web sitesine ait dokümanların gerekli metin bölümlerinin ortaya çıkartılacak şekilde işlenmesine dayanır ve bu işlemi her siteye özgün tasarlanmış bir yazılım gerçekleştirir [14]. Hiper metinlerden bilgi çıkarımı için kullanılan diğer bir yöntem ise [15]'de verildiği gibi her bir sayfanın bir küme standart soru ile sorgulanmasıdır. Böylece problem, bu özelleşmiş sorulara cevap verebilen metin bölümlerinin tespit edilmesine indirgenmiş olur. Anlaşılacağı üzere "Bilgi Çıkarımı (BÇ)" elde edilen veri kaynağının yapısından ve sunum şeklinden yararlanarak içeriğindeki veriden yeni bir bilgi elde edebilir miyim diye araştırırken "Bilgi Edinimi" açısından doküman ya da diğer deyişle veri kaynağı bir kelimeler topluluğudur. BÇ açısından ölçeklendirilebilirlik önemli bir problemdir çünkü Web'in boyutu ve devingenliğine yetişebilecek bir BÇ sistemi oluşturmak mümkün değildir. Bunun yerine belirli sitelere ve bu siteler üzerindeki belirli alanlara odaklanılır.

3.1.3 Genelleştirme

Bu aşamada ortaya çıkartılan bilgi üzerinde genellikle örüntü tanıma ve makine öğrenmesi yöntemleri kullanılır. Web madenciliği süreçlerindeki en önemli problemlerden birisi etiketleme problemidir. Veri boldur ama sınıflamaya yarayacak tanım veya kimlik bilgisine sahip değildir. Veri madenciliğinin sınıflandırma teknikleri işledikleri verinin belirli bir kavrama göre en azından pozitif veya negatif

örneklerden oluşan giriş bilgisine ihtiyaç duyarlar. Örneğin bir web sayfasının "ana sayfa" olup olmadığına göre sınıflandıracak bir sınıflandırıcı tasarlamayı istersek önceden elimizde bu şekilde bir sınıflandırmaya tabi tutulmuş ve pozitif veya negatif olarak etiketlenmiş sayfa sınıflarına ihtiyacı ortaya çıkar. Bu tip bir problemle karşılaşıldığında kullanılabilecek yöntemlerden birisi webin etkileşimli bir ortam olduğundan hareketle kullanıcılardan bazı giriş değerleri alarak ilgili sayfanın hangi sınıfa ait olduğunu tespit etmeye dayanır [16]. Burada yazarlar kullanıcının adı ve kişisel bazı bilgilerini kullanıcıdan alarak aranılan kişinin "ana sayfasını" bulmaya çalışmışlar ve bu süreç içinde taranılan veri kaynaklarından gelen bilgileri belirli etiketlerle sınıflandırarak sonuç sınıflarını oluşturmuşlardır [17]. Sınıflandırıcıların aksine kümeleme yöntemleri giriş değerlerine ihtiyaç duymaz ve web madenciliğinde yaygın bir şekilde kullanılmaktadırlar. Bu aşamada kullanılan önemli yöntemlerden bir diğeri de birliktelik kuralları madenciliğidir. Temel olarak birliktelik kuralları, X ve Y öğelerden oluşan kümeler olmak üzere X => Y tipindeki ifadelerdir. X => Y ifadesi, bir T işlemi X'i içeriyorsa büyük ihtimalle Y'yi de içerir şeklinde açıklanabilir. Olasılık veya kurala olan güven, X ve ek olarak Y'yi içeren tüm işlemlerin yüzdesinin X'i içeren tüm işlemlerle karşılaştırılması olarak tanımlanabilir. Birliktelik kurallarının madenciliği fikri "x1 ve x2 ürünlerinden alan bir müşteri %c olasılıkla y ürününden de alacak" şeklinde kuralları bulunan alışveriş tabanlı veriler üzerinde yapılan araştırmalardan kaynaklanmaktadır [18].

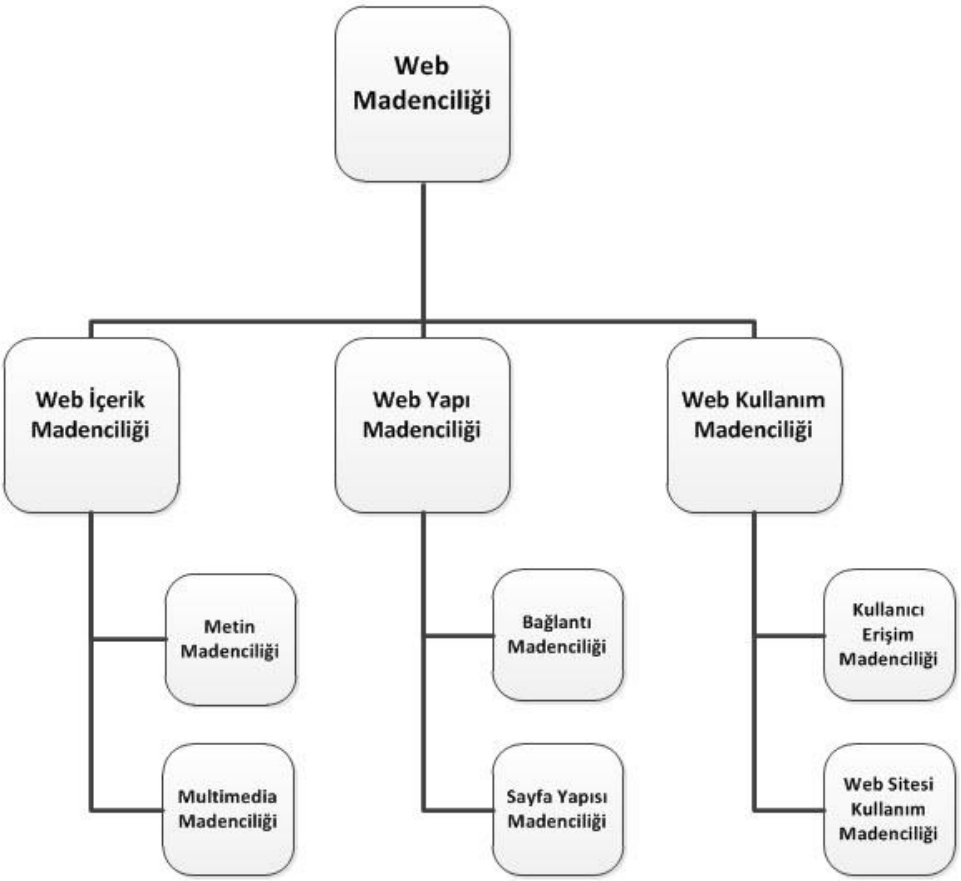
3.1.4 Analiz

Örüntü analizi Web madenciliğinin son aşamasıdır ve burada amaç bulunan örüntüler ve kurallar arasından ilginç olmayanların temizlenmesidir. En bilinen yöntemler SQL gibi bilgi sorgulama mekanizmaları kullanmak ya da eldeki kullanım verilerini OLAP işlemleri uygulamak üzere veri küplerine yerleştirmektir. Grafik çizimler, farklı değer kümelerine renkler atama şeklindeki görselleştirme teknikleri bu aşamada veri içindeki örüntüleri ve eğilimleri belirlemede yardımcı olur. En doğru analiz yöntemi web madenciliği ile hedeflenen amaca uygun olarak belirlenir [19].

Şekil 3.1'de gösterilen ve yukarda bahsedilen dört aşamaya dayanarak, *web madenciliğinin, veri madenciliği teknikleri kullanarak web dokümanları ve servislerinden bilgi keşfi için otomatik veri toplama, işlenecek veriyi ortaya çıkartma ve değerlendirme aşamalarını içerdiği söylenebilir. Burada değerlendirme hem genelleştirme hem de analizi içerir.*

3.2 Web Madenciliği Türleri

Kosala ve Blockeel [12], madenciliği yapılacak verinin tipine göre, bilgi için madencilik, bağlantı yapısı üzerinde madencilik ve kullanıcı gezinme örüntüleri için madencilik olmak üzere üç çeşit web madenciliği kategorisi önerirler. Bilgi için madencilik yani içerik madenciliği belirli bir kıstası karşılayacak şekilde bir kullanıcıya aradığı dokümanları bulmada yardım etmek için gereken tekniklerin geliştirilmesine odaklanır. Web içerik madenciliği, metin, görüntü, ses, video vs. dâhil olmak üzere web içeriğinden kullanışlı bilginin keşfine dayanır. Web yapı madenciliği ise bağlantılar şeklindeki web yapısının altında yatan model üzerinde çalışır. Keşfedip incelemeye çalıştığı bu model tanımsal olmaktan daha çok topolojik özellikleri açısından ele alınır. Farklı web siteleri arasındaki benzerlik ve ilişkileri ortaya çıkarmak açısından önemlidir. Web kullanım madenciliği ise kullanıcıların web üzerinde yaptıkları hareketler sonucu oluşan verinin üzerinde çalışan teknikleri kapsar. Kullanıcıların web üzerindeki hareketleri web sunucusu erişim günlükleri, vekil sunucu günlükleri, tarayıcı günlükleri, kullanıcı profilleri, kütük verileri, kullanıcı oturum ve işlemleri, çerezler, kullanıcı sorguları, yer imleri ve fare tıklamaları gibi kaynaklardan elde edilir.



Şekil 3.2 Web Madenciliği Sınıflandırması [20].

3.2.1 Web İçerik Madenciliği

Basit tanımıyla web sayfalarından kullanışlı bilgi keşfetme sürecidir. Margaret H. Dunham [21], Web İçerik Madenciliğini, basit arama motorlarının yaptığı işin genişletilmiş halidir şeklinde tarif etmiştir. Web içerik madenciliği, ses, görüntü ve video gibi çoklu ortam (multimedia) dâhil olmak üzere web kaynaklarının içeriğini analiz eder. Birincil kaynak ise web sayfalarıdır.

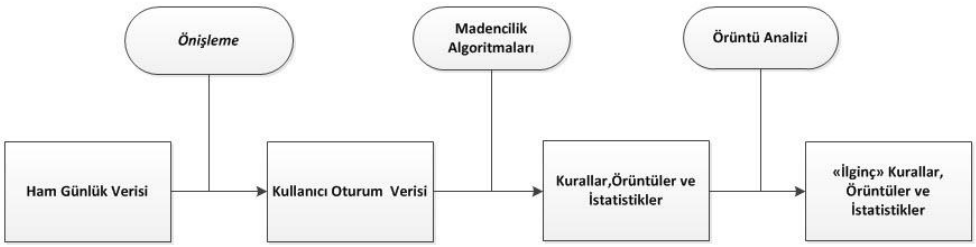
3.2.2 Web Yapı Madenciliği

Web üzerindeki bağlantı yapıları ve bunların modellemesi ile uğraşır. Web üzerindeki içerik bilgisinin yanı sıra webde bulunan kaynakların birbirleri ile olan bağlantısı da değerli bilgi içerir. Web yapı madenciliği ile veri kaynakları arasındaki bağlantılardan yararlanarak sayfa ve site değerlemesi gibi verimlilik ve kullanışlılık analizleri yapılmaktadır. Web yapı madenciliğinin ana odak noktası bağlantı bilgisidir.

3.2.3 Web Kullanım Madenciliği

Web kullanım madenciliği, kullanıcıların web ile etkileşimleri sonucu oluşan verinin üzerinde yapılan madenciliktir. Web kullanım madenciliği kaynakları; web günlükleri, kullanıcıların kayıt ve sorgu bilgileri, ticari aktivitelerin içeriği ve web hizmetleri ile ilişki veritabanlarıdır. Bunlar arasında web günlük madenciliği en önemli parçayı oluşturur. Web günlükleri olarak adlandırılan günlükler Web sunucusu üzerinde bulunan erişim günlükleri, yazılım günlükleri, hata günlükleri vs. gibi günlüklerin genel adıdır. Bu dosyalar, kullanıcı IP adresi, ziyaret edilen URL, erişim zamanı ve tarihi, ziyaret sonuçları(başarılı, başarısız, hatalı), erişim yöntemi (GET, POST) gibi kullanıcı erişim bilgilerini saklar. Web sitesi yöneticisi bu tip

bilgileri kullanarak, örneğin her kullanıcının gezinme davranışlarını kullanıcının erişim örüntülerinden ortaya çıkartarak, kullanıcılara kişiselleştirilmiş hizmetler sunma yöntemiyle hizmet kalitesini arttırabilir [20]. Şekil 3.2, Şekil 3.1'in web kullanım madenciliği için özelleşmiş halidir ve web kullanım madenciliği yapabilmek için gerçekleştirilmesi gereken temel olarak üç adım bulunmaktadır.



Şekil 3.3 Web Kullanım Madenciliği Süreci [19].

3.2.3.1 Önişleme

Önişleme, mevcut çeşitli veri kaynaklarında bulunan kullanım, içerik ve yapı bilgisinin, keşif süreci için gerekli veri yapılarına dönüştürülmesini kapsar. Kullanım verisinin önişleme adımı, mevcut ham verinin bütünlüğündeki sorunlar nedeniyle web kullanım madenciliği sürecindeki en zor adımdır.

İstemci tarafında bir izleme mekanizması kullanılmadığı sürece kullanıcıları ve sunucu oturumlarını belirleyebilmek için sadece IP adresi, bağlantıyı sağlayan program (örn. Tarayıcı) ve sunucu taraflı tıklama-akışı bilgisi kullanılabilir. Sunucu ve kullanıcı belirlemede karşılaşılan tipik problemlerden bazıları şu şekildedir [19] :

- Tek IP adresi / Çoklu Sunucu Oturumu - Internet Hizmet Sağlayıcıları tipik olarak kullanıcıların internete eriştikleri bir vekil sunucu havuzuna sahiptirler.

Tek bir vekil sunucu bir web sitesine aynı zaman aralığında erişen birçok kullanıcıya sahip olabilir.

- Çoklu IP adresi / Tek Sunucu Oturumu – Bazı servis sağlayıcılar veya gizlilik araçları bir kullanıcıdan gelen her isteği farklı IP adreslerine atarlar. Bu durumda tek bir sunucu oturumu birden fazla IP adresine sahip olabilir.

- Çoklu IP adresi / Tek kullanıcı – Farklı makinelerden internete erişen bir kullanıcı oturumdan oturuma fark eden IP adreslerine sahip olacaktır. Bu da aynı kullanıcıdan gelen tekrarlanan ziyaretlerin takibini zorlaştıracaktır.

- Çoklu Bağlantı Aracı / Tek kullanıcı – Yine burada aynı makine üzerinde bile olsa birden çok tarayıcı kullanan tek bir kullanıcı, birden çok kullanıcı gibi görünecektir.

Her kullanıcının belirlenebildiğini varsayarsak (çerezler ve sistem girişleri ile veya IP / bağlantı aracı / gezinme yolu analizi ile) bu sefer kullanıcıların oturum bilgilerini oluşturmak için her kullanıcının tıklama akışı oturumlara göre bölümlendirilmelidir. Normal olarak diğer sunucularda saklanan sayfa istekleri erişilebilir olmadığından bir kullanıcını bir web sitesini ne zaman terk ettiğini bilmek zordur. Bu yüzden bir kullanıcının tıklama akışını oturumlara bölmek için otuz dakikalık zaman aşımları varsayılan yöntem olarak kullanılmaktadır. Bu şekilde kullanıcının hangi zaman aralığında aktif oturuma sahip olduğu anlaşılabilir Oturum numarası URI içine gömülerek oturumun tanımlanması içerik sunucusuna bırakılır.

Her kullanıcı hareketi sonucu sunulan içerik, sunucu günlüklerindeki istek alanında mevcuttur. İçerik sunucuları, aktif her oturuma ait durum değişkenlerini saklayabildiğinden, bir kullanıcı isteğine karşılık hangi içeriğin tam olarak sunulduğu bilgisi URI içine her zaman konulmaz. Kullanım verisinin önişlemesi sırasında karşılaşılan son problem önbellekte tutulan sayfa referanslarının ortaya çıkartılmasıdır. Önbellekte tutulan sayfa izleme bilgilerinin doğrulanabilir tek takip yöntemi istemci tarafındaki kullanımı gözlemlemektir.

Şekil 3.3 günlük dosyası örneği içermektedir. Burada gizlilik nedeniyle IP adresleri değiştirilmiş olup satırlardaki 1.2.3.4, 3.4.5.6 sıralı sayıları IP adreslerini göstermektedir. 1 numaralı satırda 1.2.3.4 IP adresine sahip kullanıcı, "maya.cs. depaul.edu" sunucusu üzerindeki "/classes/cs589/papers.html" dosyasına erişmektedir. İstemci, "http://dataminingresources.blogspot.com/" adresinden gelmekte ve tarayıcısının Mozilla işletim sisteminin Windows olduğu ilgili satırda görülmektedir [22].

Çizelge 3.1 Örnek Web Sunucusu Günlüğü [22].

1	2006-02-01 00:08:43 1.2.3.4 - GET /classes/cs589/papers.html - 200 9221 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1;+.NET+CLR+2.0.50727) http://dataminingresources.blogspot.com/
2	2006-02-01 00:08:46 1.2.3.4 - GET /classes/cs589/papers/cms-tai.pdf - 200 4096 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1;+.NET+CLR+2.0.50727) http://maya.cs.depaul.edu/~classes/cs589/papers.html
3	2006-02-01 08:01:28 2.3.4.5 - GET /classes/ds575/papers/hyperlink.pdf - 200 318814 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1) http://www.google.com/search?hl=en&lr=&q=hyperlink+analysis+for+the+web+survey
4	2006-02-02 19:34:45 3.4.5.6 - GET /classes/cs480/announce.html - 200 3794 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1) http://maya.cs.depaul.edu/~classes/cs480/
5	2006-02-02 19:34:45 3.4.5.6 - GET /classes/cs480/styles2.css - 200 1636 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1) http://maya.cs.depaul.edu/~classes/cs480/announce.html
6	2006-02-02 19:34:45 3.4.5.6 - GET /classes/cs480/header.gif - 200 6027 HTTP/1.1 maya.cs.depaul.edu Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1) http://maya.cs.depaul.edu/~classes/cs480/announce.html

3.2.3.2 Örüntü Keşfi

Örüntü keşfi, istatistik, makine öğrenmesi, veri madenciliği ve örüntü tanıma gibi çeşitli alanlarda geliştirilen yöntem ve algoritmalardan yararlanır. Bu bölümde Web madenciliğinde uygulanagelen yöntemlerden bahsedilecektir. Web madenciliği

kapsamı nedeniyle bu yöntemlerin özelleşmiş veya uyarlanmış şekillerini kullanmak zorundadır. Örneğin veri madenciliğinde bilinen bir yöntem olan birliktelik kurallarının market sepeti uygulaması sepete atılan ürünlerin dolayısıyla kasa işlemlerinin sırasını önemsemez fakat web kullanım madenciliği açısından bakıldığında bir sunucu oturumu kullanıcının sunucudan istediği sıralı sayfalar dizisidir. Oturum tespitinin de ayrıca zorluklar içerdiğinden yukarda bahsedilmişti.

- **İstatistikî Analiz**: İstatistikî yöntemler bir web sitesi ziyaretçileri hakkında bilgi edinmek için kullanılan en genel yöntemdir. Oturum dosyasını analiz ederek sayfa görüntülenmeleri, görüntülenme zamanı ve gezinti yolu uzunluğu gibi değişkenler üzerinde tanımlayıcı istatistikî analizlerin (frekans, ortalama, medyan vs.) farklı türlerini gerçekleştirebilir. Birçok Web trafik analizi araçları, sıkça erişilen sayfalar, bir sayfanın ortalama görüntülenme süresi, bir sitedeki ortalama gezinti yolu uzunluğu gibi istatistikî bilgilerin yer aldığı periyodik raporlar üretir. Bu tip raporlar yetkisiz erişim noktaları veya hatalı URI gibi sınırlı miktarda ve aşağı seviyeli hata analiz bilgileri içerebilir. Dolayısıyla bu araçlar tarafından sunulan sonuçlar gizli kalmış eğilimleri ve açıkça görülemeyen kullanım bilgisini anlamaya yardımcı olma konusunda sınırlı yeteneğe sahiptir [23]. Bazı yeni ürünler daha yetkin ve karmaşık analiz yetenekleri sunabilmekte ama insan müdahalesi gerektirmekte ve web günlüklerinin büyük boyutundan dolayı sıklıkla örneklem üzerinde çalışma temeline dayanmaktadır [24]. Analiz derinliği açısından sınırlı olsa da bu tip araçların sunduğu bilgiler sistem performansının geliştirilmesi, sistem güvenliğinin arttırılması, web sitesinin değiştirilmesi ve pazarlama kararlarının verilmesinde destek olma açısından potansiyel olarak çok faydalı olabilir.

- **Birliktelik Kuralları**: Birliktelik kuralları üretimi aynı sunucu oturumunda erişilmiş sayfaları ilişkilendirmek için kullanılabilir. Örnek olarak Apriori algoritması veya benzerleri kullanılarak bir alışveriş sitesi üzerinde elektronik ürünler ile spor ürünleri sayfasını ziyaret eden kullanıcılar arasında bir korelasyon olup olmadığını anlamak mümkün olabilir [25]. Bu sayede site tasarımcıları birliktelik kurallarının olup olmadığına bakarak tasarımları

üzerinde iyileştirmeler/değişiklikler yapabilir, kullanıcı tarafında hissedilen sayfa yüklemeleri kaynaklı gecikmeleri azaltmak amacıyla sezgisel yöntemler ile önbellek mekanizmaları oluşturabilirler.

- **Kümeleme**: Kümeleme, nesneleri, benzerlik niteliğine göre aynı küme içine yerleştirmeyi amaçlayan gözetimsiz/müdahalesiz bir sınıflandırma yöntemidir [26]. Web kullanım madenciliğinde keşfedilebilecek özel öneme sahip iki çeşit küme vardır: Kullanıcı kümeleri ve sayfa kümeleri. Kullanıcı kümelerinin ortaya çıkartılması ile benzer gezinti tarzına sahip kullanıcılar kümelenerek aynı ilgi alanlarına sahip kullanıcılar belirlenmiş olur. Bu sayede verilen hizmetler pazarlama katmanlarına ayrılıp özelleştirilebilir. Sayfa kümelerinin ortaya çıkartılması ile benzer içeriğe veya ilgilenilme derecesine sahip sayfa grupları ortaya çıkartılmış olur ki bu da arama motorları gibi web hizmeti sunan uygulama ve kuruluşlara hizmet kalitesini arttırmak için olanak sağlar.

- **Sınıflandırma**: Sınıflandırma bir veri parçasının daha önceden belirlenmiş sınıflardan birisi ile eşleştirilmesidir [27]. Kümeleme gibi sınıflandırma da makine öğrenmesi ve veri madenciliğinin klasik yöntemlerindendir. Kümelemede olduğu gibi niteliklerin sayısı, çeşitliliği ve tek şekilli olmaması sınıflandırma yöntemlerinin hipermetinler üzerinde uygulanmasını zorlaştırmaktadır [28]. Sınıflandırma, karar ağacı sınıflandırıcıları, sade Bayes sınıflandırıcıları, K-en yakın komşu sınıflandırıcıları, Destek Vektör Makineleri vs. gibi gözetimsiz tümevarımsal öğrenme algoritmaları ile yapılabilir. Bu sınıflandırıcılar kullanılarak kullanıcı profilleri oluşturmaya çalışmak sınıflandırıcı uygulamalarına örnek olabilir. Örneğin sunucu günlüklerinde yapılan sınıflandırma şu şekilde ilginç kuralların keşfine yol açabilir: /Ürün/Elektronik kategorisinde sipariş veren kullanıcıların %30'u 18-25 yaş grubundadır ve İstanbul Avrupa yakasında yaşıyor.

- **Sıralı Örüntüler:** Sıralı örüntüleri keşfetme problemi zamana göre sıralanmış bir işlem kümesi içinde bir grup nesnenin bir başka nesne tarafından takip edilip edilmediğinin araştırılması diğer bir deyişle işlemler arası örüntü arama

faaliyetidir [29]. Web sunucuları ziyaretçilerin erişim zamanlarını kaydederler. Kullanıcıların bu ziyaret zamanlarının takibi ve bu zaman aralıkları arasındaki ilişkinin ortaya çıkartılması web sitesi yöneticilerine ve geliştiricilerine ziyaretçinin ziyaret zamanlarını tahmin etme ve buna göre reklam uygulamalarını hazırlama olanağı sağlar [30].

- **Bağımlılık Modelleme**: Bağımlılık modelleme web madenciliğindeki diğer bir faydalı örüntü keşif yöntemidir. Buradaki hedef web alanındaki çeşitli değişkenler arasında bulunan önemli bağımlılıkları temsil edebilen bir model geliştirmektir. Örnek olarak bir ziyaretçinin sanal bir alışveriş ortamında seçimlerine göre geçtiği farklı evreleri modellemek (örn. sıradan bir ziyaretçiden potansiyel ciddi bir alıcıya dönüşmesi) verilebilir. Kullanıcıların gezinme davranışlarını modellemek için uygulanabilecek çeşitli olasılıklı öğrenme yöntemleri mevcuttur. Bu teknikler Saklı Markov Modelleri ve Bayes İnanç Ağları'nı içerir. Web kullanım örüntülerini modellemek sadece kullanıcıların mevcut davranışını analiz etmek için teorik çerçeveyi sağlamakla kalmaz Web kaynaklarının gelecekteki tüketimini tahmin için de fayda sağlar. Bu tip bilgi web sitesi tarafından sunulan ürünlerin satışını arttırmak için stratejiler geliştirmeye yardımcı olabilir veya kullanıcıların gezinme deneyimlerini iyileştirebilir.

3.2.3.3 Örüntü Analizi

Örüntü Analizi Şekil.3.1'de tanımlanan tüm web kullanım madenciliği sürecinin son adımıdır. Örüntü analizinin arkasındaki gerekçe örüntü keşif sürecinde bulunan örüntüler arasındaki ilginç olmayan örüntü ve kuralların filtrelenmesidir. Bunun yanında kullanılan madencilik yöntemlerinin olumlu ve işe yarar sonuçlar üretip üretemediği de bu analiz sonucu ortaya çıkar. Analistin ortaya çıkan örüntüleri daha iyi değerlendirebilmesi ve elemeye tabi tutabilmesi için araçlara ihtiyacı vardır. SQL ve OLAP gibi mekanizmalar analiz için sıklıkla kullanılır. Örüntüleri grafik haline

getirmek veya farklı değerlere renkler atamak gibi görselleştirme teknikleri de verideki eğilimleri ve belli başlı örüntüleri vurgulayarak değerlendiricilerin işini kolaylaştırabilir.

- **Görselleştirme Teknikleri**: Görselleştirme hem gerçek hem de soyut, çeşitli tipteki olguları algılama konusunda insanlara yardımcı olur. Eğilimleri grafikleştirme, veri nesnelerine renk değerleri atama gibi yöntemlerle analistlerin ortaya çıkan örüntü sonuçlarını daha iyi kavraması hedeflenir.

- **OLAP Teknikleri**: OLAP (Çevrimiçi Analitik İşleme), iş ortamlarında stratejik veri analizi için güçlü bir yöntemdir. Sunucu günlükleri çok hızlı büyüdüğünden verinin tamamına çevrimiçi analiz sağlamak mümkün olmayabilir. Bu yüzden çevrim içi analizi mümkün kılabilmek için günlük verisini çeşitli şekillerde özetlemek gerekebilir. Web kullanım verisi analizinin bu tip ihtiyaçları için, temizlenmiş ve işlenerek belirli bir seviyede yapılandırılmış veri üzerinde OLAP ile analiz yöntemi verimli sonuçlar gösterebilir.

- **Veri ve Bilgi Sorgulama**: Madencilikle ulaşılabilecek çok fazla örüntü olduğundan analizin neye odaklanacağını belirleyecek bir mekanizmaya ihtiyaç vardır. Bu tip bir odaklanma en az iki yol ile belirlenebilir. İlk olarak örneğin, kısıtlar bir bildirim şeklinde veritabanına kaydedilerek veritabanının üzerinde madencilik yapılacak bölümü sınırlandırılabilir. İkinci olarak sorgulama, madencilik işlemi sonucu ortaya çıkartılan bilgi üzerinde yapılabilir ki bu durumda veri üzerinde sorgulama yapmak yerine bilgi üzerinde sorgulama yapabilecek bir dil gerekir [30].

3.3 Web Verisi

Veritabanlarında Bilgi Keşfi sürecindeki önemli adımlardan birisi veri madenciliği işlemleri için uygun bir hedef veri kümesi oluşturmaktır [27]. Web madenciliğinde

veri, sunucu tarafından, istemci tarafından, vekil sunuculardan veya bir organizasyonun web verisi saklayan veritabanından elde edilir. Veri toplamanın her çeşidi sadece veri kaynağının yeri açısından değil verinin tipi, verinin toplandığı veri kütlesinin katmanı ve uygulanma yöntemi açısından da farklılaşır. Web madenciliğinde kullanılabilecek çok çeşitli veri tipi vardır [19].

- İçerik: Web sayfalarındaki gerçek veridir yani web sayfalarının kullanıcıya taşımak için tasarlandığı veridir. Genellikle metin ve grafiktir ama sadece bunlarla sınırlı değildir.

- Yapı: İçeriğin organizasyonun tanımlayan veridir. Sayfa içi yapı bilgisi bir sayfada çeşitli HTML ve XML ayıraçlarının organizasyonundan ibarettir. Sayfalar arası yapı bilgisinin temel çeşidi ise sayfaları birbirine bağlayan hiper-bağlantılardır.

- Kullanım: Web sayfalarının kullanım örüntüsünü tanımlayan IP adresleri, sayfa atıfları ve erişimlerin tarih ve zamanları gibi verilerdir.

- Kullanıcı Profili: Bir web sitesi kullanıcılarına ait demografik veriyi içerir. İçeriğinde kayıt ve profil bilgisi vardır.

3.3.1 Veri Kaynakları

Farklı kaynaklardan toplanan kullanım verisi, tüm web trafiğinin farklı katmanlarına ait, tek kullanıcı tek web sitesi erişim örüntülerinden çoklu kullanıcı çoklu web sitesi erişim örüntülerine değişen gezinti örüntülerini içerir.

3.3.1.1 Sunucu Seviyeli Veri Kaynakları

Bir web sunucusu günlüğü web kullanım madenciliği uygulaması için önemli bir kaynaktır çünkü ziyaretçilerin gezinme sırasında sergiledikleri davranışları neredeyse tüm detayı ile kaydeder. Sunucu günlüklerinde kaydedilen veri, çoğu zaman birden çok kullanıcının aynı anda gerçekleştirdikleri erişimlerini içerir. Bu günlük verileri Ortak Günlük (Common log) veya Uzatılmış günlük (Extended Log) biçimleri gibi değişik dosya yapısı biçimlerinde saklanabilir. Bununla birlikte sunucu günlüklerinde kaydedilen web sitesi kullanım verisi, web ortamında çeşitli aşamalarda kullanılan önbellek mekanizmalarından dolayı tam güvenilir olmayabilir. Önbelleğe alınmış sayfalara ait görüntülenmeler sunucu günlüklerine kaydedilmez. Dahası POST metodu ile gönderilen önemli herhangi bir bilgi sunucu günlüğünde bulunamayacaktır [19]. Giden gelen veri paketlerinin analizi (sniffing) sunucu günlüklerinden veri toplanmasına alternatif bir teknolojidir. Paket analizi yapan programlar bir web sunucusuna gelen ağ trafiğini kontrol ederek kullanım verisini doğrudan TCP/IP paketlerinden alırlar. Web sunucusu çerezler ve sorgu verisi gibi diğer türde kullanım verilerini de farklı günlüklerde saklayabilir. Çerezler web sunucusu tarafından site ziyaretçilerini otomatik olarak takip edebilmek amacıyla istemci tarayıcıları için üretilen dosyalardır. HTTP protokolünün durum bilgisi saklamayan bağlantı modeli yüzünden bireysel kullanıcıları takip etmek kolay bir görev değildir. Sorgu verisi de çevrimiçi kullanıcılar tarafından bilgi gereksinimlerine göre yaptıkları sayfa aramaları sırasında üretilir. Sunucu tarafı içerik verisi, kullanım verisi dışında, yapı bilgisi ve web sayfaları hakkında bilgi (bir dosyanın boyutu ve son değiştirilme zamanı vs.) gibi bilgiler de sağlar.

Sunucu günlük dosyaları çoktan bire ilişkisine sahiptir. Birden çok kullanıcı bir web sitesini ziyaret eder ve o web sitesi hakkındaki kullanıcı davranışı buna erişimlere göre değerlendirilir [31].

3.3.1.2 İstemci Seviyeli Veri Kaynakları

İstemci taraflı verinin toplanabilmesi, uzak bilgisayarda çalışan bir program aracılığı ile (örn. Javascript veya Java appletleri) veya var olan bir tarayıcının kaynak dosyaları, veri toplama yeteneklerinin arttırılması amacıyla değiştirilerek başarılabilir. İstemci taraflı veri toplama uygulaması bu yüzden kullanıcının işbirliğine ihtiyaç duyar çünkü her iki yöntemde de kullanıcı onayının alınması gerekir. İstemci taraflı veri toplama sunucu taraflı veri toplamaya göre avantajlıdır çünkü hem önbellek kaynaklı sorunları hem de oturum tespiti problemlerini giderir. Bununla birlikte Java appletleri bir sayfanın gerçek görüntülenme zamanını belirleme açısından sunucu günlüklerinden daha iyi performans gösteremezler. Aslında Java appletleri özellikle ilk defa yüklendiğinde fazladan zaman kaybına sebep olabilir. Diğer taraftan JavaScript kodları biraz yorumlama zamanı harcar ve tüm kullanıcı tıklamalarını yakalayamazlar. Bu yöntemler sadece tek kullanıcı tek web sitesi olan durumlarda davranış yakalayabilirler. Kullanıcı davranışını yakalayabilmek için değiştirilmiş bir tarayıcı daha beceriklidir ve bir kullanıcının birden fazla web sitesi ile ilişkili verisini toplayabilir. Bu yöntemin en zor tarafı, kullanıcıları günlük web aktivitelerinde bu tip bir tarayıcıyı kullanmaya ikna etmektir. Sonuç olarak gerçek kullanıcı davranışı, istemci günlük dosyasından edinilebilir [32]. Kullanıcı davranışını göstermesi açısından istemci günlük dosyaları en özgün ve doğru kaynaklardır ama her istemci için tarayıcı değişikliği yapmak çok zordur ve kullanıcı işbirliği gerektirir [33].

3.3.1.3 Vekil Sunucu Veri Kaynakları

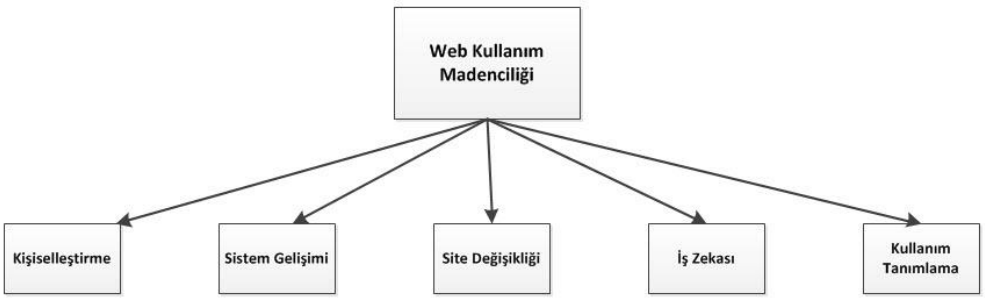
Bir web vekil sunucusu istemci tarafındaki tarayıcılar ile web sunucuları arasında önbellek mekanizmasının ara katmanı olarak davranır. Vekil sunucu önbellek mekanizması hem kullanıcılar tarafından deneyimlenen web sayfası yüklenme zamanını azaltmak hem de istemci ve sunucu tarafındaki trafik yükünü azaltmak için

kullanılır [34]. Vekil sunucu önbelleklerinin performansı gelecekteki sayfa isteklerini doğru tahmin edebilme yeteneklerine bağlıdır. Vekil sunucu kayıtlarındaki izler birçok istemciden birçok Web sunucusuna gerçek HTTP isteklerini barındırır. Bu açıdan vekil sunucular aynı vekil sunucuyu paylaşan anonim bir grup kullanıcının gezinti davranışını tanımlamak için iyi birer kaynak olarak değerlendirilebilir.

Vekil sunucu günlük dosyaları en karmaşık günlük dosyalarıdır ve kullanıcının davranışını doğru biçimde açığa çıkartmak çok zordur. Aynı IP adresi birden çok kullanıcı tarafından kullanılır [31].

3.4 Web Madenciliği Uygulama Alanları

Web madenciliği teknikleri yukarıda bahsedilen kaynaklarda bulunan verileri anlamak, analiz etmek için uygulanır ve web üzerindeki elektronik iş uygulamalarında satış, pazarlama ve müşteri desteği operasyonlarını geliştirmeye destek sağlayacak şekilde işe yarar bilgiye dönüştürebilir. Bulunan örüntüler sayesinde gelecekteki müşteri davranışları tahmin edilebilir, maliyet düşürücü önlemler alınabilir, pazarlama stratejilerine yön verilebilir, site kullanımı ve iş akış süreçlerindeki verimliliği arttırıcı kararlar alınabilir.



Şekil 3.5 Web Kullanım Madenciliğinin Ana Uygulama Alanları [19].

- Müşterileri Elde Tutma

Her müşteride önemli müşteri olduğu duygusunu yaratmak için kişiye özel hizmet uygulaması yapılabilir. Bunun için web kullanım madenciliği, kullanım verilerinden yola çıkarak her müşterinin gezinti örüntülerini bularak o müşteriye özel site deneyimi yaratılmasına yardımcı olabilir. Müşteri memnuniyetini arttırmada web kullanım madenciliği olumlu sonuçlar verecektir.

- Web Sitelerinin Bakımı ve Yeniden Yapılandırılması

Bir elektronik iş ortamındaki müşterilerin alışveriş ve gezinti alışkanlıkları, web verisi madenciliği ile yakalanarak web sitesinin yapılanması için en iyi yolun ne olduğunu belirlemek amacıyla kullanılabilir. Buradaki hedef, içeriklerini, yapılarını ve sunumlarını da kapsayacak şekilde web sitelerinin daha iyi yapılandırılarak web hizmetlerinin ve performanslarının iyileştirilmesidir. Oluşturulan model web sayfalarını ve kullanıcıları sınıflandırma yeteneğine sahiptir [35]. Kullanıcı profil kümelerine göre yeni ziyaretçilere veya aynı ziyaretçinin tekrar gelişinde önerilerde bulunulabilir. Örneğin bir şirketin web sitesine Cuma günü akşam 6-8 arası gelip eğitim ürünlerine erişmeye çalışan kullanıcılar akademik kullanıcılar olarak değerlendirilip ona göre odaklanılabilir. Madenciliğin web sitesinin optimizasyonu ile ilgili sunduğu diğer bir avantaj da tasarımcıların, kendilerine önerilerde bulunan uzmanların tecrübe ve sezgilerinden ziyade kullanıcı davranışlarındaki eğilimleri web kullanım madenciliği ile ölçümleyip site tasarımını değiştirebilmelerinin mümkün olmasıdır [36].

- Eğilim Tahmini

Web madenciliği eldeki veriye dayanarak gelecekteki değerleri belirlemek amacıyla eğilimi tahmin edebilir. Mesela bir elektronik açık arttırma şirketi açık arttırmadaki ürünler ve önceki mezat detayları ile ilgili veri barındırıyorsa tahmini modelleme yöntemi, var olan veriyi analiz edip gelecekteki mezatlara katılacak insan sayısı ve mezattaki ürünlerin son fiyatlarını tahmin etmek için kullanılabilir [30].

- Web sayfalarını kategorize etmek

Web madenciliği, web sayfalarını sınıflandırmak için farklı web siteleri arasındaki benzerlik ve ilişkileri keşfedebilir. Bu kategorizasyon, tüm web yerine istenilen dokümanların bu kategoriler içinde daha verimli aranmasına olanak tanır. Kategorizasyon kümeleme veya sınıflandırma teknikleri kullanılarak elde edilebilir.

- Saldırı Tespiti ve Güvenlik

Kullanıcının olağan kullanımına ait bir şablon, örüntüler analiz edilerek oluşturulabilir. Mevcut kullanıcının davranışı şablondan çok farklıysa alarm verecek bir güvenlik sistemi oluşturulabilir. Oluşturulacak bu güvenlik sistemi ayrıca kullanıcının belirli içeriğe erişimini kısıtlamakta da kullanılabilir.

- Elektronik Ticaret için Potansiyel Müşterileri Hedefleme

Web erişim günlüklerinin sınıflandırılması ve kümelenmesi bir şirketin pazarlama stratejilerini belirli bir grup müşteriye yönlendirmede yardımcı olabilir. Örneğin sınıflandırma kuralı madenciliği, belirli bir yerdeki belirli bir yaş grubundaki insanların belirli bir grup ürünü alma eğiliminde olduğunu keşfedebilir. Potansiyel reklam yerleri de yine böyle bir sistemle tespit edilebilir.

Müşteri kümesi elektronik işin önemli bir yönüdür. Benzer gezinme davranışlarına göre gruplama ve ortak özellik analizi ile elektronik iş sahipleri müşterilerini daha iyi tanıyabilir ve daha iyi hizmetler sağlayabilirler [37].

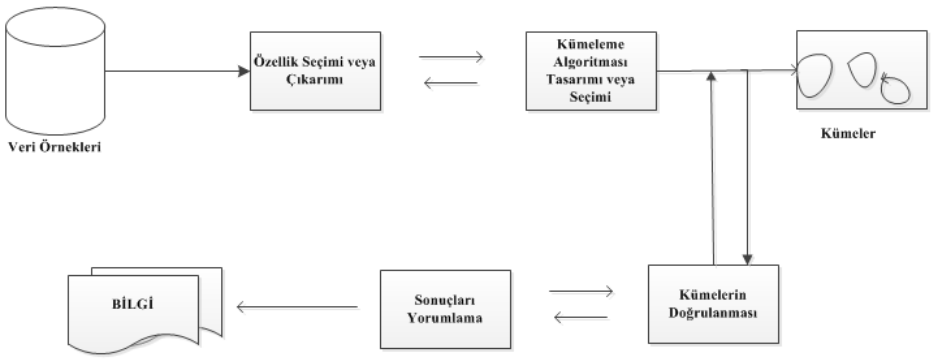
4. KÜMELEME ANALİZİ

Veri ile dolu bir dünyada yaşıyoruz ve her gün bu verileri analiz etmek ve yönetmek üzere depoluyoruz. Bu veriler ile başa çıkmanın en önemli yollarından birisi onları kategoriler veya kümeler haline getirmek üzere sınıflandırmak veya gruplamaktır. Aslında sınıflandırma, insanın tarihi gelişimi göz önüne alındığında en temel ve önemli aktivitelerden birisidir [42]. Yeni bir nesne veya olgu ile ilgili öğrenme veya anlama çabası içindeki insan ilk olarak o nesne veya olguyu tanımlayacak özelliklerin arayışı içine girer, yakınlık olarak genelleştirilebilecek benzerlik veya benzemezlik kavramaları temelinde daha önceden bildiği nesne ve olgularla karşılaştırmaya çalışır [39].

Kümeleme analizi, insanın öğrenme sürecinde önemli bir eylemdir. Çocukluğumuzdan itibaren bitkilerle hayvanları, kedilerle köpekleri nasıl ayırt edebileceğimizi farkında olmadan geliştirdiğimiz kümeleme yöntemleri ile öğreniriz. Otomatikleştirdiğimiz kümeleme yeteneğimiz sayesinde nesne uzayındaki yoğun ve seyrek bölgeleri tespit edip, geneldeki dağılım örüntülerini ve veri özellikleri arasındaki korelasyonları keşfedebiliriz [1]. Ayrıca insanlar özetleme teknikleri olmadan veri yığınları arasındaki bilgiyi kolayca keşfedemezler. Ortalama, varyans gibi basit istatistikî yöntemler veriyle ilgili olarak başlangıç düzeyinde fikir oluştururlar. Bununla birlikte nesneler arası, özellikler arası ve nesneler ile özellikler arası karmaşık ilişkiler kümeleme analizi ile keşfedilebilir [41]. Kümeleme analizi, pazar araştırması, örüntü tanıma, veri analizi ve görüntü işleme gibi çeşitli uygulamalarda kullanılmaktadır. İş hayatında kümeleme, pazarlamacılara, müşterilerini daha iyi inceleme fırsatı vererek onları alışveriş örüntülerine göre gruplandırmalarına ve farklı yöntemlerle yaklaşmalarına olanak vermektedir. Biyolojide benzer işlevselliklere sahip genleri gruplandırma, bitki ve hayvanların özelliklerine dayanarak soyağaçları oluşturmak gibi kolaylıklar sağlarken bilgi keşfi için web üzerindeki dokümanların sınıflandırılması için de kullanılabilir. Sınıflandırma her ne kadar nesneleri grup veya sınıflara ayırmanın etkili bir yolu olsa

da çoğunlukla sınıflandırıcının model olarak kullanacağı geniş bir etiketlenmiş alıştırma verisine veya örüntülerine ihtiyaç duyar. İşte bu nokta veri madenciliği açısından sınıflandırmadan farklı anlam ve yönteme sahip "kümeleme" kavramının ortaya çıktığı yerdir. Veri madenciliğindeki sınıflandırma kavram ve ilişkili yöntemleri eldeki veri kümesinin kategorilere ayrılması için baştan etiketlenmiş veya tanımlanmış alıştırma verisine ihtiyaç duyarken kümeleme yöntemleri bu kategorilere ayırma işini ön tanımlamalara ihtiyaç duymadan yapmaya çalışır. Gerçek hayatta genelde işe yarayan ve istenen de bu ters yöndeki çalışma şeklidir. Kümeleme olarak adlandırabileceğimiz yöntem ile eldeki veri, ilk önce benzerliklerine göre parçalanır ve bu ilk veri kümesine göre küçük bir şekilde parçalanmış veri kümeleri daha sonra etiketlenir. Kümeleme tabanlı bu sürecin diğer bir avantajı da değişikliklere uyum sağlayabilmesi ve farklı grupları ortaya çıkarmada yardımcı olacak özellikleri seçmede yardımcı olmasıdır [1].

Kümeleme analizi, örnekleri, belirli bir ilişki ölçüsüne dayanan otomatik sınıflandırmayla gruplara ayırmaya yarayan yöntemler topluluğudur öyle ki bir gruba ait olan örnekler birbirine benzerdir ama diğer gruptakilere benzer değildir [38].



Şekil 4.1 Kümeleme Analizi Adımları [39].

Şekil 4.1 genel olarak kümeleme analizi sürecinde gerçekleştirilen işlemleri göstermektedir. Bu dört adım şu şekildedir [39].

1) *Özellik Seçimi veya Çıkarımı*: Jain ve diğerleri [43], [44] ve Bishop [45], tarafından da belirtildiği gibi "özellik seçimi" adaylar kümesi içinden ayırt edici özellikleri seçerken, "özellik çıkarımı" orijinal özellikler arasından kullanışlı ve yeni özellikler üretmek amacıyla bazı şekil dönüşümlerinden yararlanır. Hem seçim hem de çıkarım kümeleme algoritmalarının etkinliği açısından çok önemlidir. Özelliklerin dikkatlice seçimi hem izleyen süreçteki iş yükünü önemli derecede azaltır hem de algoritma tasarım sürecini basitleştirir. Genel olarak ideal özellikler, farklı kümelere ait örüntüleri ayırt etmede işe yaramalı, gürültüye karşı dayanıklı olmalı, çıkarımı ve yorumlanması kolay olmalıdır.

   Bu adım uygun özellik seçimini ve veri elemanları üzerinde önişleme yaparak seçilen özellikler kümesinin değerlerini ölçmeyi içerir. Problem uzayının boyutsallığını azaltabilmek için çoğunlukla mevcut özellikler kümesinin bir alt kümesi seçilmeye çalışılır [46]. Kümelemede kullanılacak özelliklerin sayısının artması ilerleyen adımlarda iş yükünü arttırır. Özellik seçiminin bu hassasiyeti nedeniyle çalışılan alanla ve veri analizi ile ilgili iyi bilgiye sahip olmak gerekmektedir.

2) *Kümeleme Algoritması Tasarımı veya Seçimi: Bu* adım genelde beraber kullanılacak uzaklık ölçüsünün seçimi ve kıstas fonksiyonunun oluşturulmasını da içerir. Örüntüler birbirlerine benzeme durumlarına göre gruplandırılır ve sonuç kümelerinin oluşumunda seçilen uzaklık ölçüsünün etkisi büyüktür. Hemen hemen tüm kümeleme algoritmaları bir uzaklık ölçüsü kullanır. Hatta bazıları doğrudan uzaklık(benzerlik) matrisi üzerinde çalışır.

3) *Küme Doğrulaması*: Her kümeleme algoritması belirli bir yapıda olsun veya olmasın verilen veri kümesinde bölünmeler ortaya çıkartabilir. Dahası farklı algoritmalar veya yaklaşımlar genelde farklı sonuçlara yol açabildiği gibi aynı algoritmada bile giriş parametrelerindeki tanım farklılıkları ve örüntülerindeki sıralama değişiklikleri sonuçları etkileyebilir. Bu yüzden kullanıcıların, kullanılan algoritmalardan elde edilen sonuçlara, belirli bir derecede güven duymaları açısından etkin değerlendirme standartları ve kriterlerinin varlığı önemlidir. Değerlendirmeler nesnel olmalıdır ve veri içinde kaç tane kümenin gizli olduğu, elde edilen kümelerin anlamlı mı yoksa algoritma tarafından yapay olarak ortaya mı konduğu veya neden bir diğeri değil de bu algoritmayı seçtiğimiz gibi sorulara cevap verme açısından kullanışlı olmalıdırlar. Genel olarak üç tane test kıstası vardır: dış göstergeler, iç göstergeler ve görece göstergeler. Dış göstergeler, veri hakkındaki ön bilginin bir yansıması olan önceden belirlenmiş bir yapıya dayanır ve kümeleme çözümlerinin doğrulanması için standart olarak kullanılır. İç testler dış bilgiye yani ön bilgiye bağlı değildir tersine kümeleme yapısını orijinal veriye göre incelerler. Görece kıstaslar ise farklı kümeleme yapılarının karşılaştırılmasına dayanarak nesnelerin karakteristiklerini ortaya en iyi koyanın hangisi olduğunu bulmaya odaklanır.

4) *Sonuçların Yorumlanması*: Kümelemenin nihai hedefi, kullanıcılara karşılaştıkları problemleri verimli bir şekilde çözmeleri için orijinal veri ile ilgili anlamlı çıkarımlar sağlamaktır. İlgili alandaki uzmanlar veri bölümlerini yorumlarlar. Çıkarılan bilginin güvenilirliğini garanti etmek adına ileri analizler hatta deneyler gerekebilir.

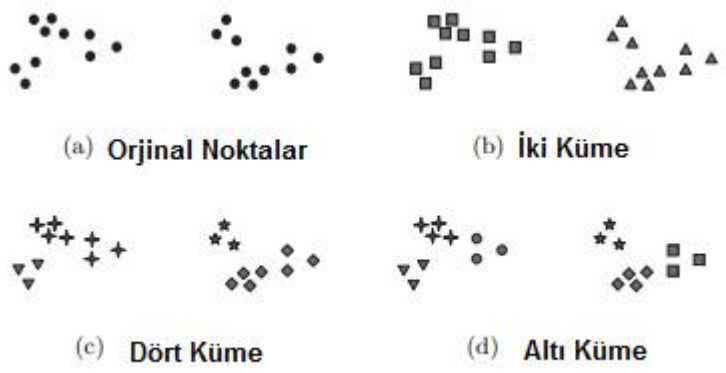
Şekil 4.1'de fark edilmesi gereken önemli bir nokta geri beslemenin varlığıdır. Kümeleme analizi bir seferde başlayıp biten bir süreç değildir. Çoğu durumda denemeler ve tekrarlar içerir ve dahası kümelemede kullanılacak özelliklerin ve kümeleme algoritmasının seçimi için genel kabul görmüş etkin kriterler mevcut değildir. Doğrulama kriterleri, kümeleme çözümlerinin güvenilirliği ve kalitesi ile

ilgili fikir verebilir fakat doğrulamaya uygun kriterlerin seçimi de uğraş gerektiren bir problemdir [39].

4.1 Kümeleme

Kümeleme analizi verinin yapısını keşfetmek için bir araçtır ve kümeleme analizinin merkezinde "kümeleme" yöntemi bulunmaktadır. Kümelemede, nesneler, özelliklerinin sahip olduğu değerler ve diğer nesneler ile olan ilişkileri (karşılıklı uzaklık, benzerlik) aracılığı ile tanımlanırlar [41].

Birçok uygulamada kümeleme kavramı iyi tanımlanmamıştır. Bir veri topluluğunda bulunan kümelerin yapılanması ile ilgili karar verme sürecinin ne kadar zor olabileceğini daha iyi anlayabilmek için Şekil 4.2 incelenebilir. Şekil 4.2'de yirmi nokta ve bunları kümelemenin üç farklı yolunu gösterilmektedir. Noktaların şekilleri küme üyeliklerine göre farklılaşmaktadır. Şekil 4.2(b) ve 4.2(d) veriyi sırasıyla iki ve altı kümeye bölmektedir. Bununla birlikte 4.2(b)'de görülen iki büyük kümenin her birisinin Şekil 4.2(d)'de üç alt kümeye bölünmesi, insan görsel algılamasının bir yapaylığı olarak ele alınabilir. Ayrıca Şekil 4.2(c)'de görüldüğü gibi orijinal verinin aslında dört kümeden oluştuğunu söylemek de çok mantıksız değildir. Kısacası Şekil 4.2'nin bütününden anlaşılacağı gibi küme tanımı belirsizliğe sahiptir ve en iyi küme tanımı eldeki verinin doğasına ve istenen sonuçlara bağlı olabilmektedir [40].



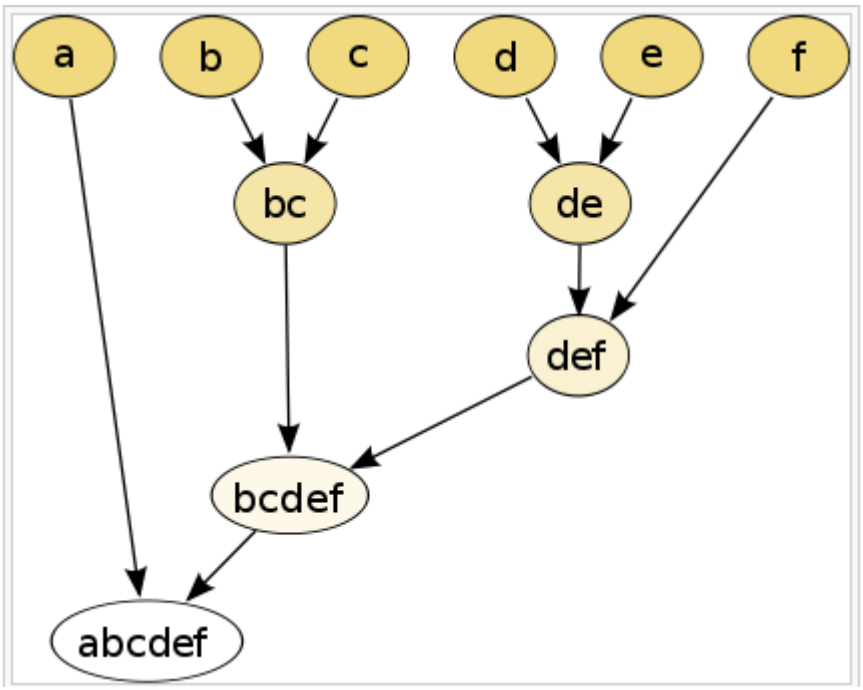
Şekil 4.2 Kümeleme ve İstenen Sonuçlar Arasındaki Fark [40].

Kümeleme, nesneleri gruplara bölmek için kullanılan diğer tekniklerle ilişkilendirilir. Örneğin nesneleri sınıf etiketleri ile tanımlaması nedeniyle sınıflandırmanın bir türü olarak kabul edilir. Hâlbuki kümeleme sınıf etiketlerini verinin içinden kendisi oluşturur. Sınıflandırma ise bunun tersine nesneleri daha önceden bilinen sınıf nesneleri ile oluşturulmuş bir modele göre etiketler ve bu yüzden "gözetimli öğrenme tekniği" olarak adlandırılır. Kümeleme ise veri topluluğundaki kendi yöntem sürecinde ortaya çıkardığı ve daha önceden belirlenen bir modele dayanmayan kategorileri etiketlendirdiği için "gözetimsiz öğrenme tekniği" olarak tanımlanır. Bu yüzden veri madenciliği terminolojisinde "sınıflandırma" denildiğinde gözetimli öğrenme akla gelir.

Kümeleme yöntemlerinin birçok sınıflandırma şekli bulunsa da en belirgin ayrımlardan birisi süreçler sonucu oluşan kümelerin "yuvalanmış" veya "yuvalanmamış" olmasına göre ya da diğer bir deyişle kümeleme yönteminin bölümlemeli veya hiyerarşik olmasına göre yapılan sınıflandırmadır. Bölümlemeli kümeleme yöntemlerinde oluşan sınıflar "yuvalanmamış" yani hiçbir küme diğerinin üzerini örtmemektedir ve her veri nesnesi sadece ve sadece bir kümenin elemanı olabilmektedir. Şekil 4.2 (b), (c) ve (d)'de bulunan kümeler yuvalanmamış yani

bölümlemeli kümelerdir. Eğer kümeler yuvalanmış şekilde olursa hiyerarşik yapıda bir kümeleme elde etmişiz demektir.

Geleneksel olarak kümeleme teknikleri hiyerarşik ve bölümlemeli (partitioning) olarak ayrılır. Hiyerarşik teknikler en tepede her şeyi kapsayan tek bir küme altında tekil kümelerden oluşan yuvalanmış küme dizisi üretirler. Her ara seviye bir alttaki kümelerin birleşimi ya da bir üstteki kümenin bölünmüş hali olarak değerlendirilebilir. Hiyerarşik kümeleme algoritmasının sonucu dendrogram olarak da adlandırılan ağaç yapısıdır. Bu ağaç yapısı iç içe yuvalanmış kümelerin birleşimini ve ara kümelerin oluşturduğu seviyeleri grafik olarak gösterir [46]. Şekil 4.3'de en üstte sıralı harfler veriyi, aşağı doğru ters ağaçtaki ara seviyeler ara kümeleri en alttaki düğüm ise tüm veriyi kapsayacak kümeyi göstermektedir.



Şekil 4.3 Hiyerarşik Kümeleme Gösterimi – Dendrogram [68].

Bölümlemeli kümeleme ise hiyerarşik kümelemenin tersine birbirini örtmeyen kümeler ve her veri elemanının sadece bir kümede olduğu kümelemeyi hedefler. Teker teker ele alındığında Şekil 4.2 (b-d)'deki küme toplulukları bölümlemeli kümelerdir.

4.2 Benzerlik Ölçüleri ve Uzaklık

Benzerlik iki nesne veya özellik arasındaki ilişkinin gücünü gösteren büyüklüktür [47]. Bu büyüklük genelde [-1, 1] aralığında değişmekle beraber [0, 1] aralığına da normalize edilebilir.

Uzaklık ise benzemezliğin ölçümüdür. Benzemezlik de çeşitli özelliklere dayanarak iki nesne arasındaki farklılığı ölçer. Benzemezlik iki nesne arasındaki uyumsuzluğun bir ölçüsü olarak da görülebilir. Uzaklık ve benzerliğin birçok çeşidi vardır. Uzaklık aşağıdaki kuralların en az ilk üçünü sağlayan nicel bir değişkendir. i ve j özellik olmak üzere:

- $d_{ij} \geq 0$
- $d_{ii} = 0$
- $d_{ij} = d_{ji}$
- $d_{ij} \leq d_{ik} + d_{jk}$ *(üçgen eşitsizliği)*

Eğer uzaklık yukarıdaki şartların dördünü de sağlarsa "metrik" olarak adlandırılır. Bu yüzden üçgen eşitsizliği kuralından dolayı tüm uzaklık fonksiyonları metrik değildir ama tüm metrikler uzaklıktır. Eğer özellik veya nesne i ile özellik veya nesne j arasındaki benzerlik s_{ij} ile ve benzemezlik δ_{ij} ile gösterilirse [0,1] aralığındaki benzerlik için benzemezlik ile arasındaki ilişki aşağıdaki gibi ifade edilebilir:

$$s_{ij} = 1 - \delta_{ij}$$

Denklem 4.1 [0,1] Aralığında Benzerlik ve Benzemezlik İlişkisi.

Burada benzerlik bir ise yani nesneler tam olarak benzer ise benzemezlik sıfırdır ve benzerlik sıfır ise yani tamamıyla farklılarsa benzemezlik birdir. Eğer benzerlik değer aralığı [-1,1], benzemezlik değer aralığı [0,1] olarak alınırsa benzerlik ve benzemelik arasındaki ilişki şu şekilde gösterilir:

$$s_{ij} = 1 - 2\delta_{ij}$$

Denklem 4.2 [-1,1] Aralığında Benzerlik ve Benzemezlik İlişkisi.

Benzemezlik 1 olduğunda (nesneler çok farklı) benzerlik -1 olur ve benzemezlik 0 olduğunda (çok benzer) benzerlik 1 olur. Birçok durumda benzemezliği diğer bir deyişle uzaklığı ölçmek benzerliği ölçmekten daha kolaydır. Benzemezliği yani uzaklığı ölçtüğümüzde kolayca normalize edip benzerlik ölçüsüne çevirebiliriz.

Kümeleme farklı veri yapıları (ikili - binary, nominal-kategorik, ordinal veya sayısal) için farklı uzaklık (ya da benzerlik) tanımlarından yararlanılarak yapılır. Sayısal değerler barındıran veri yapıları için kullanılabilen uzaklık hesaplama yöntemlerinden bazıları aşağıda açıklanmıştır.

4.2.1 Minkowski Metriği

Öklid uzayındaki genel uzaklık metriğidir [49]. Öklid, Manhattan ve Chebyshev uzaklıkları, formüldeki kuvvet değiştirilip Minkowski uzaklığından türetilebilir. Aşağıdaki formülde p üssü için 2 kullanılırsa Öklid uzaklığı, 1 kullanılırsa Manhattan uzaklığı, ∞ kullanılırsa Chebyshev uzaklığı elde edilir. Hem ordinal hem de nicel değişkenler için kullanılabilir [50].

$$P = (x_1, x_2, \ldots, x_n) \quad \text{ve} \quad Q = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^n$$

olmak üzere

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Denklem 4.3 Minkowski Metriği.

4.2.2 Öklid Metriği

Aralıklı değerlere sahip veri yapılarında kullanır. Öklid bağıntısına dayanır.İki nesne arasındaki "kuş uçuşu" mesafeyi hesapladığı da söylenebilir [48]. (p_1, p_2,…p_n) koordinatlarına sahip P nesnesi ile (q_1, q_2,…q_n) koordinatlarına sahip Q nesnesi arasındaki uzaklık aşağıdaki şekilde hesaplanır.

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}.$$

Denklem 4.4. Öklid Metriği.

4.2.3 Manhattan Metriği

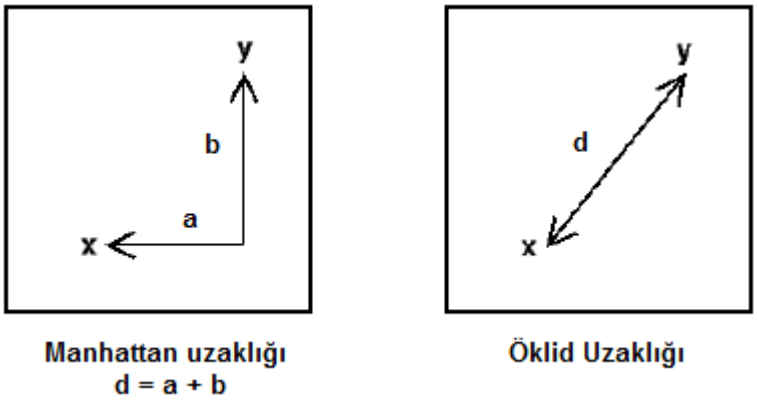
Şehir blok uzaklığı veya mutlak değer uzaklığı da denir. Verilen iki nesnenin kooordinatları arasındaki mutlak farkı inceler.

$$\mathbf{p} = (p_1, p_2, \ldots, p_n) \quad \text{ve} \quad \mathbf{q} = (q_1, q_2, \ldots, q_n)$$

olmak üzere Manhattan uzaklığı şu şekilde hesaplanır:

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

Denklem 4.5 Manhattan Metriği.



Şekil 4.4 Manhattan ve Öklid Uzaklıkları [51].

4.2.4 Açısal Ayırım ve Kosinüs Uzaklığı

İki vektör arasındaki açının kosinüsünü temsil eder. Uzaklıktan veya benzemezlikten çok benzerliği ölçer. Bu yüzden açısal ayrımın değeri ne kadar büyükse nesneler arası benzerlik o kadar fazladır. Açısal ayrımın değer aralığı kosinüs fonksiyonu gibi [-1,1] aralığındadır ve sıkça "korelasyon katsayısı" olarak adlandırılır [53]. Veri madenciliğinde kümeler arasındaki ilişkiyi ölçmek metin madenciliğinde ise özellikle doküman karşılaştırmak için kullanılır. A ve B gibi özellik değerleri tutan iki vektörün kosinüs benzerliği Θ skaler çarpım olarak şu şekilde ifade edilir.

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Denklem 4.6 Kosinüs Benzerliği- Açısal Ayrım.

4.2.5 Pearson Korelasyonu

Korelasyon iki değişkenin birbiri ile ne dereceye kadar ilişkili olduğunu veya beraber ne kadar değiştiklerini gösterir. En genel ölçüm yöntemi Pearson korelasyonudur. Bir metrik değildir. İki değişkenin farklılığının büyüklüğünü tespit etme yeteneği yoktur. İki değişken arasındaki Pearson korelasyon katsayısı bu iki değişkenin kovaryansının standart sapmalarının çarpımına bölümü olarak tanımlanır. -1 ve + 1 arasında değişen değerlerden pozitif olanlar pozitif bir korelasyonu negatif olanlar ise negatif korelasyonu gösterir.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}.$$

Denklem 4.7 Pearson Korelasyonu.

4.3 Kümeleme Algoritmaları

Literatürde birçok algoritma bulunmakla birlikte kategorilere ayırma durumunda her algoritmanın farklı kategorilerden özellikler barındırması nedeniyle kesin çizgilerle sınıflandırılmaları kolay değildir. Bununla birlikte çalışmalarda kolaylık olması nedeniyle göreceli de olsa bir kategorizasyona gidilebilir. En çok bilinen kümeleme yöntemleri ele alındığında aşağıdaki gibi bir sınıflandırma yapılabilir [1].

*Bölümlemeli (Partitioning) Yöntemler***:** n tane nesneden oluşan bir veri kümesinde bölümlemeli bir kümeleme yöntemi veriyi, k ≤ n olmak üzere k adet kümeye ayırır.

Dolayısıyla veriyi, (1) her küme en az bir nesne içermeli (2) her nesne sadece bir kümeye ait olmalı kurallarına uygun olacak şekilde k tane grup halinde sınıflar.

Oluşturulacak küme sayısı, k, verildiğinde yöntem bir başlangıç bölümlemesi yapar ve daha sonra kümeler arası tekrarlı yer değiştirmelerle başlangıç bölümlemesini iyileştirmeye çalışır. İyi bölümlemenin en önemli kuralı oluşturulan kümelerin son halinde küme içindeki veri elemanlarının yakın ya da birbirine benzer diğer küme elemanlarına ise uzak ya da farklı olmasını sağlamaktır.

Bölümlemeli kümelemede en uygun sonuca ulaşabilmek için mümkün olan tüm bölümlerin oluşturulması gerekir. Bunun yerine çoğu uygulama k-ortalamalı (k-means) veya k-medoids gibi sezgisel (heuristic) yöntemlerin uyarlanmasını seçerler. K-means algoritmasında kümeler, elemanlarının ortalaması ile temsil edilirken k-medoids algoritmasında kümeler, küme merkezine en yakın elemanlardan birisi ile temsil edilir. Bu sezgisel kümeleme yöntemleri küçük ve orta büyüklükteki veri topluluklarında küresel kümeler bulma açısından iyi çalışır. Daha büyük veri yığınları içinde karmaşık şekilli kümeler bulmak istenirse bölümlemeli yöntemler geliştirilmelidir [1]. Bu tip veri yığınları ile uğraşırken, CLARA (Clustering LARge Applications) gibi örnekleme tabanlı bir algoritma kullanılabilir.

Hiyerarşik Yöntemler: Bir hiyerarşik yöntem verilen veri nesnelerinin hiyerarşik bir çözümlemesini oluşturur. Hiyerarşik yöntemler de hiyerarşik çözümlemenin nasıl yapıldığına bağlı olarak toplamalı (agglomerative) veya bölmeli (divisive) yöntemler olarak ayrılır. Toplaşımlı yöntemler veya aşağıdan yukarıya yaklaşım her bir nesnenin ayrı bir küme olarak ele alınması ile başlar. Daha sonra tekrarlı birleştirmelerle nesnelerin ve kümelerin birbirlerine yakınlığına göre yeni kümeler oluşturarak hiyerarşinin en tepesinde tüm veri elemanlarından oluşan bir küme oluşuncaya veya bir koşul sağlanıncaya kadar çalışır. Yukarıdan aşağıya yaklaşım olarak da adlandırılan bölmeli yaklaşımda ise tüm veri elemanları aynı kümenin

içinde yer alacak şekilde başlanır. İlerleyen her adımda kümeler, her eleman bir kümeye ait oluncaya veya bir sonlanma koşulu sağlanıncaya kadar daha küçük kümelere bölünür.

Hiyerarşik ağaçlar verinin farklı soyutlama seviyelerinde görüntülenebilmesine olanak sağlar. Farklı çözünürlük seviyelerinde tutarlı kümeleme çözümlemeleri veri analizi sırasında kolaylık sağlayarak etkileşimli keşif ve görselleştirme için olanak sağlar [54]. BIRCH, ROCK (A Hierarchical Clustering Algorithm for Categorical Attributes), Chameleon (A Hierarchical Clustering Algorithm Using Dynamic Modeling) bu tip algoritmalardandır.

Yoğunluk Tabanlı Yöntemler: Çoğu bölümlemeli yöntemler nesneleri aralarındaki uzaklığa göre kümelerler. Bu tip yöntemler sadece küresel şekilli kümeleri bulmakta iyidirler ve değişik şekilli kümeleri bulurken zorlukla karşılaşırlar. Buna karşın bazı kümeleme yöntemleri de yoğunluk fikri üzerine geliştirilmişlerdir. Bu tip yöntemlerde genel fikir verilen bir kümenin komşuluğundaki yoğunluk (veri elemanlarının sayısı) belirli bir eşik değerini aşıncaya kadar kümenin genişletilmesidir yani verilen küme içindeki her bir veri noktasının belirli bir komşuluk yarıçapında eşik olarak belirlenmiş minimum sayıda nokta bulunmalıdır. Bu tip bir yöntem ayrık değerleri (outliers) ve farklı şekillerdeki kümeleri tespit etmede başarılıdır. DBSCAN ve ondan türetilmiş OPTICS bu tip yoğunluk tabanlı kümeleme algoritmalarına örnektir.

Grid Tabanlı Yöntemler: Grid tabanlı yöntemler nesne uzayını sayısallaştırarak bir ızgara yapısı oluşturan sınırlı sayıdaki hücrelere dönüştürür. Tüm kümeleme işlemleri bu ızgara yapısı yani sayısallaştırılmış uzayda gerçekleştirilir. Bu yaklaşımın en önemli avantajı veri nesnelerinin sayısına değil de sayısallaştırılmış uzayda bulunan her boyuttaki hücre sayısına bağlı olan hızlı çalışma zamanıdır. STING bu sınıftaki algoritmalara örnek olarak verilebilir.

Model Tabanlı Yöntemler: Model tabanlı yöntemler her küme için bir model önerirler ve verinin önerilen modele en uygun halini bulmaya çalışırlar. Model tabanlı bir algoritma veri noktalarının uzamsal dağılımını yansıtan bir yoğunluk fonksiyonu kullanarak kümeleri oluşturur. Model tabanlı yöntemler ayrıca istatistikî yöntemlere dayanarak küme sayılarını otomatik tespit etme, gürültü ve ayrık değerleri göz önünde bulundurma ve dolayısıyla daha sağlam kümeleme yöntemleri ortaya koyma konusunda başarılıdır. EM, COBWEB ve SOM (Self Organizing Maps) bu sınıfa ait bilinen yöntemlerdir.

Kümeleme yönteminin seçimi hem eldeki verinin tipine hem de oluşturulacak uygulamanın amacına bağlıdır. Eğer kümeleme analizi tanımlama veya keşif amacı ile kullanılacaksa çeşitli algoritmalar aynı veri üzerinde denenip verinin ortaya ne çıkartacağı gözlemlenebilir. Bazı algoritmalar farklı kümeleme yöntemlerine ait çözümleri barındırdıklarından her algoritmayı her zaman belirli bir yöntem kategorisine sokmak mümkün olmayabilir. Bazı durumlarda da uygulamalar amaçları doğrultusunda öyle kriterlere sahiptirler ki farklı algoritmaların beraber çalışıp uygulamanın amacına hizmet edecek şekilde işbirliği içinde kullanılmaları gerekir [1].

4.3.1 K - Means Algoritması

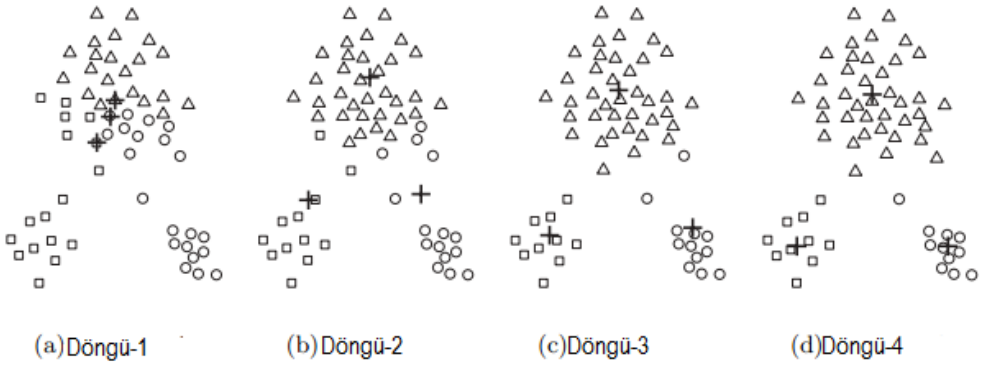
En eski kümeleme algoritmalarından olan k-means, 1967 yılında J.B. MacQueen tarafından geliştirilmiştir. En yaygın kullanılan gözetimsiz öğrenme yöntemlerinden birisi olan K-means'in atama mekanizması, her verinin sadece bir kümeye ait olabilmesine izin verir. Merkez noktanın kümeyi temsil etmesi ana fikrine dayalı bir metottur [1]. Eşit büyüklükte küresel kümeleri bulmaya eğilimlidir.

Kümeleme problemini çözen en basit gözetimsiz öğrenme algoritmalarından birisidir. Yöntem eldeki veri kümesini önceden belirlenmiş belirli sayıdaki kümeye

atamaya dayanan kolay bir yol izler. Ana fikir her küme için bir tane olmak üzere k tane merkez nokta (centroid) tanımlamaktır. Bu başlangıç merkez noktaları akıllıca seçilmelidir çünkü farklı seçilimler farklı sonuçlara yol açmaktadır. Bu yüzden en iyi seçim mümkün olduğunca birbirlerinden uzağa yerleştirmektir [55]. K-Means akış olarak aşağıda tanımlanmıştır.

```
k tane başlangıç merkez noktası seç
Repeat
    Her noktayı en yakın merkez noktaya atayarak k
    tane kümeyi oluştur
    Her kümenin merkez noktasını tekrar hesapla
Until merkez noktalar artık değişmiyor
```

Yukarıda da belirtildiği gibi kullanıcı tarafından oluşturulacak küme sayısı belirtilir. Bu küme sayısı, k, aynı zamanda başlangıçta seçilecek merkez nokta sayısıdır. Veri noktalarının her biri daha sonra bu seçilen merkez noktalara atanır. Bu atama sonucu ilk kümeler ortaya çıkmış olur. Daha sonra oluşan bu kümelerdeki verilerden yararlanarak merkez noktalar yeniden hesaplanır. Bu atama ve merkez noktaların güncellenmesi işlemleri kümelerdeki veri noktalarının yerleri değişmeyinceye yani farklı kümeler arası veri noktası alışverişi duruncaya veya merkez noktalar değişmeyinceye kadar devam eder [40].



Şekil 4.5 K-Means Döngüleri [40].

Şekil 4.5'te her döngüde "+" ile gösterilen küme merkezlerinin nasıl yer değiştirdiğine ve farklı küme elemanlarını gösteren küçük üçgen, kare ve çemberlerden bazılarının her döngüde farklı kümelere atanmalarından dolayı birbirine dönüşümleri görülmektedir. Son olarak burada en yakın merkez noktanın bulunması amacıyla kullanılabilecek sayısal bir yakınlık ölçüsünden bahsetmek gerekir. Öklid uzayında bu Öklid uzaklığı veya Manhattan uzaklığı olurken doküman kümeleme gibi uygulamalarda kosinüs benzerliği veya Jaccard daha uygun olabilmektedir. K- Means yöntemi çok sayıda değişken söz konusu olduğunda "k" değeri de küçük olursa hiyerarşik kümelemeye göre hızlıdır. Ayrıca eğer veri içindeki kümeler küresel yapıdaysa k-means hiyerarşik kümelemeye göre daha sıkı kümeler oluşturur [56].

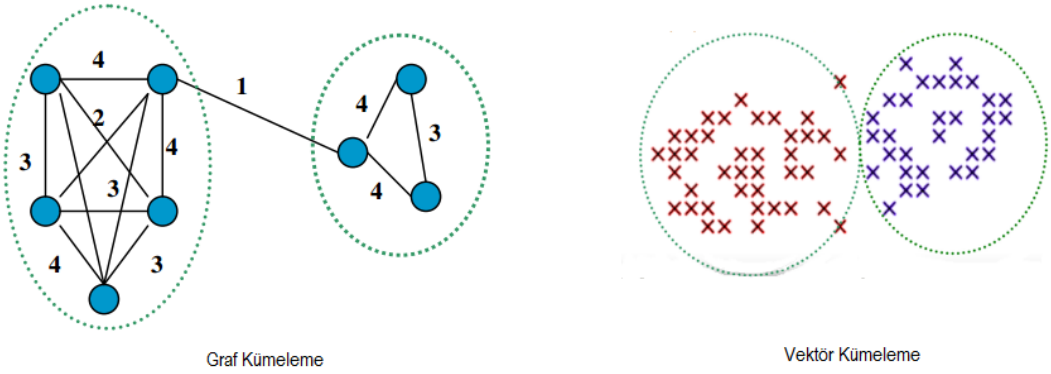
4.3.2 Graf Tabanlı Kümeleme

Veriyi temsil etme açısından güzel yollardan birisi her köşenin (vertex) bir veri elemanını, her kenara (edge) ait ağırlığın da bağlı olduğu iki köşenin benzerliğini temsil ettiği bir graf yapısıdır. Bu şekilde bir yapı oluşturulduktan sonra kümeleme,

graflar içindeki kenarların ağırlığının yüksek, graflar arası kenarların ağırlığının düşük olduğu alt graflara bölme problemine dönüşür [57].

Graf tabanlı kümelemenin arkasındaki hipotez şu şekilde ifade edilebilir: (1) Esas veri kümesini temsil eden büyük graf öyle yoğun alt graflara sahiptir ki bu alt grafların içlerindeki köşelerin (veri noktası) birbirleriyle olan bağlantısı alt grafları birbirleri ile bağlayan kenarların bağlantısından daha iyidir [57]. (2) Bir alt grafdaki rastgele gezinti o alt grafa ait köşelerin çoğu ziyaret edilinceye kadar sürecektir [58]. (3) Tüm köşeler arasındaki tüm en kısa yollar göz önüne alındığında yoğun alt graflar arasındaki bağlantılar birçok en kısa yol içinde yer alacaktır [58].

Yukarıdaki üç ifade aslında birbirleriyle çok güçlü bir şekilde bağlıdır ve sosyal bir ağ üzerinden açıklanabilir. Sosyal araştırma alanları açısından birbirlerine bağlanmış ve her birisinin bir köşe olarak ifade edildiği araştırmacılar ağı örnek olarak ele alınabilir. Bu ağda ilgi alanlarına göre topluluk olarak da adlandırılabilecek kümeler vardır ve yukarıdaki ifadelere göre değerlendirecek olursak: (1) Aynı topluluktaki araştırmacılar aynı konferanslara ve aynı projelere katılmak gibi aktiviteler aracılığı ile diğer topluluklardakine göre daha fazla etkileşimde ve ilişkide bulunacaklardır. (2) Bir araştırmacı düzenli bir şekilde makale okuyorsa büyük olasılıkla aynı topluluk içindeki araştırmacıların makalelerini okuyacaktır veya webde tüm araştırmacıların sayfalarını ziyaret ediyorsa büyük olasılıkla aynı toplulukta bulunan araştırmacıların sayfalarını ziyaret edecektir. (3) Birden çok toplulukla bağlantıları olan araştırmacılar büyük olasılıkla topluluklar arası işbirliğini geliştirecek ve araştırmacıları birbirleri ile tanıştıracaktır.



Şekil 4.6 Vektör ve Graf Kümeleme arasındaki görsel fark.

Şekil 4.6’ da vektör kümeleme görüntüsünde noktalar bir x,y ve renk değeri tutan vektöre sahiptir. Graf kümeleme de ise her kenar, noktalar arasındaki benzerliği ağırlık olarak taşımaktadır.

5. KULLANICI SORGULARI KÜMELEME UYGULAMASI

Bu bölümde kullanıcı sorgularının kümelenmesi amacıyla Maltepe Üniversitesi vekil sunucusu üzerinden derlenmiş altı aylık veri ile yapılan işlemlerden, ilgili yöntemlerden ve elde edilen sonuçlardan bahsedilmiştir.

5.1 Uygulamanın Amacı

Kullanıcı davranışlarının analiz edilip yorumlanabilir hale getirilebilmesi sanal dünyada özellikle ticari uygulamalar açısından önem taşımaktadır. Son dönemde istatistiki yöntemlerle olduğu kadar veri madenciliği gibi daha karmaşık yöntemler de kullanılarak, verimli sonuçlar elde etmek için yapılan çalışmalar artmıştır. Çoğunlukla kullanıcıların gezinme eğilimlerinin analiz edilmesi şeklindeki bu çalışmalar, ticari uygulamaların gerisinde yatan iş süreçlerinin optimizasyonu ve müşteri memnuniyeti sağlama amacını taşımaktadır.

Kullanıcı sorgularının diğer bir deyişle kullanıcıların web üzerinde arama yaparken kullandıkları anahtar kelimelerin kümelenmesi ile internet kullanıcılarının gruplandırılmasında farklı bir yaklaşım sergilenmeye çalışılmıştır. Kullanıcı ilgi alanı ve başka kullanıcılarla olan benzerlik, bu çalışmada kullanıcılar tarafından yapılan sorguların içerdiği kelimeler arasındaki benzerlik ile belirlenmeye çalışılmıştır.

5.2 Geliştirme Ortamı ve Kullanılan Araçlar

Verinin önişleme aşaması Java programlama dili ile Eclipse geliştirme ortamında gerçekleştirilmiştir [61]. Bunun başlıca nedeni Türkçe dili ile ilgili doğal dil işleme (NLP) kütüphanesi olarak Java ile geliştirilmiş olan Zemberek'in kullanılmasıdır

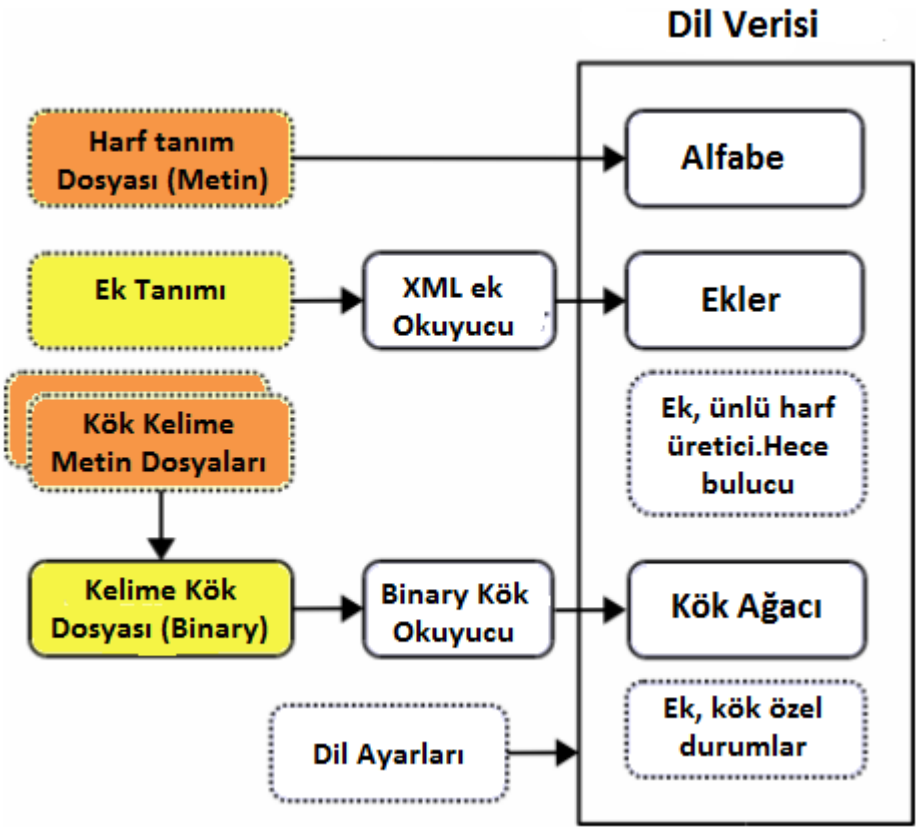
[59]. Önişleme sonrası ortaya çıkan IP-Terim matrisi Prof.Dr.Alexander Strehl (buradan sonra Strehl olarak bahsedilecektir) tarafından MATLAB'da geliştirilen kümeleme ve görüntüleme kütüphanesi kullanılarak incelenmiştir [60].

5.2.1 Zemberek Doğal Dil İşleme Kütüphanesi

Doğal dil işleme bilgisayar bilimlerinde en güncel ve en zor çalışma alanlarından biridir. Bitişken yapılı dillerdeki yoğun ek kullanımı bu dillere dayalı doğal dil işleme çalışmalarına fazladan zorluk katmaktadır.

Zemberek, kelimelere yoğun olarak sondan ekleme yapan başta Türkçe olmak üzere tüm Türki dillere yönelik açık kaynak kodlu, platform bağımsız, genel amaçlı bir doğal dil işleme kütüphanesidir [59]. Türkçe diline ilişkin çeşitli bilgi işlem sorunlarına çözüm bulmak amacıyla geliştirilmiştir. Basit doğal dil işleme işlemleri olan yazım denetimi, biçimsel çözümleme, kök bulma, kelime yapılandırma, kelime önerme, sadece ASCII karakterler kullanılarak yazılmış kelimeleri dönüştürme, hecelere ayırma gibi yetenekler sunmaktadır.

Kütüphane iki ana bölümden oluşmaktadır; bunlar dil yapı bilgisi ve NLP işlemleri modülleridir. Kütüphanenin ana bölümü NLP'ye özel algoritmalar içerir ve ilgili dile ilişkin uygulamaların kullanabilmesi için araçlar sunar. Her dil uygulaması önceden tanımlanmış dilbilgisi kuralları ile uyumlu olmak ve dile özgü verileri sağlamak zorundadır. Çoğunlukla NLP ile ilgili işlemlerin yeni bir dil uygulaması için değiştirilmesi gerekmez. Dile özgü veriler sağlandıktan sonra temel NLP fonksiyonları bu bilgileri kullanarak kullanıcılara erişimi kolay yazılım arayüzü ile hizmet vermeye başlar. Geliştiricilere kolaylık sağlaması açısından dile özgü veriler metin tabanlı kütüphane dışı dosyalarda tutulur. Bununla birlikte performans açısından özel durumlar ve ek üretme mekanizması, kütüphane içindeki program kodlarında tutulmuştur [63].



Şekil 5.1 Zemberek Dilbilgisi Elemanları ve Harici Dil Dosyaları [63].

Sol taraftaki yapılar NLP geliştiricileri tarafından sisteme sağlanması gereken dil tanımlarını sağ taraftaki yapılar ise kütüphane içinde bulunan yazılım sınıflarını ve dil ayarlarını içeren yapılandırma dosyalarını göstermektedir. Bu yapıya göre her dile ait alfabe bilgisi tanımı yapılmalıdır. Bunun için basit bir metin tabanlı dosya bulunmaktadır. Tüm Türkî dillerde çokça ek kullanılmaktadır. Bunun için Zemberek sistemi içinde ekler özel bir XML dosyasında tutulmaktadır. Eklere ilişkin özel durumlar mevcuttur ve Zemberek bunların üstesinden gelebilmektedir: Ara kökünün şimdiki zaman 3'üncü tekil hali ar-ıyor şeklindedir "ara-yor" veya "ara-ıyor" değil (Türkçede şimdiki zaman eki eklendiği kelimedeki son sesli harfi düşürür) veya edilgen ekler son ünsüze göre değişebilmektedir [63]:

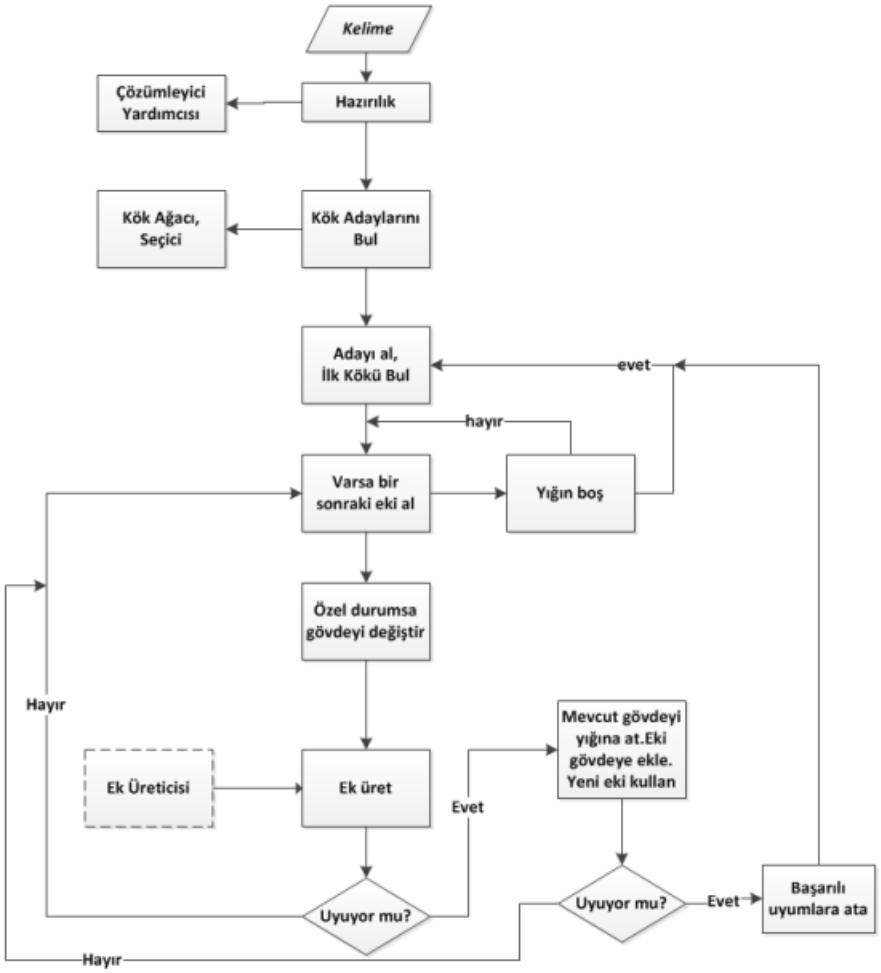
- Kes-il-mek
- gel-in-mek (gel-il-mek değil)

Kök kelime ek olmadan anlam taşıyan en temel kelimedir. Zemberekte kök kelimeler bir metin dosyasında sözlük halinde tutulmaktadır. Kök kelime, kelime tipi ve varsa özel durumlar şeklinde saklanan bu dosyayı, Zemberek çalışma zamanı performans nedeniyle doğrudan kullanmamakta bunun yerine Şekil 5.1'de de görülen "kök ağacı" denilen özel bir ikili ağaç yapısı kullanmaktadır. Ayrıca Türkçedeki kök ile ilgili özel durumlar hem kodun içinde hem de dil ayarları kapsamında özel dosyalarda tutulmaktadır. Köklerle ilgili bazı özel durumlar şu şekildedir [63] :

- burun → burnu
- su → suyu (sunu değil)
- ben → bana (bene değil)

Zemberek kütüphane yapısında bu çalışma açısından en önemli bileşen Türkçe kelime köklerini bulmada doğrudan işlevsel olan yapısal çözümleyicidir. Yapısal çözümleyici basit tanımıyla, verilen bir kelimenin mümkün olan kök ve eklerini bulur. Yapısı basit olarak sözlük tabanlı tepeden aşağı çözümleyicidir. Giriş değeri olarak verilen bir kelimenin yapısal analizinde aşağıdaki basamaklar vardır [63]:

- Kelimeyi hazırlama veya önişleme (aksan veya tire gibi işaretleri temizlemek, küçük harflere dönüştürmek vs.).
- Uygun kök seçici ile verilen kelime için aday kökleri bulma.
- Her kök adayına giriş kelimesi oluşuncaya ya da ekleyecek kök kalmayıncaya kadar mümkün olan ekleri özel durumları uygulayarak ekleme.
- Çözümleyici sonuçlarını işleme.



Şekil 5.2 Zemberek Kelime Çözümleyici Akışı [63].

Zemberek kütüphanesi Java ile geliştirilmiş olup gönüllü katılımcılar tarafından kelime veritabanı ve uygulama üzerinde çalışılarak sürekli geliştirilmektedir. Bu çalışma sürecinde Zemberek versiyonu 2.1.1 kullanılmıştır

5.2.2 MATLAB

MATLAB adı Matrix Laboratory(Matris Laboratuarı) kelimelerinin ilk üç harfinin birleştirilmesiyle oluşturulmuştur. MathWorks adlı firma tarafından geliştirilen

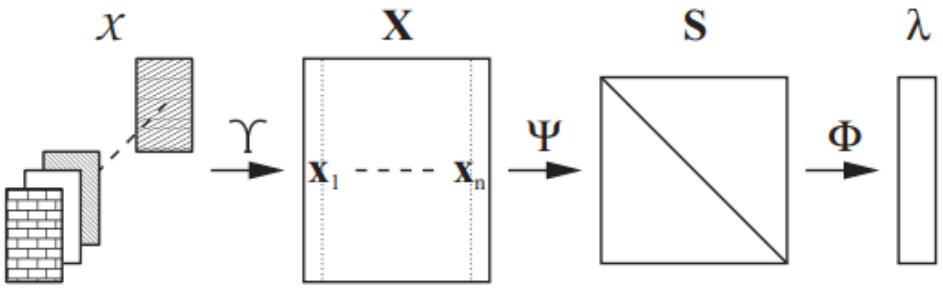
MATLAB, sayısal hesaplama ortamı ve 4. nesil bir programlama dilidir. Matris işlemleri, fonksiyon ve veri grafiği çizimi, algoritma kodlama, kullanıcı arayüzü geliştirme ve C, C++, Fortran gibi programlama dillerinde yazılmış programlarla arayüz oluşturabilme yeteneklerine sahiptir. Mühendislik, bilim ve ekonomi alanlarında kullanılmaktadır [62].

Tez çalışması sırasında MATLAB, Strehl kütüphanesine ait kümeleme ve görüntüleme fonksiyonlarının çalıştırılması amacıyla kullanılmıştır.

5.2.3 Strehl Küme Analizi Kütüphanesi

Strehl'in makine öğrenmesi için geliştirdiği kodlar arasından seçip oluşturduğu MATLAB fonksiyonlarından oluşan bir kütüphanedir [60]. ClusterVisual, ClusterBasics ve ClusterEnsemble isimli üç alt modülü olan bu yapının bu çalışmada ilk iki modülü kullanılmıştır.

ClusterBasics Modülü : Bu modül çeşitli kümeleme fonksiyonlarının MATLAB uygulamalarını içerir. Strehl'e göre kümeleme, nesneleri kendi aralarındaki ilişkilere veya benzerliklere göre gruplamak olduğundan kümeleme çalışması sırasında orijinal özellik uzayı yerine benzerlik uzayı kullanılabilir. Buradaki önemli nokta eğer problemin var olduğu alana uygun olarak özelliklere dayanan bir benzerlik ölçüsü bulunabilirse nesneler arası yakınlığı temsil edebilecek bir sayı bulunabilir. Daha sonra yapılması gereken analizler, bu sayılara dayanarak gerçekleştirilebilir. Böylece oluşturulan benzerlik uzayı son aşamadaki küme sonuçlarını görselleştirme tekniğinde de kolaylık sağlar [41]. Veriyi yüksek boyutlu özellik uzayından, benzerlik uzayına taşımak çok boyutluluğun getirdiği çalışma zamanı ve karmaşıklık sorunlarından kurtarır. Strehl bu yönteme "ilişkiye dayalı kümeleme" adını vermektedir [41,65].



Şekil 5.3 İlişkiye Dayalı Kümeleme [41].

Şekil 5.3'te, χ olarak gösterilen ham nesne tanımları, özellik uzayında X olarak vektörel ifadeye dönüştürülmüş, daha sonra benzerlik uzayı S'de ilişkisel olarak tanımlanmıştır. Süreç çıktı olarak λ küme etiketleri ile sonlandırılmıştır. Örnek olarak web sayfası kümelemede χ, n adet sayfadan oluşan bir web sayfası topluluğudur. Belirli özelliklere önem verilip kullanılmalarına karar verilerek ortaya çıkartılan X vektörleri bu sayfalardaki kelimelere ait köklerin frekanslarını tutan özellik vektörleridir. Bu vektörlerin benzerlikleri de kosinüs benzerliği ile hesaplanıp n x n boyutlarındaki S benzerlik matrisi oluşturulur. En sonunda da küme etiketleri vektörü, graf kümeleme algoritması gibi bir algoritmayla (burada Φ ile gösterilmektedir) hesaplanır.

Bu modülde gerçeklenen algoritmalar, k-means, graf tabanlı kümeleme algoritması olan "ağırlıklı graf bölümleme", hiper graf bölümleme ve Kohonen ağlarına dayalı "kendi kendini düzenleyen özellik haritaları algoritmalarıdır" (SOFM) [64]. Bu algoritmalar, yazarın market-sepeti kümelemesi ile ilgili çalışmaları sırasında yaptığı denemeler için geliştirilmiş olup, yazar, web dokümanları ve web günlüklerini kümelemek için de vektör uzayında yapılan hiyerarşik kümeleme yöntemleri yerine graf tabanlı kümeleme yöntemi önermektedir [41,65]. Strehl graf kümeleme algoritması olarak market sepeti verileri için özelleştirilmiş benzerliğe dayalı

kümeleme yapan OPOSSUM'u önermektedir. OPOSSUM diğer graf tabanlı kümeleme tekniklerinden farklı olarak uygulama kaynaklı olarak kümelerin dengeli hale getirilmesine olanak sağlar, metrik olmayan benzerlik ölçülerini kullanabilir ve uygun k (küme sayısını) bulabilmek için görselleştirmeden yararlanma olanağı sunar [41].

OPOSSUM'un önemli özelliği bir büyük küme ve k-1 tane ayrık tekil küme gibi kümeleme açısından değersiz sonuçları dengeleme yöntemi ile engellemesidir. Bu sayede her kümede eşit sayıda küme elemanı oluşturarak değerlendirme açısından kolaylık sağlamaktadır. Bunu iki şekilde yapar: "Örnek dengelemeli" ki burada her küme eşit sayıda örnek içerir (n/k) ve "değer dengelemeli", her küme kabaca aynı sayıda özellik değeri içerir [41]. Strehl ve diğerleri istenilen dengeleme özelliklerini her nesneye (müşteri, doküman, web oturumu) bir ağırlık atayarak ve sonra her kümedeki ağırlıkları sınırlandırarak elde etmişlerdir. Örnek dengelemeli kümelemede her x_j örneğine n nesne sayısı olmak üzere aynı ağırlık yani $w_j = 1/n$ atanmıştır. Değer dengelemeli yöntemde ise x_j örneğinin ağırlığı Denklem 5.1 ile bulunur. Burada dikkat edilmesi gereken tüm örneklerin ağırlık toplamı 1'dir.

$$w_j = \frac{1}{v} \sum_{i=1}^{d} x_{i,j}$$

Denklem 5.1 Değer dengelemeli örnek ağırlığı

OPOSSUM'da kümeleme problemi, köşelerinde kümelenecek nesneler olan, kenarların ağırlığının ise bu nesneler arasındaki benzerlik değerlerinin olduğu bir grafı, k adet birbirinden bağımsız ve dengeleme kıstası tarafından tanımlanmış eşit boyutlara sahip bileşenlere bölme problemidir. Köşe ağırlıklı graf bölümleme adı verilen bu yöntemde kümelenecek nesneler köşe kümesini oluşturur (V = { x_1,

x_2,....., x_n}). x_a ve x_b gibi iki köşe $(a,b) \in E$ olan yönsüz ve pozitif ağırlığı benzerlik matrisinde bulunan, s(x_a, x_b), bir kenar ile birbirine bağlıdır. Bu G = (V,E) grafını tanımlar. Bir "kenar ayırıcı", ΔE ise kaldırıldığında G grafını k adet bağlantısız alt grafa bölen kenar kümesidir. Burada kümeleme işi grafı k adet bağımsız parçaya bölecek en az kenar ağırlığına sahip kenar ayırıcıyı bulmaktır. Aşağıdaki formül "minimum kesik hedefi" de denilen bu kıstası göstermektedir :

$$\min_{\Delta E} \sum_{(a,b)\in \Delta E} s(x_a, x_b)$$

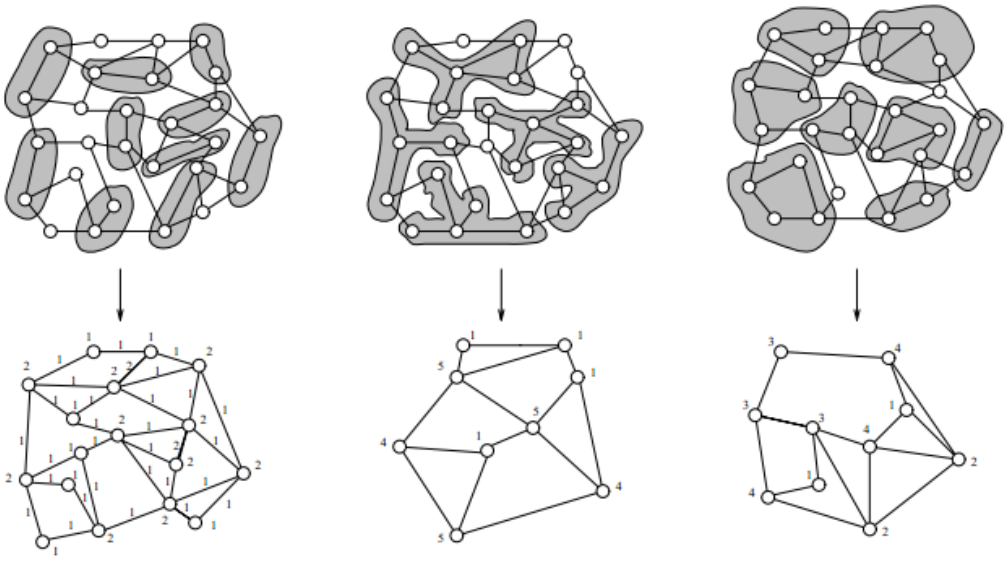
Denklem 5.2 Minimum Kesik Hedefi.

Minimum kesik hedefi için uğraşırken, w_j köşe ağırlıkları olmak üzere dengeleme kısıtı da göz önünde bulundurulmalıdır:

$$k.\, l \in \max_{\{1,\ldots,k\}} \sum_{\lambda_j = l} w_j \leq t$$

Denklem 5.3 Dengeleme Kısıtı.

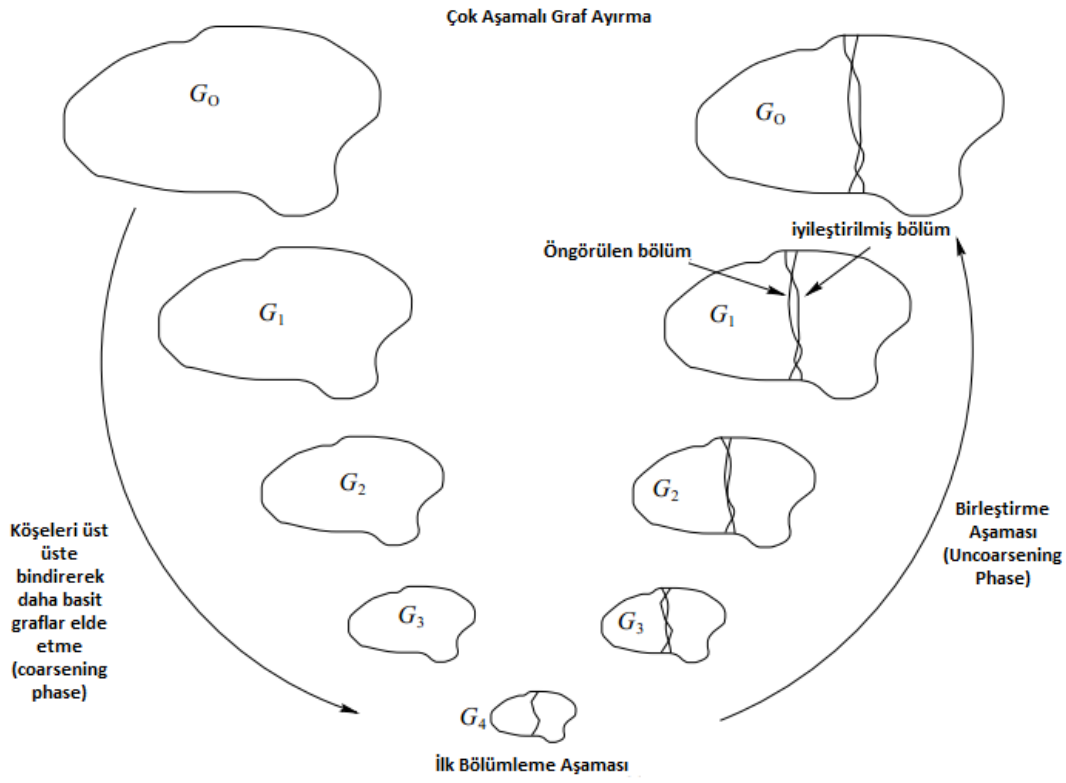
Burada graf bölümleme, kısıtlara sahip optimizasyon problemi haline gelir. Bu tip bir optimal bölümleme arayışı NP zorluk derecesine sahiptir ve bunun için Strehl'in önerisi çok aşamalı bölümleme algoritması olan METIS algoritmasıdır. METIS çok kısıtlı, çok hedefli graf bölümleme sorununu üç aşamada ele alır [66]. Kabalaştırma veya indirgeme (coarsening) denilen ilk aşamada grafı Şekil 5.4'teki gibi bitişik köşeleri üst üste getirerek detayı azaltılmış graflar haline getirir. Graf boyutu sürekli azaltılır.



Şekil 5.4 Bir grafı indirgemek [66].

Bölümleme olarak adlandırılan ikinci aşamada kabalaştırılmış graf köşe ağırlığı toplamı aynı olan ve farklı alt kümelerde köşeleri olan kenar ağırlıkları toplamı minimize edilmiş k adet ayrık alt kümeye bölünür. Bu bölümlemeler sonucu ortaya çıkan n elemanlı P bölüm vektörü, n elemanlı V kümesinin her elemanı için P[v] içinde 1 ile k arası bir sayı tutarak, v köşesinin hangi kümede olduğunu saklar. Bir P bölümü için "kenar kesiği" (edge-cut), farklı alt kümelere ait köşeleri bulunan kenarların sayısıdır. Bu bölümlemeleri elde etmek için İzgesel İkiye Ayırma(spectral bisection), Kernighan-Lin algoritması, Kapsam Büyüterek Graf Bölümleme(Graph Growing Partitioning) algoritması gibi yöntemler kullanılabilir.

Son aşama olan detaylandırma (uncoarsening) olarak da adlandırılabilecek geriye dönüş aşamasında en kaba halindeki G grafın üst üste binmiş köşeleri iyileştirmeler yapılarak açılır [66].



Şekil 5.5 METIS aşamaları [66]

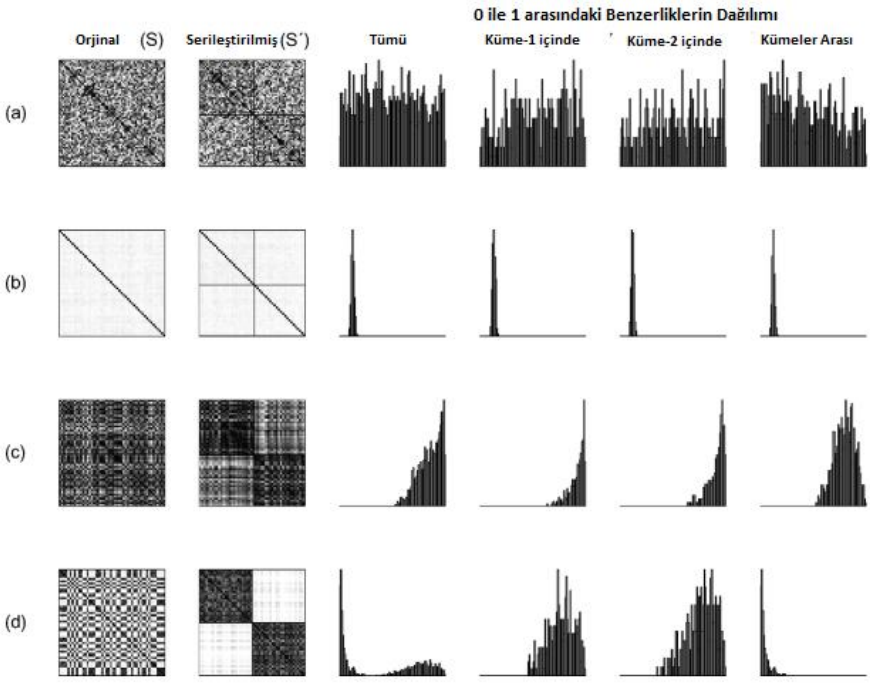
Yukarıda bahsedilen algoritmalar ve benzerlik ölçüleri Strehl kütüphanesi içinde clcgraph.m, clgraph.m, clhgraph.m, clkmeans.m, cmetis.m, evalbalance.m, hmetis.m, metis.m, simcorr.m, simcosi.m, simeucl.m, simxjac.m, wgraph.m dosyalarında gerçeklenmiştir.

ClusterVisual modülü: Strehl kütüphanesinin bu modülü CLUSION isimli küme görselleştirme aracının MATLAB gerçeklemesini içerir. Clusion (CLUSter visualizatION-Küme Görselleştirme Aracı), çok boyutlu veriyi algısal olarak daha uygun bir hale getirerek insan gözüyle verinin içinde yer alan ilişkilerin görülebilmesini sağlar, kümeleme sürecine yardımcı olur ve sonuçların kalitesini doğrulamayı kolaylaştırır [65].

Clusion kümeleme işleminin sonucuna bakar, aynı küme etiketine sahip veri noktalarını bitişik olacak şekilde yeniden düzenler ve sonra sonuçta ortaya çıkan değiştirilmiş benzerlik matrisini ($\mathbf{S}'$) görselleştirir. Bu işleme "kaba serileştirme" denilmektedir ve benzer yapıların yakınlaştırması anlamında antropoloji ve arkeoloji gibi disiplinlerden alınmıştır.

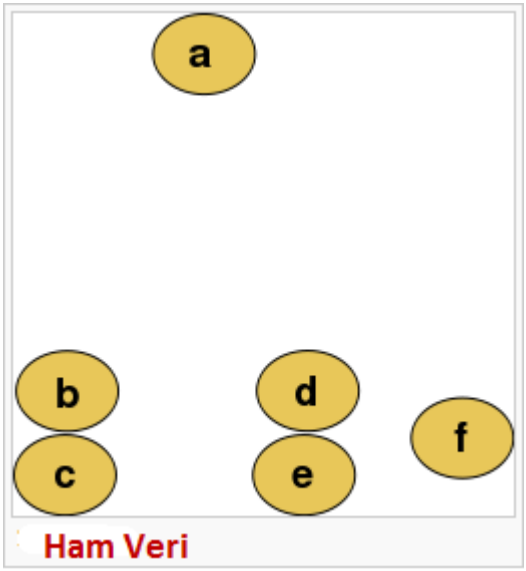
Benzerlik matrisinin serileştirilmesi, ($\mathbf{S}'$), görselleştirme açısından çok önemlidir. Benzerlik matrisi iki boyutlu olduğu için beyaz (siyah) piksellerin en düşük (en büyük) benzerlik 0(1) şeklinde değerlendirilmesi ile gri-seviyeli görüntüye dönüştürülmeye hazırdır. Satır a ve sütun b'deki pikselin grilik seviyesi x_a ve x_b örnekleri arasındaki benzerliğin büyüklüğüne göre artar. Görüntüye bakarken benzerlik değeri s'nin 0 ile 1 arasında değer alan rastgele bir değişken olduğunu göz önünde bulundurmak faydalıdır. Bu nedenle l kümesi içinde beklenen benzerlik matrisin ana köşegeni üzerinde kenarı n olan kare alanı içindeki ortalama yoğunluk ile temsil edilir. Köşegene yakın olmayan dikdörtgensel bölgeler de kümeler arası ilişkiyi gösterir. Dikdörtgensel bölgelerdeki parlaklık dağılımı kümeleme kalitesine ve mümkün olabilecek iyileştirmelere ilişkin bilgi verir. Bu bölgelerin daha belirgin olabilmesi için yatay ve düşey çizgiler kullanılarak dikdörtgensel bölgeler içindeki bölümler gösterilmiştir. Benzerlik uzayının bu şekilde görselleştirilmesi verideki kümeler hakkında hızlı bir izlenim elde etmeyi mümkün kılar. Çok miktarda veri noktaları söz konusu olsa bile, bir veri setindeki yaklaşık k adet kümenin varlığı fark edilebilir [65]. Benzerlik matrisinin dokunulmamış ve serileştirilmiş halinin görseli Şekil5.6'da uç örnekler halinde verilerek CLUSION yönteminin kavranılması daha kolay hale getirilmiştir. Tüm satırlar farklı veri topluluklarını göstermektedir ve her satırda iki küme olduğu varsayılmıştır. Sağdaki dört sütun (S) benzerlik değerlerinin tüm veri açısından, küme-1 içindeki, küme-2 içindeki ve kümeler arasında kalanların dağılımı gösterilmiştir. Eğer veri içinde doğal olarak iki küme varsa ve kümeleme algoritması iyiyse ikinci ve üçüncü sütunlardaki histogramlar, bir ve dördüncü sütunlardakilere göre sağa daha fazla yakın olacaktır. Şekil 5.6'da "a" satırında görülen desen içinde benzerlik rastgeledir ve küme bulunmamaktadır. ($\mathbf{S}'$) matrisinde köşegene yakın ve uzak bölgelerde güçlü bir görsel farklılık bulunmamaktadır. Bu

kümelemenin etkisiz olduğunu gösterir ki benzerlik matrisinde de belirgin bir yapı görünmemektedir. “b” satırında ise sadece köşegen ortaya çıkmıştır ki bu da verinin tekil ve ilişkisiz (singleton) yapılardan oluştuğunu ve veri içinde belki de tek tek veri noktalarına kadar varacak daha fazla bölünmenin mümkün olduğunu göstermektedir. “c” satırındaki tekil sayılabilecek veride ise veri noktaları arasındaki çok sayıdaki benzerlik koyu renkli S benzerlik matrisi ile gösterilmiştir. Verinin bölünmesi köşegenden uzak koyu bölgelerin oluşmasında neden olmuştur ve bu da köşegene yakın karesel bölgelerin aslında birbirine çok benzediğini ve birleştirilebileceğini gösterir. Şekil 5.6(d) satırında ise birbirinden farklı iki küme köşegene yakın karesel iki alanla belirlenmiş, köşegene uzak açık renkli kareler de bu köşegen üzerindeki koyu bölgelerle temsil edilen kümelerin birbirinden ayrık kümeler olduğunu vurgulamıştır [65].



Şekil 5.6 Benzerlik Matrisinin Orijinal ve Serileştirilmiş Clusion Desenleri [65].

CLUSION görselleştirme yöntemi ile oluşan desenlerin değerlendirilmesi ile ilgili olarak diğer bir örnek Şekil 5.7'de verilmiştir.

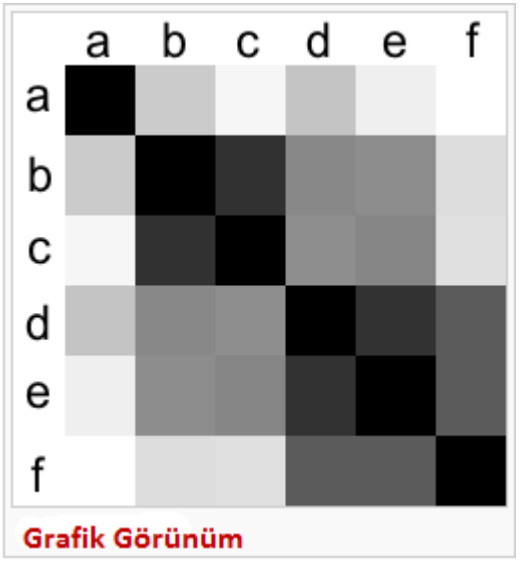


(a)

	a	b	c	d	e	f
a	0	184	222	177	216	231
b	184	0	45	123	128	200
c	222	45	0	129	121	203
d	177	123	129	0	46	83
e	216	128	121	46	0	83
f	231	200	203	83	83	0

Öklid Uzaklık Matrisi

(b)



(c)

Şekil 5.7 CLUSION açıklaması – Isı Haritası [69].

Şekil 5.7 (a)'da örnek verinin uzayda yerleşimi görülmektedir. Veri nesnelerinin piksel değerlerinden yararlanılarak hesaplanan öklid uzaklıklarına göre oluşturulan uzaklık matrisi ise Şekil 5.7(b)'de yer almaktadır. Bu uzaklık matrisi Şekil5.7 (c)'de "ısı haritası" olarak da adlandırılan grafik şeklinde gösterilebilir. Burada siyah 0 uzaklığı, beyaz ise maximum uzaklığı göstermektedir. Veri nesnelerinin kendilerine uzaklığı 0 olduğundan matrisin köşegeni siyah, birbirine uzamsal olarak en uzak noktalarda bulunan a ve f nesneleri arasındaki uzaklık ise beyaz ile gösterilmiştir. Diğer nesneler arasındaki uzaklıklar yakınlık derecelerine göre siyah ve beyaz arasında farklı tonlara sahip renkler ile derecelendirilmiştir.

5.3 Kullanılan Veri Seti

Bir web vekil sunucusu istemci tarafı tarayıcılar ile web sunucuları arasında ara seviyeli önbellek katmanı oluşturur. Vekil sunucunun sağladığı önbellek mekanizması sayesinde kullanıcı tarafında yaşanan geç sayfa yüklenmesi gibi performans sorunları aşılırken sunucu tarafında ağ trafiğinden kaynaklanan yoğunluk azaltılır. Vekil sunucu üzerindeki günlük dosyaları, birden çok istemci tarafından birden çok web sunucusuna yapılan gerçek HTTP isteklerini içerir. Bu sayede aynı ortak vekil sunucuyu paylaşan anonim kullanıcıların gezinme davranışları analiz edilebilir. Daha özel olarak belirtmek gerekirse vekil sunucu günlükleri web dokümanlarına erişmeye çalışan belirli bir grup kullanıcının (mesela aynı servis sağlayıcının müşterileri) isteklerini kaydeder. Erişim günlüğündeki her satır tek bir doküman için yapılan isteğe ait detayları tutar. Her günlük satırından isteği yapan makine adı, isteğin yapıldığı zaman ve istenilen dokümanın adı gibi bilgileri elde etmek mümkündür. Aynı kayıt, sunucunun isteği karşılayıp karşılayamadığı, eğer karşılayamadıysa nedeni ve sunucu tarafından kaç byte iletildiği gibi sunucunun bu isteğe yanıtına ilişkin bilgileri de içerir [67].

Çizelge 5.1 Üzerinde çalışılan vekil sunucu örnek verisi.

1.237.535.466.430	98	1	TCP_MISS/20	3659	GET	http://www.notasarim.org/no/main.asp?cid=3&lang=1	10.1.190.96	DEFAULT_PARENT/127.text/html
1.237.535.466.451	9015	2	TCP_MISS/20	496	GET	http://ad.e-kolay.net/orfad.a2?target=vatan_fotogaleri&fqlist=&l	10.1.50.15	DEFAULT_PARENT/127.text/html
1.237.535.466.452	36	3	TCP_DENIED/	0	GET	http://platform.ak.facebook.com/www.new/app_full_proxy.php	10.1.120.33	DEFAULT_PARENT/127.-
1.237.535.466.556	430	4	TCP_MISS/20	807	GET	http://imageserver.ebscohost.com/WebImages/graphicPixel.gif	10.1.55.21	DEFAULT_PARENT/127.-
1.237.535.466.589	1670	5	TCP_MISS/20	26909	GET	http://us.mc317.mail.yahoo.com/mc/showFolder?fid=%2540B%	10.1.180.106	DEFAULT_PARENT/127.-
1.237.535.466.674	383	6	TCP_MISS/20	24025	GET	http://www.google.com.tr/search?hl=tr&q=diziport+preason+l	10.1.180.106	DEFAULT_PARENT/127.text/html
1.237.535.466.750	7530	7	TCP_MISS/20	403465	POST	http://web.ebscohost.com/ehost/resultsadvanced?vid=4&hid=1(	10.1.55.21	DEFAULT_PARENT/127.-
1.237.535.466.766	209	8	TCP_MISS/20	1844	GET	http://imageserver.ebscohost.com/img/imageqv/small_thumb/3	10.1.55.21	DEFAULT_PARENT/127.-
1.237.535.466.777	2110	9	TCP_MISS/20	0	POST	http://salvador.ebuddy.com/dispatch	10.1.130.10	DEFAULT_PARENT/127.text/plain

Çizelge 5.1'de bu çalışma sırasında işlenen dosyanın bir bölümü örnek olarak verilmiştir. Gizlilik nedeniyle istemci makinelerin IP adresleri kırmızı sayılarla değiştirilmiştir. Çizelge 5.2'de ise çalışmada kullanılan veriden alınan gerçek bir satır ile vekil sunucu günlüğü yapısı açıklanmıştır.

Çizelge 5.2 Vekil Sunucu Satırı Alan Tanımları.

1237535434.903 330 10.1.70.60 TCP_MISS/200 23266 GET http://www.google.com.tr/search?q=replikas+konser&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:tr:official&client=firefox-a 10.1.70.60 DEFAULT_PARENT/127.0.0.1 text/html

Alan Adı	Açıklama	Değer
Zaman (Time)	İstemci soketinin (IP ve TCP adres birleşimi) kapandığı zamandır. Milisaniye duyarlılığında 1 Ocak 1970'den o zaman kadar geçen saniyelerin sayısıdır. UNIX işletim sistemine özgü formattadır	*1237535434.903*
Süre (Duration)	İstemci soketinin kabul edilmesi ile kapanması arasında geçen süredir. İsteğin milisaniye cinsinden sürdüğü zamandır	*330*
Ağ Bilgisayarı (Remote Host)	İstemcinin IP adresidir	*10.1.70.60*
Kod (Code)	İşlem sonucunu belirtir. İsteğin tipini, nasıl sağlandığını veya nasıl başarısız olduğuna ilişkin tanım bilgisidir	*TCP_MISS/200*
Byte Sayısı	İstemciye iletilen veri miktarıdır	*23266*
Metot	HTTP istek metodudur (POST/GET)	*GET*
URL	Erişilmek istenilen URL'dir	*http://www.google.com.tr/search?q=replikas+konser&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:tr:official&client=firefox-a*
rfc931	Kullanıcının giriş kimliğini (login name) tutar.Mevcut değilse eksi işareti yerleştirilir	*10.1.70.60*
Karşı Taraf Statüsü / Karşı Taraftaki Bilgisayar	Nesnenin nasıl ve nereden getirildiğinin tanımıdır	*DEFAULT_PARENT/127.0.0.1*
Tip	HTTP cevap başlığında görüldüğü şekliyle nesnenin içerik tipidir. Eğer mevcut değilse eksi işareti yerleştirilir	*text/html*

Bu çalışmada kullanılan veri, Maltepe Üniversitesi vekil sunucusu üzerinde günlük dosyaları halinde depolanmış 2011 yılına ait 6 aylık verilerdir. Kullanılan metin verisinin boyutu 3,2 GB' dır. Aynı veri üzerinde daha önce yapılan belirli bir çalışma

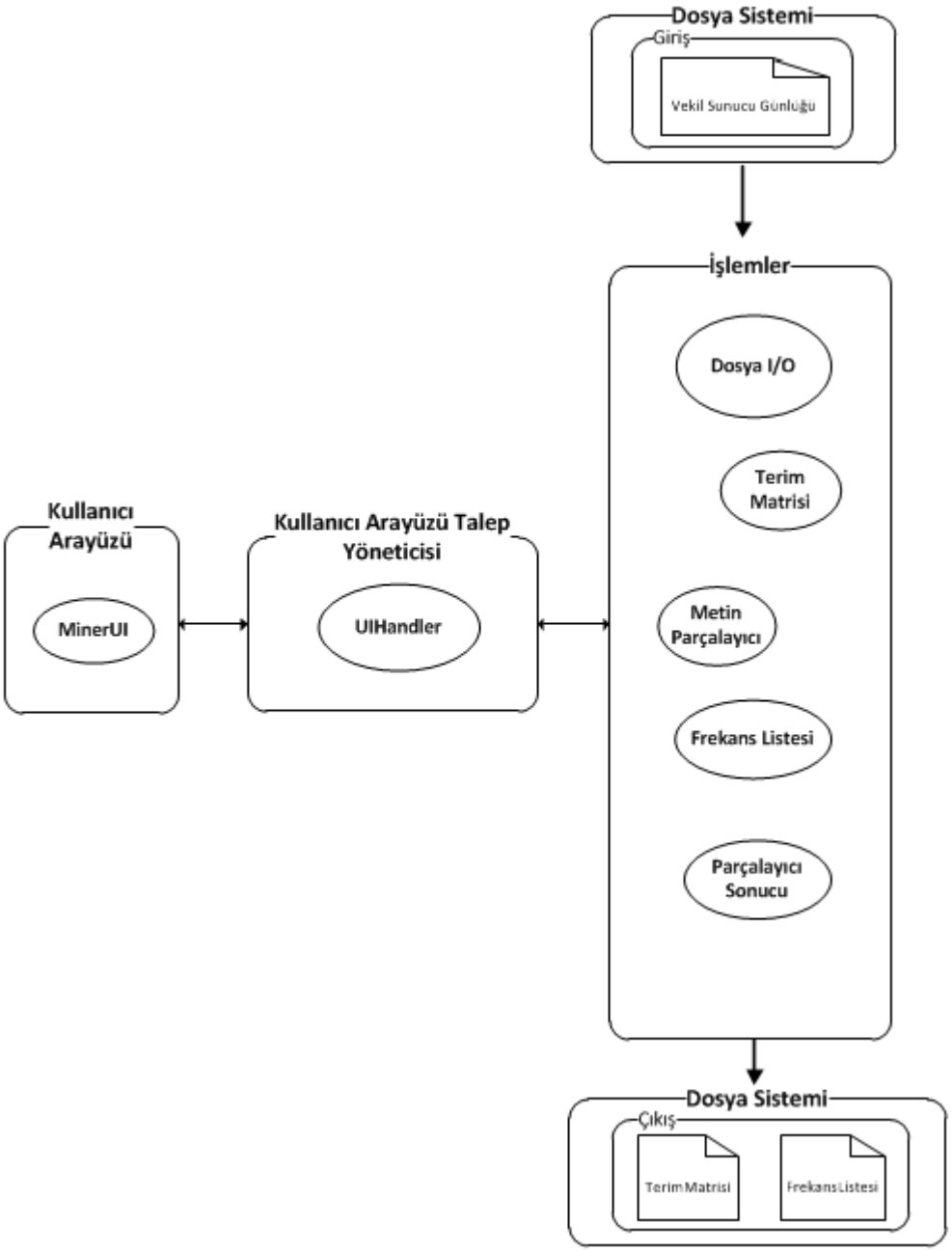
olmadığından ve literatürde kullanıcıları, yaptıkları sorgulara göre kümeleme çalışmasına rastlanmadığından karşılaştırma yapılabilecek sonuç bulunmamaktadır.

5.4 Uygulamanın Geliştirme Adımları

Uygulama Java programlama dilinde yazılan önişleme, Strehl kütüphanesinden faydalanılarak MATLAB üzerinde gerçekleştirilen kümeleme ve görüntüleme adımlarından oluşmaktadır.

5.4.1 Verinin Önişlemesi

Önişleme aşaması için Java ile geliştirilen kod üç katmandan oluşmaktadır. Şekil 5.8'de gösterilen basitleştirilmiş yapıda seviyeler ve her seviyedeki farklı görevlere sahip Java sınıfları görülmektedir. Önişleme aşamasında temel amaç günlük dosyası içindeki her satırda kullanıcı sorgusu yapılıp yapılmadığını anlamak ve eğer yapıldıysa kullanıcı sorgusunu parçalayıp tekil anahtar kelimelere ulaşıp bunları sonraki kümeleme aşaması için uygun bir yapıda depolamaktır. Seviyeler ve içeriği ile ilgili açıklamalar aşağıdadır.



Şekil 5.8 Vekil Sunucu Günlüğü Önişleme Modülü Mimarisi.

5.4.1.1 Kullanıcı Arayüzü

Kullanıcının süreci başlatıp önişleme için dosya belirleyebilmesine yardımcı olur. Önişleme sırasında üretilen frekans listesi ve metin parçalama ile ilgili dosyaların oluşturulması ile ilgili kullanıcının seçim yapmasına olanak verir. Son olarak da MATLAB tarafında çalıştırılan Strehl kütüphanesine ait fonksiyonların benzerlik matrisini kullanarak oluşturduğu kümeleme sonuçlarının tekrar işlenme sürecini başlatmak için kullanıcı bu arayüzü kullanır.

5.4.1.2 Kullanıcı Arayüzü Talep Yöneticisi

Kullanıcı arayüzünden gelen istekleri almaktan ve bir alt modüldeki sınıflar arasında argüman alışverişini yönetmekten sorumludur.

5.4.1.3 İşlemler

Bu katmanda önişleme sürecine katkıda bulunan sınıflar yer almaktadır. Sınıflar arası argüman alışverişleri çoğunlukla bu katmanda olmakla birlikte bir üst seviyedeki talep yöneticisi sayesinde de olabilmektedir. Bu seviyede bulunan sınıflar ve yaptıkları işlemler ile ilgili açıklamalar şu şekildedir:

TextParser(Metin Parçalayıcı): Satır olarak gelen günlük verisini parçalayarak ParserResult nesnelerine yerleştirir. Günlük dosyasında sadece www.google.com üzerinde yapılan aramalar ele alınmıştır. Uygulama, aranan URL'in değiştirilmesi gibi küçük değişikliklerle başka arama motorları için de kullanılır hale getirilebilir. Bu yüzden günlük dosyası satır satır okunmuş ve sadece "http://www.google.com.tr/search?" içeriğine sahip olan satırlar ele alınmıştır. Yine

aynı satırda bulunan IP değeri de kullanıcıları ayırt edici bir özellik olarak saklanmıştır.

*1237535391.305 748 10.1.180.106 TCP_MISS/200 122318 GET http://www.google.com.tr/search?hl=tr&q=**hal%C4%B1+temizli%C4%9Fi**&meta= 10.1.180.106 DEFAULT_PARENT/127.0.0.1 text/html*

Şekil 5.9 Günlük Dosyası Satır Örneği

Şekil 5.9'da görüldüğü gibi ilk IP değerini sakladıktan sonra "http" ve ikinci IP yani *rfc931* arasındaki metin yani URL, işlenecek veriyi kapsamaktadır. Şekilde görüldüğü gibi metin kodlamasından kaynaklanan sorunlar yaşanmıştır. URL kısmında bulunan arama metni UTF-8 kodlama şekline dönüştürülmüştür. Sonuçta elde edilen URL metni Şekil 5.10'daki gibi olmuştur:

*http://www.google.com.tr/search?hl=tr&q=**halı temizliği**&meta=*

Şekil 5.10 Parçalanan Günlük Satırı

IP ve URL kısmını vekil sunucu günlüğü satırından ayırabildikten sonra URL içinden aranılan kelimeleri çıkartılması aşamasına geçilmiştir. Metin içinde sorgulanan kelimelerin bulunduğu yer, www.google.com.tr/search? ifadesinden sonra "&q=", "?q=" veya "?as_q=" örüntülerini takip eden konumda yer almaktadır. Bahsedilen bu üç örüntü ve "&" işareti arasında kalan bölüm, aranan kelimelerin bulunduğu yer olarak değerlendirilmiş ve hem IP değeri hem de sorgu cümlesinden ayrılan bu kelimeler ilişkilerini koruyacak şekilde ParserResult nesneleri içinde saklanmıştır. Bu nesnelerden oluşan dizi daha sonra IP-Terim matrisi oluşturmada kullanılmıştır.

ParserResult(Parçalayıcı Sonucu): ParserResult nesnesi vekil sunucu günlüğünden alınıp parçalanan satırlar içindeki IP ve onunla ilişkili sorgu terimlerini tutar. URL içinde bulunan sorgu kelimeleri, kullanıcının giriş yaptığı şekilleriyle TextParser sınıfındaki fonksiyonlarda ayrıştırılmış, ParserResult sınıfı içinde ise temizlenmiş ve filtrelenmiştir.

Temizleme aşamasında sorgu kelimeleri içinde yanlışlıkla yazılan veya kümeleme işlemi açısından anlamı olmayan, kelime içlerine ve aralarına dağılan rakamlar ile noktalama işaretlerinin ayıklanması gerçekleştirilmiştir (örn. "m<altepe", "devexpress+crack+forum", "7 - 50x70").

Filtreleme işlemi sırasında ise öncelikle Türkçede "ve, de, ki...", İngilizcede "a, an, org, of, and" gibi çokça geçen ve üç harften küçük kelimeler elenmiştir. Bu işlem sonrasında elde kalan kelimelerden Türkçe olanların kökü Zemberek yardımı ile bulunmuştur. Bu durum ilerleyen aşamalarda terim matrisi oluşturulurken "güneşli", "güneş" ve "güneşlik" gibi aynı köke sahip kelimelerin farklı kelimeler gibi değerlendirilmemesini sağlamak amacıyla Türkçenin bitişken yapısından kaynaklanan bir gerekliliktir. Türkçe olmayan kelimelerde kök ayrıştırması gerçekleştirilmemiştir.

Terim Matrisi : Önceki adımlarda elde edilen temizlemiş ve köklerine indirgenmiş terimler(kelimeler) sütunlarını, IP'ler ise matrisin satırlarını oluşturmaktadır.

Matris oluşturulurken ilk olarak her birisi bir IP'yi tutacak şekilde satırlara IP'ler, her birisi bir sütunu temsil edecek şekilde de terimler atanmıştır. Daha sonra da önceki adımlarda oluşturulan ParserResult nesneleri tek tek dolaşılarak her IP'ye karşılık gelen terimin sayısını tutan matris hücresinin değeri bir arttırılmıştır. Bu şekilde oluşan yapı, metin parçalama, temizleme ve filtrelemeden sonra elde edilen verinin matrise yerleştirilmiş halidir. Sonrasında matris, satır bazında ve sütun

bazında taranarak belirli değerin altında toplamlara sahip satır ve sütunlar yani çok az aranmış kelimeler ile çok az arama yapmış IP'ler elenmiştir. Örneğin sütun toplamı üçten küçük bir kelime veya satır toplamı beşten küçük bir satır kümeleme sırasında ayrık değer(outlier) yaratacağından filtrelenmiştir. Benzer şekilde IP yani satır bazında toplamı 200 üzerinde olan satırlar yani IP'ler de elenmiştir. Bu eleme işleminde hangi aralıktaki değerlerin saklanacağını belirlerken FrequencyList nesnesinin ürettiği kelimelerin toplam kullanım sayısından ve matris ilk oluşturulurken hesaplanan satır ve sütun toplamlarından faydalanılmıştır. Sonuçta ortaya çıkan matrisin bir bölümü Çizelge 5.3'de görülmektedir.

Çizelge 5.3 IP-Terim Matrisi.

	cumhuriyet	makale	resmi	gazete	halı	temiz	konser	pegasus	diziport	break	sezon	bölüm	yık	yardım	hazırla	facebook	flash
10_1_30_61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
10_1_145_84	0	0	0	9	20	0	0	0	0	0	0	0	0	0	0	0	3
10_1_200_116	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	1
10_1_70_60	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
10_1_30_209	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10_1_40_56	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
10_1_190_57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10_1_70_39	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
10_1_30_62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	21	0
10_1_40_16	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28
10_1_30_95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

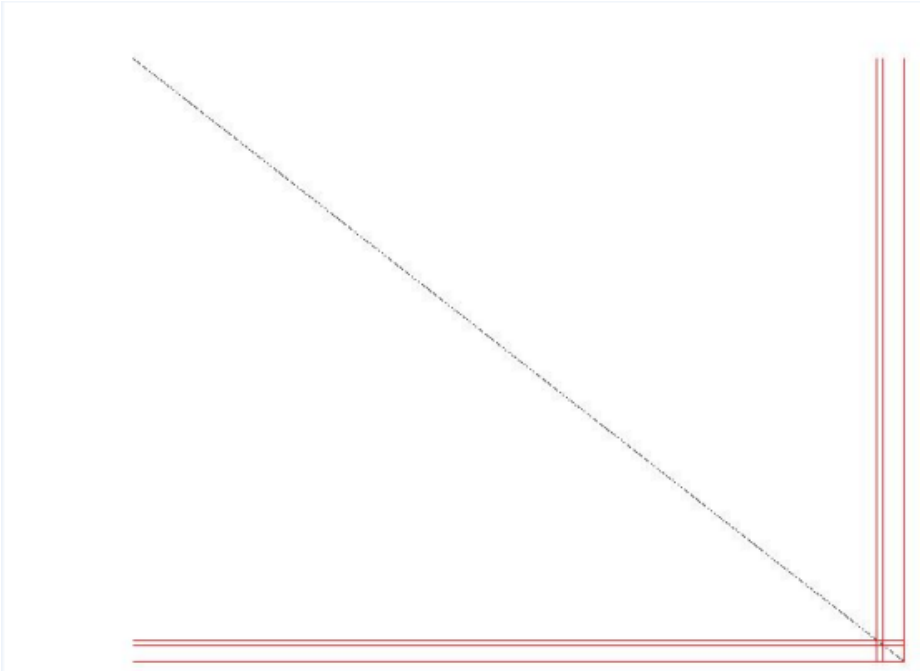
Çizelge 5.3'de satırbaşları IP, sütunlar ise aranan kelimeleri göstermektedir. Matrisin her hücresi satırdaki IP'nin ilgili sütundaki kelimeyi kaç defa aradığını göstermektedir.

5.4.2 Verinin Kümelenmesi ve CLUSION ile görselleştirme

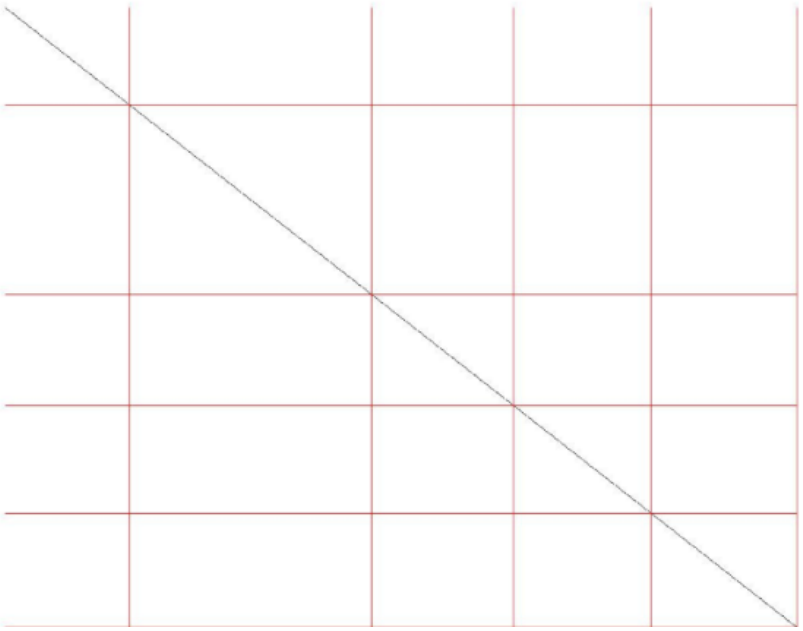
Çizelge 5.3'de gösterilen matris içeriği, MATLAB içine aktarılarak üzerinde Strehl kütüphanesi fonksiyonları çalıştırılmıştır [60]. MATLAB üzerinde ilk önce Strehl kütüphanesinde bulunan benzerlik fonksiyonları *simcosi* (Kosinüs uzaklığı), *simcorr* (Pearson korelasyonu), *simeucl* (Euclid Uzaklığı) ve *simxjac* (Uzatılmış Jaccard) ile IP-Terim matrisi benzerlik matrisine dönüştürülmüştür. Ortaya çıkan benzerlik

matrisi yine kütüphanede bulunan hiyerarşik k-means, toplamalı (agglomerative) kümeleme ve graf tabanlı kümeleme yöntemleri olan "kenar ağırlıklı graf bölümlemeli" *cgraph* ve "kenar ağırlıklı değer dengelemeli" *clcgraph* algoritmaları ile kümelenmiştir. Kümeleme için k değeri olarak sırasıyla 3,4,5 ve 6 denenmiştir.

Genel olarak Öklid benzerliği ile oluşturulan matrislerin tüm kümeleme yöntemleri ile çalışılmasında belirgin kümeler oluşmamıştır. Şekil 5.11 ve 5.12, k =3 ve k = 5 olarak alınarak K-Means yöntemi ve "kenar ağırlıklı değer dengelemeli – *clcgraph*" ile yapılan kümeleme sonuçlarının CLUSION grafiklerini göstermektedir. Şekillerde de görüldüğü gibi herhangi bir küme oluşumu köşegen boyunca görülmemektedir.

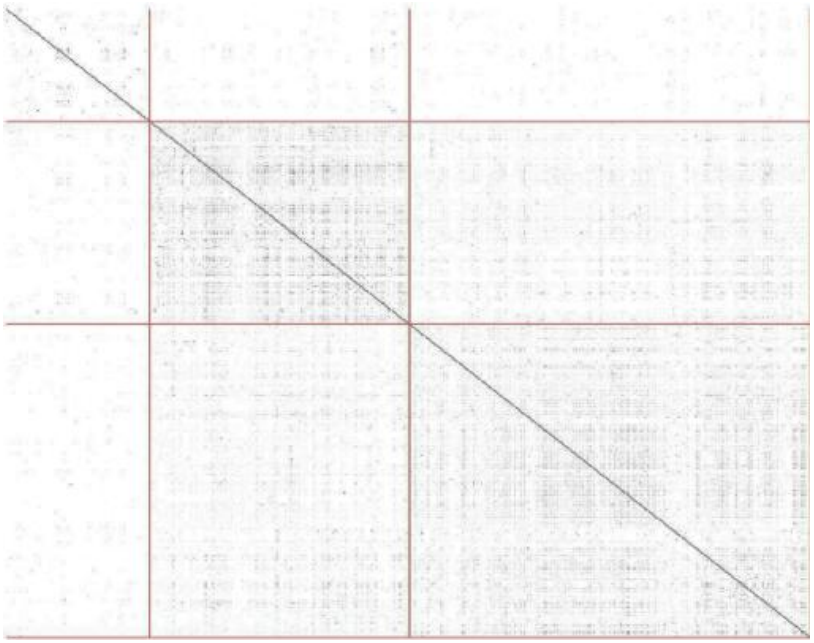


Şekil 5.11 Öklid benzerlik matrisinin K-Means kümelemesi (k = 3)



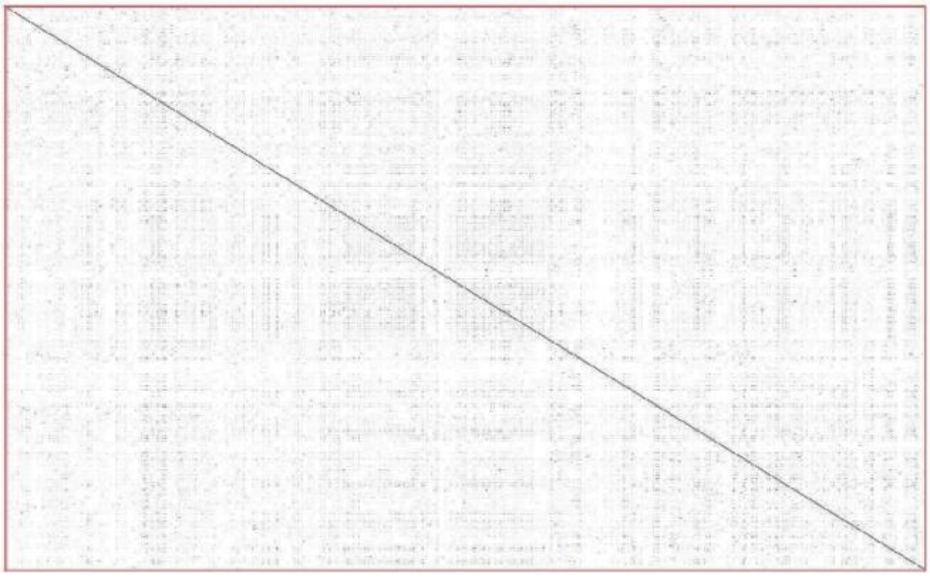
Şekil 5.12 Öklid benzerlik matrisinin *clcgraph* kümelemesi (k = 5)

Uzatılmış Jaccard (simxjac) ile oluşturulan benzerlik matrisinin farklı k değerleri ve farklı kümeleme yöntemleri ile oluşan CLUSION görselleri aşağıdadır. Şekil 5.13 ve 5.14'de görüldüğü gibi Jaccard benzerliği K-Means ve toplamalı kümeleme yöntemlerinde(agglomerative hierarchical clustering) köşegen boyunca belirgin yoğunluklar oluşturmamıştır. Graf Tabanlı kümeleme yöntemi olan *clcgraph* ise aynı k değeri için görece belirgin kümelerle Jaccard benzerliği üzerinde daha iyi performans göstermiştir. Genel anlamda ise Jaccard benzerliği ile yapılan kümelemeler diğerlerine göre başarısız olmuştur.

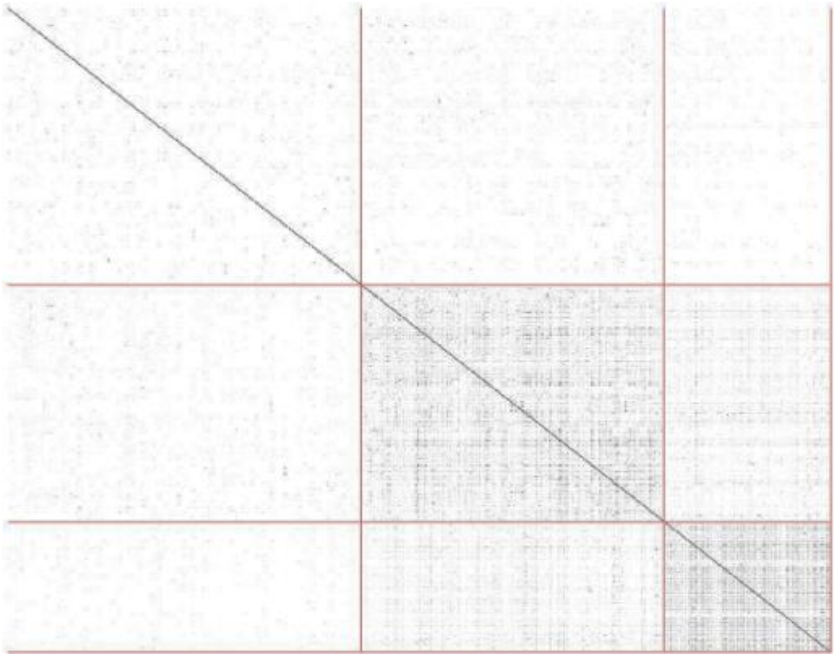


Şekil 5.13 Uzatılmış Jaccard benzerlik matrisinin K-Means ile kümelenmesi (k = 3)

Şekil 5.14'de görüldüğü gibi "toplamalı" yöntemle yapılan kümelemede herhangi bir küme bulunamamıştır. Uzatılmış jaccard benzerlik matrisinin toplamalı yöntemle kümelenmesinde tüm k değerleri için benzer CLUSION grafiği elde edilmiştir.



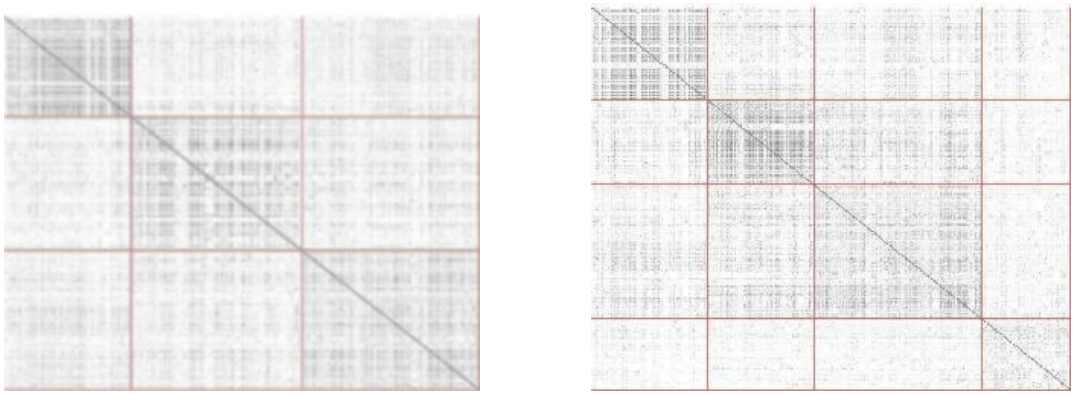
Şekil 5.14 Uzatılmış Jaccard benzerlik matrisinin Toplamalı kümelenmesi (k = 3)



Şekil 5.15 Uzatılmış Jaccard benzerlik matrisinin Graf Tabanlı Kümelemesi (k = 3)

Şekil 5.15'de görüldüğü gibi graf tabanlı kümeleme ile k= 3 için köşegen üzerinde birbirine yakın ve önceki kümeleme yöntemlerine göre daha belirgin küme elde edilmiştir.

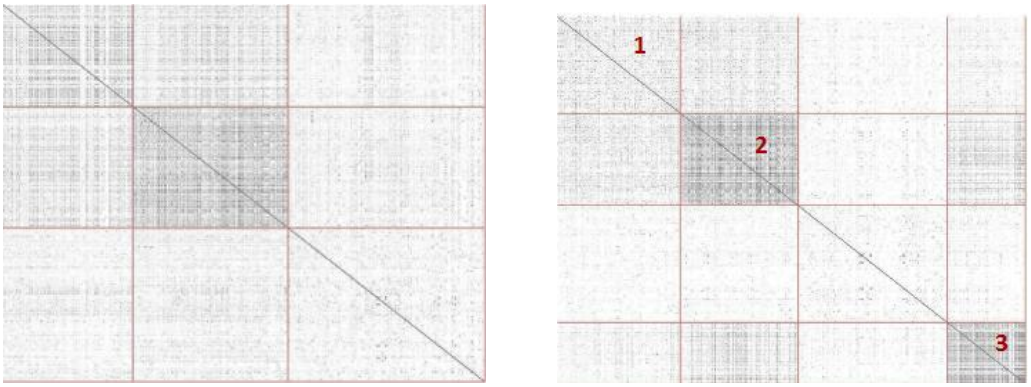
Pearson Korelasyon katsayısı ve Kosinüs uzaklığı ile elde edilen benzerlik matrislerinin K-Means ve "kenar ağırlıklı değer dengelemeli – *clcgraph*" ile kümelenmesi en net kümeleri oluşturmuştur.



(a) (b)

Şekil 5.16 K - Means ile kümeleme (a) k = 3 Pearson Korelasyonu (b) k = 4 Kosinüs uzaklığı

Şekil 5.16 (a)'da K-Means ile köşegene uzak bölgelerde seyrek dağılımlar olsa da köşegen üzerinde görece yoğun bölgeler oluşmuştur. Burada üç tane kümenin varlığından ve bu kümelere girmeyen dağınık veriden bahsedilebilir. (b) kısmında ise kosinüs uzaklığı ile elde edilen benzerlik matrisinin k = 4 için K-Means kümelemesi ile elde edilen iki yoğun küme bir tane görece daha az yoğun ama büyük küme ve bir tane de küçük ve seyrek bir küme görülmektedir.



(a) (b)

Şekil 5.17 Graf tabanlı *clcgraph* ile kümeleme (a) k = 3 Pearson Korelasyonu (b) k = 4 Kosinüs uzaklığı

Şekil 5.17 (a)'da köşegen üzerindeki yoğun bölgelerden de görüleceği gibi iki tane yoğun küme ve birbirine benzemeyen veri elemanlarının oluşturduğu görece daha büyük bir veri yığını, (b)'de ise yine 2 ve 3 ile numaralandırılmış iki tane yoğun küme elde edilmiştir. 1 numaralı bölge ise 2 numaralı küme elemanlarına aynı kümeye girebilecek kadar benzemeyen ama yakın benzerlik değerlerine sahip elemanların oluşturduğu daha seyrek bir kümeyi göstermektedir.

6. SONUÇ

Bu bölümde tez çalışması süreci sonucunda ortaya çıkan kümeler, içerdikleri terimlerle birlikte gösterilmiş, oluşan grupların karakteristiği incelenmiş ve sonuçlar yorumlanmıştır.

6.1 Değerlendirmeler

Tez çalışması kapsamında gerçekleştirilen deneylerde hiyerarşik kümeleme yöntemlerinden olan toplamalı kümeleme, farklı benzerlik ölçüleri ile denenmesine rağmen belirgin kümeler oluşturmamıştır. Aynı benzerlik ölçüleri ile K-Means daha belirgin kümeler oluşturmuştur. En iyi kaliteye sahip kümeler kosinüs uzaklığı ve korelasyon katsayısı benzerliği metriklerinin, "kenar ağırlıklı değer dengelemeli graf bölümleme" algoritması ile birlikte kullanılmasıyla elde edilmiştir. Elde edilen sonuçların kalitesi CLUSION grafikleri yardımı ile gözlemlenmiştir.

Gözlemlerde en iyi kümelenme sağlayan Şekil 5.17 b'deki CLUSION grafiğinden her bir kümedeki IP adresleri elde edilmiştir. Kümeleri oluşturan IP adresleri ve aradıkları terimler tablolar halinde Çizelge 6.1, 6.2 ve 6.3'de verilmiştir. Kümelerin eleman sayılarının çok fazla olması sebebiyle küçük bir kısmı tablolarda görülmektedir. Çizelgelerde arama terimleri arama sıklığına göre sıralı değildir.

Çizelge 6.1 Küme 1'e ait IP'ler ve aradıkları terimler.

KÜME 1	
10.1.30.61	facebook, systems, face....
10.1.145.84	halı, sigorta, identity,birkiye....
10.1.70.60	pegasus....
10.1.30.209	face, keygen...
10.1.70.39	pegasus, canlıdizi....
10.1.30.62	facebook, face....
10.1.30.95	mehmet, hava...
10.1.80.38	ankara,mehmet, kimlik, birkiye....
10.1.190.26	vitra, with, radio, face, kitap....
10.1.80.49	bebek, canlıdizi, dizi, kuram,.....
10.1.145.113	canlıdizi, tubitak, with, ankara,....
…	…

Çizelge 6.2 Küme 2'ye ait IP'ler ve aradıkları terimler.

KÜME 2	
10.1.200.242	bölüm, yık, izle, para, sözlük, tubitak, uludağ, hava, earth,…
10.1.201.77	identity, ilhan, samanyolu, tabir, sonuç, radio, albüm, ...
10.1.190.19	vitra, çevre, tubitak, ilhan, tabir, sonuç, radio, albüm,.....
10.1.200.54	bebek, para, üniversite, hava, radio, tooltip, ...
10.1.200.32	bebek, identity, sözlük, tubitak, ankara, uludağ, radio, türk,...
10.1.145.70	bölüm, flash, bebek, para,ev, uludağ, hava, yayın,...
10.1.200.140	flash, para, ingilizce, tubitak, sosyal, açıklama, computer, ...
10.1.200.82	bebek, izle, para, ingilizce, sözlük, tubitak, ilhan, sonuç,...
10.1.201.10	diziport, star, sosyal, ev, hava, öğretmen, earth,....
10.1.200.122	identity, para, üniversite,hava, radio, açıklama, itiraf,.....
10.1.200.181	break, bebek, identity, izle, para, ilhan, ankara, uludağ, hava
10.1.145.99	bebek, çevre, izle, para, samanyolu, aids, açıklama,.....
10.1.200.34	para, uludağ, radio, shutterstock, itiraf,.....
10.1.145.43	halı, vitra, izle, ankara, computer, marmara, kurum
…	…

Çizelge 6.3 Küme 3'e ait IP'ler ve aradıkları terimler.

KÜME 3	
10.1.200.116	halı, flash, bebek, bilkent, program, between, ...
10.1.40.56	halı, bebek, çevre, sorun, ankara, aids, indir,…
10.1.40.16	flash, maltepe, bebek,vitra, çevre, tubitak, ankara, aids,...
10.1.200.136	maltepe, bebek, star, sigorta, ankara, milli, indir, ...
10.1.30.87	bebek, çocuk, anadolu, piyango, seramik,....
10.1.70.53	flash, canlıdizi,...
10.1.145.93	gazete, halı, vitra, aids, ...
10.1.190.80	vitra, çevre, sorun, nezih, systems,yeditepe, piyango,....
10.1.190.32	flash, bebek, çocuk, şiddet, translate,...
10.1.30.39	flash, bebek, çevre, ankara, şirket
10.1.30.104	gazete, halı, bebek, vitra, ankara, sosyal, piyango,...
10.1.145.167	bebek, star, sonuç, aids, indir, between
10.1.145.42	bebek, çevre, tubitak, merkezi, yeditepe, between,.....
…	…

Kullanıcıların yaptıkları aramalardan elde edilen terimlerin büyük bir bölümü genel terimlerden oluştuğu için bu yaygın terimlerin filtrelenmesine ihtiyaç duyulmuştur. Filtreleme ölçütü olarak ilgili terimin Google arama motorundan gelen indekslenme sayısı kullanılmıştır. Google'ın indekslediği web sitelerinde 100.000.000'dan daha fazla sayıda kullanılan kelimeler genel terim olarak kabul edilmiştir ve bu terimler kümelerden çıkartılmıştır. Genel olmayan terimleri içeren kümeler Çizelge 6.4, 6.5 ve 6.6'da verilmiştir.

Çizelge 6.4 Küme 1'e ait terimler

KÜME 1	
Aranan Kelime	**Google'da indekslenme sayısı**
sulhi	449.000
tubitak	497.000
diziport	568.000
bazal	752.000
canlıdizi	1.640.000
akyıldız	2.300.000
inönü	3.130.000
açıköğretim	5.830.000
kuram	9.830.000
vitra	10.100.000
pansiyon	11.400.000
ayar	13.100.000
ilhan	18.900.000
uludağ	19.000.000
psikolojik	21.900.000
kurum	23.000.000
tuzla	25.900.000
tooltip	32.400.000
redd	33.500.000
marmara	36.700.000
öldür	38.000.000
bölge	40.600.000
hava	41.200.000
çevre	42.400.000
ingilizce	62.600.000
belge	68.800.000
aşık	70.200.000
halı	81.000.000
anadolu	81.000.000
medya	83.800.000
dizi	87.800.000
bebek	93.500.000

Çizelge 6.5 Küme 2'ye ait terimler

KÜME 2	
Aranan Kelime	**Google'da indekslenme sayısı**
sulhi	449.000
bazal	752.000
iktisadi	5.090.000
tubitak	5.510.000
tabir	8.720.000
vitra	10.100.000
seramik	11.700.000
küresel	12.500.000
istanbulun	16.400.000
ilhan	18.900.000
uludağ	19.000.000
erçetin	20.300.000
tepe	21.700.000
psikolojik	21.900.000
tuzla	25.900.000
melek	30.100.000
dönem	35.000.000
öldür	38.000.000
bölge	40.600.000
farid	41.200.000
öğretmen	41.500.000
çevre	42.400.000
yık	44.100.000
güçlü	45.300.000
sözlük	55.700.000
anket	60.000.000
fragman	60.700.000
shutterstock	61.000.000
ingilizce	62.600.000
güney	65.500.000
durum	66.200.000
tasarım	67.800.000
belge	68.800.000
yayın	70.700.000
beşiktaş	78.800.000
halı	81.000.000
söz	82.200.000
açıklama	87.700.000
bebek	93.500.000
kitap	99.300.000

Çizelge 6.6 Küme 3'e ait terimler

KÜME 3	
Aranan Kelime	**Google'da indekslenme sayısı**
canlıdizi	34.100
diziport	118.000
bazal	175.000
kamilkoç	185.000
tubitak	971.000
bilkent	1.040.000
kuram	1.780.000
sulhi	2.680.000
küresel	2.850.000
uludağ	2.870.000
şiddet	2.990.000
alyans	3.240.000
istanbulun	3.370.000
iett	3.440.000
nezih	3.810.000
lisans	5.360.000
tiyatro	7.000.000
marmara	7.350.000
güçlü	7.780.000
tabir	8.720.000
dönem	8.760.000
shutterstock	9.320.000
vitra	9.990.000
seramik	11.700.000
parça	11.900.000

KÜME 3	
Aranan Kelime	**Google'da indekslenme sayısı**
durum	14.400.000
yayın	16.000.000
çanakkale	16.500.000
halı	17.800.000
ilhan	18.900.000
zeka	20.200.000
kurum	20.800.000
belge	23.500.000
tepe	24.500.000
tuzla	25.900.000
kolej	28.900.000
açıklama	29.600.000
tooltip	31.600.000
şirket	33.600.000
kraloyun	36.600.000
öğretmen	36.700.000
bölge	37.500.000
başkan	41.200.000
ankara	52.000.000
tasarım	59.800.000
abstracts	65.300.000
redtube	86.300.000
oku	86.600.000
dizi	88.000.000
bebek	93.500.000

Sonraki aşamada, aynı anda birden fazla kümede bulunan terimler, net bir ayrıştırma sağlamadıkları için kümelerden çıkartılmıştır. Elde edilen son kümeler ve içerdiği terimler Çizelge 6.7'de verilmiştir.

Çizelge 6.7 Kümeleri tanımlamak için kullanılan terimler

KÜME 1	**KÜME 2**	**KÜME 3**
akyıldız	iktisadi	kamilkoç
inönü	erçetin	bilkent
açıköğretim	melek	kuram
kuram	dönem	şiddet
pansiyon	farid	alyans
ayar	yık	iett
redd	sözlük	nezih
hava	anket	lisans
aşık	fragman	tiyatro
anadolu	güney	dönem
medya	beşiktaş	parça
	söz	çanakkale
	kitap	zeka
		kolej
		şirket
		kraloyun
		başkan
		ankara
		abstracts
		redtube
		oku

Küme1 ele alındığında göze çarpan kelimelerden iki tanesi "anadolu" ve "açıköğretim"dir. Üniversite kampüsü içinde bu anahtar kelimelerin kullanılması, akla arama yapan kişilerin üniversite mezunu olmayan veya mevcut yüksek öğrenim derecelerinin üzerine açıköğretimde okuyan "çalışanları" getirmektedir. "Redd" (bir Türk rock grubu) ve "aşık" kelimeleri yaş grubu olarak orta yaş altını işaret etmekte hem de ilk önermeyi bu açıdan desteklemektedir. Küme2 grubu için {"kitap", "erçetin", "fragman"} kelimeleri kültürel aktivitelere merakı, {"iktisadi", "sözlük", "dönem", "anket"} kelimeleri ise bu gruptaki insanların akademik hayata bağlantısını işaret etmektedir. Küme3'te yer alan {"bilkent", "ankara", "kamilkoç"} kelimeleri Ankara ile ilgileri olan insanların bu kümede yer aldığını, yine "kamilkoç" ve "iett" kelimeleri ait oldukları gelir grubunu işaret etmektedir. {"nezih" (kitabevi), "lisans", "abstracts", "dönem", "kuram"} kelimeleri ise bu gruptaki insanların akademik dünya ile ilişkileri olabileceğini akla getirmektedir. Küme2 ve Küme3 genel

hatlarıyla benzer karakterler sergilese de {“kraloyun”, “redtube”} gibi eğlence ve cinsel içerikli site adlarının Küme3 içindeki kişiler tarafından aranmasından dolayı farklılaşmaktadırlar.

6.2 Öneriler

Oluşturulan sistemde günlük dosyası, geliştirilen yazılım ile çözümlendikten sonra Strehl kütüphanesi kullanılarak MATLAB ortamında kümelenmekte, kümeleme sonuçları metin dosyası aracılığı ile tekrar uygulamaya aktarılarak kümelerin içeriği görüntülenmektedir. Gelecekteki çalışmalarda, farklı ortamlarda ara transfer dosyaları kullanılarak yapılan bu işlemler yerine akışın bozulmadığı, tümleşik bir yapı geliştirilmeye yönelik çalışmalar gerçekleştirilecektir.

Bu çalışmada kullanılan günlük dosyası geçmişteki belirli bir zaman aralığına ait verileri analiz etmektedir. Sürekli olarak büyüklüğü artan dosyayı incelemek istendiğinde tümünü ele almak zorunda kalınmaktadır. Daha verimli olan yaklaşım günlük dosyasını artımlı (incremental) olarak inceleyebilmektir. Bu alanda uğraş verecek araştırmacılara günlük dosyasının artımlı analizi konusunda çalışmaları önerilmektedir. Ayrıca, araştırmacılara, bu çalışmadaki benzerlik ölçülerini ve graf tabanlı algoritmaları, literatürdeki farklı yöntemler ile karşılaştırarak performans değerlendirmesi yapmaları önerilmektedir.

KAYNAKLAR

1. Han J. Kamber M., "Data Mining Concepts and Techniques, Second Edition", Morgan Kaufmann, ISBN 13: 978-1-55860-901-3, San Francisco, 2006.

2. Daniel A. Keim, "Information Visualization and Visual Data Mining", IEEE Transactions on Visualization and Computer Graphics, Vol. 7, N0. 1, January-March 2002

3. http://www.gartner.com/technology/it-glossary/#3_0 (Erişim Tarihi: 02.02.2012)

4. Peter Cabena, Pablo Hadjinian, Rolf Stadler, Jaap Verhees, and Alessandro Zanasi, Discovering Data Mining: From Concept to Implementation, Prentice Hall, Upper Saddle River, NJ, 1998

5. Ron Kohavi, "Data Mining and Visualization", National Academy of Engineering(NAE), US Frontiers of Engineering 2000

6. Osmar R.Zaiane, "CMPUT690 Principles of Knowledge Discovery in Databases", University of Alberta Department of Computer Sciences, 1999

7. Maniatty William A., Zaki Mohammed J., "Systems Support for Scalable Data Mining", SIGKDD Explorations, December 2000

8. Ricardo Baeza-Yates,Aristides Gionis,"An Introduction to Web Mining-Part -1", Yahoo! Research, ECML/PKDD 2008 Antwerp

9. Sankar K.Pal, Varun Talwar, Pabitra Mitra, "Web Mining in Soft Computing Framework:Relevance, State of the Art and Future Directions", IEEE Transactions on Neural Networks, Vol.13, No.5, September 2002

10. Brijendra Singh, Hemant Kumar Singh, "Web Data Mining Research: A Survey",Department of Computer Sceience,University of Lucknow, 2010 IEEE 978-1-4244-5967-4

11. Qingyu Zhang and Richard s. Segall," Web mining: a survey of current research,Techniques, and software", in the International Journal of Information Technology & Decision Making Vol. 7, No. 4 (2008) 683– 720

12. Kosala and Blockeel, "Web mining research: A survey," SIGKDD:SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining, ACM, Vol. 2, 2000

13. O.Etzioni. "The World Wide Web: Quagmire or Gold Mining", Communicate of the ACM, (39)11:65-68, 1996;

14. N. Kushmerick, “Gleaning the web,” IEEE Intell. Syst. , vol. 14, no. 2, pp. 20–22, 1999

15. D. Freitag, “Information extraction from html: Application of a gen-eral machine learning approach,” in Proc. 15th Conf. Artificial Intell.AAAAI-98, 1998, pp. 517–52

16. O. Etzioni, J. Shakes, and M. Langheinrich, “Ahoy! the homepage finder,” presented at the Proc. 6th WWW Conf., Santa Carla, CA, Apr.1997.

17. http://www.ra.ethz.ch/cdstore/www6/technical/Paper039/Paper39.html (Erişim Tarihi: 05.02.2012)

18. J. Hipp, U. Guntzer, and J. Nakhaeizadeh, “Algorithms for assocation rule mining a general survey and comparison,” ACM SIGKDD Explo-rations , vol. 2, pp. 58–65, July 2000

19. Jaideep Srivastava,Robert Cooley,Mukund Deshpande, Pang-Ning Tan, “Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data”, University of Minnesota, SIGKDD Explorations Jan 2000

20. Chun-Ling Zhang, Zun-Feng Liu, Jing-Rui Yin, “The Application Research on Web Log Mining in E-Marketing”, Hebei Polytechnic University, 978-1-4244-5895-0 IEEE 2010

21. Margaret H. Dunham, “Data Mining Introductory & Advanced Topics”, Pearson Education 2003, ISBN-10**:** 8177587854

22. B. Mobasher, “Web Usage Mining (Invited Chapter)”, *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data (by Bing Liu)* Springer Berlin-Heidelberg, 2006, ISBN-10**:** 3540378812

23. Osmar R.Zaiane, “Web Usage Mining for a Better Web-Based Learning Environment”, Department of Computing Science, University of Alberta

24. H. A. Edelstein, Pan for Gold in the Clickstream, Informationweek, March 2001, http://www.informationweek.com/828/mining.htm (Erişim Tarihi: 23.01.2012)

25. R. Agrawal and R. Srikant., “Fast algorithms for mining association rules”. In Proc.ofthe 20thVLDBConfer-ence, pages 487-499, Santiago, Chile, 1994

26. Chu-Hui Lee, Yu-Hsiang Fu, “Web Usage Mining based on Clustering of Browsing Features”, Eighth International Conference on Intelligent Systems Design and Applications, IEEE 2008

27. U. Fayyad, G. Piatetsky-Shapiro, and P.Smyth. “From data mining to knowledge discovery: An overview”, In Proc. ACM KDD, 1994

28. Charabarti, Soumen, “Mining The Web-Discoevring Knowledge from HyperText Data”, The Morgan Kaufmann Publishers, 2003, ISBN: 1-55860-754-4

29. R. Srikant and R. Agrawal, “Mining sequential patterns: Generalizations and performance improvements”, Pmc. of the Fifth In17 Conference on Extending Database Technology, Avignon, France, 1996

30. Liu Jian-guo, Huang Zheng-hong , Wu Wei-ping, “Web Mining for Electronic Business Application”, 0-7803-7840-7/03 – IEEE 2003

31. Tasawar Hussain, Dr. Sohail Asghar, Dr. Nayyer Masood, “Web Usage Mining: A Survey on Preprocessing of Web Log File”, Department of Computer Science, Muhammad Ali Jinnah University, Islamabad, Pakistan

32. Murata, T. and K. Saito (2006). Extracting Users' Interests from Web Log Data. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06) 0-7695-2747-7/06

33. Stermsek, G., M. Strembeck, et al. (2007). A User Profile Derivation Approach based on Log-File Analysis. IKE 2007: 258-264

34. E. Cohen, B. Krishnamurthy, and J. Rexford. “Improving end-to-end performance of the web using server volumes and proxy filters”. InProc. ACM SIGCOMM,pages 241-253, 1998

35. Mobasher, B., Cooley, R., & Srivastave, J., “Automatic Personalization based on Web Usage Mining”, KDD99 workshop on web usage analysis and user profiling (WEBKDD’99). Aug. San Diego, CA. ACM

36. Cheng Yu,Xiong Ying, “Application of Data Mining Technology in E-Commerce”, 2009 International Forum on Computer Science Technology and Applications, IEEE 2009, 978-0-7695-3930-0/09

37. Wen-Hai Gao, “Research on Client Behaviour Pattern Recognition System Based onWeb Log Mining”, Proceedings of the Ninth International Conference on Machine Learning and Cybernetics, Qingdao, 11-14 July 2010

38. Mehmed Kantardzic, “Data Mining – Concepts, Models,Methods and Algorithms” , Wiley-IEEE Press; 1 edition (October 25, 2002), ISBN -10 : 0471228524

39. Rui Xu , Donald Wunsch II, “Survey of Clustering Algorithms”, IEEE Transactions on Neural Networks, Vol. 16, No. 3, May 2005

40. Pang-Ning Tan,Michael Steinbach,Vipin Kumar, “Introduction to Data Mining”, Pearson International Education, ISBN 0-321-42052-7

41. Strehl, Alexander, “Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining”, 2002 Doctoral Dissertation, University of Texas

42. M. Anderberg, Cluster Analysis for Applications . New York: Aca-demic, 1973

43. A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review, " IEEE Trans. Pattern Anal. Mach. Intell. , vol. 22, no. 1, pp. 4 –37, 2000.

44. A. Jain, M. Murty, and P. Flynn, "Data clustering: A review, " ACM Comput. Surv. , vol. 31, no. 3, pp. 264 –323, 1999

45. C. Bishop, Neural Networks for Pattern Recognition. New York: Ox-ford Univ. Press, 1995

46. I. K. Ravichandra Rao, "Data Mining and Clustering Techniques", DRTC Workshop on Semantic Web 8th – 10th December, 2003 DRTC, Bangalore

47. http://people.revoledu.com/kardi/tutorial/Similarity/WhatIsSimilarity.html (Erişim Tarihi :10.09.2011)

48. http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Euclidean_and_Euclidean_Squared_Distance_Metrics.htm (Erişim Tarihi :10.10.2011)

49. http://en.wikipedia.org/wiki/Minkowski_distance (Erişim Tarihi :09.11.2011)

50. http://people.revoledu.com/kardi/tutorial/Similarity/MinkowskiDistance.html (Erişim Tarihi :10.109.2011)

51. http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Manhattan_Distance_Metric.htm (Erişim Tarihi: 10.12.2011)

52. http://en.wikipedia.org/wiki/Cosine_similarity (Erişim Tarihi: 24.12.2011)

53. http://people.revoledu.com/kardi/tutorial/Similarity/AngularSeparation.html (Erişim Tarihi: 10.02.2011)

54. Ying Zhao, George Karypis,"Evaluation of Hierarchical Clustering Algorithms for Document Datasets", Department of Computer Science and Engineering University of Minnesota, 2002

55. JinHuaXu, HongLiu, "Web User Clustering Analysis based on KMeans Algorithm", 2010 International Conference on Information, Networking and Automation (ICINA)

56. http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/K-Means_Clustering_Overview.htm (Erişim Tarihi: 21.01.2012)

57. Zheng Chen, Heng Ji, "Graph -based Clustering for Computational Linguistics: A Survey", Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing, ACL 2010, pages 1–9,Uppsala, Sweden, 16 July 2010.

58. van Dongen. 2000. Graph Clustering by Flow Si-mulation. PhD thesis, University of Utrecht.

59. http://code.google.com/p/zemberek/ (Erişim Tarihi: 08.10.2011)

60. http://strehl.com/soft.html (Erişim Tarihi: 10.10.2011)

61. http://www.eclipse.org/ (Erişim Tarihi: 10.10.2012)

62. http://en.wikipedia.org/wiki/MATLAB (Erişim Tarihi: 10.10.2012)

63. Ahmet Afşin Akın, Mehmet Dündar Akın, "Zemberek, an open source NLP framework for Turkic Languages", http://code.google.com/p/zemberek/ downloads/detail?name=zemberek_makale.pdf ,14.01.2012

64. A. Strehl, J. Ghosh and R. Mooney, "Impact of Similarity Measures on Web-page Clustering", Proc. of the 17th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search (AAAI 2000), July 2000, Austin, Texas

65. A. Strehl and J. Ghosh, "Relationship-based Clustering and Visualization for High-dimensional Data Mining", INFORMS Journal on Computing, pages 208-230, Spring 2003

66. G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal of Scien-tific Computing, 20(1):359–392, 1998.

67. Athena Vakali, George Pallis, "Web Data Management Practices:Emerging Techniques and Technologies", Idea Group Inc. 2007

68. http://en.wikipedia.org/wiki/Dendrogram (Erişim Tarihi: 13.02.2012)

69. http://en.wikipedia.org/wiki/Distance_matrix (Erişim Tarihi: 03.03.2012)

.

ÖZGEÇMİŞ

Mustafa Koray Aytekin, 1977 yılında Ankara'da doğdu. Öğrenimlerini sırasıyla Üsküdar Paşakapısı İlkokulu, Kadıköy Anadolu Lisesi (Ortaokul), İstanbul Atatürk Fen Lisesi'nde tamamladı. 1995 yılında Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği bölümünü kazandı ve 2001 yılında mezun oldu. Haziran 2002 –Nisan 2004 arasında Fintek A.Ş.'de, Nisan 2004 – Temmuz 2005 arasında Siemens A.Ş.'de Yazılım Mühendisi pozisyonlarında çalıştı. Ağustos 2007'den bu yana Roche A.Ş Türkiye'de çalışmakta 2011 Haziran ayından bu yana Doğu Avrupa, Ortadoğu bölgesi E-Pazarlama Proje Yöneticisi olarak görev yapmaktadır. Maltepe Üniversitesi'de 2009 yılında MBA tamamlamıştır. Eylül 2009'da Maltepe Üniversitesi'nde başladığı Bilgisayar Mühendisliği Yüksek Lisans programına tez aşamasında devam etmektedir.