FİYAT KARŞILAŞTIRMALI ÜRÜN ARAMA MOTORU GELİŞTİRME

FURKAN GÖZÜKARA

MERSİN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

MERSİN
HAZİRAN - 2012

FİYAT KARŞILAŞTIRMALI ÜRÜN ARAMA MOTORU GELİŞTİRME

FURKAN GÖZÜKARA

MERSİN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

Danışman
Yrd. Doç. Dr. Zeki YETGİN

MERSİN
HAZİRAN – 2012

Furkan GÖZÜKARA tarafından Yrd. Doç. Dr. Zeki YETGİN danışmanlığında hazırlanan "Fiyat Karşılaştırmalı Ürün Arama Motoru Geliştirme" başlıklı bu çalışma aşağıda imzaları bulunan jüri üyeleri tarafından oy birliği ile Yüksek Lisans Tezi olarak kabul edilmiştir.

İmza

Prof. Dr. Caner ÖZDEMİR .....

Yrd. Doç. Dr. Zeki YETGİN .....

Yrd. Doç. Dr. Evren DEĞİRMENCİ .....

Yukarıdaki Jüri kararı Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 23/07/2012 tarih ve 2012/15/437 sayılı kararıyla onaylanmıştır.

Prof. Dr. A. Murat GİZİR
Enstitü Müdürü

FİYAT KARŞILAŞTIRMALI ÜRÜN ARAMA MOTORU GELİŞTİRME

Furkan GÖZÜKARA

ÖZ

Son yıllarda, dünyada ve ülkemizde hızla yaygınlaşan internet kullanımı, beraberinde pek çok e-ticaret iş olanaklarını ve çevrimiçi web marketleri geliştirmiştir. Bu iş olanaklarından belki de en büyüğü internet siteleri üzerinden etkileşimli ürün satışıdır. İnternet siteleri üzerinden etkileşimli ürün satışı yapan firma sayısı her geçen gün artmaktadır. Firmalar arası rekabetten dolayı, aynı ürünün satış fiyatları, firmalar arasında büyük farklılıklar göstermektedir. İnternet üzerinden ürün satın alacak kişilerin, bütün web marketleri ve diğer e-ticaret sitelerini tarayarak, alacağı ürünün en ucuzunu bulması hem zor hem de çok zaman alıcı bir iştir. Kişilerin bütün ürün satan siteleri taraması yerine, ürün satışı yapan siteler program aracılığı ile taranabilir ve aynı ürünler gruplanarak, müşterilere fiyatlarıyla beraber listelenmesi sağlanabilir. Fakat farklı web kaynaklarından toplanan aynı ürünlerin gruplanması zor bir problemdir. Ürünlere ait bilgiler doğası gereği hatalı, eksik, fazla ya da çelişkili veriler içerebilmektedir. Örneğin, müşterileri etkilemek için ürünün tanımlanmasında fazladan kelimeler kullanılabilmekte, ya da insan hatasıyla kelimeler yanlış ya da eksik yazılabilmektedir. Gürültü olarak isimlendirdiğimiz bu durum aynı ürünlerin kümelenmesinde kesinlikle çözülmesi gereken sorunlardan birisidir. Literatürde web tarama yoluyla toplanan ürün bilgilerinin işlenmesine yönelik çalışmalar çok azdır. Aynı ürünlerin gözetimsiz olarak kümelenmesi ve bu kümeleme işlemenin gerektirdiği ön işlemlerden olan özellik vektörlerinin çıkarılması ve gürültülere karşı normalleştirme henüz çalışılmamış konular arasındadır. Literatürde ürünlerin otomatik olarak kümelenmesine yönelik çalışmalar mevcut olmakla beraber, bu çalışmalar geniş ölçekli kataloglama olarak karşımıza çıkmakta ve bir veya daha fazla açıdan benzer olan ürünlerin kümelenmesini amaçlamaktadır. Bu tez çalışmasında, e-ticaret ve çevrimiçi web market siteleri geliştirilen tarama ajanı ile taranarak farklı kaynaklardan birçok ürün bilgileri toplanmıştır. Bu ürün bilgileri geliştirilen veri tabanı sisteminde saklanmış ve daha sonra özel olarak geliştirilen normalleştirme ve özellik çıkarma yöntemiyle, gürültülerin elenmesi ve özellik vektörlerinin çıkarılması sağlanmıştır. Özellikleri çıkarılan bu ürünler geliştirilen kümeleme algoritması ve standart kümeleme algoritmaları üzerinde test edilerek performans analizleri gerçekleştirilmiştir. Ayrıca, bu tez kapsamında aynı ürünlerin gruplanmasında hata oranını hesaplayan yeni performans ölçütleri de önerilmiştir. Son olarak gruplanan ticari ürünler, geliştirilen web ara yüzü ile kullanıcıların ihtiyacını karşılayacak hale getirilmiş ve interaktif kullanıma açılmıştır.

Anahtar Kelimeler: Fiyat Arama Motoru, Ticari Ürünlerin Gruplanması, Ürün Sınıflandırma, Ürün Kümeleme

Danışman: Yrd. Doç. Dr. Zeki YETGİN, Mersin Üniversitesi, Bilgisayar Mühendisliği Ana Bilim Dalı

DEVELOPING PRODUCT PRICE COMPARISON SEARCH ENGINE

Furkan GÖZÜKARA

ABSTRACT

Recently, the rapidly growing usage of internet both in the world and in our country has brought many e-commerce and online web markets opportunities together. Probably the biggest job opportunity among these opportunities is online product selling on websites. The number of companies which sells online products is increasing more and more. Due to the competition between companies, the same product prices could be a lot different between companies. For the persons who are going to buy an online product, scanning all of the web markets and other e-commerce sites in order to find the cheapest product they are looking for is very hard and time consuming process. Instead of searching individually, all of the online product selling web sites can be crawled by software and same products can be grouped to display to the users with their prices. But grouping same products which are collected from different web sources is a difficult problem. The products information may have faulty, missing, redundant or inconsistent data by their nature. For example, in order to impress customers there may be extra words when defining product features or missing or incorrectly written words by human errors. This situation that we call as noise is certainly a problem that has to be solved when clustering same products. There are few works about processing products information obtained via web crawling in literature. Unsupervised clustering of the same products, and its priori steps, feature vectors extraction and normalization against noises are topics that are not studied till now. There are works about automatic products clustering in literature but these works are wide scaled categorization and aim to cluster products having one or more similar aspects. In this thesis, e-commerce and online web markets are crawled via our crawling agent and many products information are collected from many different sources. These products' information is saved in a designed database and then the noises are eliminated and feature vectors are generated using the proposed normalization and feature vectors extracting methods. The products having extracted features are tested via newly developed clustering algorithm and standard clustering algorithms, and performance analysis is presented. In addition, in this thesis new error metrics are proposed that calculates the error rate in clustering of the same products. Finally, clustered commercial products are opened to public use via the developed web interface that will satisfy users' demands.

Keywords: Price Search Engine, Clustering of Commercial Products, Product Classification, Product Clustering

Advisor: Asst. Prof. Dr. Zeki YETGİN, Mersin University, Computer Engineering Department

TEŞEKKÜR

Tüm eğitim hayatım boyunca desteklerini eksik etmeyen aileme ve yüksek lisans eğitimim süresince, çalışmalarımın her aşamasında bilgi ve tecrübelerini benimle paylaşan, yardımını esirgemeyen danışmanım Sayın Yrd. Doç. Dr. Zeki YETGİN'e teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZ ..... **i**
ABSTRACT ..... **ii**
TEŞEKKÜR ..... **iii**
İÇİNDEKİLER ..... **iv**
ÇİZELGELER DİZİNİ ..... **vii**
ŞEKİLLER DİZİNİ ..... **viii**
SİMGE VE KISALTMALAR DİZİNİ ..... **ix**

1. GİRİŞ **1**

2. KAYNAK ARAŞTIRMASI **4**

2.1. WEB TABANLI ARAMA MOTORLARI 8

2.1.1. Genel Amaçlı Web Tabanlı Arama Motoru Nedir 8
2.1.2. Özelleşmiş Web Tabanlı Arama Motoru Nedir 8

2.2. WEB SİTELERİNİ TARAYAN BOT YAZILIMLAR 9

2.2.1. Web Sitelerini Tarayan Bot Yazılım Nedir 9
2.2.2. Web Sitelerini Tarayan Bot Yazılımların Çalışma Mantığı 9

2.3. VERİ MADENCİLİĞİ 10

2.4. METİN İŞLEME 10

2.4.1. Normalleştirme Ve Gürültü Eleme 10
2.4.2. Parametrelerin Çıkarılması 11
2.4.3. Özellik Vektörlerinin Oluşturulması 11
2.4.4. Özellik Vektörlerinin Seçilmesi (Boyut İndirgeme) 11
2.4.5. Gözetimsiz Öğrenme Yöntemi İle Metin Tabanlı Kümeleme Algoritmaları . 12
2.4.5.1. K-means algoritması 12
2.4.5.2. Hiyerarşik kümeleme algoritması 13
2.4.6. Test Süreci 14

2.5. KULLANICI ARAYÜZÜ 15

2.6. E-TİCARET ÜRÜN KÜMELEME SİSTEMLERİ 15

3. MATERYAL VE YÖNTEM **17**

3.1. MICROSOFT VISUAL STUDIO 2010 17

3.2. MİCROSOFT SQL SERVER 2008 R2 17

3.3. MATLAB R2011A 17

3.4. WEB SİTELERİNİ TARAYACAK BOT YAZILIM 18

3.4.1. Sitelerin Taranması 18
3.4.2. Taranan Sitelerden Gerekli Bilgilerin Çıkartılması – Veri Toplama 22
3.4.2.1. Ürün sayfalarından çıkartılan bilgiler 22
3.4.2.2. Çalışma prensibi 23

3.5. NORMALLEŞTİRME VE GÜRÜLTÜ ELEME 24

3.5.1. Harf Bazında Normalleştirme Ve Gürültü Eleme 24
3.5.2. Kelime Bazında Normalleştirme Ve Gürültü Eleme 25

3.6. PARAMETRELERİN BELİRLENMESİ, ÖZELLİK VEKTÖRLERİNİN OLUŞTURULMASI VE BOYUT İNDİRGEME 25

3.6.1. Parametrelerin Belirlenmesi 25
3.6.2. Özellik Vektörlerinin Oluşturulması 26
3.6.2.1. İlk gruplama ile daha odaklı veri işleme 26
3.6.2.2. Grupların gözetimsiz olarak eğitilerek özellik vektörlerin oluşturulması 29
3.6.3. Ham Verilerin İşlenme Süreci Örnek 33

3.7. KÜMELEME ALGORİTMASI İLE ÜRÜNLERİN GRUPLANMASI 38

3.7.1. Aşama 1 Dereceli Azalma Tabanlı Kümeleme 38
3.7.2. Aşama 2 Dereceli Birleştirme Tabanlı Kümeleme 40

3.8. TEST SÜRECİ 42

3.8.1. Kümeleme Algoritmasının Başarısını Ölçen Algoritma 42
3.8.1.1. Kaçırma tespiti 43
3.8.1.2. Yanlış alarm 45
3.8.1.3. Toplam hata 46

3.9. WEB ARAYÜZÜ 46

4. BULGULAR VE TARTIŞMA 48

4.1. HİYERARŞİK KÜMELEME ALGORİTMALARININ TEST EDİLMESİ VE SONUÇLARI 49

4.2. TEZ KAPSAMINDA GELİŞTİRİLEN ÖZEL KÜMELEME ALGORİTMASININ TEST EDİLMESİ VE SONUÇLARI 54

5. SONUÇLAR VE ÖNERİLER 56

KAYNAKLAR 57

ÖZGEÇMİŞ VE ESERLER LİSTESİ 64

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 2.1. Hiyerarşik kümeleme algoritması veriler arasındaki benzeşmezliği hesaplayacak yaygın ölçütler [40-41] 14
Çizelge 2.2. Hiyerarşik kümeleme algoritması veri setleri arasındaki benzeşmezlik uzaklığını hesaplayan yaygın bağ ölçütleri [40,42-43] 14
Çizelge 3.1. Taranan e-ticaret siteleri ve istatistikleri 22
Çizelge 3.2. Oluşan eğitim ürün gruplarına örnek 29
Çizelge 3.3. Oluşturulan eğitim grupları içindeki kelime frekanslarına örnek 30
Çizelge 3.4. Eğitim gruplarının ilk aşama eğitilmesi sonrası matris 30
Çizelge 3.5. Eğitim gruplarının ikinci aşama eğitilmesi sonrası matris 31
Çizelge 3.6. Manual sınıflandırılmış Canon renkli kartuş ürün kümesinin işlenmemiş hali 33
Çizelge 3.7. Manual sınıflandırılmış Samsung netbook ürün kümesinin işlenmemiş hali 34
Çizelge 3.8. Manual sınıflandırılmış Canon renkli kartuş ürün kümesinin harf bazında gürültü elemesi yapıldıktan sonraki hali 35
Çizelge 3.9. Manual sınıflandırılmış Samsung netbook ürün kümesinin harf bazında gürültü elemesi yapıldıktan sonraki hali 36
Çizelge 3.10. Manual sınıflandırılmış Canon renkli kartuş ürün kümesinin gürültü elemesi yapan ve özellik vektörlerini çıkartan algoritma ile 60 eşik değeri ve 10 döngü sayısı kullanılarak özellik vektörleri oluşturulduktan sonraki hali 37
Çizelge 3.11. Manual sınıflandırılmış Samsung netbook ürün kümesinin gürültü elemesi yapan ve özellik vektörlerini çıkartan algoritma ile 60 eşik değeri ve 10 döngü sayısı kullanılarak özellik vektörleri oluşturulduktan sonraki hali 37
Çizelge 3.12. Kaçırma tespiti örnek 44
Çizelge 3.13. Yanlış alarm örnek 45
Çizelge 4.1. Hiyerarşik kümeleme en yakın mesafe algoritması performans sonuçları 50
Çizelge 4.2. Hiyerarşik kümeleme en uzak mesafe algoritması performans sonuçları 51
Çizelge 4.3. Hiyerarşik kümeleme ağırlıksız ortalama mesafe algoritması performans sonuçları 51
Çizelge 4.4. Hiyerarşik kümeleme ağırlıklı ortalama mesafe algoritması performans sonuçları 52
Çizelge 4.5. Hiyerarşik kümeleme ağırlıksız kitlesel mesafenin merkezi algoritması performans sonuçları 52
Çizelge 4.6. Hiyerarşik kümeleme ağırlıklı kitlesel mesafenin merkezi algoritması performans sonuçları 53
Çizelge 4.7. Hiyerarşik kümeleme içsel kare mesafe algoritması performans sonuçları 53
Çizelge 4.8. Tez kapsamında geliştirilen özel kümeleme algoritmasının performans sonuçları 55

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. Tam bir KAM tasarımı için gereken aşamalar 7
Şekil 2.2. Tam bir KAM tasarımında tekrar eden aşamalar 8
Şekil 2.3. WTA'ların gerçekleştirdiği genel işlemler 10
Şekil 2.4. K-means algoritması adımlar [39] 13
Şekil 3.1. Site tarama ajanı ilk aşama akış şeması 19
Şekil 3.2. Site tarama ajanı 2. aşama akış şeması 21
Şekil 3.3. Taranan bir sayfadan ürün bilgilerinin çıkartılması 24
Şekil 3.4. Tüm ürünlerin diğer tüm ürünler ile gruplanması akış şeması 28
Şekil 3.5. Özellik vektörlerini oluşturan algoritmanın akış şeması 32
Şekil 3.6. Aşama 1 dereceli azalma tabanlı kümeleme akış şeması 40
Şekil 3.7. Aşama 2 dereceli birleştirme tabanlı kümeleme akış şeması 41
Şekil 3.8. Web ara yüzü örnek ürün arama sorgusu ve dönen sonuçlar 46
Şekil 3.9. Web ara yüzü ürünlerin satıldığı siteler hakkında detaylı bilgi sayfası 47

SİMGE VE KISALTMALAR DİZİNİ

KAM : Karşılaştırmalı Arama Motoru

WTA : Web Tarama Ajanı

URL : Aynı Tarzda Kaynak Konum Bulucu

MVS : Microsoft Visual Studio

MSS : Microsoft SQL Server

WPF : Windows Presentation Foundation

HAP : Html Agility Pack

T-SQL : Transact-SQL

SQL : Yapılandırılmış Sorgu Dili

ID : Kimlik Numarası

TOD : Tanımlayıcı Olasılık Değeri

1. GİRİŞ

Son yıllarda, dünyada ve aynı zamanda Türkiye'de de yaygınlaşan internet kullanımı, beraberinde pek çok ticari iş olanaklarını getirmiştir. Bu iş olanaklarından belki de en büyüğü internet siteleri üzerinden interaktif ürün satışıdır. Her geçen yıl katlanarak artan internet üzerinden yapılan alışverişler, artık günlük hayatın bir parçası haline gelmiştir. İnternet üzerinden yapılan alışverişler e-ticaret olarak adlandırılmaktadır. Türkiye'nin e-ticaret hacmi 2003 yılında 262 milyon TL iken 2011 yılı itibari ile 22,9 milyar TL'ye yükselmiştir [1]. 2010 yılında ise e-ticaret hacmi 15,2 milyar TL olarak gerçekleşmiştir [1]. Yani 1 senede gerçekleşen artış %50 seviyesinde olmuştur. Bu veriler Türkiye'de e-ticaretin ne denli hızlı bir şekilde geliştiğinin kanıtı niteliğindedir.

Gelişen e-ticaret sistemleri ile istenilen ürünler artık evlerden kısa sürede sipariş edebilmekte ve mağazalardan satın almaya göre çok daha geniş bir ürün yelpazesine sahip olunabilmektedir. Bu şekilde internet üzerinden yapılan alışverişler, herkese sadece tek bir tıklama mesafesinde olduğu için bu alandaki rekabet de mağaza satışlarına göre çok daha fazla olmaktadır. İnsanlar alacakları bir ürüne rahatça ulaşabilmenin yanında, fiyat karşılaştırması da yaparak en ucuz ürünü almaya yönelmektedir.

E-ticaret alanındaki gelişmeler bahsedildiği üzere birçok avantajlar getirdiği gibi, paralelde bu avantajlarla beraber bazı problemlerde doğurmaktadır. Bir ürün satın almak için her geçen gün sayısı artan web sitelerinin tamamını taramak saatler, hatta belki de günler alacaktır. Ayrıca bütün web sitelerinin adreslerini ürün satın almak isteyen kişilerin bilmesi de neredeyse olanaksızdır. Bu yüzden bir kişinin ürün talebiyle, bütün web sitelerini tarayarak aralarında o ürünü en ucuza satan web sitesini bulması oldukça zordur. Gelişen teknoloji ile ürün fiyatları da sürekli güncellenmekte ve pek çok web sitesi sıklıkla kampanyalar düzenlemektedir. Bir ürünü almaya karar veren kişi, 2 gün sonra o ürünün düşen fiyatını bulmak için bütün web sitelerini yeniden taramak zorunda kalacaktır ve bu da çok ciddi zaman kaybına neden olacaktır.

Tüm bu sebeplerden dolayı bu tezde amaçlanan, internet üzerinden ürün satışı yapan web sitelerinin bütün ürünlerini otomatik olarak tarayacak, bulduğu bütün ürünlerin aynı olanlarını gözetimsiz öğrenme metotları ile kümeleyecek,

herhangi bir ürünü satın almak isteyen kişiye aradığı ürünün satıldığı yerleri, fiyatları ile beraber listeleyecek, özelleşmiş bir arama motoru geliştirmektir. Bu şekilde listelemeyi yapan aracı sistemlere karşılaştırmalı alışveriş motoru (KAM) ismi verilmektedir [2]. Bu özelleşmiş arama motoru ile insanlar satın almak istedikleri ürünlerin, nerelerde, kaça satıldığına, anında en güncel şekilde ulaşabilecek ve bu sayede oldukça yorucu ve zaman kaybına sebep olan satın almak istedikleri ürünü arama işinden kurtulacaklardır.

Fiyat karşılaştırmalı arama motoru yapabilmek için gerekli olan çeşitli unsurlar vardır. İlk olarak ticari web sitelerinin tamamını tarayacak ve ürün bilgilerini taradığı sayfaların işlenmemiş içeriğinden çıkartabilecek bot yazılımı geliştirilmesi gerekmektedir. Daha sonra bilgilerin doğru şekilde saklanmasını sağlayacak veri tabanı sistemi tasarlanmalı ve tarayıcı bot yazılımı ile bütünleştirilmelidir. Bilgiler elde edildikten sonra bu bilgileri işlemden geçirerek gürültüden ayıklayacak ve kümeleme algoritmaları için uygun şekilde özellik vektörleri oluşturacak algoritma geliştirilmesi gerekmektedir. Özellik vektörleri elde edildikten sonra bu bilgileri kullanarak doğru bir şekilde farklı sitelerde satılan aynı ürünleri sınıflandıracak kümeleme algoritması tasarlanmalıdır. Ürünler kümelendikten sonra son yapılması gereken ise kullanıcılar arama yaptıklarında aradıkları ürünü, ürünün satıldığı siteler ve fiyatları ile beraber kullanıcıya döndürecek bir web ara yüzü tasarımının yapılmasıdır.

Özelleşmiş fiyat arama motoru tasarımındaki en önemli unsur elde edilen ürünlerin gruplanmasıdır. E-ticaret siteleri ürünlerini daha cazip hale getirmek için listelerken genellikle ürün tanımlamalarına gereksiz kelimeler eklemektedirler. Ayrıca ürün tanımlamalarında hatalı veya eksik bilgilerin olması sıklıkla rastlanan bir durumdur. Ek olarak birden çok web sitesinden ürün bilgilerinin çekilmesi, doğal olarak ürün tanımlamalarında farklılığa sebep olmaktadır. Bir kullanıcı bir ürünü aradığında, KAM'nin bu ürünü doğru şekilde tespit etmesi ve sadece bu ürünün satıldığı tüm ticari web sitelerini listelemesi gerekmektedir. Sadece aranan ürünün satıldığı tüm web sitelerini listelemek ise, bütün ürünlerin doğru şekilde gruplandırılmasına bağlıdır. Gruplamanın doğru şekilde yapılabilmesi için ise bahsedilen sebeplerden dolayı oldukça gürültülü olarak elde edilen ürün bilgilerinin işlenerek gürültüden arındırılması ve kümeleme algoritmalarına uygun hale

getirilmesi gerekmektedir. Bu tez kapsamında önerilen en önemli yeniliklerden bir tanesi, bu gürültülü ürün bilgilerini işleyerek gürültüden arındıracak ve kümeleme algoritmaları için uygun özellik vektörlerini oluşturacak algoritmadır. Yüzlerce e-ticaret sitesinde milyonlarca ürün satıldığı göz önüne alınırsa bu algoritmanın ölçeklenebilir olması şarttır. Bu yüzden geliştirilen algoritma tamamıyla gözetimsiz öğrenme metotları ile çalışmaktadır. Geliştirilen gürültü eleme ve özellik vektörlerini çıkartma algoritması bilinen standart kümeleme sistemleri ve yeni geliştirilen özel kümeleme sistemi ile test edilerek oldukça başarılı sonuçlar elde edilmiştir.

Tez kapsamında önerilen diğer bir önemli yenilik ise bu tarz problemlerin hata miktarını hesaplayabilecek hata hesaplama algoritmasıdır. Literatürde bu alanda çok sınırlı sayıda çalışma olduğu için yaygın bir hesaplama yöntemi bulunamamıştır. Önerilen hata hesaplama yöntemi kaçırma tespiti ve yanlış alarm metotlarını kullanarak oldukça başarılı şekilde aynı ürünlerin gruplanması sonucu oluşan kümelerin hata miktarını hesaplayabilmektedir.

Sonuç olarak web tarama yoluyla ürünlerin gruplanması alanında literatür çalışması oldukça az olup, bu tez kapsamında yapıldığı şekliyle bir çalışma bulunamamıştır. Tez kapsamında geliştirdiğimiz normalleştirme ve özellik çıkarma yöntemleri özgün değerlerdendir. Yine, aynı ürünlerin gruplanmasında hata tespitini ölçen başarı metrikleri özgün değerlerdendir. Bir bütün olarak, web tarama botu ile web marketlerin taranması, özellik vektörlerinin çıkarılıp, kullanıcıların arayabileceği bir arama motoruna dönüştürülmesi de özgün bir çalışmadır. Literatüre kazandırılan yeni değerler gelecekte bu alanda yapılacak çalışmalara ışık tutacak ve insanlığın faydasına olacak yeni algoritma ve yazılımların geliştirilmesine katkı sağlayacaktır.

2. KAYNAK ARAŞTIRMASI

KAM tasarımının yapıldığı 2 adet literatür çalışması bulunmuştur. İlk çalışma 1997 yılında Robert B. ve arkadaşları tarafından [3], ikinci çalışma ise Jaeyoung Y. ve arkadaşları tarafından 2000 yılında gerçekleştirilmiştir [2]. Bu 2 çalışmaya ek olarak, Liyi Zhang ve arkadaşları ve Ig-hoon Lee ve arkadaşları tarafından gerçekleştirilen, tam bir KAM tasarımı değil fakat gene ürünlerin özelliklerini kullanarak aranan özelliklere göre aynı veya benzer ürünleri listelemek için isim tabanlı çalışan ürün bilgi edinme sistem tasarımı çalışmaları mevcuttur [4-5].

Robert B. ve arkadaşları tarafından yapılan çalışmada web sitesi bağımlılığı olmayan bir KAM geliştirmiş ve bunu ShopBot olarak isimlendirmişlerdir. Bu çalışmada geliştirilen KAM sistemi, çevrimdışı olarak ve gözetimsiz öğrenme metotlarını kullanarak ticari ürün satışı yapan web sitelerinin özelliklerini çıkartabilecek şekilde tasarlanmıştır. Ayrıca geliştirilen sistem ticari web sitelerinin arama sistemlerini de otomatik olarak tespit edebilmekte ve kullanıcıya sıralı ürün listelemesi yapmak için ticari web sitelerinin kendi arama sistemlerinden faydalanmaktadır. Sistem çalışma mantığı olarak kullanıcıdan arama girdisini aldıktan sonra daha önce çevrimdışı olarak öğrendiği arama yapma sistemini kullanarak, sisteminde kayıtlı olan ticari ürün sitelerinin arama sistemlerine, kullanıcıdan aldığı girdiyi yolluyor. Daha sonra ticari sitelerden dönen arama sonuçlarını, daha önceden çevrimdışı olarak öğrendiği bilgi ayıklama sistemine vererek sonuçları kullanıcıya gösteriyor. Sistem çok ciddi sorunları da içerisinde barındırmaktadır. Çevrimdışı ve gözetimsiz olarak, web sitelerinin arama sistemlerini öğrenebilmek ve bilgi çıkartma sistemini geliştirebilmek için çok ciddi web sitesi bağımlı varsayımlar yapmaktadır. Yayının yapıldığı 1997 yılında henüz internet tarayıcıları çok gelişmiş olmadıkları için, web siteleri daha düzenli tasarım yapmak zorundaydı. Bu yüzden bu varsayımlar o yılda kısmen başarılı olmuş olabilir. Fakat günümüzde web tarayıcıları çok geliştiği ve hataları çok iyi derecede düzeltebildikleri için, tasarımlar oldukça düzensiz hatta yanlış hale gelmiştir. Bu sebeple yapılan varsayımlar, çalışmanın yapıldığı yıla göre çok daha az başarılı olacaktır. Diğer bir sorun ise, ticari web sitelerinin sahip oldukları dâhili arama sistemlerine güvenmektir. Ticari web sitelerinin arama sistemlerin oldukça zayıf

olabilmekte, hatta hiçbir arama sistemine sahip olmayabilmektedirler. Bu yüzden yapılacak aramalar çok zayıf şekilde doğru sonuçları döndürecektir. Son olarak ise sistem, kullanıcı arama girdisini girdikten sonra, ticari web sitelerine bağlanmakta ve kullanıcıdan aldıkları sorguyu web sitelerinin arama sistemlerine vermektedir. Bu durumda ciddi ağ gecikmeleri yaşanacaktır. Ayrıca ticari web siteleri gereksiz miktarda yük altına gireceği için, sistemin, ağlarına bağlanmasını da engelleyebilirler. Bu durumda hiç sonuç dönmeyebilir, çok zayıf sonuç dönebilir ve çok uzun süre sonuç bekleme durumları gerçekleşebilir. Bu tez kapsamında önerilen sistemde ise, web siteleri tamamıyla özel olarak geliştirilen tarama programı ile taranmakta, veri tabanında saklanmakta ve ürünler tamamıyla gözetimsiz öğrenme algoritmaları gruplanmaktadır. Bu sayede kullanıcı girdisi alındıktan sonra, Robert B. ve arkadaşları tarafından önerilen sisteme göre çok daha hızlı ve çok daha doğru sonuçlar kullanıcıya döndürülebilmektedir.

Jaeyoung Y. ve arkadaşları tarafından yapılan çalışma, Robert B. ve arkadaşları tarafından yapılan çalışma ile aynı mantığa sahiptir. Bu çalışmanın farkı ise, web sitelerinden bilgi çıkarma sisteminin varsayım miktarının daha aza indirgenmiş olması, böylece daha yüksek başarı sağlanmasıdır. Bu sistemde de gene çevrimdışı olarak ticari web sitelerini analiz etmekte ve web sitelerinin arama sistemlerini tespit edip, dönen sonuçları nasıl doğru şekilde yorumlayacağını öğrenmektedir. Çevrimdışı olarak arama yapılacak her web sitesi için gerekli arama sistemi tespiti yapıldıktan ve dönen sonuçları doğru şekilde yorumlayacak bilgi çıkarma sistemi geliştirildikten sonra, kullanıcıdan alınan sorgular ticari web sitelerinin dahili arama sistemlerine gönderilmekte ve çıkan sonuçlar yorumlanarak kullanıcılara gösterilmekte. Bu sisteminde çok ciddi problemleri bulunmaktadır. Öncelikle her ne kadar varsayım sayısı azaltılmış olsa da gene çeşitli varsayımlar mevcuttur. Mesela arama sisteminin html form değil de javascript ile çalışması durumunda sistem web sitesinin arama sistemini tespit edememektedir. Web sitelerinin arama sistemlerinin oldukça az gelişmiş olması, arama sisteminin hiç olmaması ve ağ bağlantısı gecikmesi gibi sorunlar bu sistemde de mevcuttur.

Ayrıca tam bir KAM tasarımı olmayan fakat gene e-ticaret sitelerinde satılan ürün bilgilerini kullanarak kullanıcılara ürün listeleri döndüren ve bu tez kapsamında yapılandan farklı olarak isim tabanlı çalışan sistem çalışmaları

bulunmaktadır [4-6]. Bu sistemlerin tez kapsamında yapılan çalışma ile alakalı olmasının sebebi, bu sistemlerde de e-ticaret sitelerinde satılan ürünlerin özellikleri çıkartılmış, geliştirilen isim tabanlı e-ticaret bilgi edinme sistemleri ile ürünlere belirli özellikler atanmış ve daha sonra kullanıcıdan alınan ürün özellikleri girdisi ile bu özellikleri sağlayan ürünlerin listesi döndürülmüştür. Bu sistemlerin tez kapsamında geliştirilen sistemden farklılığı ise, farklı e-ticaret sitelerinde satılan aynı ürünleri fiyatları ile kullanıcıya listeleyerek en ucuzunu satan siteyi kullanıcıya göstermek değil, kullanıcının aradığı özelliklere sahip bütün ürünleri kullanıcıya döndürmektedir. Ayrıca geliştirilen bu sistemlerde tez kapsamında gerçekleştirildiği şekliyle herhangi bir kümeleme veya sınıflandırma mevcut değildir. Tez kapsamında tasarlanan KAM sisteminde ise, farklı web sitelerinde satılan aynı ürünler kümelenmektedir.

Bu çalışmaların dışında direk olarak tam bir KAM sisteminin geliştirildiği literatür çalışması bulunamamıştır. Fakat benzer olarak e-ticaret sistemlerinde, bu tez kapsamında yapıldığı gibi ürünlerin bilgilerinin kullanılarak ürün kataloglanması için çalışmalar mevcuttur [7-9]. Bu çalışmalar genel olarak ticari yazılımları anlatan çalışmalardır ve kullanılan algoritmalar açık olarak belirtilmemiştir.

KAM aslında genel amaçlı web tabanlı arama motorlarının özelleşmiş bir biçimidir ve tam bir KAM tasarımının yapılabilmesi için gereken unsurlar Şekil 2.1'de sırası ile verilmiştir. Şekil 2.2'de ise tasarım sürecindeki tekrar eden aşamalar gösterilmiştir. Bu tez kapsamında tüm bu unsurlar geçekleştirilmiş ve tam bir KAM tasarımı yapılmıştır.

Şekil 2.1. Tam bir KAM tasarımı için gereken aşamalar

Şekil 2.2. Tam bir KAM tasarımında tekrar eden aşamalar

2.1. WEB TABANLI ARAMA MOTORLARI

2.1.1. Genel Amaçlı Web Tabanlı Arama Motoru Nedir

Web tabanlı arama motorları keşfedebildikleri bütün web dokümanlarını tarayarak veri tabanlarında saklayan ve kullanıcı girdisi aldıklarında kendi içsel sıralama ve kümeleme algoritmaları ile kullanıcı için en alakalı olduğunu düşündükleri sonuçları kullanıcıya dönen sistemlerdir [10-12]. Web sitelerinin ziyaretçi istatistiklerini tutan web tabanlı bilgi firması Alexa'ya göre şu an için en popüler arama motoru Google olup hakkında 10.000'lerce literatür çalışması yapılmıştır [12-13]. Google arama motorunu sırası ile Yahoo, Baidu ve Bing takip etmektedir [14-16].

Genel amaçlı arama motorları herhangi konu veya doküman kısıtlaması yapmadan bütün web'i tararlar ve arama yapan kullanıcıya elde ettikleri bütün dokümanlar arasından sonuç döndürürler.

2.1.2. Özelleşmiş Web Tabanlı Arama Motoru Nedir

Özelleşmiş web tabanlı arama motorları, web üzerindeki sadece bir konu veya kategori üzerine yoğunlaşırlar [17]. Tüm web yerine belirli bir konu veya kategori üzerine yoğunlaşarak, kullanıcıya, o konu veya kategori hakkında tüm web'i

tarayan arama motorlarına göre çok daha iyi sonuçlar döndürebilirler. Ayrıca tüm web'i tarayan arama motorlarına göre çok daha kolay yönetilebilir veri tabanları olacak ve çok daha az donanımsal güce ihtiyaç duyacaklardır. Bu tez kapsamında tasarlanan KAM'de özelleşmiş bir web tabanlı arama motorudur. Tasarlanan KAM sadece belirlenen web sitelerini taramakta ve amacı kullanıcılara, kullanıcıların aradıkları ürünün satıldığı tüm ticari siteleri fiyatları ile beraber listelemektir.

2.2. WEB SİTELERİNİ TARAYAN BOT YAZILIMLAR

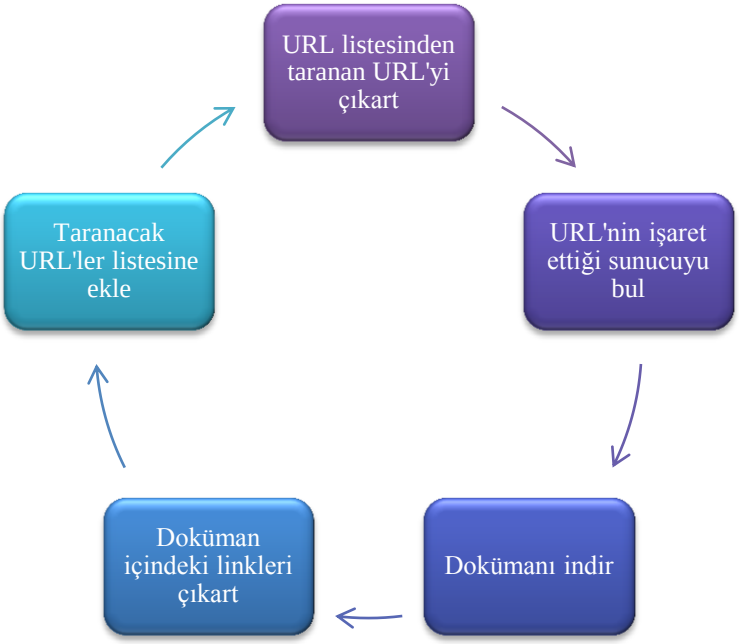
2.2.1. Web Sitelerini Tarayan Bot Yazılım Nedir

Web siteleri hakkında bilgi toplayabilmek için web sitelerini, barındırıldıkları bilgisayarlardan (web sunucuları) istekte bulunarak yerel hafızasına indirecek ve indirdiği ham bilgilerden istediği bilgileri çıkartacak yazılımlara ihtiyaç vardır. Bu tip yazılımlara genel olarak web tarama ajanı (WTA) denilmektedir. Bir WTA, web örümceği veya web robotu, yinelemeli ve otomatik olarak web sayfalarını indirebilen, indirdiği web dokümanlarının HTML kısmından istediği bilgileri ve linkleri çıkartabilen, tek bir program veya programlar grubudur [18].

Tez kapsamında yapılan çalışma direk olarak ürün satışı yapan web siteleri ile alakalı olduğu için bu siteleri tarayacak, yerel hafızaya indirecek ve indirdiği ham bilgilerden gerekli olan bilgileri çıkartacak bir WTA tasarımı yapılmıştır.

2.2.2. Web Sitelerini Tarayan Bot Yazılımların Çalışma Mantığı

Her WTA tarafından gerçekleştirilen temel adımlar, girdi olarak tohum kaynak konum bulucu (URL) listesi almak, listeden sıradaki URL'yi seçmek, seçilen URL'lerin daha önce taranmadığından emin olmak ve Şekil 2.3'deki adımları tekrar tekrar gerçekleştirmektir [19]. Tez kapsamında tasarlanan özel WTA, WTA'ların genel çalışma mantığından çok farklı değildir. Genel çalışma mantığına ek olarak, ticari web sitelerinden ürünlerin sınıflandırılması için gerekli olan bilgileri çıkartabilecek özel fonksiyonlar eklenmiştir.



Şekil 2.3. WTA'ların gerçekleştirdiği genel işlemler

2.3. VERİ MADENCİLİĞİ

Veri madenciliği, içerisinden anlamlı kuralları ve kalıpları çıkartmak için, çok büyük miktarda verinin keşfedilmesi ve işlenmesidir [20]. Tam bir KAM tasarımı yapılırken veri madenciliğinin önemi oldukça büyüktür. Taranan ürün satışı yapan web sitelerinden, ürün gruplamak için işe yarayacak yararlı bilgiler çıkartılmalıdır. Bu işlem tam olarak veri madenciliği alanına girmektedir. Keşfedilen verilerin işlenmesi için gerekli olan algoritmalar, tez kapsamında oluşturulan WTA ile bütünleştirilmiştir. Tasarlanan WTA sadece web sitelerini tarama işlemi yapmamakta, aynı zamanda veri madenciliği de yaparak taradığı dokümanları işlemekte ve gerekli bilgileri çıkartarak kaydetmektedir.

2.4. METİN İŞLEME

2.4.1. Normalleştirme Ve Gürültü Eleme

Normalleştirme ve gürültü eleme, pek çok metin işleme ve bilgi toplama uygulamasının çok önemli bir unsurudur [21]. Bu tez kapsamında da pek çok normalleştirme ve gürültü eleme yapılmıştır. Normalleştirmeye bütün kelimelerin küçük harfe dönüştürülmesi örnek olarak verilebilirken, gürültü elemeye örnek

olarak, web sitelerinden çekilen metinlerdeki gereksiz karakterlerin atılmasını verebiliriz.

2.4.2. Parametrelerin Çıkarılması

Parametre tahmininin zor kısmı, çıkışı doğru şekilde elde edebilen ve modeli çok karmaşık hale getirmeyen, genel olarak gürültülü ve fazlalık olan özelliklerin doğru birleşimini bulmaktır [21]. Metin işleme sistemlerinde doğru parametrelerin önemi çok büyüktür. Tez kapsamında farklı web sitelerinde satılan aynı ürünlerin gruplanması için geliştirilen algoritmaların en yüksek doğruluk ile çalışması için pek çok farklı parametre denenmiştir. Ayrıca daha iyi sonuçlar elde edebilmek çok çeşitli parametre iyileştirilmesi yapılmıştır.

2.4.3. Özellik Vektörlerinin Oluşturulması

Özellik vektörleri çıkartılarak, tahmin edicilerin performansının artması, daha hızlı ve maliyet etkin tahmin ediciler elde etmek ve veriyi oluşturan alt kademe işlemleri daha iyi anlamak hedeflenmektedir [22]. Tez kapsamında geliştirilen kümeleme algoritması da, genel metin tabanlı kümeleme algoritmaları gibi, metinin özelliklerine ihtiyaç duymaktadır. E-ticaret sitelerinin taranması sonucunda elde edilen ürün bilgileri ham haldedir ve daha başarılı bir kümeleme yapılabilmesi için işlenmesi gerekmektedir. Bu yüzden elde edilen bilgiler, özel geliştirilen algoritmalar ile işlenmiş ve bu bilgilerden özellik vektörleri oluşturulmuştur. Ham bilgi yerine oluşturulan özellik vektörleri kullanılarak oldukça daha başarılı sonuçlar elde edilmiştir.

2.4.4. Özellik Vektörlerinin Seçilmesi (Boyut İndirgeme)

Genel olarak metin tabanlı sistemlerde, kök uzay boyutu (kelimeler, cümleler, vb.) oldukça büyük olduğundan pek çok algoritma, boyut indirgemesi yapılmadan çalışamamaktadır ve çalışsa bile oldukça büyük performans kaybı yaşanmaktadır [23]. Bu durum göz önüne alındığında, metin tabanlı kümeleme sistemlerinde uzay boyutunun indirgenmesinin zorunlu olduğu varsayılabilir. Fakat boyut indirgemesi yapılırken, kümeleme performansının kaybolmaması, hatta iyileşmesi yönünde çalışma yapılmalıdır. Tez kapsamında yapılan boyut indirgemesi

ile hem kümeleme performansı hem de kaynak kullanımı performansı yönünden iyileştirme yapılmıştır.

2.4.5. Gözetimsiz Öğrenme Yöntemi İle Metin Tabanlı Kümeleme Algoritmaları

Otomatik olarak yapılan metin tabanlı kümeleme, prensip olarak gözetimsiz (insan girdisi almadan) ve gözetimli (insan girdisi alarak) olmak üzere iki kısma ayrılır [24].

Gözetimli öğrenme, insan girdisi gerektirdiği için çok maliyetlidir ve gerçek yaşam problemlerinde uygulanması genel olarak çok zordur. Bu tez kapsamında gerçekleştirilen KAM tam bir gerçek yaşam problemidir ve çok büyük veri setleri ile kümeleme yapılmaktadır. Bu yüzden gözetimsiz öğrenme yöntemi ile çalışan algoritma geliştirmek çok büyük maliyet ve hız avantajı sağlayacaktır.

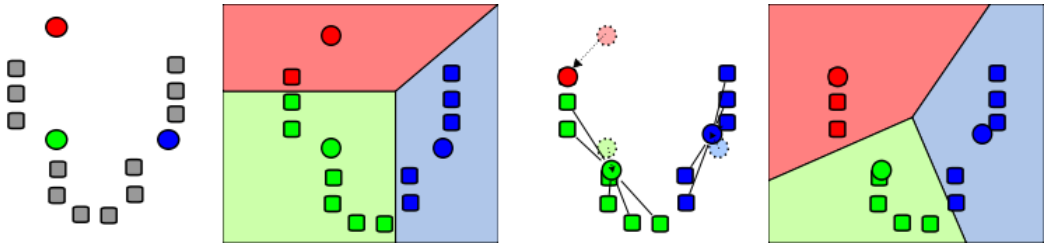
Gözetimsiz öğrenme yöntemi ile metin tabanlı kümeleme sistemleri üzerine çok sayıda literatür çalışması bulunmaktadır [24-37]. Çalışmaların çokluğu, giderek kontrolsüzce ve herhangi bir düzene sahip olmadan sayısı büyüyen elektronik dokümanların (internet) sınıflandırılmasının önemini ortaya koymaktadır. Yapılan literatür taramasında, aynı e-ticaret ürünlerinin tespit edilmesine yönelik sınıflandırma çalışmasına rastlanmamıştır.

Kümeleme algoritmaları 4 sınıfta toplanabilir [38]:

- Hariç Tutulan Kümeleme (Örnek: K-Means)
- Üst Üste Kesişen Kümeleme (Örnek: Fuzzy C-means)
- Hiyerarşik Kümeleme (Örnek: Hiyerarşik Kümeleme)
- Olasılıklı Kümeleme (Örnek: Mixture of Gaussians)

2.4.5.1. K-means algoritması

K-means algoritması kullanıcıdan kaç adet sınıf olduğu girdisi bilgisini alarak çalışır. İlk olarak algoritma rastgele verilen sınıf sayısı kadar merkez noktası atar ve daha sonra en yakın ortalamalara göre etraftaki öğeleri bu sınıflara koyar. Daha sonra oluşturulan bu sınıfların geometrik ortalamaları, yeni sınıf merkez noktaları olur. 2. ve 3. adım artık sonuçlar değişmez hale gelene kadar devam eder. Şekil 2.4'de adımlar gözükmektedir [39].



Şekil 2.4. K-means algoritması adımlar [39]

2.4.5.2. Hiyerarşik kümeleme algoritması

Hiyerarşik kümeleme algoritması aşağıdan yukarı ve yukardan aşağı olmak üzere 2 farklı yöntem ile çalışır. Aşağıdan yukarı yönteminde bütün veriler tek başına bir sınıftır ve bu veriler birleştirilerek yeni sınıflar elde edilir. Yukardan aşağı yönteminde ise bütün veriler tek bir sınıftır ve bu sınıf daha ufak sınıflara parçalanılarak devam edilir.

Parçalama veya birleştirme işlemleri için veriler arasındaki benzeşmezliği hesaplayacak ölçütlere ve veri setleri arasındaki benzeşmezlik uzaklığını hesaplayacak bağ ölçütlerine ihtiyaç vardır [40].

Çizelge 2.1'de benzeşmezliği hesaplamak için kullanılan yaygın ölçütler [40-41] ve Çizelge 2.2'de benzeşmezlik uzaklığını hesaplamak için kullanılan yaygın bağ ölçütleri gösterilmiştir [40,42-43].

Çizelge 2.1. Hiyerarşik kümeleme algoritması veriler arasındaki benzeşmezliği hesaplayacak yaygın ölçütler [40-41]

İsimler	Formüller
Öklid Mesafesi	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Kare Öklid Mesafesi	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan Mesafesi	$\|a-b\|_1 = \sum_i \|a_i - b_i\|$
Maksimum Mesafe	$\|a-b\|_\infty = \max_i \|a_i - b_i\|$
Mahalanobis Mesafesi	$\sqrt{(a-b)^\top S^{-1}(a-b)}$
Kosinüs Benzerliği	$\frac{a \cdot b}{\|a\|\|b\|}$

Çizelge 2.2. Hiyerarşik kümeleme algoritması veri setleri arasındaki benzeşmezlik uzaklığını hesaplayan yaygın bağ ölçütleri [40,42-43]

İsimler	Formüller
Maksimum (Tam Bağ) Kümelemesi	$\max\{\, d(a,b) : a \in A,\, b \in B \,\}.$
Minimum (Tek-Bağ) Kümelemesi	$\min\{\, d(a,b) : a \in A,\, b \in B \,\}.$
Orta (Ortalama) Bağ Kümelemesi	$\frac{1}{\|A\|\|B\|} \sum_{a \in A} \sum_{b \in B} d(a,b).$
Minimum Enerji Kümelemesi	$\frac{2}{nm} \sum_{i,j=1}^{n,m} \|a_i - b_j\|_2$ $- \frac{1}{n^2} \sum_{i,j=1}^{n} \|a_i - a_j\|_2$ $- \frac{1}{m^2} \sum_{i,j=1}^{m} \|b_i - b_j\|_2$

2.4.6. Test Süreci

Metin tabanlı sınıflandırma problemlerinde parametre sayısı oldukça fazla olmakta, parametreler kolayca tahmin edilememekte, parametreler çalışılan yönteme çok bağımlık gösterebilmekte ve bu yüzden parametrelerin seçimi performansı

oldukça etkileyebilmektedir [44]. Tüm bu sebeplerden dolayı, geliştirilen algoritmanın etkili bir test sürecine tabi tutularak olabildiğince doğru parametrelerin elde edilmesi ve algoritmanın gerçek performansının ortaya konması oldukça büyük öneme sahiptir. Doğru seçilmeyen parametreler, algoritmanın başarısına direk etki edebileceği gibi, performansa da oldukça etki edebilmektedir. Bu tez kapsamında, oldukça etkili bir test süreci yürütülerek, algoritmanın performansında iyileştirmeler gerçekleştirilmiştir.

2.5. KULLANICI ARAYÜZÜ

Kullanıcı ara yüzü geliştirmek, yazılım geliştirmenin temel taşlarından bir tanesidir ve yazılımın başarısının büyük bir kısmı kullanıcı ara yüzüne bağımlıdır [45]. Tam bir KAM tasarımının yapılabilmesi için web tabanlı kullanıcı ara yüzü tasarımı gerekmektedir. Tasarlanacak kullanıcı ara yüzü, arama yapacak kişilerden arama girdisini alacak, aldığı girdi ile daha önceden çevrimdışı olarak, geliştirilen algoritmalar ile kümelenmiş ürün grupları arasında tarama yapacak ve kullanıcıya aradığı ürün gruplarını listeleyecektir.

Web tabanlı olarak çalışması gereken KAM kullanıcı ara yüzünün, güncel tarayıcılar ile uyumlu olması, aranan ürünün bilgilerini detaylı şekilde listeleyebilecek ve kolayca kullanılabilir olması oldukça önemlidir. Bu tez kapsamında geliştirilen KAM için etkili bir web ara yüzü geliştirilmiştir ve hizmete açılmıştır.

2.6. E-TİCARET ÜRÜN KÜMELEME SİSTEMLERİ

Farklı web sitelerinde satılan aynı ürünleri tespit etmeye yönelik KAM tasarımının yapıldığı 2 adet çalışma mevcuttur [2-3]. Bu çalışmalarda çevrimdışı çalışarak farklı web sitelerinde satılan aynı ürünleri sınıflandırıp, daha sonra arama yapan kullanıcılara döndüren bir sistem yerine, ürünleri satan web sitelerinin dâhili arama sistemlerini otomatik olarak keşfeden ve kullanıcıdan girdi alındığı zaman bu sistemleri kullanarak sonuç döndüren bir yapı kullanılmıştır. Kullanılan bu sistem, bu tez kapsamında yapılan sisteminden tamamıyla farklıdır.

Gene e-ticaret sistemlerinde satılan ürünlerin özelliklerini kullanarak kullanıcılara ürün listesi döndürecek şekilde çalışan literatür çalışmaları

bulunmaktadır [4-6]. Bu çalışmalarda aynı ürünlerin tespiti hedeflenmemiş, kullanıcıdan alınan özellikleri sağlayan tüm ürünleri listeleme hedeflenmiştir. Ayrıca önerilen sistemlerde herhangi bir ürün kümelemesi veya sınıflandırma mevcut değildir.

Bu çalışmalardan başka, e-ticaret sistemlerindeki ürünlerin bilgilerini kullanarak kümeleme yapan sistemler bulunmaktadır [8-9,46-57]. Bu sistemlerde farklı web sitelerinde satılan aynı ürünleri kümelemek yerine, bir e-ticaret sistemindeki ürünleri kategorize etmek ve kataloglamak üzerine çalışılmıştır. Ayrıca gene ürün özelliklerini kullanarak yapılan farklı bir çalışma türünde, ürünlerin özellikleri ve çeşitli başka unsurlar kullanılarak, kullanıcıların ilgisini çekebilecek diğer en iyi ürünleri kullanıcıya göstermek üzerine çalışılmıştır [58].

3. MATERYAL VE YÖNTEM

Bu tez kapsamında yazılan gerekli yazılımlar ve geliştirilen algoritmalar için Microsoft Visual Studio (MVS) 2010 [59], Microsoft SQL Server (MSS) 2008 R2 [60], Matlab R2011a [61] programları kullanılmıştır.

3.1. MICROSOFT VISUAL STUDIO 2010

MVS, Microsoft tarafından üretilen dünyadaki en popüler yazılım geliştirme araçlarından bir tanesidir. Tez kapsamında yazılan web sitelerini tarayacak ve işleyecek bot yazılımı tamamıyla 0'dan ve MVS 2010 kullanılarak geliştirilmiştir.

Programlama dili olarak Visual C# [62], uygulama olarak C# 4.0 Windows Presentation Foundation (WPF) [63] ve alt yapı olarak Net Framework 4.0 [64] seçilmiştir. MVS 2010 ile gelen standart kütüphanelere ek olarak, Html Agility Pack (HAP) [65] isimli HTML çözümleyicisi kullanılmıştır.

3.2. MİCROSOFT SQL SERVER 2008 R2

MSS, Microsoft tarafından geliştirilen dünyadaki en popüler veri tabanı sistemlerinden bir tanesidir. Tez kapsamında geliştirilen tüm yazılımlar için veri tabanı sistemi olarak MSS 2008 R2 kullanılmıştır.

MSS standart Transact-SQL (T-SQL) komutları ile çalışmaktadır. Tez kapsamında geliştirilen uygulamaların kullandığı tüm yapılandırılmış sorgu dili (SQL) komutları 0'dan yazılmıştır. Herhangi bir yardımcı araç veya hazır kod kullanılmamıştır.

3.3. MATLAB R2011A

Matlab, algoritma geliştirmek, veri analizi yapmak, görselleştirme ve sayısal işlemler yapmak için dünyadaki kullanılan en meşhur programlama platformlarından bir tanesidir. Tez kapsamında Matlab sürüm R2011a ve R2009b kullanılmıştır.

Matlab kullanılarak algoritma geliştirilmiş, geliştirilen algoritmanın diğer meşhur algoritmalar ile karşılaştırılması yapılmıştır.

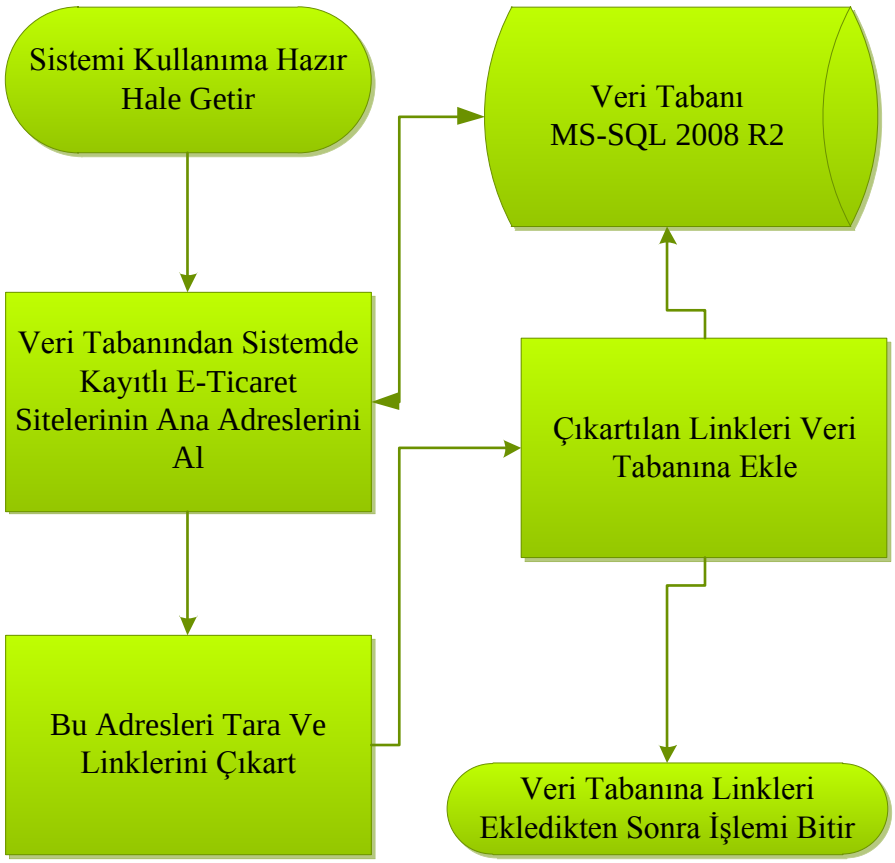
3.4. WEB SİTELERİNİ TARAYACAK BOT YAZILIM

3.4.1. Sitelerin Taranması

Tasarlanan KAM sisteminde, sitenin sadece ana adresi (tohum URL) verilmektedir. Tasarlanan site tarama ajanı, verilen e-ticaret sitelerinin ana adresinden tarama işlemine başlamaktadır. Her bir sitenin, 100,000'lerce sayfası olabildiği için tasarlanacak yazılımın çok iş parçacıklı çalışabiliyor olması şarttır. Bu yüzden tasarım çok iş parçacıklı çalışabilir şekilde tasarlanmıştır.

Tarama ajanı geliştirilirken diğer bir karşılaşılan zorluk ise, e-ticaret sitelerinin çok verimsiz şekilde tasarlanmış link yapısı olmuştur. Pek çok sitede sayfalardan elde edilen linkler direk olarak tarandığında görülmüştür ki, aynı sayfaya sonsuz sayıda link ile erişilmektedir. Bu şekilde herhangi bir iyileştirme yapılmadan tarama yapıldığında, bir sitedeki sayfaların taranmasının hiçbir zaman sonuçlanmayacağı veya çok sayıda gereksiz tarama ve aynı dokümanın indirilmesinin yapılacağı tespit edilmiştir. Bu yüzden programda, taranan linklerden atılması gereken kısımlar oluşturulmuş, hatta bazı sitelere özel düzenlemeler yapılması gerekmiştir. Optimum seviyeye ulaşana kadar milyonlarca sayfa taraması gerçekleştirilmiştir.

Program 2 ana aşama ile çalışmaktadır. İlk aşamada sistem kullanıma hazır hale getirilir. Bu işlemde her web sitesinin ana adresi taranır ve içerdikleri linkler çıkartılır. İlk aşamanın çalışma mantığı Şekil 3.1'de gösterilmiştir. Bu aşamadan sonra program istenildiği zaman kapatılabilir, yeniden başlatılabilir. Program her başlatıldığında henüz taramadığı linkleri taramaya devam eder. Bundan sonra döngüsel şekilde hiçbir taranmamış link kalmayana tarama ve link çıkartma işlemine devam edilir. Bu şekilde bütün e-ticaret sitelerinin içerdikleri bütün sayfalar taranır.



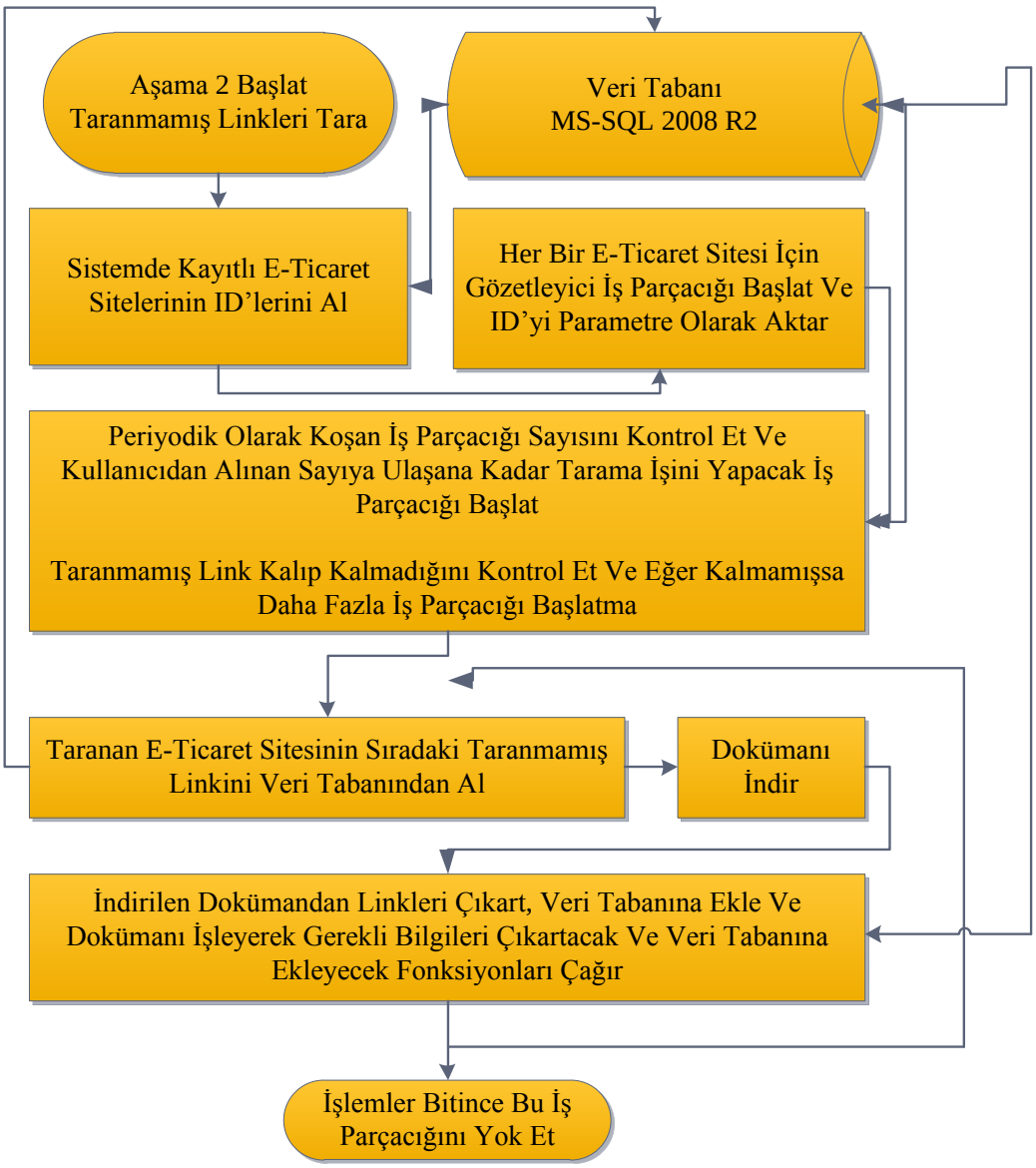
Şekil 3.1. Site tarama ajanı ilk aşama akış şeması

İlk aşama bittikten sonra yazılım başlatılırken kaç adet tarama ve işleme ajanının her bir site için başlatılacağı kullanıcı tarafından verilmektedir. Yani yazılımın, her bir e-ticaret sitesi için kaç adet iş parçacığını aynı anda yürüteceği. Örnek olarak eğer 5 adet iş parçacığı girdi olarak verilirse, sistemde kayıtlı olan her bir e-ticaret sitesini taramak için sürekli olarak 5 iş parçacığı çalışacaktır. Böylece saniyede 10'larca sayfa taranabilecek ve işlenebilecektir. Buradaki en büyük zorluk, farklı iş parçacıkları arasındaki iletişimi sağlayabilmektedir. Her iş parçacığından dönen sonuç, yeni başlayacak iş parçacıklarının alması gereken parametreyi direk olarak etkilemektedir. Bu yüzden çok sayıda iş parçacığı aynı anda çalıştırılırken çok detaylı bir sistem tasarımı yapılması zorunludur.

İş parçacığı sayısı verilerek yazılım başlatıldıktan sonra, yazılım ilk olarak veri tabanını sorgulayarak sistemde kayıtlı olan e-ticaret sitelerinin kimlik numarasını (ID) alır. Daha sonra her bir ID için gözlemci iş parçacığı başlatır. Bu gözlemci iş parçacıkları sonsuz döngü içerisinde sürekli olarak koşan iş parçacığı sayısını kontrol eder. Buradaki sayı, kullanıcıdan alınan her bir e-ticaret sitesi için

kaç adet iş parçacığının aynı anda çalışacağını belirleyen sayıdır. Sürekli olarak iş parçacıklarının koşup koşmadıklarını kontrol ederek koşan iş parçacığı sayısını sabit olarak kullanıcıdan alınan kadar tutar. Örnek olarak kullanıcı her e-ticaret sitesi için maksimum 5 iş parçacığı belirlemişse, gözlemci iş parçacığı sürekli olarak görevli olduğu e-ticaret sitesini tarayacak 5 iş parçacığı koşturacaktır. Bu tez kapsamında 20 adet e-ticaret sitesi [66-85] seçildiği için, toplam koşan tarayıcı iş parçacığı sayısı bir web sitesinde taranacak link kalmayana kadar daima 100 olacaktır. Gözlemci iş parçacığı, başlattığı her bir iş parçacığı işini bitirdikten sonra veri tabanını sorgulayarak görevli olduğu e-ticaret sitesi için taranmamış link kalıp kalmadığına bakar. Eğer taranmamış link kalmamışsa daha fazla iş parçacığı başlatmaz ve görevini sonlandırır.

Görevli iş parçacıkları tarafından oluşturulan her iş parçacığı veri tabanına sorgu yollayarak taraması gereken sıradaki taranmamış linki alır ve o sayfayı indirir. İşe başlayan parçacıkların aynı linki veri tabanından çekerek işlemeye başlamaması için, yazılan SQL sorguları ile veri tabanı sistemi, iş parçacıkları arasındaki koordinasyonu sağlamak için kullanılmıştır. İndirilen sayfanın içerdiği tüm linkler çıkartılır ve veri tabanında mevcut bulunmayanları veri tabanına eklenir. İndirilen sayfanın içeriğinden linklerin çıkartılması için HAP kütüphanesinden faydalanılmıştır. Daha sonra indirilen sayfanın içeriği, her bir e-ticaret sitesi için özel olarak yazılmış işleme fonksiyonlarına verilir ve işlenerek tez kapsamında geliştirilen KAM sisteminin çalışması için gerekli veriler çıkartılır. Çıkartılan bu veriler yazılan fonksiyonlar ile veri tabanına kaydedilir. Tüm bu işlemler bittikten sonra iş parçacığı yok edilir. Böylece gözetimci iş parçacığı koşan iş parçacıklarını kontrol ettiğinde işi biten iş parçacığını tespit eder ve yeni bir tarayıcı iş parçacığı başlatır. Aşama 2'nin çalışma mantığı Şekil 3.2'de gösterilmiştir.



Şekil 3.2. Site tarama ajanı 2. aşama akış şeması

Tarama ajanı daha önce eklediği bir linki tekrardan veri tabanına eklememek için, elde ettiği linkleri gürültü elemesinden geçirmiş ve normalleştirmiştir. Bunun için uzun analizler sonucunda, elde edilen linklerden atılması gereken parametreler listesi oluşturulmuş, ayrıca linkler kesin link formuna dönüştürülmüştür. Tüm bunlara rağmen e-ticaret sitelerinin düzgün normlara uymadan tasarlanan yapıları sebebi ile aynı ürünü farklı linkler ile listeledikleri tespit edilmiştir. Bu yüzden tüm tarama işlemi bittikten sonra, 2 veya daha fazla defa farklı

linkler ile listelenen ürünler, teke indirilmiştir. Bu sayede bir e-ticaret sitesinde satılan ürünlerin sadece 1 defa veri tabanında saklanması sağlanmıştır.

3.4.2. Taranan Sitelerden Gerekli Bilgilerin Çıkartılması – Veri Toplama

Tarama ajanı yardımı ile taranan sayfaların ham içeriğinin işlenerek, tez kapsamında geliştirilen KAM tasarımı için gerekli olan bilgilerin çıkartılması gerekmektedir. Türkiye'deki e-ticaret sitelerinin pek çoğu aynı alt yapıyı kullansa da, siteler arası farklılıklar olmaktadır. Bu yüzden her bir e-ticaret sitesi için ayrı ayrı ham html verisini işleyerek gerekli bilgileri döndürecek fonksiyonlar yazılmıştır. İlk başta bu işlem çok maliyetli gibi gözükse de, bilgilerin tam olarak doğru elde edilmesi, geliştirilen algoritmaların doğruluğu açısından çok önemlidir. Ayrıca Türkiye'de sınırlı sayıda e-ticaret sitesi bulunmakta ve sitelerin pek çoğu aynı alt yapıyı kullanmaktadır. Bu yüzden bir e-ticaret sitesi için yazılan fonksiyon ufak veya hiçbir değişiklik yapılmadan diğer bir e-ticaret sitesi için çalışmaktadır. Ayrı ayrı fonksiyon yazılması kabul edilebilir maliyettedir. Taranan siteler ve istatistikleri Çizelge 3.1'de gösterilmiştir.

Çizelge 3.1. Taranan e-ticaret siteleri ve istatistikleri

Site Adresi	Bulunan Ürün Sayısı	Taranan Sayfa Sayısı
vatanbilgisayar.com	5.803	6.611
hepsiburada.com	177.310	313.946
gold.com.tr	6.018	7.383
hizlial.com	84.046	166.197
ereyon.com.tr	68.960	92.076
webdenal.com	69.979	121.853
darty.com.tr	3.183	4.228
exa.com.tr	2.144	3.142
eksenbilgisayar.com	4.245	4.558
pratikev.com	63.170	69.275
incehesap.com	3.857	9.547
netsiparis.com	40.525	59.294
pcdepo.com	478	607
teknobiyotik.com	2.165	2.937
webdenbul.com	12.085	16.379
birnumaram.com	28.361	35.598
sanalmarketim.com	13.059	18.060
bimeks.com.tr	8.149	11.741
inventus.com.tr	1.001	1.266
novabilgisayar.com	1.334	1.849

3.4.2.1. Ürün sayfalarından çıkartılan bilgiler

Planlanan KAM tasarımında bir ürün sayfasından, sayfa başlığı, sayfa tanımı, sayfa anahtar kelimeleri, ürün ismi, ürün fiyatı ve ürün kategorisi bilgilerinin çıkartılması planlanmış ve fonksiyonlar verilen ham html verisinden bu bilgileri

çıkartacak şekilde tasarlanmıştır. İlk başta bu bilgilerin tamamının kümeleme algoritmalarında kullanılması planlanmışken, geliştirme sürecinde en verimli sonuçların sadece ürün ismi kullanılarak elde edildiği tespit edilmiştir.

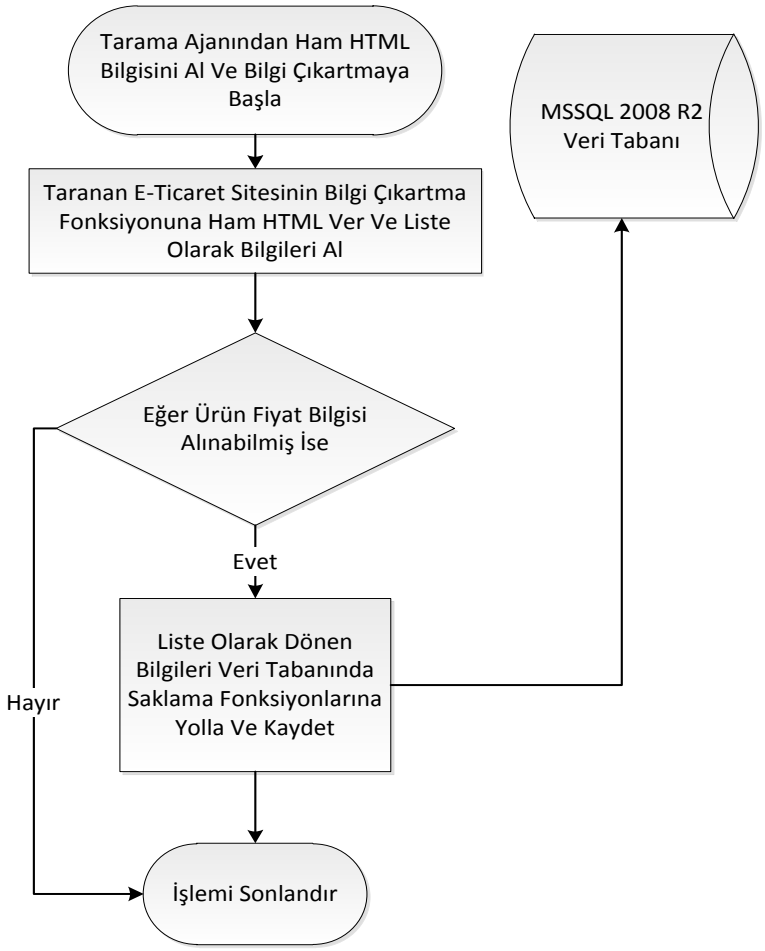
3.4.2.2. Çalışma prensibi

Tarama ajanı sayfayı indirdikten sonra taradığı e-ticaret sitesinin bilgi çıkartma fonksiyonuna, indirdiği sayfanın ham html içeriğini yollar. Fonksiyondan dönen sonuçlar ise veri tabanına kaydetme fonksiyonlarına gönderilir ve veri tabanında saklanır. Sadece bilgi çıkartma fonksiyonları sitelere özel yazılmış, diğer fonksiyonlar ise genel olarak tasarlanmıştır. Eğer bilgi çıkartma fonksiyonu ürünün fiyat bilgisini alamaz ise, kendisine yollanın sayfanın ürün sayfası olmadığını varsayar. Fiyat bilgisi alınamayan sayfaların diğer bilgileri de veri tabanında saklanmaz.

Bir sayfadan fiyat bilgisinin alınmaması iki durumda olmaktadır. Eğer o sayfa bir ürün sayfası değil ise veya o ürün stoklarda kalmamış ise. Ürün araması yapan kullanıcı stoklarda olmayan ürünleri görmek istemeyeceği için ürün sayfası değil şeklinde varsayılmıştır.

Bilgi çıkartma işleminde karşılaşılan en büyük zorluk, tamamlanmamış html öğelerinin olduğu veya herhangi bir ayırt edici sınıf ve kimlik numarası kullanılmamış e-ticaret sistemleri olmuştur. Bu tarz tasarımlara sahip sitelerden doğru şekilde bilgileri çekebilmek için oldukça gelişmiş fonksiyonlar yazılması gerekmiştir.

Gerekli bilgilerin çıkartılması sürecindeki aşamalar Şekil 3.3'de gözükmektedir.



Şekil 3.3. Taranan bir sayfadan ürün bilgilerinin çıkartılması

3.5. NORMALLEŞTİRME VE GÜRÜLTÜ ELEME

Geliştirme sürecinde elde edilen ürün bilgilerinin direk olarak kullanılması yerine çeşitli işlemlerden geçirilerek kullanılmasının daha iyi sonuçlar ürettiği test aşamasında tespit edilmiştir. Türkiye'deki siteler İngilizce ve Türkçe terimler kullanmakta ve her iki dilde de kelimeler boşluk karakteri ile ayrıldığı için elde edilen ürün bilgileri boşluk karakteri ile kelimelere dönüştürülmüştür.

3.5.1. Harf Bazında Normalleştirme Ve Gürültü Eleme

Bir kelimeyi bazı e-ticaret siteleri Türkçe karakter kullanarak bazıları ise kullanmayarak yazdığı için, Türkçe karakterler İngilizce karşılıklarına dönüştürülmüştür. Örnek olarak ş harfi s harfine dönüştürülmüştür. Sınıflandırmada anlamlı olmayacak karakterleri elemek için, a-Z, 0-9 haricindeki karakterler boşluk karakteri haline getirilmiştir. Metin karşılaştırmalarında tam performans elde

edebilmek için tüm kelimeler küçük harfe çevrilmiştir. Son olarak 1 karakter boyutundaki kelimeler kullanılmamıştır.

3.5.2. Kelime Bazında Normalleştirme Ve Gürültü Eleme

Elde edilen verilerin analizi göstermiştir ki e-ticaret siteleri, ürün bilgilerini girerken sıklıkla harf hataları yapmakta, ayrık kelimeleri bitişik yazmakta veya bitişik kelimeleri ayrık yazmakta ve kelimeleri eksik yazmaktadır. Tüm bu hataların sonucu oluşan kelimeler ise karşımıza gürültü olarak çıkmaktadır. Bu tarz kelimelerin yok edilmesi veya doğru haline dönüştürülmesi oldukça karmaşık algoritmalar gerektirmektedir. Geliştirilen özellik vektörü oluşturma algoritmasında kelime bazında gürültü eleme ve normalleştirme başarılmıştır.

3.6. PARAMETRELERİN BELİRLENMESİ, ÖZELLİK VEKTÖRLERİNİN OLUŞTURULMASI VE BOYUT İNDİRGEME

Metin tabanlı sistemlerde belirlenen parametreler direk olarak sonuçların başarısına etki etmektedir. Bu yüzden parametrelerin doğru şekilde belirlenmesi büyük bir önem arz etmektedir. Hangi parametrelerin kullanılacağını ise baştan belirlemek oldukça zordur.

3.6.1. Parametrelerin Belirlenmesi

Metin tabanlı kümeleme sistemlerinin genelinde, elde var olan bilgiler ilk olarak özellik vektörlerine dönüştürülür ve bilgilerin işlemden geçirilmiş hali olan bu vektörler kullanılarak kümeleme işlemi gerçekleştirilir. Bu şekilde kümeleme algoritmalarının doğruluk oranı ve performansı iyileştirilir. Tez kapsamında geliştirilen site tarama ajanı her ürün sayfasından 6 adet bilgiyi veri tabanına kaydetmiş fakat yapılan veri analizleri ve testler sonucunda özellik vektörlerinin oluşturulması sürecinde sadece ürün isimlerinin kullanılmasının daha mantıklı olduğu ve iyi sonuçlar üreteceği tespit edilmiştir.

Gözetimsiz öğrenme yöntemi kullanan metin tabanlı algoritmalarda mutlaka çeşitli eşik değerlerinin belirlenmesi gerekmektedir. Gözetimsiz öğrenme yöntemi ile çalışan algoritmaların en büyük zorluklarından bir tanesi de budur. Yanlış şekilde belirlenen eşik değerleri algoritmaların ulaşabilecekleri başarı seviyelerinin oldukça

altında kalmalarına sebep olabilmektedir. Özellik vektörlerini çıkartan algoritmada belirlenen eşik değerleri mantıksal çıkarımlar, veri analizleri ve testler yapılarak belirlenmiştir.

3.6.2. Özellik Vektörlerinin Oluşturulması

Tez kapsamında geliştirilen KAM'nin en güçlü algoritması özellik vektörlerini çıkartan algoritma olmuştur. Geliştirilen algoritma ile elde edilen özellik vektörleri, hiç işlenmemiş şekilde ham bilgilerin kullanılarak elde edilen özellik vektörlerinin kullanılmasına göre oldukça iyi sonuçlar ürettiği testler ile gözlemlenmiştir.

3.6.2.1. İlk gruplama ile daha odaklı veri işleme

Kelime bazında gürültü elemesinin daha verimli yapılabilmesi için, evrensel olarak tüm ürünler üzerinde çalışmak yerine, bir ürünün sadece kelime bazında diğer ürünlerle gruplanması ve bu grup içindeki kelimelerin kullanılarak gürültü elemesinin yapılmasının daha iyi sonuçlar ürettiği gözlemlenmiştir.

İlk üründen başlanarak tüm ürünler, geri kalan tüm ürünler ile kelime bazında karşılaştırılır ve eşik değerini aşanlar aynı gruba girmiş sayılır. Oluşturulan bu gruplar eğitim setleri olarak kullanılır. Vektör (i) i'inci ürünün özelliklerini göstersin, min fonksiyonu ise verilen vektörlerin kelime boyutu küçük olanın kelime boyutunu döndürsün. Karşılaştırma için kullanılan yöntem Eşitlik (3.1)'de gösterilmiştir. Bu karşılaştırma yöntemi yerelde çalıştığı için Normal Benzerlik olarak adlandırılmıştır.

$$\text{Benzerlik}\,(i,j) = \frac{\text{eleman sayısı}\,\{\text{vektör}(i) \cap \text{vektör}(j)\}}{\min\{\text{vektör}(i), \text{vektör}(j)\}} \times 100 \qquad (3.1)$$

İkinci bir karşılaştırma yöntemi daha geliştirilmiştir. Bu karşılaştırma yöntemi ise Matlab'ın bir özelliğinden faydalanmaktadır. Matlab rakamlar üzerine kurulu olduğu için rakam tabanlı olarak çok daha hızlı çalışmaktadır. Bu nedenle kelimeler rakamlara dönüştürülerek işlemler gerçekleştirilmiştir. Kelimeler rakamlara dönüşürken Matlab kelimeleri sıralayarak indis atamıştır. Bu sebeple çok

benzer kelimeler çok yakın indislere sahip olmuştur. Örnek olarak okul ve okula kelimeleri varsa bunların indisleri peş peşe gelecektir. İndis atama işlemi evrensel olarak gerçekleştirildiği için bu yöntem Evrensel Mesafe benzerliği olarak adlandırılmıştır. Vektör (i) i'inci ürünün özelliklerini, kelime (i) i'inci kelimenin karşılık gelen indis değerini göstermiş olsun. Min ve max fonksiyonları benzerlik matrisin en küçüğünü ve en büyüğünü bulan fonksiyonlardır. Bu durumda verilen 2 vektör, vektör1, vektör2 arasındaki benzerlik oranı Eşitlik (3.2)-(3.4)'de gösterildiği şekilde hesaplanmıştır.

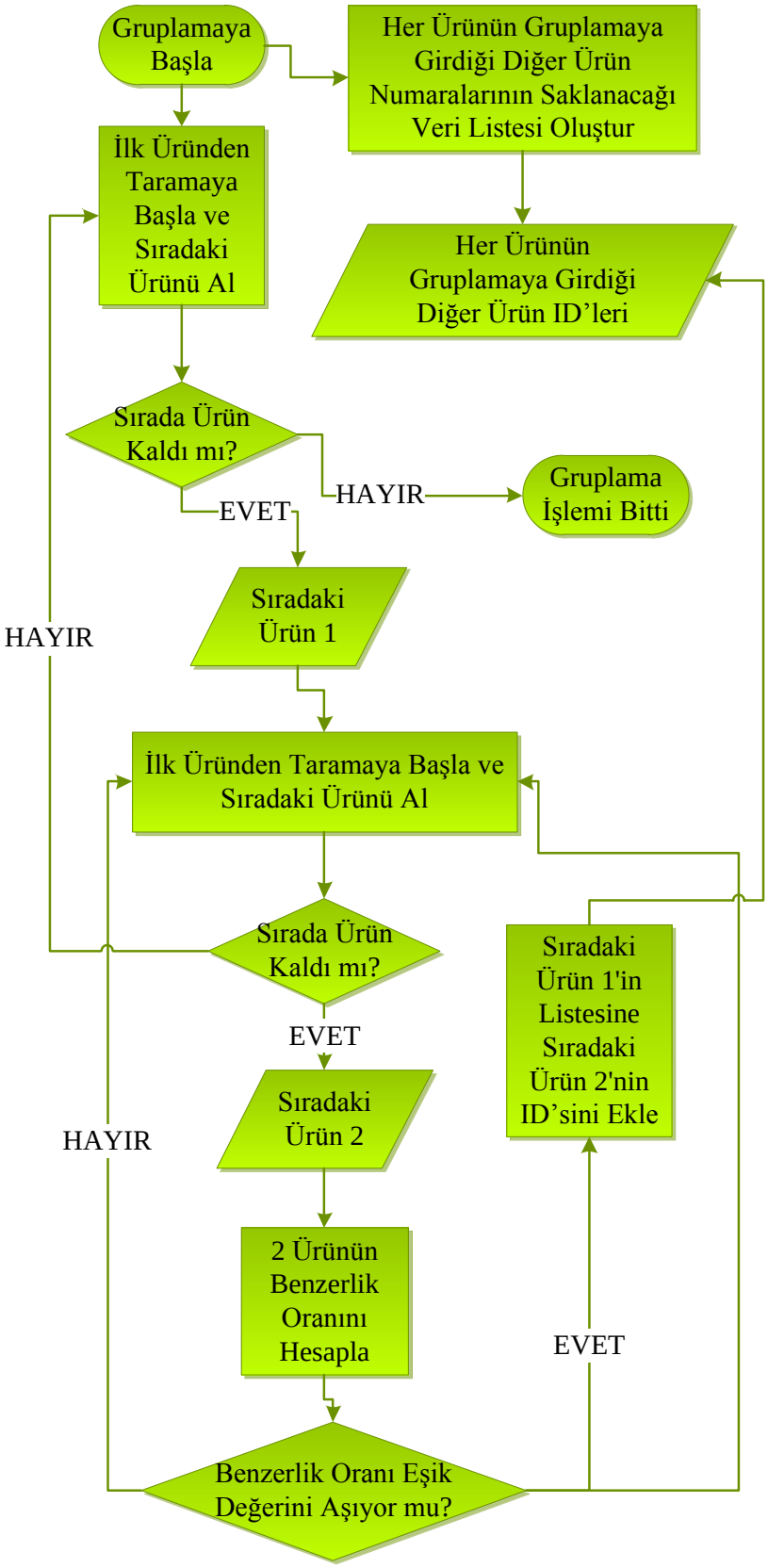
$$\text{Benzerlik (i, j)} = \sum_{k\epsilon\{vektör(i) \cap vektör(j)\}} kelime(k) \tag{3.2}$$

$$\text{Benzerlik (i, j)} = Benzerlik\ (i,j) - \min(Benzerlik) \tag{3.3}$$

$$\text{Benzerlik (i, j)} = \frac{Benzerlik\ (i,j)}{\max(Benzerlik)} \tag{3.4}$$

Eşitlik (3.2)-(3.4)'deki yöntem sadece standart kümeleme algoritmalarının Matlab'da test edilmesi sürecinde kullanılmıştır. Standart kümeleme algoritmalarına elde edilen yeni özellik vektörlerinin arasındaki mesafe verilmiştir ve bu mesafeyi hesaplamak için Evrensel Mesafe benzerliği yöntemi kullanılmıştır.

Eğitim setleri oluşturulurken eşik değeri %60 olarak belirlenmiştir. İç içe 2 döngü ile tüm ürünlerin diğer tüm ürünler ile karşılaştırılması sağlanmıştır. Yazılım çok iş parçacıklı olarak çalışabilmekte ve bu sayede süreyi oldukça kısaltmaktadır. Dış döngünün o anki mevcut ürününe Ürün 1, iç döngünün o anki mevcut ürününe Ürün 2 dersek, bu iki ürünün bilgileri karşılaştırılır ve benzerlik oranı hesaplanır. Eğer eşik değerini aşıyorsa Ürün 1'in gruplandığı diğer ürünler listesine, Ürün 2'nin ID'si eklenir. İşlem 1. döngüdeki tüm ürünler taranana kadar tekrar eder. Şekil 3.4'de tüm ürünlerin diğer tüm ürünler ile gruplanmasının akış şeması gösterilmiştir.



Şekil 3.4. Tüm ürünlerin diğer tüm ürünler ile gruplanması akış şeması

3.6.2.2. Grupların gözetimsiz olarak eğitilerek özellik vektörlerin oluşturulması

Özellik vektörlerini oluştururken doğru öğelerin seçilmesi, kümeleme algoritmalarının sonuçlarını direk olarak etkilemektedir. Bu yüzden çok başarılı şekilde oluşturulmuş özellik vektörleri, başarısız kümeleme algoritmalarını bile kabul edilebilir düzeye getirecektir.

Tez kapsamında gerçek hayat verisi ve problemi üzerinde çalışıldığı için elde edilen veriler oldukça fazla gürültüye sahiptir. Gürültünün elenmesi için harf bazında iyileştirilmeler yapılmış olsa da, kelime bazında anlamsal olarak gürültü elemesinin yapılması gözetimsiz olarak oldukça zordur. Gözetimli olarak yapılması ise maliyet açısından mümkün değildir. Bu problemin üzerinden gelmek için geliştirilen algoritma, bir önceki aşamada oluşturulan eğitim setlerini kullanmaktadır. Oluşan eğitim setlerine örnek Çizelge 3.2'de gösterilmiştir.

Çizelge 3.2. Oluşan eğitim ürün gruplarına örnek

Eğitim Seti	Aynı Gruba Giren Diğer Ürün ID'leri										
24	22	23	25	26	27	28	29	30	31		
28	22	23	24	25	30						
244	236	237	238	239	241	246					
245	240	243									
316	327	334	354	360							
320	297	299	321	322	323	326	330	332	333	335	340

Her ürün Çizelge 3.2'de gözüktüğü gibi başka ürünlerle gruplanmıştır. Yani ürün sayısı kadar grup oluşmuştur. Algoritma 1. gruptan (1. ürünün oluşturduğu) başlayarak bütün grupları eğitmektedir. Eğitim sürecinin temel prensibi 4 temel aşamadan oluşmaktadır.

1. Bir ürünü tanımlayan kelimelerin tanımlayıcı özelliği o kelimenin gruptaki olasılığı olarak görülebilir (tanımlayıcı olasılık değeri (TOD)). Bu durumda ürünü tanımlayan kelimelerin olasılıkları toplamı 1'e yaklaşmalıdır.
2. Aynı ürünü tanımlayan kelimelerin tanımlayıcı özelliği aynı ürünlerin kelime ortalamalarına yaklaşmalıdır.

3. Adım 1 ve adım 2 algoritma bir değere yakınsayana kadar ya da istenilen döngü sayısı elde edilene kadar grubu eğitir.
4. Özellik vektörlerinin uzunluklarının gruptaki ortalamaya yaklaştırılması. Bu şekilde gürültüye ve uzunluktaki dengesizliğe karşı normalleştirme sağlanır.

İlk olarak grubun içindeki tüm kelimelerin frekansları çıkartılır. Daha sonra o grubun içindeki ürün bilgilerinin kelimeleri yerine bu frekans değerleri yazılır. Her grup için oluşan matrislere örnek Çizelge 3.3'de gösterilmiştir.

Çizelge 3.3. Oluşturulan eğitim grupları içindeki kelime frekanslarına örnek

samsung	n150				4	3		
samsung	intel	atom	n570		4	2	2	1
samsung	n150	jp0xtr	atom		4	2	1	2
samsung	n150	intel			4	3	2	

Bu aşamadan sonra aşağıdaki adımlar gerçekleştirilir:

1) Birinci aşamada her satırdaki kelimelerin frekansı satırdaki toplam frekansa bölünür (Eşitlik (3.5)). Çıkan sonuç, kelimenin TOD'si olarak düşünülür. TOD'lar bulunduktan sonra kelimelerin tanımlayıcısı olarak kullanılır. Bu işlemden sonra örnek matris Çizelge 3.4'de gösterilmiştir.

$$Kelimenin\ yeni\ TOD'si\ = \frac{Kelimenin\ mevcut\ TOD'si}{O\ satırdaki\ TOD'ler\ toplamı} \tag{3.5}$$

Çizelge 3.4. Eğitim gruplarının ilk aşama eğitilmesi sonrası matris

samsung	n150				4/7	3/7		
samsung	intel	atom	n570		4/9	2/9	2/9	1/9
samsung	n150	jp0xtr	atom		4/9	2/9	1/9	2/9
samsung	n150	intel			4/9	3/9	2/9	

2) İkinci aşamada kelimelerin elde edilen TOD'leri sütun bazında toplanır ve ortalaması alınır. Ortalama TOD o kelimenin yeni TOD'si olur. Burada sütun bazında alınırken sadece o kelimenin tanımlayıcı TOD'si alınır. Sütun sırasının önemi yoktur. İkinci işlemden sonraki örnek matrisin son hali Çizelge 3.5'de gösterilmiştir. Yapılan işlem Eşitlik (3.6)'de gösterilmiştir.

Çizelge 3.5. Eğitim gruplarının ikinci aşama eğitilmesi sonrası matris

samsung	n150				47/100	32/100		
samsung	intel	atom	n570		47/100	2/9	2/9	1/9
samsung	n150	jp0xtr	atom		47/100	32/100	1/9	2/9
samsung	n150	intel			47/100	32/100	2/9	

$$Kelimenin\ yeni\ TOD'si = \frac{Tüm\ satırlardaki\ o\ kelimenin\ TOD'leri\ toplamı}{O\ kelimenin\ bulunduğu\ satır\ sayısı} \quad (3.6)$$

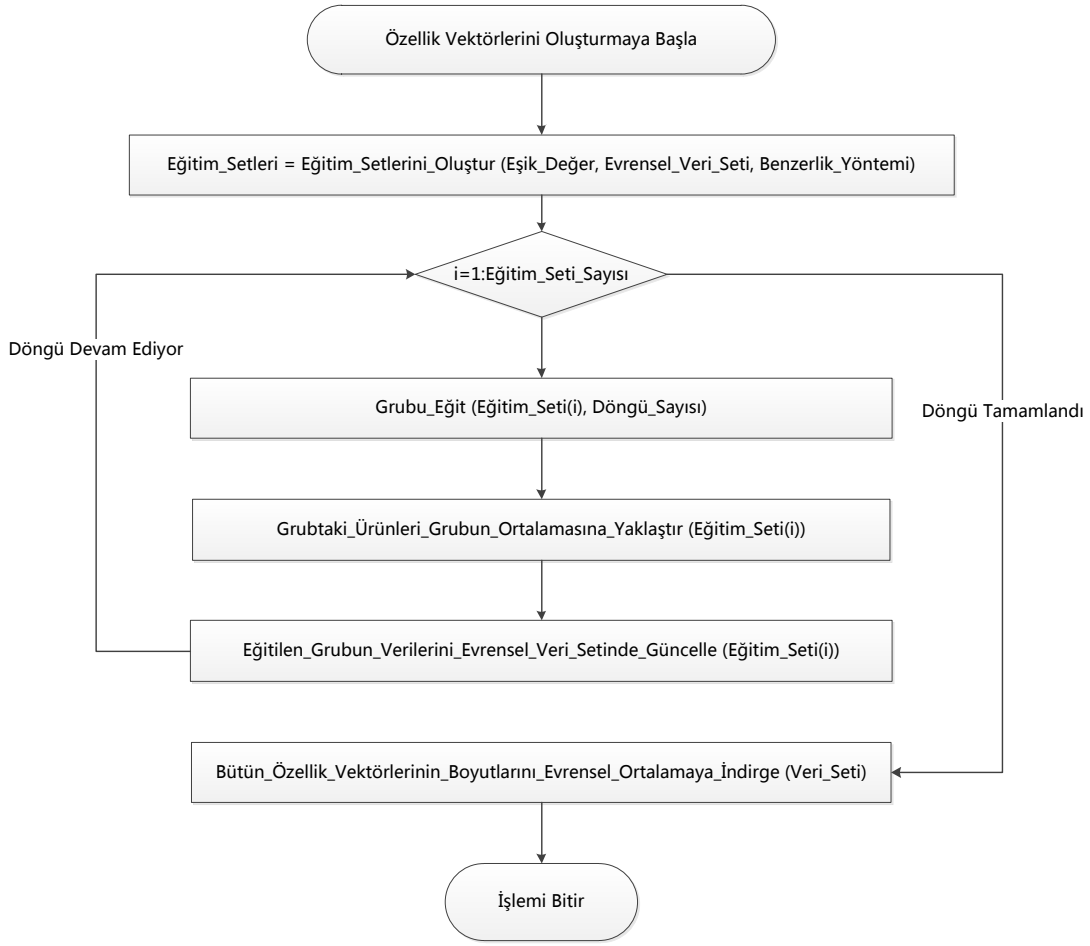
Üçüncü aşama olarak belirlenen döngü sayısı kadar bu 2 aşama tekrarlanır. Yapılan denemeler sonrası 20 döngünün gayet iyi sonuçlar elde ettiği tespit edilmiştir.

Dördüncü aşamada döngüler bittikten sonra kelimeler, son durumdaki TOD'lerine göre büyükten küçüğe doğru sıralanarak bir diziye aktarılır. Daha sonra gruptaki ürünlerin ortalama kelime sayısı bulunur. Ortalama kelime sayısından daha az kelimeye sahip ürünlere, kelime TOD'lerine göre sıralanmış listeden içermedikleri 2 adet yeni kelime eklenir. 2 adet kelimenin bir tanesi kategori adayı diğeri detaylı ürünü tanımlayan tanımlayıcı olarak düşünülmüştür. Daha sonra gruptaki bütün ürünlerin kelimeleri, yeni TOD'lerine göre büyükten küçüğe doğru sıralanır. En son olarak o grubun ortalama kelime sayısından daha büyük sayıda kelime içeren ürünlerinin sondan başlanarak ortalama kelime sayısına gelene kadar kelimeleri atılır. Bu aşamada bittikten sonra bu ürünlerin son halleri evrensel ürün listesinde güncellenir.

Bu şekilde bütün ürünlerin oluşturdukları gruplar işlenir. Bir önceki grubun sonucu bir sonraki grubu etkilemektedir. Sebebi ise her grupta eğitilen ürün

bilgilerinin evrensel ürün bilgisi listesinde güncellenmesidir. Bütün grupların işlenmesi bittikten sonra, evrensel ürün listesinin ortalama kelime boyutu hesaplanır ve yeniden bu boyutun üzerindeki ürünlerin kelimeleri sondan başlanarak ortalama boyuta gelene kadar çıkartılarak boyut küçültme yapılır.

Algoritmanın, harf hatası veya ayrık yazılan kelimeleri doğru halleri ile değiştirdiği ve tanımlayıcı unsuru olmayan kelimeleri ürün bilgisinden çıkartarak çok güçlü bir şekilde kelime bazında gürültü elemesi yaptığı gözlenmiştir. Bu aşamadan sonra ürünlerin bilgileri kümeleme algoritmalarında kullanılmak üzere işlenmiş özellik vektörlerine dönüşmüştür. Geliştirilen yeni algoritmanın akış şeması Şekil 3.5'de gösterilmiştir.



Şekil 3.5. Özellik vektörlerini oluşturan algoritmanın akış şeması

3.6.3. Ham Verilerin İşlenme Süreci Örnek

Çizelge 3.6 ve Çizelge 3.7'da manuel olarak oluşturulmuş test setindeki bir Canon renkli kartuş ürününün ve Panasonic video kamera ürünün satıldığı tüm web sitelerindeki listelenme bilgilerinin işlenmemiş hali gösterilmiştir. Algoritmanın başarısının daha iyi gösterilmesi için yapılan örneklerde küçük harfe dönüştürürken Türkçe karakterler İngilizceye dönüştürülmemiştir.

Çizelge 3.6. Manual sınıflandırılmış Canon renkli kartuş ürün kümesinin işlenmemiş hali

Ürün ID	**Ürün İsmi**
34632	Canon CL-41 Mürekkep Kartuş Renkli (41)
1417152	Canon CL-41 Renkli Kartuş
131538	CANON CL-41 RENKLİ
15088	Canon CL-41 Renkli MP150/160/170 Kartuş CAN22309
2521523	CANON CL-41 RENKLİ KARTUŞ
424323	Canon KARTUŞ CL-41 RENKLİ MP140/150/160/170
188038	CANON 1159186 CL-41 RENKLİ
2388138	Canon Cl-41 Mürekkep Kartuş
477899	CANON CL-41 RENKLİ KARTUŞ(IP1600/2200/6220/ML150)
1505334	CANON CL-41 RENKLI KARTUS
2401575	Canon CL-41 (41) Üç Renkli Kartuş (IP1200 / IP1600 / MP150)
2436897	CANON CL-41 Üç Renkli Kartuş MP140 MX310 MP450
2406650	Canon CL-41

Çizelge 3.7. Manual sınıflandırılmış Samsung netbook ürün kümesinin işlenmemiş hali

Ürün ID	**Ürün İsmi**
9065	Samsung N150-JP0XTR N570 2GB 320GB 10.1"" W7STR
4907	SAMSUNG N150-JP0XTR BEYAZ INTEL ATOM N570 1.66 GHz-2048MB DDR3-320GB -10.1"-CAM-BT-W7STR
4875	SAMSUNG N150-JP0XTR Atom N570 1.66GHZ 2GB 320GB 10.1" Netbook W7S Beyaz
41242	Samsung N150-JP0XTR Intel Atom N570 1.66Ghz 2Gb 320Gb 10.1" Win7Starter
169120	Samsung N150JP0XTR N570 2G 320GB 10.1 W7S BEYAZ
168088	SAMSUNG N150-JP0XTR W
243325	SAMSUNG NP-N150-JP0XTR Beyaz Atom N570 2GB 320GB Paylaşımlı Vga GMA3150 10.1" Win 7 Starter
2351726	Samsung NP-N150-JP0XTR NC110 INTEL ATOM N570 2/320/ Win7S/ 10.1 BEYAZ
1326311	SAMSUNG N150-JP0XTR NETBOOK N570/2GB/320GB/10,1/BEYAZ
2398358	Samsung Intel Atom N570 1.66GHz 2GB 320GB 10.1" Beyaz Netbook (N150-JP0XTR)
2451624	NP-N150-JP0XTR SAMSUNG NC110 INTEL ATOM N570 2/320/ Win7S/ 10.1 BEYAZ
2504542	NP-N150-JP0XTR SAMSUNG NC110 INTEL ATOM N570 2/320/ Win7S/ 10.1 BEYAZ
2421713	SAMSUNG NP-N150-JP0XTR Beyaz Atom N570 1.66GHz 2GB 320GB 10.1" Win 7 Starter
2405496	Samsung NP-N150-JP0XTR (White)
12037	SAMSUNG N150-JP0XTR Atom N570 1.66GHZ 2GB 320GB 10,1" Windows 7 Starter Beyaz Netbook

Çizelge 3.6 ve Çizelge 3.7'da gözüktüğü üzere bazı web siteleri daha detaylı ürün acıkmaları yazarken, bazı web siteleri ise oldukça kısıtlı tanımlama yapmışlardır. Bazı web siteleri ürünü tanımlamayan çok genel bilgileri ürün tanımlarına eklemişken, bazı web siteleri bitişik yazması gereken kelimeleri ayrı yazmış, bazı web siteleri de ayrı yazması gereken kelimeleri bitişik yazmıştır. Bazı web sitelerinde ise gereksiz kelimeler ürün tanımlamalarında kullanılmıştır. Bu bilgilerin herhangi bir işlemden geçirilmeden metin tabanlı algoritmalar ile kümelenmesi ise oldukça zordur. Veriler harf bazında gürültü elemesi (1. Seviye) yapıldıktan sonra kümeleme algoritmalarında işe yaramayacak bir kısım işaretler ve tek harfli kelimelerden arındırılmış ayrıca bütün kelimeler küçük harfe dönüştürülerek normalleştirme yapılmıştır. Bu işlemlerin ardından Çizelge 3.6 Çizelge 3.8'deki hale, Çizelge 3.7 ise Çizelge 3.9'deki hale dönüşmüştür.

Çizelge 3.8. Manual sınıflandırılmış Canon renkli kartuş ürün kümesinin harf bazında gürültü elemesi yapıldıktan sonraki hali

Ürün ID	**Ürün İsmi**
34632	canon cl 41 mürekkep kartuş renkli 41
1417152	canon cl 41 renkli kartuş
131538	canon cl 41 renkli
15088	canon cl 41 renkli mp150 160 170 kartuş can22309
2521523	canon cl 41 renkli kartuş
424323	canon kartuş cl 41 renkli mp140 150 160 170
188038	canon 1159186 cl 41 renkli
2388138	canon cl 41 mürekkep kartuş
477899	canon cl 41 renkli kartuş ip1600 2200 6220 ml150
1505334	canon cl 41 renkli kartus
2401575	canon cl 41 41 üç renkli kartuş ip1200 ip1600 mp150
2436897	canon cl 41 üç renkli kartuş mp140 mx310 mp450
2406650	canon cl 41

Çizelge 3.9. Manual sınıflandırılmış Samsung netbook ürün kümesinin harf bazında gürültü elemesi yapıldıktan sonraki hali

Ürün ID	**Ürün İsmi**
9065	samsung n150 jp0xtr n570 2gb 320gb 10 w7str
4907	samsung n150 jp0xtr beyaz intel atom n570 66 ghz 2048mb ddr3 320gb 10 cam bt w7str
4875	samsung n150 jp0xtr atom n570 66ghz 2gb 320gb 10 netbook w7s beyaz
41242	samsung n150 jp0xtr intel atom n570 66ghz 2gb 320gb 10 win7starter
169120	samsung n150jp0xtr n570 2g 320gb 10 w7s beyaz
168088	samsung n150 jp0xtr
243325	samsung np n150 jp0xtr beyaz atom n570 2gb 320gb paylaşımlı vga gma3150 10 win starter
2351726	samsung np n150 jp0xtr nc110 intel atom n570 320 win7s 10 beyaz
1326311	samsung n150 jp0xtr netbook n570 2gb 320gb 10 beyaz
2398358	samsung intel atom n570 66ghz 2gb 320gb 10 beyaz netbook n150 jp0xtr
2451624	np n150 jp0xtr samsung nc110 intel atom n570 320 win7s 10 beyaz
2504542	np n150 jp0xtr samsung nc110 intel atom n570 320 win7s 10 beyaz
2421713	samsung np n150 jp0xtr beyaz atom n570 66ghz 2gb 320gb 10 win starter
2405496	samsung np n150 jp0xtr white
12037	samsung n150 jp0xtr atom n570 66ghz 2gb 320gb 10 windows starter beyaz netbook

Çizelge 3.8 ve Çizelge 3.9 incelendiğinde, fazlalık ve hataları kelimelerin hala ürün bilgilerinde bulunduğu ve oldukça önemli bazı ürün bilgilerin eksik olduğu gözlenecektir. Bu aşamadan sonra ise tez kapsamında geliştirilmiş olan özel gürültü elemesi yapan ve özellik vektörlerini oluşturan algoritma ile veriler işlenir. Veriler işlendikten sonra Çizelge 3.8 Çizelge 3.10'a ve Çizelge 3.9'de Çizelge 3.11'a dönüşür.

Çizelge 3.10. Manual sınıflandırılmış Canon renkli kartuş ürün kümesinin gürültü elemesi yapan ve özellik vektörlerini çıkartan algoritma ile 60 eşik değeri ve 10 döngü sayısı kullanılarak özellik vektörleri oluşturulduktan sonraki hali

Ürün ID	**Ürün İsmi**
34632	canon 41 cl kartuş renkli 3010
1417152	canon 41 cl kartuş renkli 3010
131538	canon 41 cl kartuş renkli 3010
15088	canon 41 cl kartuş renkli 3010
2521523	canon 41 cl kartuş renkli 3010
424323	canon 41 cl kartuş renkli 3010
188038	canon 41 cl kartuş renkli 3010
2388138	canon 41 cl kartuş renkli 3010
477899	canon 41 cl kartuş renkli 3010
1505334	canon 41 cl kartuş renkli 3010
2401575	canon 41 cl kartuş renkli 3010
2436897	canon 41 cl kartuş renkli 3010
2406650	canon 3010 41 cl kartuş

Çizelge 3.11. Manual sınıflandırılmış Samsung netbook ürün kümesinin gürültü elemesi yapan ve özellik vektörlerini çıkartan algoritma ile 60 eşik değeri ve 10 döngü sayısı kullanılarak özellik vektörleri oluşturulduktan sonraki hali

Ürün ID	**Ürün İsmi**
9065	10 n570 samsung 320gb jp0xtr n150 white
4907	10 n570 samsung 320gb jp0xtr n150 beyaz
4875	10 n570 samsung 320gb jp0xtr n150 beyaz
41242	10 n570 samsung 320gb jp0xtr n150 atom
169120	10 n570 samsung 320gb white beyaz 2g
168088	10 n570 samsung 320gb jp0xtr n150 white
243325	10 n570 samsung 320gb jp0xtr n150 beyaz
2351726	10 jp0xtr n150 n570 samsung atom beyaz
1326311	10 n570 samsung 320gb jp0xtr n150 white
2398358	10 n570 samsung 320gb jp0xtr n150 beyaz
2451624	10 jp0xtr n150 n570 samsung atom beyaz
2504542	10 jp0xtr n150 n570 samsung atom beyaz
2421713	10 n570 samsung 320gb jp0xtr n150 beyaz
2405496	10 n570 samsung jp0xtr n150 white 2g
12037	10 n570 samsung 320gb jp0xtr n150 beyaz

Çizelge 3.10 ve Çizelge 3.11'de gözüktüğü üzere ürün bilgileri oldukça daha tanımlayıcı ve normalleşmiş bir hale gelmiştir. Gereksiz kelimeler bilgilerden atılmış ve eksik kelimeler eklenmiştir. Bu hale gelen ürün bilgilerinin kümeleme algoritmaları ile kümelenmesi oldukça daha yüksek performans vermektedir.

3.7. KÜMELEME ALGORİTMASI İLE ÜRÜNLERİN GRUPLANMASI

Ürün gruplarının eğitilmesinin ardından oluşan yeni ürün bilgileri oldukça isabetli hale gelmiştir ve özellik vektörü olarak kullanılmaya hazırdır. Bu bilgiler hiçbir gelişmiş kümeleme algoritmasına tabi tutulmadan sadece %100 aynılarını aynı kümeye al şeklinde basit bir mantık kullanılarak kümelenmesi durumunda bile oldukça başarılı sonuçlar elde etmektedir. Tez kapsamında geliştirilen algoritmaların en güçlüsü ürün gruplarının eğitilerek özellik vektörlerini oluşturan algoritma olmuştur.

Ürünleri gruplamak için ilk olarak ürünlerin yeni bilgileri kelime tabanlı olarak listeler haline dönüştürülür ve özellik vektörleri elde edilir. Daha sonra 2 aşamadan oluşan kümeleme işlemine geçilir. İlk aşamada hiçbir ürün hiçbir kümeye girmemiştir ve ürünler kümelere konulur. İkinci aşamada ise ilk aşamada oluşan kümeler birleştirilerek daha büyük kümeler oluşturulur.

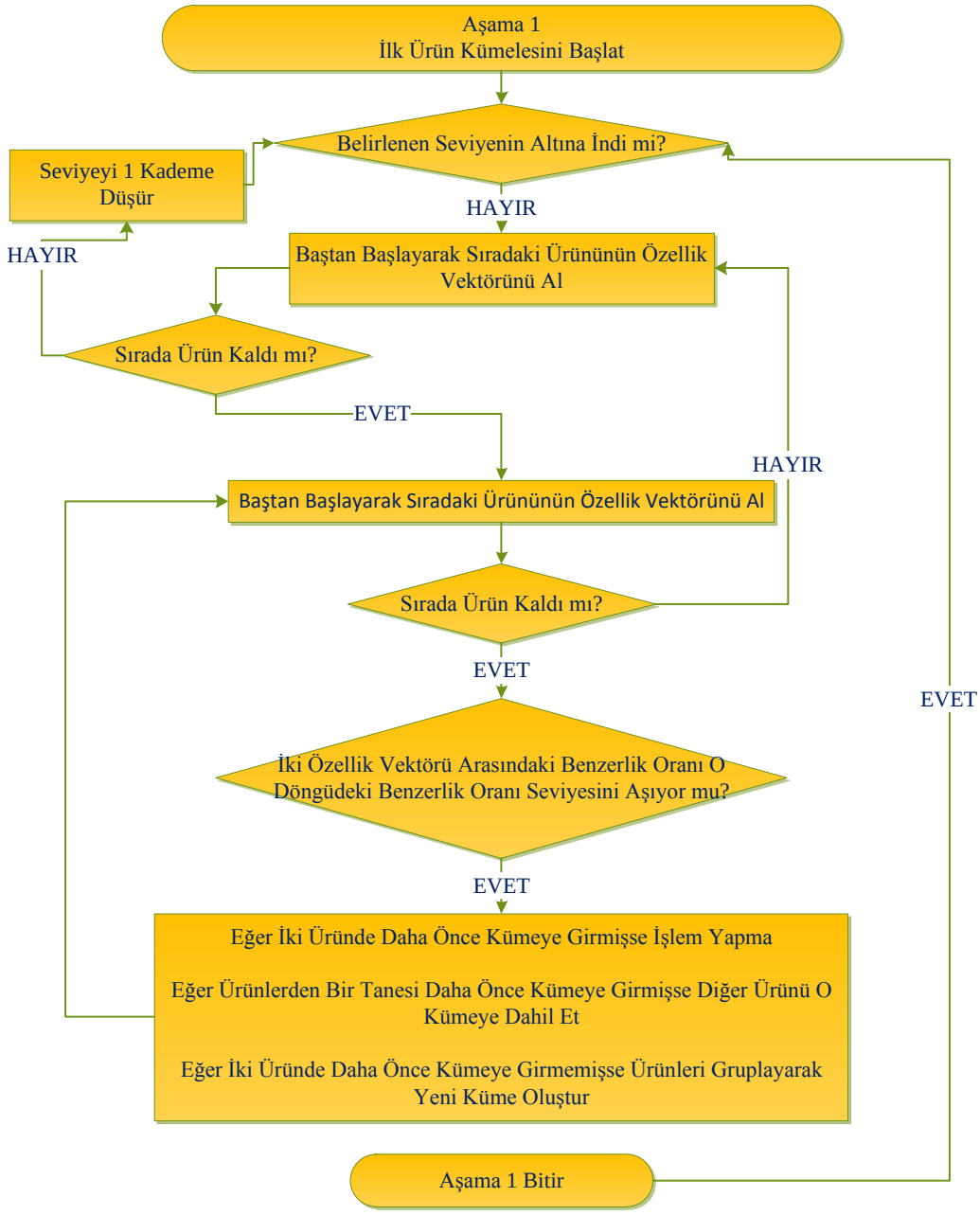
3.7.1. Aşama 1 Dereceli Azalma Tabanlı Kümeleme

İlk aşamada ürünler benzerlik oranlarına göre kümelere eklenirler. Benzerlik oranı %100 den başlayarak belirlenen seviyeye azalana kadar döngü gerçekleşir ve altına düştüğünde daha fazla kümeleme işlemi yapılmaz böylece aşama 1 tamamlanır.

Kümeleme işlemi ürünlerin özellik vektörlerinin benzerlik oranlarının karşılaştırılması ile yapılır. Benzerlik oranın hesaplanması için içerdikleri aynı kelime sayısı kullanılmıştır. Benzerlik oranının hesaplanma formülü daha önce eşitlik (3.1)'de gösterilmiştir. Karşılaştırma işlemi şöyle gerçekleşir. İlk döngüde tüm ürünler diğer tüm ürünler karşılaştırılır ve %100 aynı özellik vektörüne sahip ürünler grup oluştururlar. Bu grup belirli bir hash ile saklanır ve karşılaştırma hızı diğer döngüler için arttırılır. Böylece ilk döngüde %100 aynı özellik vektörüne sahip ürünler aynı kümeye girmiş olurlar. Birinci döngüde dâhil olmak üzere iç döngülerin

her seferinde sadece herhangi bir gruba girememiş ürünler gruba atanmak için taranır. Aşama birdeki amaç hiçbir gruba giremeyen ürünleri bir gruba atamaktır. İkinci döngüde ise %100 oranı bir alt kademeye çekilir. Bunun için evrensel ürün listesinin özellik vektörlerinin ortalama kelime boyutu kullanılır. Örnek olarak eğer evrensel ürün listesinin özellik vektörlerinin ortalama boyutu eğer 7 ise bir sonraki döngüde 6 ürünün aynı olması aranılacaktır. Bunun için de yeni benzerlik oranı 6/7 yani %85 olacaktır. İkinci ve daha sonraki döngülerde gruba girememiş ürünler tüm ürünler ile karşılaştırılırlar. İki ürünün karşılaştırması eğer o döngüdeki benzerlik oranını aşıyorsa, iki üründe gruba girip girmeme durumlarına bakılır. Eğer ürünlerden bir tanesi daha önce bir gruba girmişse diğer üründe o gruba dâhil edilir. Eğer iki üründe daha önce bir gruba girememişse ürünler yeni bir küme oluştururlar.

Her döngü sonunda benzerlik oranı bir alt kademeye çekilir ve oran belirlenen seviyenin altına inince bir daha karşılaştırma yapılmaz. Tez kapsamında algoritmanın geliştirilmesi sürecinde ilk olarak bu oran %50 olarak belirlenmiştir. Bu oranın altına inince boşta kalan ürünler herhangi bir kümeye dâhil edilmek için daha fazla taranmaz ve tek başlarına küme oluşturdukları varsayılır. Aşama 1'in akış şeması şekil Şekil 3.6'da gösterilmiştir.



Şekil 3.6. Aşama 1 dereceli azalma tabanlı kümeleme akış şeması

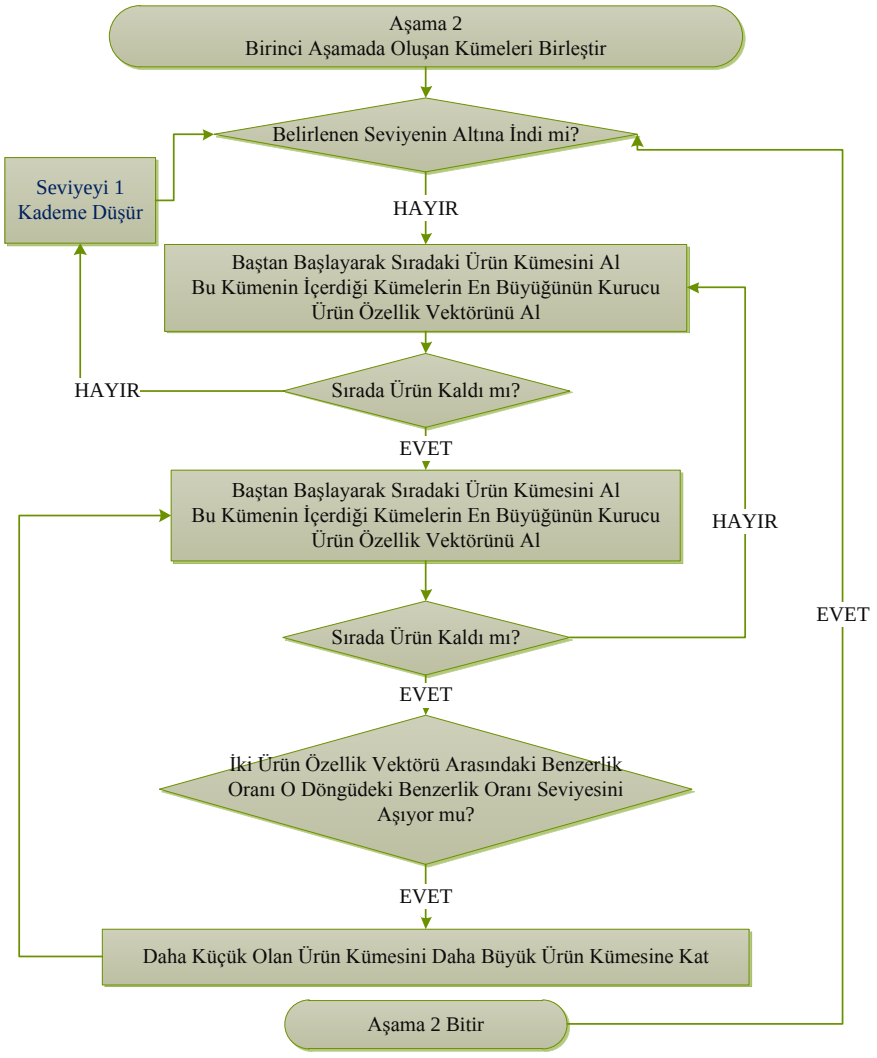
3.7.2. Aşama 2 Dereceli Birleştirme Tabanlı Kümeleme

İkinci aşamada tarama ürünler arasında değil ürün kümeleri arasında yapılır. Karşılaştırma yapılırken o kümeyi oluşturan ürünlerin özellik vektörlerinin ortalaması yerine, o kümeyi ilk oluşturan ürünün özellik vektörü kullanılmıştır.

İkinci aşamada birinci aşamada olduğu gibi %100'den değil bu sefer ikinci kademeden başlanır. Sebebi ise zaten %100 aynı olanların birinci aşamada aynı

gruba girmiş olmasıdır. İkinci aşamada da ürün vektörleri arasında karşılaştırma yapılır fakat bu sefer gruplar bir biri ile birleştirilmektedir.

İkinci aşamada gruplar birleştirildikçe daha büyük kümeler oluşmaktadır. İki grup karşılaştırılırken grubun içindeki hangi ürünün vektörünün karşılaştırma için kullanılacağı ise, o grup içindeki gruplardan hangisi birinci aşamada en büyük grubu oluşturmuşsa, o en büyük grubu kuran ürünün vektörünün seçilmesi ile belirlenir. Birinci aşamada olduğu gibi gruplar arası karşılaştırma belirlenen benzerlik seviyesinin altına düşene kadar devam eder. İşlemler sonucunda oluşan ürün grupları, farklı e-ticaret sitelerinde satılan aynı ürünlerin gruplanmasını oluşturur. Aşama 2'nin akış şeması Şekil 3.7'de gösterilmiştir.



Şekil 3.7. Aşama 2 dereceli birleştirme tabanlı kümeleme akış şeması

3.8. TEST SÜRECİ

Test süreci, sadece belirli bir aşamadan sonra değil, tüm tez boyunca gerçekleşmiştir. Algoritmalar geliştirilirken sürekli olarak parametre ve yöntem denemeleri yapılmış ve daha iyi sonuçlar elde edilmeye çalışılmıştır. Fakat en son aşamadaki kümeleme algoritmasının ardından farklı e-ticaret sitelerinde satılan aynı ürünlerin gruplanmasının başarısı, tez kapsamında geliştirilen başarı ölçme algoritması tamamlanana kadar tespit edilememiştir. Başarının hesaplandığı algoritmanın gelişimine kadar yorum yöntemi ve gözlem ile iyileştirmeler yapılmıştır.

Tez kapsamında üzerinde çalışılan sistemin benzerinin literatürde çok az bulunması nedeniyle, geliştirilen algoritmanın başarısını mantıklı bir şekilde tespit edebilecek yeni bir algoritma geliştirilmesi gerekmiştir. Başarıyı ölçmek için kullanılacak algoritmanın aşağıdaki iki durumda da çalışması gerekmektedir.

- Çok sayıda farklı ürün grubu birleştirilerek çok büyük bir grup oluştuğunda
- Ürünler yeterince gruplanmayıp çok sayıda grup oluştuğunda

İki durum aslında bir birinin zıttı olmaktadır. Eğer algoritma sadece ürünün gruplanması gereken diğer ürünler ile aynı gruba girip girmediğine bakarsa, birinci seçenekteki çok sayıda grubun birleştirilip çok büyük grup oluşması durumunda başarı çok yüksek olacaktır. Örnek olarak tüm ürünler sadece tek bir gruba girerse başarı oranı %100 çıkacaktır. Hâlbuki bu çok büyük bir başarısızlık olacaktır. İşte bunu tespit etmeye kaçırma tespiti denilmektedir.

İkinci durum ise aynı gruba girmesi gereken ürünler çok sayıda grup oluştururursa başarı nasıl belirlenecektir sorusunu gündeme getirmektedir. Sadece grup sayısına bakılması veya sadece doğru ürünler ile gruplanıp gruplanmadığına bakılması, gene başarısız bir sonuç üretecektir. Bu duruma ise yanlış alarm denilmektedir. Bu 2 durumu da göz önüne alan ve bu problemlere çözüm üreten bir algoritma geliştirilmiştir.

3.8.1. Kümeleme Algoritmasının Başarısını Ölçen Algoritma

Tez kapsamında geliştirilen kümeleme algoritması, aynı ürünleri gruplamaya yöneliktir. Bu tarz bir algoritmanın başarısını ölçen bir algoritmaya

literatürde rastlanmamıştır. Bu bakımından geliştirilen yeni başarı ölçme algoritması ilklerden olma özelliğine sahiptir.

Kümeleme algoritmasının ürettiği sınıfların listesi sol tarafta ve bu ürünlerin gerçek sınıflarının listesi sağ tarafta olacak şekilde düşünülürse, geliştirilen performans algoritması soldan sağa ve sağdan sola haritalama yaparak performans sonucunu bulmaktadır.

i. Kaçırma Tespiti (KT): Gerçekte aynı ürün olduğu halde algoritmanın farklı ürün olarak sınıfladığı ürün çiftlerinin sayısı. Kaçırma Tespiti oranı O_{KT}, gerçekte aynı olan toplam ürün çiftlerinin sayısı N_{KT} ile temsil edilirse formül Eşitlik (3.7)'deki gibi olur.

$$O_{KT} = \frac{KT}{N_{KT}} \; x \; \%100 \tag{3.7}$$

ii. Yanlış Alarm (YA): Gerçekte farklı ürün olduğu halde algoritmanın aynı ürün olarak sınıfladığı ürün çiftlerinin sayısı. Yanlış Alarm oranı O_{YA}, gerçekte farklı olan toplam ürün çiftlerinin sayısı N_{YA} ile temsil edilirse formül Eşitlik (3.8)'deki gibi olur.

$$O_{YA} = \frac{YA}{N_{YA}} \; x \; \%100 \tag{3.8}$$

iii. Toplam Hata (TH): Kaçırma Tespiti veya Yanlış Alarm olarak tespit edilen toplam ürün çiftlerinin sayısı. Buna göre toplam hata oranı formülü Eşitlik (3.9)'da verilmiştir.

$$O_{TH} = \frac{KT + YA}{N_{KT} + N_{YA}} \; x \; \%100 \tag{3.9}$$

3.8.1.1. Kaçırma tespiti

Kaçırma tespiti hesaplanırken kümeleme algoritmasının oluşturduğu sınıflar solda ve orijinal sınıflar sağda olacak şekilde, soldan sağa doğru haritalama yapılır. Sağ taraftaki her bir sınıf için sol tarafta kaç adet sınıf oluştuğuna bakılır ve oluşan

sayıya göre KT oranı hesaplanır. Örnek olarak sağ tarafta 8 adet sınıf 1 olsun. Sol tarafta bu 8 ürün ise kümeleme algoritması tarafından 2 adet sınıf 2, 5 adet sınıf 6 ve 1 adet sınıf 3 olarak gruplansın Çizelge 3.12'de gözüktüğü gibi.

Çizelge 3.12. Kaçırma tespiti örnek

Kümeleme Algoritması Sonucu Ürünlerin Sınıflanması	Bu Ürünlerin Orijinal Sınıfları
3	1
2	1
6	1
2	1
6	1
6	1
6	1
6	1

Bu sonuçlara göre sol tarafta tek 1 sınıf oluşması gerekirken 1 ürünlü, 2 ürünlü ve 5 ürünlü olmak üzere 3 sınıf oluşmuştur. Hata ise şu şekilde hesaplanır. Sağ taraftaki o sınıf için ürün sayısının 2'li kombinasyonundan, sol taraftaki sınıfların içerdikleri eleman sayılarının 2'li kombinasyonlarının toplamı çıkartılır ve sağ taraftaki eleman sayısının 2'li kombinasyonuna bölünür. Bu örnek için hata Eşitlik (3.10)'da gözüktüğü şekilde hesaplanır.

$$\frac{\binom{8}{2} - \left(\binom{1}{2} + \binom{2}{2} + \binom{5}{2}\right)}{\binom{8}{2}} = \frac{17}{28} \tag{3.10}$$

Her bir orijinal sınıf için KT oranı hesaplanır ve çıkan değerler evrensel pay ve paydaya eklenir.

3.8.1.2. Yanlış alarm

Yanlış alarm, kaçırma tespiti ile aynı mantıkta çalışmaktadır. Bu sefer sağ tarafta kümeleme algoritması tarafından oluşturulan sınıflar sıra ile sıralanır ve sol tarafta ise bu ürünlerin orijinal sınıfları yazılır. Bu şekilde yanlış alarm hesaplaması yapılır. Örnek olarak kümeleme algoritması tarafından sınıf 4 diye kümelenmiş 8 ürün olsun ve bunların gerçek sınıfları 2 adet 2, 3 adet 5 ve 3 adet 7 olsun Çizelge 3.13'da gözüktüğü gibi.

Çizelge 3.13. Yanlış alarm örnek

Kümeleme Algoritması Sonucu Ürünlerin Sınıflanması	Bu Ürünlerin Orijinal Sınıfları
4	2
4	2
4	5
4	5
4	5
4	7
4	7
4	7

Bu sefer orijinal sınıflara göre değil, kümeleme algoritması tarafından oluşturulan sınıflara göre sıralama yapılmıştır. Soldan sağa haritalamanın tam tersi. Kaçırma tespitinin hesaplanmasındaki yöntemin aynısı uygulanır ve hata hesaplanır. Bu örnek için sonuç Eşitlik (3.11)'de gözüktüğü gibi çıkacaktır.

$$\frac{\binom{8}{2} - (\binom{2}{2} + \binom{3}{2} + \binom{3}{2})}{\binom{8}{2}} = \frac{21}{28} \tag{3.11}$$

Sol taraftaki tüm sınıflar taranıncaya kadar döngüsel olarak hata hesaplama işlemi devam eder ve elde edilen pay ve paydalar evrensel pay ve payda tutucusuna eklenir.

3.8.1.3. Toplam hata

Bütün ürünler için kaçırma tespiti oranı ve yanlış alarm değerleri hesaplandıktan sonra oluşan evrensel pay ve payda değerleri ile toplam hata hesaplanır. Yukarıda verilen iki örneğin toplam hatası (21+17)/(28+28)=%67 olarak çıkacaktır.

3.9. WEB ARAYÜZÜ

Geliştirilen KAM motorunun insanlar tarafından kullanılabilmesi için ilk etapta deneme sürümü olacak şekilde web ara yüzü geliştirilmiştir. Web ara yüzünün geliştirilmesi için ASP.net ve C# alt yapısı kullanılmıştır.

Web ara yüzü kullanıcıdan aldığı bilgiyi boşluk karakterine göre kelimelere bölerek veri tabanına sorgu yollar ve bu kelimeleri içeren ürünlerin ait oldukları sınıfları kullanıcıya döndürür. Örnek bir ürün arama sorgusu ve dönen sonuçlar Şekil 3.8'de gösterilmiştir.

Şekil 3.8. Web ara yüzü örnek ürün arama sorgusu ve dönen sonuçlar

Veri tabanında hızlı bir şekilde kelimeyi içeren ürünleri bulabilmek için tam-metin indis sistemi kullanılmıştır. Bu sayede kullanıcı tarafından yollanan sorgudaki kelimeleri bulmak, normal arama yapmaya göre çok daha hızlı olmaktadır. Aranan kelimeleri ürünlerin ait oldukları sınıf ID'leri elde edilir ve bu sınıfa giren ürünlerin sayıları ve fiyat aralık bilgileri çekilir. Şekil 3.8'de bu bilgiler gözükmektedir. Daha sonra kullanıcı ilgilendiği ürünün hangi e-ticaret sitelerinde satıldığını görmek için Ürünleri Göster tuşuna basarak ürünlerin satıldığı yerler hakkında detaylı bilgi sayfasına ulaşır Şekil 3.9'da gözüktüğü gibi.

Şekil 3.9. Web ara yüzü ürünlerin satıldığı siteler hakkında detaylı bilgi sayfası

Bu sayfada ürünün satıldığı yere ulaşmak için kullanıcı Ürüne Git tuşuna basarak o ürünün satıldığı e-ticaret sayfasına ulaşır.

4. BULGULAR VE TARTIŞMA

Tez kapsamında geliştirilen tüm algoritmalar gözetimsiz öğrenme yöntemi ile çalışmaktadır ve bu yüzden %100 olarak ölçeklenebilir durumdadırlar. Tek yapılması gereken taranacak sitelerden bilgileri çekecek fonksiyonların yazılmasıdır. Bu sayede 1000'lerce site herhangi bir kullanıcı girdisi olmadan taranabilmekte ve insanların hizmetine sunulabilmektedir. Fakat test yapabilmek için manuel olarak seçilmiş ürün seti gerekmektedir. Bunun için yazılan bir fonksiyon ile 100 adet rastgele ürün seçilmiştir. Ürünler EXA bilgisayarın [73] listesinden seçilmiştir. Bunun sebebi ise EXA bilgisayarın bilgisayar parçaları üzerine yoğunlaşmış olmasıdır. Tez kapsamında belirlenen 20 adet e-ticaret sitesinin ortak özelliği, hepsinin bilgisayar ürünü satıyor olmasıdır.

Seçilen ürünlerin tamamı farklı ürünlerdir. Bunun sebebi ise bir e-ticaret sitesinin aynı ürünü 2 defa listelememesi, farklı linkler ile listelese bile tasarlanan KAM'nin 2 veya daha fazla nüshası bulunan ürünleri teke indirmesidir. Bu ürünlerin veri tabanında başka hangi sitelerde listelendiği tek tek manuel olarak tespit edilmiştir. Bu sayede ürünlerin başka hangi ürünler ile aynı sınıfa girmesi gerektiği çıkartılmıştır. Seçilen ürünlerin satıldığı diğer sitelerde tespit edildikten sonra toplam test seti 936 ürüne ulaşmıştır.

Manuel olarak elde edilen test seti üzerinde mevcut kümeleme algoritmaları ve tez kapsamında geliştirilen kümeleme algoritmaları test edilmiştir. Özellik vektörü çıkartma algoritmasının performansı da testler sırasında ölçülmüştür. Özellik vektörü çıkarma algoritmasının performansı bu algoritmanın devre dışı bırakıldığı ve etkinleştirildiği durumdaki performans sonuçları ile test edilmiştir. Bu algoritma devre dışı bırakıldığında kelime tabanlı gürültü elemesi ve boyut küçültme ile normalleştirme işlemleri gerçekleşmemiştir. Ayrıca test sürecinde algoritmaları daha fazla zorlamak için kelimeler küçük harfe dönüştürülürken Türkçe karakterler İngilizce karşılığına dönüştürülmemiştir. Örnek olarak "ş" harfi "s" harfine dönüştürülmemiştir. Bulgular ve tartışma kapsamında verilen performans sonuçları bu şekildeki veri seti üzerinde elde edilmiştir.

Gözetimsiz öğrenme algoritmalarında parametrelerin doğruluğu sonuçların başarısını çok yüksek oranda etkilemektedir. Birden çok parametrenin olduğu durumlarda doğru parametrelerin bulunması ise başlı başına bir araştırma alanı

olmaktadır ve tek başına bir tez konusu olabilecek kapsamdadır. Bu tez kapsamında doğru parametrelerin bulunması araştırma konusu olmadığı için yazılımsal olarak test seti üzerinde parametre denemeleri yapılmış ve çok iyi değerler üreten parametreler bulunmuştur. Bulunan parametreler en iyi parametreler olmaktan uzak olmasına rağmen oldukça iyi sonuçlar üreterek tez kapsamında geliştirilen algoritmaların ne kadar etkili olduğunu ortaya koymaktadır.

Standart hiyerarşik kümeleme algoritmaları ve tez kapsamında geliştirilen yeni kümeleme algoritması test edilmiştir. Standart hiyerarşik kümeleme algoritmalarının test edilmesi için Matlab programının sahip olduğu hazır fonksiyonlardan faydalanılmıştır. Bu algoritmalara tez kapsamında geliştirilen özellik vektörlerini oluşturan algoritma ile üretilen ve bu algoritmaya tabi tutulmadan e-ticaret sitelerinden elde edildiği şekliyle üretilen veri setleri verilerek geliştirilen özellik vektörü çıkartma algoritmasının başarısı hesaplanmıştır. Harf bazında gürültü elemesi her iki durumda da gerçekleştirilmiştir. Tez kapsamında geliştirilen kümeleme algoritmasının da başarısı ortaya konulmuştur.

Özellik vektörlerini oluşturan algoritmanın etkin olduğu testlerde 2 adet parametre kullanılmıştır. Bu parametrelerin ilki ilk gruplamadaki eşik değeri, diğer parametre ise grupların gözetimsiz olarak eğitilerek özellik vektörlerinin oluşturulması sürecindeki döngü sayısıdır. Standart algoritmaların testinde ek olarak 2 farklı ilk benzerlik hesaplama metodu kullanılmıştır. Eşitlik (3.1)'deki metot Normal ve eşitlik (3.4)'deki metot ise Evrensel olarak adlandırılmıştır. Bu benzerlik hesaplamalarının yapılış sebebi ise Matlab'daki hazır kümeleme algoritmalarının verilen özellik vektörleri arasındaki benzerlik mesafesini alarak çalışmasıdır.

4.1. HİYERARŞİK KÜMELEME ALGORİTMALARININ TEST EDİLMESİ VE SONUÇLARI

Hiyerarşik kümeleme algoritmalarının test edilmesi için Matlab programının sahip olduğu linkage fonksiyonu kullanılmıştır. Bu fonksiyon belirtilen algoritmayı kullanarak hiyerarşik küme ağacı oluşturmaktadır. Aşağıdaki algoritmalar fonksiyona verilebilmektedir:

- 'single' : en yakın mesafe
- 'complete' : en uzak mesafe

- ‘average’ : ağırlıksız ortalama mesafe (UPGMA)
- ‘weighted’ : ağırlıklı ortalama mesafe (WPGMA)
- ‘centroid’ : ağırlıksız kitlesel mesafenin merkezi (UPGMC)
- ‘median’ : ağırlıklı kitlesel mesafenin merkezi (WPGMC)
- ‘ward’ : içsel kare mesafe

Yukarıda belirtilen algoritmaların hepsi test edilmiştir. Bu algoritmalar sonucu elde edilen küme ağaçları Matlab programının sahip olduğu standart cluster fonksiyonuna verilmiştir. Bu fonksiyon aldığı küme ağacından kümeleri oluşturmaktadır. Fonksiyon küme ağacının haricinde standart 2 parametre daha almaktadır. Tüm testlerde ‘Cutoff” ve ‘1.15’ parametreleri verilmiştir. Cluster fonksiyonun bu parametreler ile tez kapsamında geliştirilen kümeleme için en iyi sonuçları ürettiği tespit edilmiştir.

Özellik vektörü oluşturan algoritma eşik değeri ve döngü sayısı parametrelerini kullanmaktadır. Bu yüzden çizelgelerde aktif değil iken eşik değeri ve döngü sayısına göre sonuçlar değişmemektedir.

Çizelge 4.1’de en yakın mesafe algoritması ile hiyerarşik kümelemenin sonuçları gösterilmiştir. Tez kapsamında geliştirilen özellik vektörlerinin çıkartılması algoritması doğru parametreler ile kullanılınca %66,20’lik bir performans kazancı sağlamıştır.

Çizelge 4.1. Hiyerarşik kümeleme en yakın mesafe algoritması performans sonuçları

En Yakın Mesafe	Aktif Değil Hata %	Aktif Hata %	Performans Kazancı %
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Normal	26,06	18,34	29,62
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Evrensel	38,85	29,05	25,22
Eşik Değeri: 58 Döngü Sayısı: 12 Metot: Normal	26,06	16,08	38,29
Eşik Değeri: 68 Döngü Sayısı: 12 Metot: Evrensel	38,85	**13,13**	66,20

Çizelge 4.2'de en uzak mesafe algoritması ile hiyerarşik kümelemenin sonuçları gösterilmiştir. Bu yöntemde denenen tüm kümeleme algoritmaları arasındaki en iyi hata değeri olan %8,68 değerine ulaşılmıştır.

Çizelge 4.2. Hiyerarşik kümeleme en uzak mesafe algoritması performans sonuçları

En Uzak Mesafe	Aktif Değil Hata %	Aktif Hata %	Performans Kazancı %
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Normal	37,31	19,87	46,74
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Evrensel	25,58	34,26	-33,93
Eşik Değeri: 67 Döngü Sayısı: 48 Metot: Normal	37,31	13,94	62,63
Eşik Değeri: 84 Döngü Sayısı: 33 Metot: Evrensel	25,58	**8,68**	66,06

Çizelge 4.3'de ağırlıksız ortalama mesafe algoritması ile hiyerarşik kümelemenin sonuçları gösterilmiştir. Yazılımsal parametre denemeleri sonucu elde edilen en iyi parametreler ile %52'lik bir performans artışı yakalanabilmiştir.

Çizelge 4.3. Hiyerarşik kümeleme ağırlıksız ortalama mesafe algoritması performans sonuçları

Ağırlıksız Ortalama Mesafe	Aktif Değil Hata %	Aktif Hata %	Performans Kazancı %
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Normal	31,23	20,16	35,44
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Evrensel	36,18	44,55	-42,65
Eşik Değeri: 60 Döngü Sayısı: 66 Metot: Normal	31,23	15,75	49,56
Eşik Değeri: 76 Döngü Sayısı: 50 Metot: Evrensel	36,18	**14,97**	52,06

Çizelge 4.4'de ağırlıksız ortalama mesafe algoritması ile hiyerarşik kümelemenin sonuçları gösterilmiştir. Yazılımsal parametre denemeleri sonucu elde edilen en iyi parametreler ile %53'lük bir performans artışı yakalanabilmiştir.

Çizelge 4.4. Hiyerarşik kümeleme ağırlıklı ortalama mesafe algoritması performans sonuçları

Ağırlıklı Ortalama Mesafe	Aktif Değil Hata %	Aktif Hata %	Performans Kazancı %
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Normal	37,39	19,93	46,69
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Evrensel	26,72	25,25	5,5
Eşik Değeri: 64 Döngü Sayısı: 14 Metot: Normal	37,39	17,50	53,19
Eşik Değeri: 80 Döngü Sayısı: 18 Metot: Evrensel	26,72	**16,09**	39,78

Çizelge 4.5'de ağırlıksız kitlesel mesafenin merkezi algoritması ile hiyerarşik kümelemenin sonuçları gösterilmiştir. Yapılan denemelerde yüzde tabanlı olarak en yüksek performans kazancı %75'lik bir değer ile bu algoritmada yakalanmıştır.

Çizelge 4.5. Hiyerarşik kümeleme ağırlıksız kitlesel mesafenin merkezi algoritması performans sonuçları

Ağırlıksız Kitlesel Mesafenin Merkezi	Aktif Değil Hata %	Aktif Hata %	Performans Kazancı %
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Normal	33,78	18,90	44,04
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Evrensel	59,99	29,76	50,39
Eşik Değeri: 64 Döngü Sayısı: 90 Metot: Normal	33,78	**14,50**	57,07
Eşik Değeri: 68 Döngü Sayısı: 98 Metot: Evrensel	59,99	14,54	75,76

Çizelge 4.6'da ağırlıklı kitlesel mesafenin merkezi algoritması ile hiyerarşik kümelemenin sonuçları gösterilmiştir. Tez kapsamında geliştirilen özellik çıkartma ve verileri eğitme algoritması yaklaşık %72'lik bir başarı göstermiştir.

Çizelge 4.6. Hiyerarşik kümeleme ağırlıklı kitlesel mesafenin merkezi algoritması performans sonuçları

Ağırlıklı Kitlesel Mesafenin Merkezi	Aktif Değil Hata %	Aktif Hata %	Performans Kazancı %
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Normal	37,76	19,80	47,56
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Evrensel	41,41	24,15	41,68
Eşik Değeri: 68 Döngü Sayısı: 46 Metot: Normal	37,76	14,53	61,52
Eşik Değeri: 72 Döngü Sayısı: 34 Metot: Evrensel	41,41	**11,65**	71,86

Çizelge 4.7'de içsel kare mesafe algoritması ile hiyerarşik kümelemenin sonuçları gösterilmiştir. Tez kapsamında geliştirilen özellik çıkartma ve verileri eğitme algoritması yaklaşık %61'lik bir başarı göstermiştir.

Çizelge 4.7. Hiyerarşik kümeleme içsel kare mesafe algoritması performans sonuçları

İçsel Kare Mesafe	Aktif Değil Hata %	Aktif Hata %	Performans Kazancı %
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Normal	34,10	17,34	49,14
Eşik Değeri: 60 Döngü Sayısı: 20 Metot: Evrensel	30,67	33,60	-9,55
Eşik Değeri: 70 Döngü Sayısı: 22 Metot: Normal	34,10	12,97	61,96
Eşik Değeri: 82 Döngü Sayısı: 16 Metot: Evrensel	30,67	**12,05**	60,71

Standart hiyerarşik kümeleme algoritmaları ile yapılan testlerin sonucu olarak tamamıyla gözetimsiz öğrenme yöntemleri kullanılarak hata oranı %10'un altına inebilmiştir. Tez kapsamında geliştirilen verileri gözetimsiz olarak eğiterek özellik vektörlerini oluşturan algoritma çok büyük oranda performans artışı getirmektedir. Testler ayrıca gözetimsiz öğrenme yöntemindeki büyük problemlerden biri olan parametrelerin önemini de ortaya koymaktadır. Doğru parametrelerin seçilmesi ile çok iyi derecede performans elde edilebilmektedir. Testlerdeki parametreler sınırlı tez süresi boyunca bulunan parametrelerdir ve tez konusunda en iyi parametreleri elde etmek olmadığı için bu sonuçlardan çok daha iyi sonuçlara ulaşmak mümkündür.

4.2. TEZ KAPSAMINDA GELİŞTİRİLEN ÖZEL KÜMELEME ALGORİTMASININ TEST EDİLMESİ VE SONUÇLARI

Tez kapsamında geliştirilen algoritma 4 farklı parametre almaktadır. Bunlardan ilk 2 tanesi özellik vektörlerini oluşturan algoritmanın aldığı eğitim için gereken döngü sayısı ve ilk gruplama için gereken eşik değeridir. Diğer iki tanesi ise geliştirilen özel kümeleme algoritmasının ilk aşama ve ikinci aşamada küme oluşturmak için aldığı eşik değerleridir.

Çizelge 4.8'de tez kapsamında geliştirilen özel kümeleme algoritmasının performans sonuçları verilmiştir. Eşik değeri 1 özellik vektörlerini oluşturan algoritmanın aldığı eşik değerini ifade etmektedir. Eşik değeri 2 tez kapsamında geliştirilen kümeleme algoritmasının ilk kümeleme işlemi için aldığı eşik değerini ifade ederken eşik değeri 3 ise ikinci kümeleme işlemi için gereken eşik değerini ifade etmektedir. Geliştirilen özel kümeleme algoritması mevcut standart kümeleme algoritmalarına göre daha düşük performans vermiş olmasına rağmen gözetimsiz öğrenme yöntemi ile çalışan bir algoritma için iyi sayılabilecek bir değer olan %16 değeri elde edilebilmiştir. Algoritmanın geliştirilmesi ile çok daha iyi değerler elde edilmesi mümkündür.

Çizelge 4.8. Tez kapsamında geliştirilen özel kümeleme algoritmasının performans sonuçları

	Aktif Değil Hata %	Aktif Hata %	Performans Kazancı %
Eşik Değeri 1: 60 Döngü Sayısı: 20 Eşik Değeri 2: 50 Eşik Değeri 3: 50	27,77	23,61	14,98
Eşik Değeri 1: 60 Döngü Sayısı: 100 Eşik Değeri 2: 50 Eşik Değeri 3: 50	27,77	23,26	16,24
Eşik Değeri 1: 60 Döngü Sayısı: 250 Eşik Değeri 2: 50 Eşik Değeri 3: 50	27,77	25,45	8,35
Eşik Değeri 1: 60 Döngü Sayısı: 500 Eşik Değeri 2: 50 Eşik Değeri 3: 50	27,77	25,81	7,05
Eşik Değeri 1: 60 Döngü Sayısı: 1000 Eşik Değeri 2: 50 Eşik Değeri 3: 50	27,77	26,98	2,84
Eşik Değeri 1: 68 Döngü Sayısı: 59 Eşik Değeri 2: 55 Eşik Değeri 3: 25	18,86	**16,08**	14,74

5. SONUÇLAR VE ÖNERİLER

Tez kapsamından geliştirilen yazılımlar, algoritmalar ve gerçekleştirilen eylemler aşağıdaki şekilde listelenebilir:

- Elde edilen bilgilerin saklanacağı veri tabanı sistemi
- E-ticaret sitelerinin sadece ilk adreslerini aldıktan sonra, sitelerin sahip olduğu bütün sayfaları tarayarak, sayfaların ham içeriğinden ürün bilgilerini çıkartacak ve veri tabanında saklayacak tarayıcı bot yazılımı
- Veri tabanından bilgileri okuyarak gürültü elemesi yapacak algoritmalar
- Gürültü elemesi yapıldıktan sonra kümeleme algoritmaları için uygun şekilde özellik vektörleri oluşturacak algoritma
- Oluşturulan özellik vektörlerini kullanarak aynı ürünlerin kümelenmesini yapacak yeni bir kümeleme algoritması
- Oluşturulan özellik vektörlerinin standart kümeleme algoritmaları ile test edilerek performans analizleri
- Kullanıcıdan aldığı bilgi doğrultusunda kullanıcının istediği ürünün satıldığı web sitelerini fiyatları ile beraber listeleyecek web arama motoru

Teze ilk başlanıldığında tamamıyla gözetimsiz öğrenme yöntemleri ile ölçeklenebilir kümelemenin başarısı %70 olarak hedeflenmişti. Fakat geliştirilen gürültü eleme ve özellik vektörlerini oluşturan algoritma ile %92 gibi oldukça iyi bir başarı oranı yakalanabildi. %92 başarı oranı tamamıyla gözetimsiz öğrenme yöntemi ile çalışan sistemler için oldukça iyi bir orandır. Ayrıca bu veriler tamamıyla gerçek hayat verisi üzerinde elde edilmiştir. Gerçek hayat verisi genel olarak literatürde kullanılan sanal verilerden çok daha gürültülü ve üzerinde çalışması zor olmaktadır.

Tez kapsamında özel olarak geliştirilen kümeleme algoritması henüz yeterince olgunlaştırılamamıştır. Özel olarak geliştirilen kümeleme algoritmasının üzerinde çalışılarak çok daha iyi sonuçlar elde edilmesi mümkündür. Kümeleme için standart algoritmaların kullanılması durumunda ise en uygun parametre değerlerinin belirlenebilmesi için ek çalışmalar yapılması gerekmektedir.

KAYNAKLAR

[1] Yönetici, S., "e-ticaret 2011′de %50 artışla 22,9 milyar TL, 2012 hedefi 34 milyar TL", Sanalmimarlar, http://blog.sanalmimarlar.com/2012/02/e-ticaret-2011-sonu-2012-hedef/ (14.02.2012)

[2] Yang, J., Choi, J., Kim, J. and Ham, H., "A More Scalable Comparison Shopping Agent", Engineering of Intelligent Systems, 766-772, (2000).

[3] Doorenbos, R., Etzioni, O. and Weld, D., "A Scalable Comparison-Shopping Agent For The World Wide Web", In Proceeding of the First International Conference on Autonomous Agents (Agents-97), Marina del Rey, CA, 39-48, (1997).

[4] Zhang, L., Zhu, M., Huang, W., "A Framework for an Ontology-based E-commerce Product Information Retrieval System", Journal Of Computers 4(6), 436-443, (2009).

[5] Lee, I., Lee, S., Lee, T., Lee, S., Kim, D., Chun, J., Lee, H. and Shim, J., "Practical issues for building a product ontology system", in Proceedings of the 2005 International Workshop on Data Engineering Issues in E-Commerce (DEEC'05), IEEE Computer Society, (2005).

[6] Lee, T., Chun, J., Shim, J., Lee, S.-g., "An Ontology-Based Product Recommender System for B2B Market places", International Journal of Electronic Commerce 11, 125-154, (2006).

[7] Hans Friedrich, W. & Fabian, S., "Information Structuring and Product Classification", TSST 2006 Beijing, (2006).

[8] Ding, Y., Korotkiy, M., Omelayenko, B., Kartseva, V., Zykov, V., Klein, M., Schulten, E. and Fensel, D., "GoldenBullet: Automated Classification of Product Data in Ecommerce", In Proceedings of Business Information Systems Conference, (BIS 2002), Poznan, Poland, (April 2002).

[9] Wolin, B., "Automatic Classification in Product Catalogs", Proceedings of the 25th Annual Interna-tional ACM SIGIR Conference on Research and Development in Information Retrieval, (2002).

[10] Brin, S., Page, L., "Anatomy of a Large-Scale Hypertextual Web Search Engine", Proc. 7th International World Wide Web Conference, (1998).

[11] Hanumantha Rao, G., Narender, G., Srinivasa Rao, B. and Srilatha, M., "Web Search Engine", International Journal of Scientific & Engineering Research Volume 2, Issue 12, (December-2011).

[12] Alexa the Web Information Company, http://www.alexa.com, (2012).

[13] Google Scholar, http://scholar.google.com, (2012).

[14] Yahoo!, http://www.yahoo.com, (2012).

[15] 百度一下，你就知道，http://www.baidu.com, (2012).

[16] Bing, http://www.bing.com, (2012).

[17] STEELE, R. "Techniques for specialized search engines", In Proc. Internet Computing 2001, Las Vegas, (2001).

[18] Thelwall, M. (2001a), "A web crawler design for data mining", Journal of Information Science 27(5), 319-325, (2001).

[19] Heydon, A. and Najork, M. A., "Mercator: A scalable, extensible web crawler", World Wide Web, 2(4):219-229, (Dec. 1999).

[20] Gordon S., L. and Michael J. A., B., "Data Mining Techniques: For Marketing, Sales, And Customer Relationship Management, 3nd ed.", Wiley Publishing Inc., Indianapolis, Indiana, 2 s., (2011).

[21] Jianfeng, G., Galen, A., Mark, J. and Kristina, T., "A comparative study of parameter estimation methods for statistical natural language processing", In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL'07), 824-831, (2007).

[22] Guyon, I. and Elisseeff, A., "An introduction to variable and feature selection", J. Mach. Learn. Res. 3, 1157-1182, (2003).

[23] Yang, Y., Pedersen J., "A Comparative Study on Feature Selection in Text Categorization", In: Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, 412-420, (1997).

[24] Nomoto, T. and Matsumoto, Y., "A new approach to unsupervised text summarization", In Proc. of the 24th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval, (2001).

[25] M. Surdeanu, J. Turmo and A. Ageno, "A Hybrid Unsupervised Approach for Document Clustering", In Proc. KDD'05, 685-690, Chicago, Illinois, USA, (2005).

[26] Kondadadi, R. And Kozma, R., "A modified fuzzy ART for soft document clustering", Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN '02, vol. 3, 2545-2549, (2002).

[27] Rooney, N., Patterson, D., Galushka, M. and Dobrynin, V., "A scaleable document clustering approach for large document corpora", Inf. Process. Manage., 42(5):1163-1175, (2006).

[28] Merkl, D. and Rauber, A., "Document classification with unsupervised neural networks", In F. Crestani & G. Pasi (Eds.), Soft computing in information retrieval (pp. 102-121), Germany: Physica Verlag and Co, ISBN:3790812994, (2000).

[29] Slonim, N. and Tishby, N., "Document Clustering using Word Clusters via the Information Bottleneck Method", In ACM SIGIR 2000, (2000).

[30] Rigouste, L., Capp´e, O. and Yvon, F., "Evaluation of a probabilistic method for unsupervised text clustering", In International Symposium on Applied Stochastic Models and Data Analysis, Brest, France, (May 2005).

[31] Larsen, B. and Aone, C., "Fast and Effective Text Mining Using Linear-time Document Clustering," KDD-99, San Diego, California (1999).

[32] Srinivasan S., H., "Features for Unsupervised Document Classification", In Proceedings of the Conference on Computational Natural Language Learning, 36-42, Taipei, Taiwan, (2002).

[33] Hofmann, T., "The Cluster-Abstraction Model: Unsupervised Learning of Topic Hierarchies from Text Data", Proc. 16th Int'l Joint Conf. Artificial Intelligence (IJCAI-99), 682-687, (1999).

[34] Doucet, A., Lehtonen, M. "Unsupervised classification of text-centric XML document collections", In: Workshop of the INitiative for the Evaluation of XML Retrieval, (2006).

[35] Slonim, N., Friedman, N. and Tishby, N., "Unsupervised document classification using sequential information maximization", In Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval, 129-136, (2002).

[36] Caillet, M., Pessiot, J. F., Amini, M. R., Gallinari, P., "Unsupervised Learning with Term Clustering for Thematic Segmentation of Texts", Proceedings of RIAO, (2004).

[37] Reinberger, M. L. and Spyns, P., "Unsupervised text mining for the learning of dogma-inspired ontologies", In P. Buitelaar, P. Cimiano, and B. Magnini, editors, Ontology Learning from Text: Methods, Applications and Evaluation. IOS Press, (2005).

[38] Clustering-Introduction, http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/, (2012).

[39] k-means clustering - Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/K-means_clustering, (2012).

[40] Hierarchical clustering - Wikipedia, the free encyclopedia , http://en.wikipedia.org/wiki/Hierarchical_clustering, (2012).

[41] The DISTANCE Procedure: Proximity Measures :: SAS/STAT(R) 9.2 User's Guide, Second Edition, http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_distance_sect016.htm, (2012).

[42] The CLUSTER Procedure: Clustering Methods :: SAS/STAT(R) 9.2 User's Guide, Second Edition , http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_cluster_sect012.htm, (2012).

[43] Székely, G. J. and Rizzo, M. L., "Hierarchical clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method", Journal of Classification 22, 151-183, (2005).

[44] Koster, C. H. A. and Beney, J. G., "On the Importance of Parameter Tuning in Text Categorization", In: Perspectives of Systems Informatics. Vol. 4378. Lecture Notes in Computer Science Springer Berlin / Heidelberg, 270-283, (2007) .

[45] Stone, D., Jarrett, C., Woodroffe, M., and Minocha, S., "User Interface Design and Evaluation. San Francisco", CA: Morgan Kaufmann, book review by David Sturtz, Information Visualization (2006) 5, 77-78, (2005).

[46] Schulten, E., Akkermans, H., Botquin, G., Dörr, M., Guarino, N., Lopes, N., "The E-Commerce Product Classification Challenge", IEEE Intelligent Systems, 16(4), 86-89, (2001).

[47] Fensel, D., Ding, Y., Omelayenko, B., Schulten, E., Botquin, G., Brown, M., Flett, A., "Product Data Integration in B2B E-Commerce", IEEE Intelligent Systems 16, 54-59, (2001).

[48] Bergamaschi, S., Guerra, F., and Vincini, M., "A Data Integration Framework for e-Commerce Product Classification", Proc. First International Semantic Web Conference (Sardinia, Italy), 379-393, (June 9-12, 2002).

[49] J. Leukel, V. Schmitz, F. Dorloff, "A modeling approach for product classification systems", in: Proceedings of the DEXA Workshops, (2002).

[50] Beneventano, D., Guerra, F., Magnani, S., "A Web Service based framework for the semantic mapping amongst product classification", Journal of Electronic Commerce Research, 5:114-127, (2004).

[51] Leukel, J., Schmitz, V. and Dorloff, F. D., "Modeling and exchange of product classification systems using XML", Proceedings of the 4th IEEE International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems, (2002).

[52] Bergamaschi, S., Guerra, F. and Vincini, M., "Product Classification Integration for E-commerce", Proceedings of the 2nd International Workshop on Electronic Business Hubs (WEBH 2002), Aix-en-Provence, France, (2002).

[53] Kim, D., Lee, S.-g., Chun, J. and Lee, J., "A Semantic Classification Model for e-Catalogs", Paper presented at the IEEE Conference on E-Commerce Technology (CEC'04), San Diego, CA, USA, (2004, July 6-9).

[54] D. Beneventano, S. Magnani, "A framework for the classification and the reclassification of electronic catalogs", in Proc. of the 2004 ACM symposium on Applied computing, (2004).

[55] Omelayenko, B. and Fensel, D., "A Two-Layered Integration Approach for Product Information in B2B E-commerce", In Proceedings of the Second

International Conference on Electronic Commerce and Web Technologies (EC-WEB 2001), Munich, Germany, (September 2001).

[56] Lee, T., Lee, I., Lee, S., Lee, S., Kim, D., Chun, J., Lee, H. and Shim, J., "Building an operational product ontology system", Electronic Commerce Research and Applications, 5(1):16-28, (2006).

[57] D. Kim, S.-G. Lee, J. Chun, S. Park, and J. Oh, "Catalog management in e-Commerce systems", in Proceedings of Computer Science and Technology, Cancun, (2003).

[58] Mohanty, B. K. and Bhasker, B., "Product classification in the Internet business—a fuzzy approach", Decis. Support Syst., vol. 38, no. 4, 611-619, (Jan. 2005).

[59] Overview of Visual Studio 2010 Ultimate | Microsoft Visual Studio, http://www.microsoft.com/visualstudio/en-us/products/2010-editions/ultimate/overview, (2012).

[60] Microsoft SQL Server | Previous Versions, http://www.microsoft.com/sqlserver/en/us/editions/previous-versions.aspx, (2012).

[61] MATLAB - The Language of Technical Computing, http://www.mathworks.com/products/matlab/, (2012).

[62] Visual C#, http://msdn.microsoft.com/en-us/vstudio/hh388566, (2012).

[63] Windows Presentation Foundation - WindowsClient.net, http://windowsclient.net/wpf/, (2012).

[64] .NET Downloads, Developer Resources & Case Studies | Microsoft .NET Framework, http://www.microsoft.com/net, (2012).

[65] Html Agility Pack, http://htmlagilitypack.codeplex.com/, (2012).

[66] VATAN COMPUTER - Türkiye'nin Teknoloji Hiperstore'u, http://www.vatanbilgisayar.com, (2012).

[67] Hepsiburada.com | Türkiye'nin en büyük alışveriş merkezi, http://www.hepsiburada.com, (2012).

[68] Gold Bilgisayar - Türkiye'nin En Büyük Online Alışveriş Sitesi, http://www.gold.com.tr, (2012).

[69] Hizlial.com - Hızlı ve Güvenilir Online Alışveriş, http://www.hizlial.com, (2012).

[70] Ereyon.com.tr Alışverişin Güvenli Adresi, http://www.ereyon.com.tr, (2012).

[71] Webdenal.com - En Ucuz Fiyatlar Laptop, Projektör, Cep Telefonu, LCD Monitör, Ütü, Süpürge, Netbook, Taşınabilir Harddisk, http://www.webdenal.com, (2012).

[72] Darty TR, http://www.darty.com.tr, (2012).

[73] EXA Bilgisayar - Hesaplı Alışveriş, http://www.exa.com.tr, (2012).

[74] www.eksenbilgisayar.com, http://www.eksenbilgisayar.com, (2012).

[75] PratikEv.com : Alışverişin İnternet Adresi : Küçük Ev Aletleri, Cep Telefonu, Bilgisayar, LCD Televizyon ve Daha Binlerce Ürün En Uygun Fiyatlarla..., http://www.pratikev.com, (2012).

[76] Notebook | Notebook Fiyatları | Cep Telefonu Fiyatları | Cep Telefonu İncehesap, http://www.incehesap.com, (2012).

[77] Online Alışveriş - Netsiparis.com - Laptop, Notebook, Dizüstü Bilgisayarlar, http://www.netsiparis.com, (2012).

[78] Laptop , dell laptop , asus laptop , hp laptop , toshiba laptop , notebook , dizüstü bilgisayar, http://www.pcdepo.com, (2012).

[79] TeknoBiyotik: Ana Sayfa, http://www.teknobiyotik.com, (2012).

[80] En uygun fiyatları webdenbul, http://www.webdenbul.com, (2012).

[81] Birnumaram.com - Güvenli Bir Alışveriş İçin, http://www.birnumaram.com, (2012).

[82] Sanalmarketim.com - En Taze Teknoloji! - Internetten Güvenli Hızlı Alışveriş, http://www.sanalmarketim.com, (2012).

[83] Bimeks - Teknolojinin Kalbi Burada Atıyor, http://www.bimeks.com.tr, (2012).

[84] inventus - Ana Sayfa, http://inventus.com.tr, (2012).

[85] .:.Nova Bilgisayar.:. : Teknoloji Rehberiniz, http://www.novabilgisayar.com, (2012).

ÖZGEÇMİŞ VE ESERLER LİSTESİ

Adı Soyadı: Furkan GÖZÜKARA

Doğum Tarihi: 15/06/1987

Öğrenim Durumu:

Derece	Bölüm/Program	Üniversite	Yıl
Lisans	Bilgisayar Mühendisliği	İstanbul Teknik Üniversitesi	2004-2009
Yüksek Lisans	Bilgisayar Mühendisliği	Mersin Üniversitesi	2010-2012