İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ONTOLOJİ TABANLI İLİŞKİSEL ÜRÜN ÖNERİ SİSTEMİ

YÜKSEK LİSANS TEZİ

Hikmet KAPUSUZOĞLU

Anabilim Dalı : Bilgisayar Mühendisliği

Programı : Bilgisayar Mühendisliği

HAZİRAN 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ONTOLOJİ TABANLI İLİŞKİSEL ÜRÜN ÖNERİ SİSTEMİ

YÜKSEK LİSANS TEZİ
Hikmet KAPUSUZOĞLU
(504081519)

Tezin Enstitüye Verildiği Tarih : 5 Mayıs 2011
Tezin Savunulduğu Tarih : 7 Haziran 2011

Tez Danışmanı : Doç.Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ (İTÜ)
Diğer Jüri Üyeleri : Yrd. Doç. Dr. Şima ETANER UYAR (İTÜ)
Prof. Dr. Çoşkun SÖNMEZ (YTÜ)

HAZİRAN 2011

Aileme,

ÖNSÖZ

Çalışmalarım boyunca bana değerli bilgi ve deneyimleri ile yol gösteren hocam Sayın Doç. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ'ye ilgisi ve sabrı için içtenlikle teşekkür ederim.

Yüksek lisans eğitimim boyunca bana değerli katkılar veren İTÜ Bilgisayar Mühendisliği ailesinin değerli hocalarına ayrı ayrı teşekkür ederim.

Tez çalışmamda gerekli olan veri setini sağlayan Yapı Kredi Yayınları A.Ş'ye teşekkürü borç bilirim.

Son olarak her zaman bana inanan, güvenen ve destek olan annem Hülya KAPUSUZOĞLU, babam Ahmet KAPUSUZOĞLU ve tez verilerinin hazırlanmasında yanımda olan kardeşim Nermin KAPUSUZOĞLU'na çok teşekkür ediyorum.

Mayıs 2011

Hikmet Kapusuzoğlu

(Bilgisayar Mühendisi)

İÇİNDEKİLER

Sayfa

KISALTMALAR

GA : Genetik Algoritma
GKB : Genişletilmiş Kosinüs Benzerliği
GÖU-1 : Genişletilmiş Öklid Uzaklığı-1
GÖU-2 : Genişletilmiş Öklid Uzaklığı-2
İF : İşbirlikçi Filtreleme
İTF : İçerik Temelli Filtreleme
KB : Kosinüs Benzerliği
OSD : Ortalama Saflık Değeri
ÖU : Öklid Uzaklığı

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

ONTOLOJİ TABANLI İLİŞKİSEL ÜRÜN ÖNERİ SİSTEMİ

ÖZET

İnternet üzerinden alışverişin yaygınlaşmasıyla, elektronik ticaret Web sitelerinde ürün önerme daha önemli bir hale gelmiştir. Ürün önerme sistemleri, kullanıcıların Web sitesine ait sayfalarda dolaşırken bıraktığı bilgileri kullanarak kullanıcılara öneride bulunur. Ürün önerme sistemlerinin amacı kullanıcılarının karar vermelerini kolaylaştırmak ve ilgilendikleri ürünlere hızlı ve kolayca ulaşmalarını sağlamaktır. İşbirlikçi filtreleme ve içerik tabanlı filtreleme yöntemleri, ürün önerme sistemlerinde elektronik ticaret Web siteleri için en yaygın kullanılan yöntemlerdir. İşbirlikçi filtreleme kullanıcı tercihlerinin benzerliğine, içerik temelli filtreleme ürünlerin benzerliğine dayalı olarak ürün öneren sistemlerdir. İşbirlikçi filtrelemede benzer özellikler gösteren kullanıcılar gruplanırken, içerik temelli filtrelemede benzer özellikler gösteren ürünler gruplanır. Öneri aşamasında işbirlikçi filtrelemede kullanıcı profilinin en yakın olduğu grup; içerik temelli filtrelemede ise kullanıcının ziyaret ettiği ürünlere en yakın olan grup belirlenerek bu grup içerisinden öneride bulunulur. Ancak bu yöntemler soğuk başlangıç, eleman seyrekliği ve karmaşık ürünlerin önerimindeki yetersizlik problemleri ile karşı karşıyadır. Örneğin, sisteme yeni katılan bir ürün veya kullanıcı, bir gruba dahil olmadığı için öneride bulunulamaz veya ürünlerin az bir kısmının kullanıcılar tarafından beğenilmesi durumunda hep aynı ürünün önerilme riski bulunmaktadır. Ayrıca bu yöntemler ürünlerin derin anlamsal ilişkilerini yakalayamadığından karmaşık ürünlerin öneriminde yetersizdir. Etkili ve doğru bir öneride bulunmak için son çalışmalar, etki alanı ontolojisini de önerme işlemine dahil ederek verinin anlamsal özelliklerinden yararlanmayı hedeflemektedir. Bu çalışmalarda etki alanı ontolojisi sadece öneride bulunulacak ürünün çeşit ve niteliklerini kapsamaktadır ve ürünün ilişkisel özellikleri göz ardı edilmektedir. Aslında öneride bulunulacak ürünün ilişkide olduğu diğer kavramların ontolojisinin de öneri sistemine dahil edilmesi gerekmektedir. Bu çalışmada ilişkisel verinin etki alanı ontolojisi kullanılarak öneri sistemine dahil edilmesine odaklanılmış ve gerçeklenmesi kolay bir altyapı geliştirilmiştir. Bu altyapı kullanılarak kullanıcılara kitap önerisinde bulunan bir çalışma gerçeklenmiştir. Önerilen altyapının performansı, bir internet kitap mağazasına ait veriler üzerinde değerlendirilmiş ve sonuçlar önerilen altyapının ilişkisel verilerde etkin bir şekilde kullanılabileceğini göstermiştir.

A RELATIONAL RECOMMENDER SYSTEM BASED ON DOMAIN ONTOLOGY

SUMMARY

Product recommendation on electronic commerce Web sites becomes more important with the widespread use of Internet-based shopping. Recommendation systems utilize the information that users leave while navigating the Web pages in order to make recommendation. The aims of the recommendation systems are to simplify the making decision for users and provide a fast and easy way to access the products which they are interested. Collaborative filtering and content based filtering methods have been commonly used for this task by electronic commerce Web sites. Collabarative filtering is a recommendation system which is based on similarity of preference of the users and content based filtering is a recommendation system which is based on similarity of products. Collabarative filtering system groups users who are similar and content based filtering systems group products which are similar. In recommendation phase the closest group to user profile is determined in collabarative filtering and the closest group to the product which is in active session is determined in content based filtering. These methods have several shortcomings, such as cold start problem, sparseness and insufficient recommendation for complex objects. For instance a new user or a new product is not included in any group so recommendation is not available for this user or this product will never be recommended. Moreover if the knowledge about products which are prefered by users is sparse, same products may be recommended every time. Finally, this methods can not handle with deep semantic knowledge of products. Thus they are insufficient for recommendation of complex objects. In order to produce effective and accurate recommendations, recent approaches utilize the semantic properties of data by integrating the domain ontology into the recommendation process. In these studies, the domain ontology covering only the types and properties of the product to be recommended is considered where the relational nature of the product data is omitted. However, the domain ontology of the features related to the product may also provide useful information during recommendation process. In this study, we focus on integrating domain ontology of relational data into the recommendation process. We design a framework for an easy implementation of a recommendation system on relational data. Using this framework, we implement as a case study a recommendation model that recommends books to the users. We evaluated the performance of our model on real data obtained from a Turkish Internet book store. Our experimental results show that our proposed method can be effectively used for recommending items in relational data.

1. GİRİŞ

Elektronik ticaret, işlem kolaylığı nedeniyle artan bir ivme ile yaygınlaşmaktadır ancak elektronik ticaret Web sitelerindeki fazla ürün çeşitliliği kullanıcıların karar vermelerini ve ilgilendikleri ürünlere hızlı ve kolayca ulaşmalarını zorlaştırmaktadır. Kullanıcıların Web sayfalarını gezerken karar vermelerini kolaylaştırmak ya da ilgilenebilecekleri ürünleri kullanıcılara sunmak için öneri sistemleri yeni bir gereksinim olarak ortaya çıkmıştır. Öneri sistemleri, kullanıcının sitede dolaşırken bıraktığı verilerden bilgi üreterek, kullanıcıya kullanıcının ilgilenebileceği ürünleri önerir. Öneri sistemleri, Amazon [1] ve Pandora [2] gibi pek çok elektronik ticaret Web sitesi tarafından kullanılmaktadır. Elektronik ticaret Web sitelerinin sayısındaki hızlı artış sonucu oluşan rekabet, bu Web sitelerinin kullanıcılarına sunduğu önerilerin başarısını artırmak için çeşitli yöntemlerin oluşumunu tetiklemiştir. İşbirlikçi Filtreleme (İF) ve İçerik Temelli Filtreleme (İTF) ürün önerme sistemlerinde en çok kullanılan yöntemlerdendir.

İF, kullanıcı tercihlerinin benzerliğine dayalıdır. Benzer özellikleri beğenen veya benzer özellikler gösteren kullanıcılar gruplanır ve öneri yapılacak kullanıcı profiline en yakın olan grup belirlenir, bu gruptaki kullanıcıların tercihleri öneri olarak sunulur. İF eleman seyrekliği ve soğuk başlangıç adı verilen, yeni eleman problemleri ile karşı karşıyadır [3]. Sisteme yeni bir içerik (ürün) dahil edildiğinde, bu içerik daha önce hiçbir kullanıcı tarafından tercih edilmediğinden kullanıcılara önerilemeyecektir. Tercih edilme, kullanıcı tarafından sayfanın ziyaret edilmesi, içeriğin satın alınması veya kullanıcının içerik hakkında oy kullanması şeklinde olabilir. İF'nin kullanıcılar arasındaki ortak tercihlere dayalı olması ve yeni bir kullanıcının sistemde kayıtlı bir tercihi bulunmaması da yeni bir kullanıcıya öneride bulunmayı engellemektedir. İF'ye ait bir diğer problem ise seyrekliktir: içeriklerin az bir kısmı için kullanıcı beğenilerinin bulunması durumu, hep aynı içeriklerin önerilmesine neden olacaktır [4].

İTF' de, içerikler (ürünler) belirli özelliklerine göre gruplanır ve kullanıcıyı temsil eden bir kullanıcı profilinine en yakın olan grup belirlenerek bu grup içerisindeki içerikten önermede bulunulur. Kullanıcı profili, kullanıcının daha önce satın aldığı, beğendiğini ifade ettiği veya ziyaret ettiği içerikler (ürünler) ile belirlenir. İTF yöntemi, limitli çeşitlilik ve karmaşık ürünlerin önerimindeki yetersizlik problemleri ile karşı karşıyadır. İTF sisteminde bulunmayan niteliklere sahip bir içeriğin kullanıcılara önerilmesi mümkün olmayacaktır [5]. İTF'ye ait bir diğer problem ise karmaşık ürünlerin önerimindeki yetersizliktir. İTF, ürünlerin derin anlamsal ilişkilerini yakalayamaz. Örneğin bir ürün sadece nitelikleri ile temsil edilirse, film ve yönetmen arasındaki ilişki veya ders ve öğrenci arasındaki ilişki göz ardı edilmiş olur [6].

Her iki yöntemin de eksikliklerini tamamlamak için iki yöntemi birleştiren melez çalışmalar da vardır. İki yöntemin birleştirilmesi farklı yöntemlerle gerçekleştirilebilmektedir. Her iki yöntem de ayrı ayrı uygulayıp sonuçlar ağırlıklandırılarak birleştirilebilir, bazı İTF özellikleri İF yöntemine dahil edilebilir veya bazı İF yöntemi özellikleri İTF yöntemine dahil edilebilir [7].

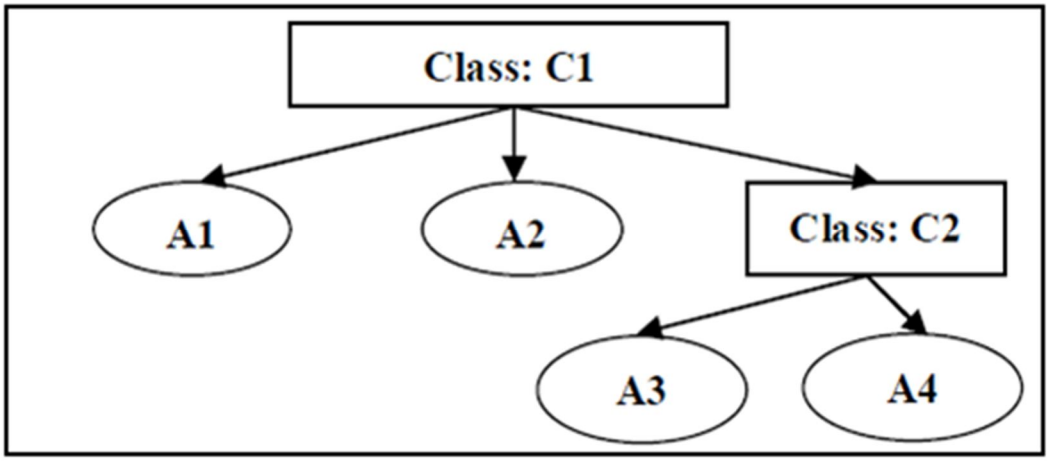
İF ve İTF yöntemlerinde bulunan problemleri çözmek ve daha doğru ürün öneriminde bulunmak için güncel çalışmalarda ontolojiler kullanılmaktadır. Ontoloji, Thomas Gruber'in ifadeleriyle bir kavramsallaştırmanın tanımlanmasıdır [8]. Belirli bir etki alanındaki kavramlar, bu kavramlar arasındaki ilişkiler ve kavramlara ait örnekler ontolojiyi oluşturur. Ontoloji kullanan sistemlerin temelinde içerik hakkındaki bilgilerin ve bağıntıların kullanılması vardır. Ontolojiler kullanılarak, nesnelerin derin anlamsal karakterlerinden faydalanmak ve karmaşık nesnelerin önerilmesinde başarıyı artırmak mümkündür [9].

1.1 Tezin Amacı

Ürün öneriminde ontoloji kullanan pek çok çalışmada, ürün ontolojisinin homojen bir yapıda olduğu varsayılmıştır. Bu çalışmalarda ontolojiler tek bir sınıfı içerecek şekilde ele alınmış ve/veya bu sınıfa ait niteliklerin karmaşık yapıda olabileceği göz ardı edilmiştir.

Ontoloji temelli ürün önerme sistemlerinde yüksek başarı elde etmek için, ontolojiler önerilecek sınıfın ilişkide olduğu diğer sınıfları da kapsamalıdır çünkü kullanıcıların yaptıkları tercihlerde bu sınıf elemanlarının da katkısı vardır. Örneğin bir ürünün kullanıcılar tarafından beğenilme olasılığını belirlerken, sadece ürünün niteliklerini değil, ürünün üreticisinin niteliklerinin de hesaba katılması gerekmektedir. Bu doğrultuda oluşturulmuş bir veri kümesinde, ürünlerin farklı sayıda ilişkide oldukları nesne ve her nesnenin çoklu değer kümesine sahip niteliği olabilir. Soyut bir etki alanı ontolojisi Şekil 1.1'de verilmiştir. Bu şekilde, *C1* sınıfı, nesnelerinin önerileceği sınıftır. Bu sınıf "hedef sınıf" olarak adlandırılmıştır. *C1* sınıfının iki niteliği bulunmaktadır: *A1* ve *A2*, ayrıca *C1* sınıfı başka bir sınıfla, *C2* sınıfı ile ilişkilidir.

C2 sınıfı, *C1* sınıfının alt sınıfıdır çünkü *C2* sınıfı hedef sınıf olan *C1* sınıfı ile ilişkilidir ve kendi nitelikleri vardır (*A3* ve *A4*). Örneğin bir kitap öneri sisteminde *C1*, kitap sınıfına; *A1* ve *A2* kitap sınıfına ait konu ve basım yılı niteliklerine; *C2*, yazar sınıfına ve *A3* ile *A4* yazarların aldığı ödüller ve yazarların yaşlarını belirten niteliklere karşılık gelebilir.



Şekil 3.1: Soyut Etki Alanı Ontolojisi.

Bu çalışmada ontoloji temelli, ilişkisel ve anlamsal öneri sistemi için yeni bir altyapı geliştirilmiştir. Çalışmanın çıkış noktası önerilecek ürünlerin niteliklerinin ve ürünlerin ilişkide olduğu diğer nesnelerin niteliklerinin ilişkisel olarak kullanılması ve katsayılar atanarak ağırlıklandırılmasıdır. Niteliklerin katsayıları genetik algoritma ve İF temelli bir yöntem kullanılarak 2 farklı yöntemle tespit edilmiştir.

Önerilen altyapı 5 aşamadan oluşmaktadır: İlk aşamada elektronik ticaret Web sitesine ait erişim kütüklerindeki kayıtlar, kullanıcı oturumlarına dönüştürülür. Kullanıcı oturumları, kullanıcılar tarafından bir oturum boyunca ziyaret edilmiş ürünlerden oluşmaktadır. İkinci aşamada, ürünün nitelikleri altsınıfın niteliklerini de kapsayacak şekilde genişletilir. Sonraki aşamada, genetik algoritma temelli yöntemler kullanılarak nitelikler ağırlıklandırılır. Dördüncü aşamada, elektronik ticaret Web sitesinden elde edilen nitelikler üçüncü aşamada elde edilen katsayılar kullanılarak kümelenirler; böylece arama uzayının küçültülmesi hedeflenmiştir. Son aşama öneri aşamasıdır. Kullanıcı tarafından en son ziyaret edilmiş ürüne en yakın küme merkezi belirlenir ve bu küme içerisindeki ürüne en çok benzeyen ürünler kullanıcıya önerilir. Bu altyapıyı kullanarak bir ontoloji tabanlı ilişkisel ve anlamsal öneri modeli geliştirilmiş ve performansı bir Web kitap mağazasında değerlendirilmiştir. Altyapı diğer ürünler için de kullanılabilir.

1.2 Tezin Yapısı

Tez 5 temel bölümden oluşmuştur. Bölüm 1'de çalışmaya ait temel bilgiler belirtilmiştir. Bölüm 2'de konu ile ilgili yapılan önceki çalışmalar belirtilmiştir. Bölüm 3'de bazı kuramsal bilgiler verilmiştir. Bölüm 4'de önerilen model detaylarıyla açıklanmıştır. Bölüm 5'de deneysel sonuçlar verilmiştir.

2. ÖNCEKİ ÇALIŞMALAR

2.1 İşbirlikci Filtreleme ve İçerik Filtreleme Tabanlı Çalışmalar

Ürün önerme sistemlerine ait çalışmaların büyük bir bölümü İF ve İTF yöntemlerini kullanırlar. İF yöntemi, kullanıcıları bir bağıntı veya benzerlik hesabı kullanarak bir vektör olarak temsil eder ve vektörler arasındaki benzerliğe dayalı önerilerde bulunur [10]. Örneğin [11]'de belirtilen çalışmada kullanıcıların ürünlere verdiği oylardan oluşan bir kullanıcı-ürün matrisi oluşturulmuş ve matristeki seyrek veri yapısının neden olabileceği başarısızlığı önlemek için tahmin algoritması geliştirilerek matristeki eksik verilerin azaltılması hedeflenmiştir. Tahmin algoritması, kullanıcılar arasındaki benzerliği kullanarak eksik verileri tamamlamayı amaçlar. Kullanıcılar arası benzerlik, kullanıcıların oy verdikleri ürünler kullanılarak Pearson korelasyon katsayısı ile belirlemiştir. Eksik verinin tamamlanması, tüm eksik veriler için değil sadece doğruluğu belli bir eşik değerinin üzerindeki veriler için uygulanmıştır. Öneri aşamasında aktif kullanıcının vektörel temsiline en yakın vektörler kullanılmıştır.
Kullanıcı sayısının çok büyük olduğu durumlarda bu yöntemin kullanılması zorlaşmaktadır. Bu problemi aşmak için öneri sistemlerinde ürünler arasındaki ilişkiden yararlanılmaya başlanmıştır [12]. İTF' ye dayalı yöntemler tüm kullanıcıların tercihi yerine sadece öneri yapılacak kullanıcının tercihleri ile ilgilenir. İTF' ye ait örnek bir çalışma olan Pazzani ve diğerlerinin çalışmasında doküman öneri sistemi gerçeklenmiştir [5]. Bu çalışmada önce her döküman için indeks oluşturulur. Bir kullanıcıya bu indeksleri içeren bir doküman görüntülediğinde sistem önceden indekslenmiş dokümanlar içerisinden öneride bulunur. İlgili dokümanın seçiminde Bayesian sınıflandırma yöntemi kullanılmıştır. İTF yöntemi kullanan çalışmalardan bir diğeri ise Pandora [2] için geliştirilmiş Müzik Genom Projesidir. Bu projede müzik önerisi, müzik parçalarının niteliğine ve kullanıcı tercihlerine göre gerçeklenmiştir. Öneride bulunurken parçada piyano çalınıp çalınmadığı gibi 400'den fazla nitelik değerlendirilmiştir [13].

İF ve İTF yöntemlerini içeren melez sistemler ile soğuk başlangıç ve limitli çeşitlilik gibi İF ve İTF yöntemlerindeki problemleri çözmeyi ve daha yüksek oranda bir başarı elde etmeyi amaçlayan çalışmalar da vardır. Souvik ve diğerlerinin çalışmasında, İTF yöntemi kullanılmış ve içeriğe ait benzerlikler hesaplanırken nitelikler İF yöntemi ile katsayılar atanarak ağırlıklandırılmıştır. Katsayıların belirlenmesindeki temel nokta kullanıcıların hangi niteliklere önem verdiğinin tespit edilmesidir. Niteliklere ait katsayıların belirlenmesi için düğümlerin önerilecek ürünler olduğu ve ayrıtların iki düğümü de seçen kullanıcı sayılarının olduğu bir sosyal ağ oluşturulmuş ve bu ağın ayrıtları ve düğümleri kullanılarak elde edilecek doğrusal bağlanım denklemleri ile katsayılar belirlenmiştir [14] . Bu çalışmaya ait başarım testlerinde İTF kullanılarak niteliklerin ağırlıklandırılması durumunda başarının arttığı görülmüştür. Melez yöntemleri kullanan çalışmalara ait bir diğer örnek ise Melville ve diğerlerinin çalışmasıdır. Bu çalışmada var olan kullanıcı verisi İTF tahmin yöntemleri kullanılarak genişletilmiş ve İF kullanılarak öneride bulunulmuştur [9]. Böylece daha önce tercih edilmemiş bir içeriğin de önerilebilinmesi sağlanmıştır.

2.2 Ontoloji Tabanlı Çalışmalar

Ontoloji tabanlı çalışmalar, ürün önerme sistemlerinde kullanılan son yaklaşımlardandır. Bu çalışmaların büyük bir çoğunluğu, kullanıcının ziyaret ettiği Web sayfalarını ontolojideki sınıfın nesneleri ile eşleyerek kullanıcının Web sayfalarından oluşan işlem dizisini, ontolojik nesnelerden oluşan işlem dizisine dönüştürür ve elde edilen işlem dizisi üzerinde veri madenciliği yöntemleri uygular. Ontoloji kullanan pek çok çalışma, ontolojiyi tek bir sınıf içeren basit bir veri yapısında ele almış, ontolojik sınıfların ilişkisel bir yapıda bulunduğunu göz ardı etmiştir.

Dai ve Mobasher'in çalışmalarında Web sayfası önerisinde bulunan ontoloji tabanlı bir altyapı anlatılmıştır [15]. Altyapı, Web sayfalarından oluşan oturumları, ontolojideki nesnelerden oluşan oturumlara dönüştürdükten sonra bu oturumları tek bir nesne ile temsil edebilmek için birleştirme fonksiyonları kullanmıştır. Birleştirme fonksiyonlarının her nitelik için önceden verildiğini varsayılmaktadır. Oturumdaki nesnelerin tek bir nesneye indirgenmesinde kullanılan birleştirme fonksiyonlarında her nesneye bir katsayı atanmıştır.

Katsayılar kullanıcıların oturumlardaki sayfa ziyaret süreleri olarak kabul edilmiştir. Her oturum için gerçekleştirilen birleştirme işlemi sonrası elde edilen ve oturumu temsil eden nesneler kümelenmiştir. Öneride bulunulacak aktif oturum da birleştirme fonksiyonu ile bir nesne ile temsil edilmiş ve bu nesneye en yakın küme merkezi belirlenerek bu kümedeki en yakın nesneler kullanılarak öneride bulunulmuştur. Çalışmada önerinin nasıl gerçekleneceğine dair detay bulunmamaktadır.

Ontolojideki nesnelerden oluşan oturumları kullanan bir diğer ürün öneri sistemi ise yaygın örüntülerin tespitine dayalıdır [16]. Bu sistemde erişim kütüğü dosyalarındaki Web sayfaları, ontolojideki nesnelere dönüştürülür ve SPADE algoritması ile ilişkilendirme kuralları tespit edilir. Öneri aşamasında ise öneride bulunulacak oturumdaki sayfalar ontolojideki nesnelere çevirilir ve belirlenen ilişkilendirme kurallarına göre önerilecek nesne belirlenir. Belirlenen nesne temsil ettiği Web sayfasına yeniden dönüştürülerek bu sayfanın önerilmesi sağlanır.

İlişkilendirme kuralları ve benzerlik yöntemleri gibi iki farklı yöntemi bir arada kullanan ontoloji tabanlı çalışmalar da vardır. Mabroukeh ve Ezeife, "Semrec" adını verdikleri yöntemde [17] öncelikle tüm nesnelerin birbirlerine olan uzaklıklarından oluşan bir matris oluşturmuşlardır. Sonraki aşamada ontolojideki nesnelerden oluşan oturumları Markov zinciri modelinde kullanarak geçiş olasılıklarını elde etmiş ve öneride bulunulacak aktif oturum için önerilebilecek nesneleri bu olasılıklara göre belirlemişlerdir. Önerilebilecek bu nesneler arasından aktif oturumdaki nesnelere uzaklığı matriste en az olan nesneler seçilmiş ve önerilmiştir.

Ontoloji tabanlı bazı öneri sistemlerinde ise ontolojinin birden fazla sınıftan oluşan karmaşık ve ilişkili bir yapıda olduğu belirtilmiş ancak bu yapının ürün öneriminde nasıl ele alınacağı açıklanmamıştır. Örneğin bir çalışmada [6] öneri sistemi 3 aşamadan oluşacak şekilde ele alınmıştır: verinin hazırlanması, örüntünün bulunması ve öneri aşamaları.

Verinin hazırlanması aşamasındaki amaç oturumlardaki sayfaları ontolojideki nesnelerle eşleyerek kullanıcının Web sitesindeki gezintisinden oluşan işlem dizisini, anlamsal işlem dizisine dönüştürmektir. Bu çalışmada ontolojiler birden fazla sınıfı içerebilecek şekilde ele alındığından, örüntü bulunması aşamasında her sınıf için ontolojideki nesnelerden oluşan oturumlar birleştirme fonksiyonları kullanılarak tek bir nesne formuna indirgenir.

Film ontolojisi örneğinde film ve oyuncular olarak iki sınıf olduğu kabul edilirse; film ve oyuncu nesnelerini içeren bir oturumdan, birleştirme fonksiyonları yardımıyla film sınıfı için bir tane ve oyuncu sınıfı için bir tane sanal nesne yaratılmış olacaktır. Birleştirme fonksiyonları için genel bir kural verilmemiş, ontolojideki nesnelerin niteliklerine göre değişebileceği belirtilmiştir. Örneğin yıl belirten bir nitelik için birleştirme fonksiyonu ortalama hesaplayabilir veya yıl aralığını 10 yıl içeren bir yapıya dönüştürebilir. Birleştirme fonksiyonları ontolojideki her sınıf için oturumlarda ayrı ayrı çalıştırılmış ve sonuçta her sınıf için oturumların birleştirilmesiyle oluşan bir sanal nesneler kümesi elde edilmiştir.

Öneri aşamasında, öneri yapılacak aktif oturumun profili ontolojideki her sınıf için birleştirme fonksiyonları ile tek bir nesne ile temsil edilecek yapıya indirgenmiş ve her nesnenin bir önceki aşamada elde edilen sanal nesneler kümelerindeki nesnelere benzerliği belirlenmiştir. Her sınıf için benzerliği belli bir eşik değerinin üzerinde olan nesneler birleştirilerek önerilecek nesne formuna getirilmiştir. Film örneğinde, en yakın film nesnesi ve en yakın oyuncu nesnesi belirlenmiş ve bu nesnelerin birleşiminden oluşan nesneye karşılık gelebilecek ürün önerilmiştir. Bu çalışmada ontolojinin ve ontolojideki nesnelerin var olduğu kabul edilmiş, ancak ontolojideki nesneler arasındaki benzerliğin nasıl hesaplanacağı gibi konular açık bırakılmıştır.

3. KURAMSAL BİLGİLER

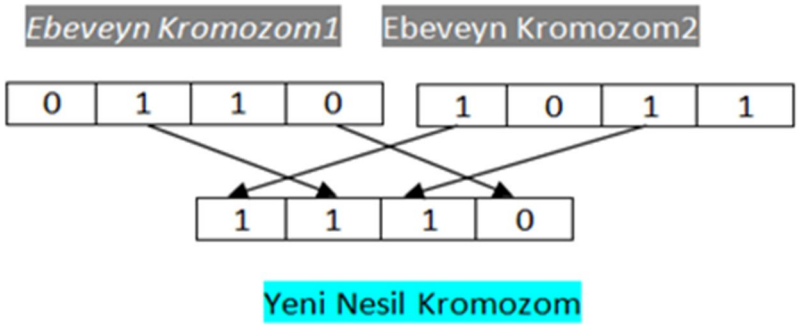
Detayları Bölüm 4'de açıklanmış olan ontoloji tabanlı ilişkisel ürün öneri sisteminin gerçeklenmesinde kullanılan yöntemler ile ilgili bilgiler bu bölümde verilmiştir. Önerilen bu sistemde ontolojideki sınıfların nitelikleri İF temelli bir yöntem ve genetik algoritma temelli bir yöntem kullanılarak 2 farklı yöntemle katsayılar atanarak ağırlıklandırılmıştır. Genetik algoritmaya ait kuramsal bilgiler Bölüm 3.1'de verilmiştir. Ontolojideki sınıfların nesnelerinin birbirleri ile olan uzaklıklarının ve benzerliklerinin hesaplanmasında yararlanılan "ilişkisel verilerde uzaklık hesaplama yöntemleri" Bölüm 3.2'de açıklanmıştır.

3.1 Genetik Algoritma

Genetik algoritma (GA) doğada gözlemlenen evrimsel süreci taklit eden, arama tabanlı, sezgisel bir yöntemdir. Optimizasyon problemlerinde geniş bir kullanım alanı vardır. GA'lar optimizasyon problemlerinde parametre kümesini değil, bu kümenin kodlanmış biçimini girdi olarak alırlar böylece çözüm uzayının tamamı yerine belirli bir kısmını taradıklarından daha kısa sürede çözüme ulaşırlar [18]. Kodlanmış bu biçime kromozom adı verilir ve optimize edilecek parametreler kromozomlar ile gösterilir. Her bir parametre gen olarak adlandırılır. Bu nedenle bir kromozom, optimize edilecek parametre sayısı kadar genden oluşur. Popülasyon ise belirli bir sayıdaki kromozomların oluşturduğu yapıdır ve başlangıçta genlere atanan rastgele değerlerden oluşur. Uygunluk değer fonksiyonu, her biri bağımsız bir çözüm olan kromozomların problemi ne ölçüde başarılı olarak çözdüğünü belirler.

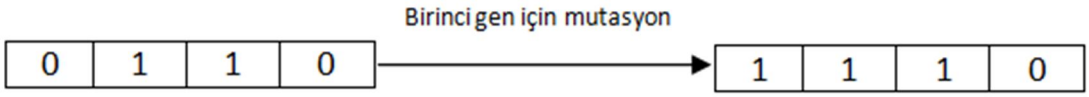
GA, seleksiyon, çaprazlama ve mutasyon gibi işlemler ile doğal seçim prosesini taklit eder ve arama uzayında en iyinin hayatta kalması ilkesine göre en iyi çözümü arar. Kromozomlar, uygunluk değerleri göz önüne alınarak bir sonraki nesli oluşturmak için çaprazlanır. Buradaki amaç iyi uygunluk değerlerine sahip kromozomları birleştirerek daha iyi uygunluk değerine sahip kromozomlar elde etmektir. Çaprazlama ebeveyn kromozomların genlerinin kullanılarak yeni nesil kromozomlar elde edilmesidir.

Örneğin, tıpta 4 farklı etken maddenin hangilerinin birarada seçileceğinin belirlenmesine dair bir problemde genler 0 ve 1 şeklinde ifade edilebilir. Maddenin seçimi durumu 1 ile, seçilmemesi durumu 0 ile ifade edilmiştir. Bu problemde 4 tane optimize edilecek değer olduğu için her kromozomda 4 adet gen olacaktır. Kromozomlara ait örnek bir çaprazlama işlemi Şekil 3.1'de belirtilmiştir.



Şekil 3.1: Genetik Algoritma Çaprazlama Örneği.

Çaprazlamalar sonucu kromozomların birbirini tekrarlamaması ve farklı çözümlere de ulaşılabilinmesi için mutasyon uygulanır. Mutasyon kromozomdaki genlerin değiştirilmesidir. Şekil 3.2'de örnek bir mutasyon işlemi verilmiştir.



Şekil 3.2: Genetik Algoritma Mutasyon Örneği.

GA'ların temel işleyişi aşağıdaki gibi açıklanabilir [19]:

- Bir kromozomun çözüm oluşturabilecek şekilde kodlarına karar verilmesi
- Başlangıç popülasyonunun rastgele veya tecrübe ile oluşturulması.
- Popülasyondaki kromozomların uygunluk değerlerinin belirlenmesi.
- Kromozomların uygunluk değerine göre yeni nesil kromozomların oluşmasını sağlamak için eşleşme ve mutasyonun gerçeklenmesi.
- Yukarıdaki işlemlerin belli bir kıstas sağlanınca kadar tekrarlanması.
- Popülasyondaki kromozomlardan uygunluk değeri en yüksek olanın çözüm olarak kabul edilmesi.

İşlem tekrar sayısını belirleyen kıstas, kromozomlardan birinin belli bir uygunluk değerine ulaşması veya çaprazlama sayısının belli bir değere ulaşmasıdır [20].

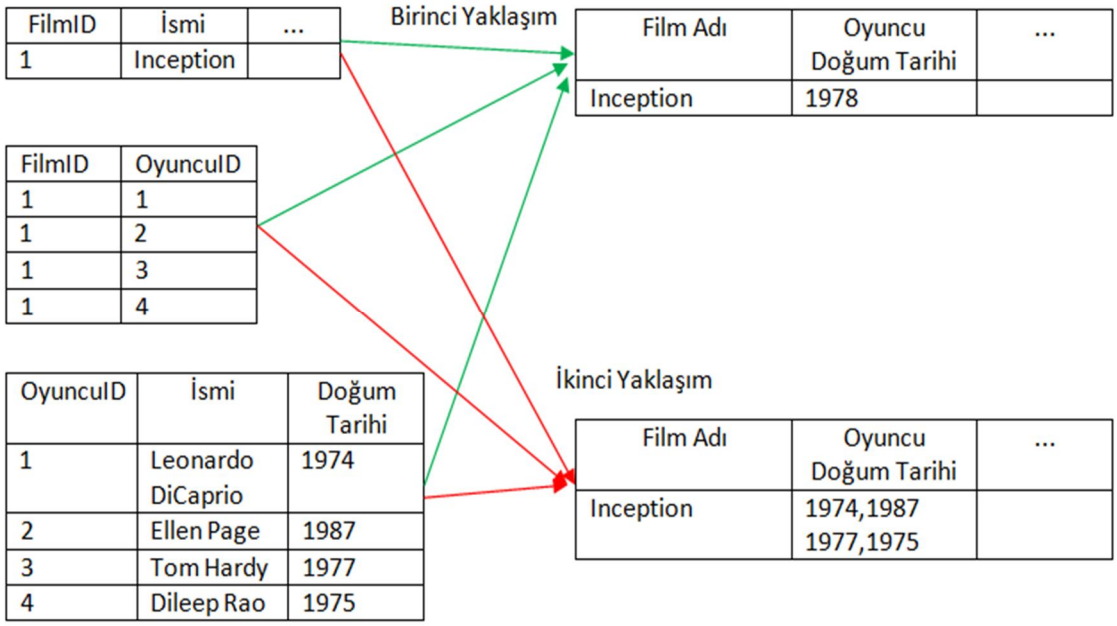
Genetik algoritmaların diğer optimizasyon çözümlerinden farkı aşağıdaki gibi listelenebilir [21]:

- GA, parametrelerin kendileri ile değil, parametre takımının kodlanış şekliyle ilgilenir.
- GA'nın yerel optimum noktaya takılma riski yoktur çünkü tek bir nokta ile çalışmaz, bir nokta kümesi ile çalışır.
- GA, uygunluk fonksiyonunun bir takım türevlerini değil doğrudan uygunluk fonksiyonunun kendisini kullanır.
- GA, deterministik değildir, rastlantısal geçiş kurallarını kullanır.

3.2 İlişkisel Verilerde Uzaklık Hesaplama

Öklid (Euclidean) uzaklığı gibi geleneksel uzaklık ölçümleri, iki vektörün uzayda ne kadar yakın olduğunun belirlenmesine dayalıdır ve iki vektör arasındaki uzaklığın N boyutlu uzayda ölçülmesi için kullanılırlar. Halbuki ilişkisel veriler genellikle karmaşık bir yapıdadır ve bir vektör olarak temsil edilebilmeleri güçtür. Bu nedenle geleneksel uzaklık ölçüm yöntemleri ilişkisel verilere uygulanamaz [22]. İlişkisel verilerin, veritabanı sistemlerinde birden fazla tablo ile temsil edilebilmesi ve niteliklerinin farklı sayılarda değerler alabilmesi, geleneksel uzaklık ölçüm yöntemlerinin kullanılabilinmesini engeller.

Bu probleme çözüm sunan farklı yaklaşımlar vardır: Birinci yaklaşımda, tablolar birleştirilir ve birden çoğa veya çoktan çoğa ilişkiye neden olan nitelikler, birleştirme fonksiyonları kullanılarak tek bir değere indirgenir; İkinci yaklaşımda ise tablolar birleştirilir ve birden çoğa veya çoktan çoğa ilişkiye neden olan nitelikler çok terimli bir değer kümesine sahip olabilecek şekilde modellenir [23]. Her iki yaklaşıma ait bir örnek Şekil 3.3'de verilmiştir.



Şekil 3.3: Tabloların Birleştirilmesi.

İlişkisel verinin tek bir tabloya indirgenmesi bilgi kaybına veya yüksek boyutlu ve seyrek bir veri yapısının elde edilmesine neden olur [24]. İlişkisel verilerde uzaklık hesaplamak için bir diğer çözüm ise uzaklık hesaplama yöntemlerinin birden çok tablo ile direk işlem yapabilecek duruma getirilmesidir [16]. Bu çözüm iki farklı yöntemle gerçeklenebilir. Birincisi, ilişki içerisinde bulunan her bir tablonun her bir sütunundaki niteliklere ait uzaklığın ayrı ayrı hesaplanması ve sonuçların aritmetik ortalamasının uzaklık olarak alınmasıdır. Örneğin Çizelge 3.1'de verilen *O* sınıfının

3 tane niteliği vardır ve bu niteliklerden Nitelik3, başka bir sınıfa, *SC* sınıfına bir referanstır. *SC* sınıfının Çizelge 3.2'de görüldüğü gibi 2 niteliği bulunmaktadır. Bu durumda *O* sınıfının nesnelerinden *O1* ve *O2* arasındaki uzaklık (3.1)'deki gibi hesaplanır. Formüldeki *n* Çizelge 3.1'deki; *k* ise Çizelge 3.2' deki sütun sayısıdır ve *dist(O1.At(i),O2.At(i))* fonksiyonu, *O1* ve *O2* nesnelerinin *i.* niteliklerinin uzaklık hesap fonksiyonudur.

$$dist(O1,O2)=\frac{\sum_{i=1}^{n} dist\left(O1.At(i),O2.At(i)\right)+\sum_{i=1}^{k} dist\left(SC1.At(i)\right),SC2.At(i))}{n+k-1} \qquad \textbf{(3.1)}$$

Çizelge 3.1 : "O" Sınıfı.

"O" sınıfının nesneleri	Nitelikler		
	Nitelik1	**Nitelik2**	**Nitelik3**
O1	At1	At2	SC1
O2	At1	At2	SC2

Çizelge 3.2 : "SC" sınıfı.

Instances Of Class "SC"	Nitelikler	
	Nitelik1	**Nitelik2**
SC1	SAt1	SAt2
SC2	SAt3	SAt4

İkinci yöntem, öncelikle referans tabloda bulunan *SC1* ve *SC2* nesneleri arasındaki uzaklığı hesaplar, sonra *O1* ve *O2* nesnelerindeki başka bir tabloya referansı bulunmayan niteliklerin (Nitelik1, Nitelik2) uzaklığını hesaplar ve bunların ortalamasını *O1* ve *O2*'nin uzaklığı olarak alır. Bu yöntem referans tablolardaki nesnelerin uzaklığını rekürsif olarak maksimum derinliğe ulaşıncaya kadar hesaplar [25]. Çizelge 3.1'de verilen *O1* ve *O2* nesneleri arasındaki uzaklık bu yaklaşımla aşağıdaki fonksiyonla hesaplanacaktır.

$$dist(O1,O2) = \frac{\sum_{i=1}^{n} dist\left(O1.At(i), O2.At(i)\right) + \frac{\sum_{i=1}^{k} dist\left(SC1.At(i), SC2.At(i)\right)}{k}}{n} \quad \textbf{(3.2)}$$

4. ÖNERİLEN MODEL

Bu çalışmada önerilen model 5 temel işlemden oluşmaktadır: oturum tanımlama, oturumu genişletme, katsayıları belirleme, kümeleme ve öneride bulunma.

4.1 Oturum Tanımlama

Kullanıcılara ait oturum bilgisi çerezlerden (cookie), vekil sunuculardan (proxy), uygulamalarda tutulan etkileşim bilgilerinden veya Web sunucusu erişim kütüklerinden elde edilebilir [26]. Ancak diğer veriler genellikle bulunmadığı veya erişimi zor olduğu için oturum bilgileri genellikle Web sunucusu erişim kütüklerinden elde edilir.

Web sunucularının erişim kütükleri, istemcilerden gelen isteklerin kayıt edilmesiyle oluşur ve tarih, saat, istemci IP adresi, istemci tarayıcı versiyonu, erişilen sayfanın URL bilgisi ve isteğin durumu gibi bilgileri içerir. Aşağıda örnek bir sunucu erişim kütüğü kaydı verilmiştir:

2010-12-14 23:58:30 212.154.28.114 GET /tanim.asp sid=XN0OUP49YT2LVEU1GGH 80 - 85.96.228.45 Mozilla/5.0+(Macintosh;+U;+Intel+Mac+OS+X+10.6;+en-US;+rv:1.9.1.16)+Gecko/20101130+Firefox/3.5.16 200 0 0

Sunucu erişim kütüklerinde tarihe göre oluşturulmuş farklı dosyalarda bulunan bu kayıtlar LogParser programı [27] gibi Web sunucusu erişim kütüğü analiz programlarıyla gerekli alanlar filtrelenerek birleştirilir. Çizelge 4.1'de bu işlem sonucu elde edilmiş örnek bir erişim kütüğü içeriği belirtilmiştir.

Çizelge 4.1 : Web Sunucusu Erişim Kütüğü İçeriği.

İstemci IP	Tarih ve Saat	İstemci Tarayıcı Versiyonu	URL	Durum Kodu
212.154.28.114	2010-12-14 00:01:03	Compatible;+Googlebot/2.1	/sid=KOU6G4GTZWGY0MA	200
195.87.213.60	2010-12-14 08:03:50	MSIE+7.0;+Windows+NT+5.1	/satis/body-background.jpg	200
78.163.137.96	2010-12-14 18:13:50	Mozilla/5.0Windows+NT+5.1;+en-US)	/sid=SFDKEWJR4WSD2JSA	500
77.100.56.23	2010-12-14 20:13:02	MSIE+6.0;+Windows+NT+5.1	/sid=ASF3FCSDVJ3GSICS3	200

Erişim kütüğünde bulunan kayıtlardan bir kısmı oturum tanımlamada kullanılamayacak kayıtlardır. Bu kayıtlar şu şekilde belirtilebilir:

- Durum kodu 200 den farklı olan istekler. Bu istekler bir hata ile sonuçlanmıştır.
- Resim dosyaları gibi çoklu ortam (multimedia) dosyası istekleri. Bu isteklerin URL alanı bir Web sayfasına karşı düşmemektedir.
- Örümcekler tarafından oluşturulmuş kayıtlar. Bir kaydın örümcek tarafından oluşturulup oluşturulmadığı istemci tarayıcı versiyonundan belirlenebilir. Örümcek olarak kabul edilen bazı istemci tarayıcı versiyonları aşağıda belirtilmiştir:

  Mozilla/5.0+(compatible;+Googlebot/2.1;++http://www.google.com/bot.html)

  Mozilla/5.0+(compatible;+bingbot/2.0;++http://www.bing.com/bingbot.htm)

  Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+SV1+LM+8;+MSIECrawler)

  Mozilla/5.0+(compatible;+YandexBot/3.0;++http://yandex.com/bots)

  Mozilla/5.0+(compatible;+MSIE+8.0;+Windows+NT+6.0;)

Oturum belirlemede kullanılmayacak, yukarıda belirtilen kayıtlar erişim kütüğünden temizlenir. Örneğin Çizelge 4.1'de verilen kayıtlar temizlendikten sonra Çizelge 4.2 elde edilir.

Çizelge 4.2 : Temizlenmiş Web Sunucusu Erişim Kütüğü İçeriği.

İstemci IP	Tarih ve Saat	İstemci Tarayıcı Versiyonu	URL (Sayfa)	Durum Kodu
77.100.56.23	2010-12-14 20:13:02	MSIE+6.0;+Windows+ NT+5.1	/sid=ASF3FCSDVJ3GSI CS3	200

Bu aşamada Çizelge 4.2'de örneği verilmiş ve tüm isteklerden oluşan bir liste ve bu isteklerde bulunan toplam *m* tane farklı sayfayı içeren bir sayfa kümesi "*P*" elde edilmiştir.

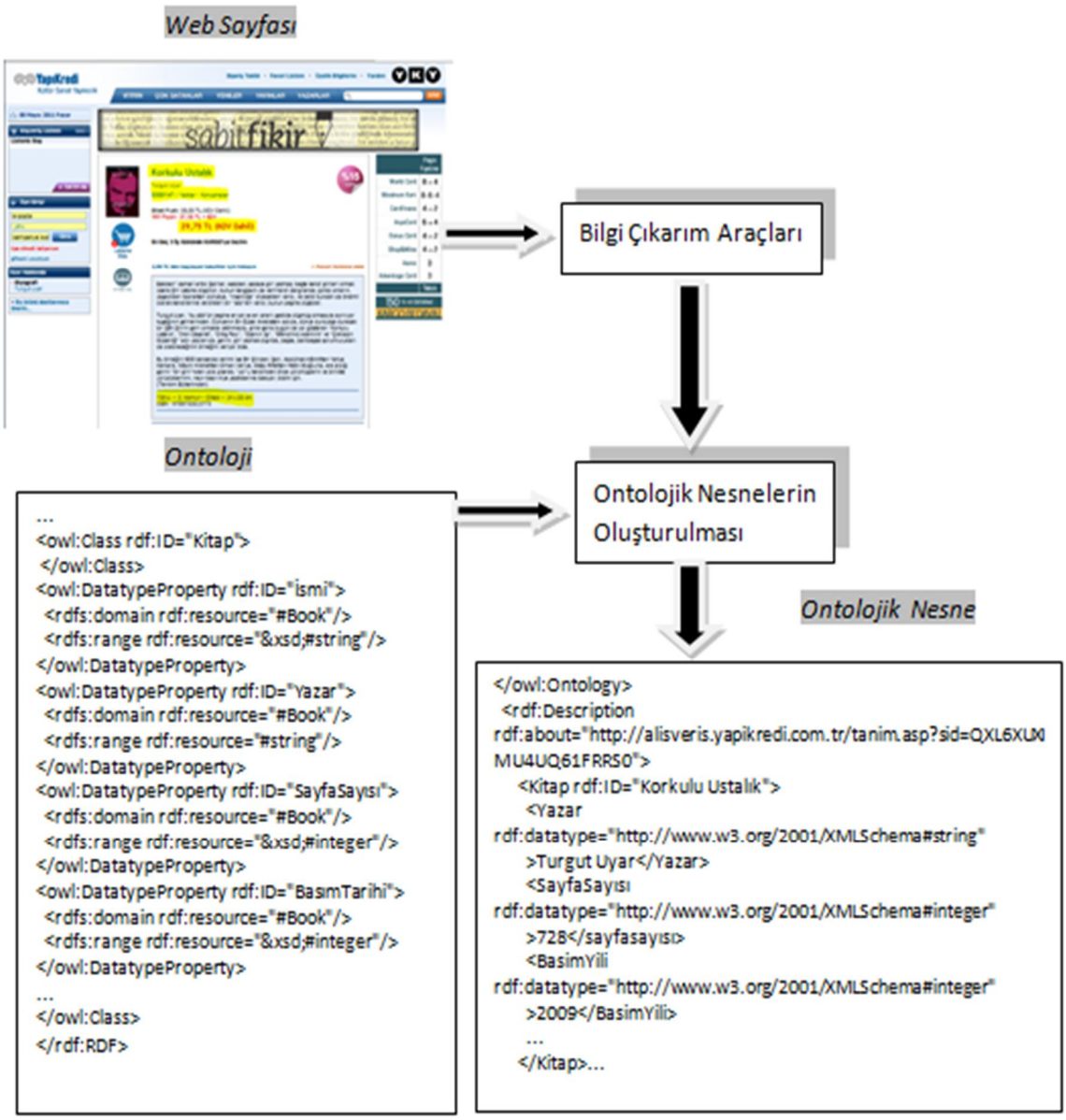
$$P = \left\langle p_1 , p_2 ,..., p_m \right\rangle \quad \textbf{(4.1)}$$

Oturumları belirlemek için tekil kullanıcılar tespit edilir. Tekil kullanıcılar, bir kullanıcının aynı IP ve istemci tarayıcı versiyonuna sahip olacağının varsayılmasıyla belirlenir. Bir oturum kullanıcı tarafından ilk isteğin gelmesiyle başlar ve belli bir eylemsizlik süresi sonra sona erer. Bu süre pek çok Web sunucusu için 20 dakikadır. Çizelge 4.1'de verilen erişim bilgilerinden, tekil kullanıcı bilgileri ve eylemsizlik süresi göz önüne alınarak oturumlar belirlenir. Bu aşamanın sonunda Web sayfalarından oluşan bir oturum kümesi *S* elde edilir. *S* kümesindeki *n* boyutlu S_i oturumu **(4.2)**'de belirtilmiştir.

$$S_i = \left\langle p_1^i, p_2^i,..., p_n^i \right\rangle \quad \text{Öyle ki } p_k^n \in P, \text{ ve } 1 \le k \le n . \quad \textbf{(4.2)}$$

Her bir Web sayfasında tek bir ürün bulunduğu varsayılmıştır, bu nedenle birden çok ürünün bulunduğu ortak sayfalar göz ardı edilir. Web sayfalarını ontolojiye yansıtmak için öncelikle sayfalarda bulunan ürünün ontolojisi öğrenilmelidir.

Ontoloji, doğal dil işleme teknikleriyle öğrenilebilir veya daha önce oluşturulmuş ve kabul görmüş olan aynı içeriğe sahip bir ontoloji kullanılabilir. Bu çalışmada kullanılan ontoloji elle oluşturulmuştur çünkü küçük ve statik Web siteleri için ontolojinin el ile oluşturulmasının daha elverişli olduğu daha önceki çalışmalarda belirtilmiştir [6]. *P* kümesinde bulunan her Web sayfasındaki ürüne ait kullanılabilir bilgiler, manuel geliştirilen veya mevcut bilgi çıkarımı araçlarıyla [28,29] alınır ve bu bilgilerle ontolojiye ait bireyler oluşturulur. Buradaki kullanılabilir bilgiler ontolojideki sınıflara ait niteliklerdir. Web sayfalarının, ontojideki sınıflara ait nesnelere yansıtılması Şekil 4.1'de gösterilmiştir.



Şekil 4.1: Web Sayfalarının Ontoloji Bireylerine Yansıtılması.

Ontolojideki sınıfa ait nesneler belirlendikten sonra (4.2)'deki örnek oturum *Si* Web sayfaları yerine, bu sayfaların yansıtıldığı ontolojideki sınıflara ait nesneleri içerecek şekle dönüştürülür. "*O*", Şekil 1.1'de belirtilen *C1* sınıfının nesneler kümesi olmak üzere *n* elemanlı *Si* oturumu **(4.3)** ile temsil edilir.

$$S_i = \left\langle o_1^i, o_2^i, ..., o_n^i \right\rangle \text{ Öyle ki } o_k^n \in O \text{ ve } 1 \le k \le n \qquad \textbf{(4.3)}$$

Öneri aşamasında hangi sayfanın hangi nesneye yansıtıldığı bilgisi gerekeceğinden sayfalar ve yansıtıldıkları nesneleri kayıt eden bir tablo bu aşamada oluşturulur.

4.2 Oturum Genişletme

Bölüm 4.1'de oturumlar, ontolojideki sınıfın nesnelerinden oluşacak şekilde elde edilmiştir. Bu sınıfın niteliklerinden bazıları da kendisine ait nitelikleri olan bir sınıf olabilir. Bu tarz sınıfları altsınıf olarak adlandıracağız. Altsınıfların kendilerine ait nitelikleri vardır ve başka bir sınıfın niteliğidirler. Bölüm 4.1'de elde edilen oturumlarda bu alt sınıflar sadece isimlerinin olduğu bir nitelik olarak temsil edilmektedir. Oturum genişletme aşamasında bu alt sınıflara ait etki alanı ontolojisi ve ontolojik nesneleri oluşturulur ve oturumlar hem hedef sınıf nesnesini hem de alt sınıf nesnesini kapsayacak şekilde genişletilir. Bu aşamada da ontoloji el ile oluşturulmuş ve ontolojideki sınıfa ait nesneler Şekil 4.2'de belirtildiği gibi bilgi çıkarım araçlarıyla edinilmiştir. Bu aşamanın sonunda oluşan örnek oturum *Si*, **(4.4)**'de verilmiştir. *SO*, Şekil 1.1'de belirtilen *C2* sınıfına ait nesneler kümesidir.

$$S_i = \left\langle \left\{o_1^i, so_1^i\right\}, \left\{o_2^i, so_2^i\right\}, ..., \left\{o_m^i, so_n^i\right\} \right\rangle \text{ Öyle ki } o_k^n \in O \text{ ve } so_k^n \in SO \qquad \textbf{(4.4)}$$

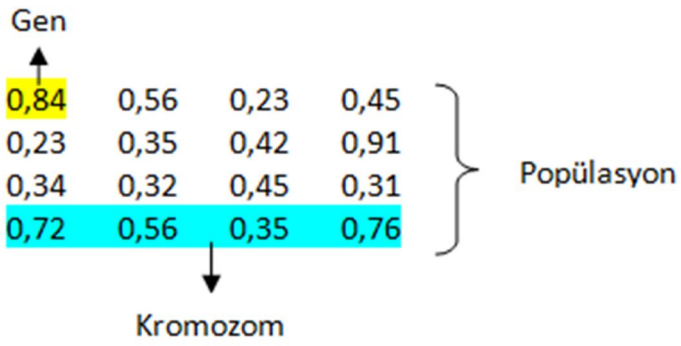
Buradaki temel nokta **(4.3)**'de verilen oturumun sadece *C1* sınıfına ait nesnelerden oluşurken, **(4.4)**'de verilen oturumun hem *C1* sınıfına ait nesnelerden hem de *C2* sınıfına ait nesnelerden oluşmasıdır. Gerçek bir veri kümesinde *C1* sınıfı kitap sınıfına, *C2* sınıfı yazar sınıfına karşı gelebilir. Bu veri kümesinden oluşan oturumlar için **(4.3)**'de sadece kitap sınıfının nesneleri varken, **(4.4)**'de hem kitap sınıfının hem de yazar sınıfının nesneleri vardır. **(4.3)**'de yazar sadece isimden oluşan bir nitelik iken, **(4.4)**'de kendi nitelikleri olan (yaşı, ödülleri vb.) bir altsınıftır.

4.3 Katsayıları Belirleme

Benzerlik hesaplamalarında sınıflara ait nitelikler katsayılar atanarak ağırlıklandırılmıştır. Katsayıları belirlemek için 2 farklı yöntem kullanılmıştır: genetik algoritma temelli yöntem ve İF temelli yöntem. Her iki yöntemde de katsayılar 0,0 ile 1,0 arasında değiştiğinden uzaklık veya benzerlik hiçbir zaman 1,0 üzerine çıkamamıştır. Katsayıların atanmasındaki amaç önemsiz olarak tespit edilen niteliklere 0,0'a yakın katsayılar atanarak benzerlik hesaplamalarındaki etkilerinin azaltılmasıdır.

4.3.1 Genetik algoritma temelli yöntem

Niteliklerin ağırlıklandırılmasında kullanılan yöntemlerden birisi genetik algoritma temelli yöntemdir. Bu çalışmadaki her gen, katsayı atanması gereken bir niteliktir ve her kromozom nitelik sayısı kadar gen içerdiği için niteliklere optimum değer atama probleminin bir çözümüdür. Kromozomlar, rastgele değer atanmış genlerden oluşan popülasyonlar üretilerek elde edilir. Genlere atanan rastgele değerler 0,0 ile 1,0 arasındadır. Bu yöntemde kullanıcının seçim yapmasında önemli olan niteliklere yüksek katsayı değeri atanması amaçlanmıştır. Şekil 4.2'de gen, kromozom ve popülasyon örneği verilmiştir.



Şekil 4.2: Gen, Kromozom, Popülasyon.

Kromozomlara ait uygunluk değeri, kromozomdaki genlerin katsayı olarak kullanılmasıyla gerçekleştirilmiş başarılı öneri sayısının, toplam öneri sayısına oranıdır.

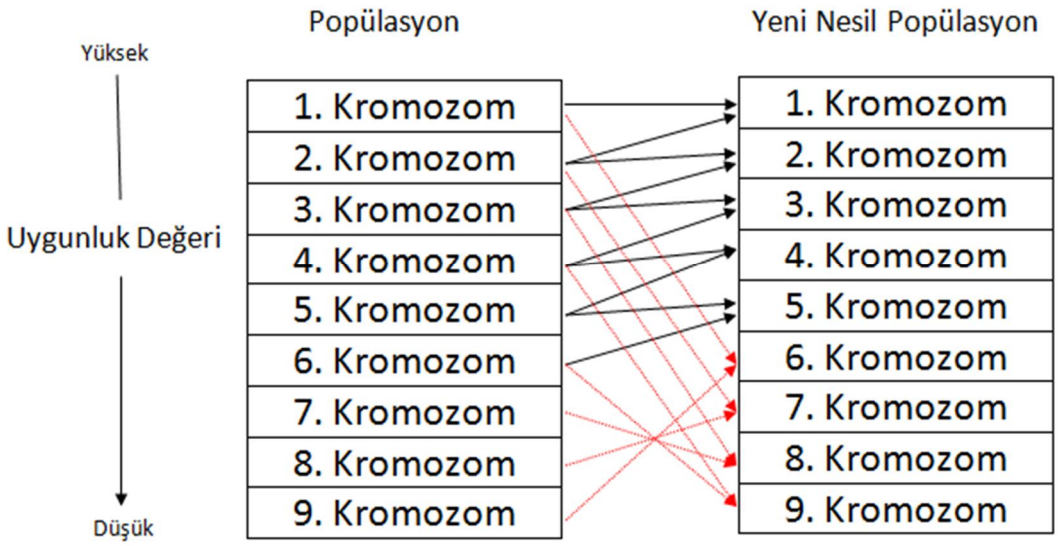
Hangi kromozomların çaprazlanacağını belirlemek için (Seleksiyon) [20]:

- Tüm kromozomlar uygunluk değerlerine göre sıralanırlar.

- *i.* ve *(i+1).* kromozomları arasında *i<(P/2)+1* olduğu sürece çaprazlama uygulanır. *P*, popülasyondaki kromozom sayısıdır.

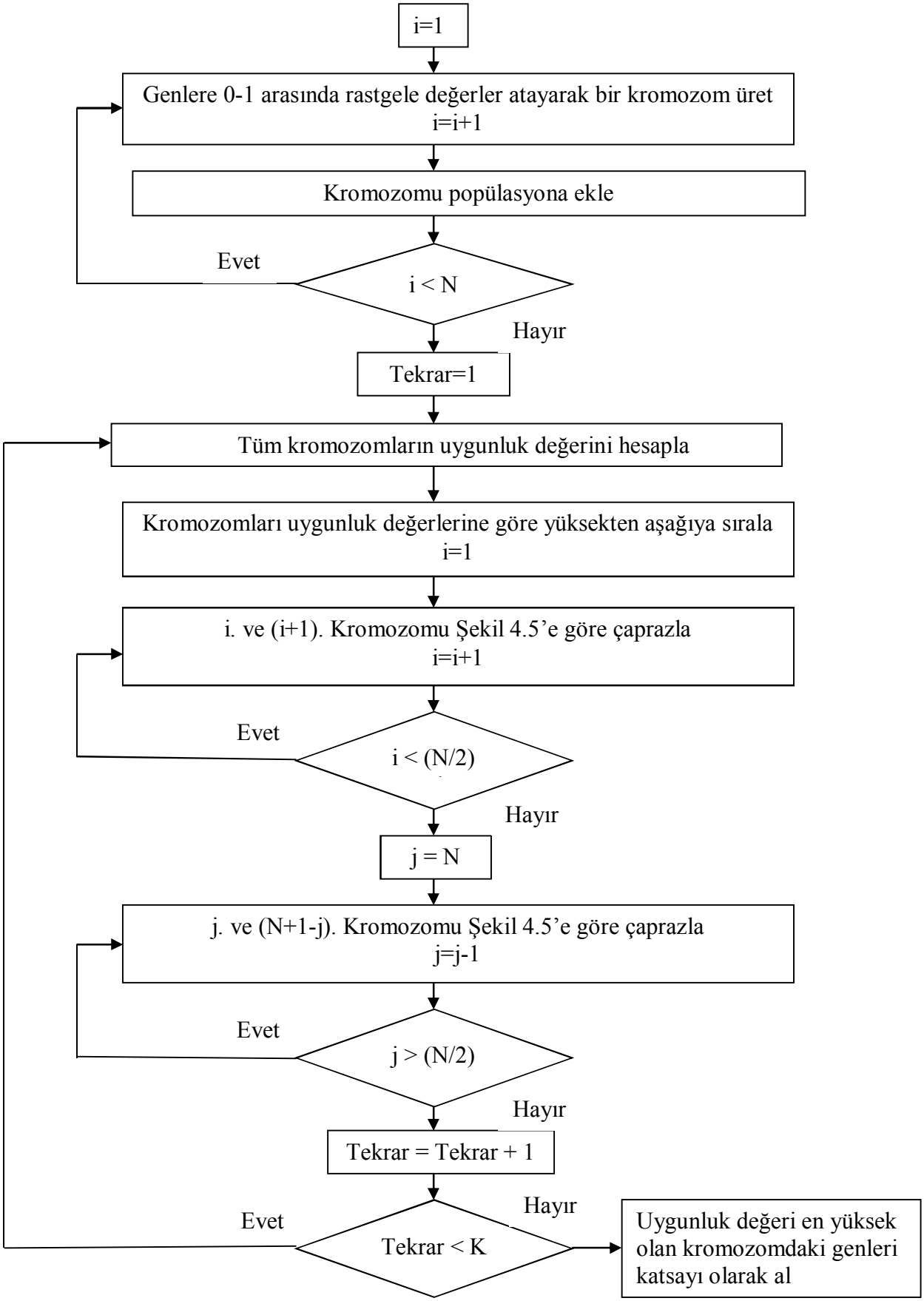
- Diğer çaprazlamalar *j.* ve *(p+1-j).* kromozomlarına uygulanır.

Böylece çaprazlamaların yarısı, uygunluk değeri en yüksek olan kromozomların birbiri ile gerçekleşir. Çaprazlamaların diğer yarısı ise uygunluk değeri en yüksek olan kromozomlar ile en düşük olan kromozomlar arasında gerçekleştirilir. Bu çaprazlamalarda, uygunluk değeri bir ebeveynin diğerine göre daha yüksek olduğundan, yeni nesil kromozomun uygunluk değerinin en az bir ebeveyninkinden yüksek olması beklenir. Şekil 4.3'de seleksiyon gösterilmiştir.

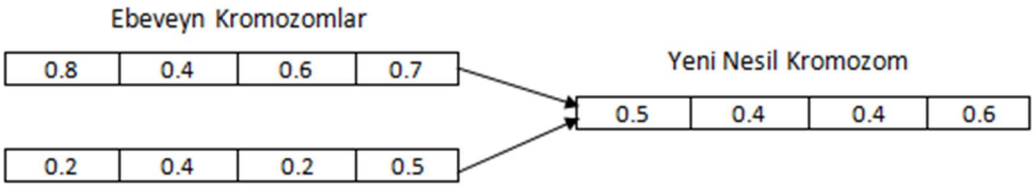


Şekil 4.3: Seleksiyon.

Çaprazlamalar, klasik genetik algoritmalardaki gen alışverişi şeklinde değil, bir çeşit mutasyon ile birleştirilmiş şekilde, ebeveynlerin aritmetik ortalaması alınarak gerçeklenmiştir. Ebeveynlerdeki genlerin aritmetik ortalaması oğul genlere aynı sırada olacak şekilde atanmıştır. Böylece yeni nesil kromozomlardaki genler de 0,0 ile 1,0 arasında olacak şekilde oluşurlar. Çaprazlama oranı 1,0 olarak alınmış ve tüm kromozomlar çaprazlanmıştır. *N* boyutlu bir popülasyonun *K* kadar çaprazlanması ile katsayıların belirlenmesi yöntemine ilişkin akış diyagramı Şekil 4.4'de, örnek bir çaprazlama ve mutasyon ise Şekil 4.5'de verilmiştir.



Şekil 4.4: GA ile Katsayı Belirleme İşlemi Akış Diyagramı.



Şekil 4.5: Çaprazlama ve Mutasyon.

Bu çalışmada çaprazlama sayısı önceden belirlenmiş bir sayı *(K)* kadar tekrarlanmıştır ancak çaprazlama bir uygunluk eşik değerine ulaşılıncaya kadar da tekrarlanabilir. Farklı *N* (popülasyondaki kromozom sayısı) değerleri için sonuçlar üretilmiştir. Sonuçta en yüksek uygunluk değerine sahip kromozomdaki genler niteliklere ait katsayılar olarak alınmıştır.

4.3.2 İşbirlikçi filtreleme temelli yöntem

Kullanıcıların elektronik ticaret Web sitelerindeki gezintileri, ürün sınıfının nitelikleri ile değişmektedir [16]. Örneğin kitap ticareti yapan bir site için, bazı kullanıcılar sadece kitabın konusuna dikkat ederek gezinti yapmaktadır ya da bazı kullanıcılar sadece yeni kitaplar ile ilgilenebileceğinden, site üzerinde sadece yeni kitaplara ait içeriğe sahip olan sayfaları ziyaret etmektedir. İF temelli yöntemin amacı, kullanıcıların, bir ürünü hangi nitelikleri için tercih ettiğini bulmak ve bu nitelikleri önemli nitelik, diğerlerini ise önemsiz nitelik olarak belirleyebilmektir. Önemsiz olan nitelikler belirlendikten sonra, bu niteliklere 1,0'dan düşük bir katsayı atanarak benzerlik hesabındaki etkileri azaltılır; önemli niteliklere katsayı atanmaz, 1,0 olarak bırakılır. Önemli ve önemsiz niteliklerin belirlenmesi için:

- Oturumlardaki nesnelerin her bir niteliği için bir vektör üretilir. Örneğin nesneleri Çizelge 4.3'de belirtilen ve **(4.4)**'de verilen *S1* oturumu için **(4.5)**'deki gibi 3 tane vektör üretilir

Çizelge 4.3 : S1 Oturumundaki Nesneler ve Nitelikleri.

Nesneler	**Nitelikler**		
	Attribute1	**Attribute2**	**Attribute3**
Obj1	A	X	M
Obj2	A	Y	N
Obj3	A	W	K
Obj4	A	W	L

$S1 = \{Obj1, Obj2, Obj3, Obj4\}$ **(4.4)**

$V1 = <A, A, A, A>$ → Attribute1 için. **(4.5)**

$V2 = <X, Y, W, W>$ → Attribute 2 için.

$V3 = <M, N, K, L>$ → Attribute3 için.

- Vektörlerdeki saflığı bulmak için bir saflık fonksiyonu kullanılır. Saflık, vektördeki elemanların ne kadar farklı ya da ne kadar aynı olduğunun belirlenmesidir. Vektördeki elemanların hepsi aynı ise saflık maksimum; hepsi farklı ise saflık minimumdur. Bu çalışmada kullanılan saflık fonksiyonu, vektördeki tüm elemanları birbiri ile karşılaştırır ve bu karşılaştırmalar sonucu aynı olduğu belirlenen eleman çifti sayısının toplam karşılaştırma sayısına oranını saflık değeri olarak alır. Örneğin **(4.6)**'da verilen *n* boyutlu *V* vektörü için saflık değeri **(4.7)**'deki gibi hesaplanır.

V = <a1, a2, a3,… a_n> **(4.6)**

$$Saflik(V) = \sum_{i=1}^{n} \sum_{j=1}^{i} f(a_i, a_j) \Big/ n * \frac{n+1}{2} \qquad \textbf{(4.7)}$$

Formül (4.7)'deki payda kısmı, n elemanlı bir vektördeki toplam karşılaştırma sayısıdır. Pay kısmındaki $f(a_i, a_j)$, fonksiyonu ise a_i ve a_j değerlerinin aynı olup olmadığını kontrol eden (4.8)'de verilmiş basit bir karşılaştırma fonksiyonudur.

$$f(a_i, a_j) = \begin{cases} 1 & \text{Eğer } a_i = a_j \\ 0 & \text{Eğer } a_i \neq a_j \end{cases} \qquad \textbf{(4.8)}$$

- Rastgele üretilmiş oturumlardan oluşan bir oturum kümesi elde edilir. Bu oturumların içeriği Bölüm 4.1'de elde edilmiş oturum kümesindeki oturumlarda bulunan elamanlar arasından rastgele seçilir. Rastgele üretilen bu oturum kümesindeki oturumların boyutları Bölüm 4.1'de elde edilmiş oturum kümesindeki oturumların boyutları ile aynıdır. Örneğin Bölüm 4.1'de boyutu *K* olan toplam *N* adet oturum varsa, rastgele yaratılan oturum kümesinde de boyutu *K* olan *N* tane oturum yaratılacaktır. Böylece iki faklı oturum kümesi elde edilmiş olur: Bölüm 4.1'de Web sunucusu erişim kütüğünden elde edilen oturum kümesi ve rastgele üretilmiş oturum kümesi.

- Ontolojideki sınıfların her niteliği için hem erişim kütüğünden elde edilmiş oturum kümesindeki hem de rastgele üretilmiş oturum kümesindeki oturumların ortalama saflık değeri (OSD), formül **(4.7)**'ye göre hesaplanır. Sınıftaki niteliklerin katsayısı OSD'leri kullanılarak formül **(4.9)**'deki gibi belirlenir.

  X=Erişim Kütüğü Dosyalarından Üretilen Oturumların OSD **(4.9)**

  Y= Rastgele Üretilen Oturumların OSD

  Oran= X/Y

  K = Önceden Belirlenmiş Bir Eşik Değeri

  Katsayı = 1 Eğer Oran > K

  = Maksimum(Oran, 0,9) Eğer Oran < K

Bir niteliğin kullanıcı tercihini ne kadar etkilediği **(4.9)**'daki *Oran* değeri ile belirlenir, bu değer belli bir eşik değerinden (*K*) büyükse nitelik, önemli nitelik olarak kabul edilir ve katsayısı 1,0 olarak alınır. Eşik değerinden küçükse bu nitelik önemsiz bir nitelik olarak kabul edilir ve katsayı olarak tespit edilen *Oran* değeri alınır, bu değerin 0,9 ile eşik değeri (*K*) arasında olması durumunda katsayı 0,9 olarak alınacaktır.

4.4 Kümeleme ve Benzerlik Hesaplamaları

Bölüm 4.2'de elde edilen yapı normalize edilmiş veritabanı sistemine benzerdir. İlişkisel bir veritabanı şeması, anlamsal olarak ve dış anahtarlarla birbirine bağlanmış birden çok tablodan oluşur ve bir ontoloji örneğidir [6]. Bu nedenle ilişkisel veri tabanı yöntemleri Bölüm 4.2'de elde edilen yapı için uygundur.

Bu yapının ilk problemi benzerlik veya uzaklık ölçümü sırasında altsınıfların ve onların niteliklerinin nasıl ele alınacağının belirlenmesidir. Bu çalışmada kosinüs benzerliğini kullanan 2 farklı benzerlik hesaplama ve Öklid uzaklığını **(3.1)** ve **(3.2)**'ye göre kullanan 3 farklı uzaklık hesaplama gerçeklenmiştir.

Birinci Kosinüs Benzerliği (KB) hesaplamada, sadece hedef sınıfın nitelikleri girdi olarak kabul edilmiştir. İkinci kosinüs benzerliği hesaplamada ise hem hedef sınıfın hem de altsınıfın nitelikleri girdi olarak kabul edilmiştir. Örneğin kitap benzerliğinin hesaplanmasında yazar sınıfının nitelikleri de girdi olarak alınmıştır. Bu hesaplama "Genişletilmiş Kosinüs Benzerliği" (GKB) olarak adlandırılmıştır.

Birinci Öklid Uzaklığı (ÖU) hesaplamasında sadece hedef sınıfın nitelikleri girdi olarak alınmıştır. İkinci ve üçüncü Öklid uzaklığı hesaplamalarında ise altsınıfın nitelikleri de girdi olarak alınmıştır. Bu nedenle alt sınıfa ait nesneler bölüm 4.2 de elde edilmiştir. İkinci ve üçüncü Öklid uzaklık hesaplamalarının farkı, ikincisinin **(3.1)**'e göre; üçüncünün ise **(3.2)**'ye göre uzaklığı hesaplamasıdır. Bu çalışmada ikinci hesaplama "Genişletilmiş Öklid Uzaklığı-1" (GÖU-1) ; üçüncü hesaplama "Genişletilmiş Öklid Uzaklığı-2" (GÖU-2) olarak adlandırılmıştır.

Tüm nitelikler normalize edildiğinden uzaklık değerleri 0,0 ile 1,0 arasında değişmektedir. Bu nedenle *O1* ve *O2* nesneleri arasındaki uzaklık aşağıdaki gibi benzerliğe dönüştürebilir.

$$Benzerlik(O_1, O_2) = 1 - Uzaklik(O_1, O_2) \quad \textbf{(4.9)}$$

İlişkisel yapılardaki bir diğer problem ise çoklu değerli niteliklerdir. Sınıflara ait bazı nitelikler çoklu değerli olabilir. Örneğin bir film birden fazla aktöre sahiptir. Genellikle bu problemin çözümü için çoklu değerli nitelikleri tek bir değere indirgeyen birleştirme fonksiyonları kullanılır [22]. Bu çalışmada kullanılan KB, GKB, ÖU, GÖU-1 ve GÖU-2 yöntemleri, çoklu değerli nitelikler üzerinde çalışabilecek şekilde geliştirilmiştir. Bu yöntemler iki nesnenin çoklu değerli nitelikleri arasındaki benzerliği, nesnelerin ilgili nitelik değer kümelerinin kesişimindeki eleman sayısının birleşimindeki eleman sayısına oranı olarak kabul eder. O_1 ve O_2 nesneleri için f(O_1), O_1 nesnesinin çoklu değerli bir niteliğinin değer kümesine ait eleman sayısı ise bu nesnelerin ilgili niteliklerinin benzerliği **(4.10)**'daki gibi hesaplanır.

$$S(O_1, O_2) = f(O_1 \cap O_2) / \left[f(O_1 \cap O_2) + f(O_1 / O_2) + f(O_2 / O_1) \right] \quad \textbf{(4.10)}$$

Sayısal çoklu değerli niteliklerde benzerlik hesabı, O_1 nesnesinin değer kümesindeki her değerin, O_2 nesnesinin değer kümesindeki her değer ile benzerliklerinin bulunması ve bunların aritmetik ortalamasının hesaplanması ile sağlanır. Örneğin A,B,C,X,Y reel sayılar olmak üzere $f(O_1)$ ve $f(O_2)$, **(4.11)**'de verildiği gibiyse bu niteliğe ait benzerlik **(4.12)**'deki gibi hesaplanır.

$$f(O_1) = \{ \text{A, B, C} \} \text{ ve } f(O_2) = \{ \text{X,Y} \} \quad \textbf{(4.11)}$$

$$S(O_1, O_2) = (\text{S(A,X)} + \text{S(A,Y)} + \text{S(B,X)} + \text{S(B,Y)} + \text{S(C,X)} + \text{S(C,Y)}) / 6 \quad \textbf{(4.12)}$$

KB ve GKB yöntemlerinde kosinüs benzerliği kullanılmaktadır. Kosinüs benzerliğinde O_1^i, O_1 nesnesinin *i.* niteliği olduğu düşünülürse O_1 ve O_2 nesneleri arasındaki benzerlik aşağıdaki gibi hesaplanacaktır.

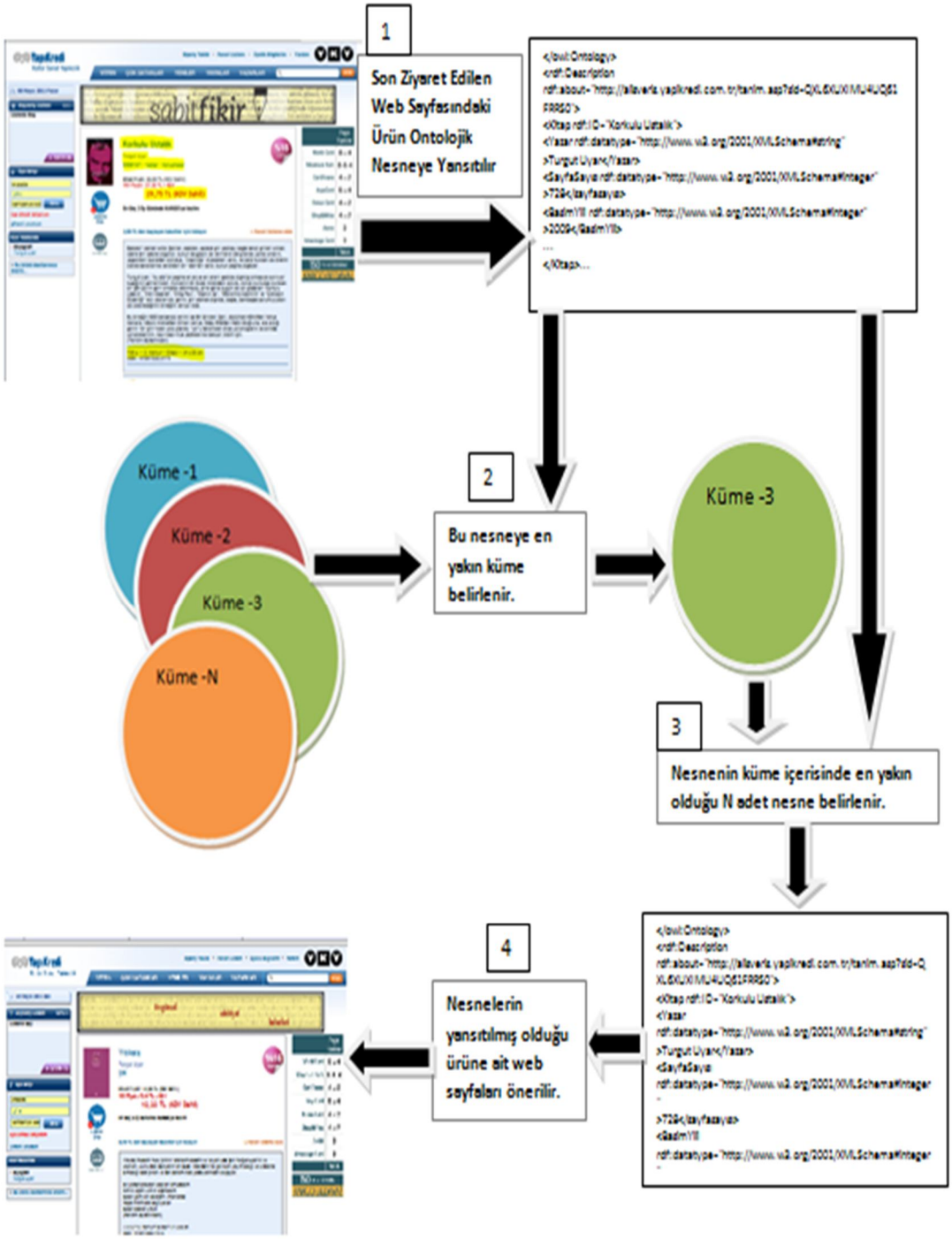
$$S(O_1, O_2) = \frac{\sum_{i=1}^{n} O_1^i * O_2^i}{\sqrt{\sum_{i=1}^{n} (O_1^i)^2} + \sqrt{\sum_{i=1}^{n} (O_2^i)^2}} \quad \textbf{(4.13)}$$

O_1 ve O_2 nesneleri için *i*. nitelik çoklu değerli ise, nesnelerin bu niteliğinin değer kümeleri **(4.10)** ve **(4.12)**'deki formüllerde kullanılarak bir *X* değeri elde edilir ve **(4.13)**'deki fonksiyonda O_1^i değeri *1*; O_2^i değeri *X* olarak alınır.

Benzerlik hesaplama yöntemi, çoklu değerli nitelikler üzerinde hesap yapabilecek şekilde uyarlandığından, k-means gibi uzaklık temelli geleneksel kümeleme teknikleri veri üzerinde kullanılabilir. Kümeleme ile arama uzayının daraltılması hedeflenmiştir. Kümeleme aşamasındaki benzerlik hesaplamalarında niteliklere katsayılar atanarak ağırlıklandırılmıştır. Katsayıların belirlenmesi Bölüm 4.3'de açıklanmıştır.

4.5 Öneride Bulunma

Öneri aşamasında, aktif kullanıcı tarafından son ziyaret edilmiş sayfa dikkate alınır ve bu sayfanın etki alanı ontolojisine yansıtılmış nesnesi belirlenir. Sonraki aşamada bu nesnenin küme merkezlerine olan uzaklığı hesaplanarak en yakın küme merkezi belirlenir ve en yakın kümedeki nesnelere olan uzaklık hesaplanır. Kümedeki nesnelerden aktif oturumdaki nesneye en yakın olan N tanesinin karşı geldiği Web sayfaları kullanıcıya öneri olarak sunulur. Hangi Web sayfasının hangi nesneye yansıtıldığı Bölüm 4.1'de belirlenmiş ve kayıt edilmiştir. Bu işlemler Şekil 4.5'te görselleştirilerek verilmiştir.



Şekil 4.6: Öneride Bulunma.

5. DENEYSEL SONUÇLAR

Bölüm 4'de detayları açıklanmış olan ürün öneri sistemi, Yapı Kredi Yayınları'na ait Web sitesi[1] için test edilmiştir. Bu Web sitesi üzerinden kitapları aramak, görüntülemek ve satın almak mümkündür. Çalışmada kullanılan erişim kütüğü dosyaları, 9 Aralık 2010 ile 2 Şubat 2011 tarihleri arasına aittir. Bu bölümde ürün öneri sisteminin Yapı Kredi Yayınları'nın Web sitesine uygulanması ve performansının değerlendirilmesi ile ilgili detaylar açıklanmıştır.

5.1 Ürün Öneri Sisteminin Uygulanması

Web sitesinde bulunan ürün (kitap) için ontoloji elle oluşturulmuştur. Oluşturulan kitap ontolojisinde iki adet sınıf bulunmaktadır: kitap ve yazar. Yayınevi, tüm kitaplar için aynı olduğundan, yayınevi sınıfı ontolojiye dahil edilmemiştir. Kitap sınıfı, nesnelerinin önerileceği hedef sınıftır. Yazar sınıfı ise hem kendi nitelikleri olan bir sınıf hem de hedef sınıfın bir niteliği olduğu için altsınıftır. Çizelge 5.1'de belirtilen kitap sınıfının nitelikleri aşağıda açıklanmıştır:

- Alan: Kitabın fiziksel ölçülerini temsil eden çift duyarlı kayan noktalı sayı tipinde bir niteliktir; kitabın en ve boyunun çarpımına karşı gelmektedir.
- Basım Yılı: Kitabın son basım yılını belirten tam sayı tipinde bir niteliktir.
- Cilt Tipi: Kitabın ciltli veya ciltsiz olduğunu belirten boole tipinde bir niteliktir.
- Fiyat: Kitabın fiyatını belirten çift duyarlı kayan noktalı sayı tipinde bir niteliktir.
- Kalite: Kitabın kağıt kalitesini belirten çift duyarlı kayan noktalı sayı tipinde bir niteliktir. Kuşe kağıt için 1,00; 1. hamur kağıt için 0,66; 2. Hamur kağıt için 0,33 ve 3. Hamur kağıt 0.00 olarak kabul edilmiştir.

[1] http://alisveris.yapikredi.com.tr

- Kategori: Kitabın konusunu belirten karakter katarı tipinde bir niteliktir.
- Yazar: Kitabın yazarını belirten sınıf tipinde bir değişkendir.
- Yeni Yayın: Kitabın 2011 yılında baskısının olup olmadığını belirten boole veri tipinde bir niteliktir.

Çizelge 5.1: Kitap Sınıfı Nitelikleri.

Kitap
Alan : Çift Duyarlı Kayan Noktalı Sayı (Double)
Basım Yılı: Tam Sayı (Integer)
Cilt Tipi:Boole (Boolean)
Fiyat: Çift Duyarlı Kayan Noktalı Sayı (Double)
İsim: Karakter Katarı (String)
Kalite: Çift Duyarlı Kayan Noktalı Sayı (Double)
Kategori: Karakter Katarı (String)
Yazar: Sınıf (Class)
Yeni Yayın: Boole (Boolean)

Çizelge 5.2'de belirtilen yazar sınıfının nitelikleri aşağıda açıklanmıştır:

Doğum Tarihi: Yazarın doğum tarihini belirten tamsayı tipinde bir niteliktir. Aykırı noktaları engellemek için en az değeri 1900 olarak kabul edilmiş, doğum tarihi 1900'den önce olan yazarlar için bu niteliğe ait değer 1900 olarak alınmıştır.

Kitap Sayısı: Yazarın Yapı Kredi Yayınları'ndan yayınlanmış kitap sayısını belirten tam sayı tipinde bir değişkendir.

Kategoriler: Yazarın daha önce yazdığı kitapların kategorilerinden oluşan karakter katarı dizisi tipinde bir değişkendir.

Çizelge 5.2: Yazar Sınıfı Nitelikleri.

Yazar
İsim: Karakter Katarı (String)
Doğum Tarihi: Tam Sayı (Integer)
Kitap Sayısı: Tam Sayı (Integer)
Kategoriler: Karakter Katarı Dizisi (String Array)

Etki alanı ontolojisinin tanımlanmasından sonra Web sitesine ait sayfalarda bulunan ürünler etki alanı ontolojisine yansıtılmıştır. Bu ürünlere ait bilgilerin Web sayfaları üzerinden elde edilmesinde bilgi çıkarım araçları kullanılmıştır. Bilgi çıkarım araçları C# dilinde gerçeklenmiştir, bu araçlar ile Web sayfasının kaynak kodunun görüntülenmesi ve bu kod içerisinden gerekli bilginin düzenli ifadeler (regular expression) kullanılarak elde edilmesi hedeflenmiştir. Düzenli ifadeler, bilgi çıkarımı için uzun yıllardır kullanılan en pratik yöntemdir [30]. Web sayfalarına ait kaynak kod içerisinde bir defa tekrarlayan bir düzenli ifade tespit edilmiş ve bu düzenli ifade tanımlanarak sayfalardaki ürünlerle ilgili bilgiler elde edilmiştir. Örneğin, "Kitapla Direniş" kitabının bulunduğu sayfanın kaynak kodunun bir kısmı aşağıda verilmiştir:

<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1254">
<title>Yapı Kredi Yayınları - Kitapla Direniş - </title>
<META HTTP-EQUIV="Copyright" CONTENT="Copyright © Yapı Kredi Kültür Yayınları">

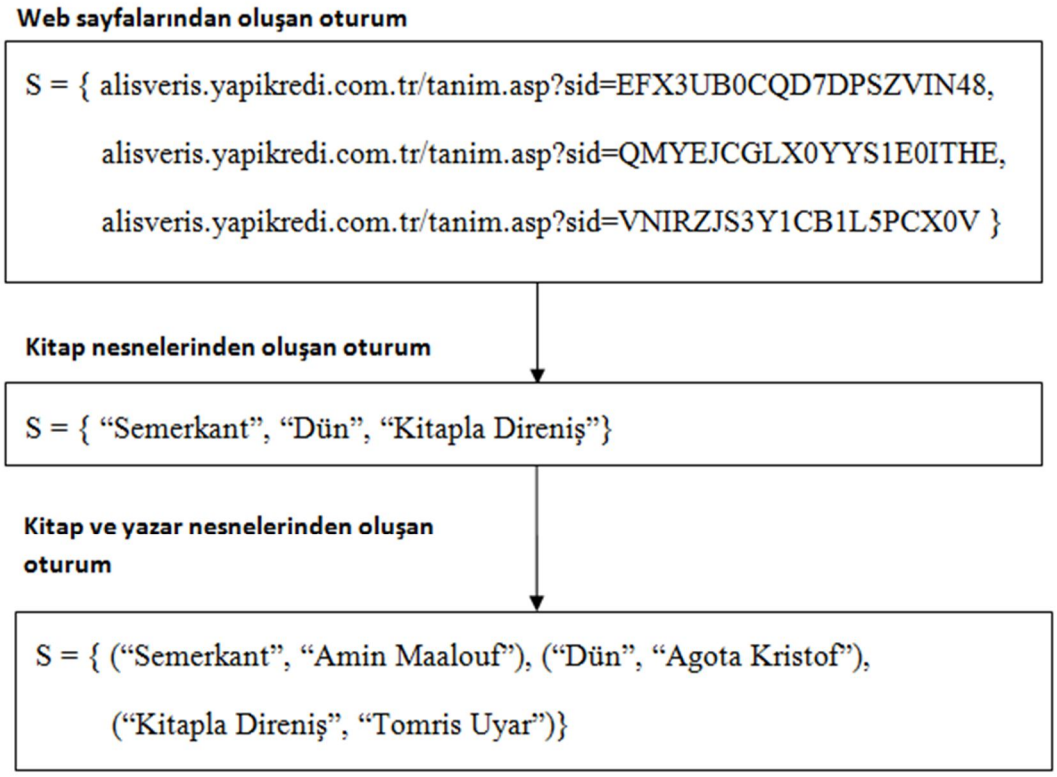
Bu kod içerisinde "*<title>Yapı Kredi Yayınları -"* karakter katarı ile başlayan tek bir satır bulunduğundan, bu karakter katarı ile *"- </title>*" karakter katarı arasındaki kelimelerden kitabın isim bilgisine ("*Kitapla Direniş")* ulaşılması için ""*<title>Yapı Kredi Yayınları –(.*?)- </title>*" şeklinde bir düzenli ifade kullanılabilir. Her nitelik için gerekli olan düzenli ifadeler, Web sayfalarının kaynak kodları incelenerek belirlenmiştir.

Hem yazar sınıfının hem de kitap sınıfının nesnelerini oluşturmak için gerekli olan bilgiler öncelikle Yapı Kredi Yayınları'na ait Web sitesinden elde edilmeye çalışılmıştır. Buradan erişilemeyen bilgilere, Kibo'ya ait olan Web sitesinden[2] ulaşılmaya çalışılmıştır. Her iki site üzerinde de bulunmayan bilgiler ise çeşitli kaynaklardan yararlanılarak bulunmuştur. Böylece 2222 adet kitap nesnesi ve 925 adet yazar nesnesi tanımlanmıştır.

Tanımlanan kitap nesnelerinin birbirleri ile olan uzaklıklarının belirlenmesinde, Bölüm 4.4'de detayları açıklanan KB, GKB, ÖU, GÖU-1 ve GÖU-2 yöntemleri kullanılmıştır. Bu yöntemlerin başarı testi sonuçları ve kümeleme ile ilgili ayrıntılar Bölüm 5.3.1'de açıklanmıştır. Niteliklere katsayı atama yöntemlerinin başarı sonuçları ise Bölüm 5.3.2'de açıklanmıştır.

Katsayıların belirlenmesi aşamasında ontolojideki nesnelerden oluşan oturumlar kullanılmaktadır. Oturumları belirleyebilmek için, Web sunucusu erişim kütüğü dosyaları LogParser programı [27] kullanılarak Bölüm 4.1'de belirtildiği gibi gereksiz kayıtlardan arındırılarak birleştirilmiştir. LogParser programında çalıştırılan komut Ek A.1'de verilmiştir. Ziyaret edilen sayfa birden fazla ürün içeren bir sayfa ise oturuma dahil edilmemiştir. Tekil kullanıcı tespiti istemci tarayıcı versiyonu ve IP adresine göre gerçeklenmiş ve eylemsizlik süresi 20 dakika seçilerek oturumlar belirlenmiştir. Önerilen sistemin başarısının belirlenmesinde boyutu birden fazla olan oturumlar gerekeceğinden sadece boyutu birden fazla olan oturumlar dikkate alınmıştır. 55 günlük erişim kütüğü dosyalarından toplam 4317 oturum elde edilmiştir, bunların 2791 tanesinin boyutu birden fazladır. Boyutu birden fazla olan oturumların ortalama boyutu ise 3,18'dir. Başlangıçta elde edilen oturumlar, Web sayfalarından oluşmaktadır, bu Web sayfalarındaki ürünlerin etki alanı ontolojisine yansıtılmış nesneleri önceden elde edildiğinden, oturumlar Web sayfaları yerine bu sayfalardaki ürünlerin karşı geldiği nesnelerden oluşacak şekilde düzenlenmiştir. Böylece oturumlar kitap sınıfına ait nesnelerden oluşturulmuştur. Bir sonraki aşamada oturumlar hem kitap sınıfına ait nesnelerden hem de yazar sınıfına ait nesnelerden oluşacak şekilde genişletilmiştir. Örnek bir *S* oturumunun genişletilmesi Şekil 5.1'de verilmiştir.

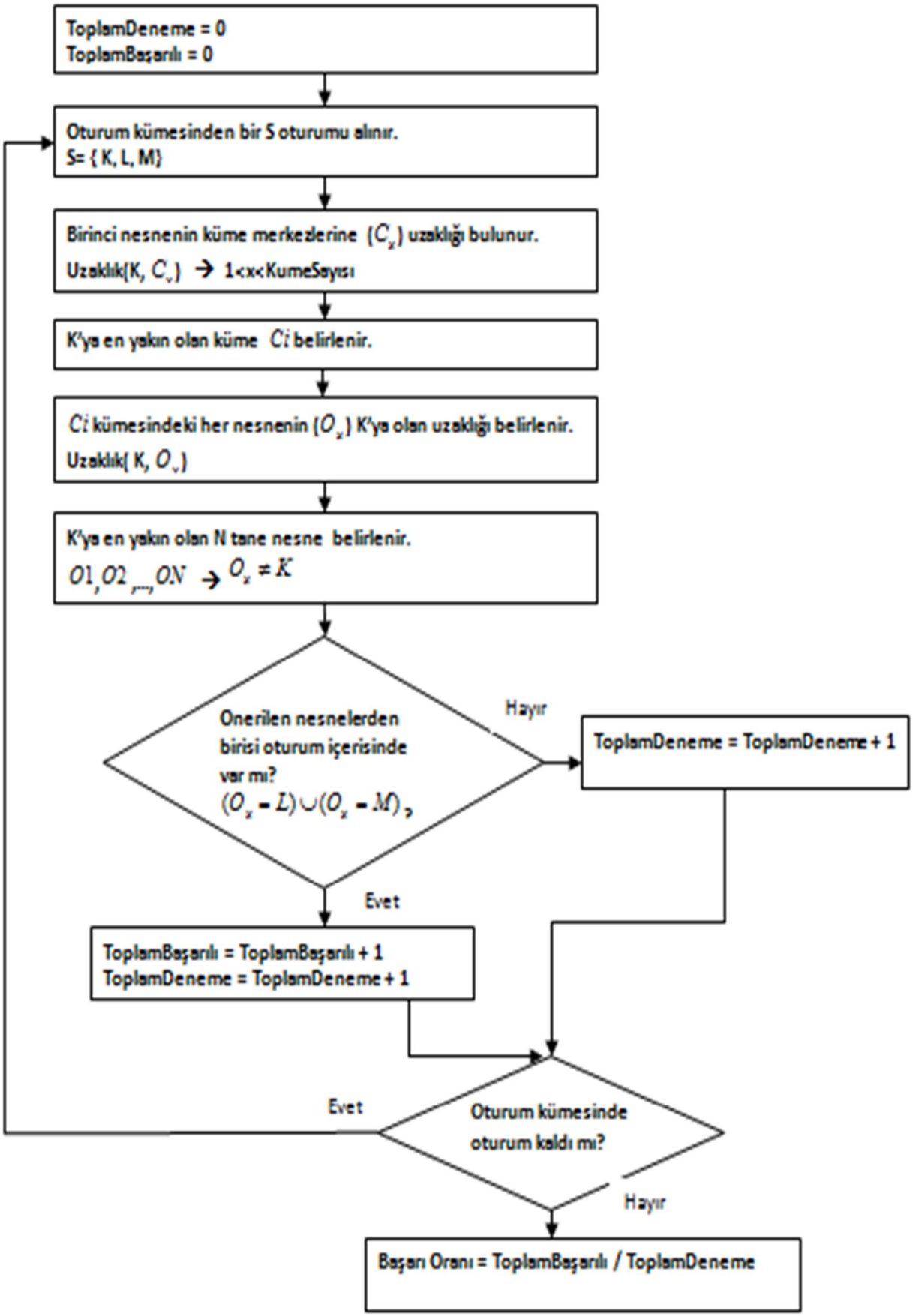
[2]http://www.kibo.com.tr



Şekil 5.1: Oturumun Genişletilmesi.

5.2 Değerlendirme Kriteri

Sistemin performansını değerlendirmek için gerekli kıstas, başarılı olan öneri sayısının toplam gerçekleştirilmiş öneri sayısına oranı olarak kabul edilmiştir. Öneriler, oturumların ilk nesneleri dikkate alınarak gerçeklenmiş ve önerilen ürünün oturum içerisinde bulunması durumunda başarılı sayılmıştır. Şekil 5.2'de başarı oranı belirlemeyi açıklayan bir akış diyagramı verilmiştir.

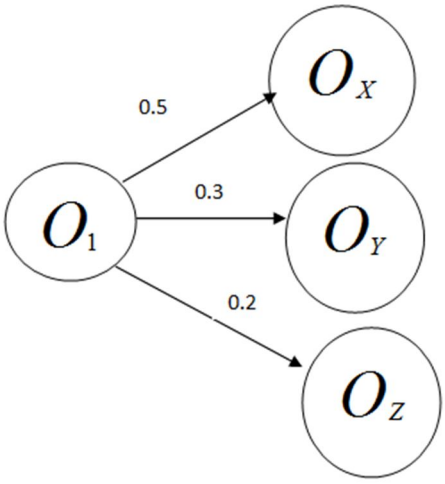


Şekil 5.2: Başarı Oranı Belirleme Akış Diyagramı.

5.3 Başarım Testleri

5.3.1Benzerlik hesaplama yöntemleri testleri

Bölüm 4'de detayları açıklanmış olan ontoloji tabanlı ilişkisel ürün öneri sisteminin başarı sonuçlarını, farklı bir yöntemin başarı sonuçları ile karşılaştırmak için Markov zinciri modeli kullanılmıştır. Karşılaştırma için bu modelin seçilmesinin nedeni, bir çok çalışmada bu modelin ürün öneriminde başarılı sonuçlar verdiğinin belirtilmesidir [31,32]. Markov zinciri modeli, kullanıcıların Web sunucularındaki erişim kütüklerine bıraktıkları hareketlerin benzerliğine dayalı bir öneri yöntemidir. Bu çalışmada birinci derece ve ikinci derece Markov zinciri modeli kullanılmıştır. Kullanıcının bir sonraki adımda ziyaret edeceği ürün, birinci derece Markov zinciri modelinde son ziyaret ettiği ürüne göre; ikinci derece Markov zinciri modelinde ise son ziyaret ettiği iki ürüne göre belirlenir. Her ürün için kendisinden sonra hangi ürünün ziyaret edilebileceği olasılığı, Web sunucusu erişim kütüğündeki kayıtlar kullanılarak belirlenir. Örneğin Şekil 5.3'de kullanıcıların O_1 ürününe ait Web sayfasından sonra hangi ürüne ait Web sayfasına geçebileceği oranlarıyla verilmiştir. Bu durumda O_1 ürününü ziyaret eden kullanıcıya tek bir ürün önerilecek ise O_x önerilir.



Şekil 5.3: Birinci Derece Markov Zinciri.

Markov modeli başarımı 10 kat çapraz doğrulama modeli ile gerçekleştirilmiştir. Bölüm 5.1'de elde edilen 2790 oturum, her biri 279 oturum içeren 10 oturum kümesine ayrılmıştır. Her defasında 9 küme öğrenme kümesi,1 küme test kümesi olarak kullanılarak 10 farklı test gerçekleştirilmiştir. Önerilen kitap sayısına göre 10 testin ortalama başarısı Çizelge 5.3'de verilmiştir. Testlerin başarı sonuçlarının standart sapması 1 kitap önerim testinde 2,34 ve 3 kitap önerim testinde 3,12'dir. Tüm test sonuçları Ek A.2'de verilmiştir.

Çizelge 5.3: Bir Dereceli Markov Zinciri Başarı Oranı.

Önerilen Kitap Sayısı	Başarı Oranı
1	%14,5
3	%36

İki dereceli Markov modelinin başarımı da 10 kat çapraz doğrulama modeli ile test edilmiştir. Boyutu 2 den fazla olan 1240 oturum, her biri 124 oturum içeren 10 farklı kümeye bölünmüş ve 9 küme öğrenme kümesi, 1 küme test kümesi alınarak 10 farklı test gerçeklenmiştir. Testlerin başarı sonuçları Çizelge 5.4'de verilmiştir. Testlerin başarı sonuçlarının standart sapması 1 kitap öneriminde 2,91 ve 3 kitap öneriminde 4,02'dir.

Çizelge 5.4: İki Dereceli Markov Zinciri Başarı Oranı.

Önerilen Kitap Sayısı	Başarı Oranı
1	%12,3
3	%31,8

İki dereceli markov zinciri modelinde başarının bir dereceli Markov zinciri modeline göre yaklaşık %15 azaldığı tespit edilmiştir. Yüksek dereceli Markov modelleri durum uzayını eksponansiyel olarak artırır ve aşırı büyük bir veri seti gerektirdikleri için öneri başarısının düşmesine neden olabilirler [33]. Web sayfası önerisi gibi hızlı öneri yapmak gereken durumlarda da yüksek dereceli Markov modeli uygun değildir [34].

Önerilen ontoloji tabanlı ilişkisel ürün öneri sistemi, KB ve ÖU yöntemlerini geliştirmiş ve detayları Bölüm 4.4'de verilen GKB, GÖU-1 ve GÖU-2 yöntemlerini kullanmıştır. GKB, girdi olarak hem hedef sınıfın hem alt sınıfın niteliklerini alan kosinüs benzerliği yöntemidir. GÖU-1 ve GÖU-2 yöntemleri de girdi olarak hem hedef sınıfın hem de alt sınıfın niteliklerini kabul eden Öklid uzaklığı temelli yöntemlerdir. Bu nedenle KB ve ÖU yöntemlerinde sadece kitap sınıfının nitelikleri (yazar adı, fiyatı, basım yılı, kategori, yeni basım ve alan) kullanılmıştır. GKB, GÖU-1 ve GÖU-2 yöntemlerinde ise hem kitap hem yazar sınıfının nitelikleri (yaşı, kitap sayısı, yazdığı kitap konuları) kullanılmıştır.

Nesneler arasındaki benzerlik hesaplanırken yazar sınıfının çoklu değerli kategoriler niteliğine ait benzerliği **(4.10)**'a göre hesaplanmıştır. Kitap sınıfının kategori niteliğinde benzerlik aynı kategoridekiler için 1,0; aynı kategori grubundakiler için 0,5; farklı kategori grubundakiler için 0,0 olarak kabul edilmiştir. Kategoriler ve kategori grupları Ek A.3'de verilmiştir. KB ve ÖU benzerliğinde yazar niteliği karakter katarı tipindedir ve aynı yazarlar için benzerlik 1,0; farklı yazarlar için benzerlik 0,0 olarak alınmıştır. GKB, GÖU-1, GÖU-2 yöntemlerinde ise yazar niteliği sınıf tipindedir.

KB ve ÖU yöntemleri, GKB, GÖU-1 ve GÖU-2 yöntemlerinin başarılarını karşılaştırmak için gerçeklenmiştir. Gerçekleştirilen ilk testte sadece bu yöntemlerin başarılarını karşılaştırmak hedeflendiğinden benzerlik hesaplamalarında niteliklere katsayı atanmamış ve kümeleme işlemi gerçeklenmemiştir. Öneride bulunulacak oturumdaki ilk kitabın sistemdeki tüm kitaplarla benzerliği hesaplanmış ve en benzer kitaplar önerilmiştir. Bu testte benzerlik hesaplamalarında katsayılar ağırlıklandırılmadığı için oturumlardan veya kitaplardan oluşan bir öğrenme kümesine ihtiyaç yoktur. Bu nedenle oturumların tamamı (2791 adet) için yöntemlerin başarısı test edilmiştir. Çizelge 5.5'de kümeleme ve nitelik ağırlıklandırılması yapılmadan gerçeklenmiş test sonuçları verilmiştir.

Çizelge 5.5: Benzerlik Hesaplama Yöntemleri Başarım Oranları.

Önerilen Kitap Sayısı	KB	GKB	ÖU	GÖU-1	GÖU-2
1	%7	%9.5	%12.5	%17	%15
3	%17	%26	%34	%44	%40

Test sonuçlarında görüldüğü gibi altsınıfın niteliklerini de benzerlik hesabına dahil etmek başarıyı artırmıştır. Örneğin KB'nin 3 kitap önerim testindeki başarısı sadece kitap sınıfının niteliklerini kullandığında %17 iken hem kitap hem yazar sınıfının niteliklerini kullandığında (GKB) %26 olmuştur. ÖU yönteminde de başarı %34'den, %44'e çıkmıştır. Veri kümesinde bir yazara 2,4 kitap düşmektedir. Bu nedenle kitaplar için sadece yazar adı iyi bir ayırt edici özellik olamayacağından yazar sınıfının niteliklerinin kullanılması başarıyı artırmıştır.

Gerçekleştirilen ikinci testte ise kitaplar kümelenmiş ve oturumdaki ilk nesneye en yakın küme tespit edildikten sonra bu kümedeki en yakın nesneler önerilmiştir. Benzerlik hesaplamaları geleneksel yöntemlerin (Kosinüs Benzerliği, Öklid Uzaklığı) ilişkisel veriler üzerinde işlem yapabilecek şekilde genişletilmiş formları ile gerçeklendiğinden benzerlik hesaplanmasında bir araç kullanılmamıştır. Benzerlik hesapları C# dilinde gerçeklenmiş bir program ile yapılarak benzerlik matrisi oluşturulmuştur. Kümeleme ise girdi olarak kümelenecek nesnelerin benzerlik matrisini kabul eden CLUTO yazılım aracı [35] içerisindeki *scluster* demetleme yöntemi ile gerçeklenmiştir. Kümeleme işlemi *scluster* yönteminin "agglo" anahtarı ile yığılmalı hiyerarşik yöntem kullanılarak gerçekleştirilmiştir.

Yığılmalı hiyerarşik kümeleme yöntemi aşağıdaki adımlardan oluşmaktadır [36]:

- Her nesne, G_1, G_2, ..., G_n ile ifade edilen ayrı bir küme olarak kabul edilir ve bu nesnelerin benzerlik matrisi hesaplanır.
- Benzerlik matrisinde n küme sayısı, $i = 1, 2, ..., n$, $j = 1, 2, ..., n$ ve $i \neq j$ olmak üzere tüm en az $D(G_i , G_j)$ uzaklığına sahip iki küme belirlenir ve bu iki küme birleştirilerek yeni bir küme oluşturulur.
- Yeni oluşan kümeler de dikkate alınarak benzerlik matrisi güncellenir.
- Yukarıdaki adımlar tek bir küme elde edilinceye (dendogramın köküne ulaşılınca kadar) tekrarlanır.

  Küme sayısının 5 olduğu durum için başarı test sonuçları Çizelge 5.6'da; 10 olduğu durum için başarı test sonuçları Çizelge 5.7'de verilmiştir. Bu testlerde de nitelikler ağırlıklandırılmadığı için bir öğrenme kümesine ihtiyaç duyulmadığından testler oturumların tamamı üzerinde gerçeklenmiştir.

Çizelge 5.6: 5 Küme İçin Benzerlik Hesaplama Yöntemleri Başarım Oranları.

Önerilen Kitap Sayısı	KB	GKB	ÖU	GÖU-1	GÖU-12
1	%5	%7	%9	%12	%11
3	%12	%18	%23	%32	%27

Çizelge 5.7: 10 Küme İçin Benzerlik Hesaplama Yöntemleri Başarım Oranları.

Önerilen Kitap Sayısı	KB	GKB	ÖU	GÖU-1	GÖU-12
1	%4	%6	%7.5	%11	%9
3	%10	%16	%20	%26	%23

Kitapların önce kümelendiği sonra en yakın kümenin belirlenip, bu küme içerisindeki en yakın kitapların önerildiği durumda: 5 küme için başarı yaklaşık %28 azalmış, arama uzayı %63 küçülmüştür; 10 küme için ise başarı yaklaşık %40 azalmış, arama uzayı %71 küçülmüştür.

Oturumlardaki en son ziyaret edilen Web sayfasında bulunan ürün dikkate alınarak ürün öneriminde bulunulduğundan benzerlik hesaplamaları çevrimdışı zamanda gerçekleştirilebilir. Bu durumda arama uzayının daraltılması çok önemli bir konu olmayabilir ancak oturumlardaki son *N* tane ürünün birleştirme fonksiyonu ile tek bir ürüne indirgendiği durumlarda benzerlik hesaplamaları çevrimiçi zamanda gerçekleneceğinden daha önemli olacaktır. Bu durumda kümelemeden kaynaklanan başarıdaki düşüş oranı, arama uzayının küçülme oranına göre çok daha az ve kabul edilebilir olduğu için ürünler kümelenerek arama uzayı daraltılabilir. Test veri seti için arama uzayı çok büyük olmadığından 5.3.2'de detayları belirtilen nitelik ağırlıklandırma testlerinde kümeleme yapılmamıştır.

5.3.2 Nitelik ağırlıklandırma testleri

GA ve İF temelli yöntem ile iki farklı şekilde nitelikler ağırlıklandırılmıştır. GA ile nitelikleri ağırlıklandırmada: 7 tanesi kitap sınıfının nitelikleri ve 4 tanesi yazar sınıfının nitelikleri için olmak üzere toplam 11 tane gen içeren kromozomlardan oluşan bir popülasyon oluşturulmuştur. Popülasyon, Bölüm 4.3.1'de açıklandığı gibi çaprazlama ve mutasyonlara tabi tutularak, popülasyondaki kromozomların uygunluk değerleri artırılmıştır. Uygunluk değeri, kromozomlardaki genlerin katsayı olarak kullanılması durumunda gerçekleşen öneri başarı oranıdır.

İF temelli yöntemde, boyutları Web sunucusu erişim kütüğünden elde edilen oturum kümesindeki oturumların boyutları ile aynı olacak şekilde rastgele bir oturum kümesi yaratılmıştır. Örneğin Web sunucusu erişim kütüğünden Bölüm 4.1'de açıklandığı gibi oturumları elde ettiğimizde, boyutu 2 olan 1542 oturum var olduğundan yeni yaratılan oturum kümesinde de boyutu 2 olan 1542 oturum rastgele yaratılmıştır. Rastgele yaratılan oturumlardaki elemanlar, erişim kütüğünden elde edilmiş oturumlardaki elemanlar arasından rastgele seçilmiştir. Her iki oturum kümesi için de (erişim kütüğündeki kayıtlardan elde edilen ve rastgele yaratılan) Bölüm 4.3'de belirtilen OSD değeri hesaplanmıştır. Niteliklerin katsayılarının belirlenmesinde **(4.9)**'daki *K* değeri deneysel olarak belirlenerek 1,20 alınmıştır. Böylece bir nitelik, kullanıcıların ürün seçimindeki tercihini en az %20 etkileyebiliyorsa önemli bir nitelik olarak kabul edilmiştir. Farklı *K* değerlerine (eşik değerlerine) göre oluşan başarı oranları Çizelge 5.9'da verilmiştir.

Her iki yöntemin testi için de 10 kat çapraz doğrulama tekniği kullanılmıştır. 2790 oturum, 279 oturum içerecek şekilde 10 farklı oturum kümesine bölünmüş ve 9 oturum kümesi öğrenme kümesi olarak kullanılarak katsayılar belirlenmiş, belirlenen katsayılar ile 1 oturum kümesinde test gerçekleştirilmiştir.

GÖU-1 yönteminde niteliklerin ağırlıklandırılması ile ilgili test sonuçları Çizelge 5.8'de verilmiştir. 3 kitabın önerildiği durumda başarı %44'den GA ile %62'ye; İF ile %58'e çıkmıştır. Niteliklerden kullanıcı tercihlerinde önemsiz olanlarının belirlenmesi ve bunların benzerlik hesabındaki etkilerinin azaltılması başarıyı artırmıştır. Testlerin başarı sonuçlarının standart sapması GA'da 1 kitap önerim testi için 1,08 ve 3 kitap önerim testi için 1,39'dur; İF'de 1 kitap önerim testi için 1,37 ve 3 kitap önerim testi için 3,87'dir. Testlerin başarı sonuçlarının varyansı GA'da 1 kitap önerim testi için 1,16 ve 3 kitap önerim testi için 1,93'dür; İF'de 1 kitap önerim testi için 1,87 ve 3 kitap öneri testi için 14,97'dir Tüm test sonuçları Ek A.4'de verilmiştir.

Elemanları eşleşmiş olmayan ve normal dağılım göstermeyen iki grubun istatistiksel anlamlılığının tespitinde Mann Whitney U Testi kullanılabilir [37]. Bu nedenle önerilen modelin başarısının, Markov zinciri modelinin başarısına göre istatistiksel anlamlılığının tespitinde Mann Whitney U Testi kullanılmıştır. GÖU-1 yöntemi ve GA yönteminin kullanıldığı önerilen sisteminin başarısının, Markov zinciri modelinin başarısına göre $p<0,005$ anlamlılık düzeyinde anlamlı bir fark gösterdiği ve başarıyı önemli ölçüde artırdığı tespit edilmiştir. ($p=0,0002$)

Çizelge 5.8: GÖU-1 Yönteminde Nitelik Ağırlıklandırma Yöntemlerinin Başarıları.

Önerilen Kitap Sayısı	**Yöntemler**			
	Bir Dereceli Markov Zinciri	Ağırlıklandırma Olmadan	İF Temelli	GA
1	%14,5	%17	%25	%27
3	%36	%44	%58	%62

Çizelge 5.9: *K* değerine göre İF Temelli Nitelik Ağırlıklandırma Başarıları.

K değeri	Başarı Oranı
1,1	%49
1,2	%58
1,3	%51
1,4	%50

GA ile niteliklerin ağırlıklandırılmasında çaprazlama oranı 1,0 olarak alınmış ve yeni nesil kromozomların elde edilmesinde tüm kromozomlar Şekil 4.3'e göre çaprazlanmıştır. Çeşitli popülasyon boyutları ile çeşitli çaprazlama tekrar sayıları kullanılarak farklı başarı oranları elde edilmiştir. Çaprazlama sayısı ve popülasyondaki kromozom sayısının başarıya olan etkisi Çizelge 5.10'da verilmiştir.

Çizelge 5.10: Populasyondaki Kromozom Sayısı ve Çaprazlama Tekrar Sayısı.

Popülasyondaki Kromozom Sayısı	**Çaprazlama Tekrar Sayısı**							
	1	5	9	13	17	21	25	29
5	13%	17%	19%	22%	22%	22%	22%	22%
30	11%	18%	20%	21%	23%	23%	24%	24%
100	12%	20%	23%	25%	25%	27%	27%	27%

GÖU-1 yöntemi kullanılarak gerçeklenen ikinci testte kullanıcının ziyaret ettiği son iki ürün dikkate alınarak ürün öneriminde bulunulmuştur. Bu testte kullanıcının son ziyaret ettiği ürün ve sondan bir önce ziyaret ettiği ürün için iki farklı öneri kümesi oluşturulmuş ve kullanıcıya bu kümelerin kesişimindeki ürünler önerilmiştir. Kümelerin kesişimindeki ürün sayısının önerilecek ürün sayısından az olması durumunda son ziyaret edilen ürün için oluşturulmuş öneri kümesindeki ürünlerden son ziyaret edilen ürüne en benzer olanlar önerilmiştir.

Gerçeklenen bu testin başarısı boyutu 2'den büyük olan 1240 oturumun 10 kümeye bölünmesi ile 10 kat çapraz doğrulama modeli kullanılarak gerçeklenmiştir. Test sonuçları Çizelge 5.11'de verilmiştir. İF temelli yöntemin başarı testinde standart sapma 1 kitap öneriminde 2,09 ve 3 kitap öneriminde 4,77; GA temelli yöntemin başarı testinde standart sapma 1 kitap öneriminde 1,75 ve 3 kitap öneriminde 2,43'dür. Önerilen sistemin son iki ürüne göre öneride bulunduğu yapının başarısının da iki dereceli Markov zinciri modelinin başarısına göre $p<0,005$ anlamlılık düzeyinde anlamlı olduğu ve başarıyı önemli ölçüde artırdığı Mann Whitney U Testi ile tespit edilmiştir. ($p=0,003$)

Çizelge 5.11: GÖU-1 ileNitelik Ağırlıklandırma Yöntemlerinin Başarıları -2.

Önerilen Kitap Sayısı	**Yöntemler**			
	İki Dereceli Markov Zinciri	Ağırlıklandırma Olmadan	İF Temelli	GA
1	%12,3	%16	%22	%23
3	%31,8	%42	%52	%55

GKB yönteminde niteliklere katsayı atanması durumunda **(4.14)**'de belirtilen benzerlik hesaplama formülü W^i, *i.* niteliğe ait katsayı olmak üzere aşağıda belirtildiği gibi olacaktır:

$$S(O_1, O_2) = \frac{\sum_{i=1}^{n} O_1^i * O_2^i * W^i}{\sqrt{\sum_{i=1}^{n} (O_1^i)^2} + \sqrt{\sum_{i=1}^{n} (O_2^i)^2}} \qquad \textbf{(5.1)}$$

GKB yönteminde de GÖU-1 yönteminde olduğu gibi 10 kat çapraz doğrulama testi uygulanmıştır. GA testinde popülasyondaki kromozom sayısı 25 olarak alınmış ve 100 defa çaprazlama uygulanmıştır. On test sonucunun ortalama başarısı Çizelge 5.12'de verilmiştir. Testlerin standart sapması İF'de 1 kitap önerim testi için 1,09 ve 3 kitap önerim testi için 1,16; GA'da 1 kitap önerim testi için 0,67 ve 3 kitap önerim testi için 1,34'dür. . Testlerin varyansı İF'de 1 kitap önerim testi için 1,18 ve 3 kitap önerim testi için 1,34; GA'da 1 kitap önerim testi için 0,44 ve 3 kitap önerim testi için 1,79'dur. Tüm test sonuçları Ek A.5'de verilmiştir. GKB yönteminde de GÖU-1 yönteminde olduğu gibi önemsiz niteliklere düşük katsayılar atanarak etkilerinin azaltılması ürün önerimindeki başarıyı artırmıştır. GKB ve GA yöntemlerinin kullanıldığı önerilen sistemin başarısının, Markov zinciri modelinin başarısına göre 0,05 anlamlılık düzeyinde anlamlı olduğu Mann Whitney U testi ile tespit edilmiştir. (p=0,01 ve u=15,5)

Çizelge 5.12: GKB Yönteminde Nitelik Ağırlıklandırma Yöntemlerinin Başarıları.

Önerilen	**Yöntemler**		
Kitap Sayısı	Ağırlıklandırma Olmadan	İF Temelli	GA
1	%7	%15	%17
3	%17	%35	%39

6. SONUÇ VE ÖNERİLER

Ürün öneri sistemlerinde geniş bir kullanım alanı olan İF ve İTF yöntemlerindeki soğuk başlangıç, eleman seyrekliği ve limitli çeşitlilik gibi problemleri çözmek ve ürünlerin derin anlamsal ilişkilerinden faydalanarak başarımı artırmak için ontoloji tabanlı ürün öneri sistemleri geliştirilmiştir. Ontolojinin temel içeriği belirli bir alandaki kavramlar ve bu kavramlar arasındaki ilişkilerdir. Dolayısıyla bir ürünün ontolojisi hem ürünün sınıfını hem de ürünün ilişkide olduğu diğer sınıfları içermelidir. Bu tarz bir yapı ilişkisel veri yapısıdır ve ontolojiler üzerinde veri madenciliği işlemlerini gerçekleştirmek için ilişkisel veri madenciliği teknikleri kullanılabilir. Bu noktadan hareketle yüksek başarılı bir ürün öneri sistemi oluşturmak için etki alanı ontolojilerinden yararlanan ilişkisel bir ürün öneri sistemi geliştirilmiş ve test edilmiştir. Deneysel sonuçlar, ontolojiye ürünün ilişkisi olduğu diğer sınıfları da dahil etmenin başarıyı artırdığını göstermiştir. Çalışmada üzerinde durulan bir diğer konu ise niteliklere katsayılar atanarak ağırlıklandırmaktır. Bir ürünün niteliklerinin hepsi kullanıcılar için ayırt edici değildir. Kullanıcıların tercihini etkilemeyen niteliklerin tespiti ve bu niteliklerin katsayısının düşük tutularak benzerlik hesaplamasındaki etkilerinin azaltılması başarıyı artırmıştır. Deneysel sonuçlar bir kitap ontolojisi için test edilmiştir ancak önerilen sistem diğer ürünlerin ontolojilerine de uygulanabilir. Aktif bir oturumu etki alanı ontolojisindeki bir sınıfa ait nesne ile temsil edebilecek birleştirme fonksiyonu tanımlamak gibi ilginç konular gelecek çalışmalar için açık bırakılmıştır.

KAYNAKLAR

[1] **Url-1** <www.amazon.com>, alındığı tarih 30.04.2011.

[2] **Url-2** <www.pandora.com>, alındığı tarih 30.04.2011.

[3] **Pazzani, M., Muramatsu, J., Billsus, D.,** 1996. Syskill & Webert: Identifying interesting Web sites, *Proceedings of 9th National Conference on Artificial Intelligence,* California, Irvine.

[4] **Adomavicius, G., Tuzhilin, A.,** 2005. The Next Generation of Recommender Systems: A Survey of the State-of-the-Arts and Possible Extensions, *IEEE Transactions and data engineering,* Volume **17***, No:6.*

[5] **Pazzani J.M., Billsus D**. 2007. Content-Based Recommendation Systems, *The Adaptive Web Methods and Strategies of Web Personalization*, Springer, pp. 325-341.

[6] **Dai, H., Mobasher, B.,** 2005. Integrating semantic knowledge with Web usage mining for personalization, *In: Web Mining: Applications and Techniques,* IRM Press, Idea Group Publishing.

[7] **Pazzani, J.M.,** 1999. A Framework for Collaborative, Content-Based, and

Demographic Filtering, *In Journal of Artificial Intelligence Review - Special issue on data mining on the Internet*., Volume **13**, Issue 5-6 pp. 393-408.

[8] **Gruber, T.,** 1995. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies* Volume **43**, Issues 5-6. p.907-928.

[9] **Melville, P., Mooney, R., Nagarajan R.,** 2001. Content-boosted collaborative filtering, *Proceedings of SIGIR-2001 Workshop on Recommender Systems,* New Orleans, LA.

[10] **Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.,** 1999. An algorithmic framework for performing collaborative filtering, *Proceedings of the 22nd ACM Conferenceon Research and Development in Information Retrieval*, Berkeley.

[11] **Ma, H., King, I., Lyu, M. R.** 2007. Effective missing data prediction for collaborative filtering, *Proceedings of the 30th annual international ACM conference on research and development in information Retrieval,* NewYork, pp. 39-46.

[12] **Sarwar, B., Karypis, G., Konstan, J., Riedl, J.,** , 2000. Analysis of recommendation algorithms for E-Commerce, *Proceedings of the International Conference on Electronic Commerce*, Seoul, Korea, pp. 158–167.

[13] **Çubuk, M.,** 2009. Hybrid recommendation engine based on anonymous users, *MSc Thesis*, Eindhoven University of Technology, Eindhoven, Holland.

[14] **Debnath, S., Ganguly, N., Mitra P.,** 2008. Feature weighting in content based recommendation system using social network analysis, *Proceeding of the 17th international conference on World Wide Web,* New York, USA.

[15] **Dai, H., Mobasher, B.,** 2002. Using ontologies to discover domain-level Web usage profiles, *Proceedings of the 2nd Semantic Web Mining Workshop at ECML/PKDD 2002*. Helsinki, Finland.

[16] **Salin, S., Senkul, P.,** 2009. Using semantic information for Web usage mining based recommendation, *ISCIS 2009*, pp. 236-238

[17] **Nizar R.M., Ezeife, C.I.,** 2009. Using domain ontology for semantic Web usage mining and next page prediction, *Proceeding of the 18th ACM conference on Information and knowledge management,*New York, USA.

[18] **Goldberg D.E.** 1989. Genetic Algorithms in Search, Optimization and Machine Learning, *Addison-Wesley*, USA, pp. 1-7.

[19] **Salma, V.,** 2010. Zeki Optimizasyon Teknikleri-3 (Genetik Algoritma) <*http://volkansalma.wordpress.com/2010/02/14/zeki-optimizasyon-teknikleri-3-genetik-algoritma*>, alındığı tarih 23.04.2011

[20] **Ishii, N., Wang, Y.,** 1998. Learning feature weights for similarity measures using genetic algorithms. *Proceedings of IEEE International Joint Symposia on Intelligence and Systems*, USA.

[21] **İşçi, Ö., Korukoğlu, S.,** 2003. Genetik algoritma yaklaşımı ve yöneylem araştırmasında bir uygulama, *Yönetim ve Ekonomi Dergisi*, Cilt:10, Sayı:2, Celal Bayar Üniversitesi, Manisa.

[22] **Ryu, T., Eick, C.F.,** 1998. A unified similarity measure for Attributes with set or bag of values, *Proceedings of 6th International Workshop on Rough Sets, Data Mining and Granular Computing (RSDMGrC'98),* Research Triangle Park (NC).

[23] **Neville, J., Jensen, D., Gallagher, B.,** 2003. Simple Estimators for Relational Bayesian Classifiers, *Proceedings of the Third IEEE International Conference on Data Mining*, Amherst, MA 01003 USA

[24] **Zhang, M.Z., Yu, P.S., Wu, X.,** 2006. A General model for relational clustering, *Proceedings of Statistical Relational Learning workshop in the 23th ICML*, Pittsburgh, PA, USA. (SRL2006)

[25] **Gulli, A.,** 2011. Web mining with relational clustering, <*www.di.unipi.it/~gulli/tutorial/relational_Web_mining.pdf*> alındığı tarih: 23 Nisan 2011.

[26] **Cooley, R., Tan, P.N, Srivastava, J.,** 2000. Web usage mining: discovery and applications of usage patterns from Web data, *ACM SIGKDD Explorations Newsletter Volume 1, Issue 2*, pp. 12-23.

[27] **Url-3** <*www.microsoft.com/downloads/en/details.aspx?FamilyID=890cd06b-abf8-4c25-91b2-f8d975cf8c07&displaylang=en*>, alındığı tarih 23.04.2011

[28] **Url-4** < http://rbytes.net/software/pad-information-extraction-tool-review/>, alındığı tarih 30.04.2011

[29] **Url-5** < http://www.webinfoextractor.com />, alındığı tarih 30.04.2011

[30] **Li, Y., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S**., 2008. Regular expression learning for information extraction, *Proceeding of the Conference on Empirical Methods in Natural Language Processing,* Stroudsburg, USA.

[31] **Fouss, F., Faulkner, S., Kolp, M., Pirotte, A., Saerens, M.**, 2005 . Web recommendation system based on a Markov-chain model, *Journal of International Conference on Enterprise Information Systems (ICEIS 2005)*", Miami, USA

[32] **Liu, Y., Huang, X., An, A.,** 2007. Personalized recommendation with adaptive mixture of Markov Models, *Journal of the American Society for Information Science and Technology*, Wiley, USA

[33] **Levene, M., Loizou, G.,** 2003. Computing the entropy of user navigation in the Web, *Journal of Information Technology and Decision Making*, Volume 2, pp. 459-476.

[34] **Deshpande, M., Karypis, G.,** 2011. Selective Markov Models for Predicting Web-Page Accesses, *Journal of 1st SIAM International Conference on Data Mining*, Chicago, USA.

[35] **URL-6** <*http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview*>, alındığı tarih 30.04.2011

[36] **Servi, T.,** 2009. Çok değişkenli karma dağılım modeline dayalı kümeleme analizi*, PhD Thesis*, Çukurova University, Adana, Turkey.

[37] **Motulsky, H.,** 2010. Intuitive Biostatistics, *Oxford University Press*, New York, USA, pp.387-389

EKLER

EK A.1

Log Parser Sorgu Cümlesi:

"select date,time,c-ip,cs-uri-query,cs(User-Agent) into C:\Users\02481361\Desktop\c\filt.log from *.log where cs-uri-query like'%sid=%' and cs(User-Agent) not like '%bot%' and cs(User-Agent) not like '%arama%' and cs(User-Agent) not like '%ask%' and cs(User-Agent) not like '%slurp%' and cs(User-Agent) not like '%Crawler%' and cs(User-Agent) not like '%findlinks%' and cs(User-Agent) not like '%spider%'" and cs(User-Agent) not like '%OfficeLiveConnector%' and cs(User-Agent) not like '%Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+Trident/4.0;+SLCC2;+.NET+CLR+2.0.50727;+.NET+CLR+3.5.30729;+.NET+CLR+3.0.30729;+Media+Center+PC+6.0;+InfoPath.2)%' and cs(User-Agent) not like '%Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.0;+Trident/4.0;+SLCC1;+.NET+CLR+2.0.50727;+Media+Center+PC+5.0;+InfoPath.2;+.NET+CLR+3.5.30729;+.NET+CLR+3.0.30729)%' and cs(User-Agent) not like '%Mozilla/5.0+(Macintosh;+U;+PPC+Mac+OS+X+10_4_11;+en)+AppleWebKit/533.18.1+(KHTML,+like+Gecko)+Version/4.1.2+Safari/533.18.5%' and cs(User-Agent) not like Mozilla/4.0+(compatible;+ MSIE+8.0;+ Windows+ NT+ 6.1;Trident/4.0;+GTB6.5;+SLCC2;+.NET+CLR+2.0.50727;+.NET+CLR+3.5.30729;+.NET+CLR+3.0.30729;+Media+Center+PC+6.0;+InfoPath.2;+AskTbBT5/5.9.1.14019)%' and c-ip not like '%192.165.213.18%' and c-ip not like '%212.175.133.91%' and cs(User-Agent) not like '%Mozilla/5.0+(Windows;+U;+Windows+NT+5.1;+tr;+rv:1.9.2.12)+Gecko/20101026+Firefox/3.6.12%' and cs(User-Agent) not like '%Mozilla/4.0+(compatible;+MSIE+7.0;+Windows+NT+5.1;+.NET+CLR+1.1.4322;+.NET+CLR+2.0.50727;+.NET+CLR+3.0.04506.648;+.NET+CLR+3.5.21022)%' '%Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+WOW64;+Trident/4.0;+GTB6.6;+SLCC2;+.NET+CLR+2.0.50727;+.NET+CLR+3.5.30729;+.NET+CLR+3.0.30729;+Media+Center+PC+6.0;+HPNTDF;+InfoPath.2)%' and cs(User-Agent) not like '%Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+5.1;+Trident/4.0;+InfoPath.2;+.NET+CLR+2.0.50727;+.NET+CLR+3.0.4506.2152;+.NET+CLR+3.5.30729;+handyCafeCFW/3.3.34;+handyCafeCln/3.3.20)%' and cs(User-Agent) not like '%Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.0;+Trident/4.0;+SLCC1;+.NET+CLR+2.0.50727;+Media+Center+PC+5.0;+.NET+CLR+3.5.30729;+InfoPath.1;+.NET+CLR+3.0.30729;+.NET4.0C)%' and cs(User-Agent) not like '%Mozilla/5.0+(Windows;+U;+Windows+NT+5.1;+tr;+rv:1.9.2.3)+Gecko/20100401+Firefox/3.6.3%' -i:IISW3C -o:W3C"

EK A.2

Çizelge A.2: Markov Modeli 10 Kat Çapraz Doğrulama Testleri Başarıları

Test	Başarım Sonucu (%)
Test 1	17.3
Test 2	17.2
Test 3	16.7
Test 4	15.8
Test 5	14.7
Test 6	14.6
Test 7	13.4
Test 8	13.3
Test 9	11.9
Test 10	10.2

EK A.3

Çizelge A.3: Kategoriler Ve Kategori Grupları.

Kategori Grupları			
Sanat Kitapları	**Bilgi & Düşünce Kitapları**	**Çocuk Kitapları**	**Edebiyat Kitapları**
Sanat	Başvuru Kitapları	Çocuk Kitapları 8-12 Yaş	Anı
Ressamlar	Özel Dizi	Çocuk ve Eğitimi	Anlatı
Sanat Dünyamız	Gelişim Kitaplığı	Gençlik Kitapları	Biyografi
Sergi Kitapları	Genel Kültür	İlkGençlik Kitapları	Deneme
Oyun	Tıp Kitapları Dizisi	Okul Çağı Kitaplığı	Deneme-Eleştiri
Senaryo	Koç Üniversitesi Yayınları	Seçme Öyküler	Edebiyat Söyleşileri
Tiyatro	Politika	Doğan Kardeş Dergisi	Gazete Yazıları
Türkiye'de Güncel Sanat	COTİGO (Düşünce)	Çizgi Roman	Gezi
Şehir Monografileri	Sosyoloji	Çocuk-Okul Dönemi	Günlük
Kültürel Çalışma Dizisi	Tarih	Çocuk Kitapları Dizisi	Hikayeler
	Belgeler	Masal	İnceleme
	Düş İzleri		Köşe Yazıları
			Makale
			Mektuplar
			Otobiyografi
			Öykü
			Roman

EK A.4

Çizelge A.4: GÖU-1 10 Kat Çapraz Doğrulama Testleri Başarıları

Test	İF		GA	
	Kitap Sayısı=1	Kitap Sayısı=3	Kitap Sayısı=1	Kitap Sayısı=3
Test 1	27,4	65,3	28,5	64,1
Test 2	26,9	63,4	27,8	63,8
Test 3	26,7	60,2	27,7	63,5
Test 4	25,8	58,4	27,5	63,2
Test 5	25,1	57,3	26,9	63,6
Test 6	24,6	56,6	26,8	62,1
Test 7	24,5	55,4	26,4	61,5
Test 8	24,2	55,1	26,1	61
Test 9	24,1	54,6	25,9	60,6
Test 10	23,3	53,7	24,8	60,2

EK A.5

Çizelge A.5: GKB 10 Kat Çapraz Doğrulama Testleri Başarıları

Test	İF		GA	
	Kitap Sayısı=1	Kitap Sayısı=3	Kitap Sayısı=1	Kitap Sayısı=3
Test 1	16,8	36,8	18,1	41,1
Test 2	16,4	36,6	17,8	40,9
Test 3	15,9	36,4	17,6	40,4
Test 4	15,5	36,2	17,4	39,8
Test 5	15,3	35,5	17,2	39,4
Test 6	15,1	35,4	16,8	39
Test 7	14,9	34,6	16,5	38,6
Test 8	14,4	34,2	16,4	38,2
Test 9	13,7	34	16,4	37,5
Test 10	13,4	33,6	16,2	37,3

ÖZGEÇMİŞ

Ad Soyad: Hikmet KAPUSUZOĞLU

Doğum Yeri ve Tarihi: Nevşehir, 19.03.1985

Adres: Kağıthane, İstanbul

Lisans Üniversite: İstanbul Teknik Üniversitesi

Yayın Listesi:

- Kapusuzoğlu, H., Gündüz Öğüdücü, S., 2011: A Relational Recommender System Based on Domain Ontology. *Proceedings of the 2nd International Conference on Emerging Intelligent Data and Web Technologies,* Tirana, Albania.