Web Kullanıcıları için Melez bir Web Öneri Sistemi

M. Göksedef
goksedef@itu.edu.tr

Ş. Öğüdücü
sgunduz@itu.edu.tr

İstanbul Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü

Özetçe

Bu makalede, web kullanıcıları için birden fazla öneri sistemini kullanan yeni bir melez öneri sistemi önerilmiştir. World Wide Web'in(www) hızlı gelişimiyle birlikte, haber, ekonomi, kültür, eğitim, sağlık hizmetleri ve reklam gibi bir çok alanda bilgi kaynağı olan İnternet ortamında, kullanıcı kendisi için gerekli bilgileri bulmakta çoğu zaman zorlanmaktadır. Bunun nedeni sorgulama araçlarının kısıtlı olması ve bilgilerin fazlalığı olarak görülmektedir. Web öneri sistemleri bu karmaşık bilgi ortamında kullanıcılara kendileri için gerekli olan bilgiyi bulmakta yardımcı olurlar. Son zamanlarda, kullanıcıların İnternet ortamındaki davranışları incelenerek bir sonraki davranışını öngörmeye çalışan çalışmalar yapılmıştır. Bu makalede, başarılı öneriler üretebilmek için iki farklı öneri modelinin sonuçlarını birleştirilmiştir. Deneysel sonuçlar, standart öneri sistemleriyle karşılaştırılınca önerdiğimiz yöntemin daha başarılı sonuçlar ürettiğini göstermiştir.

Abstract

In this paper, we propose a new hybrid recommendation model for Web users which is based on multiple recommender systems working in parallel. With the rapid growth of the World Wide Web (www), it becomes a critical issue to find useful information from the Internet. Web recommender systems help people make decisions in this complex information space where the volume of information available to them is huge. Recently, a number of approaches have been developed to extract the user behavior from her navigational path and predict her next request as she visits Web pages. In this paper, we integrate the results of two different recommendation models in order to achieve better recommendation accuracy. The experimental evaluation shows that our method can achieve a better prediction accuracy compared to standard recommendation systems while still guaranteeing competitive time requirements.

1. Giriş

İnternetin yaygınlaşması ve her alanda bilgi sağlaması günlük yaşantımıza hızla girmesine neden olmuştur. Haber, ekonomi, kültür, eğitim, sağlık hizmetler ve reklâm gibi birçok alanda bilgi kaynağı olan İnternet ortamında, kullanıcı kendisi için gerekli bilgileri bulmakta çoğu zaman zorlanmaktadır. Bunun nedeni sorgulama araçlarının kısıtlı olması ve bilgilerin fazlalığı olarak görülmektedir. Verilere ulaşım sürecinde kullanıcıyı doğru ve hızlı olarak yönlendirmek hem web sitesinin etkili kullanımı açısından hem de elektronik ticaret sitelerinin amaçlarına ulaşmaları açısından önemlidir. Bu konu web madenciliğinin uygulama alanlarından biri olan öneri modelleriyle çözümlenebilir. Web öneri sistemleri kullanıcıların ihtiyacı olan bilgileri öngörür ve onlara önerilerde bulunarak ziyaretlerini kolaylaştırır. İnternet ortamında kullanılan öneri sistemlerinin girişleri kullanıcıların davranış modeli ve önerilebilecek sayfalar (veya ürünler) olarak düşünülebilir. Çıktıları ise bu sayfaların (ürünlerin) alt kümesidir. Bu sistemlerin amacı kullanıcıların davranışlarını inceleyerek gelecekteki davranışları için öngörüde bulunmaktır. Kullanıcıların davranışlarını modellemek için bir veri madenciliği uygulaması olan web kullanım madenciliği ana yaklaşımlardan biridir. Web kullanım madenciliği, web sunucusu erişim günlükleri, yetkili sunucu günlükleri, tarayıcı günlükleri, kullanıcı oturumları[1], kayıt olma verileri, kullanıcı profilleri, çerezler, kullanıcı sorguları ve yer imi verileri gibi ikincil veriler üzerinde veri madenciliği teknikleri ile kullanıcı örüntülerini bularak web tabanlı uygulamaların ihtiyaçlarını anlamaya ve onlara daha iyi hizmet etmeye çalışır. Son yıllarda web kullanım madenciliği konusunda, öneri sistemleri, internet sitelerinin tasarımı, kullanıcı profillerinin bulunması gibi birçok çalışma geliştirilmiştir. Bütün bu çalışmalarda amaç daha etkin bir öngörü algoritmasının bulunmasıdır. Öngörünün temel konusu kullanıcının gelecekteki isteklerini bulan algoritmanın geliştirilmesidir. Bunun için en başarılı yol kullanıcının geçmiş davranışlarının incelenerek modellenmesidir.

[1] [18]'de *kullanıcı oturumları* terimi, kullanıcın bir ziyaretinde oluşturduğu tıklama izi olarak kullanılmıştır. Bu çalışmada, bu terim *sunucu oturumu* terimiyle değişimli olarak kullanılacaktır.

Kullanıcının davranış örüntülerinin bulunması ve işbirlikçi filtreleme teknikleri öneri sistemlerinin kullandığı temel yöntemlerdir. Örüntülerin bulunması için ardışık örüntüler, ilişkilendirme kuralları, markov modelleri ve demetleme gibi veri madenciliği teknikler kullanılır.

İşbirlikçi filtreleme teknikleri, bir kullanıcının zevkine en yakın olan başka kullanıcıların beğendikleri ürünleri önerir. Bu yaklaşımının bazı eksiklikleri vardır. Sistemdeki ürün sayısı artıkça başarılı öneri yapmak zorlaşır. Her ne kadar başarılı öneri yapan sistemler olsa da bu sistemleri başarılı hale getirmek için yapılan işlemler sistemin çalışma maliyetini artırır [16].

Öneri sistemlerinde ilişkilendirme kuralları [12], ardışık örüntüler [1] ve markov modelleri [4, 6, 15] gibi yöntemler de kullanılmıştır. Bu sistemler karmaşık yapıya sahip olmayan internet sitelerinde başarılı olmuştur. Fakat karmaşık ve sayfalar arasındaki bağlantı sayısının çok fazla olduğu sitelerde yapılan deneyler, bu sistemlerin büyük bellek ve zamana gereksinimlerinin olduğunu göstermiştir. Yüksek derecedeki markov modelleri dışındaki sistemlerin kullanıcının tüm davranışını anlaması imkânsızdır. Bu modellerin ise parametre sayısı çok olduğundan, fazla sayfa (markov modelindeki durum sayısı) içeren internet sitelerinde yüksek dereceli markov modellerini öğrenmek olanaklı değildir.

Yukarıda bahsedilen yöntemlere alternatif olarak, [13]'te gerçekleştirilen sistemde, demetleme bulanık mantıkla yapılır. Bu sayede sayfa veya kullanıcı birden fazla demete atanabilir. Fakat bu çalışmada elde edilen demetlerin öneri için kullanımı gösterilmemiştir. [2,19]'daki yöntemlerde, kullanıcı oturumları birbirlerine olan benzerliklerine bakılarak demetlere atanmıştır. [10]'daki sayfa önerileri sunucu günlüklerinden elde edilen sayfaların oluşturduğu demetleri temel almıştır. [11]'deki öneri sisteminde de yine sunucu günlüklerinden elde edilen kullanıcı verilerinden ilişkilendirme kuralları çıkarılmıştır.

Web madenciliğinde, öneri sistemleri yoğun olarak incelenmiştir. Buna rağmen, öneri kalitesi ve kullanıcı memnuniyeti hala istenilen seviyede değildir. Bu nedenle, bu çalışmada öneri kalitesini artıran, aynı zamanda hızlı öneride bulunan yeni bir öneri sistemi tasarımı üzerinde yoğunlaştık. Bu çalışmada, iki farklı öneri sistemini birbirlerinin eksiklerini gidermek için birleştiren bir melez öneri sistemi oluşturmak temel amaçtır. Eğer bu yöntemler düzgün bir şekilde birleştirilirse son öneri başarımı artırılabilir. Bu çalışmada web sunucusu günlük verilerinin temizlenmesinden ve önişlenmesinden çok, doğru bir öneri sistemi oluşturmak amaç edinildiği için, verilerin kullanıcı oturumlarından oluşan bir dizi olduğunu varsaymak yeterli olacaktır. Bir kullanıcı oturumu, kullanıcının bir sitede sırayla ziyaret ettiği farklı sayfalardan oluşan bir kümedir.

Önerilen yaklaşımdan özetle bahsetmek gerekirse sistem iki ana bölümden oluşmaktadır: çevrim-dışı ve çevrim-içi bölüm. Çevrim-dışı bölümde web sunucusu günlük verileri önişlemeden geçirilir ve kullanıcı oturumlarından oluşan kullanıcı örüntüleri ortaya çıkarılır. Çevrim-içi bölümde ise kullanıcı örüntüleri temel alınarak öneri üretilir. Öneri üretmek için, iki farklı öneri modeli birleştirilmiştir: "A Web Page Prediction Model Based on Click-Stream Tree Representation of User Behavior" (TİA-Model) [7] ve "Model-Based Clustering and Visualization of Navigation Patterns on a Web site" (Markov-Model) [4]. İki yöntem de, birbirlerine benzer kullanıcı ziyaretlerini gruplamak için kullanıcı oturumlarını demetlere atar. İlk model çizge tabanlı demetleme yöntemini kullanırken, diğer öneri sistemi model tabanlı bir demetleme algoritması kullanır. Çevrim-dışı çalışmanın sonunda, iki farklı demetleme algoritmasından iki farklı demetleme sonucu elde edilir. Çevrim-içi bölümde ise, kullanıcı tarafından bir istek gelince, iki öneri sistemi paralel çalışarak kullanıcının daha önceden ziyaret etmediği, herbiri dört sayfadan oluşan iki öneri kümesi oluşturur. Bu öneri kümeleri birleştirilerek son öneri kümesi elde edilir. Sistemin daha önceden yaptığı önerilerin başarısına göre birleştirme yöntemi öneri üretme sırasında güncellenir. Üç farklı web sunucu günlüğünden elde edilen sonuçlar önerilen yöntemin önemli bir iyileştirme sağladığını göstermiştir. Bu makalenin yaptığı katkıları şunlardır:

1. İki farklı öneri sistemini birleştiren yeni bir melez öneri modeli sunulması
2. Yüksek başarımlı bir öneri modeli geliştirilmesi

Makalenin geri kalanı şu şekilde düzenlenmiştir. 2. bölümde, çalışmada kullanılan farklı iki öneri sistemi açıklanacaktır. 3. bölümde, melez öneri sisteminin tasarımı sunulacaktır. 4. bölümde deney sonuçları gösterilecektir. Son bölüm olan 5. bölümde ise elde edilen sonuçlar ve gelecekte yapılacak çalışmalar gösterilecektir.

2. Melez Öneri Sisteminin Modülleri

Bu bölümde, önerdiğimiz melez öneri sistemini oluşturan iki farklı öneri modeli kısaca açıklanacaktır. Bu öneri modelleri melez öneri isteminin modüllerini oluşturmaktadır. 1. bölümde bahsedildiği gibi çevrim-dışı çalışmada, web sunucularından elde edilen web sunucusu günlük verileri kullanılarak, kullanıcı davranışları modellenir. Çevrim-içi aşamada ise bu model kullanılarak kullanıcıya önerilerde bulunulur. Bu çalışmada, kullanıcı davranışları iki farklı yöntemle modellenmiş ve bu modellerin ürettiği öneriler birleştirilerek kullanıcıya sunulmuştur. Şimdi bu iki öneri modeli kısaca açıklanacaktır.

2.1. Tıklama İzi Ağacı Modülü

Tıklama İzi Ağacı (TİA) modülünde öneriler [7] çalışmasındaki gibi üretilir. [7] çalışmasında, kullanıcı oturumları arasındaki ikili benzerlikleri hesaplamak için dizi hizalama teknikleri kullanılmış ve yeni bir yöntem önerilmiştir. Bu yöntemde kullanıcıların hangi

sayfayı hangi sıra ile ziyaret ettiği ve bu sayfalarda ne kadar süre geçirdikleri bilgisi göz önüne alınmıştır. Ancak bizim çalışmamızda sistemi hızlandırmak ve basitleştirmek için süre bilgisi kullanılmamıştır. Hesaplanan ikili benzerlikler ile düğümleri kullanıcı oturumları olan bir çizge oluşturulmuştur. Çizgedeki ayrıtların ağırlığı, bu ayrıtları oluşturan düğümlerdeki kullanıcı oturumları arasındaki benzerliğe eşittir. Bu çizge elde edildikten sonra çizge tabanlı bir demetleme algoritması ile, kullanıcı oturumları demetlere ayrılır. Böylece her demet birbirine en çok benzeyen kullanıcı oturumlarından oluşur ve TİA adı verilen bir ağaç ile temsil edilir. Bu ağacın düğümleri temsil ettiği demetteki kullanıcı oturumlarındaki sayfalardır. Aktif kullanıcıdan bir istek geldiğinde en yakın kullanıcı oturumu[2] bulunur. Daha sonra bu kullanıcı oturumunun yer aldığı TİA'dan yararlanılarak, kullanıcının daha önce ziyaret etmediği ve ihtiyaçlarına en iyi şekilde cevap veren dört sayfa belirlenir. Aktif kullanıcının ilk iki isteğinde önerilecek sayfalar tüm demetlerde yani tüm ağaçlarda aranır. Bundan sonraki isteklerde ise sistem kullanıcıya öneri yapmak için kullanıcının en çok benzerlik gösterdiği N demeti bularak geri kalan önerileri üretmek için bu demetleri temsil eden TİA'ları kullanır.

2.2. Markov Modülü

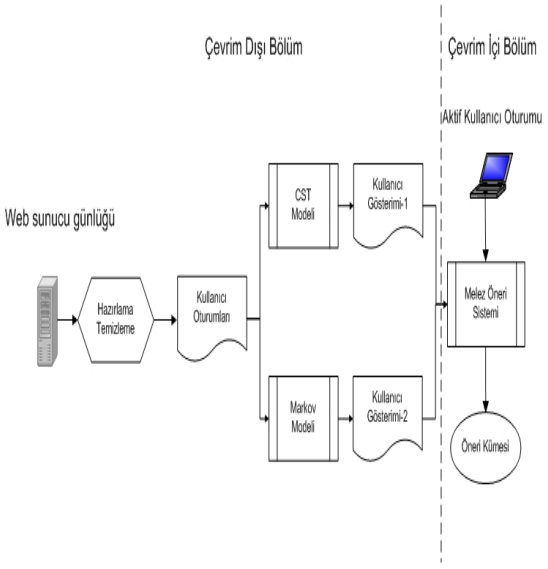
[4]'te önerilen model ise sistemde yer alacak ikinci öneri sistemidir. Bu modelde, kullanıcı oturumları, bu oturumlarda yer alan sayfaların ziyaret sırasına göre demetlere atanır. Her demet için ilk durum olasılık vektörü ve bağlantı olasılık matrisinden oluşan bir markov modeli vardır. Expectation Maximiazation (EM) algoritması ile birinci dereceden markov modellerinin parametreleri öğrenilir. Bu modülde kullanıcı oturumları şu şekilde modellenmiştir:

1) Kullanıcı oturumu belirli bir olasılıkla bir demete atanır.
2) Kullanıcı oturumunda ziyaret edilen sayfalar, oturumun ait olduğu demetin markov modeli parametreleri ile üretilir.

[4]'teki çalışma gezinti örüntülerinin görselleştirilmesi üzerine olduğundan bu model sayfa önermez. Bu sebeple, bu çalışmada Markov-Modeli adı verilen bir öneri motoru geliştirildi. Öneri motorunun detayları gelecek bölümlerde açıklanacaktır.

3. Melez Öneri Modeli

Bu bölümde, önerilen model anlatılacaktır. Çevrim-dışı çalışmada veri temizleme ve kullanıcı örüntüleri bulunmuştur, çevrim-içi çalışmada ise bulunan kullanıcı örüntülerinden yararlanılarak öneriler üretilmiştir. Şekil 1'de yapısı verilmiş olan melez öneri modelinin detayları aşağıda açıklanmıştır.



Şekil 1. Melez Öneri Sistemi Mimarisi

3.1. Veri hazırlama ve temizleme

Web sunucuları kendilerine gelen her bir istem için erişim günlüğüne bir kayıt düşerler. Bu kayıtlar, web kullanım madenciliği için önemli bir veri kaynağıdır. Sunucu kendisine gelen her istemin tarihini ve zamanını kaydeder. Ayrıca istemin hangi adresten geldiği, hangi dosyanın istendiği, istenilen dosyanın büyüklüğü, gönderim sırasında hata oluşup oluşmadığı, dosyanın nasıl gönderildiği gibi bilgiler de sunucu kayıtlarında yer alır. Veri temizleme aşamasında, çalışmanın hedefi için gereksiz olan veriler temizlenir. Bu aşamada yapılması gereken bir diğer işlem kullanıcı oturumlarının, bu oturumlarda ziyaret edilen sayfaların ve bu sayfalarda kalış sürelerinin belirlenmesidir. Web ortamında bir oturum, kullanıcın bir web sitesine girdiği ve ayrılana kadar kaldığı süre içinde yaptığı etkinlikler olarak tanımlanabilir. Ancak hem başka web sitelerinin sunucu kayıtlarına erişmek mümkün olmadığından, hem de bir kullanıcı bir web sitesini bir veya daha fazla kere ziyaret edebileceğinden, kullanıcının bir web sitesi için oturumunun hangi sayfada başlayıp hangi sayfada bittiğini kestirmek oldukça güçtür. Bu konudaki çalışmalarda genellikle aynı kullanıcıya ait iki istek arasında belli bir süre geçtiğinde o kullanıcı için yeni bir oturum başlatılır.

Bu çalışmada da çevrim-dışı aşamada web sunucu kayıt dosyası içindeki gereksiz bilgiler ayıklanarak web sitesini ziyaret eden herhangi bir kişinin o ziyareti sırasında izlediği sayfalar dizisi çıkarılacaktır. Bu bilgi sıralı diziler yapısındadır. Öneri modellerinde çevrim-dışı aşamada, işlenmemiş web sunucusu günlük verileri temizlenir ve kullanıcı örüntülerinin

[2] Aktif kullanıcıya en çok benzerlik gösteren kullanıcı oturum.

bulunması için hazırlanır. [5, 17, 20] çalışmalarında temizlik için temel yöntemler incelenmiştir. Veri hazırlama işlemi ile sunucu günlük verileri kullanıcı oturumlarına çevrilmiştir. m uzunluğundaki s_i kullanıcı oturumu şu şekilde ifade edilir: $s_i = (p_i^1, p_i^2, ..., p_i^m)$. p_i oturum sırasında ziyaret edilen sayfalardır ve P'nin bir alt kümesidir. $(p_i^1, p_i^2, ..., p_i^m) \subset P \quad P = \{p_1, p_2, ..., p_n\}$,de sitede bulunan tüm sayfalardır.

3.2. Kullanıcı örüntülerinin gösterimi

TİA ve Markov modelleri, veri hazırlama ve temizleme aşamasında elde edilen kullanıcı oturumlarından kullanıcı davranışları için örüntüler elde eder. Daha önce belirtildiği gibi her iki model kullanıcı örüntülerini farklı biçimde temsil eder. TİA modelinde kullanıcı örüntüleri ağaç ile temsil edilirken Markov modelinde örüntüler Markov modelinin parametreleri olarak temsil edilir. Bu çalışmada melez öneri sisteminin modülleri olarak bu iki modelin seçilmesinin nedenleri şunlardır:

1. İki öneri sistemi de aktif kullanıcı için öneri üretirken ziyaret edilen sayfaların sırasını göz önüne aldıkları için birbirleriyle uyumludur.
2. TİA-Modeli öneride bulunurken kullanıcının o oturumdaki daha önceki bütün isteklerine bakar. İki kullanıcı oturumunun TİA'ya aşağıdaki gibi eklendiğini varsayalım:

TİA'ya eklenmiş olan kullanıcı oturumları:

$p_1 \quad p_2 \quad p_3$

$p_2 \quad p_4$

Eğer yeni kullanıcının ilk isteği p_2 ise, TİA modeli p_4 sayfasını önerir. Bununla birlikte, eğer kullanıcı p_1 ve p_2 sayfalarını ziyaret ettikten sonra öneri yapılırsa, p_3 önerilir. Bu sebeple, bu model birinci derecedeki markov modelini karışık dereceli markov modeli gibi davranarak tamamlar.
3. İki modelin de başarımı yüksektir.

Bunlara ek olarak, TİA-Modelinin tüm kullanıcı oturumlarını demetlere ayırmadan, tek bir TİA'da tutabilme özelliği vardır. Bu başarımı artırır fakat öneri yapılırken, aktif kullanıcı oturumunun en çok örtüştüğü oturum tüm ağaçta arandığı için öneri üretme süresi demetleme yapıldığı duruma göre daha uzundur [7].
Melez öneri sisteminin modüllerini oluşturan her iki öneri modeli kullanıcı örüntülerini farklı şekilde temsil eder. TİA-Modeli kullanıcı örüntülerini temsil etmek için bir ağaç yapısı kullanır. Kullanıcı oturumları demetlere ayrıldıktan sonra her demet TİA şeklinde gösterilir. Demetteki tüm kullanıcı oturumları TİA' da bir dal olarak gösterilir. Eğer bir dal daha önce ağaçta yer aldıysa o dalda yer alan düğümlere ait sayaç değerleri bir artırılır. Her TİA' da null olan bir kök düğümü bulunur. Kök düğümü haricindeki tüm düğümler üç alana sahiptir: veri, sayaç ve sonraki düğüm. Veri alanı sayfa bilgisini tutar. Sayaç alanında, o düğüme ulaşan kullanıcı oturumu sayısı tutulur. Sonraki düğüm alanı ise ağaçta aynı veri alanı bilgisine sahip bir sonraki düğümü gösteren işaretçidir. Her TİA için ayrıca bir başlık tablosu bulunur. Başlık tablosunun her elemanı iki alandan oluşur: Veri alanı ve ilk düğüm. Ağaçta yer alan her farklı veri alanı bilgisi için (farklı sayfa) başlık tablosunda bir eleman bulunur ve bu elemanın veri alanında da aynı sayfa bilgisi bulunur. Başlık tablosundaki bir elemanın ilk düğümü ise veri alanındaki sayfa bilgisinin ağaçta yer aldığı ilk düğümü işaret eder. Bu şekilde p_i sayfasının yer aldığı bütün oturumları bulmak için ilk önce başlık tablosunda veri alanı p_i olan eleman bulunur ve ilk düğüm işaretçisinin ağaçta gösterdiği düğüm elde edilir. Bu düğüm ağaçta p_i sayfasının yer aldığı ilk düğümdür. Bu düğümün, sonraki düğüm alanının işaret ettiği düğüm ise p_i sayfasının ağaçta yer aldığı bir sonraki düğüm olacaktır. Sırayla sonraki düğüm alanlarının işaret ettiği düğümler elde edilerek p_i sayfasının yer aldığı bütün oturumlara ulaşılır.
İkinci kullanıcı örüntü gösterimi ise Markov-modelinin gösterimidir. Markov-modelinin parametreleri şunlardır: Kullanıcı oturumlarının demetlere atanma olasılığı ($p(c_k)$, c_k k'ıncı demeti temsil eder.) ve demet parametreleri. Demet parametreleri durum uzayı, ilk değer olasılıkları ve x_i ve x_j durumları arasındaki t_{ij} bağlantı olasılıklarından oluşur. t_{ij} bağlantı değeri, sistemin x_i durumundan x_j durumuna geçme olasılığını gösterir. Bu çalışmada durum uzayı, web sitesindeki sayfalardır. x_i'den x_j'ye geçme olasılığı ise p_i sayfasından sonra p_j sayfasının ziyaret edilme olasılığıdır ($P(p_j | p_i)$). [4] Çalışmasında kullanıcı davranışlarını modellemek için birinci dereceden bir Markov modeli kullanılmıştır. Böyle bir modelde bir kullanıcın davranışını sadece bir önceki davranışı belirlemektedir. s_i, m uzunluğundaki bir kullanıcı oturumu olsun. Markov-modeli bu s_i oturumunun şu şekilde oluştuğunu varsayar:

$$p(s_i) = \sum_{k=1}^{K} p(s_i \mid c_k) p(c_k)$$

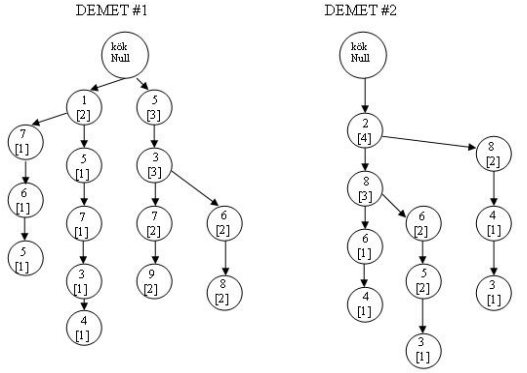
$$p(s_i \mid c_k) = p(p_i^1 \mid c_k) \prod_{j=2}^{m} p(p_i^j \mid p_i^{j-1}, c_k) \qquad (1)$$

Bu formülde K demet sayısını, $p(p_i^1 \mid c_k)$ ise k'ıncı demetteki p_i^1 için ilk durum olasılığını gösterir.

Tablo 1. Örnek 1 için kullanıcı oturumları

S	Sayfalar	Demet Numarası (TİA-Model)	Demet Numarası (Markov-Model)
S1	[p1, p5, p7, p3, p4]	1	2
S2	[p5, p3, p7, p9]	1	1
S3	[p5, p3, p6, p8]	1	1
S4	[p5, p3, p7, p9]	1	1
S5	[p1, p7, p6, p5]	1	1
S6	[p2, p8, p4, p3]	2	1
S7	[p2, p8, p6, p5]	2	1
S8	[p2, p8, p6, p4]	2	1
S9	[p2, p8, p6, p5, p3]	2	2

Örnek 1. İki farklı kullanıcı örüntü gösterimini bir örnek üzerinde gösterelim. Dokuz internet sayfasından oluşan bir P kümemiz olsun P = { p_1, ... , p_9 } . Kullanıcı oturumları ve hangi demete ait oldukları Tablo.1 de gösterilmiştir. Şekil 2 ve 3'te bu örneğe ait TİA-Modeli ve Markov-Modeli için kullanıcı oturumlarının gösterimi bulunmaktadır. Şekil 2 de iki demet için oluşturulmuş TİA gösterilmektedir. Her düğüm, veri alanına (Şekil 2'de ziyaret edilen sayfanın numaraları olarak gösterilmiştir) ve sayaç alanına (Şekil 2'de [sayaç] olarak gösterilmiştir) sahiptir. Kolaylık olsun diye veri alanında sayfalar sadece numaralarla ifade edilmiştir. Aşağıdaki Markov-Model parametreleri ise yine iki demetten oluşan bir model için bulunmuştur. Bu modelde her kullanıcı oturumu bir demete atanmıştır. Bir kullanıcı oturumunun birinci demete atanma olasılığı $p(c_1)$ = 0.776 ikinci demete atanma olasılığı ise $p(c_2)$ = 0.223 olarak bulunmuştur. Şekil 3'te her demet için bulunan demet parametreleri gösterilmiştir. Π_1 ve Π_2 demetlerin ilk durum olasılıklarını gösterir. Demetlerin bağlantı matrisleri ise sırayla T_1 ve T_2 olarak ifade edilir.



Şekil 2. İki demet için TİA

$$\pi_1 = \begin{bmatrix} 0.1 \\ 0.6 \\ 0 \\ 0 \\ 0.3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad T_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.3 & 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0.7 \\ 0 & 0 & 0 & 0.3 & 0 & 0.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\pi_2 = \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0 \\ 0.5 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad T_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Şekil 3. İki demet için Markov parametreleri

3.3. Öneri üretmek için Algoritmalar

Bundan önceki bölümlerde çevrim-dışı aşamada yapılan işlemler anlatılmıştır. Öneri motoru, öneri sisteminin gerçek zamanlı tek bileşenidir. Bu bileşen kullanıcın geçmişteki hareketlerine bakarak bir sonraki davranışı için öngörüde bulunur ve buna yönelik öneriler üretir. Çevrim-içi yapılan önerilerde, öneri sisteminin hızlı öneri yapması çok önemlidir. Bu sebeple, melez öneri sistemini oluşturan iki model paralel olarak çalışıp dört sayfadan oluşan öneri kümeleri oluştururlar. Bu öneri kümeleri birleştirilip yine, dört sayfadan oluşan aktif kullanıcıya sunulmak üzere son öneri kümesi elde edilir. Sonuçları birleştirmek için uyarlanabilir ağırlıklı ortalama yöntemi kullanılmıştır.

Aktif bir kullanıcı oturumu için, TİA-Modeli ve Markov-Modeli kendi oluşturdukları kullanıcı örüntülerinden yararlanarak önerilerde bulunurlar. TİA-Modelinin öneri üretme algoritması [7]'de verilmiştir. Markov modeli kullanarak dört sayfadan oluşan öneri kümesi ise şu şekilde oluşturulur: Aktif kullanıcı oturumu (s_a) için formül 1 ile her demette bulunma olasılığı hesaplanır ve s_a, olasılığı en yüksek olan demete (c_a) atanır. s_a kullanıcı oturumunda en son ziyaret edilen sayfa x_i ise, c_a demetinin bağlantı matrisindeki t_{ij} değerleri büyükten küçüğe doğru sıralanır. En büyük dört t_{ij} değeri için x_j durumuna karşılık gelen sayfalar öneri kümesini oluşturur. Öneri üretmek için kullanılan algoritma şu şekildedir:

Girdi: Aktif kullanıcı oturumu s_a, TİA ve Markov modellerinin oluşturdukları kullanıcı örüntü gösterimleri
Çıktı: dört sayfadan oluşan öneri kümesi

1. TİA ve Markov modelleri ile dörder sayfadan oluşan iki öneri kümesi üretilir (ÖS1 ve ÖS2). Öneri kümeleri üretilirken her iki model paralel olarak çalışır. Böylece melez öneri sisteminde iki model kullanmamızdan dolayı bu adımda bir yavaşlama olmaz.
2. Yeni kullanıcı oturumu için öneri kümesi oluşturmadan önce seçilen yönteme göre ağırlıklar güncellenir. Bu ağırlıklar son öneri kümesi belirlerken melez öneri sistemini oluşturan modüllerden

hangisinin daha etkin olacağına karar vermek için kullanılır.
3. ÖS1 ve ÖS2 aşağıda gösterildiği gibi birleştirilir.

$$\text{ÖS} = w_1 \times \text{ÖS}_1 + w_2 \times \text{ÖS}_2$$

$w_1 + w_2 = 1$ ve $w_1, w_2 \in \{1/4, 2/4, 3/4\}$.

Öneri kümelerini birleştirmek için kullanılan ağırlıklar güncellenirken, modellerin başarımları birbirinden bağımsız olarak değerlendirilir. Eğer kullanıcı bir model tarafından üretilmiş bir sayfayı seçerse, öneriyi yapan modelin ağırlığı artırılır. Eğer bu sayfa iki model tarafından üretilmiş ise iki modelin de ağırlığı artırılır. Ağırlıklar güncellenirken üç farklı yöntem kullanılmıştır:
Yöntem 1. Eğer bir model bir önceki kullanıcı oturumu için daha başarılı olmuşsa, başarılı olan modelin ürettiği öneri kümesinin ağırlığı w=3/4 olarak belirlenir.
Yöntem 2. Aktif kullanıcı oturumu için her modelden iki sayfa içeren iki farklı öneri yapıldıktan sonra, modeller değerlendirilir. Bundan sonra daha başarılı olan modelin ürettiği öneri kümesinin ağırlığı w=3/4 olarak belirlenir.
Yöntem 3. Kullanılan modellerin genel başarımına bakılır. Aktif kullanıcı oturumundan önceki oturumlar için hangi model daha başarılı öneriler yapmışsa o modelin ürettiği öneri kümesinin ağırlığı w=3/4 olarak belirlenir.

4. Deney Sonuçları

Bu bölümde, üç farklı veri kümesi üzerinde yapılan deneyler açıklanacaktır. Birinci veri kümesi NASA Kennedy Uzay Üssü sunucularından elde edilmiştir. İkinci web sunucusu günlük verisi Baltimore Washington DC bölgesinin internet servis sağlayıcısı olan ClarkNET (C.NET veri kümesi) web sunucularındandır [8]. Son veri kümesi ise Saskatchewan Üniversitesinin (SU veri kümesi) sunucularındandır [14]. Veri kümelerinin özellikleri Tablo 2' de gösterilmiştir. Bu veri kümeleri, kullanıma açık oldukları ve daha önceki çalışmalarda kullanıldıkları için seçilmiştir. Bu veriler temizlendikten sonra yaklaşık %30'luk bir kısmı rasgele seçilerek sınama kümesi olarak kullanılmıştır. Kalan kısım eğitim kümesini oluşturur. Kullanıcı oturumundaki her sayfa için, dört sayfadan oluşan bir öneri kümesi oluşturulur. Bu testlerde Linux 2.6 üzerinde çalışan (Intel Xeon 3 Ghz) çift işlemcili bir bilgisayar kullanılmıştır. Programlar Java ile geliştirilmiş olup, herhangi bir kod optimizasyonu yapılmamıştır.

Tablo 2. Temizlenmiş günlük verilerinin özellikleri

Veri Kümesi	Sayfa Sayısı	Oturum Sayısı
NASA	92	15369
C.Net	67	6846
UOS	171	7452

Başarım ölçümü için Vuruş-Oranı adı verilen bir ölçüt kullanılmıştır. Bu ölçüt şu şekilde tanımlanır. Eğer önerilen dört sayfadan bir tanesi kullanıcının bir sonraki isteği ise bu bir vuruştur. Vuruş-Oranı ise toplam yapılan öneri sayısının vuruş sayısına oranıdır. C.NET ve SU veri kümeleri için yapılan deneylerde kullanıcı oturumları demetlere ayrılmadan tek bir TİA kullanılmıştır. C.NET ve SU veri kümelerinde az sayıda kullanıcı oturumu bulunduğundan tek bir TİA kullanımı bu sistemleri çok yavaşlatmaz. Tek TİA ile elde edilen sonuçlar TİA-Modelinin elde edebileceği en yüksek başarımı sağlar [7]. Tek bir TİA kullanıldığı durumda yeni bir kullanıcıya öneri üretmek için geçmişteki tüm kullanıcı oturumları incelenir. Nasa veri kümesindeki kullanıcı oturumlarının sayısı fazla olduğu için bu veri kümesinde tek bir TİA kullanarak öneri üretmek sistemi yavaşlatır. Arama uzayını küçültmek için Nasa veri kümesinde bulunan kullanıcı turumları demetlendikten sonra her bir demet bir TİA ile temsil edilmiştir. Bu veri kümesi için demet sayısı beş olarak belirlenmiştir.

Markov-Modelinin parametrelerinin belirlenmesi için farklı demet sayıları ile deneyler yapılmıştır. Beş ile yirmi arasında değişen her farklı demet sayısı için EM algoritması on kere çalıştırmıştır. Her çalışmada EM algoritması için verilen ilk durum değerleri farklı olmaktadır. Bu çalışmalar içinde en iyi sonucu veren model belirlenmiştir. Tablo 3'de verilen sonuçlar her iki öneri sisteminin birbirinden bağımsız olarak çalıştığında elde ettikleri Vuruş-Oranı'nı göstermektedir. Ayrıca Tablo 3'te melez öneri sisteminin sonuçları mevcuttur. Bu sonuçlar bize melez öneri sisteminin daha başarılı olduğunu göstermektedir. Bu da farklı modellerin sonuçları etkin bir biçimde birleştirilince daha başarılı öneriler üretileceğini kanıtlar.

Tablo 3. Vuruş-Oranı yüzdeleri

Veri Kümesi	TİA-modeli	Markov-modeli	Melez Öneri Sistemi
NASA	62	59	66,4
C.Net	60,3	60,5	61,9
UOS	67,5	67,3	69,9

Bir sonraki deneyde, melez öneri sistemini oluşturan öneri modüllerinin doğru öneri üretmekteki başarıları gösterilmektir. Bunun için başarılı önerilerin hangi model tarafından üretildiği belirlenmiştir. Her modülün tek başına yapmış olduğu başarılı öneri yüzdesi ile ikisinin birlikte yapmış olduğu başarılı öneri yüzdesi Tablo 4'te gösterilmiştir. Bu tabloda herbir modelin başarılı önerilere katkısı verilmiştir. Örnek olarak, Nasa veri kümesi için toplam Vuruş-Oranı %66,4'dür (Tablo 3). Bu başarılı önerilerin %19,9'u TİA modeli tarafından, %8,9 Markov-Modeli tarafından, %37,7'si ise her iki model tarafından üretilen önerilerdir. Bu deneyde, her iki modelin de genelde öneri için ortak sayfalar ürettikleri gözlemlenmiştir. Eğer farklı başarılı sayfalar öneren modüller kullanılırsa melez öneri sisteminin başarısının daha artacağı düşünülmektedir.

Tablo 4. Modüllerin başarı yüzdeleri

Veri Kümesi	TİA-modeli	Markov-modeli	TİA+Markov-modeli
NASA	19,9	8,9	37,7
C.Net	15,7	8,5	37,7
UOS	11,6	4,9	53,5

Öneri kümelerini birleştirirken ağırlıkları güncelleme yöntemlerinin Vuruş-Oranına etkisi Tablo 5'te gösterilmektedir. Yöntem 1 ve yöntem 2 arasında büyük bir fark olmamasına rağmen yöntem 2 diğerlerinden daha iyi sonuç vermektedir. Bu deney, ağırlıkları sadece aktif kullanıcının davranışına göre güncellemenin en doğru yaklaşım olduğunu göstermiştir. Melez öneri sisteminin, tek bir öneri modeli kullanıldığı duruma göre ek yük getiren aşaması son öneri kümesini oluşturma aşamasıdır. Fakat yapılan ölçümler, bu aşamanın öneri üretmek için gereken zamanı %3 kadar uzattığını göstermiştir. Bu da bizim çalışmamızda yaklaşık olarak 6 μs 'ye eşittir.

Tablo 5. Ağırlık güncelleme yöntemlerine göre Vuruş Oranı başarımı

Veri Kümesi	Yöntem 1	Yöntem 2	Yöntem 3	Yöntem 2 + Yöntem 3
NASA	66,0	66,4	63,7	64,2
C.Net	61,8	61,9	61,1	61,2
UOS	69,9	69,9	69,6	69,4

5. Sonuçlar ve Gelecek Çalışmalar

Bu çalışmada, internet ortamındaki kullanıcılar için yeni bir öneri yöntemi geliştirildi. Geliştirilen bu melez öneri sistemi iki farklı öneri modelinin sonuçlarını birleştirerek tek bir öneri sonucu oluşturur. Sonuçları birleştirmek için üç farklı yöntem kullanılmış va başarımları incelenmiştir. Bu yöntemlerle yapılan deneyler, melez öneri sisteminin başarımının kullandığı iki modelden de daha iyi olduğunu göstermiştir.

Melez öneri sistemini farklı şekilde geliştirmeye devam etmekteyiz. Melez öneri sistemini oluşturan modüller için farklı öneri modelleri kullanılabilir. Birbirinden farklı sayfalar üreten öneri modelleri kullanıldığı zaman sistemin başarımı daha artabilir. Ayrıca melez öneri sisteminin modülleri olarak ikiden daha fazla öneri modeli kullanılarak başarım artırılabilir.

Kaynaklar

[1] R**. Agrawal and R. Srikant**. Mining sequential patterns. In *Proceedings of the International Conference on Data Engineering (ICDE)*, March 1995. Taipei, Taiwan.

[2] **A. Banerjee and J. Ghosh**. Clickstream clustering using weighted longest common subsequences. In *Proceedings of the Wokshop on Web Mining, SIAM Conference on Data Mining*, pages 33–40, 2001. Chicago, IL.

[3] **J. S. Breese, D. Heckerman, and C. Kadie.** Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[4] **I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White**. Model-based clustering and visualization of navigation patterns on a web site. *Data Min. Knowl. Discov.*, 7(4):399–424, 2003.

[5] **R. Cooley, B. Mobasher, and J. Srivastava.** Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1(1):5–32, 1999.

[6] **M. Deshpande and G. Karypis**. Selective markov models for predicting web-page accesses. In *Proceedings of the First SIAM International Conference on Data Mining (SDM'2001)*, 2001.

[7] **S. Gündüz and M. T. Özsu**. A web page prediction model based on click-stream tree representation of user behavior. In *Proceedings of Ninth ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 535–540, August 2003. Washington, DC, USA.

[8] C. W. S. Log. http://ita.ee.lbl.gov/html/contrib/ClarkNet-HTTP.html.

[9] N. K. S. C. Log. http://ita.ee.lbl.gov/html/contrib/NASAHTTP.html.

[10] **B. Mobasher, H. Dai, T. Luo, and M. Nakagawa.** Discovery of aggregate usage profiles for web personalization. In *Proceedings of the Web Mining for E-Commerce Workshop (WebKDD'2000)*, 2000.

[11] **B. Mobasher, H. Dai, T. Luo, and M. Nakagawa**. Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd ACM Workhop on Web Information and Data Management*, November 2001. Atlanta, USA.

[12] **A. Nanopoulos, D. Katsaros, and Y. Manolopoulos**. Effective prediction of web-user accesses: a data mining approach. In *Proceedings of WEBKDD workshop*, 2001. San Francisco, CA, USA.

[13] **O. Nasraoui, R. Krishnapuram, and A. Joshi**. Mining web access logs using a fuzzy relational clustering algorithm based on a robust estimator. In

Proceedings of Eight International World Wide Web Conference, 1999. Toronto, Canada.

[14] T. U. of Saskatchewan Log. http://ita.ee.lbl.gov/html/contrib/Sask-HTTP.html.

[15] **R. R. Sarukkai**. Link prediction and path analysis using markov chains. In *Proceedings of the Ninth International World Wide Web Conference*, 2000. Amsterdam.

[16] **B.M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl**. Application of dimensionality reduction in recommender system –a case study. In *Proceedings of the WebKdd 2000 workshop at the ACM SIGKDD 2000*, 2000.

[17] **J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan**. Web usage mining: Discovery and application of usage patterns from web data. *ACM SIGKDD Explorations*, 1(2):12–23, 2000.

[18] **J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan**. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.

[19] **W. Wang and O. R. Zaiane**. Clustering web sessions by sequence alignment. In *Proceedings of 13th International Workshop on Database and Expert Systems Applications, DEXA'02*, 2002. Aix en Provence, France.

[20] **O. R. Zaiane**. Web usage mining for a better web-based learning environment. In *Proc. Conference on Advanced Technology for Education*, pages 60–64, June 27–28 2001. Bannf, Alberta.