BİRLİKTELİK KURALLARINA VE ÖNERİ SİSTEMLERİNE DAYALI KESTİRİMCİ ANALİTİK YÖNTEMLER

Ayhan DEMİRİZ

Bölüme Genel Bakış

Birliktelik kuralları, aynı işlemde kayıt altına alınan birden fazla ürün, parça, hizmet v.b. varlıkların birlikte bulunma durumuyla ilgili genelleme yapan çıkarımlardır. Öneri sistemleri ise en basit olarak belirli ürün ve hizmeti daha önce almış veya yorumda bulunmuş (ya da beğenmiş) kullanıcılar (müşteriler) için ilave olarak hangi hizmet veya ürünlerin tavsiye edilebileceğini tespit eden sistemlerdir. Elbette işlem tarihçeleri dikkate alındığında birliktelik analizinin öneri sistemlerinde doğal olarak kullanımı düşünülebilir. Uygulamada ise başarılı bir çözümün dikkate alması gereken özel konular, farklı yaklaşım ve algoritmaların kullanımını zorunlu hale getirmiştir. Bu bölüm, bazı özel algoritma ve çözümleri dikkate alarak birliktelik madenciliği ve öneri sistemleri hakkında bilgi verir.

Anahtar Kavramlar

Market Sepet Analizi

İlişki Madenciliği

Birliktelik Kuralları

Apriori Algoritması

Sıra Madenciliği

Adım Madenciliği

Öneri Sistemleri

Müşteri ve Ürün Bazlı Öneri sistemleri

Giriş: Müşteri tercihlerinin ekonomik faydaya dönüşümü

İşletmelerde toplanan verinin büyümesi ve çeşitlenmesi veri madenciliğinin ortaya çıkması ve yaygınlaşması ile ilgili olarak en önemli sebep olarak gösterilir. Aslında bilişim sistemlerinin ticari boyutta yaygınlaşmaya başlaması ile 1960'lı yıllardan bu yana büyük işletmeler gerek üretim ve gerekse satış kanallarında veri toplamayı, paylaşmayı ve bunları karar verme süreçlerinde faydalı hale getirmeyi sürdüre gelmişlerdir. Bilişim alt yapısı olarak baktığımızda bu tip ticari bir bakış açısı birçok teknolojinin ortaya çıkmasına yardımcı olmuştur. Kuşkusuz bunların en başında geleni ise veritabanı yönetim sistemleridir. Bu tip sistemlerin zaman içinde artan bir şekilde ve çeşitlilikte veri toplamaya imkân vermesi müşterilerine daha iyi hizmet etmek, kârlılık ve verimliliklerini arttırmak isteyen işletmeleri bu verilerden faydalanmaya itmiştir. Geleneksel olarak baktığımızda pazarlama, planlama, üretim ve diğer birçok işletme biriminde iş analitiği, operasyon yönetimi ve yönetim bilimi gibi birçok yaklaşım, çeşitli analitik ve optimizasyon yöntemleri ile bu tür veriler üzerine inşa edilen karar verme süreçlerini desteklemiştir.

En basit anlamda bir örnek vermek gerekirse satış verilerinden yola çıkılarak yapılan talep tahminleri planlama, üretim ve stok kontrol gibi önemli faaliyetlerin temelini teşkil etmektedir. Bunlar işletmelerin operasyonel olarak yürüttüğü standart karar verme süreçlerinden bazılarıdır. Temel olarak baktığımızda müşterilerden ve genel anlamda satışlardan toplanan verileri kullanılarak firmalar için kritik faydaya dönüşen bir yapı söz konusudur. Başlangıçta genel olarak toplanabilen veriler (örneğin satış verileri), müşteri özelinde toplanılmaya başlandığında ve müşterinin elektronik olarak gerçekleştirdiği işlem bazına indirgendiğinde artık ekonomik fayda daha da artmıştır. Yine de unutmamak gerekir ki toplanan müşteri verilerinin ekonomik faydaya dönüşümü yeni bir olgu değildir, teknik olarak ticaretin yapıldığı her dönemde mevcuttur.

Acaba müşteri verilerinin geniş ölçekte kullanımı ekonomik faydaya nasıl dönüştürülebilir? Bu sorunun cevabı için işlem bazında (detayında) toplanan müşteri verilerine dayanan aksiyonların alınması gerekmektedir. Aslında veri madenciliği bu noktada anlam kazanmaktadır. Geniş ölçekli verilerden müşteri davranışları ve tercihleri hakkında çıkarımlar yaparak bunların aksiyon alınabilecek kurallar kümesi haline sokulması ile bunu başarabiliriz.

Günümüzde bunun başarılı sonuçlarına örnek olarak Netflix, Prime Video ve Spotify gibi dijital içerik platfromlarının yaygınlaşması ve Amazon.com önderliğinde e-ticaret platfomlarının ticaretin merkezi haline gelmeleri verilebilir. Netflix örneği aslında öneri sistemlerinin bir başarısı olarak verilmelidir. Özellikle ara vermeden yapılan video izlemelerinde (binge watching) müşterinin beğeneceği içeriğin sağlanabilmesi çok önemlidir. Benzer platformların elbette yerel alternatifleri söz konusudur. Bu platformların başarısı elbette geniş ölçekli müşteri verilerini ekonomik faydaya yani kârlılığa dönüştürebilmeleri ile ilintilidir.

Bu bölümde müşteri davranış ve tercih verilerinden azami faydanın sağlanmasına katkıda bulunan birliktelik kurallarının keşfi ve öneri sistemlerinin çalışması hakkında uygulamalı bilgi verilmeye çalışılacaktır.

Sepet Verileri ve Birliktelik Kuralları

Birliktelik madenciliğinin en yaygın kullanım alanı market sepet verilerinin analizidir. Bu tür verilerin ortak özelliği gerçekleşen bir işlem içerisinde aynı anda birden fazla ürün, parça, hizmet v.b. varlığa ait verilerin bulunmasıdır. Toplanan verilerin incelenmesi sonucunda sıklıkla birlikte alınan ürün kümeleri kolaylıkla elde edilebilir. Sıklıkla alınan ürün kümelerinden birliktelik kuralları elde edilir. Bu kurallar $if\ A\ \&\ C\ \rightarrow\ B$ şeklinde ifade edilebilen önermelere sahiptir. Bahsi geçen kural, sepet analizi özelinde "eğer A ve C ürünleri birlikte alınmışlarsa belirli bir güven (*confidence*) ve destek (*support*) oranları ile B ürünü de alınmıştır" şeklinde ifade edilebilir. Bu tür kurallar bir süpermarket ortamında hangi ürünlerin indirime girebileceğinin belirlenmesinde kolayca kullanılabilir. Bunun yanında doğal olarak "Hangi ürünler raflarda fiziksel olarak yakın yerleştirilmelidir ki ortak satışları arttırılmış olsun?" sorusuna cevap vermek için kullanılır. Hazır giyim sektörü özelinde birlikte satın alınan veya giyilmesi tercih edilen ürünlere günlük kullanımda da "kombine" denildiğini göz önünde bulundurmalıyız. Yani bir anlamda kombine ürünler keşfedilir.

Kavramların anlaşılması için ufak bir örnek vermek gerekirse, bir marketten gevrek ve süt alan müşterinin durumunu düşünelim. Her ne kadar tekil bir işlem gibi gözükse de kasa verileri incelendiğinde gevrek ve sütün sıklıkla alındığı görülebilir. "Gevrek alan, süt alır" (Gevrek → Süt) şeklinde bir kural bulabiliriz. Acaba "Süt alan, Gevrek alır" diye bir kural daha doğru olmaz mıydı? Bu sorunun cevabı kuralın güven (c) ve destek (s) oranlarının anlaşılmasıyla verilebilir. İncelenen tüm kasa işlemlerinin sayısının n olduğunu varsayalım. Bu işlemlerde alınan toplam gevrek ve süt sayılarının sırasıyla $n(G)$ ve $n(S)$, Gevrek ve Sütün birlikte alındıları, yani ortak alındıkları işlemlerin sayısının da $n(G \cap S)$ olduğunu varsayalım. Bu durumda $s = n(G \cap S)/n$ şeklinde destek oranını hesaplayabiliriz. Destek oranı tüm işlemler göz önüne alındığında Gevrek ve Süt ürünlerinin birlikte satın alınma oranıdır. Aslında bu $P(GS)$ olasılığıdır. Güven oranı ise birliktelik kuralları hakkında hesaplanabilir. Gevrek → Süt şeklinde verilen kuralın güven oranı $c = n(G \cap S)/n(G)$ olarak hesaplanır. Güven oranı, c, $P(S|G)$ koşullu olasılığına karşılık gelir. Hem destek hem de güven oranları için alt limit (minimum destek ve güven) belirlenir. Destek ölçeği için oran yerine sayı limiti de verilebilir. Bu durumda sayı olarak minimum destek limitinin üzerinde olan ürün veya ürün kümeleri sıklıkla karşılaşılanlar olarak kabul edilir. Güven oranı limitinin üzerinde kalan kurallar ise birliktelik kuralları olarak kabul edilir. Örneğimizde Süt → Gevrek kuralı belki güven limitinin altında kaldığı için önem görmez.

Genel olarak birliktelik madenciliği iki aşamadan meydana gelir. İlk aşamada sıklıkla alınan tüm ürün ve ürün kümeleri bulunur. İkinci aşamada ise sık ürün kümelerinden yola çıkılarak güven ve destek

oranlarının alt limit şartlarını (minimum) sağlayan birliktelik kuralları keşfedilir. Elbette tüm tarihsel (geçmiş) veriler dikkate alındığında market sepet analizi geçmişte yapılan indirimlerin etkilerini de istemeden yansıtabilir. Yani geçmişte yapılmış indirim dönemlerinde yaşanan aşırı hareketlilik veya bazı ürün kümelerinin birlikte indirime girmeleri, verilerin belirli bir ürün veya ürün grubu üzerinde yoğunlaşmasına sebep olabilir. Bu yüzden çıkarım yapılan kurallar bazen yanıltıcı olabilir. Veriden kaynaklanan yanlılık (bias) her zaman dikkatlice göz önüne alınmalıdır.

Apriori Algoritması

Geniş ölçekli verilerde sık ürün kümelerinin keşfedilmesi sadece bilgisayar hesaplama gücüne dayanarak yapılamayacak sayıda kombinasyon gerektirebilir. Bu açıdan hesaplama gereksinimini azaltacak algoritmalar geliştirilmiştir. Bunlardan en bilineni Apriori algoritmasıdır (Agrawal v.d., 1996; Agrawal & Srikant, 1994). Algoritma isminin çağrıştırdığı gibi bu yaklaşım "ön bilgi" kullanarak bazı faydasız hesaplamalardan kaçınıp en etkin şekilde sık ürün kümelerini keşfeder.

Şimdiye kadar bahsettiğimiz sık ürün kümelerinin k tane üründen oluşan bir küme olduğunu hatırlamamız gerekir. Kolay ifade edebilmek için buna S_k diyelim. Doğal olarak ürünler tekil olarak ele alındığında S_1 sık karşılaşılan ürünler olarak kabul edilir. Apriori algoritması "Apriori özelliği" olarak adlandırılan bir kabule dayanmaktadır. Buna göre herhangi bir k ürün küme eğer minimum destek oranı şartını sağlamıyorsa, bu kümeye eklenen bir ürün ile oluşan yeni $k+1$ ürün kümesi de minimum destek şartını sağlamayacaktır. Bu kabul, anti-monotonluk olarak adlandırılan kurala dayanmaktadır. Bu kurala göre, eğer bir küme herhangi bir testi geçemezse, altkümesi olduğu tüm üstkümeler de bu testi geçemeyecektir. Örneğin I olarak adlandırdığımız k-ürün kümesi sık olmasın. Bu I kümesine A ürününü eklediğimizi düşünelim. O zaman $I \cup A$ kümesi de sık ürün kümesi olmayacaktır.

Apriori algoritması iki ana adımdan oluşur:

1- Aday ürün kümelerinin bulunması: S_k sık k ürün kümelerinin bulunabilmesi için A_k aday kümelerinin listelenmesi gerekir. Bu amaçla S_{k-1}ürün kümeleri arasında birleşme işlemi yapılarak A_k aday kümeleri elde edilebilir. S_{k-1}ürün kümelerinde $k-2$ ortak ürüne sahip kümeler birleştirilerek A_k elde edilir.
2- A_k aday kümeleri için sayım yapılarak minimum destek şartını sağlayan adaylardan S_k elde edilir ve $k = k+1$ olarak güncellenir. Bu iki adım arzu edilen veya elde edilebilecek en büyük k değerine kadar devam eder.

İşlem No	Ürünler (veya Hizmetler)
1	41368, 47036, 55822
2	41368, 42267
3	42267, 47036, 55822
4	41368, 42267, 47036, 55822

Tablo 0-1 Örnek Veri Seti

Hizmet Kodu	**Tanım**
41368	Kişisel 800'lü Numara
42267	Temel Mesaj Servisi
47036	Arayan Numara
55822	Rehberde Numara Gizle

Tablo 0-2 Hizmet Tanımları

Küçük bir örnek üzerinde algoritmanın çalışmasını göstermek için Tablo 0-1 ve Tablo 0-2'de bazı Telekom hizmetleri hakkında işlem verileri ve açıklamalar verilmiştir. Buna göre ürün (hizmet) bazında sayma yaptığımızda Tablo 0-3'de verilen frekanslar karşımıza çıkmaktadır. Bu örnekte destek limitini 2 kabul edersek tüm ürünler sık olarak karşımıza çıkmaktadır. Burada A_2 kümesi tüm ikili kombinasyonlardır. Tablo 0-4'de gösterildiği gibi, biri dışında tüm kombinasyonlar 2 kez karşılaşılmıştır ve hepsi sıktır. 47036 ve 55822 ürünleri (hizmetleri) 3 kez birlikte alınmıştır. S_2 kümesi Tablo 0-4'de verilmiştir. Tablo 0-5, üçlü aday ürün kümelerini listelemektedir. Bazı ürün kümeleri sadece 1 kez birlikte görüldükleri için sık değildirler. Bu tablodaki son iki satır S_3 kümesini vermektedir.

Hizmet Kodu	Frekans
41368	3
42267	3
47036	3
55822	3

Tablo 0-3 Hizmet Kullanım Frekansı

Hizmet Kodu	Frekans
41368, 42267	2
41368, 47036	2
41368, 55822	2
42267, 47036	2
42267, 55822	2
47036, 55822	3

Tablo 0-4 İkili Sıklıklar

Hizmet Kodu	Frekans
~~41368, 42267, 47036~~	~~1~~
~~41368, 42267, 55822~~	~~1~~
41368, 47036, 55822	2
42267, 47036, 55822	2

Tablo 0-5 Üçlü Sıklıklar

Minimum güven oranı, c %100 olarak belirlendiğinde aşağıdaki birliktelik kuralları Tablo 0-4 ve Tablo 0-5'ten elde edilebilirler.

47036 →55822

55822 →47036

47036 41368 →55822

41368 55822 →47036

47036 42267 →55822

55822 42267 →47036

Birliktelik kuralların keşfi elbette uygulamada çok faydalar sağlayabilir. Ancak kurallar dikkatli bir analizden geçmek durumundadırlar. Sadece destek ve güven oranları anlamlı kural bulmak için yeterli olmayabilir. Örneğin yukarıda tabloda gözüktüğü gibi sol ve sağ tarafları yer değiştirmiş kurallar bulunabilmektedir. Bunlar çoğu zaman birbirini tekrar eden ya da alan uzmanının zaten bildiği kurallar olabilir. Apriori algoritması ile gereksiz birçok kuralın keşfedilmesi gayet mümkündür. Bir kuralı önemli yapan en önemli özelliği ilginçliğidir. İlginçliğin hesaplanabilmesi için kuralın sol ve sağ taraflarından yola çıkarak ihtimal tablosu (contingency table) hesaplanmalıdır. Gevrek → Süt kuralı örneğimizdeki ihtimal tablosu şu şekilde Tablo 0-6'da bulabiliriz.

	S	$\overline{S}$	
G	f_{11}	f_{10}	f_{1+}
$\overline{G}$	f_{01}	f_{00}	f_{1+}
	f_{+1}	f_{+0}	$\|T\| = n$

Tablo 0-6 Kural Keşfi için Hata Matrisi

f_{11}, G ve S'nin desteği

f_{10}, G ve $\overline{S}$'nin desteği

f_{01}, $\overline{G}$ ve S'nin desteği

f_{00}, $\overline{G}$ ve $\overline{S}$'nin desteği

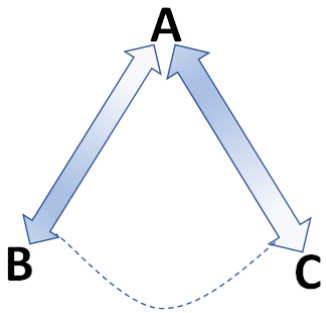
Bu tablodan yola çıkarak destek ve güven oranlarının yanında birçok istatistiksel ölçüt hesaplanabilir. Bunlar Gini, Lift, İlginçlik, F-Değeri, Jaccard-Değeri gibi değerlerdir. Örneğin,

$$\text{İ}lginçlik = \frac{P(GS)}{P(G)P(S)} = \frac{f_{11}/n}{\frac{n(G)n(S)}{n^2}} = \frac{nf_{11}}{n(G)n(S)}$$

$$Lift = \frac{P(S|G)}{P(S)}$$

şeklinde hesaplanabilir. İhtimal tablosuna dayanarak diğer birçok istatistiksel kural gücü ölçütü kolaylıkla hesaplanabilir. İstatistiksel ölçütlerin yanında öznel sıralamalar yapılarak kurallar hakkında yorumlar da yapılabilir.

Birliktelik kuralları genel anlamda aralarında birlikte alınma, birlikte görülme gibi ilişkiler olan ürünler veya olaylar için kullanılır. Acaba iki ürünün birlikte alınmaması da ilginç veya keşfe değer olabilir mi? Aslında yukarıda "kombine" olarak adlandırdığımız birbirini tamamlayan ürünler ilişki madenciliği ile bulunmaktadır. Birbirlerine alternatif olan yani birbirleri için ikame olan ürünlerin keşfi de önemlidir. Birliktelik madenciliği ikame ilişkilerin keşfinde de kullanılmıştır (Savasere v.d., 1998; Tan v.d., 2001). Negatif ilişki olarak adlandırılan bu tür ilişkiye bir örnek Şekil 0.1'de verilmiştir. Bu şekilde **AB** ve **AC** ürünleri sıklıkla birlikte görüldüğü ve bu kümelerin sık olduğunu kabul edelim. Eğer **BC** kümesi sık olarak karşımıza çıkmıyorsa B ve C ürünleri A ürünü üzerinden negatif ilişkiye (ikame) sahiptir sonucuna varırız. İkame ürün veya negatif ilişkiye verilebilecek en iyi örnek Pepsi ve Coca Cola'nın birlikte alınmamasıdır. Fakat her iki ürünün de cipsle birlikte sık alındıkları görülebilir. Cips üzerinden Pepsi ve Coca Cola'nın negatif ilişkiye sahip oldukları kolaylıkla bulunabilir.



Şekil 0.1 İkame İlişkiler (Negatif İlişki)

Diğer Yaklaşımlar

Apriori algoritmasının performansını arttırmak için çeşitli çözümlerin önerildiği bazı iyileştirmeler geliştirilmiştir. Bunlardan ilki anahtarlı tabloların (hash table) kullanılarak aday A_k kümeleri daha etkin bir şekilde elde edilmesidir. Özellikle A_2 kümelerinin bulunmasında anahtarlı tabloların kullanımı süreci çok hızlandıracaktır. Diğer bir hızlandırma yöntemi ise artık gerek görülmeyen işlemlerin (transaction) veritabanından silinmesidir. Yani sık ürün kümeleri bulundukça bazı işlemlerin sık ürün kümesi içermediği anlaşıldığında bunlar çalışma yapılan veritabanında keşif esnasında silinebilir. Böylelikle gereksiz taramalar önlenmiş olur.

Veritabanının alt bölümlere ayrılması ile de keşif süreci hızlandırılabilir. Örneğin veritabanı ayrık 4 farklı parçaya bölünmüşse, her bir alt parçada daha hızlı bir keşif süreci olabilir. Elbette alt parçalar için belirlenen minimum destek oranı tüm veritabanı için geçerli olmayacaktır. Alt parçalar için belirlenmiş minimum destek oranını 4 ile çarparsak tüm veritabanı için geçerli olacak minimum destek oranı elde edilebilir. Bu durum aslında hassasiyet ölçüsünde gerileme oluyor ama önemli bir hız kazanımı yaşanabilir. Alt parçalarda bulunan sık ürün kümelerinin bütüne taşınması tüm alt parçadakilerin birleştirilmesi ile gerçekleşir.

Apriori algoritmasına en önemli alternatif aday FP-Growth algoritmasıdır (Han & Kamber, 2006). Bu algoritma, aday kümelerin bulunması adımının ortadan kaldırılması fikrine dayanır. FP-Growth algoritması da veritabanının alt parçalara ayrılıp alt çözümler bulunması (divide-and-conquer) stratejisine dayanır. Apriori algoritmasında olduğu gibi S_1 kümelerinin bulunması için bir kez veritabanı

baştan sona taranır. Daha sonra FP-Agaç yapısı olarak adlandırılan özel veri yapılarının oluşturulabilmesi için tekrar tam bir tarama yapılır. Bu özel veri yapısı üzerinde ön ek (prefix) ve son ek (suffix) kavramları yoluyla yinelemeli (iteratif) olarak FP-Ağaç yapısı büyütülerek keşif işlemi yapılır.

Büyük ölçekli işlem (transaction) verilerinin depolama kaynağı olarak ilişkisel veri tabanları olduğu dikkate alındığında SQL üzerinde de çeşitli tümleştirme (aggregate) sorguları ile sık ürün kümeleri bulunabilir. Bunun yanında Apriori'nin SQL'de uygulamasına örnek olarak (Rajamani v.d., 1999; Sarawagi v.d., 1998) verilebilir.

Eclat (Mohammed J Zaki v.d., 1997) algoritması da farklı bir yaklaşımla derinlik öncelikli bir yaklaşımla sık ürün kümelerini bulmaya çalışır. Veritabanını tek taramada sık kümeleri bulabilmektedir. Apriori ile karşılaştırıldığında hafıza kullanımı anlamında avantajları bulunmaktadır.

Sırasal Veri Analizi

Birliktelik (ilişki) madenciliğinde gerçekleşen işlemlerin sırası veya zamanı dikkate alınmaz. Dolayısıyla incelenen veri zamansal (temporal) değildir. Web sitelerinde tıklanma verilerinin toplandığı log dosyalarının incelenmesi gibi problemlerle karşılaşan erken veri bilimciler başlangıçta Apriori ve benzeri algoritmaları kullanarak aynı oturumda birlikte ziyaret edilen sayfaları kolaylıkla bulmuşlardır. Ancak zamansal iş kuralların bulunmasında yani sayfaların ziyaret edilme sırasını dikkate alan iş kurallarının keşfinde yeni bir problem ile karşılaşmışlardır. Bu tür kuralların Apriori ve benzeri algoritmalarla bulunması (keşfi) gerçekten zordur. Sırasal (zamansal) veriler dikkate alınamadığından bazen geçersiz bazen de anlamsız kuralların keşfi mümkündür. Benzer bir durum elbette gen ve protein gibi biyolojiyi ilgilendiren verilerin incelenmesinde de yaşanmış biyoinformatik gibi alanlarda yeniliklerin ortaya çıkmasına yol açan gelişmeler sağlanmıştır.

Zamansal veriler birçok alanda karşımıza çıkmaktadır. Örneğin gezgin satıcı probleminde karşımıza çıkan şehirlerin ziyaret sırası bile doğası itibariyle zamansal elbette mekânsal yapıya da sahiptir. Tıklama dizisi (click stream) verileri elbette sırasal veri analizi konusunda en erken örneklerdendir. Doğal olarak müşteri bilgisi olan alışveriş (transaction) verileri zamansal olarak ele alındığında da bu tür bir analiz ile karşılaşırız. Örneğin bebek bekleyen bir çiftin yaptığı bazı alışverişler ileride yapılması gereken alışverişlere işaret olabilir. Bu bölümde sırasal verilerin analizinde kullanılan bazı yöntemler hakkında bilgi verilecektir.

Tıklama dizilerin incelenmesinde olduğu gibi bulunan sık sıralı diziler birbirlerini ardışık olarak izlemesi gerekmemektedir. Örneğin bir Telekom operatörünün fatura sayfasına önce bakılabilir, sonra bir ürün satın alınabilir, sonra müşteri temsilcisi ile görüşme yapılabilir. Bu olayların arasında müşteri, Telekom operatörünün farklı web sayfalarını da ziyaret edebilir. Gerçekleşen bu olayların ardışık olması zorunluluğu yoktur. Sadece akış içerisinde birinin daha önlerde veya daha arkalarda olması yeterlidir. Ancak gezgin satıcı probleminde karşılaşılan sıralı diziler ise ardışık bir yapıya sahiptir. Her iki farklı durum bu bölümde incelenecektir.

Sıra (Sequence) Madenciliği

Sırasal verilerin keşfi problemi ilk olarak (Agrawal & Srikant, 1995)'da incelenmiştir. Yazarlar bu tip bir problemi çözmek için üç farklı yaklaşım önermişlerdir (Agrawal & Srikant, 1995). Bu üç yöntem içinden *AprioriAll* algoritması en başarılı bulunmuştur. Daha sonra *AprioriAll* algoritmasına göre 20 kat daha hızlı olan GSP algoritması geliştirilmiştir (Srikant & Agrawal, 1996). GSP yinelemeli bir algoritma olarak k-uzunlukta sıralı dizileri bulabilmesi için k kez verinin tümünü gözden geçirmesi gerekmektedir. GSP, sık sıra dizilerini etkin biçimde sayabilmek için hash (anahtar) ağaçları gibi özel veri yapıları kullanmaktadır. Ayrıca sıralı dizilerin keşfinde kullanılan maksimum ve minimum aralık, kayan pencere kısıtı gibi bazı kavramlar ilk kez (Srikant & Agrawal, 1996)'da kullanılmıştır. Temel

olarak maksimum (minimum) aralık iki olay arasında en fazla (en az) olması gereken olay (mesela sayfa ziyareti) sayısını verir. Kayan pencere kısıtı ise keşif sürecini sadece belirli bir çerçevede (aralıkta) yapılmasını sağlar.

Yukarıda bahsedilen ilk örnekler farklı işlemlerde (transaction) tekrarlanan sıralı dizilerin keşfiyle ilgilidir. Uzun ve tek bir sıralı veride sıklıkla karşılaşılan küçük bölümler (olay zincirleri) ise farklı bir yaklaşım gerektirir (Mannila v.d., 1995; Mannila & Toivonen, 1996). PrefixScan Algoritması sıralı verilerin incelenmesinde kullanılan genel amaçlı ilk başarılı algoritmadır (Pei v.d., 2001). PrefixScan çoklu veritabanı projeksiyonlarını kullanarak etkin bir biçimde sık sıralı dizileri keşfeder.

Temel olarak sıra madenciliği ilişki (birliktelik) madenciliğinin zamansal verilere uygulanmış hali olarak düşünülebilir. İlişki madenciliği olaylar içindeki ilişkilerle yani bir işlem içinde birlikte görülen olaylarla ilgilenmekle birlikte sıra madenciliği olaylar arası ilişkileri keşfeder. Sıra madenciliği için en güzel örneklerden biri müşteri bazında zamansal olarak market sepet verilerinin incelenmesidir. Bu yüzden *AprioriAll* ve GSP gibi erken algoritmalar birliktelik kurallarının keşfi için geliştirilmiş fikirler üzerine kurulmuştur. Örneğin sık sıralı dizilerin belirlenmesinde de destek düzeyi kullanılan bir ölçüdür.

Apriori algoritmasının birçok uygulaması Tablo 0-1'deki gibi verileri yatay şekilde alırlar. Aslında veriler dikey şekilde de işlem no, ürün no (transaction ID, Item ID) çiftleri olarak algoritmaya verilebilir. Aslında bu tür bir format ilişkisel veritabanlarında tutulan veriler için daha uygundur. Dikey formatta veri kullanılarak geliştirilen ilk yöntem SPAM dikey bir bitmap kullanmıştır (J. Ayres v.d., 2002). SPAM derinliğine öncelikli arama yapan ve iki farklı budama adımı olan bir algoritmadır: S ve I adımları. PrefixScan ve SPADE (M J Zaki, 2001a) algoritmaları ile karşılaştırılmıştır (J. Ayres v.d., 2002). Simülasyon verileri üzerinde yapılan deneyler sonucunda özellikle büyük veri setlerinde her iki algoritmadan da daha iyi sonuçlar vermiştir. Aslında SPADE algoritmasının (M J Zaki, 2001a, 2001b) paralel versiyonu olmakla birlikte, SPAM algoritması ile karşılaştırılırken paralel olmayan bir versiyonu kullanılmıştır. Bunun yanında SPAM algoritmasının SPADE'e göre daha fazla hafıza gerektirdiği gözlemlenmiştir. Bunun sebebi SPAM'in tam matris şeklinde veri kullanırken SPADE ise seyrek matris formatına yakın dikey veri kullanmasıdır. Seyrek matris biçimindeki yapılar her zaman için hesaplamalı işlemlerde üstün performans sağlarlar.

SPADE algoritması sıralı dizi keşfini küçük alt problemlere bölerek gerçekleştirdiği için diğer çözümlere göre veritabanını daha az kez baştan sona tam tarar. Yani karmaşıklığı daha azdır ve özel veri yapıları gerektirmez. Veritabanını birinci kez sık olayların keşfi, ikinci kez ise sık olay çiftlerinin (ikili sıra dizileri) keşfinde, üçüncü kez ise diğer uzunluktaki sıralı dizilerin bulunmasında tam olarak tarar. Diğer algoritmalara göre çok etkindir. Sıra madenciliği ile keşfedilen kurallar zamansal özelliğe sahiptir. Örneğin $A \rightarrow B$ kuralının anlamı eğer A olayı gerçekleşirse daha sonrasında B olayı gerçekleşir, diğer bir ifade ile eğer A sayfası ziyaret edilirse aynı oturumun ilerleyen adımlarında B sayfası da ziyaret edilir. Hemen akabinde olması gerekmez, daha sonraki bir adımda da olabilir.

Adım (Path) Madenciliği

En genel anlamıyla bilinen sıra madenciliğinde müşteri numarası, işlem numarası ve gerçekleşen olayların bilinmesi gerekir ki anlamlı ve sık karşılaşılan sıralı diziler keşfedilebilsin. Gerçekleşen olaylar dizisi verildiğinde iki farklı seçenek vardır. İlkinde olayın gerçekleştiği zaman, ikincisinde ise sırayı belirten indeks değeri verilebilir (veya olduğu varsayılabilir). Elbette işlemler her bir müşteri bazında zamansal olarak indekslenmelidir.

Tablo 0-7'de müşteriden bağımsız olarak tutulan işlemlere ait örnek olaylar verilmiştir. İlk işlemde önce A, sonra D ve en sonunda F olayı gerçekleşmiştir. Bu olayların indeks sırası 1, 2 ve 3 olarak doğrudan atanabilir. Bu bölümde indeks olarak veya zamansal olarak ardışık gerçekleşen olayların oluşturduğu sık karşılaşılan sıra dizilerinin keşfi incelenecektir. Bir sonraki olayı adım atmakla denk tuttuğumuzdan bu tip sıra dizinlerinin keşfi adım madenciliği olarak adlandırılmıştır.

İşlem No	Sıralı Olaylar
1	A, D, F
2	A, C, D
3	A, B, D, F, B, D
4	A, C, F, C
5	A, C, B, D
6	A, D, F

Tablo 0-7 Sıralı Dizi Örnek Veri Seti

Olay dizileri verilen bir işlemde birden fazla tekrar edebilirler. Örneğin Tablo 0-7'de 3 nolu işlemde B, D ardışık olaylar iki kez tekrarlamıştır. İlişki madenciliğinde genelde işlem bazında destek düzeyi belirlenir. Fakat sıra madenciliğinde aynı işlemde tekrarlanan sıra dizinleri olabilir. Bu yüzden destek düzeyi (ya da oranı) tüm olay sayılarının üzerinden belirlenebilir. Yani yapılan farklı işlem sayılarından değil, gerçekleşen tüm olayların sayısı üzerinden destek oranı belirlenebilir. Destek düzeyi de sıklık sayımı olarak olayların karşılaşılma sayısını verir (A. Demiriz, 2002, 2005). Bu durumda minimum destek düzeyi tüm veriler içinde bir olayın (veya olay dizisinin) gerçekleşme sayısını verir. Sık tekil olaylar kümesi F_1 olarak tanımlanabilir.

Örnek veri setinde toplam 23 olay gerçekleşmiştir. Minimum destek seviyesi 2 olarak seçildiğinde Tablo 0-8'de verilen sık olaylar kümesi elde edilir. Burada verilen bir işlemde olayın gerçekleşme sırası (anı) indeks numarası olarak alınmıştır. Örneğin A olayı hep birinci adımda gerçekleşmiş olup, indeks numarası 1'dir. Minimum destek oranı da $\frac{2}{23} \approx \%9$ şeklinde bulunabilir.

A	
İ No	İnd.
1	1
2	1
3	1
4	1
5	1
6	1

B	
İ No	İnd.
3	2
3	5
5	3

C	
İ No	İnd.
2	2
4	2
4	4
5	2

D	
İ No	İnd.
1	2
2	3
3	3
3	6
5	4
6	2

F	
İ No	İnd.
1	3
3	4
4	3
6	3

Tablo 0-8 Sık Olaylar F_1

$A \rightarrow$ C	
İ No	İnd.
2	2
4	2
5	2

$A \rightarrow$ D	
İ No	İnd.
1	2
6	2

$B \rightarrow$ D	
İ No	İnd.
3	3
3	6
5	4

D $\rightarrow$ F	
İ No	İnd.
1	3
3	4
6	3

$A \rightarrow$ D $\rightarrow$ F	
İ No	İnd.
1	3
6	3

Tablo 0-9 Sık Adımlar F_2 ve F_3

Tablo 0-7'de verilen örnek veri seti için tüm tekil olaylar, belirlenen minimum destek düzeyinde sık olarak belirlenmiştir. F_1 kümesi Tablo 0-8'de verilmiştir. Artık F_2 ve F_3 yani ikili ve üçlü sıra diziler (adımlar) keşfedilebilir. Bir sonraki adımda F_2 kümesi için adaylar F_1'in kendisi ile birleşmesi ile (join operation $F_1 * F_1$) kolayca bulunabilir. Bu birleşme operatörü için iki şartın sağlanması gerekiyor:

1- Aynı işlemde olmak
2- Bir birim indeks farkının olması (yani ardışık).

Yani işlem numaraları aynı ve indeksleri bir farklı olan kayıtlar eşleştirilebilir. D ve F olayları birleştirilip $D \rightarrow F$ adım dizisi elde edilmeye çalışıldığında yukarıdaki birleşme şartlarını sağlayan

$\{(1:2)(1:3),(3:3)(3:4),(6:2)(6:3)\}$ D ve F çiftleri mevcuttur. Etkin bir biçimde eşleşme sağlanırsa aynı operasyonda F→D adım dizisinin de sık adım olup olmadığı belirlenebilir. Bu örnek için sık olmadığı açıktır. Bunun yanında B→D yolu üçüncü işlemde iki kez alınmıştır. Bu yüzden bu yol üçüncü işlemden iki kayıta sahiptir.

Daha uzun sık yolların keşfi F_2'den biraz farklıdır ve daha az karmaşıktır. F_k'nın bulunması için F_{k-1} ve F_2'nin birleştirilmesi yeterlidir. Örneğin $A \rightarrow D$ ve $D \rightarrow F$ kümelerini birleştirdiğimizde $A \rightarrow D \rightarrow F$ yolunun sık olup olmadığı bulunabilir. F_2 ve F_3 kümeleri Tablo 0-9'da verilmiştir. Bu bölümde gösterilen yöntem hakkında (A Demiriz, 2004)'da daha detaylı bilgi bulunmaktadır.

Öneri Sistemleri

E-Ticaret sitelerinin 1990'lı yıllarda kurulmaya başlanması sonrasında müşterilerin aradıklarını bulamaması veya almak istemedikleri ürünlerle ilgili reklam mesajlarına boğulması, müşterileri ile iyi ilişkiler yürütmek isteyen e-ticaret şirketlerini arayışlara itmiştir. Aslında müşterilere yapılan öneriler geleneksel ticarette satış elemanlarının günlük rutinlerinden biridir. Örneğin bir kitapçıda aradığı kitabı bulamayan bir müşteriye satış elemanları yardım ettiklerinde aranılan kitabın alternatifleri hakkında da bilgi verebilirler. Satış elemanlarının, müşterinin ilgisini daha fazla çekmek için o an aranılan kitabın yanında diğer hangi kitapların da okunmasının faydalı olacağını söylemeleri işlerinin bir parçasıdır. Bu gibi önerilerle karşılaşan bir müşterinin kitapçıdan tek kitap yerine birden fazla kitap satın alıp ayrılması gayet normaldir.

Müşteri İlişki Yönetimi'nin (CRM) bir parçası olan öneri sistemleri, müşterileri ile iyi ilişkiler sürdürmek isteyen e-ticaret siteleri için vazgeçilemez bir araçtır. Müşterilerinin ne tür ürünlerden hoşlanıp, hoşlanmayacağını bilmeleri firmalar için kritik bir bilgidir. Bir müşteriye stoklarımızda olan (veya kolayca tedarik edebileceğimiz) ürünleri önerebileceğimiz için ürün yelpazesinin çeşitliliği, en çok tercih edilen ürünlerin stok seviyelerinin her zaman yeterli tutulması operasyonel anlamda firma için en önemli hedeflerdendir. Bu açıdan müşteri tercihlerinin bilinmesi ve ekonomik faydaya dönüştürülmesi öneri sistemlerinin yanında diğer sistemler için de önemlidir.

Müşteri tercihlerinin rafine bir şekilde iş kurallarına dönüşebilmesi ve kolayca aksiyon alınması sonucunda öneri sistemleri az bulunan bir özellikten ziyade zorunluluk olarak e-ticaretin bir parçası olmuştur. Bu tür sistemlerin özellikle ürün arama esnasında müşteriyi yönlendirmeleri ve çapraz satışa imkân vermeleri yadsınamaz. Hatta Netflix gibi platformlarda artık tüm ürün kataloğu kolayca erişilememekte ve çoğu zaman sadece müşterinin ilgisini çekebilecek ürünler listelenmektedir. Bu tür sistemler adeta ürün aramaya yeni bir kimlik vererek müşterilerin önerilerle yönlendirilmesine imkân sağlamışlardır.

Geleneksel mağazacılıkta rafların kısıtlı alana sahip olması sadece kısıtlı sayıda ürünün teşhirine imkân verir. Bu durumda mağaza satış elemanları ya mevcut ürünleri tavsiye edebilirler ya da diğer mağazalardan kolaylıkla getirebilecekleri ürünleri tavsiye edebilirler. Çok az durumda ise rakip bir mağazaya yönlendirebilirler. Elbette müşterinin eliboş çıkmaması için tüm satış taktiklerine başvurulup alternatif bir ürün ile müşterinin memnuniyeti sağlanabilir. Online mağazacılıkta ise raf kısıtı olmamakla birlikte müşterinin dikkatinin çekilebileceği kısıtlı bir süre sorunu vardır. Ürün arama sonucunda ihtiyaç duyulan ürün veya ürünlere tatmin edici içerikle cevap verilmelidir. Öneri sistemleri bu noktada vazgeçilmez bir araç haline gelmişlerdir.

En basit haliyle öneri sistemleri liste şeklinde sabit önerilerdir. Özellikle e-ticaret sitesinin kategori yönetimi tarafından yayınlanan "Favori Listeleri" veya "Editörün Şeçtikleri" bu tür önerilere örnektir. Bunun yanında bazı iş kurallarının sağlanması durumunda çeşitli öneriler senaryo halinde e-ticaret sitesi üzerinde kurgulanabilir. Bunlar manuel olarak kurgulanan öneri sistemleridir. Bunların yanında özet olarak adlandırılabilecek ve belirlenmiş bir süreyi kapsayacak şekilde Top 10, En Popüler, Son Alınanlar (İzlenenler) listeleri de yayınlanabilir. Genel eğilimi temsil eden bir yapıya sahip olduklarından bu tür listeler müşterilerin ilgilerini çekmektedir.

Öneri sistemleri yukarıda bahsedilen şekilde basit algılanmamalıdır. Geçmiş veriler dikkate alındığında müşteri tercihleri ürünlerin belirli bir skala üzerinden değerlendirilmesi (örneğin izlenilen filmlere verilen puanlar) veya en basit anlamda ürünü satın-alıp almamak açısından takip edilebilir. En temel haliyle bu tür veriler müşteri-ürün matrisi şeklinde bir yapıya sahiptir. Bu yapıdaki veriler üzerinde en basit anlamda korelasyon analizi yapılarak ürünler arasındaki ilişki bulunabilir. Verilerin içeriği dikkate alındığında korelasyon analizinin anlamlı ve uygulanabilir sonuç vereceği şüphelidir. Bu yüzden daha kapsamlı modeller yardımıyla müşteri tercihleri anlamlı bir şekilde faydaya dönüşebilir. Tablo 0-1'de verilen örnek işlemlerin farklı müşteriler tarafından yapıldığını varsayarsak Tablo 0-10'da öneri sistemlerinde kullanılabilecek basit bir veri yapısı verilmiştir.

Müşteri\Ürün	41368	42267	47036	55822
1	1	0	1	1
2	1	1	0	0
3	0	1	1	1
4	1	1	1	1

Tablo 0-10 Örnek Müşteri-Ürün Matrisi

Müşteri-Ürün matrisi özellikle ürünlerin puanlandırması durumunda seyrek bir haldedir. Zira tüm müşterilerin (kullanıcıların) tüm ürünleri (mesela film) değerlendirmesi mümkün olmayabilir. Benzer bir şekilde her bir müşteri stokta bulunan ürünlerin çok az bir kısmını geçmişte satın almış olabilir. Diğer bir deyişle müşteri başına değerlendirmesi yapılmış veya satın alınmış çok az ürün olabilir. Bunun yanında yeni bir ürün eklendiğinde veya yeni bir müşteri kazanıldığında başlangıçta hiç veri olmayacaktır. Müşteri-ürün matrisi çok seyrek olsa bile, öneri sistemleri bu tür verilerden yola çıkılarak geliştirilir.

Müşteri-ürün matrisi göz önüne alındığında probleme hangi açıdan yaklaştığımıza göre iki ana öneri sistemi mevcuttur. Eğer müşteriler arasındaki benzerlikten yola çıkarsak yani müşteri-ürün matrisinde satırlarda işlem yaparsak müşteri-tabanlı öneri sistemi geliştirilmiş olur. Diğer taraftan, matrisin sütunları arasında benzerlikler dikkate alınarak bir sistem geliştirilirse ürün-tabanlı öneri sistemi geliştirilmiş olur. Müşteri ekseninde bir model geliştirilirse öneri yapılmak istenen müşteriye benzer müşterilerin davranışları bizim için belirleyici olur. İlgilenen müşteriye benzer müşterilerin bulunması gerekir. Ürün ekseninde bir model geliştirilmişse mevcut ürünler dikkate alındığında bulunması/alınması/puanlanması en yüksek olabilirliğe sahip ürünler önerilir. Bu durumda ürün ekseninde benzerlikler öne çıkar.

Bir müşteri profilindeki tüm ürünleri (öğeleri) ya da puanlanan tüm öğelerin içeriklerini dikkate alarak öneri sistemleri geliştirebiliriz. Bu durumda ürünlerin (öğelerin) içerikleri hakkında daha detaylı bir veri hazırlanması gerekir ve içeriklerden yola çıkılarak ürünler arasındaki ilişkiler keşfedilmelidir. Burada içerik denildiğinde aslında o öğeyi tanımlayan özellikler (öznitelik) akla gelmelidir. Bu özellikler bazında öğeler arasında ilişkiler ortaya çıkarılır. Bu tip öneri sistemleri içerik bazlı öneri sistemi olarak adlandırılır. Örneğin metinler üzerine bir öneri sistemi geliştiriliyorsa kullanıcının (müşterinin) geçmişte beğendiği öğelerin içeriğine yakın yeni metinler önerilebilir. Bu durumda sadece müşteri-ürün matrisi değil ürünlerin özelliklerinin bulunduğu farklı bir veri setine de ihtiyaç duyulmaktadır.

Öneri sistemleri için birçok yaklaşım vardır. Bunların önde gelenleri olarak Bayesci Ağlar, korelasyon analizi, vektör benzerliği ve birliktelik madenciliği sayılabilir. Müşteri veya ürün eksenli komşuluk yöntemleri de yaygın olarak karşılaşılan yöntemlerdir. Bu bölümde yaygın kullanılan yöntemler hakkında bilgi verilecektir.

Müşteri Ekseninde Öneri Sistemleri

Kabul gören yaygın görüşe göre benzer zevkleri olan kişiler benzer tercihlerde bulunacaktır. Profil olarak, örneğin geçmiş ürün tercihleri bakımından, birbirlerine benzer müşteriler birbirleri için davranış bakımından önemli işaretler taşımaktadırlar. Diğer bakımdan ürün profilleri dışında çeşitli demografik

ya da farklı alanlardan müşteriler hakkında toplanan veriler de müşteri profili oluşturulmasına yardımcı olabilir. Bu tür bir veri seti yani salt müşteri-ürün matrisi haricindeki profil verisi mevcut ise en basit yapılacak modelleme müşterilerin segmentasyonu ile birbirlerine benzer müşteri gruplarının bulunmasıdır. Oluşan müşteri grupları içinde popüler ürünler dikkate alınarak grup içinde ürün önerileri basit bir şekilde yapılabilir. Burada kabul edilen en önemli varsayım birbirlerine benzer müşteri profillerinde tercih edilen ürünler benzer olacaktır.

Diğer bir yaklaşım ise en basit anlamında müşteriler arasında vektör benzerliğinden yani Öklit uzaklığından yola çıkarak birbirine benzer (yakın) olan müşterilerin benzer ürünleri tercih ettikleri göz önüne alınarak öneri yapılır. Öklit uzaklığı elbette müşteri-ürün matrisinde ikili (0-1) verilerin olduğu düşünüldüğünde başarılı bir benzerlik ölçütü olmayabilir. Özellikle ikili verilerden oluşan iki vektör arasında benzerlik hesaplamanın en sağlam yolu bu vektörler arasındaki açının kosinüsünü hesaplamaktır. Böylelikle iki vektörün birbirlerine ne kadar benzer oldukları anlaşılabilir. Tam örtüşen vektörler arasında 0°, zıt vektörler arasında ise 180° bir açı olup, ilgili kosinüs değerleri 1 ile -1 arasında değişecektir. Denklem 12.1, iki vektör arasında kosinüs değerini hesaplar.

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\left||\vec{a}|\right|_2 * \left|\left|\vec{b}\right|\right|_2} \qquad (12.1)$$

Kosinüs değerinin hesaplanmasının yanında iki vektör arasında korelasyon hesaplanarak birbirine benzer müşteriler kolayca bulunabilir. Pearson Korelasyon katsayısı Denklem 12.2'de verilmiştir.

$$corr_{a,b} = \frac{\sum_i (u_{ai} - \overline{u_a})(u_{bi} - \overline{u_b})}{\sqrt{\sum_i (u_{ai} - \overline{u_a})^2 \sum_i (u_{bi} - \overline{u_b})^2}} \qquad (12.2)$$

Her iki benzerlik de kolayca hesaplansa bile milyonlarca müşterinin olduğu bir sistem düşünüldüğünde müşteri tabanlı yaklaşımlar için benzerliklerin hesaplanması ve komşulukların bulunması neredeyse imkânsız hale gelebilir. Bu yüzden yaygın kullanımı için özel veri yapılarının (kd-Ağaçları gibi) kullanılması gerekmektedir.

Ürün Ekseninde Öneri Sistemleri

Denklem 12.1 ve 12.2'de verilen ölçütler ürün ekseninde de öneri sistemlerinin uygulamasında kullanılabilir. Müşteri-ürün matrisi göz önüne alındığında sütun bazında benzerlikler bulunmalıdır. Ürünler arasında benzerlikler bulunduktan sonra tahmin adımında müşterinin profilindeki ürünlere en benzer ürünler tavsiye edilebilir.

Müşteriyle yapılan bir işlem sırasında, CRM sisteminin parçası olan öneri sistemi eğer İlk-5 (Top-5) ürün şeklinde bir tavsiyede bulunuyorsa, müşteri profilindeki ürünlere en çok benzeyen 5 ürün müşteriye sunulur. Bu tür ürün sunumları (önerileri) bazı iş kuralları ve senaryoları bütünlüğünde yapılır. Ürün temelli öneri sistemleri geniş ölçekli müşteri veritabanı durumunda çok etkin öneri sistemleridir (G. Karypis, 2001; Sarwar v.d., 2001). Koşullu olasılık hesapları da ürünler arasındaki benzerliklerin bulunmasında kullanılabilir. Geniş ölçekli verileri için hem kosinüs hem de koşullu olasılık hesapları verilerdeki bazı ürünlerin fazla karşılaşılması durumlarının olumsuz etkilerini kaldırmak için değiştirilebilir (G. Karypis, 2001).

Bu kısımda bahsedilen müşteri ve ürün eksenli sistemler müşteri-ürün matrisi üzerinde gerçek zamanlı işlem yapılmasını gerektirdiğinden hafıza temelli öneri sistemleri olarak da adlandırılır. Bir sonraki bölümde daha çok model temelli bir yapıya sahip olan İşbirliğine Dayalı Filtreleme öneri sistemleri hakkında bilgi verilecektir.

İşbirliğine Dayalı Filtreleme

İşbirliğine dayalı filtreleme yöntemleri de ürün ekseninde geliştirilen model tabanlı öneri sistemleridir. Bu terim ilk olarak (Goldberg v.d., 1992)'de bahsedilmiş olsa da konu ile ilgili öncü çalışmalar (Resnick & Varian, 1997)'de geniş bir şekilde tanıtılmıştır. Bu ilk önemli yayında GroupLens (Konstan v.d., 1997) ve Siteseer (Rucker & Polanco, 1997) projeleri öne çıkmıştır. GroupLens Usenet Gruplarında mesaj tavsiye ederken, Siteseer ise kullanıcıların sık kullandıkları (bookmark) web sitelerine göre web sayfası önermekteydi. Bu ilk projelerde en önemli kabul kullanıcılar arasında önemli benzerliklerin olmasıydı.

Müşteri-ürün matrisi verilerini ele aldığımızda ürünler arasındaki nedensel ilişkilerin modellenmesi için en doğal yaklaşım Bayesci Ağlardır. Diğer bir deyişle bu tür bir yaklaşım ile ürünler arasındaki ilişkilerden yola çıkılarak öneri sistemleri kolaylıkla geliştirilebilir. Ancak Bayesci Ağlar ile bulunan ilişkilerin uzman olmayan bir analist tarafından yorumlanması çok güçtür. Bu yüzden ürünler ya da değişkenler arasındaki koşulllu olasıklardan faydalanan Bağımlılık Ağları (Dependency Networks) geliştirilmiştir (Heckerman v.d., 2000). Bağımlılık ağları ile tahmin ve korelasyona dayanan ilişkiler anlaşılabilir şekilde kullanıcıya sunularak aksiyon alınabilecek kurallar elde edilebilir (Heckerman v.d., 2000; J. Breese v.d., n.d.). Bunun yanında Bayesci ağlara göre bağımlılık ağlarının bulunması hesaplama açısından daha az karmaşıktır. Bir ürün (veya değişkenin) diğerleri verildiğindeki koşullu dağılımı herhangi bir olasılıksal sınıflandırma veya regresyon yöntemi ile kolaylıkla bulunduğundan (Heckerman v.d., 2000) bağımlılık ağlarının oluşturulması ve görselleştirilmesi daha basittir.

Bağımlılık ağlarının nasıl çalıştığını göstermek için şu örneğe bakalım. Farz edelim ki ilgilendiğimiz ürün kümesi $\mathcal{X} = (X_1, X_2, X_3)$ olsun. Bu durumda yapmamız gereken $p(x_1|x_2, x_3)$, $p(x_2|x_1, x_3)$ ve $p(x_3|x_1, x_2)$ olmak üzere üç koşullu olasılık dağılımının bulunmasıdır. Bu dağılımları lojistik regresyon, yapay sinir ağları veya karar ağaçları ile kolayca bulabiliriz. Kullandığımız sınıflandırma yönteminin değişken seçebilme özelliği olduğunu kabul edelim. Örneğin açıklayıcı olması açısından X_1, X_3 için ve X_3 de X_1 için tahmin edici (predictor) değişken olmadığı kullandığımız sınıflandırma yöntemi tarafından bulunsun. Bu durumda bağımlılık ağı $X_1 \leftrightarrow X_2 \leftrightarrow X_3$ şeklinde kolaylıkla bulunabilir. Bu şekilde ürünler (değişkenler) arasında ilişkiler kolaylıkla bulunup, öneri sisteminde kullanılabilir. Örneğin profilinde X_3 ürünü olan müşteriye X_2 ürünü tavsiye edilirken X_1 ürünü tavsiye edilmez.

Elbette müşteri-ürün matrisinin çok seyrek olduğu durumlarda öneri sistemlerinin başarılı sonuçlar vermesi zor hale gelebilir (A. Demiriz, 2004). Bu durumda müşteri -ürün matrisi gruplandırarak veya köşegenleştirilerek alt müşteri-ürün grupları bulunarak yani belirli müşteri grubunun tercih ettiği alt ürün grupları bulunarak seyrekliği daha az alt kümeler (matrisler) elde edilebilir. Müşterilere bu alt problemleri dikkate alan öneri sistemleri ile hizmet verilebilir (A. Demiriz, 2004).

Uygulama Örnekleri

Bu bölümde bahsedilen bazı algoritmaların R platformundaki uygulamaları hakkında bazı örnekler aşağıda verilmiştir. İhtiyaç duyulan R paketleri şu şekilde yüklenebilir.

```
install.packages("arules","arulesSequences")
library(arules)
library(arulesSequences)
```

Örnek veri setleri aşağıdaki şekilde web sitesinden erişilebilir. Burada ilişki madenciliği ve sıra madenciliği için iki farklı veri seti paylaşılmıştır. Bu veri setleri daha önce başka bir kaynakta

paylaşılmamıştır. Veri dosyaları (A. Demiriz, 2002; A. Demiriz, 2004) eserlerinde bahsedilen sistemlerde kullanılan verilerin anonimize edilip örneklenmesi ile elde edilmişlerdir.

```
if(!file.exists("sampleProdData.txt")) {
download.file("https://www.ayhandemiriz.com/sampleProdData.txt",
"sampleProdData.txt")
}
if(!file.exists("clickDataSpade.txt")) {
download.file("https://www.ayhandemiriz.com/clickDataSpade.txt",
"clickDataSpade.txt")
}
```

İlk dosyada bulunan veriler `arules` (Hahsler, n.d.) paketinin read.transactions fonksiyonu ile aşağıdaki şekilde yüklenebilir. Verilerin dikey formatta olduklarına dikkat edelim. Bu yüzden "`single`" opsiyonu kullanılmıştır.

```
tr <- read.transactions("sampleProdData.txt", format = "single",
cols = c(1,2))
summary(tr)
```

`Summary` fonksiyonu ile veri seti incelendiğinde 130379 işlem verisi, uzunluğu 2 ile 20 ürün arasında değişecek şekilde dağılmaktadır. Yani iki ile yirmi arasında uzunluğu olan işlemler mevcuttur. En sık görülen ürünler 40, 15, 8, 1 ve 9 nolu ürünlerdir. `Summary` fonksiyonun çıktısı aşağıdaki gibidir.

```
transactions as itemMatrix in sparse format with
 130379 rows (elements/itemsets/transactions) and
 51 columns (items) and a density of 0.0825563

most frequent items:
     40      15       8       1       9 (Other)
  76695   70777   49252   42278   30116  279826

element (itemset/transaction) length distribution:
sizes
    2     3     4     5     6     7     8     9    10    11    12    13    14
29852 29313 23752 17152 12107  7980  4901  2639  1435   672   334   130    79
   15    16    17    18    20
   16     9     4     3     1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    3.00    4.00    4.21    5.00   20.00
```

```
includes extended item information - examples:
  labels
1      1
2     10
3     11

includes extended transaction information - examples:
  transactionID
1    1000093026
2    1000100049
3    1000100820
```

İki farklı destek oranı için (0.05 ve 0.01) sık ürün kümeleri aşağıdaki şekilde bulunur. `inspect` fonksiyonu ile bu kümeler ekrana listelenebilir. İlk durumda 89 farklı küme varken, ikinci durumda ise 814 sık küme bulunmaktadır.

```
itemsets <- apriori(tr,parameter = list(supp = 0.05, conf = 0.55,
target = "frequent itemsets"))
inspect(itemsets)


itemsets <- apriori(tr,parameter = list(supp = 0.01, conf = 0.55,
target = "frequent itemsets"))
inspect(itemsets)
```

Farklı destek ve güven oranları kullanılarak apriori algoritması yardımıyla birliktelik kuralları aşağı şekilde elde edilebilir. Daha büyük destek seviyesi için 56 kural bulunurken, diğer destek oranı için 878 kural bulunmuştur.

```
rules <- apriori(tr,parameter = list(supp = 0.05, conf = 0.55,
target = "rules"))
inspect(rules)


rules <- apriori(tr,parameter = list(supp = 0.01, conf = 0.55,
target = "rules"))
inspect(rules)
```

Eclat algortiması kullanılarak sık ürün kümeleri ve birliktelik kuralları aşağıdaki şekilde bulunabilir.

```
itemsets <- eclat(tr,parameter = list(supp = 0.01, maxlen = 5))
inspect(itemsets)

rules <- ruleInduction(itemsets, tr, confidence = .55)
inspect(rules)
```

Sıralı verilerin analizi için `arulesSequences` paketi yardımıyla çeşitli analizler yapabiliriz. Bu pakette SPADE (M J Zaki, 2001a) algoritması kullanılmıştır. Kullandığımız örnek veride 61522 farklı oturumda ziyaret edilen 936 sayfaya ait tıklanma verileri bulunmaktadır. Verilerin seyrekliğini ifade etmek açısından yoğunluk oranı 0.001068376 olarak rapor edilebilir. Bu durumda tüm matris göz önüne alındığında yaklaşık olarak %99,9 oranında seyreklik söz konusudur.

```
clkData <- read_baskets(con =  "clickDataSpade.txt",
info = c("sequenceID","eventID","SIZE"))

summary(clkData)

freqseqs <- cspade(clkData, parameter = list(support = 0.05),
control = list(verbose = TRUE))

inspect(freqseqs)

seqrules <- ruleInduction(freqseqs, confidence = .5,
control = list(verbose = TRUE))

inspect(seqrules)
```

SPADE algoritması 0.05 destek oranı ile kullanıldığında 103 sık sayfa kümesi bulunmuştur. Bu sık kümelerden 0.5 güven oranı kullanıldığında 28 adet kural bulunmuştur. Kural çıkarımı için `ruleInduction` fonksiyonu kullanılmıştır. `arulesSequences` paketi ile test verisi de sağlanmıştır. Aşağıda bu test verisinin yüklenmesi ve analizi ile ilgili kod verilmiştir. Test verisinde 90748 farklı işlemde 77 farklı ürüne ait veriler bulunmaktadır. Yoğunluk 0.03452103'tür. Dolayısıyla yaklaşık olarak %96.65 oranında seyreklik söz konusudur. 0.5 güven ve 0.0133 destek seviyelerinde 63961 kural bulunmuştur.

```
tst <- read_baskets(con = system.file("misc", "test.txt", package
="arulesSequences"),
info = c("sequenceID", "eventID", "SIZE"))
summary(tst)

seqtst <- cspade(tst, parameter = list(support = 0.0133),
control = list(verbose = TRUE))
summary(seqtst)

k <- support(seqtst, tst, control = list(verbose = TRUE))
table(size(seqtst), sign(quality(seqtst)$support -k))

rls <- ruleInduction(seqtst, confidence = .5,
control = list(verbose = TRUE))
inspect(rls)
```

REFERANSLAR

Goldberg, D., Nichols, D., Oki, B. M. ., & Terry, D. . (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, *35*(12), 61–70.

Heckerman, D., Chickering, D., Meek, C., Rounthwaite, R., & C. Kadie. (2000). Dependency Networks for Density Estimation, Collaborative Filtering, and Data Visualization. *Journal of Machine Learning Research*, *1*(Oct), 49–75.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, *40*(3), 77–87.

Resnick, P., & Varian, H. R. (1997). Recommender Systems. *Communications of the ACM*, *40*(3), 56–58.

Rucker, J., & Polanco, M. J. (1997). Personalized Navigation for the Web. *Communications of the ACM*, *40*(3), 73–75.

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., & Verkamo, A. I. (1996). Fast Discovery of Association Rules. In U. Fayyad & et al (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 307–328). AAAI Press, Menlo Park, CA.

Agrawal, R., & Srikant, R. (1995). Mining Sequential Patterns. *11th Intl. Conf. on Data Engg.*

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th VLDB Conference,Santiago, Chile*, 487–499.

Demiriz, A. (2002). webSPADE: a parallel sequence mining algorithm to analyze web log data. *2002 IEEE International Conference on Data Mining, 2002.*

Demiriz, A. (2005). On analyzing web log data: A parallel sequence-mining algorithm. In *Next Generation of Data-Mining Applications*. https://doi.org/10.1109/9780471696650

Demiriz, A. (2004). Asipath: A simple path mining algorithm. *In Proceedings of the 16th International Conference on Parallel and Distributed Computing and Systems (PDCS 2004)*, 9.

Demiriz, Ayhan. (2004). Enhancing product recommender systems on sparse binary data. *Data Mining and Knowledge Discovery*, *9*(2), 147–170.

G. Karypis. (2001). Evaluation of Item-Based Top-N Recommendation Algorithms. *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM)*.

Hahsler, M. (n.d.). *arules R Paketi*. 20 Mayıs 2021'de erişilmiştir https://michael.hahsler.net/research/arules_RUG_2015/demo/

Han, J., & Kamber, M. (2006). *Data Mining Concepts and Techniques* (2nd ed.). Morgan Kaufmann.

J. Ayres, J. Gehrke, T. Yiu, & J. Flannick. (2002). Sequential Pattern Mining using a Bitmap Representation. In D. Hand, D. Keim, & R. Ng (Eds.), *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press.

J. Breese, D. Heckerman, & Kadie, C. (n.d.). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, 43–52.

Mannila, H., & Toivonen, H. (1996). Discovering generalized episodes using minimal occurences. *2nd Intl. Conf. Knowledge Discovery and Data Mining*.

Mannila, H., Toivonen, H., & Verkamo, I. (1995). Discovering frequent episodes in sequences. *1st Intl. Conf. Knowledge Discovery and Data Mining*.

Pei, J., Han, J., Pinto, H., Chen, Q., Dayal, U., & Hsu, M.-C. (2001). PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. *Proceedings of International Conference on Data Engineering (ICDE'01)*.

Rajamani, K., Cox, A., Iyer, B., & Chadha, A. (1999). Efficient Mining for Association Rules with Relational Database Systems. *Proceedings of the International Database Engineering and Applications Symposium (IDEAS'99)*.

Sarawagi, S., Thomas, S., & Agrawal, R. (1998). Integrating association rule mining with relational database systems: Alternatives and implications. *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, 343–354.

Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the Tenth International World Wide Web Conference on World Wide Web*, 285–295.

Savasere, A., Omiecinski, E., & Navathe, S. B. (1998). Mining for Strong Negative Associations in a Large Database of Customer Transactions. *ICDE*, 494–502.

Srikant, R., & Agrawal, R. (1996, March). Mining Sequential Patterns: Generalizations and performance improvements. *5th Intl. Conf. Extending Database Technology.*

Tan, P.-N., Kumar, V., & Kuno, H. (2001). Using SAS for Mining Indirect Associations in Data. *Western Users of SAS Software Conference*.

Zaki, M J. (2001a). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning Journal*, *42*(1/2), 31–60.

Zaki, M J. (2001b). Parallel Sequence Mining on Shared-Memory Machines. *Journal of Parallel and Distributed Computing*, *61*(3), 401–426.

Zaki, Mohammed J, Parthasarathy, S., Ogihara, M., & Li, W. (1997). New Algorithms for Fast Discovery of Association Rules. *KDD*.

Ayhan Demiriz, 1972 yılında Erzincan'da doğdu. 1989-1993 yılları arasında İstanbul Teknik Üniversitesi İşletme Fakültesi Endüstri Mühendisliği bölümünde lisans eğitimini tamamladı. 1995-2000 yılları arasında Decision Sciences and Engineering Systems alanındaki bütünleşik doktora çalışmasını NY eyaletinin Troy şehrinde bulunan Rensselaer Polytechnic Institute'de yaptı. 2000 ve 2004 yılları arasında Verizon isimli Telekom şirketinde yazılım mühendisi olarak çalıştı. 2004 ve 2016 yılları arasında Sakarya Üniversitesi Endüstri Mühendisliği Bölümü'nde farklı kadrolarda akademisyen olarak görev yapmıştır. 2016 Ekim ayından bu yana Gebze Teknik Üniversitesi Endüstri Mühendisliği Bölümü'nde kurucu bölüm başkanı ve Profesör olarak görev yapmaktadır. Kendisinin ayrıca Verikar Yazılım şirketi altında bazı girişimleri bulunmaktadır.