Veri Madenciliğinde Sepet Analizi Uygulamaları

Market Basket Analysis for Data Mining

Mehmet Aydın Ulaş, Ethem Alpaydın

(Boğaziçi Üniversitesi Bilgisayar Mühendisliği)

Nasuhi Sönmez, Ataman Kalkan

(GİMA Türk A.Ş.)

ÖZET

Elektronik ticaret uygulamalarının da gelişmesiyle firmaların elinde yüksek miktarda veri birikmiş oldu. Bu yüksek miktardaki ham veriden hızlı bilgi çıkarmak için kullanılan yöntemlerden bir tanesi de Veri Madenciliğidir. Veri Madenciliğinin amacı veri içerisinde bağıntılar bulmaktır.

Sepet Analizi Veri Madenciliğinin ana uygulama alanlarından bir tanesidir. Sepet Analizinin amacı verilen satış hareketleri üzerinden mallar arasında ilintiler ortaya çıkarmaktır.

Sepet analizi Gima Türk A.Ş.'nin çeşitli mağazalarından alınan satış verilerinin bir kısmı üzerinde uygulandı ve ümit verici sonuçlar alındı.

ABSTRACT

With the e-commerce applications growing rapidly, the companies have a significant amount of data in their hands. Data Mining is one of the methods for extracting useful information from this raw data. The aim of Data Mining is to find relations in the data.

Mining Association Rules (Market Basket Analysis) is one of the main application areas of Data Mining. Given a set of customer transactions on items, the aim is to find correlations between the sales of items.

We applied Market Basket Analysis to some subset of the data taken from some stores of Gima Türk A.Ş. and we obtained promising results.

1. GİRİŞ

Veritabanlarında "Bilgi Keşfi" olarak da bilinen Veri Madenciliğinin amacı ileriki aşamalardaki kararlara yardımcı olması için veri içerisinde yönsemeler, örüntüler, ilintiler ve sapaklıklar bulmaktır. Veri Madenciliği sihir değildir. Veri Madenciliği sadece uzmanlara ileride daha iyi kararlar verebilmelerine yardımcı olmaktadır. Veri Madenciliği Veritabanları, Yapay Zekâ ve Yapay Öğrenme konularının bir kesişimidir. Veri Madenciliğine örnek olarak:

- Sepet Analizi (İlişki Madenciliği): Sepet analizinin amacı bir veri kümesi içerisinde bağıntılar bulmaktır.

- Sınıflama: Sınıflama belli özniteliklere bakarak veriyi önceden belli olan sınıflardan birisine vermektir.
- Regresyon: Regresyon verinin bazı özelliklerini kullanarak diğer özelliklerini tahmin etmek ya da veriyi kullanarak belli sonuçlar çıkarmak için kullanılır.
- Öbekleme: Öbekleme veriyi daha önceden belli olmayan sınıflar içerisine dağıtmaktır. Öbeklemede amaç öbek içi farklılıkları azaltmak ve öbeklerarası farklılıkları artırmaktır.

Son zamanlarda İlişki Madenciliği olarak da bilinen sepet analizi Veri Madenciliğinin üzerinde yoğunlaşılan konularından birisi oldu. İlişki Madenciliği ilk olarak [1]'de tanıtıldı. Çok fazla hareket olan bir market veritabanını düşünelim. *"X → Y"* bir ilişki kuralı, *"X"* ***neden*** (antecedent) ve *"Y"* ***sonuç*** (consequent) olarak adlandırılır. *"X"* ve *"Y"* birer öğekümedir (itemset) ve kural *X* alanların *%c*'si *Y* almaktadır demektir. Burdaki *c* sayısına ***güven*** (confidence) denmekte ve *X* alanların yüzde kaçının *Y* aldığını belirtmektedir. Örnek olarak "Sigara alanların yüzde sekseni kibrit almaktadır" gibi kurallar çıkarılabilir. Bu kurallar kullanılarak "Coca Cola ne ile satılıyor?" gibi sorulara cevap bulunabilir, ayrıca *A* ve *B* maddeleri arasındaki bağıntılar merak ediliyorsa neden kısmı A olan ve sonuç kısmı B olan kurallar bulunabilir.

Genel amaç, verilen bir müşteri hareket veritabanını kullanarak kurallar çıkarmaktır. Veritabanlarının boyutları çok büyük olduğu için genel olarak algoritmalar programları hızlandırmak üzerine kurulmuştur.

1.1. Problemin Tanımı

$I = (i_1, i_2, .., i_n)$ bir hareket kümesi (transaction set) olsun. Her i_i ***öğe*** (item) olarak adlandırılır. Tüm hareketlerin kümesine *D* denir ve *T* bir hareket ve $T \subset D$ olmak üzere her *T* bir ***öğekümedir*** (öğeler kümesidir). Her hareketin ***TID*** denilen tek bir numarası vardır. İçinde *k* öğe bulunan öğekümeye ***k-öğeküme*** (k-itemset) denir. *X* ve *Y* birbirinden farklı birer öğeküme olsun. Bir *X* öğekümesinin ***desteği*** (support) *X* öğekümesini kapsayan kümelerin sayısının tüm öğekümelere oranıdır. *|X|, X*'i kapsayan öğekümelerin sayısı; *|D|,* tüm öğekümelerin sayısı ve *|X.Y|* de *X* ve *Y*'yi kapsayan öğekümelerin sayısı olsun. Bir öğekümenin desteği aşağıdaki gibi tanımlanır:

$$Destek(X) = \frac{|X|}{|D|}$$

Eğer *X* alanların *%s* kısmı *Y* almış ise *X→ Y* kuralının desteği *X* ve *Y*'yi beraber alanların sayısının tüm öğekümelerin sayısına oranıdır.

$$Destek(X \Rightarrow Y) = \frac{|X.Y|}{|D|}$$

Destek bir kuralın istatistiksel olarak önemini güven ise kuralın kuvvetini belirtmektedir. Eğer *X* içeren hareketlerin *%c* kadarı aynı zamanda *Y* de içeriyorsa *X → Y* kuralının güveni *c* denir.

$$Güven(X \Rightarrow Y) = \frac{Destek(XY)}{Destek(X)}$$

D hareketler kümesi verildiğinde sepet analizinin amacı destek ve güvenleri önceden belirlenmiş minimum güven ve minimum destek değerlerinden daha büyük olan tüm *X → Y* kurallarını çıkarmaktır. Bir öğekümenin desteği eğer minimum destek değerinden daha büyükse bu öğekümeye ***geniş*** (large) denir.

İlişki Madenciliğini iki adıma ayırabiliriz: İlk adımda minimum destek kullanılarak tüm geniş öğekümeler çıkartılır ve ikinci adımda minimum güven ve geniş öğekümeler kullanılarak tüm kurallar çıkartılır.

İlişki Madenciliğini gerçekleyen algoritmalar veri üzerinde birden çok kere geçerler. Çoğu algoritma ilk önce geniş öğekümeleri bulur ve daha sonra da bunları kullanarak kuralları çıkartır. Geniş öğekümeleri bulmak için öğekümenin boyu birer birer artırılır ve daha sonra tüm hareketler üzerinden sayılarak bu

öğekümenin geniş olup olmadığı kontrol edilir. Zor olan kısım geniş öğekümeleri bulmak olduğu için araştırmalar genelde bu alanda yoğunlaşmıştır.

Problem birçok açıdan birçok algoritmayla çözülmeye çalışılmıştır. [1]'de Agrawal, Imielinski ve Swami destek ve güven kavramlarını tanıtmışlar ve sonucunda sadece bir öğe olan bir algoritma gerçeklemişlerdir.

[2]'de Agrawal ve Srikant Apriori ve AprioriTID algoritmalarını tanımlamış ve sonucunda birden fazla öğe olabilen bu algoritmaları gerçeklemişlerdir. Apriori algoritmasını ileriki bölümlerde daha açık bir şekilde anlatıp Gima Türk A.Ş.'nin verilerinin bir kısmı kullanılarak elde edilen sonuçları vereceğiz.

2. YÖNTEM

Testlerimizde Apriori algoritmasını kullanarak ilişki kurallarını çıkardık.

2.1. Apriori Algoritması

Geniş Öğekümelerin Bulunması

Apriori Algoritması şu şekilde çalışmaktadır: Önce desteği minimum destekten daha büyük olan 1-öğekümeler bulunur. Bütün öğeler teker teker tüm hareketler üzerinden sayılır ve desteği minimum destekten daha büyük desteği olan 1-öğekümeler seçilir. Daha sonra Apriori_Gen algoritması kullanılarak geniş 2-öğekümeler bulunur. Daha sonra bir budama işlemi kullanılarak 2-öğekümeler azaltılır. Daha sonra 3-öğekümeler bulunarak bu şekilde budanacak öğeküme kalmayana kadar devam edilir. Öğekümeler sözlük sıralamasıyla sıralanmıştır.

Öğeküme Oluşturma

Bir *k-öğekümeye m* öğe daha eklersek yeni öğekümeye ilk öğekümenin ***m-uzatması*** (m-extension) denir. k-öğekümeleri kullanarak (k+1)-öğekümeleri şu şekilde yaratıyoruz: Tüm k-öğekümeler ikişer ikişer ele alınarak ilk k-1 öğelerinin aynı olup olmadıklarına bakılır. Eğer aynı iseler ilk öğeküme ile ikinci öğekümenin son öğesi ile 1-uzatmasını alarak yeni k+1-öğeküme oluşturulur.

Tablo 1'de bu algoritmaya bir örnek görülmektedir. Minimum desteği 0.5 alalım. İlk önce tüm 1-öğekümeler sayılır ve desteği minimum destekten küçük olanlar L1'e alınmazlar. Daha sonra L1 kullanılarak C1 oluşturulur ve C2'deki tüm öğekümeler sayılır. Görüldüğü gibi {1, 2} ve {1, 5} destekleri düşük olduğu için elenirler. Yaratılabilecek tek öğeküme {2, 3, 5}'dir. Bu öğekümeyi L3'de sayarız. Daha başka öğeküme kalmadığı için algoritma burada sonra erer.

Budama

X bir öğeküme olsun. Eğer X geniş bir öğeküme ise $Y \subset X$ olacak şekilde tüm Y öğekümeleri de geniştir. Bu fikri kullanarak her k+1-öğeküme oluşturulduğunda bu öğekümenin tüm k-öğeli altkümelerinin (k-item subset) geniş olup olmadığı kontrol edilir. Eğer herhangi bir altkümesi geniş değilse bu öğekümeyi tüm öğekümeler içinde saymaya gerek kalmadığından bu öğeküme budanır.

Kural Oluşturma

Bütün geniş öğekümeler oluşturulduktan sonra bu öğekümeler kullanılarak kurallar çıkartılır.

Tablo 1. Apriori Örneği

D

TID	Öğeler
100	1, 3, 4
200	2, 3, 5
300	1, 2, 3, 5
400	2, 5

C_1

Öğeküme	Destek
1	0.5
2	0.75
3	0.75
4	0.25
5	0.75

L_1

Öğeküme	Destek
1	0.5
2	0.75
3	0.75
5	0.75

C_2

Öğeküme	Destek
1, 2	0.25
1, 3	0.5
1, 5	0.25
2, 3	0.5
2, 5	0.75
3, 5	0.5

L_2

Öğeküme	Destek
1, 3	0.5
2, 3	0.5
2, 5	0.75
3, 5	0.5

C_3

Öğeküme	Destek
2, 3, 5	0.5

L_3

Öğeküme	Destek
2, 3, 5	0.5

Eğer *a → (I – a)* kuralı oluşturulamıyorsa $a' \subset a$ olacak şekilde *a' → (I – a')* kuralları üretilemez. Algoritma bu fikre dayanmaktadır. Örneğin *X = {A, B, C}* ve *Y = {D}* olsun. Eğer *{A, B, C} → D* kuralı çıkarılamıyorsa *{A, B} → {C, D}* kuralı çıkarılamaz çünkü ilk kural her zaman ikincisinden daha fazla güvene sahiptir. Yukarıdaki ifadeyi *(I – r) → r* kuralını çıkarabilmek için $r' \subset r$ olacak şekilde *(I – r') → r'* kuralını çıkarabilmek gerekir şeklinde ifade edebiliriz. Yukarıdaki örneğe dönersek *{A, B} → {C, D}* kuralını çıkarabilmek için *{A, B, C} → {D}* ve *{A, B, D} → C* kurallarının ikisinin de çıkarılabilmesi gerekir.

3. VERİ KÜMESİ VE SONUÇLAR

Testlerimizde GİMA Türk A.Ş.'nin Haziran 2000 ve Ağustos 2000 arasında bir şubesinden alınan verilerinin bir kısmını kullandık. Tablo 2 ve Tablo 3'de testlerde kullanılan veritabanındaki tabloların özellikleri gözükmektedir.

Tablo 2. Fis_Baslık

Alan Adı	Alan Tanımı
Tarih	Hareketin Tarihi
Kasa_No	Kasanın Numarası
Fis_No	Fişin Numarası
Musteri_No	Müşterinin Numarası

Tablo 3. Fis_Detay

Alan Adı	Alan Tanımı
Tarih	Hareketin Tarihi
Kasa_No	Kasanın Numarası
Fis_No	Fişin Numarası
Fis_Sirano	Malın Fiş İçindeki Sırası
Mal_No	Malın Numarası
Miktar	Malın Miktarı

Kullanılan veritabanında 756,868 hareket, 140,610 öğeküme ve 7,237 çeşit mal (öğe) bulunmaktadır. Her öğe ortalama olarak 105 öğekümede yer almaktadır. Veritabanında toplam 9,985 kayıtlı müşteri bulunmaktadır.

Kullanılan veriler yaz ayında alınmış olduğu için en çok satılan ürünler domates, ekmek, salatalık gibi ürünler bunun yanında da yumurta, karpuz gibi ürünler çıktı. Bu verilerden yola çıkarak insanlarımızın yaz aylarında özellikle hafif yemeklere yöneldiklerini özellikle salatanın çok fazla tüketildiğini bunun yanında domatesin ve yumurtanın çok fazla satılmasından menemenin de çok tüketilen yemekler arasında olduğunu ayrıca meyvelerin de yazın yüksek miktarda tüketildiğini söyleyebiliriz. “Salatalık alanların yüzde altmış yedisi domates almaktadır” veya “Semizotu ve domates alanların yüzde elli beşi maydanoz almaktadır” kurallara örnek olarak gösterilebilir.

Çalışmalarımız halen değişik mal gruplarının ve şubelerin hareketlerinin karşılaştırılması yönünde sürmektedir.

KAYNAKÇA

[1] Agrawal, R., Imilienski, T., Swami, A. N. (1993). Mining Association Rules Between Sets of Items in Large Databases. Proceedings of the ACM SIGMOD International Conference on Management of Data, 207 – 216, Mayıs 1993.

[2] Agrawal, R., Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. Proceedings of the 20th International Conference on Very Large Databases (VLDB), Eylül 1994.

[3] Ganti, V., Gehrke, J., Ramakrishnan, R. (1999). Mining Very Large Databases. IEEE Computer 32, 8, 6875, Ağustos 1999.

[4] Hipp, J., Güntzer, U., Nakhaeizadeh, G. (2000). Algorithms for Association Rule Mining – A General Survey and Comparison. SIGKDD Explorations, 2, 2000.