

# Veri Madenciliği

## Bölüm 4. Birliktelik Kuralları

Doç. Dr. Suat Özdemir

<http://ceng.gazi.edu.tr/~ozdemir>

### İlişkilendirme/Birliktelik Kuralları - Association Rules

- Birliktelik kuralları olarak da bilinir
- İlişkilendirme kuralı madenciliği
  - Veri kümesi içindeki **yaygın örüntülerin** (pattern) ve nesneleri oluşturan **öğeler arasındaki ilişkilerin** bulunması
  - *"Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction"*
- İlişkilendirme kurallarının kullanım alanları
  - Hangi ürünler çoğunlukla birlikte satılıyor?
  - Yeni bir ilaca duyarlı olan DNA tipleri hangileridir?
  - Web dokümanları otomatik olarak sınıflandırılabilir mi?

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Birliktelik Kuralları

- Bilgisayar alan bir müşterinin aynı zamanda antivirus yazılımı almasına ait ilişkilendirme kuralı
  - **bilgisayar → antivirüs\_yazılımı [support=%2, confidence=%60]**
  - Support = destek (müşterilerin %2'si bilgisayar ve antivirüs yazılımını beraber almışlar)
  - Confidence = güven (bilgisayar alan müşterilerin %60'ı antivirüs yazılımı da almış)
- şeklinde gösterilir
- Geçerli olabilmesi için minimum support ve confidence değerlerini sağlaması gerekir

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Tanımlar

- Bütün öğelerden oluşan küme  $I = \{i_1, i_2, \dots, i_d\}$ 
  - $I = \{\text{ekmek, süt, bira, kola, yumurta, bez}\}$
- Transaction/Hareket  $T_j \subseteq I$ ,
  - $T_1 = \{\text{ekmek, süt}\}$
- Hareketlerden oluşan veri kümesi
  - $D = \{T_1, T_2, \dots, T_N\}$

Sepet hareketleri  
Market-Basket transactions

TID	Öğeler
1	Ekmek, Süt
2	Ekmek, Bez, Bira, Yumurta
3	Süt, Bez, Bira, Kola
4	Ekmek, Süt, Bez, Bira
5	Ekmek, Süt, Bez, Kola

İlişkilendirme kuralları örnekleri

$\{\text{Bez}\} \rightarrow \{\text{Bira}\},$   
 $\{\text{Süt, Ekmek}\} \rightarrow \{\text{Yumurta, Kola}\},$   
 $\{\text{Bira, Ekmek}\} \rightarrow \{\text{Süt}\},$

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Tanımlar

- Öğeler kümesi (Itemset)
  - Bir veya daha çok öğeden oluşan küme
  - k-öğeler kümesi (k-itemset): k öğeden oluşan küme
    - 3-öğeler kümesi: {Bez, Bira, Ekmek}
- Destek sayısı  $\sigma$  (Support count)
  - Bir öğeler kümesinin veri kümesinde görülme sıklığı
  - $\sigma(\{\text{Süt, Ekmek, Bez}\}) = 2$
- Destek  $s$  (Support)
  - Bir öğeler kümesinin içinde bulunduğu hareketlerin toplam hareketlere oranı
  - $s(\{\text{Süt, Ekmek, Bez}\}) = 2/5$
- **Yaygın öğeler (Frequent itemset)**
  - Destek değeri **minimum support** eşik değerinden daha büyük ya da eşit olan öğeler kümesi

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## İlişkilendirme Kuralları Oluşturma

- Veri kümesi D içinden en az, en küçük **destek** ve **güven** değerine sahip  $X \rightarrow Y$  şeklinde kuralların bulunması
    - $X \subset I, Y \subset I, X \cap Y = \emptyset$
    - X ve Y yaygın öğe kümeleri
- Örnek:  $\{\text{Süt, Bez}\} \rightarrow \{\text{Bira}\}$

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## İlişkilendirme Kuralları Değerlendirme

### ▪ Kuralları değerlendirme ölçütleri

- **Destek (support) s:** XUY öğeler kümesinin bulunduğu hareket sayısının toplam hareket sayısına oranı

Örnek:

$\{\text{Süt,Bez}\} \Rightarrow \text{Bira}$

$$s = \frac{\sigma(\text{Süt,Bez,Bira})}{|T|} = \frac{2}{5} = 0.4$$

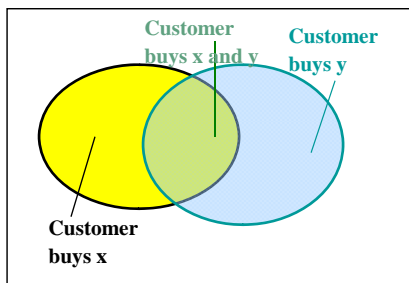
- **Güven (confidence) c:** XUY öğeler kümesinin bulunduğu hareket sayısının X öğeler kümesi bulunan hareket sayısına oranı

$$c = \frac{\sigma(\text{Süt,Bez,Bira})}{\sigma(\text{Süt,Bez})} = \frac{2}{3} = 0.67$$

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Örnek

Transaction-id	Items bought
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F



- Find all the rules  $X \rightarrow Y$  with minimum support and confidence

- support,  $s$ , probability that a transaction contains  $X \cup Y$
- confidence,  $c$ , conditional probability that a transaction having  $X$  also contains  $Y$

Let  $sup_{min} = 50\%$ ,  $conf_{min} = 50\%$

Freq. Pat.: {A:3, B:3, D:4, E:3, AD:3}

Association rules:

$A \rightarrow D$  (60%, 100%)

$D \rightarrow A$  (60%, 75%)

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Geçerli İlişkilendirme Kuralları Oluşturma

- İlişkilendirme kuralları madenciliğinde temel amaç D hareket kümesinden kurallar oluşturmak
  - kuralların destek değeri, belirlenen en küçük destek ( $\min_{sup}$ ) değerinden büyük ya da eşit olmalı
  - kuralların güven değeri, belirlenen en küçük güven ( $\min_{conf}$ ) değerinden büyük ya da eşit olmalı
- **Brute-force yaklaşım**
  - Olası bütün kuralları listele
  - Her kural için destek ve güven değeri hesapla
  - $\min_{sup}$  ve  $\min_{conf}$  eşik değerlerinden küçük destek ve güven değerlerine sahip kuralları sil
  - Hesaplama maliyeti yüksek

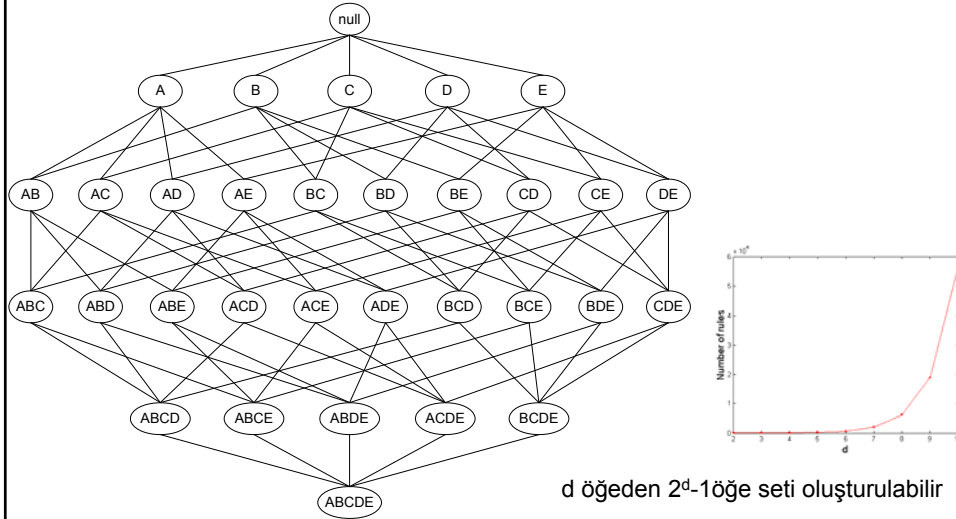
Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Geçerli İlişkilendirme Kuralları Oluşturma

- İki adımda gerçekleşir
- **1. Yaygın öğeleri belirleme:**
  - destek değeri minimum support değerinden büyük ya da eşit olan öğeler kümelerini bulma
- **2. Kural Oluşturma:**
  - Güven değeri  $\min_{conf}$  değerinden büyük ya da eşit olan ve yaygın öğelerin ikili bölünmeleri olan kurallar oluşturma
  - Güçlü kurallar

Veri Madenciliği  
Doç. Dr. Suat Özdemir

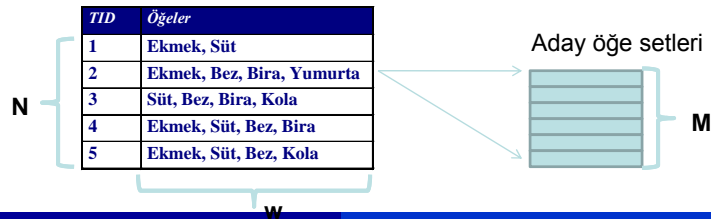
## Geçerli İlişkilendirme Kuralları Oluşturma



Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Yaygın Öğe Adayları Oluşturma

- Yaygın öğeleri bulmak maliyetli
  - $d$  öğe için  $2^d - 1$  öğe oluşabilir
  - Bu da  $2^d - 1$  yaygın öğe adayı demektir
- Brute-Force Yaklaşım
  - Her yaygın öğe adayı için veri kümesini taranarak hareketlerde yaygın öğe adayı bulunup bulunmadığı kontrol edilir
  - Yaygın öğe adayları için destek değeri bulunur
  - Destek değeri  $\min_{sup}$  değerine eşit yada büyük olanlar yaygın öğeler
  - Karmaşıklığı:  $O(NMw)$ ,  $M=2^d - 1 \Rightarrow$  hesaplaması maliyetli



Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Yaygın Öğe Oluşturma Yöntemleri

- 3 temel yaklaşım var
  - Apriori yöntemi (Agrawal & Srikant@VLDB'94)
  - Freq. pattern growth yöntemi (FPgrowth—Han, Pei & Yin @SIGMOD'00)
  - Vertical data format yöntemi (Charm—Zaki & Hsiao @SDM'02)

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Apriori yöntemi

- Apriori yöntemi
  - Rakesh Agrawal, Ramakrishnan Srikant, Fast Algorithms for Mining Association Rules, Proc. 20th Int. Conf. Very Large Data Bases, VLDB'94
  - Heikki Mannila, Hannu Toivonen, Inkeri Verkamo, Efficient Algorithms for Discovering Association Rules . AAAI Workshop on Knowledge Discovery in Databases (KDD-94).
- Temel yaklaşım:
$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$
  - Bir öğeler kümesinin destek değeri, bu kümenin herhangi bir altkümesinin destek değerinden büyük olamaz
  - anti-monotone özellik

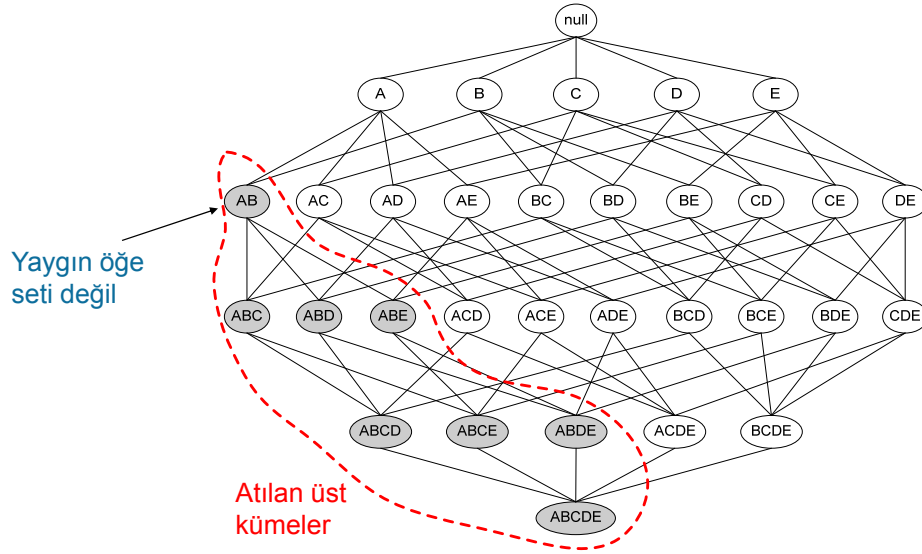
Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Apriori yöntemi

- Çıkarım: Bir yaygın ögenin herhangi bir altkümesi de yaygın ögedir
  - {Süt, Bez, Bira} kümesi yaygın öge ise {Süt, Bez} kümesi de yaygın ögedir
  - {Süt, Bez, Bira} ögeleri kümesi olan her harekette {Süt, Bez} kümesi de vardır
- Yaygın öge aday sayısını azaltma yöntemi: Yaygın öge olmayan bir kümenin üst kümeleri yaygın öge adayı olarak oluşturulmaz (destek değeri hesaplanmaz)
  - Yöntem: k-yaygın ögeleri kümesinden (k+1) yaygın öge adayları oluştur yaygın öge adayları için destek değeri hesapla

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Apriori yöntemi



Veri Madenciliği  
Doç. Dr. Suat Özdemir



**Algorithm 6.2.1 (Apriori)** Find frequent itemsets using an iterative level-wise approach.

**Input:** Database,  $D$ , of transactions; minimum support threshold,  $min\_sup$ .

**Output:**  $L$ , frequent itemsets in  $D$ .

**Method:**

```

1)  $L_1 = \text{find\_frequent\_1-itemsets}(D)$ ;
2) for ( $k = 2$ ;  $L_{k-1} \neq \phi$ ;  $k++$ ) {
3)    $C_k = \text{apriori\_gen}(L_{k-1}, min\_sup)$ ;
4)   for each transaction  $t \in D$  { // scan  $D$  for counts
5)      $C_t = \text{subset}(C_k, t)$ ; // get the subsets of  $t$  that are candidates
6)     for each candidate  $c \in C_t$ 
7)        $c.\text{count}++$ ;
8)   }
9)    $L_k = \{c \in C_k | c.\text{count} \geq min\_sup\}$ 
10) }
11) return  $L = \cup_k L_k$ ;

```

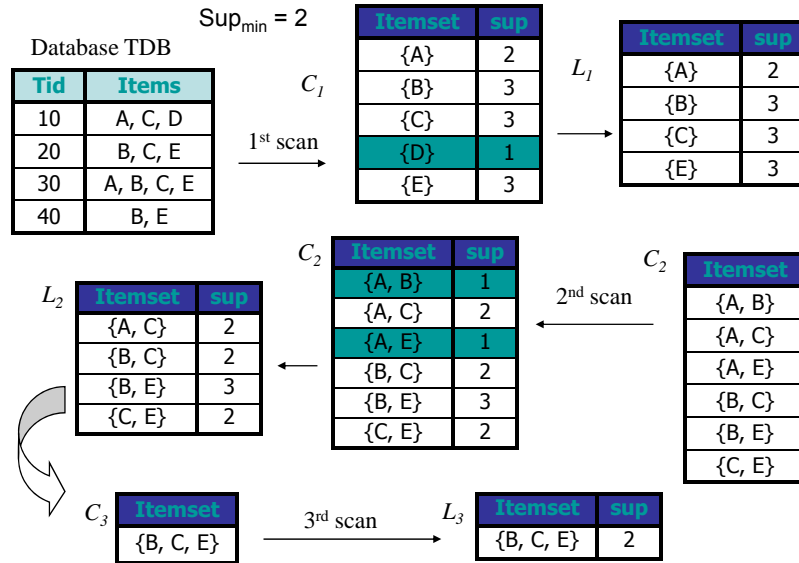
procedure  $\text{apriori\_gen}(L_{k-1}:\text{frequent } (k-1)\text{-itemsets}; min\_sup:\text{minimum support})$

```

1) for each itemset  $l_1 \in L_{k-1}$ 
2)   for each itemset  $l_2 \in L_{k-1}$ 
3)     if ( $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \dots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$ ) then
4)        $c = l_1 \sqcup l_2$ ; // join step: generate candidates
5)       if  $\text{has\_infrequent\_subset}(c, L_{k-1})$  then
6)         delete  $c$ ; // prune step: remove unfruitful candidate
7)       else add  $c$  to  $C_k$ ;
8)   }
9) return  $C_k$ ;

```

## Örnek 1



## Örnek 2

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Items (1-itemsets)



Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

Pairs (2-itemsets)

(Coke ve Eggs içeren setler dahil edilmedi)

Minimum Support = 3

Tüm öge setleri dahil olursa,

$${}^6C_1 + {}^6C_2 + {}^6C_3 = 6 + 15 + 20 = 41$$

Destek değerine göre azaltılırsa,

$${}^6C_1 + {}^4C_2 + {}^3C_3 = 6 + 6 + 1 = 13$$



Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3



Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Apriori yöntemi - Özet

- Yaygın öge setlerini belirleme
- Uzunluğu k=1 olan yaygın öge setlerini oluştur
- Yeni yaygın öğeseti kalmayana kadar tekrarlar
  - k uzunluğundaki yaygın öge setlerinden (k+1) uzunluğundaki aday öge setlerini oluştur
  - k uzunluğunda yaygın olmayan altkümelere sahip aday öge setlerini ele
    - Veri tabanını tarayarak aday öge setlerinin support countlarını hesapla
    - Yaygın olmayan aday öge setlerini ele ve sadece yaygın olanlarla devam et

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Yaygın Öğelerden İlişkilendirme Kuralları Oluşturma

- Sadece güçlü ilişkilendirme kuralları oluşuyor
  - Yaygın öğeler  $\min_{\text{sup}}$  değerini sağlıyor
- Güçlü ilişkilendirme kuralları  $\min_{\text{conf}}$  değerini sağlıyor.
- Yöntem: Her yaygın öğeler kümesi  $f$  'in altkümelerini ( $s$ ) oluştur
  - Eğer  $\text{destek}(f) / \text{destek}(s) \geq \text{minconf}$  ise, altküme  $s$  için,  $s \rightarrow (f-s)$  ilişkilendirme kuralı oluştur

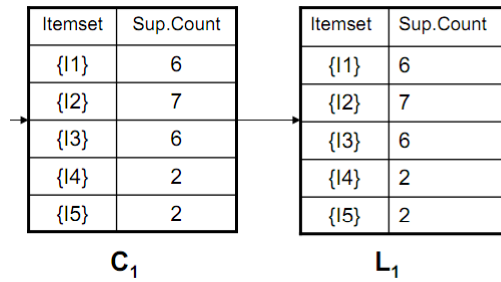
Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Örnek 3

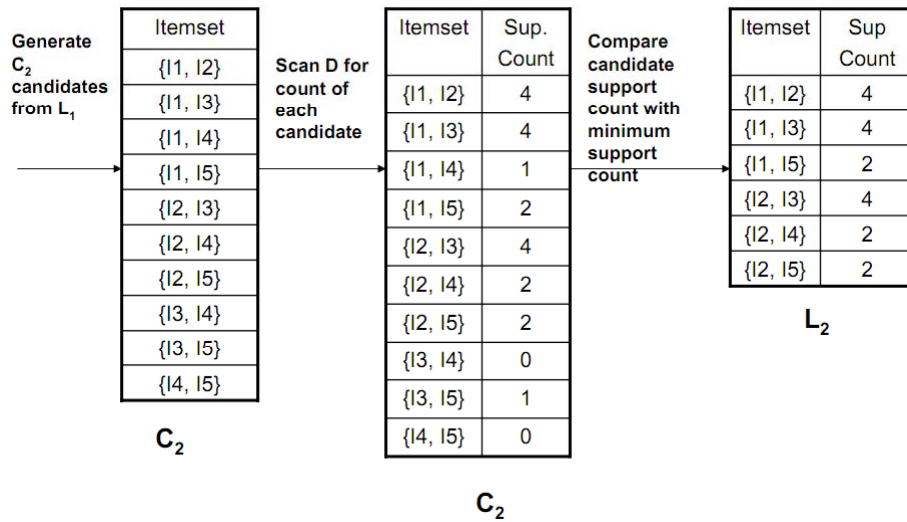
TID	List of Items
T100	I1, I2, I5
T100	I2, I4
T100	I2, I3
T100	I1, I2, I4
T100	I1, I3
T100	I2, I3
T100	I1, I3
T100	I1, I2, I3, I5
T100	I1, I2, I3

- Min support count 2
- Min confidence %70
- Apriori ile ilişkilendirme kurallarını oluşturunuz

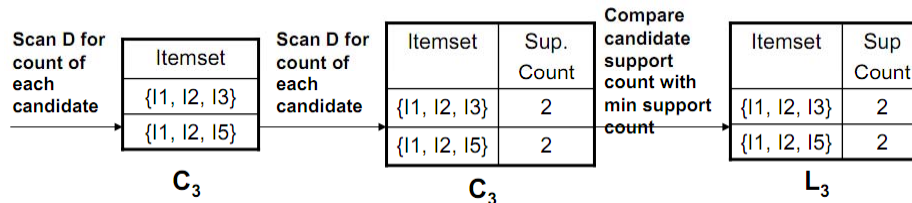
Veri Madenciliği  
Doç. Dr. Suat Özdemir



Veri Madenciliği  
Doç. Dr. Suat Özdemir



Veri Madenciliği  
Doç. Dr. Suat Özdemir



We had  $L = \{\{I1\}, \{I2\}, \{I3\}, \{I4\}, \{I5\}, \{I1,I2\}, \{I1,I3\}, \{I1,I5\}, \{I2,I3\}, \{I2,I4\}, \{I2,I5\}, \{I1,I2,I3\}, \{I1,I2,I5\}\}$ .

- Lets take  $I = \{I1,I2,I5\}$ .
- Its all nonempty subsets are  $\{I1,I2\}, \{I1,I5\}, \{I2,I5\}, \{I1\}, \{I2\}, \{I5\}$ .

Veri Madenciliği  
Doç. Dr. Suat Özdemir

The resulting association rules are shown below, each listed with its confidence.

- R1:  $I1 \wedge I2 \rightarrow I5$ 
  - Confidence =  $sc\{I1,I2,I5\}/sc\{I1,I2\} = 2/4 = 50\%$
  - R1 is Rejected.
- R2:  $I1 \wedge I5 \rightarrow I2$ 
  - Confidence =  $sc\{I1,I2,I5\}/sc\{I1,I5\} = 2/2 = 100\%$
  - R2 is Selected.
- R3:  $I2 \wedge I5 \rightarrow I1$ 
  - Confidence =  $sc\{I1,I2,I5\}/sc\{I2,I5\} = 2/2 = 100\%$
  - R3 is Selected.

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Apriori-Problemler

- Tekrarlı veri tabanı taramaları : Tüm veritabanını taramak zor
- $i_1 i_2 \dots i_{100}$  itemsetini bulmak için
  - gerekli tarama sayısı: 100
  - Aday sayısı:  $\binom{100}{1} + \binom{100}{2} + \dots + \binom{100}{100} = 2^{100} - 1 = 1.27 * 10^{30} !$
- Bottleneck: aday-üretme-ve-test
- Aday üretmeden yapabilir miyiz?

## Aday Oluşturmada Yaygın Öğeleri Belirleme

- Ana fikir → Kısa yaygın öğelere yeni öğeler eklenerek daha uzun yaygın öğeler elde etme
- Örnek: "abc" bir yaygın öğe
  - Veri kümesinde içinde "abc" öğeleri bulunan hareketler (DB|abc) olsun
  - Eğer (DB|abc) içinde d yaygın öğe olarak bulunursa: "abcd" yaygın öğe olarak belirlenir

## FP-Tree Algoritması

1. DB bir kez taranarak 1-yaygın öge bulunuyor
2. Yaygın ögeler destek sayısına göre büyükten küçüğe sıralanıyor, f-list
3. DB bir kez daha taranarak FP-ağacı oluşturuluyor.

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## FP-Ağacının Özelliği

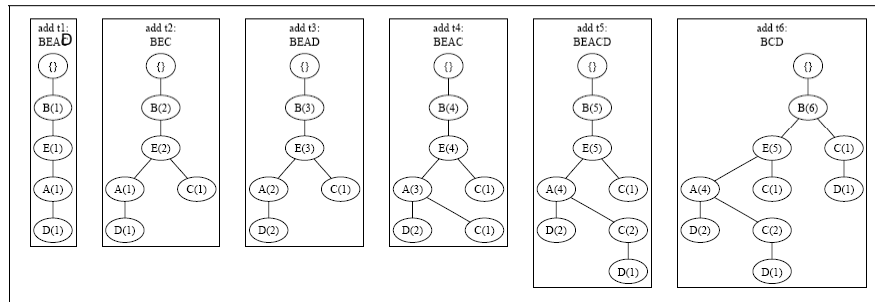
- Bütünlük
  - Yaygın ögeleri bulmak için gerekli tüm bilgiyi barındırır
- Sıkıştırılmış
  - Yaygın olmayan ögeler FP-ağacında bulunmaz
  - Destek sayısı daha büyük olan ögeler köke daha yakın
  - Asıl veri kümesinden daha büyük değil

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## FP-Tree Algoritması: Örnek 1

	$i(t)$
1	ABDE
2	BCE
3	ABDE
4	ABCE
5	ABCDE
6	BCD

F-list  $\{B(6), E(5), A(4), C(4), D(4)\}$

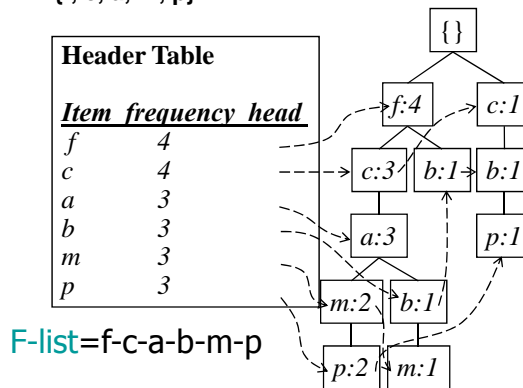


Veri Madenciliği  
Doç. Dr. Suat Özdemir

## FP-Tree Algoritması: Örnek 2

TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

$min\_support = 3$



Veri Madenciliği  
Doç. Dr. Suat Özdemir



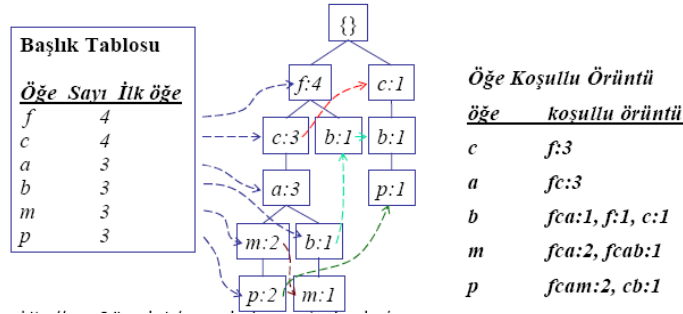
## Örüntüleri ve Veri Kümesini Bölme

- Yaygın öğeler f-listesine göre altkümelere bölünür
  - F-list=f-c-a-b-m-p
  - p öğesi bulunan örüntüler
  - m öğesi bulunan ancak p öğesi bulunmayan örüntüler
  - ...
  - c öğesi bulunan ancak a, b, m, p öğesi bulunmayan örüntüler
  - f öğesi bulunan örüntüler

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Öğeler Koşullu Örüntü Oluşturma

- Başlık tablosundan her öğenin bulunduğu ilk düğüm bulunur.
- Bu düğümden başlayarak ağaçta öğenin bulunduğu tüm düğümlere ulaşılır
- Kökten öğeye kadar olan yollar bulunur (transformed prefix paths)



Veri Madenciliği  
Doç. Dr. Suat Özdemir

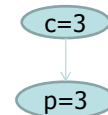
## İçinde 'p' olan tüm örüntülerin bulunması

- Başlık tablosunun en altından başla
- p:3'e tamamlanana kadar tüm yolları bul
- 'p' nin görüldüğü yollar: (f:4, c:3, a:3, m:2, p:2) ve (c:1, b:1, p:1)
- Bu yolların içinde geçen p sayısına göre değerleri (içinde p olan yolların sayısı)
  - (f:2, c:2, a:2, m:2, p:2) ve (c:1, b:1, p:1)

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## İçinde 'p' olan tüm örüntülerin bulunması

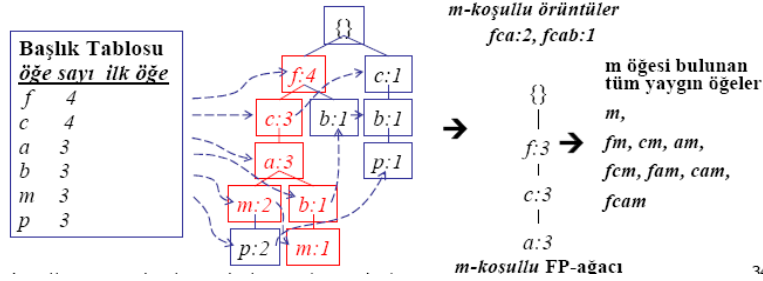
- 'p' için koşullu FP ağacı oluşturmak için 'p' yollardan atılır
  - (f:2, c:2, a:2, m:2) ve (c:1, b:1)
  - Zaten p nin bu yolların üzerinde olduğunu biliyoruz
- İçinde 'p' olan yaygın öge setlerini bulmak için koşullu FP ağacındaki yaygın öge setleri bulunur ve 'p' eklenir.
- Koşullu FP ağacı oluşturulurken yine support count lara bakılır ve eleme yapılır
  - Elimizde sadece c:3 kaldı.
  - İçinde 'p' geçen yaygın öge setleri p:3 ve cp:3



Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Koşullu FP Ağaçları Oluşturma

- Her koşullu örüntü için öğelerin destek sayısı bulunur
- Yaygın öğeler için koşullu FP-ağacı oluşturulur



Veri Madenciliği  
Doç. Dr. Suat Özdemir

## FP-Ağaçları ile Yaygın Öğeleri Bulma

- Yaygın öğelere yinelemeli olarak yeni öğeler ekleme
- Yöntem:
  - Her yaygın öğenin koşullu örüntülerini ve koşullu FP ağacını oluştur
  - İşlemi yeni oluşturulan her koşullu FP-ağacı için tekrarla
  - Oluşturulan FP-ağaçlarında hiç öğe bulunmayana kadar veya ağaçta tek bir dal kalana kadar işleme devam et
    - Ağaçta tek bir dal kaldığında yaygın öğeler dalı oluşturan öğelerin kombinasyonu

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## İlişkilendirme Kuralları Oluşturma

- L yaygın öğelerden  $f \subset L$  altkümelerinin bulunması  $f \rightarrow L - f$  kurallarının en küçük güven değeri koşulunu sağlaması gerekir
- Eğer  $\{A, B, C, D\}$  yaygın öğeler ise olası ilişkilendirme kuralları
  - $ABC \rightarrow D, ABD \rightarrow C, ACD \rightarrow B, BCD \rightarrow A,$   
 $A \rightarrow BCD, B \rightarrow ACD, C \rightarrow ABD, D \rightarrow ABC,$   
 $AB \rightarrow CD, AC \rightarrow BD, AD \rightarrow BC, BC \rightarrow AD,$   
 $BD \rightarrow AC, CD \rightarrow AB$

Veri Madenciliği  
Doç. Dr. Suat Özdemir

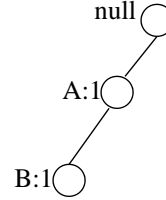
## FP-Tree Algoritması: Örnek 3

Veri Madenciliği  
Doç. Dr. Suat Özdemir

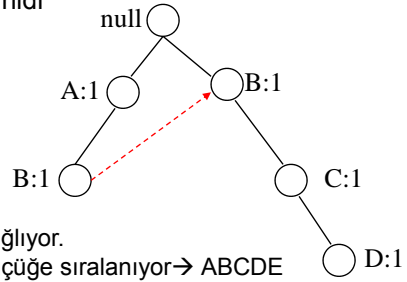
## FP-tree oluşturma

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

TID=1 yerleştirildi



TID=2 yerleştirildi



Tablodaki bütün öğeler  $\text{sup}_{\min}=2$  koşulunu sağlıyor.

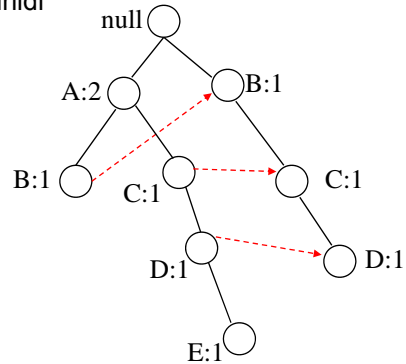
Öğeler support countlarına göre büyükten küçüğe sıralanıyor  $\rightarrow$  ABCDE

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## FP-tree oluşturma

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

TID=3 yerleştirildi



Veri Madenciliği  
Doç. Dr. Suat Özdemir

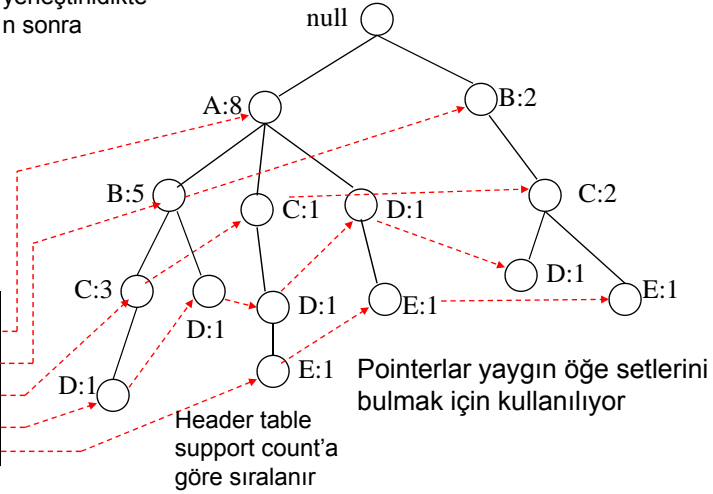
## FP-tree oluřturma

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Bütün item'lar  
yerleřtirildikten  
sonra

Header table

Item	Pointer
A	
B	
C	
D	
E	



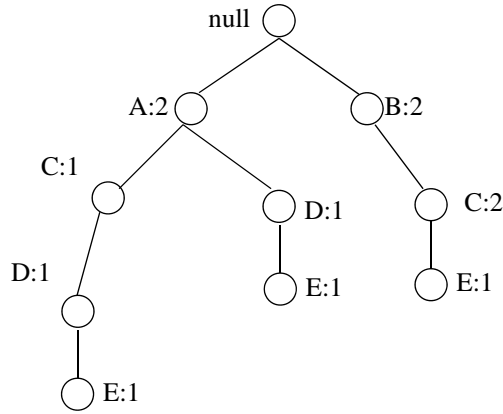
Veri Madencilięi  
Doç. Dr. Suat Özdemir

## FP-tree ile yaygın öęe seti bulma

- Header table'ın en altından bařlayarak, her item için kořullu FP-tree ler oluřturulur. {E} ile bařlanır
- Bu aęaęlar üzerinde eklemeler yapılarak ({D,E} gibi) oluřturulabilecek tüm yaygın öęe setleri bulunur (divide-and-conquer).
- Header table daki bütün öęeler için bu iřlem yapıldığında tüm yaygın öęe setleri bulunmuř olur.

Veri Madencilięi  
Doç. Dr. Suat Özdemir

## Yaygın öge seti bulma örnek



E ile biten yolları gösteren ağaç

$\text{sup}_{\min}=2$  olsun

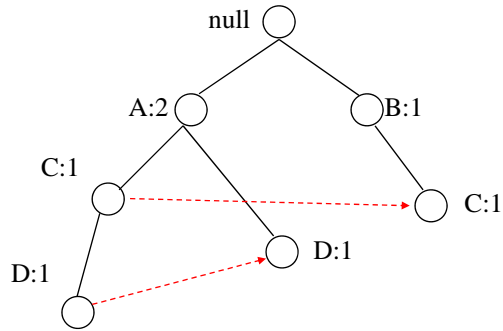
E için Conditional Pattern base (E koşullu öge setleri):

$P = \{(A:1, C:1, D:1), (A:1, D:1), (B:1, C:1)\}$

P üzerinde FP-tree algoritması recursive olarak uygulanır

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Yaygın öge seti bulma örnek



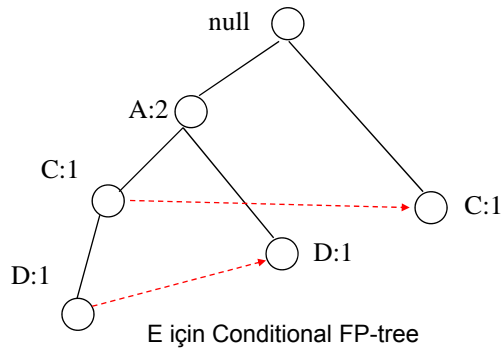
P den ağaç üretilir.

Conditional Pattern basedeki tüm setler ağaca yerleştirildi

$\text{sup}_{\min}=2$  olduğu için B ağaçta yer almamalı.

Veri Madenciliği  
Doç. Dr. Suat Özdemir

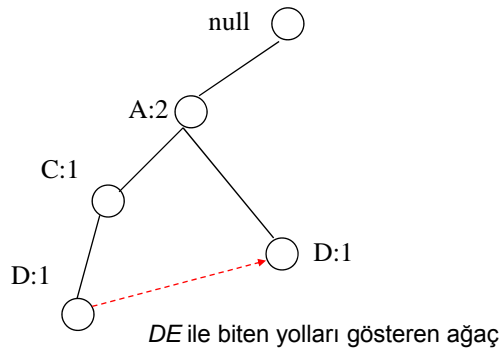
## Yaygın öge seti bulma örnek



{E} yaygın öge seti olarak bulundu. Şimdi bu ağaç üzerinden 2 elemanlı yaygın öge setlerini bulmak için FP-tree bir kez daha çalışır.

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Yaygın öge seti bulma örnek



{D,E}, {C,E} ve {A,E} koşullu yaygın öğeleri bulmak için *DE*, *CE* ve *AE* ile biten yollar bulunur.

Yandaki ağaçta D nin support countu 2 olduğu için {D,E} yaygın öge seti olarak bulunur.

DE için Conditional Pattern base (DE koşullu öge setleri):

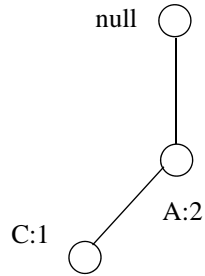
$$P = \{(A:1, C:1), (A:1)\}$$

DE için Conditional Tree oluşturulur

Veri Madenciliği  
Doç. Dr. Suat Özdemir



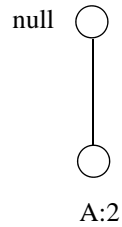
## Yaygın öge seti bulma örnek



C nin support countu 1 olduğu için ağaçta olmamalı

Veri Madenciliği  
Doç. Dr. Suat Özdemir

## Yaygın öge seti bulma örnek



DE için Conditional FP-Tree

A nın support countu 2 olduğu için {A,D,E} yaygın öge seti olarak bulunur.

Sırada CE ve AE koşullu ağaçlarının üretilmesi vardır. Gerekli ağaçlar oluşturulduğunda sadece {C,E} ve {A,E} yaygın öge setleri oluşur.

Sonuç olarak E koşullu bütün yaygın öge setleri bulunmuş olur:

**{E}, {D,E}, {A,D,E}, {C,E}, {A,E}**

Bu işlem header tablosundaki öğelerin hepsi için gerçekleştirilir.

Veri Madenciliği  
Doç. Dr. Suat Özdemir