

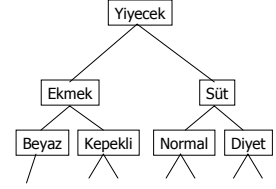
VERİ MADENCİLİĞİ İlişkilendirme Kuralları: Farklı Yöntemler

Yrd. Doç. Dr. Şule Gündüz Öğüdücü
<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

1

Çok Katmanlı İlişkilendirme Kuralları

- Nitelikler bir hiyerarşi oluşturabilir.
 - kavram hiyerarşisi
- Daha alt katmanlarda yer alan ürünlerin destek değeri daha düşük
- Alt seviyelerde oluşan kurallar özel durumları tanımlıyor
 - Beyaz ekmek => normal süt
 - Kepekli ekmek => normal süt
 - Kepekli ekmek => diyet süt



<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

2

Çok Katmanlı İlişkilendirme Kuralları

- Destek ve güven değerlerinin katmanlar arasında değişimi
 - X_1 ve X_2 niteliklerinin üst düğümünde yer alan nitelik X
 $\sigma(X) \geq \sigma(X_1) + \sigma(X_2)$
 - $\sigma(X_1 \cup Y_1) \geq \text{minsup}$, X_1 niteliğinin üst düğümünde yer alan nitelik X , Y_1 niteliğinin üst düğümünde yer alan nitelik Y
 $\sigma(X \cup Y_1) \geq \text{minsup}$, $\sigma(X_1 \cup Y) \geq \text{minsup}$
 $\sigma(X \cup Y) \geq \text{minsup}$
 - $\text{conf}(X_1 \Rightarrow Y_1) \geq \text{minconf}$ ise $\text{conf}(X_1 \Rightarrow Y) \geq \text{minconf}$

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

3

Çok Katmanlı İlişkilendirme Kuralları

- 1. Yaklaşım:
Her hareket t , üst katmanlardaki nitelikler cinsinden yazılarak genişletilmiş harekete t' dönüştürülür
 $t = \{\text{beyaz ekmek, normal süt}\}$
 $t' = \{\text{beyaz ekmek, normal süt, ekmek, süt, yiyecek}\}$
 t' hareketleri için ilişkilendirme kuralları bulunur
- Problemler:
 - Üst düğümlerde yer alan niteliklerin destek değerleri daha büyük
 - Farklı seviyelerde farklı destek değerleri
 - Veri kümesinin boyutu artıyor

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

4

Çok Katmanlı İlişkilendirme Kuralları

- Alt katmanlarda destek değerini küçültme
 - Her katmanda bağımsız hesaplama (Level-by-level independent)
 - Üst düğümleri yaygın nitelik olan niteliklerin destek değerini hesaplama
 - Tek niteliğe göre budama (Level-cross filtering by single item)
 - nitelikler kümesine göre budama (Level-cross filtering by k-itemset)

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

5

Çok Katmanlı İlişkilendirme Kuralları

- 2. Yaklaşım
İlk olarak en üst katman için yaygın nitelikler bulunur
Her aşamada bir alt katmandaki nitelikler arasındaki ilişkiler bulunur
- Problemler:
 - Kullanışlı değil: her katman için veri kümesi araştırılıyor
 - Farklı katmanlar arasındaki ilişkiler bulunamıyor

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

6

Gereksiz Kuralları Eleme

- Ürünler arasındaki ilişki nedeniyle bazı kurallar gereksiz olabilir
- Örnek
 - satın_alma (X, "Ekmek") => satın_alma(X, "Süt") [%8, %70]
 - satın_alma (X, "Beyaz Ekmek") => satın_alma(X, "Süt") [%2, %70]
- İlk kural ikinci kuralın üst kuralı
- Eğer bir kuralın destek değeri üst kuralının destek değerinden tahmin edilebiliyorsa gereksizdir

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

7

Nicel İlişkilendirme Kuralları

Kayıt No	Yaş	Medeni Durum	Araba Sayısı
1	23	Bekar	1
2	25	Evli	1
3	29	Bekar	0
4	34	Evli	2
5	38	Evli	2
...	...	...	...

- Örnek ilişkilendirme kuralları:
 - yaş:30-39, medeni durum: evli => araba sayısı:2 (destek: %40, güven: %100)
 - araba sayısı: 0-1 => medeni durum: bekar (destek: %40, güven: %66,70)

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

8

Nicel İlişkilendirme Kuralları

- İkili değişkenlerden oluşan ilişkilendirme kurallarına dönüştürülüyor:
 - Kategorik olmayan nitelik değerleri için bölmeleme
 - yaş: 20-29, 30-39
 - ikili değişken oluşturma
 - kategorik niteliklerin her değeri için bir ikili değişken
 - kategorik olmayan niteliklerin her bölümü için bir ikili değişken

Kayıt No	Yaş	Medeni Durum	Araba Sayısı
1	23	Bekar	1
5	38	Evli	2

Kayıt No	Yaş 20-29	Yaş 30-39	M.Durum Evli	M.Durum Bekar	Araba 0	Araba 1	Araba 2
1	1	0	0	1	0	1	0
5	0	1	1	0	0	0	1

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

9

Nicel İlişkilendirme Kuralları

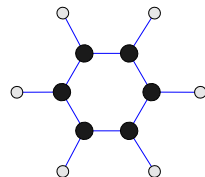
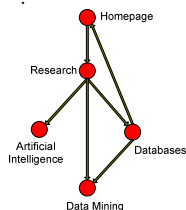
- Dönüştürmeden kaynaklanan problemler
 - bölme sayısı az: Bilgi kaybolması
 - Destek değeri küçük: fazla sayıda kural

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

10

Yaygın Altçizge Madenciliği

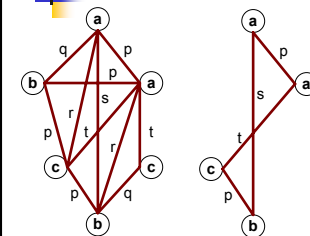
- Frequent Subgraph Mining
 - İlişkilendirme kuralları madenciliği yöntemleri uygulanarak yaygın altçizgeler bulunur
- Uygulama alanları
 - Web madenciliği
 - Biyoenformatik
 - ...



<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

11

Çizge Tanımları



(a) Etiketli Çizge

(b) Altçizge

- $V=\{x_i\}$ düğümler kümesi

- $E=\{w_{ij}\}$ x_i ve x_j düğümleri arasındaki ağırlık

- Altçizge: $G_A(V_A, E_A)$

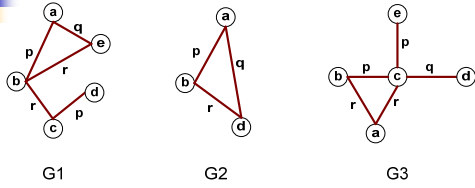
$$\bigcup_{i=1}^k V_i = V$$

$$E_i = \{ \{u, v\} \in E \wedge u, v \in V_i \}$$

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

12

Çizge-Hareket Dönüşümü



	(a,b,p)	(a,b,q)	(a,b,r)	(b,c,p)	(b,c,q)	(b,c,r)	...	(d,e,r)
G1	1	0	0	0	0	1	...	0
G2	1	0	0	0	0	0	...	0
G3	0	0	1	0	0	0	...	0
G3	...	...	...	...	...	...	...	...

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

13

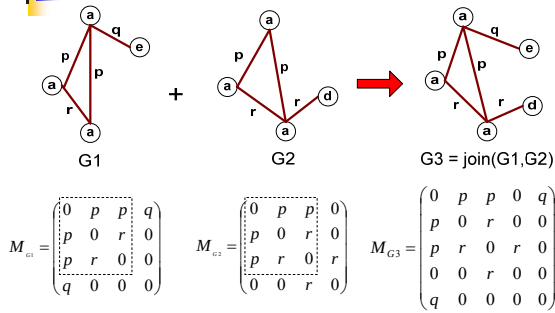
Apriori Temelli Yaklaşım

- Bir altçizgenin güven değeri:
 - Altçizgenin bulunduğu çizgelerin toplam çizgelere oranı
- Apriori temelli yaklaşım
 - yaygın (k+1)-altçizgeleri bulmak için k-altçizgelerin kullanılması
 - düğüm ekleme: k düğüm sayısı
 - ayırt ekleme: k ayırt sayısı

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

14

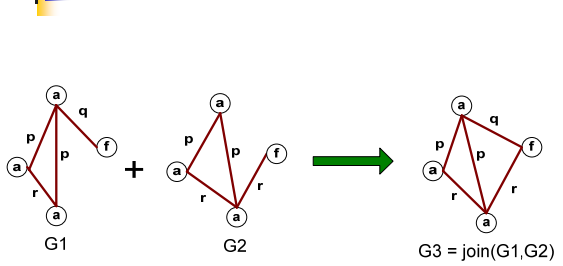
Düğüm Ekleme



<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

15

Ayırt Ekleme



<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

16

Apriori Temelli Algoritma

- Yaygın 1-altçizgelerin belirlenmesi
- Yeni yaygın altçizge oluşmayana kadar tekrarla
 - Aday oluşturma: Yaygın (k-1) altçizgeleri kullanarak k-altçizgeler oluşturma
 - Aday azaltma: (k-1)-altçizgesi yaygın olmayan adayların silinmesi
 - Destek hesaplaması: Kalan adaylar için destek değerinin hesaplanması
 - Aday eleme: min_sup değerini sağlamayan adayların silinmesi

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

17

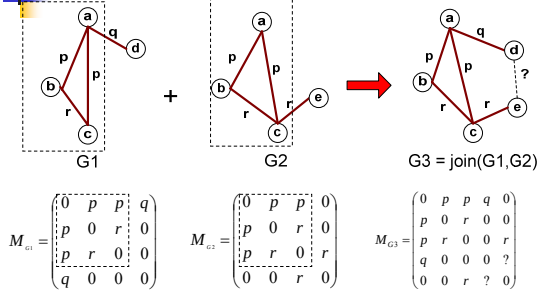
Aday Oluşturma

- Apriori yöntemi:
 - İki (k-1)-yaygın niteliklerin birleştirilmesi sonucu sadece bir k-yaygın nitelikler oluşur.
- Altçizge madenciliği
 - İki (k-1)-yaygın altçizgenin birleştirilmesi sonucu birden fazla k-yaygın altçizge oluşur
 - iki (k-1)-yaygın altçizgenin birleştirilmesi için (k-2)-altçizgelerinin aynı olması gerekir (çekirdek)

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

18

Düğüm Ekleme: Farklı Aday Oluşması

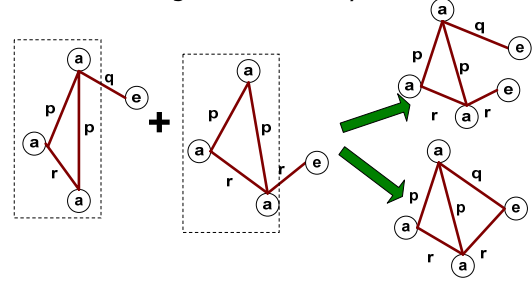


<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

19

Ayrıt Ekleme: Farklı Aday Oluşması

- 1. durum: düğüm etiketleri aynı

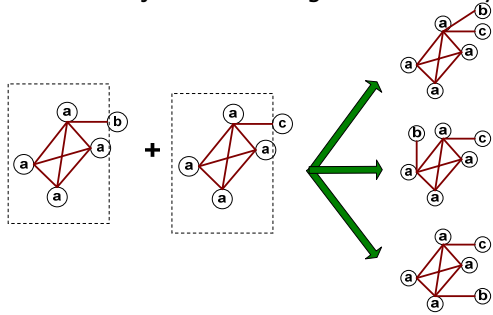


<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

20

Ayrıt Ekleme: Farklı Aday Oluşması

- 2. durum: çekirdekteki düğüm etiketleri aynı



<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

21

Çizge Madenciliği

- Jun Huan, Wei Wang, Jan Prins, and Jiong Yang, *SPIN: Mining maximal frequent subgraphs from graph databases*. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2004.
- X. Yan, X. J. Zhou, and J. Han, *Mining Closed Relational Graphs with Connectivity Constraints*, Proc. 2005 Int. Conf. on Knowledge Discovery and Data Mining (KDD'05), Chicago, IL, Aug. 2005.

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

22

Sıralı Diziler

- İlişkilendirme kuralları hareketlerde yer alan nitelikler arasındaki ilişkiyi bulmayı hedefler. Zaman bilgisini kullanmaz.



- Sıralı dizi madenciliği zaman bilgisini kullanarak hareketlerin sırasına göre nitelikler arasındaki ilişkiyi bulur.

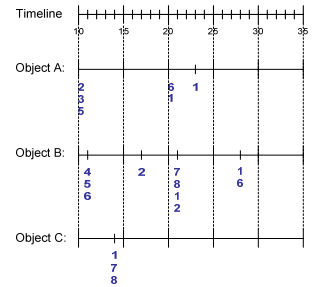


<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

23

Sıralı Dizi Verisi

Object	Timestamp	Events
A	10	2, 3, 5
A	20	6, 1
A	23	1
B	11	4, 5, 6
B	17	2
B	21	7, 8, 1, 2
B	28	1, 6
C	14	1, 8, 7



- Örnek sıralı dizi verisi:
 - müşteri hareketleri
 - tıbbi veriler
 - web verileri
 - telefon görüşme kayıtları
 - DNA dizileri

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

24

Sıralı Dizi Tanımı

Sıralı dizi veritabanı

SID	sıralı dizi
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

Elemanlar içinde niteliklerin sırası yok (liste)

Elemanlar sıralı

9-sıralı dizi

- Elemanlar nitelikler kümesi, nitelikler sıralı değildir, genelde alfabetik sırada sıralanırlar (abc)
- Elemanların sıralı listesi (<a(abc)(ac)d(cf)>)
- Elemanlar zaman veya yer ile nitelendirilir
- Sıralı dizinin uzunluğu |s|: Sıralı diziyi oluşturan nitelikler sayısı
- k-sıralı dizi: k niteliği olan sıralı dizi

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

25

Sıralı Dizi Örnekleri

Web verileri:

< (Homepage) (Electronics) (Digital Cameras) (Canon Digital Camera) (Shopping Cart) (Order Confirmation) (Return to Shopping) >

Kütüphaneden alınan kitaplar:

<(Fellowship of the Ring) (The Two Towers) (Return of the King)>

DNA dizileri

ACAAGATGCCATTGTCCCCGGCCTCTGCTGCTGCTCTCCGGGGCCACGGCC

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

26

Alt Dizi Tanımı

- $\beta = \langle b_1 b_2 \dots b_m \rangle$ sıralı dizisi içinde $\alpha = \langle a_1 a_2 \dots a_n \rangle$ sıralı dizinin bulunabilmesi için ($m \geq n$):
 - $i_1 < i_2 < \dots < i_n$ koşulunda $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \dots, a_n \subseteq b_{i_n}$
 - <a(abc)(ac)d(cf)> sıralı dizinin bir alt dizisi: <a(bc)dc>
 - <ad>, <(ad)c(bc)(ae)> sıralı dizinin alt dizisi değil
- α dizisi için β "super sequence" olarak tanımlanır.
- s dizisi hiç bir dizinin alt dizisi değilse "maximal" olarak tanımlanır.
- Bir alt dizinin destek değeri: Altdizinin bulunduğu sıralı dizilerin toplam sıralı dizi sayısına oranı
- Sıralı dizi örüntüsü: Destek değeri en küçük destek değerine (minsup) eşit ya da büyük olan alt diziler

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

27

Sıralı Dizi Madenciliği

- Sıralı dizi veritabanında en az *minsup* destek değerine sahip alt dizilerin bulunması (sıralı dizi örüntülerinin bulunması)

sıralı dizi veritabanı

SID	sıralı dizi
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

$\alpha 1 = \langle (ab)c \rangle$ minsup($\alpha 1$)=2

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

28

Zorluklar

- Veri kümesi içinde çok fazla sayıda sıralı dizi örüntüsü bulunabilir
 - Sıralı dizi madenciliği algoritmasının bütün sıralı dizi örüntülerini bulması gerekir
 - Veri kümesini az sayıda tarayarak sonuç üretmesi gerekir
 - ölçeklenebilir
 - Değişik veri tipleri üzerinde çalışabilmesi gerekir

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

29

Sıralı Dizi Örüntülerinin Temel Özelliği

Apriori:

- Sıralı dizi s, minsup değerini sağlamıyorsa, s'in hiçbir üst kümesi de minsup değerini sağlamaz.
- Örnek: <hb> minsup değerini sağlamıyor $\rightarrow$ <hab> ve <(ah)b> sıralı dizileri de minsup değerini sağlamaz.

SID	Sıralı dizi
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

minsup = 2

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

30

Apriori All

- Rakesh Agrawal and Ramakrishnan Srikant, *Mining Sequential Patterns*, In Proc. of 11h Int. Conf. on Data Engineering, Taipei, Taiwan, 1995.
- Apriori algoritmasının yaklaşımını kullanır
- Maximal olmayan diziler de dahil olmak üzere bütün sıralı dizileri çıkarır
- Aday sıralı dizileri bulmak için Apriori algoritmasının Apriori-generate fonksiyonunu kullanır. Aday sıralı dizilerin minsup değerlerini bulmak için sıralı dizi verisini tarar.

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

31

Apriori All Algoritması

- 5 aşamadan oluşur
 - Sıralama aşaması (sort phase)
Hareketlerden oluşan veri kümesinden sıralı dizileri oluşturma
 - Yaygın nitelikler kümesi aşaması (large itemset phase)
Apriori ile yaygın nitelikler kümeleri bulunur
 - Dönüştürme aşaması (Transformation phase)
Her yaygın nitelikler kümesi bir tamsayı ile eşleştirilir
 - Sıralı dizi aşaması (sequence phase)
Apriori kullanılarak bütün sıralı dizi örüntüleri çıkarılır
 - Maximal aşaması (maximal phase)
Maximal olmayan sıralı diziler elenir

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

32

Apriori All Algoritması (1)

- C_k : k uzunluğunda yaygın sıralı dizi adayları
 - L_k : k uzunluğunda yaygın sıralı diziler
- ```

 $L_1 = \{1\text{-frequent sequences}\}; // 2. \text{ aşama sonucu}$
for ($k = 1; L_k \neq \emptyset; k++$) do begin
 C_{k+1} = candidates generated from L_k
 for each sequence c in database do
 increment the count of all candidates in C_{k+1} that are contained in c
 L_{k+1} = candidates in C_{k+1} with min_support
end
return $\cup_k L_k$

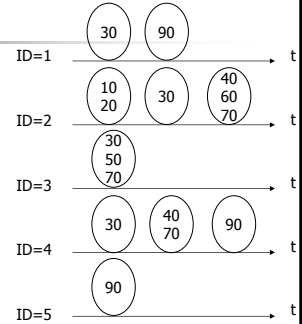
```
- Aday oluşturma:  $L_{k-1}$  kendisiyle birleştirme
- Örnek:  
 $\{1,2,3\} \times \{1,2,4\} =$   
 $\{1,2,3,4\}$  ve  
 $\{1,2,4,3\}$
- insert into  $C_k$   
 select  $p.item_1, p.item_2, \dots, p.item_{k-1}, q.item_{k-1}$   
 from  $L_{k-1} p, L_{k-1} q$   
 where  $p.item_1 = q.item_1, \dots,$   
 $p.item_{k-2} = q.item_{k-2}$

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

33

## Örnek: Sıralı Dizi Veritabanı

| ID | Zaman | Ürün     |
|----|-------|----------|
| 1  | 1     | 30       |
| 1  | 2     | 90       |
| 2  | 1     | 10,20    |
| 2  | 2     | 30       |
| 2  | 3     | 40,60,70 |
| 3  | 1     | 30,50,70 |
| 4  | 1     | 30       |
| 4  | 2     | 40,70    |
| 4  | 3     | 90       |
| 5  | 1     | 90       |



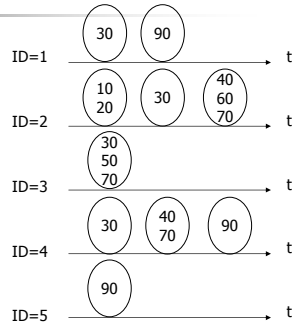
minsup=%40 => en az iki farklı ID  
 Sonuç:  $\langle (30)(90) \rangle$  (ID=1,4)  $\langle (30)(40\ 70) \rangle$  (ID2,4)  
 $\langle 30 \rangle \langle 40 \rangle \langle 70 \rangle \langle 90 \rangle$ ,  $\langle (30)(40) \rangle$ ,  $\langle (30)(70) \rangle$ ,  $\langle (40\ 70) \rangle$  sıralı dizi örüntüsü değil

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

34

## Sıralama Aşaması

| ID | Zaman | Ürün     |
|----|-------|----------|
| 1  | 1     | 30       |
| 1  | 2     | 90       |
| 2  | 1     | 10,20    |
| 2  | 2     | 30       |
| 2  | 3     | 40,60,70 |
| 3  | 1     | 30,50,70 |
| 4  | 1     | 30       |
| 4  | 2     | 40,70    |
| 4  | 3     | 90       |
| 5  | 1     | 90       |



- ID
- Zaman

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

35

## Yaygın Nitelikler Kümesi Aşaması

- Apriori algoritması kullanılarak yaygın nitelikler belirlenir.
  - Sıralı dizi örüntülerinde geçen nitelikler yaygın nitelikler olması gerekiyor
- Destek değerinin farklı hesaplanması gerekiyor
  - Bir niteliğin kaç farklı "ID" için bulunduğu

| ID | Zaman | Ürün     |
|----|-------|----------|
| 1  | 1     | 30       |
| 1  | 2     | 90       |
| 2  | 1     | 10,20    |
| 2  | 2     | 30       |
| 2  | 3     | 40,60,70 |
| 3  | 1     | 30,50,70 |
| 4  | 1     | 30       |
| 4  | 2     | 40,70    |
| 4  | 3     | 90       |
| 5  | 1     | 90       |

minsup=%40 => en az iki farklı ID  
 Yaygın nitelikler kümeleri:  
 $\{30\}\{40\}\{70\}\{40\ 70\}\{90\}$

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

36

## Dönüştürme Aşaması (1)

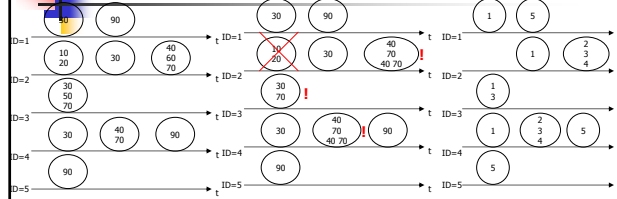
- Her yaygın nitelikler kümesi bir tamsayı ile eşleştirilir
  - İki farklı nitelikler kümesinin veritabanı içinde eşit zamanda taranması için
- Hareketler tamsayılar ile temsil edilir

| Nitelikler | Tamsayı |
|------------|---------|
| {30}       | 1       |
| {40}       | 2       |
| {70}       | 3       |
| {40 70}    | 4       |
| {90}       | 5       |

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

37

## Dönüştürme Aşaması (2)



| ID | Sıralı Diziler        |
|----|-----------------------|
| 1  | <(30) (90)>           |
| 2  | <(10 20)30(40 60 70)> |
| 3  | <(30 50 70)>          |
| 4  | <30 (40 70) 90>       |
| 5  | <(90)>                |

| Nitelikler | Tamsayı |
|------------|---------|
| {30}       | 1       |
| {40}       | 2       |
| {70}       | 3       |
| {40 70}    | 4       |
| {90}       | 5       |

| ID | Tamsayı Eşleneği  |
|----|-------------------|
| 1  | <{1} {5}>         |
| 2  | <1{2 3 4}>        |
| 3  | <{1 3}>           |
| 4  | <{1} {2 3 4} {5}> |
| 5  | <{5}>             |

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

38

## Sıralı Dizi Aşaması

- Nitelikler kümesi kullanılarak istenen destek değerini sağlayan sıralı diziler bulunuyor
- Apriori algoritması ile aynı yaklaşım

minsup=%40

| ID | Tamsayı Eşleneği  |
|----|-------------------|
| 1  | <{1} {5}>         |
| 2  | <1{2 3 4}>        |
| 3  | <{1 3}>           |
| 4  | <{1} {2 3 4} {5}> |
| 5  | <{5}>             |

| Nitelikler | Destek |
|------------|--------|
| {1}        | 4      |
| {2}        | 2      |
| {3}        | 3      |
| {4}        | 2      |
| {5}        | 3      |
| {1} {2}    | 2      |
| {1} {3}    | 3      |
| {1} {4}    | 2      |
| {1} {5}    | 2      |

|              |
|--------------|
| <30>         |
| <40>         |
| <70>         |
| <40 70>      |
| <90>         |
| <30><40>     |
| <30><70>     |
| <30 (40 70)> |
| <30><90>     |

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

39

## Maximal Aşaması

- Maximal olmayan sıralı dizileri sil
  - Başka bir sıralı dizinin alt dizisi olan dizileri sil

for  $k=n; k=1; k--$   
for each  $k$ -sequence  $s_k$  do  
delete all subsequences of  $s_k$

|              |
|--------------|
| <30>         |
| <40>         |
| <70>         |
| <40 70>      |
| <90>         |
| <30><40>     |
| <30><70>     |
| <30 (40 70)> |
| <30><90>     |

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

40

## GSP (A Generalized Sequential Pattern Mining Algorithm)

- Ramakrishnan Srikant, Rakesh Agrawal, *Mining Sequential Patterns: Generalizations And Performance Improvements*, Proc. 5th Int. Conf. Extending Database Technology, EDBT'96
- 1. Aşama:
  - Veritabanını bir kez tarayarak 1 nitelikten oluşan sıralı dizi örüntülerini bulunur
- 2. Aşama: Yeni bir sıralı dizi örüntüsü bulunmayınca kadar tekrarla
  - Aday oluşturma:  $(k-1)$ , adımda bulunan sıralı dizi örüntüleri birleştirilerek  $k$  uzunluklu sıralı dizi örüntü adayları bulunur
  - Aday azaltma:  $(k-1)$  alt dizileri minsup değerini sağlamayan adaylar elenir
  - Destek değeri hesaplama: Kalan sıralı dizi örüntü adayları için veritabanı taranarak destek değeri hesaplanır
  - Aday eleme: minsup değeri sağlamayan sıralı dizi örüntü adayları elenir.

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

41

## GSP: Aday Oluşturma

- İlk durum ( $k=2$ ): Bir uzunluklu sıralı dizilerden iki uzunluklu sıralı diziler oluşturma
  - $s_{11} = \langle i_1 \rangle$  ve  $s_{12} = \langle i_2 \rangle$  sıralı dizilerin birleştirilmesi sonucunda  $s_{21} = \langle i_1 i_2 \rangle$ ,  $s_{22} = \langle i_1 i_2 \rangle$ ,  $s_{23} = \langle i_2 i_1 \rangle$ ,  $s_{24} = \langle i_2 i_1 \rangle$  ve  $s_{25} = \langle i_1 i_2 \rangle$  sıralı dizileri oluşur
- Genel durum ( $k>2$ ):
  - $k$  uzunluklu  $s_{ki}$  ve  $s_{kj}$  sıralı dizileri şu şartlar altında birleştir:
    - $s_{ki}$  sıralı dizinin ilk niteliği ve  $s_{kj}$  sıralı dizinin son niteliği silindiğinde oluşan alt dizinin aynı olması
  - $(k+1)$  uzunluklu sıralı dizi örüntü adayı,  $s_{ki}$  sıralı diziyeye  $s_{kj}$  sıralı dizinin son niteliğinin eklenmesi ile bulunur
    - $s_{ki}$  sıralı dizinin son iki niteliği aynı elemana ait:  $s_{kj}$  sıralı dizinin son niteliği  $s_{ki}$  sıralı dizisinin son elemanının parçası olur
    - diğer durumda  $s_{ki}$  sıralı dizisinin son niteliği  $s_{kj}$  sıralı dizisinin ayrı bir elemanı olur.

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

42

## Örnek

- $s_1 = \langle (a) (b c) (d) \rangle$  ve  $s_2 = \langle (b c) (d e) \rangle$  sıralı dizileri birleştirilirse
  - aday sıralı dizi  $\langle (a) (b c) (de) \rangle$ 
    - $s_2$  sıralı dizideki son iki nitelik (d ve e) aynı elemana ait
- $s_1 = \langle (a) (b c) (d) \rangle$  ve  $s_2 = \langle (b c) (d)(e) \rangle$  sıralı dizileri birleştirilirse
  - aday sıralı dizi  $\langle (a) (b c) (d)(e) \rangle$ 
    - $s_2$  sıralı dizideki son iki nitelik (d ve e) aynı elemana ait değil
- $s_1 = \langle (a)(b f)(d) \rangle$  ve  $s_2 = \langle (a) (b) (d e) \rangle$  sıralı dizileri  $\langle (a) (b f) (de) \rangle$  sıralı dizi örüntü adayını oluşturmak üzere birleştirilemez
  - $s_2 = \langle (b f) (d x) \rangle$  veya  $s_2 = \langle (b f) (d) (x) \rangle$  şeklinde olmadığı için

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

43

## Örnek

- Bir nitelikten oluşan dizilerin bulunması
- Adaylar
  - $\langle a \rangle$ ,  $\langle b \rangle$ ,  $\langle c \rangle$ ,  $\langle d \rangle$ ,  $\langle e \rangle$ ,  $\langle f \rangle$ ,  $\langle g \rangle$ ,  $\langle h \rangle$
- Veritabanı taranarak destek sayıları hesaplanır

$min\_sup = 2$

| SID | Sıralı Dizi                     |
|-----|---------------------------------|
| 10  | $\langle (bd)cb(ac) \rangle$    |
| 20  | $\langle (bf)(ce)b(fg) \rangle$ |
| 30  | $\langle (ah)(bf)abf \rangle$   |
| 40  | $\langle (be)(ce)d \rangle$     |
| 50  | $\langle a(bd)bcb(ade) \rangle$ |

| Aday                | Des |
|---------------------|-----|
| $\langle a \rangle$ | 3   |
| $\langle b \rangle$ | 5   |
| $\langle c \rangle$ | 4   |
| $\langle d \rangle$ | 3   |
| $\langle e \rangle$ | 3   |
| $\langle f \rangle$ | 2   |
| $\langle g \rangle$ | 1   |
| $\langle h \rangle$ | 1   |

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

44

## Örnek

- iki uzunluklu sıralı dizilerin bulunması
  - 51 aday

|                     |                     |                      |                      |                      |                      |                      |
|---------------------|---------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
|                     | $\langle a \rangle$ | $\langle b \rangle$  | $\langle c \rangle$  | $\langle d \rangle$  | $\langle e \rangle$  | $\langle f \rangle$  |
| $\langle a \rangle$ |                     | $\langle aa \rangle$ | $\langle ab \rangle$ | $\langle ac \rangle$ | $\langle ad \rangle$ | $\langle ae \rangle$ |
| $\langle b \rangle$ |                     | $\langle ba \rangle$ | $\langle bb \rangle$ | $\langle bc \rangle$ | $\langle bd \rangle$ | $\langle be \rangle$ |
| $\langle c \rangle$ |                     | $\langle ca \rangle$ | $\langle cb \rangle$ | $\langle cc \rangle$ | $\langle cd \rangle$ | $\langle ce \rangle$ |
| $\langle d \rangle$ |                     | $\langle da \rangle$ | $\langle db \rangle$ | $\langle dc \rangle$ | $\langle dd \rangle$ | $\langle de \rangle$ |
| $\langle e \rangle$ |                     | $\langle ea \rangle$ | $\langle eb \rangle$ | $\langle ec \rangle$ | $\langle ed \rangle$ | $\langle ee \rangle$ |
| $\langle f \rangle$ |                     | $\langle fa \rangle$ | $\langle fb \rangle$ | $\langle fc \rangle$ | $\langle fd \rangle$ | $\langle fe \rangle$ |

Apriori özelliği kullanılmadan aday sayısı  
 $8 * 8 + 8 * 7 / 2 = 92$

Apriori aday sayısını  
 %44.57 azaltıyor

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

45

## Örnek

- 3 uzunluklu adayların bulunması
  - 2 uzunluklu sıralı dizilerden 3 uzunluklu adaylar oluşturma
    - Apriori özelliği kullanılıyor
    - $\langle ab \rangle$ ,  $\langle aa \rangle$  ve  $\langle ba \rangle$  : 2 uzunluklu sıralı diziler  
 $\rightarrow \langle aba \rangle$  : 3 uzunluklu sıralı dizi
    - $\langle (bd) \rangle$ ,  $\langle bb \rangle$  ve  $\langle db \rangle$  : 2 uzunluklu sıralı diziler  
 $\rightarrow \langle (bd)b \rangle$  : 3 uzunluklu sıralı dizi
- 46 aday oluşuyor
- 3 uzunluklu sıralı dizilerin bulunması
  - Veritabanı taranarak adayların destek sayıları hesaplanır
  - 46 adaydan 19 adet 3 uzunluklu sıralı dizi örüntüsü bulunur

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

46

## GSP Örnek

1. adım: 8 aday, 6 adet 1-sıralı dizi örüntüsü.
2. adım: 51 aday, 19 adet 2-sıralı dizi örüntüsü. 10 aday DB'de değil
3. adım: 46 aday, 19 adet 3-sıralı dizi örüntüsü. 20 aday DB'de değil
4. adım: 8 aday, 6 adet 4-sıralı dizi örüntüsü
5. adım: 1 aday, 1 adet 5-sıralı dizi örüntüsü

$minsup = 2$

| ID | Sıralı dizi                     |
|----|---------------------------------|
| 10 | $\langle (bd)cb(ac) \rangle$    |
| 20 | $\langle (bf)(ce)b(fg) \rangle$ |
| 30 | $\langle (ah)(bf)abf \rangle$   |
| 40 | $\langle (be)(ce)d \rangle$     |
| 50 | $\langle a(bd)bcb(ade) \rangle$ |

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

47

## GSP Örnek

3-sıralı dizi örüntüleri

|                                     |
|-------------------------------------|
| $\langle \{1\} \{2\} \{3\} \rangle$ |
| $\langle \{1\} \{2\} \{5\} \rangle$ |
| $\langle \{1\} \{3\} \{3\} \rangle$ |
| $\langle \{2\} \{3\} \{4\} \rangle$ |
| $\langle \{2\} \{5\} \{3\} \rangle$ |
| $\langle \{3\} \{4\} \{5\} \rangle$ |
| $\langle \{5\} \{3\} \{4\} \rangle$ |

Aday Oluşturma

|                                           |
|-------------------------------------------|
| $\langle \{1\} \{2\} \{3\} \{4\} \rangle$ |
| $\langle \{1\} \{2\} \{5\} \{3\} \rangle$ |
| $\langle \{1\} \{5\} \{3\} \{4\} \rangle$ |
| $\langle \{2\} \{3\} \{4\} \{5\} \rangle$ |
| $\langle \{2\} \{5\} \{3\} \{4\} \rangle$ |

Aday Eleme

|                                           |
|-------------------------------------------|
| $\langle \{1\} \{2\} \{5\} \{3\} \rangle$ |
|-------------------------------------------|

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

48



## GSP Algoritmasında Problemler

- Çok sayıda sıralı dizi örüntü adayı oluşturulabilir
  - 1000 adet 1-sıralı dizi örüntülerinden  $1000 \times 1000 + \frac{1000 \times 999}{2} = 1,499,500$  adet sıralı dizi örüntü adayı oluşur.
- Veritabanı birçok kez taranır
- Uzun sıralı dizileri belirlemek çok zor
  - sıralı diziler art arda eklenerek daha uzun sıralı diziler oluşuyor
  - 100-sıralı dizi oluşturmak için  $10^{30}$  sıralı dizi gerekiyor.

$$\sum_{i=1}^{100} \binom{100}{i} = 2^{100} - 1 \approx 10^{30}$$

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

49

## Sıralı Dizi Örüntü Adayı Oluşturma

- Parçala ve çöz yaklaşımı
  - veritabanı yinlemeli küçültülür
  - oluşan her veritabanında sıralı dizi örüntüleri bulunur
- Algoritmalar:
  - J. Han J. Pei, B. Mortazavi-Asi, Q. Chen, U. Dayal and M.C. Hsu. *FreeSpan: Frequent pattern-projected sequential pattern mining*. KDD'00
  - J. Pei, J. Han, B. Mortazavi-Asi, H. Pinto, Q. Chen, U. Dayal, and M.C. Hsu. *Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth*. In Proceedings of the 17th International Conference on Data Engineering, 2001.

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

50

## FreeSpan Algoritması

- Örnek: Sıralı dizi veritabanı ve min\_sup=2

### Sıralı dizi Veritabanı

```
< (bd) c b (ac) >
< (bf) (ce) b (fg) >
< (ah) (bf) a b f >
< (be) (ce) d >
< a (bd) b c b (ade) >
```

- 1. Adım: 1-sıralı dizi örüntülerinin bulunması ve destek değeri küçülen sırada sıralanmaları (f-list)

**f\_list:** b:5, c:4, a:3, d:3, e:3, f:2

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

51

## FreeSpan Algoritması

- 2. Adım: Arama uzayının bölünmesi. Sıralı dizi veritabanı altkümelerine bölünür
  - f niteliği bulunan alt diziler
  - e niteliği bulunan f niteliği bulunmayan
  - d niteliği bulunan e ve f nitelikleri bulunmayan
  - a niteliği bulunan d, e ve f nitelikleri bulunmayan
  - c niteliği bulunan a, d, e ve f nitelikleri bulunmayan
  - b niteliği bulunan
- Her alt dizi ( $\alpha$ ) için bu alt dizilerin bulunduğu sıralı dizilerden oluşan veritabanları,  $\alpha$ -yansımali veritabanları ( $\alpha$ -projected database), oluşturulur.

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

52

## FreeSpan Algoritması

- c niteliği bulunan a, d, e ve f nitelikleri bulunmayan sıralı diziler
  - <c>-yansımali veritabanı: <b c b c>, <b c b>, <b c>, <b b c b>
  - c niteliği bulunan a, d, e ve f nitelikleri bulunmayan bütün alt dizilerin bulunması: <bc>: 4, <cb>: 3
  - Veritabanı yansıtılarak sıralı diziler bulunmaya devam edilir.

### Sıralı dizi Veritabanı

```
< (bd) c b (ac) >
< (bf) (ce) b (fg) >
< (ah) (bf) a b f >
< (be) (ce) d >
< a (bd) b c b (ade) >
```

<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

53