

VERİ MADENCİLİĞİ İlişkilendirme Kuralları

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

1

İlişkilendirme Kuralları Madenciliği

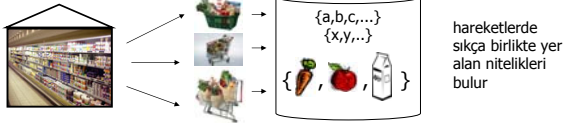
- İlişkilendirme kuralı madenciliği
 - Veri kümesi içindeki yaygın örüntülerin, nesneleri oluşturan nitelikler arasındaki ilişkilerin bulunması
- İlişkilendirme kurallarını kullanma: veri içindeki kuralları belirleme
 - Hangi ürünler çoğunlukla birlikte satılıyor?
 - Kişisel bilgisayar satın alan bir kişinin bir sonraki satın alacağı ürün ne olabilir?
 - Yeni bir ilaca duyarlı olan DNA tipleri hangileridir?
 - Web dökümanları otomatik olarak sınıflandırılabilir mi?

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

2

İlişkilendirme Kuralları Bulma

- Bir niteliğin (veya nitelikler kümesinin) varlığını harekette bulunan başka niteliklerin varlıklarına dayanarak öngörme



- Kural şekli: "Gövde → Baş [destek, güven]"

satın alma(x, "ekmek") → satın alma (x, "süt") [%0.6, %65]
öğrenci(x, "BLG"), kayıt(x, "VTYS") → not(x, "A") [%1, %75]

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

3

İlişkilendirme Kuralları Bulma

- Bütün niteliklerden oluşan küme $I = \{I_1, I_2, \dots, I_q\}$
 - $I = \{\text{ekmek, süt, bira, kola, yumurta, bez}\}$
- Hareket $T_i \in I, T_j = \{I_{j1}, I_{j2}, \dots, I_{jn}\}$
 - $T1 = \{\text{ekmek, süt}\}$
- Hareketlerden oluşan veri kümesi $D = \{T_1, T_2, \dots, T_N\}$



Market Alışveriş verisi

Hareket	Öğeler
T1	Ekmek, Süt
T2	Ekmek, Bez, Bira, Yumurta
T3	Süt, Bez, Bira, Kola
T4	Ekmek, Süt, Bez, Bira
T5	Ekmek, Süt, Bez, Kola

Yaygın nitelikler:

Bez, bira
Süt, ekmek, yumurta, kola
Bira, ekmek, süt

Bulunan İlişkilendirme Kuralları

{Bez} → {Bira},
{Süt, Ekmek} → {Yumurta, Kola},
{Bira, Ekmek} → {Süt}

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

4

Yaygın Nitelikler

- nitelikler kümesi (Itemset)
 - Bir veya daha çok nitelikten oluşan küme
 - k-nitelikler kümesi (k-itemset): k nitelikten oluşan küme
 - 3-nitelikler kümesi: {Bez, Bira, Ekmek}
- Destek sayısı σ (Support count)
 - Bir nitelikler kümesinin veri kümesinde görülme sıklığı
 - $\sigma(\{\text{Süt, Ekmek, Bez}\}) = 2$
- Destek s (Support)
 - Bir nitelikler kümesinin içinde bulunduğu hareketlerin toplam hareketlere oranı

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

5

İlişkilendirme Kuralları

- Veri kümesi $D = \{T_1, T_2, \dots, T_N\}$
- en az, en küçük destek ve güven değerine sahip $X \rightarrow Y$ şeklinde kuralların bulunması
 - $X \subset I, Y \subset I, X \cap Y = \emptyset$
- Kuralları değerlendirme ölçütleri
 - destek (support) s: $X \cup Y$ nitelikler kümesinin bulunduğu hareket sayısının toplam hareket sayısına oranı

$$\text{support}(X \rightarrow Y) = \frac{\#X \cup Y}{N}$$
 - güven (confidence) c: $X \cup Y$ nitelikler kümesinin bulunduğu hareket sayısının X nitelikler kümesi bulunan hareket sayısına oranı

$$\text{confidence}(X \rightarrow Y) = \frac{\#(X \cup Y)}{\#X}$$

TID	Öğeler
T1	Ekmek, Süt
T2	Ekmek, Bez, Bira, Yumurta
T3	Süt, Bez, Bira, Kola
T4	Ekmek, Süt, Bez, Bira
T5	Ekmek, Süt, Bez, Kola

Örnek:
{Süt, Bez} → {Bira}

$$\text{support} = \frac{\sigma(\text{süt, bira, bez})}{|T|} = \frac{2}{5} = 0.4$$

$$\text{confidence} = \frac{\sigma(\text{süt, bira, bez})}{\sigma(\text{süt, bez})} = \frac{2}{3} = 0.67$$

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

6

İlişkilendirme Kuralları Oluşturma

- İlişkilendirme kuralları madenciliğinde temel amaç: D hareket kümesinden kurallar oluşturmak
 - kuralların destek değeri, belirlenen en küçük destek ($minsup$) değerinden büyük ya da eşit olmalı
 - kuralların güven değeri, belirlenen en küçük güven ($minconf$) değerinden büyük ya da eşit olmalı
- Brute-force yaklaşım
 - Olası bütün kuralları listele
 - Her kural için destek ve güven değeri hesapla
 - $minsup$ ve $minconf$ eşik değerlerinden küçük destek ve güven değerlerine sahip kuralları sil
 - hesaplama maliyeti yüksek

$$R = 3^d - 2^{d+1} + 1$$

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

7

İlişkilendirme Kuralları Özellikleri

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

Örnek Kurallar:

$\{Süt, Bez\} \rightarrow \{Bira\}$ ($s=0.4, c=0.67$)
 $\{Süt, Bira\} \rightarrow \{Bez\}$ ($s=0.4, c=1.0$)
 $\{Bez, Bira\} \rightarrow \{Süt\}$ ($s=0.4, c=0.67$)
 $\{Bira\} \rightarrow \{Süt, Bez\}$ ($s=0.4, c=0.67$)
 $\{Bez\} \rightarrow \{Süt, Bira\}$ ($s=0.4, c=0.5$)
 $\{Süt\} \rightarrow \{Bez, Bira\}$ ($s=0.4, c=0.5$)

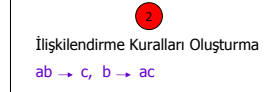
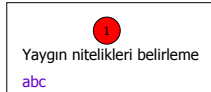
- Aynı nitelikler kümesinin ikili bölünmesine (binary partition) ait kurallar
 - $\{Süt, Bez, Bira\}$
- Aynı nitelikler kümesinden oluşan kuralların destek değerleri aynı, güven değerleri farklı
- Kurallar için destek ve güven şartları ayrı değerlendirilebilir

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

8

İlişkilendirme Kuralları Oluşturma

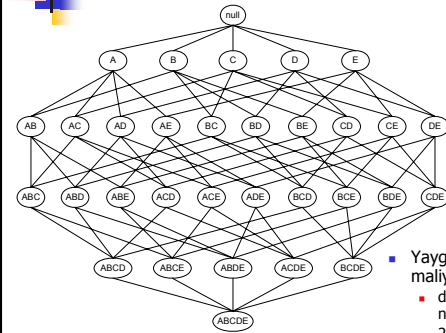
- İki adımda gerçekleşir
 - Yaygın nitelikleri belirleme:**
 - destek değeri $minsup$ değerinden büyük ya da eşit olan nitelikler kümelerini bulma
 - Kural Oluşturma:**
 - Destek değeri $minconf$ değerinden büyük ya da eşit olan ve yaygın niteliklerin ikili bölünmeleri olan kuralları oluşturma
 - Güçlü kurallar



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

9

Yaygın nitelik Adayları Oluşturma



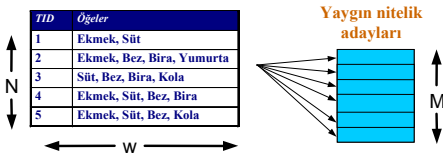
- Yaygın nitelikleri bulmak maliyetli
 - d nitelik için $2^d - 1$ yaygın nitelik oluşabilir
 - $2^d - 1$ yaygın nitelik adayı

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

10

Yaygın Nitelik Oluşturma

- Brute-Force Yaklaşım
 - Her yaygın nitelik adayı için veri kümesini taranarak hareketlerde yaygın nitelik adayı bulunup bulunmadığı kontrol edilir



- yaygın nitelik adayları için destek değeri bulunur
- Destek değeri $minsup$ değerine eşit yada büyük olanlar yaygın nitelikler
- karmaşıklığı: $O(NMw)$, $M=2^d - 1 \Rightarrow$ hesaplaması maliyetli

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

11

Yaygın nitelik Oluşturma Yöntemleri

- Yaygın nitelik aday sayısını (M) azaltma
 - Örnek: Apriori algoritması
- Hareket sayısını (N) azaltma
- Veri kümesi tarama sayısını (MN) azaltma

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

12

Apriori Algoritması

- 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 nitelikler kümesinin destek değeri altkümünün destek değerinden büyük olamaz
- anti-monotone özellik

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

13

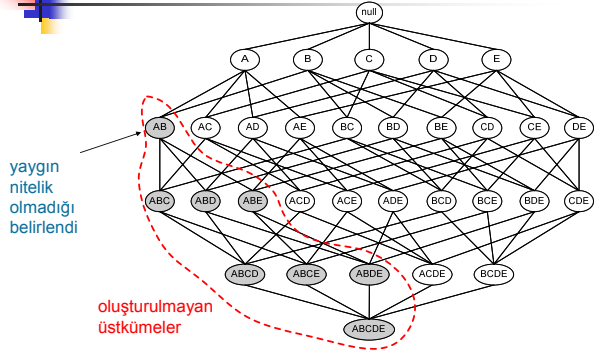
Apriori Algoritması

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

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

14

Apriori Yaygın nitelik Azaltma Yöntemi



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

15

Apriori Algoritması: Örnek

Öge	Sayı
Ekmek	4
Kola	2
Süt	4
Bira	3
Bez	4
Yumurta	1

1-nitelikler kümesi

Öğeler kümesi	Sayı
{Ekmek, Süt}	3
{Ekmek, Bira}	2
{Ekmek, Bez}	3
{Süt, Bira}	2
{Süt, Bez}	3
{Bira, Bez}	3

2-nitelikler kümesi

(kola ya da yumurta olan nitelikler kümesini yaygın nitelik adayı olarak oluşturmaya gerek yok)

minsup = 3

Her nitelikler kümesi için destek hesaplandı,
 $C_1 + C_2 + C_3 = 6 + 15 + 20 = 41$
 Yaygın nitelik sayısı azaltılınca,
 $6 + 6 + 1 = 13$

3-nitelikler kümesi

Öğeler Kümesi	Sayı
{Ekmek, Süt, Bez}	3

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

16

Apriori Algoritması

- Pseudo-code:
 - C_k : Candidate itemset of size k
 - L_k : frequent itemset of size k
 - $L_I = \{\text{frequent items}\};$
 - for ($k = 1; L_k \neq \emptyset; k++$) do begin
 - C_{k+1} = candidates generated from L_k
 - for each transaction t in database do
 - increment the count of all candidates in C_{k+1} that are contained in t
 - L_{k+1} = candidates in C_{k+1} with min_support
 - end
 - return $\cup_k L_k$

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

17

Yaygın Nitelikler Adayı Oluşturma

- L_{k-1} kümesinde nitelikler alfabetik sıralanmış olsun
- 1. adım: L_{k-1} kendisiyle birleştirme
 - insert into C_k
 - select $p.item_y, p.item_z, \dots, p.item_{k-1}, q.item_{k-1}$
 - from $L_{k-1} p, L_{k-1} q$
 - where $p.item_1 = q.item_y, \dots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$
- 2. adım: Eleme
 - forall itemsets c in C_k do
 - forall $(k-1)$ -subsets s of c do
 - if (s is not in L_{k-1}) then delete c from C_k

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

18

Örnek: Yaygın nitelikler Adayı Oluşturma

- Yaygın nitelik adayları oluşturma:
 - L_k kendisiyle birleştirilir (self join)
 - eleme
- Örnek:
 - $L_3 = \{abc, abd, acd, ace, bcd\}$
 - Kendisiyle birleştirme: $L_3 * L_3$
 - abc ve abd nitelikler kümesinden $abcd$
 - acd ve ace nitelikler kümesinden $acde$
 - Eleme
 - ade L_3 kümesinin bir elemanı olmadığından $acde$ yaygın niteliklere dahil edilmez
 - $C_{\mathcal{F}} = \{abcd\}$

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

19

Yaygın niteliklerden İlişkilendirme Kuralları Oluşturma

- Sadece güçlü ilişkilendirme kuralları oluşuyor
- Yaygın nitelikler *minsup* değerini sağlıyor
- Güçlü ilişkilendirme kuralları *minconf* değerini sağlıyor.
- Güven $(A \rightarrow B) = \frac{\text{Destek}(A \cup B)}{\text{Destek}(A)}$
- Yöntem:
 - Her yaygın nitelikler kümesi f 'in altkümelerini oluştur
 - Her altküme s için, $s \rightarrow (f-s)$ ilişkilendirme kuralı oluştur eğer: $\frac{\text{destek}(f)}{\text{destek}(s)} \geq \text{minconf}$ ise

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

20

Apriori Algoritmasını Geliştirme

- Veritabanı tarama sayısını azaltma
 - k-yaygın nitelikler kümesi için veritabanı k kez taranıyor
- Hash yöntemi ile veritabanı tarama sayısını azaltma
- Veritabanındaki hareket sayısını ve hareketlerdeki nitelik sayısını azaltma
 - yaygın olmayan niteliklerin veritabanında yer almasına gerek yok
- Arama uzayını parçalama

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

21

Eniyileme: Hash Ağacı

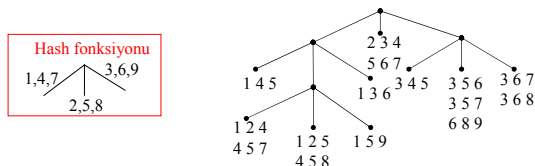
- Her yaygın nitelik adayının destek değerinin belirlenmesi için veri kümesi taranır
 - Yaygın nitelik adayı sayısı çok büyük olabilir
- Karşılaştırma sayısını azaltmak için yaygın nitelik adayları hash ağacında saklanır
 - Her hareket bütün yaygın niteliklerle karşılaştırılmak yerine hash ağacının ilgili bölümündeki yaygın nitelikler adayları ile karşılaştırır

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

22

Hash Ağacı Oluşturma

- 15 adet 3-nitelikler kümesi
 $\{1\ 4\ 5\}, \{1\ 2\ 4\}, \{4\ 5\ 7\}, \{1\ 2\ 5\}, \{4\ 5\ 8\}, \{1\ 5\ 9\}, \{1\ 3\ 6\}, \{2\ 3\ 4\}, \{5\ 6\ 7\}, \{3\ 4\ 5\}, \{3\ 5\ 6\}, \{3\ 5\ 7\}, \{6\ 8\ 9\}, \{3\ 6\ 7\}, \{3\ 6\ 8\}$
- Hash fonksiyonu kullanılarak bir hash ağacında saklanıyor
 - $h(p) = p \bmod 3$
- max yaprak boyutu: bir yaprakta bulunabilecek nitelik kümesi sayısı

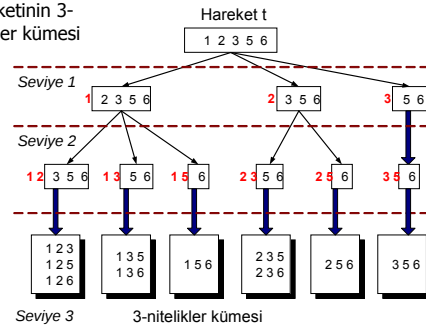


<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

23

Bir Harekette Bulunabilecek Yaygın nitelikleri Belirleme

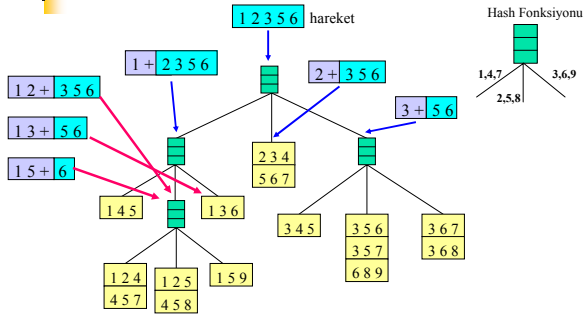
- t hareketinin 3-nitelikler kümesi



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

24

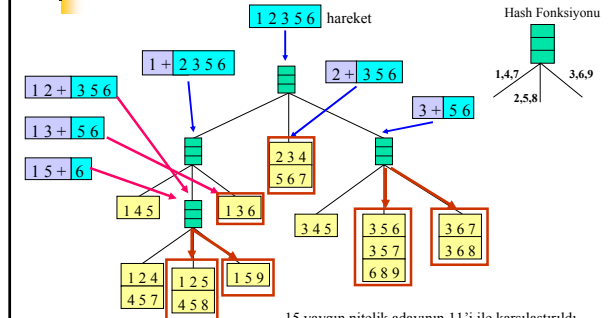
Bir Harekette Bulunabilecek Yaygın nitelikleri Hash Ağacı ile Belirleme



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

25

Bir Harekette Bulunabilecek Yaygın nitelikleri Hash Ağacı ile Belirleme



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

26

Karmaşıklığı Etkileyen Etmenler

- Veri kümesindeki boyut sayısı (nitelik sayısı)
 - her niteliğin destek değerini saklamak için daha fazla saklama alanına ihtiyaç var
- Veri kümesinin büyüklüğü
 - Veri kümesi tarandığı için veri kümesindeki hareket sayısının fazla olması algoritmanın çalışma süresini uzatır
- Hareketlerin ortalama büyüklüğü (nitelik sayısı)
 - Yoğun veri kümelerinde hareketlerdeki nitelik sayısı fazla olur
 - Yaygın nitelik kümelerindeki nitelik sayısı daha fazladır
- En küçük destek değerini belirleme
 - En küçük destek değerini küçültme daha fazla sayıda yaygın nitelik oluşmasına neden olur
 - Yaygın nitelik aday sayısının ve yaygın nitelik kümesindeki niteliklerin sayısının artmasına neden olur

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

27

Yaygın Nitelikleri Belirlemede Sorunlar

- Sorunlar
 - Veri kümesinin birçok kez taranması
 - Yaygın nitelikler aday sayısının fazlalığı
 - Yaygın nitelikleri bulmak için $1 \leq j \leq 100$
 - veri kümesini tarama sayısı: 100
 - Aday sayısı: $(100^1) + (100^2) + \dots + (100^{100}) = 2^{100} - 1 = 1.27 \times 10^{30}$
 - Yaygın nitelikler adayları için destek değerinin hesaplanması
- Çözümler
 - Veri kümesi tarama sayısını azaltma (S. Brin R. Motwani, J. Ullman, and S. Tsur. *Dynamic itemset counting and implication rules for market basket data*. In SIGMOD'97)
 - Yaygın nitelik aday sayısını azaltma (J. Park, M. Chen, and P. Yu. *An effective hash-based algorithm for mining association rules*. In SIGMOD'95)
 - Destek değerinin hesaplanmasını kolaylaştırma (M. Zaki et al. *New algorithms for fast discovery of association rules*. In KDD'97)
 - Yaygın nitelikler aday oluşturmadan yaygın nitelikler bulunabilir. (J. Han, J. Pei, and Y. Yin, *Mining Frequent Patterns without Candidate Generation*. In SIGMOD'00.)

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

28

Aday Oluşturmadan Yaygın Nitelikleri Belirleme

- Kısa yaygın niteliklere yeni nitelikler eklenerek daha uzun yaygın nitelikler elde etme
- Örnek:
 - "abc" bir yaygın nitelik
 - Veri kümesinde içinde "abc" nitelikleri bulunan hareketler (DB|abc)
 - DB|abc içinde d yaygın nitelik olarak bulunursa: "abcd" yaygın nitelik olarak belirlenir

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

29

FP-Tree Algoritması



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

30

FP-Ağacının Özelliği

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

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

31

Örüntüleri ve Veri Kümesini Bölme

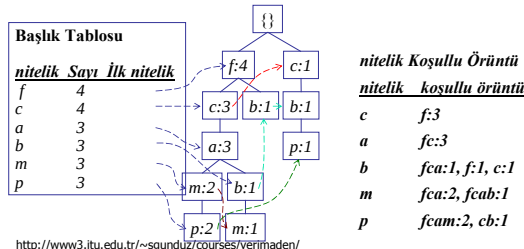
- Yaygın nitelikler f-listesine göre altkümelere bölünür
 - F-list=f-c-a-b-m-p
 - p niteliksi bulunan örüntüler
 - m niteliği bulunan ancak p niteliği bulunmayan örüntüler
 - ...
 - c niteliği bulunan ancak a, b, m, p niteliği bulunmayan örüntüler
 - f niteliksi bulunan örüntüler

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

32

Nitelik Koşullu Örüntü Oluşturma

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

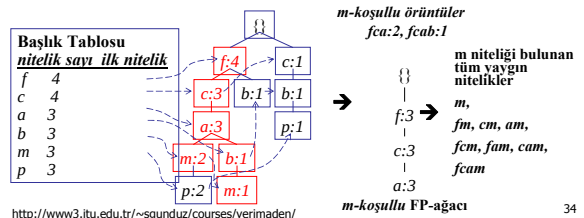


<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

33

Nitelik Koşullu Örüntülerden Koşullu FP-Ağaçları Oluşturma

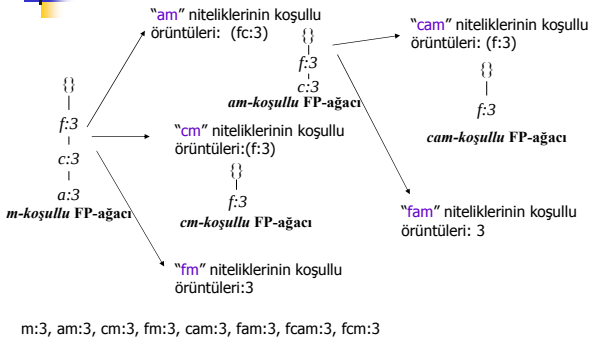
- Her koşullu örüntü için niteliklerin destek sayısı bulunur
- Yaygın nitelikler için koşullu FP-ağacı oluşturulur



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

34

Koşullu FP-Ağaçları



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

35

FP-Ağaçları ile Yaygın nitelikleri Bulma

- Yaygın niteliklere yinelemeli olarak yeni nitelikler ekleme
- Yöntem:
 - Her yaygın niteliğin koşullu örüntülerini ve koşullu FP-ağacını oluştur
 - İşlemi, yeni oluşturulan her koşullu FP-ağacı için tekrarla
 - Oluşturulan FP-ağaçlarında hiç nitelik bulunmayana kadar veya ağaçta tek bir dal kalana kadar işleme devam et
 - Ağaçta tek bir dal kaldığında yaygın nitelikler dalı oluşturan niteliklerin kombinasyonu

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

36

İlişkilendirme Kuralları Oluşturma

- L yaygın niteliklerden $f \subseteq L$ altkümelerinin bulunması
 - $f \rightarrow L-f$ kurallarının en küçük güven değeri koşulunu sağlaması gerekir
- Eğer $\{A, B, C, D\}$ yaygın nitelikler ise olası ilişkilendirme kuralları

$$\begin{array}{llll} ABC \rightarrow D, & ABD \rightarrow C, & ACD \rightarrow B, & BCD \rightarrow A, \\ A \rightarrow BCD, & B \rightarrow ACD, & C \rightarrow ABD, & D \rightarrow ABC \\ AB \rightarrow CD, & AC \rightarrow BD, & AD \rightarrow BC, & BC \rightarrow AD, \\ BD \rightarrow AC, & CD \rightarrow AB & & \end{array}$$
- $|L| = k$ için $2^k - 2$ ilişkilendirme kuralı adayı vardır
 - $L \rightarrow \emptyset$ ve $\emptyset \rightarrow L$ kuralları geçerli kurallar değildir

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

37

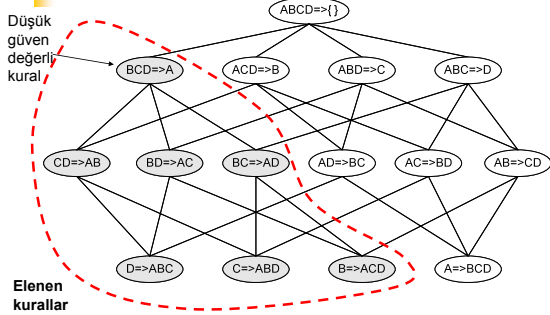
İlişkilendirme Kuralları Oluşturma

- İlişkilendirme kurallarının güven değerlerinin anti-monotone özelliği yok
 - $c(ABC \rightarrow D)$ değeri $c(AB \rightarrow D)$ değerinden küçük ya da büyük olabilir
- Aynı yaygın nitelikler kümesinden $L = \{A, B, C, D\}$ oluşan ilişkilendirme kurallarının güven değerleri için anti-monotone özelliği var
 - $c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$
 - İlişkilendirme kuralının solunda bulunan nitelik sayısı büyük olan kuralların güven değerleri de büyüktür.

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

38

Apriori Algoritması için Kural Oluşturma

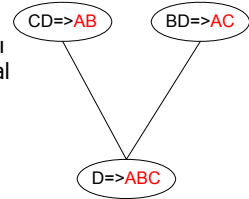


<http://www3.itu.edu.tr/~sgunduz/courses/verimaden/>

39

Apriori Algoritması için Kural Oluşturma

- Sağ taraflarındaki niteliklerin örnekleri aynı olan iki kural birleştirilir
- $(CD=>AB, BD=>AC)$ kuralları birleştirilerek $D=>ABC$ kuralı adayı oluşturulur.
- $AD=>BC$ kuralının güven değeri büyük değilse $D=>ABC$ kuralı elenir



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

40

Destek Değerinin Etkisi

- minsup değeri büyük belirlenirse veri kümesinden bazı örüntüler elde edilmeyebilir:
 - veri kümesinde daha az bulunan
 - önemli bilgi taşıyan
- minsup değeri küçük belirlenirse
 - yöntem karmaşıklaşır
 - çok fazla sayıda yaygın nitelikler kümesi elde edilir
- Tek bir destek değeri her zaman yeterli olmayabilir.

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

41

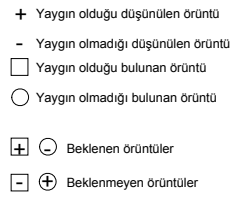
Bulunan Kuralların Önemi

- Tarafsız Ölçüt:
 - Örüntüler veri kümesinden elde edilen istatistiklere göre sıralanır
 - güven, destek, Jaccard, Gini, ...
- Taraflı Ölçüt
 - Örüntüler kullanıcının değerlendirmesine göre sıralanır
 - Bulunan örüntü kullanıcının beklentisi dışındaysa ilginçtir (Silberschatz & Tuzhilin)

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

42

-



43