

Web Madenciliđi (Web Mining)

Hazırlayan: M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliđi Bölümü

Bu dersin sunumları, "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Bing Liu, Springer, 2011." kitabı kullanılarak hazırlanmıştır.

Konular

- Birliktelik Kuralı Oluşturulması
 - Algoritma
- Birliktelik Kural Madenciliđi için Veri Formatları
- Çoklu Minimum Destek
- Sınıf Birliktelik Kural Madenciliđi
 - Algoritma

Birliktelik Kuralı Oluşturulması

- Bazı uygulamalarda **frequent itemset'leri kullanarak tüm birliktelik kurallarının oluşturulması gereklidir.**
- Bir f frequent itemset için **kural oluştururken f kümesinin tüm altkümeleri kullanılır.**
- **Confidence** değerine bağlı olarak bir frequent itemset için kural aşağıdaki gibi elde edilebilir:

$$(f - \alpha) \rightarrow \alpha, \text{ if}$$

$$\text{confidence} = \frac{f.\text{count}}{(f - \alpha).\text{count}} \geq \text{minconf},$$

- Burada $f.\text{count}$ ile $(f - \alpha).\text{count}$ tüm transaction kümesi T 'deki sayılarını gösterir.
- f frequent item için support değeri ise $f.\text{count} / n$ ile gösterilir.

Birliktelik Kuralı Oluşturulması

- Eğer f frequent itemset ise, Apriori algoritması **boş küme hariç tüm altkümelerinin de frequent itemset olmasını gerektirir.**
- Eğer $(f - \alpha) \rightarrow \alpha$ kuralı geçerliyse, $(f - \alpha_{sub}) \rightarrow \alpha_{sub}$ kuralları da geçerlidir.
- Eğer $\{A, B, C, D\}$ itemset için $(A, B \rightarrow C, D)$ kuralı varsa, $(A, B, C \rightarrow D)$ ve $(A, B, D \rightarrow C)$ kuralları da **kesinlikle geçerlidir.**
 $\{A, B, C\}.\text{count} \leq \{A, B\}.\text{count}$, $\{A, B, D\}.\text{count} \leq \{A, B\}.\text{count}$
 $(A, B, C \rightarrow D).\text{conf} \geq (A, B \rightarrow C, D).\text{conf}$
- Eğer $\{A, B, C, D\}$ itemset için $(A, B, C \rightarrow D)$ kuralı varsa, $(A, B \rightarrow C, D)$ ve $(A, C \rightarrow B, D)$ kuralları **geçerli olmayabilir.**
 $\{A, B\}.\text{count} \geq \{A, B, C\}.\text{count}$, $\{A, C\}.\text{count} \geq \{A, B, C\}.\text{count}$
 $(A, B, C \rightarrow D).\text{conf} \leq (A, B \rightarrow C, D).\text{conf}$

Konular

- Birliktelik Kuralı Oluşturulması
 - Algoritma
- Birliktelik Kural Madenciliği için Veri Formatları
- Çoklu Minimum Destek
- Sınıf Birliktelik Kural Madenciliği
 - Algoritma

5

Algoritma

- Öncelikle 1 elemana sahip ardıl kısımlardan oluşan kurallar elde edilir.

```
Algorithm genRules( $F$ ) //  $F$  is the set of all frequent itemsets
1 for each frequent  $k$ -itemset  $f_k$  in  $F$ ,  $k \geq 2$  do
2   output every 1-item consequent rule of  $f_k$  with confidence  $\geq minconf$  and
   support  $\leftarrow f_k.count / n$  //  $n$  is the total number of transactions in  $T$ 
3    $H_1 \leftarrow$  {consequents of all 1-item consequent rules derived from  $f_k$  above};
4   ap-genRules( $f_k$ ,  $H_1$ );
5 endfor
```

F_3 : {{Chicken, Clothes, Milk}:3}

Son elemanlardan oluşan küme
 $H_1 = \{\{Chicken\}, \{Milk\}\}$

1 elemanlı ardıl kısımlara sahip kurallar oluşturulur.

Rule 1: Chicken, Clothes \rightarrow Milk [sup = 3/7, conf = 3/3]
Rule 2: Chicken, Milk \rightarrow Clothes [sup = 3/7, conf = 3/4]
Rule 3: Clothes, Milk \rightarrow Chicken [sup = 3/7, conf = 3/3].

minsup = %30 ve minconf = %80

6

Algoritma

$F_3: \{\{\text{Chicken, Clothes, Milk}\}:3\}$
 $H_1 = \{\{\text{Chicken}\}, \{\text{Milk}\}\}$

```

Procedure ap-genRules( $f_k, H_m$ ) //  $H_m$  is the set of  $m$ -item consequents
1  if ( $k > m + 1$ ) AND ( $H_m \neq \emptyset$ ) then
2     $H_{m+1} \leftarrow$  candidate-gen( $H_m$ );  $H_2 = \{\{\text{Chicken, Milk}\}\}$ 
3    for each  $h_{m+1}$  in  $H_{m+1}$  do
4       $conf \leftarrow f_k.count / (f_k - h_{m+1}).count$ ;
5      if ( $conf \geq minconf$ ) then  $\{\text{Chicken, Clothes, Milk}\}.count / \text{Clothes}.count$ 
6        output the rule ( $f_k - h_{m+1}$ )  $\rightarrow h_{m+1}$  with confidence =  $conf$  and
          support =  $f_k.count / n$ ; //  $n$  is the total number of transactions in  $T$ 
7      else
8        delete  $h_{m+1}$  from  $H_{m+1}$ ;  $\text{Clothes} \rightarrow \text{Milk, Chicken}$ 
9    endfor
10   ap-genRules( $f_k, H_{m+1}$ );
11  endif

```

$F_3: \{\{\text{Chicken, Clothes, Milk}\}:3\}$
 $H_2 = \{\{\text{Chicken, Milk}\}\}$

Algoritma

Örnek

- Aşağıdaki T kümesi için $minsup = \%30$ ($3/7$) ve $minconf = \%80$ ($4/5$) olsun.
 - t_1 : Beef, Chicken, Milk
 - t_2 : Beef, Cheese
 - t_3 : Cheese, Boots
 - t_4 : Beef, Chicken, Cheese
 - t_5 : Beef, Chicken, Clothes, Cheese, Milk
 - t_6 : Chicken, Clothes, Milk
 - t_7 : Chicken, Milk, Clothes
- Aşağıdaki frequent itemset'ler elde edilir.
 - F_1 : $\{\{\text{Beef}\}:4, \{\text{Cheese}\}:4, \{\text{Chicken}\}:5, \{\text{Clothes}\}:3, \{\text{Milk}\}:4\}$
 - F_2 : $\{\{\text{Beef, Cheese}\}:3, \{\text{Beef, Chicken}\}:3, \{\text{Chicken, Clothes}\}:3, \{\text{Chicken, Milk}\}:4, \{\text{Clothes, Milk}\}:3\}$
 - F_3 : $\{\{\text{Chicken, Clothes, Milk}\}:3\}$.

Örnek – devam

- F_3 için kurallar aşağıdaki gibi elde edilebilir.
 - Rule 1: Chicken, Clothes \rightarrow Milk [sup = 3/7, conf = 3/3]
 - Rule 2: Chicken, Milk \rightarrow Clothes [sup = 3/7, conf = 3/4]
 - Rule 3: Clothes, Milk \rightarrow Chicken [sup = 3/7, conf = 3/3].
- minconf = %80 gerektiğinden dolayı **genRules()** algoritmasının **2.satırında** 1. ve 3. frequent itemset'ler alınır.
- Böylelikle $H_1 = \{\{Chicken\}, \{Milk\}\}$ olarak elde edilir.
- F_3 ve H_1 için ap-GenRules() algoritması çağrılır. **Bu algoritmanın 2.satırındaki candidate-gen()** fonksiyonuyla $H_2 = \{\{Chicken, Milk\}\}$ elde edilir.

Örnek – devam

- ap-GenRules() algoritmasının 2.satırında $H_2 = \{\{Chicken, Milk\}\}$ elde edildikten sonra aşağıdaki 4.kural oluşturulur:
 - Rule 4: Clothes \rightarrow Milk, Chicken [sup = 3/7, conf = 3/3].
- $F_3 = \{Chicken, Clothes, Milk\}$ frequent itemset'i kullanılarak aşağıdaki 4 kural elde edilmiş olur.
 - Rule 1: Chicken, Clothes \rightarrow Milk [sup = 3/7, conf = 3/3]
 - ~~Rule 2: Chicken, Milk \rightarrow Clothes [sup = 3/7, conf = 3/4].~~
 - Rule 3: Clothes, Milk \rightarrow Chicken [sup = 3/7, conf = 3/3].
 - Rule 4: Clothes \rightarrow Milk, Chicken [sup = 3/7, conf = 3/3].
- Bunlardan bir tanesi minconf değerini sağlamadığından silinir.

Konular

- Birliktelik Kuralı Oluşturulması
 - Algoritma
- Birliktelik Kural Madenciliği için Veri Formatları
- Çoklu Minimum Destek
- Sınıf Birliktelik Kural Madenciliği
 - Algoritma

11

Birliktelik Kural Madenciliği için Veri Formatları

- Birliktelik kural madenciliği ilişkisel veritabanlarında da uygulanabilir.
- VTYS içerisindeki tablonun bir transaction veri kümesine dönüştürülmesi gereklidir. Bunun için özellik-değer (**attribute-value**) ikilisinin elde edilmesi yeterlidir.
- Aşağıdaki tablodan transaction kümesi elde edilmiştir.

Attribute1	Attribute2	Attribute3
a	a	x
b	n	y

(A) Table data

t_1 : (Attribute1, a), (Attribute2, a), (Attribute3, x)

t_2 : (Attribute1, b), (Attribute2, n), (Attribute3, y)

(B) Transaction data

12

Birliktelik Kural Madenciliđi için Veri Formatları

- Eğer VTYS içerisindeki tablo kategorik veri yerine sayısal veri içeriyorsa deđer aralıđı belirli parçaya bölünerek dönüştürme işlemi yapılabilir.
- Örneđin 1-100 arasındaki sayısal deđerlere sahip bir alan için 1-20, 21-40, 41-60, 61-80, 81-100 aralıkları alınarak kategorik deđer ataması yapılabilir.

a = 1 - 20

b = 21 - 40

c = 41 - 60

d = 61 - 80

e = 81 - 100

13

Konular

- Birliktelik Kuralı Oluşturulması
 - Algoritma
- Birliktelik Kural Madenciliđi için Veri Formatları
- **Çoklu Minimum Destek**
- Sınıf Birliktelik Kural Madenciliđi
 - Algoritma

14

Çoklu Minimum Destek

- Birliktelik kural madenciliğinde tek minsup değeri kullanılması bazı uygulamalarda uygun olmayabilir.
- **Tüm veriler aynı öneme, sıklığa veya yapıya sahip olmayabilir.**
- **Bazı veriler sık olmasa da daha büyük öneme sahip olabilir.**
- Bu durumda bazı sorunlar ortaya çıkar:
 - **minsup değeri yüksek olursa** bazı önemli kurallar sık olmadığı için elde edilemez.
 - **minsup değeri düşük olursa** çok sayıda kural elde edilir ve birliktelik kural madenciliği oldukça zorlaşır.
- **E-ticaret sitelerinde bazı ürünler pahalıdır ve az sıklıkta satılır. Ancak bu ürünlere yönelik kurallarda önemlidir.**

15

Çoklu Minimum Destek

Örnek

- Bir süpermarkette az sayıda satılan FoodProcessor ve CookingPan ürünleri için kural bulmak istiyoruz.
- Az sıklıkta olduğundan minsup değerinin çok küçük olması gerekiyor.
- minsup = %0,006 için aşağıdaki gibi bir frequent itemset bulunabilir:
{FoodProcessor, CookingPan} [sup = 0.006%].
- Düşük minsup değerinden dolayı aşağıdaki anlamsız frequent itemset'lerde elde edilebilir:
 f_1 : {Bread, Cheese, Egg, Bagel, Milk, Sugar, Butter} [sup = 0.007%],
 f_2 : {Bread, Egg, Milk, CookingPan} [sup = 0.006%].
- Her iki frequent itemset gereksizdir. Çünkü hemen hemen her alışverişte bu item'lar yer alırlar.

16

Çoklu Minimum Destek

- Bu sorun veriyi bloklar halinde parçalayarak her parça için ayrı ayrı madencilik yapılarak çözülebilir. **Ancak, ayrı bloklardaki aynı itemler için kurallar elde edilemez.**
- Bu sorun için en iyi çözüm her item için farklı minimum support değeri (**minimum item support (MIS)**) belirlenmesidir.
- Bu sayede farklı sıklıkta ve farklı öneme sahip **item'lar için ayrı support değerleri kullanılarak anlamlı kurallar elde edilebilir.**
- MIS (i), i .item için minimum support değeri olsun. Bir R kuralı için minimum support değeri aşağıdaki ifade edilir:

$$R = i_1, i_2, \dots, i_k \rightarrow i_{k+1}, \dots, i_r,$$
$$\min(\text{MIS}(i_1), \text{MIS}(i_2), \dots, \text{MIS}(i_r)).$$

17

Çoklu Minimum Destek

Örnek

- Bir itemset {Bread, Shoes, Clothes} için kullanıcı tanımlı MIS değerleri aşağıdaki gibi olsun:

$$\text{MIS}(\text{Bread}) = 2\% \quad \text{MIS}(\text{Clothes}) = 0.2\% \quad \text{MIS}(\text{Shoes}) = 0.1\%.$$

- Aşağıdaki kural minimum support değerini karşılamaz. Çünkü, $\min(\text{MIS}(\text{Bread}), \text{MIS}(\text{Clothes})) = \%0,2$ 'dir (**Support değerinin 0,2'den büyük olması gerekir.**)

$$\text{Clothes} \rightarrow \text{Bread} \quad [\text{sup} = 0.15\%, \text{conf} = 70\%].$$

- Aşağıdaki kural minimum support değerini karşılar. Çünkü, $\min(\text{MIS}(\text{Shoes}), \text{MIS}(\text{Clothes})) = \%0,1$ 'dir.

$$\text{Clothes} \rightarrow \text{Shoes} \quad [\text{sup} = 0.15\%, \text{conf} = 70\%].$$

18

Çoklu Minimum Destek

Örnek

- {1, 2, 3, 4} itemset için MIS değerleri aşağıdaki olsun:
MIS(1) = 10% MIS(2) = 20% MIS(3) = 5% MIS(4) = 6%.
- Eğer 2.seviyede {1, 2} itemset için **support %9** bulunursa, **MIS(1) ve MIS(2) değerlerini sağlamaz**. Bu durumda, **Apriori algoritması {1, 2} itemset'i atar**.
- Böylelikle, **{1, 2, 3} ve {1, 2, 4} itemset'leri de üst seviyede oluşturulamaz**. Bu itemset'ler MIS(3) = %5 ve MIS(4) = %6 değerlerini sağlamaktadır.
- Ancak, **{1, 2} itemset'i silinmezse downward closure özelliği kaybolur**.

19

Çoklu Minimum Destek

- **Bir itemset içerisinde** çok sık ve çok az sayıda item'ların yer almasını önlemek için farklı yöntemler kullanılmaktadır.
- Örneğin, **maksimum ve minimum support değerlerinin uzaklığını (farkını) belirlemek için sabit fark kısıtı konulabilir**.
- Aşağıdaki ifade $sup(i)$, i .item için support değerini göstermektedir.

$$\max_{i \in S} \{sup(i)\} - \min_{i \in S} \{sup(i)\} \leq \varphi,$$

- Bu kısıt ile birlikte çok sayıda frequent itemset oluşturulması da engellenmiş olur.

20

Konular

- Birliktelik Kuralı Oluşturulması
 - Algoritma
- Birliktelik Kural Madenciliği için Veri Formatları
- Çoklu Minimum Destek
- Sınıf Birliktelik Kural Madenciliği
 - Algoritma

21

Sınıf Birliktelik Kural Madenciliği

- Bazı uygulamalarda **kurallar için kullanıcı hedef değer** belirleyebilir.
- **Örneğin bir grup doküman için hangi konuya yönelik olduğunu içeren bilgiye ulaşılmak istenebilir.**
- T kümesi n tane transaction'dan oluşursa, her transaction bir y sınıfı ile eşleştirilebilir.
- I tüm item'lar kümesi ve Y ise tüm sınıflar kümesi ise $I \cap Y = \emptyset$ dir.
- Bir **sınıf birliktelik kuralı (class association rule - CAR)** aşağıdaki gibi tanımlanır:
 $X \rightarrow y$, burada $X \subseteq I$, ve $y \in Y$ dir.
- Support ve confidence değerleri normal birliktelik kurallarındaki gibi hesaplanır.

22

Sınıf Birliklik Kural Madenciliği

- **CAR'larda hedef eleman bir tanedir ve bir sınıfı gösterir.**
- Normal birliklik kurallarında hedef bir item kümesidir ve birden fazla eleman olabilmektedir.
- **CAR madenciliğindeki amaç, kullanıcı tanımlı minsup ve minconf değerlerine sahip tüm kuralların bulunmasıdır.**

23

Sınıf Birliklik Kural Madenciliği

Örnek

- Aşağıda 8 doküman ile 2 sınıf arasındaki eşleştirme görülmektedir.

$I = \{\text{Student, Teach, School, City, Game, Baseball, Basketball, Team, Coach, Player, Spectator}\}$

$Y = \{\text{Education, Sport}\}$.

	Transactions	Class
doc 1:	Student, Teach, School	: Education
doc 2:	Student, School	: Education
doc 3:	Teach, School, City, Game	: Education
doc 4:	Baseball, Basketball	: Sport
doc 5:	Basketball, Player, Spectator	: Sport
doc 6:	Baseball, Coach, Game, Team	: Sport
doc 7:	Basketball, Team, City, Game	: Sport

- minsup = %20 ve minconf = %60 için iki kural aşağıdaki yazılabilir.

Student, School → Education [sup= 2/7, conf = 2/2]
Game → Sport [sup= 2/7, conf = 2/3].

24

Konular

- Birliktelik Kuralı Oluşturulması
 - Algoritma
- Birliktelik Kural Madenciliği için Veri Formatları
- Çoklu Minimum Destek
- Sınıf Birliktelik Kural Madenciliği
 - Algoritma

25

Algoritma

- CAR doğrudan tek adımla elde edilebilmektedir. Temel işlem minsup değerini sağlayan ruleitem'lerini bulmaktadır.
- Bir ruleitem
(*condset*, *y*)
şeklinde tanımlanır. Burada $condset \subseteq I$, ve $y \in Y$ dir.
- **condset için support değeri (condsupCount)**, *T* içerisinde **condset'i** bulunduran transaction sayısıdır.
- **ruleitem için support değeri (rulesupCount)**, *T* içerisinde **condset'i ve y sınıfını** birlikte bulunduran transaction sayısıdır.
- Her ruleitem aşağıdaki gibi bir kuralı gösterir:
condset → *y*

26

Algoritma

- $condset \rightarrow y$ şeklindeki bir kural için support ve confidence değeri
 $(condset \rightarrow y).support = (rulesupCount / n)$
 $(condset \rightarrow y).confidence = (rulesupCount / condsupCount)$
şeklinde ifade edilir.
- minsup değerini sağlayan ruleitem'lar frequent ruleitem'dir.

27

Algoritma

- ({Student, School}, Education) ruleitem için:

	Transactions	Class
doc 1:	Student, Teach, School	: Education
doc 2:	Student, School	: Education
doc 3:	Teach, School, City, Game	: Education
doc 4:	Baseball, Basketball	: Sport
doc 5:	Basketball, Player, Spectator	: Sport
doc 6:	Baseball, Coach, Game, Team	: Sport
doc 7:	Basketball, Team, City, Game	: Sport

$condsupCount(\{Student, School\}) = 2$

$rulesupCount(\{Student, School\}, Education) = 2$

$(\{Student, School\}, Education).support = 2/7$ (%28,6)

$(\{Student, School\}, Education).confidence = 2/2$ (%100)

28

Algoritma

Algorithm CAR-Apriori(T)

- 1 $C_1 \leftarrow \text{init-pass}(T);$ // the first pass over T
- 2 $F_1 \leftarrow \{f \mid f \in C_1, f.\text{rulesupCount} / n \geq \text{minsup}\};$
- 3 $CAR_1 \leftarrow \{f \mid f \in F_1, f.\text{rulesupCount} / f.\text{condsupCount} \geq \text{minconf}\};$
- 4 **for** ($k = 2; F_{k-1} \neq \emptyset; k++$) **do**
- 5 $C_k \leftarrow \text{CARcandidate-gen}(F_{k-1});$
- 6 **for each** transaction $t \in T$ **do**
- 7 **for each** candidate $c \in C_k$ **do**
- 8 **if** $c.\text{condset}$ is contained in t **then** // c is a subset of t
- 9 $c.\text{condsupCount}++;$
- 10 **if** $t.\text{class} = c.\text{class}$ **then**
- 11 $c.\text{rulesupCount}++$
- 12 **endfor**
- 13 **endfor**
- 14 $F_k \leftarrow \{c \in C_k \mid c.\text{rulesupCount} / n \geq \text{minsup}\};$
- 15 $CAR_k \leftarrow \{f \mid f \in F_k, f.\text{rulesupCount} / f.\text{condsupCount} \geq \text{minconf}\};$
- 16 **endfor**
- 17 **return** $CAR \leftarrow \bigcup_k CAR_k;$

$C_1 = \{(i, y) \mid i \in I, \text{ and } y \in Y\}$ |condSet| = 1

Aday frequent 1-ruleitem

|condSet| = 1 olan CAR kümesi

Yeni aday küme C_k oluşturuldu.
Aynı sınıfa ait olan condset'ler kullanılır.

condSet sayıldı

ruleitem sayıldı

Frequent item yapıldı.

CAR_k kümesine eklendi.

Tüm CAR kümesi elde edildi.

29

Algoritma

Örnek

- Aşağıdaki veri için minsup = %15 ($\geq 2/7$) ve minconf = %70 alınıyor.

	Transactions	Class
doc 1:	Student, Teach, School	: Education
doc 2:	Student, School	: Education
doc 3:	Teach, School, City, Game	: Education
doc 4:	Baseball, Basketball	: Sport
doc 5:	Basketball, Player, Spectator	: Sport
doc 6:	Baseball, Coach, Game, Team	: Sport
doc 7:	Basketball, Team, City, Game	: Sport

- İlk sayı condSupCount ikinci sayı ruleSupCount değerini gösterir.


$F_1:$ $\{(\{ \text{School} \}, \text{Education}) : (3, 3),$ $(\{ \text{Student} \}, \text{Education}) : (2, 2),$

$(\{ \text{Teach} \}, \text{Education}) : (2, 2),$ $(\{ \text{Baseball} \}, \text{Sport}) : (2, 2),$

$(\{ \text{Basketball} \}, \text{Sport}) : (3, 3),$ $(\{ \text{Game} \}, \text{Sport}) : (3, 2),$

$(\{ \text{Team} \}, \text{Sport}) : (2, 2)$

30



Algoritma

Örnek – devam


	Transactions	Class
doc 1:	Student, Teach, School	: Education
doc 2:	Student, School	: Education
doc 3:	Teach, School, City, Game	: Education
doc 4:	Baseball, Basketball	: Sport
doc 5:	Basketball, Player, Spectator	: Sport
doc 6:	Baseball, Coach, Game, Team	: Sport
doc 7:	Basketball, Team, City, Game	: Sport

■ İlk sayı condSupCount ikinci sayı ruleSupCount değerini gösterir.

F_1 : {({School}, Education):(3, 3), ({Student}, Education):(2, 2),
 ({Teach}, Education):(2, 2), ({Baseball}, Sport):(2, 2),
 ({Basketball}, Sport):(3, 3), ({Game}, Sport):(3, 2),
 ({Team}, Sport):(2, 2)}

CAR_1 : School → Education [sup = 3/7, conf = 3/3]
 Student → Education [sup = 2/7, conf = 2/2]
 Teach → Education [sup = 2/7, conf = 2/2]
 Baseball → Sport [sup = 2/7, conf = 2/2]
 Basketball → Sport [sup = 3/7, conf = 3/3]
 ~~Game → Sport~~ ~~[sup = 2/7, conf = 2/3]~~
 Team → Sport [sup = 2/7, conf = 2/2]

31



Algoritma

Örnek – devam

	Transactions	Class
doc 1:	Student, Teach, School	: Education
doc 2:	Student, School	: Education
doc 3:	Teach, School, City, Game	: Education
doc 4:	Baseball, Basketball	: Sport
doc 5:	Basketball, Player, Spectator	: Sport
doc 6:	Baseball, Coach, Game, Team	: Sport
doc 7:	Basketball, Team, City, Game	: Sport

■ İlk sayı condSupCount ikinci sayı ruleSupCount değerini gösterir.

C_2 : {({School, Student}, Education), ({School, Teach}, Education),
 ({Student, Teach}, Education), ({Baseball, Basketball}, Sport),
 ({Baseball, Game}, Sport), ({Baseball, Team}, Sport),
 ({Basketball, Game}, Sport), ({Basketball, Team}, Sport),
 ({Game, Team}, Sport)}

F_2 : {({School, Student}, Education):(2, 2),
 ({School, Teach}, Education):(2, 2), ({Game, Team}, Sport):(2, 2)}

CAR_2 : School, Student → Education [sup = 2/7, conf = 2/2]
 School, Teach → Education [sup = 2/7, conf = 2/2]
 Game, Team → Sport [sup = 2/7, conf = 2/2]

32

Ödev

- Birlikte kullanım kurallarının kullanım alanları hakkında detaylı bir araştırma ödevi hazırlayınız.