

# 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

- Frequent Itemsets ve Birliktelik Kuralları
- Frequent Itemsets ve Birliktelik Kurallarının Temelleri
- Support, Confidence ve Interest
- Apriori Algoritması
  - Frequent itemset oluşturulması
  - Algoritma analizi
- Büyük Veride Hash Tabanlı Frequent Itemset Bulma
  - PCY algoritması
  - Multistage algoritması
  - Multihash algoritması

## Frequent Itemsets ve Birliktelik Kuralları

- Frequent itemsets ve birliktelik kuralları (association rules) büyük veri analizinde önemli bir yere sahiptir.
- **Birliktelik kural madenciliği veri madenciliğinin temel çalışma alanlarından birisidir.**
- Birliktelik kuralları, üzerinde en çok çalışma yapılan ve yaygın uygulanan modeldir.
- **Birliktelik kuralları modelinin amacı, veri parçalarının birlikte gerçekleşme (co-occurrence) ilişkilerinin tümünü ortaya çıkarmaktır.**
- Agrawal ve arkadaşları\* tarafından 1993 yılında önerilmesinden sonra bir çok alanda başarılı bir şekilde uygulanmıştır.

\* R. Agrawal, T. Imielinski, and A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data (SIGMOD'93)*, pp. 207-216, 1993.

3

## Frequent Itemsets ve Birliktelik Kuralları

- Birliktelik kural madenciliğindeki klasik uygulama alanı market sepeti veri analizidir.
- Market sepeti analizinde, **müşterilerin aldıkları ürünler arasındaki ilişkiler ortaya çıkartılır.**

Peynir → Ekmek [support = 10%, confidence = 75%]

- Yukarıdaki kuralda, **tüm müşterilerin %10'u peynir ve ekmeği birlikte almıştır (support).**
- Yukarıdaki kuralda, **peynir alan müşterilerin %75'i ekmeği de almıştır (confidence).**

4

## Frequent Itemsets ve Birliktelik Kuralları

- **Birliktelik kural madenciliği ürünlerin sırasını dikkate almaz.**
- **Sıralı örüntü madenciliği ise ürünlerin sırasıyla ilgilenir.**
- Sıralı örüntü madenciliğinde “müşterilerin %5'i önce ekmek sonra peynir daha sonra zeytin almaktadır” şeklinde kural elde edilir.
- Bu tür kurallar Web kullanım madenciliğinde clickstream analizinde veya dilsel örüntülerin bulunması amacıyla kullanılır.

5

## Konular

- Frequent Itemsets ve Birliktelik Kuralları
- **Frequent Itemsets ve Birliktelik Kurallarının Temelleri**
- Support, Confidence ve Interest
- Apriori Algoritması
  - Frequent itemset oluşturulması
  - Algoritma analizi
- Büyük Veride Hash Tabanlı Frequent Itemset Bulma
  - PCY algoritması
  - Multistage algoritması
  - Multihash algoritması

6

## Frequent Itemsets ve Birliktelik Kurallarının Temelleri

- Birliktelik kural madenciliği aşağıdaki gibi ifade edilir:

$$I = \{i_1, i_2, \dots, i_m\}$$

$$T = (t_1, t_2, \dots, t_n)$$

- Burada,  $I$  parçalar (items) kümesi,  $T$  işlemler (transactions) kümesidir.
- Her bir işlem bir grup parçadan oluşan kümedir ve

$$t_i \subseteq I$$

şeklinde ifade edilir.

- **Birliktelik kuralı ise aşağıdaki gibi gösterilir:**

$$X \rightarrow Y, \text{ where } X \subset I, Y \subset I, \text{ and } X \cap Y = \emptyset$$

- Burada  $X$  ve  $Y$  **parça kümesi (itemset)** olarak ifade edilir.

## Frequent Itemsets ve Birliktelik Kurallarının Temelleri

### Örnek

- Bir müşteri  $I$  kümesindeki ürünlere sahip bir marketten üç tane ürün alıyor. Bu işlem aşağıdaki gibidir.

{Beef, Chicken, Cheese}

- **Bir birliktelik kuralı aşağıdaki gibi olabilir.**

Beef, Chicken  $\rightarrow$  Cheese

- Burada,

$X = \{\text{Beef, Chicken}\}$

$Y = \{\text{Cheese}\}$

şeklinde ifade edilir.

## Konular

- Frequent Itemsets ve Birliktelik Kuralları
- Frequent Itemsets ve Birliktelik Kurallarının Temelleri
- **Support, Confidence ve Interest**
- Apriori Algoritması
  - Frequent itemset oluşturulması
  - Algoritma analizi
- Büyük Veride Hash Tabanlı Frequent Itemset Bulma
  - PCY algoritması
  - Multistage algoritması
  - Multihash algoritması

## Support, Confidence ve Interest

### Support

- $t_i \in T$  transaction'ı,  $X$  itemset'ini kapsar (eğer  $X$  itemset'i  $t_i$  transaction'ının alt kümesi ise ( $X \subset t_i$ )).
- $X$  itemset'inin **destek sayısı (support count)**,  $T$  içerisinde  $X$ 'i kapsayan transaction sayısıdır ve  $X.count$  şeklinde gösterilir.
- $X \rightarrow Y$  kuralının **support değeri**,  $T$  tüm transaction'lar kümesinde  $X \cup Y$  itemset'ini içeren **transaction'ların oranıdır**.
- Bir kuralın **support değeri, olma olasılığını gösterir** ve  $\Pr(X \cup Y)$  şeklinde ifade edilir.
- $X \rightarrow Y$  kuralının support değeri aşağıdaki eşitlikle hesaplanır. Burada,  $n$  toplam transaction sayısını gösterir.

$$support = \frac{(X \cup Y).count}{n}$$

## Support, Confidence ve Interest

### Confidence

- $X \rightarrow Y$  kuralının **confidence değeri**,  $T$  tüm transaction'lar kümesinde  $X$  ile  $Y$  itemset'lerini **birlikte içeren transaction'ların,  $X$  i bulduran transactionlar içerisindeki oranıdır.**
- Bir kuralın **confidence değeri, şartlı olasılık ile ifade edilir ve  $\Pr(Y | X)$  şeklinde gösterilir.**
- $X \rightarrow Y$  kuralının confidence değeri aşağıdaki eşitlikle hesaplanır. Burada,  $n$  toplam transaction sayısını gösterir.

$$confidence = \frac{(X \cup Y).count}{X.count}$$

- **Confidence bir kuralın öngörülebilirliğini (predictability) belirler.**

11

## Support, Confidence ve Interest

### Interest

- $I \rightarrow j$  birliktelik kuralı için **interest değeri, kuralın *confidence* değeri ile  $j$ 'nin *support* değerinin farkıyla hesaplanır.**

- $Interest = (I \rightarrow j).conf - j.sup$

- **Sıfıra yakın değerler  $I \rightarrow j$  kuralının önemli olmadığını gösterir (rastgele dağılımda sıfır olur).**
- **Yüksek negatif değerler  $I$ 'nin  $j$  ile birlikte olmadığını (önemli) gösterir.**
- **Yüksek pozitif değerler  $I \rightarrow j$  kuralının önemli olduğunu gösterir.**
- $\{dog\} \rightarrow cat$ , kuralının interest değeri =  $5/7 - 6/8 = -0,036$  dir.
- $\{dog\} \rightarrow cat$ , birliktelik kuralı önemli değildir.
- $\{cat\} \rightarrow kitten$ , kuralının interest değeri =  $1/6 - 1/8 = 0,042$  dir.

1. {Cat, and, dog, bites}
2. {Yahoo, news, claims, a, cat, mated, with, a, dog, and, produced, viable, offspring}
3. {Cat, killer, likely, is, a, big, dog}
4. {Professional, free, advice, on, dog, training, puppy, training}
5. {Cat, and, kitten, training, and, behavior}
6. {Dog, &, Cat, provides, dog, training, in, Eugene, Oregon}
7. {"Dog, and, cat", is, a, slang, term, used, by, police, officers, for, a, male-female, relationship}
8. {Shop, for, your, show, dog, grooming, and, pet, supplies}

12

## Support, Confidence ve Interest

- **Birliktelik kural madenciliği**, verilen bir  $T$  transaction'lar kümesinde, **önceden belirlenmiş support ve confidence değerlerine eşit veya büyük olan birliktelik kurallarının elde edilmesini amaçlar.**
- Kullanıcı tarafından belirlenmiş olan support ve confidence değerlerine **minimum support (minsup)** ve **minimum confidence (minconf)** denilmektedir.

13

## Support, Confidence ve Interest

### Örnek

- Aşağıda 7 transaction'a sahip olan  $T$  kümesi verilmiştir.
- $I$  kümesi markette satılan tüm parçaları içermektedir.
- **minsup = 30%** ve **minconf = 80%** için kurallar bulalım.

t<sub>1</sub>: Beef, Chicken, Milk  
t<sub>2</sub>: Beef, Cheese  
t<sub>3</sub>: Cheese, Boots  
t<sub>4</sub>: Beef, Chicken, Cheese  
t<sub>5</sub>: Beef, Chicken, Clothes, Cheese, Milk  
t<sub>6</sub>: Chicken, Clothes, Milk  
t<sub>7</sub>: Chicken, Milk, Clothes

- Transaction'lar için aşağıdaki kurallar **support = 42,86% (>30%)** ve **confidence = 100% (>80%)** değerine sahiptir ve geçerlidir.

Chicken, Clothes → Milk [sup = 3/7, conf = 3/3]

Clothes → Milk, Chicken [sup = 3/7, conf = 3/3]

14

## Konular

- Frequent Itemsets ve Birliktelik Kuralları
- Frequent Itemsets ve Birliktelik Kurallarının Temelleri
- Support, Confidence ve Interest
- **Apriori Algoritması**
  - Frequent itemset oluşturulması
  - Algoritma analizi
- Büyük Veride Hash Tabanlı Frequent Itemset Bulma
  - PCY algoritması
  - Multistage algoritması
  - Multihash algoritması

15

## Apriori Algoritması

- Apriori algoritması iki aşamadan oluşmaktadır:
  - **Tüm frequent itemset'lerin oluşturulması.**  
Frequent itemset, **minsup** değerine sahip item'lar kümesidir.
  - **Frequent itemset'lerden güvenilir birliktelik kurallarının oluşturulması.**  
Güvenilir birliktelik kuralı, **minconf** değerine sahip kuraldır.
- Bir itemset içindeki **eleman sayısı k ise** bu itemset **k-itemset** olarak adlandırılır.
- Aşağıdaki itemset **minsup** değerini sağlıyorsa, **frequent 3-itemset'tir.**

{Chicken, Clothes, Milk}

16



## Apriori Algoritması

- **minsup = 30%** ve **minconf = 80%** değerlerini alalım.

{Chicken, Clothes, Milk}

- Yukarıdaki frequent 3-itemset için minsup ve minconf değerlerini sağlayan aşağıdaki birliktelik kuralları yazılabilir.

Rule 1:	Chicken, Clothes → Milk	[sup = 3/7, conf = 3/3]
Rule 2:	Clothes, Milk → Chicken	[sup = 3/7, conf = 3/3]
Rule 3:	Clothes → Milk, Chicken	[sup = 3/7, conf = 3/3]

17

## Konular

- Frequent Itemsets ve Birliktelik Kuralları
- Frequent Itemsets ve Birliktelik Kurallarının Temelleri
- Support, Confidence ve Interest
- Apriori Algoritması
  - Frequent itemset oluşturulması
  - Algoritma analizi
- Büyük Veride Hash Tabanlı Frequent Itemset Bulma
  - PCY algoritması
  - Multistage algoritması
  - Multihash algoritması

18

## Frequent itemset oluşturulması

- **Downward closure property**  
Eğer bir itemset minsup değerine sahipse, bu itemset'in boş küme hariç tüm altkümeleri de minsup değerine sahiptir.
- **Apriori property**  
Bir transaction,  $X$ 'deki item'lara sahipse,  $X$ 'in boş küme hariç tüm alt kümelerine de sahip olmak zorundadır.
- **Pruning**  
Apriori algoritması minsup değerine sahip olmayan birliktelik kural adaylarını temizler.

19

## Frequent itemset oluşturulması

- Algoritma  $I$  itemset içerisindeki elemanların tümüyle sıralı (**lexicographic order**) olduğunu varsayar.
- Bir  $k$ -itemset aşağıdaki gibi gösterilir ve  $w[1], w[2], \dots, w[k]$  birer item'dir.  
 $\{w[1], w[2], \dots, w[k]\}$
- Apriori algoritması tüm frequent itemset'lerin verileri üzerinden birden fazla geçerek işlem yapar.
- **Apriori algoritması level-wise search yapar** ve her geçişte her bir item için support değerlerini ve frequent olup olmadıklarını belirler.
- Önce her bir item için frequent 1-itemset'i oluşturur ve her iterasyonda 2-itemset, 3-itemset şeklinde artarak devam eder.

20

## Konular

- Frequent Itemsets ve Birliktelik Kuralları
- Frequent Itemsets ve Birliktelik Kurallarının Temelleri
- Support, Confidence ve Interest
- Apriori Algoritması
  - Frequent itemset oluşturulması
  - **Algoritma analizi**
- Büyük Veride Hash Tabanlı Frequent Itemset Bulma
  - PCY algoritması
  - Multistage algoritması
  - Multihash algoritması

21

## Algoritma analizi

Algorithm Apriori( $T$ )

```
1  $C_1 \leftarrow \text{init-pass}(T);$  // the first pass over  $T$ 
2  $F_1 \leftarrow \{f \mid f \in C_1, f.\text{count}/n \geq \text{minsup}\};$  //  $n$  is the no. of transactions in  $T$ 
3 for ( $k=2; F_{k-1} \neq \emptyset; k++$ ) do // subsequent passes over  $T$ 
4    $C_k \leftarrow \text{candidate-gen}(F_{k-1});$  // Candidate k-itemset oluşturulur.
5   for each transaction  $t \in T$  do // scan the data once
6     for each candidate  $c \in C_k$  do
7       if  $c$  is contained in  $t$  then
8          $c.\text{count}++;$ 
9     endfor
10  endfor
11   $F_k \leftarrow \{c \in C_k \mid c.\text{count}/n \geq \text{minsup}\}$ 
12 endfor
13 return  $F \leftarrow \bigcup_k F_k;$ 
```

Her item için support değeri hesaplanır. Candidate 1-itemset oluşturulur.

1-itemset

Candidate k-itemset oluşturulur.

Candidate k-itemset elemanlarının T içerisindeki sayısı (support) bulunur.

Frequent k-itemset oluşturulur.

22

## Algoritma analizi

```
Function candidate-gen( $F_{k-1}$ )
1  $C_k \leftarrow \emptyset$ ; // initialize the set of candidates
2 forall  $f_1, f_2 \in F_{k-1}$  // find all pairs of frequent itemsets
3   with  $f_1 = \{i_1, \dots, i_{k-2}, i_{k-1}\}$  // that differ only in the last item
4   and  $f_2 = \{i_1, \dots, i_{k-2}, i'_{k-1}\}$ 
5   and  $i_{k-1} < i'_{k-1}$  do // according to the lexicographic order
6      $c \leftarrow \{i_1, \dots, i_{k-1}, i'_{k-1}\}$ ; // join the two itemsets  $f_1$  and  $f_2$ 
7      $C_k \leftarrow C_k \cup \{c\}$ ; // add the new itemset  $c$  to the candidates
8     for each  $(k-1)$ -subset  $s$  of  $c$  do
9       if  $\{s \notin F_{k-1}\}$  then // delete  $c$  from the candidates
10        delete  $c$  from  $C_k$ ;
11     endfor
12 endfor
13 return  $C_k$ ; // return the generated candidates
```

Son elemanlar farklı

Joining aşaması

Pruning aşaması

23

## Algoritma analizi

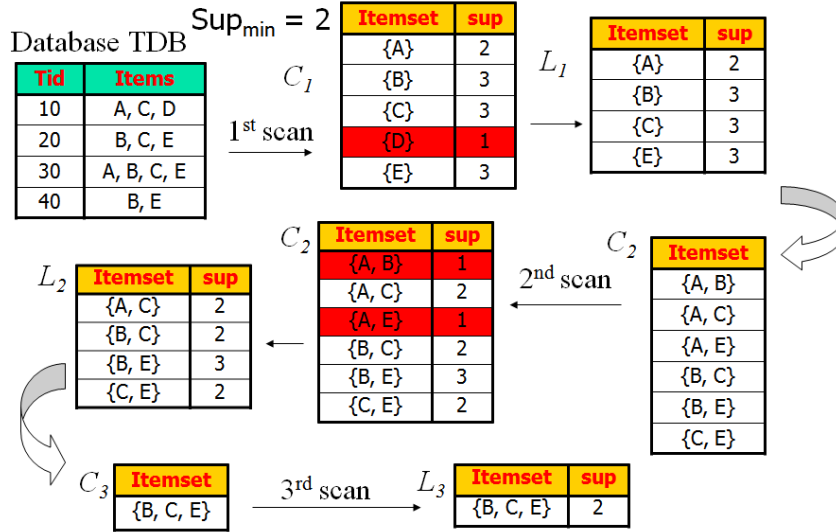
### Örnek

- Aşağıda 3. seviyede oluşturulan frequent 3-itemset verilmiştir.  
 $F_3 = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 3, 4\}\}$
- Join aşamasında  $\{1, 2, 3, 4\}$  ve  $\{1, 3, 4, 5\}$  oluşturulur.
- $\{1, 2, 3, 4\}$  birinci ve ikinci itemset'leri ile oluşturulur.
- $\{1, 3, 4, 5\}$  ise  $\{1, 3, 4\}$  ile  $\{1, 3, 5\}$  itemset'leri ile oluşturulur.
- Pruning aşamasından sonra sadece  $\{1, 2, 3, 4\}$  kalır.
- $\{1, 4, 5\}$  kümesi ve  $\{3, 4, 5\}$  kümesi, frequent 3-itemset içerisinde olmadığından  $\{1, 3, 4, 5\}$  silinir.

24

## Algoritma analizi

Örnek:  $\text{minsup}=2$



25

## Algoritma analizi

Örnek

- Aşağıda 7 transaction'a sahip bir  $T$  kümesi verilmiştir.
- $\text{minsup} = 30\%$  (en az 3/7) olarak alarak tüm frequent itemset'leri bulalım.

- $t_1$ : 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

26

## Algoritma analizi

### Örnek – devam

- Her item sonunda support değerleri verilmiştir. **Support değerinin en az 3 olması gereklidir** ( $3/7 > 30\%$ ).

$F_1$ : {{Beef}:4, {Cheese}:4, {Chicken}:5, {Clothes}:3, {Milk}:4}

$C_2$ : {{Beef, Cheese}, {Beef, Chicken}, {Beef, Clothes}, {Beef, Milk},  
{Cheese, Chicken}, {Cheese, Clothes}, {Cheese, Milk},  
{Chicken, Clothes}, {Chicken, Milk}, {Clothes, Milk}}

$F_2$ : {{Beef, Chicken}:3, {Beef, Cheese}:3, {Chicken, Clothes}:3,  
{Chicken, Milk}:4, {Clothes, Milk}:3}

$C_3$ : {{Chicken, Clothes, Milk}}

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

{Beef, Cheese, Chicken} oluşturulur.  
Ancak, {Cheese, Chicken}  $F_2$  içinde olmadığından, {Beef, Cheese, Chicken} silinir ve  $C_3$  içerisinde yer almaz.

27

## Algoritma analizi

- Exponential bir algoritmadır.  $I$  kümesindeki eleman sayısı  $m$  ise  $O(2^m)$  karmaşıklığına sahiptir.
- Market sepeti örneğinde olduğu gibi, market çok sayıda ürünü satar ancak müşteri bir kısmını alır (**sparseness**). Algoritma transaction kümesindeki veri seyrekliğine göre daha etkin çalışır.
- Algoritma tüm veriyi hafızaya yüklemeyi çalışır ve verileri teker teker tarayıp işlem yapar (sıra 5-10 arasında). **Ölçeklenebilirliği yüksektir** ve çok büyük boyutlardaki veriler üzerinde çalışabilir.
- Level-wise search** yaptığından istenilen seviyede sonlandırılabilir.
- Birliktelik kurallarında çok sayıda (binlerce) kural oluşturulabilir ve faydalı olanların bulunması çok zordur (**Interestingness problem**).

28

## Konular

- Frequent Itemsets ve Birliktelik Kuralları
- Frequent Itemsets ve Birliktelik Kurallarının Temelleri
- Support, Confidence ve Interest
- Apriori Algoritması
  - Frequent itemset oluşturulması
  - Algoritma analizi
- **Büyük Veride Hash Tabanlı Frequent Itemset Bulma**
  - PCY algoritması
  - Multistage algoritması
  - Multihash algoritması

29

## Büyük Veride Hash Tabanlı Frequent Itemset Bulma

- **A-Priori algoritması aday itemsetler ile frequent itemset'lerin adedini tutmak için hafızada çok alan gerektirir.**
- Itemset'lerin adetlerinin tutulduğu tamsayı değerler hafızaya sığmazsa disk ile hafıza arasında çok kez okuma ve yazma yapılır (**thrashing**).
- **Üst seviye itemset'lerde (3-frequent, 4-frequent, ...) hafıza gereksinimi düşer.**
- A-Priori algoritmasında **en çok hafıza gereksinimi 1-elemanlı aday itemset'ler ile 1-frequent itemset'ler** içindir.

30

## Konular

- Frequent Itemsets ve Birliktelik Kuralları
- Frequent Itemsets ve Birliktelik Kurallarının Temelleri
- Support, Confidence ve Interest
- Apriori Algoritması
  - Frequent itemset oluşturulması
  - Algoritma analizi
- Büyük Veride Hash Tabanlı Frequent Itemset Bulma
  - **PCY algoritması**
  - Multistage algoritması
  - Multihash algoritması

31

## PCY algoritması

- Park, Chen ve Yu (PCY) algoritması, **A-Priori algoritmasındaki ilk geçişte ayrılan hafıza alanı gereksinimini azaltır.**
- PCY algoritması **Bloom filtresi yaklaşımını kullanır.**
- Itemset'lerin **adetlerini tutan dizi bir hash tablosu** olarak alınır.
- Hash tablosundaki **her bucket** tamsayı olarak **adedi tutar.**
- **Her item çifti** hash tablosundaki **bucket'lara eşleştirilir.**
- Her sepette tüm çiftler oluşturulur ve **her çift için eşleştiği bucket değeri 1 artırılır.**
- **İlk geçişten sonra her bucket kendisine eşleştirilen çiftler için toplam adedi tutar.**

32



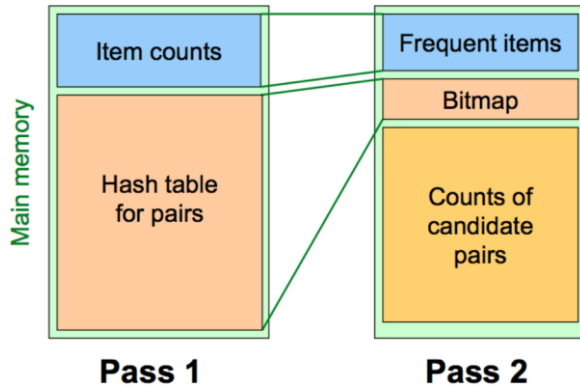
## PCY algoritması

- Bir bucket'ın değeri threshold değerinden büyükse frequent bucket olarak alınır.
- Frequent bucket'ı eşleştiren çiftler aday frequent itemset olarak alınır.
- Infrequent bucket'ı eşleştiren çiftlerin hiçbirisi frequent itemset olamazlar.
- $\{i, j\}$  aday çiftleri aşağıdaki şekilde oluşturulur:
  - $i$  ve  $j$  item'ları 1-frequent item'dır.
  - $\{i, j\}$  çifti frequent bucket'ı eşleştirir.
- En kötü durumda tüm bucket'lar frequent olur.
- Infrequent bucket sayısı arttıkça, PCY daha az hafıza alanı gerektirir.

33

## PCY algoritması

- İkinci geçişe gelmeden önce PCY algoritması hash tablosunu bitmap şeklinde özetler.
- Bitmap'teki her bit hash tablosundaki bir bucket'ı özetler.
- Bucket'taki 4 byte (32 bit) değer 1 bitle gösterilir.
- Frequent bucket için bitmap'teki bit 1, infrequent bucket için 0 yapılır.



34

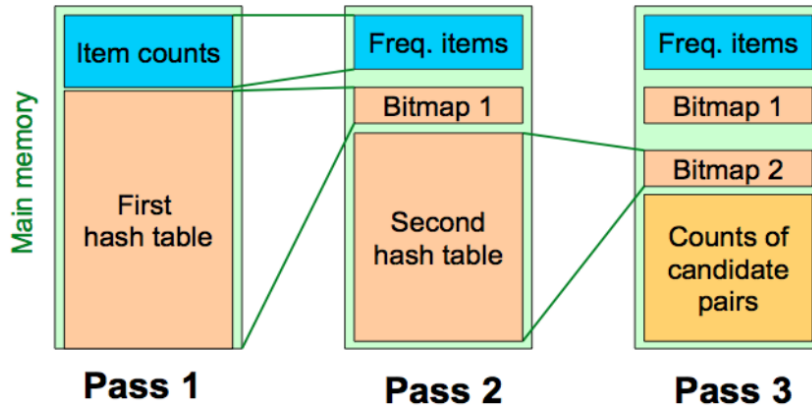
## Konular

- Frequent Itemsets ve Birliktelik Kuralları
- Frequent Itemsets ve Birliktelik Kurallarının Temelleri
- Support, Confidence ve Interest
- Apriori Algoritması
  - Frequent itemset oluşturulması
  - Algoritma analizi
- Büyük Veride Hash Tabanlı Frequent Itemset Bulma
  - PCY algoritması
  - **Multistage algoritması**
  - Multihash algoritması

35

## Multistage algoritması

- Multistage algoritması, art arda çok sayıda hash tablosu kullanarak aday çift sayısını PCY algoritmasına göre düşürür.



- Multistage algoritmasında ilk geçiş PCY ile aynıdır.

36

## Multistage algoritması

- İkinci geçişte, farklı bir hash tablosu ve farklı bir hash fonksiyonu ile ikinci bir bitmap oluşturulur.
- Her iki hash değerine göre kara verilir.
- $\{i, j\}$  aday çiftleri aşağıdaki şekilde oluşturulur:
  - $i$  ve  $j$  frequent item'dır.
  - $\{i, j\}$  ilk geçişte hash tablosunda frequent bucket'a eşleştirilmiştir.
  - $\{i, j\}$  ikinci geçişte hash tablosunda frequent bucket'a eşleştirilmiştir.

37

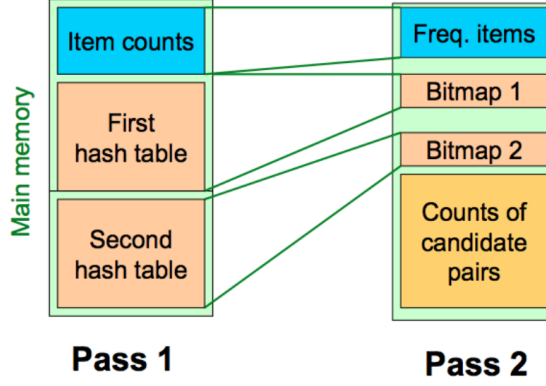
## Konular

- Frequent Itemsets ve Birliktelik Kuralları
- Frequent Itemsets ve Birliktelik Kurallarının Temelleri
- Support, Confidence ve Interest
- Apriori Algoritması
  - Frequent itemset oluşturulması
  - Algoritma analizi
- Büyük Veride Hash Tabanlı Frequent Itemset Bulma
  - PCY algoritması
  - Multistage algoritması
  - Multihash algoritması

38

## Multihash algoritması

- Multihash algoritması, art arda geçişte iki farklı hash tablosu kullanmak yerine **iki ayrı hash fonksiyonu** ve **iki ayrı hash tablosu kullanır**.
- Hash tabloları multistage'e göre daha küçük boyuttadır.



- Multihash algoritmasında **iki bitmap'te de frequent olanlar aday çift olarak alınır**.

39