

Büyük Veri Analitiği (Big Data Analytics)

M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman, Stanford University, 2011." kitabı kullanılarak hazırlanmıştır.

Konular

- **Frequent Itemsets**
 - Market sepeti modeli
 - Market sepeti veri gösterimi
 - Frequent itemsets uygulamaları
- **A-Priori Algoritması**
- **Hafızada Büyük Veri Kümelerinde İşlem**
 - PCY algoritması
 - Multistage algoritması
 - Multihash algoritması

Frequent Itemsets

- Frequent itemset'ler ve birliktelik kuralları farklı alanlardaki uygulamalarda yaygın kullanılmaktadır.
- **Market sepeti modelinde**, item'lar ile sepetler arasında **many-to-many** ilişki belirlenir.
- **Frequent itemsets problemi**, aynı sepette bulunan item kümeleriyle ilgilenir.
- **A-Priori algoritması** küçük kümelerden başlayarak **büyük kümelerin frequent itemset olup olmadığına karar verir.**
- Frequent itemset'lerin tamamını bulmak yerine, **yaklaşık frequent itemset bulan algoritmalar daha hızlı sonuç üretir.**

3

Konular

- Frequent Itemsets
 - Market sepeti modeli
 - Market sepeti veri gösterimi
 - Frequent itemsets uygulamaları
- A-Priori Algoritması
- Hafızada Büyük Veri Kümelerinde İşlem
 - PCY algoritması
 - Multistage algoritması
 - Multihash algoritması

4

Market sepeti modeli

- **Market sepeti modeli**, iki türe ait nesnelerin **many-to-many** ilişkisini tanımlamak için kullanılır.
- **Veri türleri**, **parçalar (items)** ve **sepetler (transactions)** olarak tanımlanır.
- **Her sepet bir item kümesine sahiptir (itemset).**
- Genellikle, **bir sepet içindeki item sayısı, toplam item sayısına göre çok küçüktür.**
- Toplam sepet sayısı genellikle hafızaya sığmayacak kadar fazladır.
- **Veri, market sepetlerinin sıralı gösterimi şeklinde ifade edilir.**
- **Çok sayıda sepette yer alan item kümesi frequent** olarak ifade edilir.
- s bir eşik destek değeri (**minimum support value**) ve I item kümesi iken, I item kümesinin **alt kümesi olduğu sepet sayısı, s değerine eşit veya büyükse I frequent itemset** olarak adlandırılır.

5

Market sepeti modeli

- Aşağıda her kelime bir item, her küme bir sepettir.

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}

- Dog kelimesi 7 sepette vardır ve support değeri 7/8 olur.
- Cat kelimesi 6 sepette vardır ve support değeri 6/8 olur.

6

Market sepeti modeli

- Minimum support değeri $s = 3$ olsun.
- Tek elemanlı 5 frequent itemset vardır.
 $\{dog\}, \{cat\}, \{and\}, \{a\}, \{training\}$
- İki elemanlı frequent itemset'lerde her iki elemanda frequent item olmak zorundadır.
- 1-frequent itemset'lerin birlikte yer alabileceği 10 olasılık vardır.

	training	a	and	cat
dog	4, 6	2, 3, 7	1, 2, 8	1, 2, 3, 6, 7
cat	5, 6	2, 3, 7	1, 2, 5	
and	5	2, 7		
a	none			

- İki elemanlı (**2-frequent itemset**) 5 tane frequent itemset vardır.
 $\{dog, a\}:3, \{dog, and\}:3, \{dog, cat\}:5, \{cat, a\}:3, \{cat, and\}:3$

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}

Market sepeti modeli

- Üç elemanlı frequent itemset'lerde her iki elemanlı alt küme frequent itemset olmak zorundadır.
- $\{dog, a, and\}$ üçlüsü, $\{dog, a\}$ alt kümesi 2-frequent itemset olmadığı için **3-frequent itemset** olamaz.
- $\{dog, cat, and\}$ üçlüsü, ikili alt kümelerinin tümü 2-frequent itemset olduğu için **aday 3-frequent itemset'tir**.
- $\{dog, cat, and\}$ üçlüsü **sadece 1 ve 2. sepetlerde birlikte yer aldığı için 3-frequent itemset** olamaz.
- $\{dog, cat, a\}$ üçlüsü, ikili alt kümelerinin tümü 2-frequent itemset olduğu için **aday 3-frequent itemset'tir**.
- $\{dog, cat, a\}$ üçlüsü, **2, 3 ve 7. sepetlerde birlikte yer aldığı için 3-frequent itemset'tir**.

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}

Konular

- Frequent Itemsets
 - Market sepeti modeli
 - **Market sepeti veri gösterimi**
 - Frequent itemsets uygulamaları
- A-Priori Algoritması
- Hafızada Büyük Veri Kümelerinde İşlem
 - PCY algoritması
 - Multistage algoritması
 - Multihash algoritması

9

Market sepeti veri gösterimi

- Market sepeti verisi bir file içerisinde sepetler halinde saklanır.
- **Veri dağıtık saklanabilir veya klasik bir dosyada saklanabilir.**
 $\{23, 456, 1001\}\{3, 18, 92, 145\}\{ \dots$
- Veri üzerindeki **tüm işlemler bir makine tarafından yapılabilir** veya uygulamaya göre **MapReduce** gibi yöntemlerle parçalanabilir.
- **Dağıtık işlem sonuçlarını birleştirerek genel bir threshold değerine göre itemset elde etmek bazı uygulamalarda zordur** (frequent itemset).
- **Dosya boyutları genellikle çok büyüktür ve hafızaya sığmaz.**
- Diskten okuma süresi en önemli maliyeti oluşturur.
- Genellikle **sepet boyutları küçüktür** ve üzerindeki **işlem süresi diskten okumaya göre çok küçüktür.**
- Uygulamalarda frequent itemset genellikle küçük boyuttadır (2 veya 3).

10

Market sepeti veri gösterimi

- Frequent itemset algoritmaları veri üzerinden geçerken farklı hesaplamalar da yapabilir (Örn. verideki tüm çiftlerin bulunması).
- Tüm veri hafızada tutulmazsa çiftlerin adetlerinin hesaplanması için diskten okuma gereklidir.

Örnek

- Bir algoritma ile n item içerisindeki tüm çiftlerin kombinasyonunu saymak istiyoruz.
- $C(n, 2) = n! / ((n-2)! 2!)$ adet tamsayı (yaklaşık $n^2/2$) saklamak için hafızada alana ihtiyaç duyulur (sayaç adedi).
- Her tamsayı 4 byte ile saklanırsa $2n^2$ byte alana ihtiyaç duyulur.
- Hafızada ayrılan alan 2GB ($2n^2 = 2^{31}$ byte) ise $n \leq 2^{15}$ (yaklaşık $n \leq 32.000$ adet çift) olur.

11

Market sepeti veri gösterimi

Üçgen matris yöntemi

- n elemanlı bir kümede bulunan çiftlerin adetleri tamsayı değerlerden oluşan matris kullanılarak saklanabilir.
- Matriste $\{i, j\}$ ikilisi için i . satır ve j . sütuna değer yazılır, ancak $\{j, i\}$ boş kalır. Matrisin yarısı kullanılmaz.
- Çiftlerin adetleri üçgen dizi ile tutulabilir.
- $a[k]$ dizi elemanı $\{i, j\}$ çiftinin adedini tutar (s = matris satır/sütun sayısı).

$$k = (s * i) + j - ((i * (i + 1)) / 2)$$

	0	1	2	3
0	a	b	c	d
1		e	f	g
2			h	i
3				j



	0	1	2	3	4	5	6	7	8	9
	a	b	c	d	e	f	g	h	i	j

- Tüm çiftlerin adetleri $\{1, 2\}, \{1, 3\}, \dots, \{1, n\}, \{2, 3\}, \{2, 4\}, \dots, \{2, n\}, \dots, \{n-2, n-1\}, \{n-2, n\}, \{n-1, n\}$ şeklinde tek boyutlu dizide tutulur.

12

Market sepeti veri gösterimi

Monotonicity

- Eğer, I item kümesi frequent ise, I kümesinin tüm altkümeleri de frequent'tir (monotonicity).
- $J \subseteq I$ ise, I daki item'ları bulunduran sepetler, J daki item'ları da kesinlikle bulundurur ($J.support \geq I.support$).
- {dog, cat, and} frequent itemset ise {dog, cat}, {dog, and}, {cat, and} itemset'lerin tamamı da frequent itemset'tir.
- Bir frequent itemset ile **frequent superset oluşturulamıyorsa** bu itemset **maximal** olarak adlandırılır.
- {dog, cat, and} frequent itemset, herhangi bir 4-frequent itemset'in alt kümesi değilse maximal itemset'tir.

13

Konular

- Frequent Itemsets
 - Market sepeti modeli
 - Market sepeti veri gösterimi
 - Frequent itemsets uygulamaları
- A-Priori Algoritması
- Hafızada Büyük Veri Kümelerinde İşlem
 - PCY algoritması
 - Multistage algoritması
 - Multihash algoritması

14

Frequent itemsets uygulamaları

- Market sepeti modeli gerçek alışveriş sepeti analizinde kullanılmaktadır.
- Süpermarketler veya mağaza zincirleri satılan tüm ürünleri kaydeder.
- **Bir müşterinin alışverişi bir sepeti, sepetteki her ürün item'ı ifade eder.**
- **Frequent itemset'ler bulunarak birlikte alınan ürünler belirlenir.**

Related concepts

- **Item'lar kelime ve sepetler ise doküman** (Web page, blog, tweet, ...) olarak alınır.
- **Bir doküman (sepet) kelimeleri (item'ları) içerir.**
- Aynı dokümanlarda **birlikte sık geçen kelimeler aranır**sa stopword'ler (gibi, ile, a, the, ...) elde edilir.
- **Tüm dokümanlarda yaygın kullanılmayan kelime çiftleri dokümanın konusunu daha çok yansıtır.**

15

Frequent itemsets uygulamaları

Plagiarism

- Dokümanların benzerliğinin bulunmasında kullanılır.
- **Aynı cümlelerin farklı dokümanlarda birlikte yer almasına bakılır.**

Biomarkers

- Item'lar **iki tür biomarker ile ifade edilebilir: kan proteinler/genler ve hastalıklar.**
- **Her sepet bir hasta hakkındaki bilgileri içerir** (genome, kan analiz değerleri, hastalığın medikal geçmişi, ...).
- **Bir frequent itemset bir hastalık ile bir veya daha fazla biomarker'ı içerir.**
- Bir hastalıkta yer alan biomarker'lar belirlenebilir.
- Benzer hastalığa neden olan biomarker'lar belirlenebilir.

16

Konular

- Frequent Itemsets
 - Market sepeti modeli
 - Market sepeti veri gösterimi
 - Frequent itemsets uygulamaları
- A-Priori Algoritması
- Hafızada Büyük Veri Kümelerinde İşlem
 - PCY algoritması
 - Multistage algoritması
 - Multihash algoritması

17

A-Priori Algoritması

- **Hafızada tüm çiftlerin adetlerini tutmak için yeterli alan olsa bile, tüm sepetlerdeki çiftlerin adetlerinin sayılması uzun zaman alır.**
- Eğer çok sayıda çift varsa, basit bir sayma işlemi bile hafızada yapılamaz.
- **A-Priori algoritması sayılacak çiftlerin adedini azaltır.**
- A-Priori algoritmasının literatürde farklı versiyonları geliştirilmiştir (önişlemler ile aday frequent itemset oluşturmada farklılıklar vardır).
- A-Priori algoritması verinin üzerinden her iterasyonda iki kez geçer.

18

A-Priori Algoritması

- **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

A-Priori Algoritması

- 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'dır.

$\{w[1], w[2], \dots, w[k]\}$

- 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

A-Priori Algoritması

İlk geçiş

- **A-Priori algoritması verinin üzerinden ilk geçişte**, önce item isimlerini (gerekliyse) tamsayıya dönüştürür ($1 \rightarrow n$) ve bir tabloda tutulur.
- Ardından, **item adetlerinin tutulması için başka bir tablo oluşturulur** ve başlangıçta item'ların adetleri 0 yapılır.
- **Tüm sepetlerden ilk geçişte karşılaşılan item'ın adedi bir artırılır.**
- Tüm sepetlerden geçtikten sonra **support değerini sağlayan tek item'lar frequent item olarak alınır.**
- Tek item'dan oluşan **frequent item sayısı genellikle %1 civarındadır.**
- Ardından, tüm ikili eşleştirmelerden **aday 2-frequent itemset'ler belirlenir.**

21

A-Priori Algoritması

İkinci geçiş

- **Aday 2-frequent itemset'lerin tüm sepetlerdeki adetleri sayılır.**
- **Bir çift item'ın her ikisi de ayrı ayrı frequent item değilse kendisi de frequent item olamaz.**
- İlk geçişte frequent item olmayan tek item'ların içinde olduğu tüm çiftler elimine edilmiş olur.
- **Tüm ikili aday itemset'lerin adetlerini tutmak için gereken yer yaklaşık $2n^2$ olur.**
- 1-frequent itemset sayısı m ise $2m^2$ alan gerekir.
- $m = n/2$ ise, adet tutmak için toplam gerekli alan $2(n/2)^2 = n^2/2$ olur.
- Tüm item'ların yarısı elimine edilse bile, çiftlerin adetlerinin saklanması için gerekli alan $1/4$ 'e iner.

22

A-Priori Algoritması

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     $C_k \leftarrow \text{candidate-gen}(F_{k-1});$  // scan the data once
5    for each transaction  $t \in T$  do
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.

23

A-Priori Algoritması

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
6       $c \leftarrow \{i_1, \dots, i_{k-1}, i'_{k-1}\};$ 
7       $C_k \leftarrow C_k \cup \{c\};$ 
8    for each  $(k-1)$ -subset  $s$  of  $c$  do
9      if  $(s \notin F_{k-1})$  then
10        delete  $c$  from  $C_k;$ 
11    endfor
12 endfor
13 return  $C_k;$ 

```

Son elemanları farklı

Joining aşaması

Pruning aşaması

// return the generated candidates

24

A-Priori Algoritması

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.

25

A-Priori Algoritması

Örnek

- Aşağıda 7 transaction'a sahip bir T kümesi verilmiştir.
- **minsup = 30%** (en az 3/7) olarak 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

A-Priori Algoritması

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}}

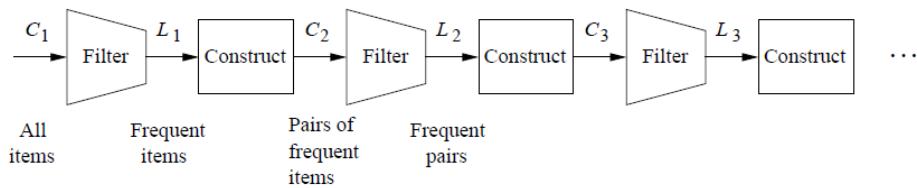
F_3 : {{Chicken, Clothes, Milk}:3}

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

27

A-Priori Algoritması

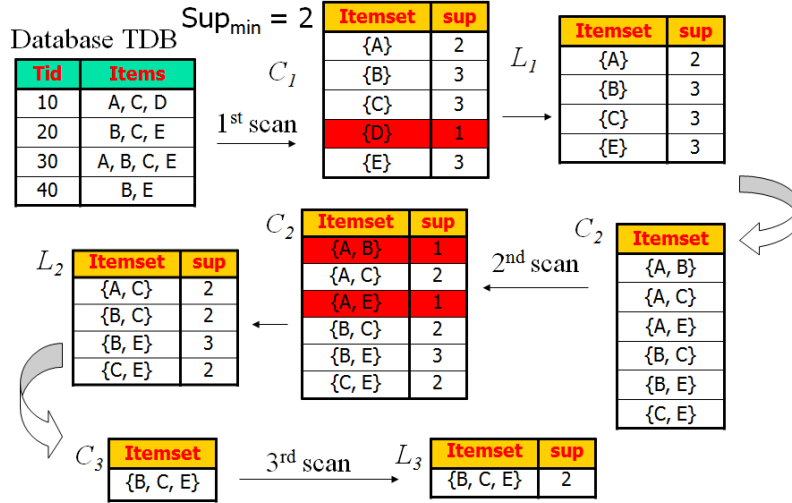
- A-Priori algoritmasında k -frequent itemset yoksa $(k+1)$ -frequent itemset yoktur (monotonicity).
- k -frequent itemset'ten $(k+1)$ -frequent itemset'e geçiş iki adımda yapılır:
 - Aday $(k+1)$ -frequent itemset'ler belirlenir.
 - Aday itemset'lerden frequent olanlar belirlenir.



28

A-Priori Algoritması

Örnek: $\text{minsup}=2$



29

A-Priori Algoritması

Algoritmanın değerlendirilmesi

- 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 (satır 5-10 arasında). **Ölçeklenebilirliği yüksektir** ve çok büyük boyutlardaki veriler üzerinde çalışabilir.
- **Level-wise search** yaptığından istenilen seviyede sonlandırılabilir.
- Birlikte kurallarda çok sayıda (binlerce) kural oluşturulabilir ve faydalı olanların bulunması çok zordur (**Interestingness problem**).

30

Konular

- Frequent Itemsets
 - Market sepeti modeli
 - Market sepeti veri gösterimi
 - Frequent itemsets uygulamaları
- A-Priori Algoritması
- Hafızada Büyük Veri Kümelerinde İşlem
 - PCY algoritması
 - Multistage algoritması
 - Multihash algoritması

31

Hafızada Büyük Veri Kümelerinde İşlem

- 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.

32

Konular

- Frequent Itemsets
 - Market sepeti modeli
 - Market sepeti veri gösterimi
 - Frequent itemsets uygulamaları
- A-Priori Algoritması
- Hafızada Büyük Veri Kümelerinde İşlem
 - PCY algoritması
 - Multistage algoritması
 - Multihash algoritması

33

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**.

34

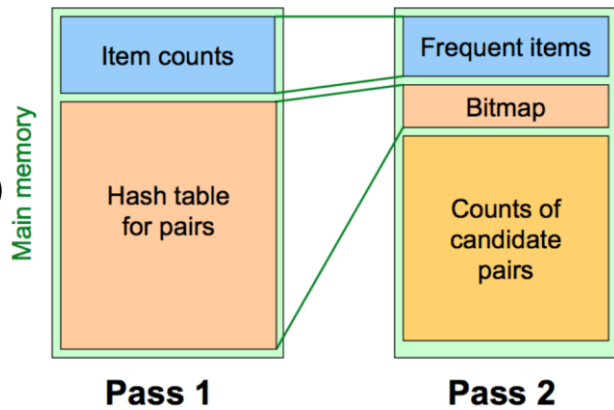
PCY algoritması

- 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.

35

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.



36

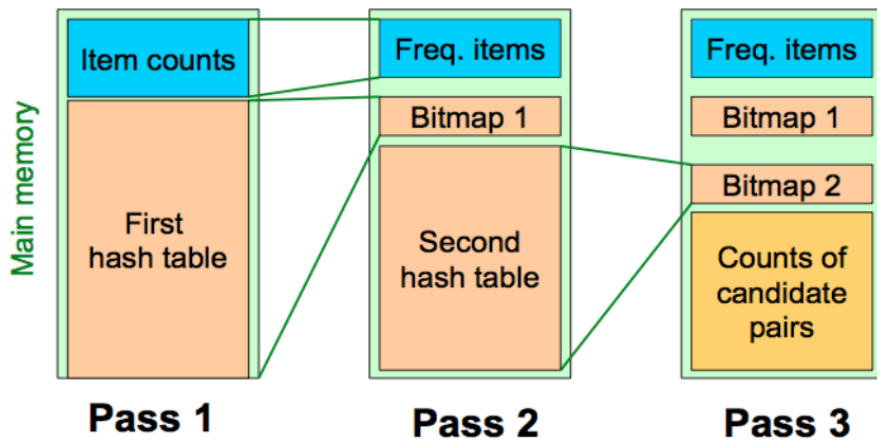
Konular

- Frequent Itemsets
 - Market sepeti modeli
 - Market sepeti veri gösterimi
 - Frequent itemsets uygulamaları
- A-Priori Algoritması
- Hafızada Büyük Veri Kümelerinde İşlem
 - PCY algoritması
 - Multistage algoritması
 - Multihash algoritması

37

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.

38

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.

39

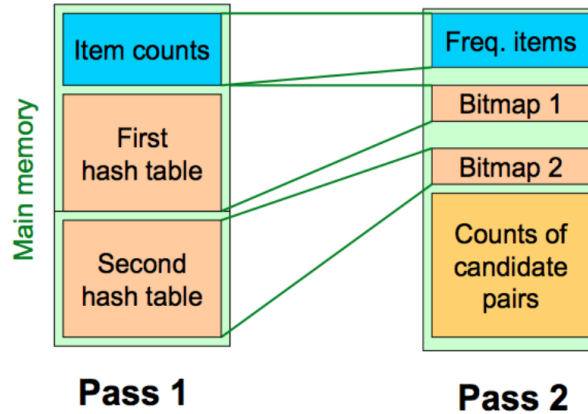
Konular

- Frequent Itemsets
 - Market sepeti modeli
 - Market sepeti veri gösterimi
 - Frequent itemsets uygulamaları
- A-Priori Algoritması
- Hafızada Büyük Veri Kümelerinde İşlem
 - PCY algoritması
 - Multistage algoritması
 - Multihash algoritması

40

Multihash algoritması

- Multihash algoritması, ard 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.**