Büyük Veri İçin İstatistiksel Öğrenme (Statistical Learning for Big Data)

M. Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Bu dersin sunumları, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2017." ve "Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman, Stanford University, 2011." kitapları kullanılarak hazırlanmıştır.

Konular

- Frequent Itemsets
  - Market sepeti modeli
  - Frequent itemsets uygulamaları
  - Birliktelik kuralları
  - Market sepeti veri gösterimi
  - 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 üretirler.**

3

Konular

- Frequent Itemsets
  - Market sepeti modeli
  - Frequent itemsets uygulamaları
  - Birliktelik kuralları
  - Market sepeti veri gösterimi
  - 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

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}

- Minimum support değeri $s = 3$ olsun.
- Tek elemanlı 5 frequent itemset vardır.

  {dog}, {cat}, {and}, {a}, {training}
- **İki elemanlı frequent itemset'lerde her iki eleman da 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

7

Market sepeti modeli

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}

- **Üç elemanlı frequent itemset'lerde her iki elemanlı alt küme frequent itemset olmak zorundadır.**
- **{dog, a, and}** üçlüsü, **{a, and}** 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.**

8

Konular

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

9

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 aranırsa** stopword'ler (gibi, ile, a, the, …) elde edilir.
- **Tüm dokümanlarda yaygın kullanılmayan kelime çiftleri dokümanın konusunu daha çok yansıtır.**

10

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'ler belirlenebilir.
- Benzer hastalığa neden olan biomarker'lar belirlenebilir.

11

Konular

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

12

Birliktelik kuralları

- **Frequent itemset'lerin genellikle if-then kuralları ile gösterilmesi istenir.**
- Bu kurallara **birliktelik kuralları** (association rules) denilmektedir.
- I item kümesi ve j bir item olmak üzere aşağıdaki gibi gösterilir:

$$I \rightarrow j$$

- ***I* item kümesindeki tüm item'lar bir sepette varsa, *j* 'de bu sepette vardır.**
- ***I* item kümesindeki tüm item'lar ile *j* item'ının birlikte bulunma oranı destek** (support) **değeridir.**
- ***I* item kümesindeki tüm item'ların bulunduğu sepetlerde *j* item'ının da bulunma oranı güven** (confidence) **değeridir.**

13

Birliktelik kuralları

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}

- {cat, dog} → and, kuralının **support** değeri **3/8**, **confidence** değeri **3/5** tir.
- {cat} → kitten, kuralının **support** değeri **1/8**, **confidence** değeri **1/6** dır.

14

Birliktelik kuralları

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}

- ***I → j* birliktelik kuralı için** interest değeri, kuralın ***confidence*** değeri **ile *j'* nin *support* değerinin farkıyla hesaplanır.**
- ***Interest* = (*I → j*).*conf* – *j.sup***
- **Sıfıra yakın değerler** $I \rightarrow j$ kuralının önemli olmadığını gösterir (rastgele dağılımda sıfır olur).
- **Yüksek negatif değerler** I' nın j ile birlikte olmadığını (önemli) gösterir.
- **Yüksek pozitif değerler** $I \rightarrow j$ kuralının önemli olduğunu gösterir.
- $\{dog\} \rightarrow cat$, kuralının interest değeri = $5/7 - 6/8 = -0{,}036$ dır.
- $\{dog\} \rightarrow cat$, birliktelik kuralı önemli değildir.
- $\{cat\} \rightarrow kitten$, kuralının interest değeri = $1/6 - 1/8 = 0{,}042$ dir.

15

Konular

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

16

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}{...
```

- 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).

17

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.
- 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 $n \leq 33.000$ adet çift) olmalıdır.

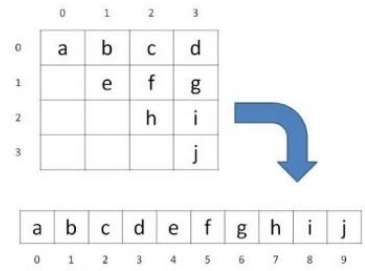
18

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).

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



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

19

Market sepeti veri gösterimi

Monotonicity

- **Eğer I item kümesi frequent ise, I kümesinin tüm altkümeleri de frequent'tır** (monotonicity).
- **$J \subseteq I$ ise, I daki item'ları bulunduran sepetler, J deki 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.

20

Konular

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

21

A-Priori algoritması

- **Hafızada tüm çiftlerin adetlerini tutmak için yeterli alan olsa bile, tüm sepetlerdeki çiftlerin adetlerinin sayılması uzun zaman alır.**
- Eğer çok fazla sayıda çift varsa, basit bir sayma işlemi bile hafızada yapılamaz. **A-Priori algoritması sayılacak çiftlerin adedini azaltır.**
- **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 sahiptir.
- **Pruning:** Apriori algoritması minsup değerine sahip olmayan birliktelik kural adaylarını temizler.

22

A-Priori algoritması

- Algoritma, I itemset içerisindeki elemanların tümüyle sıralı **(lexicographic order)** olduğunu varsayar.
- Bir k-itemset aşağıdaki gibi gösterilir ve $w[1]$, $w[2]$, ..., $w[k]$ birer item'dır.

  $\{w[1], w[2], \ldots, w[k]\}$
- 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 değerini ve frequent olup olmadığını belirler.
- Önce her bir item için frequent 1-itemset'i oluşturur ve her iterasyonda 2-itemset, 3-itemset şeklinde artarak devam eder.

23

A-Priori algoritması

- **A-Priori algoritması, tüm sepetlerden ilk geçişte karşılaşılan item'ın adedini bir artırır.**
- 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, 1-frequent item'ların tüm ikili eşleştirmelerinden **aday 2-frequent itemset'ler belirlenir.**
- İlk geçişte frequent item olmayan tek item'ların içinde olduğu tüm çiftler elimine edilmiş olur.

24

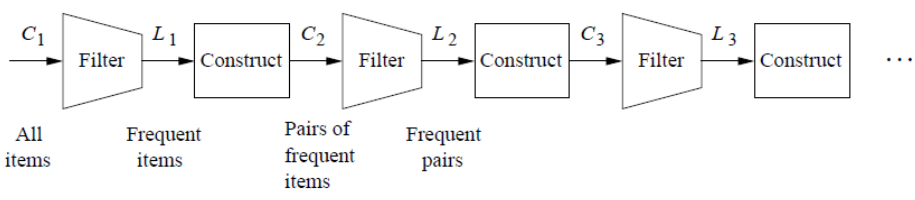
A-Priori algoritması

- Aday 2-frequent itemset'lerin sepetlerdeki adetleri sayılır.
- Sayma için ikili kombinasyon kadar **(n! / (n-2)! 2!)** yer gerekir.
- **Tüm ikili aday itemset'lerin adetlerini tutmak için gereken yer yaklaşık $2n^2$ byte olur** (Her sayaç 4 byte).
- Aday 2-frequent itemset'ler için sayaç $m = n/2$ olsa, toplam sayaç için gerekli alan $2(n/2)^2 = n^2/2$ olur.
- Sayılacak çiftlerin yarısı elimine edilse bile, adetlerin tutulması için gerekli alan $1/4$'e iner.

25

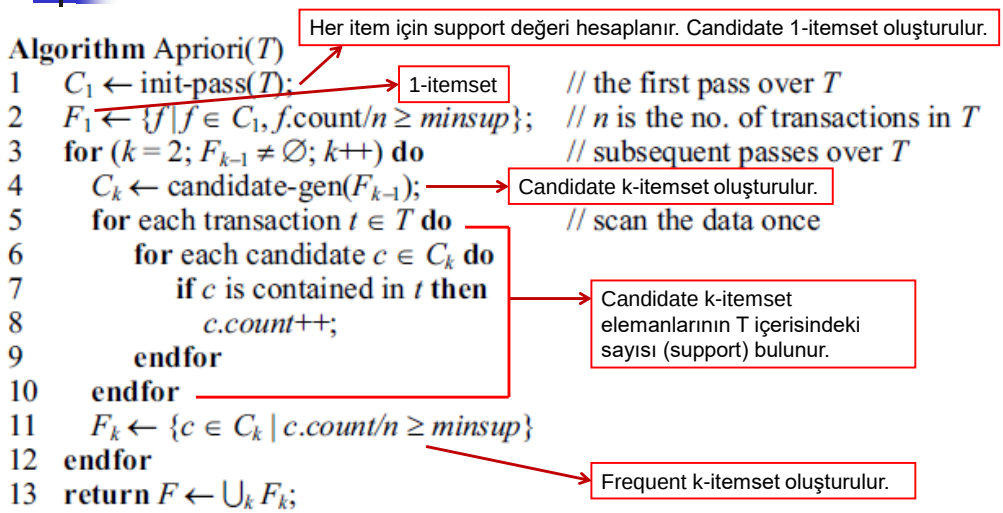
A-Priori algoritması

- **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.



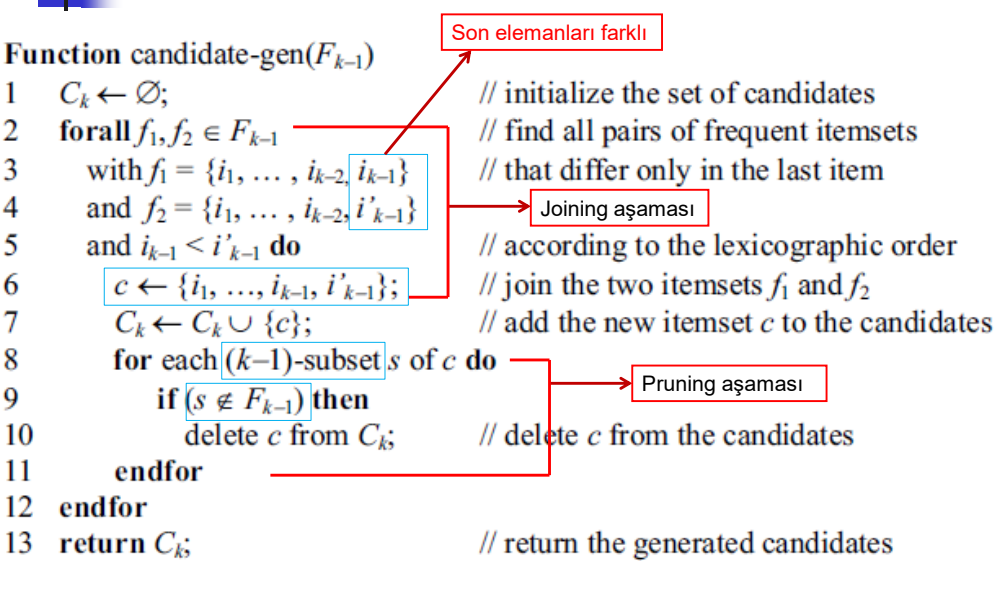
26

A-Priori algoritması




```
Algorithm Apriori(T)
1   C1 ← init-pass(T);                    // the first pass over T
2   F1 ← {f | f ∈ C1, f.count/n ≥ minsup}; // n is the no. of transactions in T
3   for (k = 2; Fk−1 ≠ ∅; k++) do         // subsequent passes over T
4     Ck ← candidate-gen(Fk−1);
5     for each transaction t ∈ T do       // scan the data once
6       for each candidate c ∈ Ck do
7         if c is contained in t then
8           c.count++;
9       endfor
10    endfor
11    Fk ← {c ∈ Ck | c.count/n ≥ minsup}
12  endfor
13  return F ← ⋃k Fk;
```


A-Priori algoritması




```
Function candidate-gen(Fk−1)
1   Ck ← ∅;                                  // initialize the set of candidates
2   forall f1, f2 ∈ Fk−1                     // find all pairs of frequent itemsets
3     with f1 = {i1, … , ik−2, ik−1}         // that differ only in the last item
4     and f2 = {i1, … , ik−2, i'k−1}
5     and ik−1 < i'k−1 do                    // according to the lexicographic order
6       c ← {i1, …, ik−1, i'k−1};            // join the two itemsets f1 and f2
7       Ck ← Ck ∪ {c};                       // add the new itemset c to the candidates
8       for each (k−1)-subset s of c do
9         if (s ∉ Fk−1) then
10          delete c from Ck;                // delete c from the candidates
11      endfor
12  endfor
13  return Ck;                               // return the generated candidates
```

A-Priori algoritması

Örnek

- Aşağıda 3. seviyede oluşturulan frequent 3-itemset verilmiştir.

  $F_3 = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 3, 4\}\}$

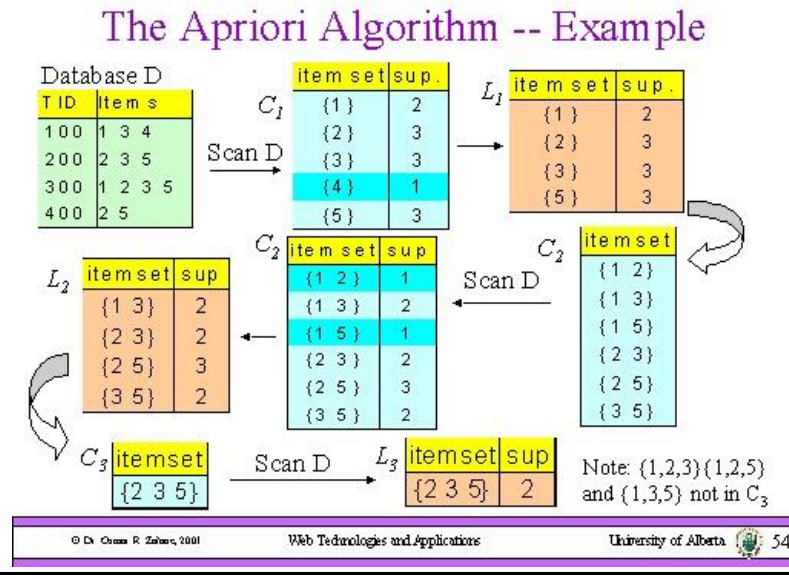
- Join aşamasında {1, 2, 3, 4} ve {1, 3, 4, 5} oluşturulur.
- {1, 2, 3, 4} birinci ve ikinci itemset'leri ile oluşturulur.
- {1, 3, 4, 5} ise {1, 3, 4} ile {1, 3, 5} itemset'leri ile oluşturulur.
- Pruning aşamasından sonra sadece {1, 2, 3, 4} kalır.
- **{1, 4, 5} kümesi ve {3, 4, 5} kümesi, frequent 3-itemset içerisinde olmadığından {1, 3, 4, 5} silinir.**

29

A-Priori algoritması

Örnek

minsup=2



30

Konular

- Frequent Itemsets
  - Market sepeti modeli
  - Frequent itemsets uygulamaları
  - Birliktelik kuralları
  - Market sepeti veri gösterimi
  - 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 okunur ve yazılır (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'lerde** olur.

32

Konular

- Frequent Itemsets
  - Market sepeti modeli
  - Frequent itemsets uygulamaları
  - Birliktelik kuralları
  - Market sepeti veri gösterimi
  - 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

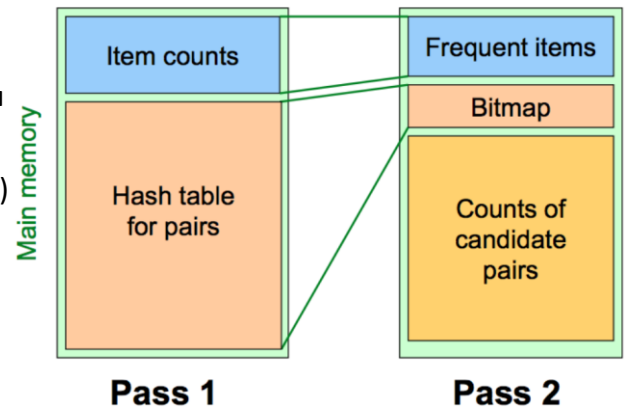
PCY algoritması

- **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

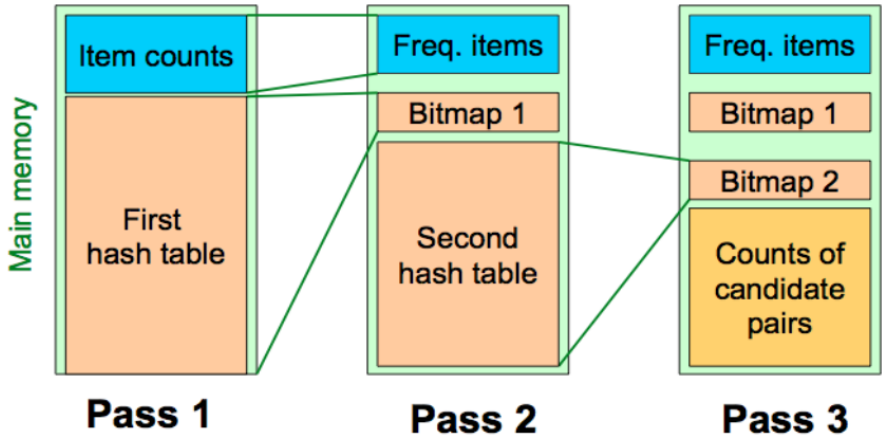
Konular

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

37

Multistage algoritması

- Multistage algoritması, **ard 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.**
- **Birinci geçişte frequent bucket'a atananlar** ile her bir item'ı frequent olanlar **ikinci geçişte tekrar hash'lenir.**
- $\{i, j\}$ 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

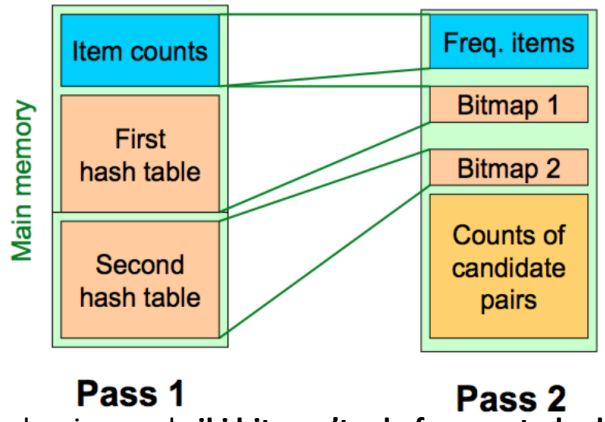
Konular

- Frequent Itemsets
  - Market sepeti modeli
  - Frequent itemsets uygulamaları
  - Birliktelik kuralları
  - Market sepeti veri gösterimi
  - 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 algorimasında **iki bitmap'te de frequent olanlar aday çift olarak alınır.**

41