Ιστογράμματα

Γιάννης Κωτίδης

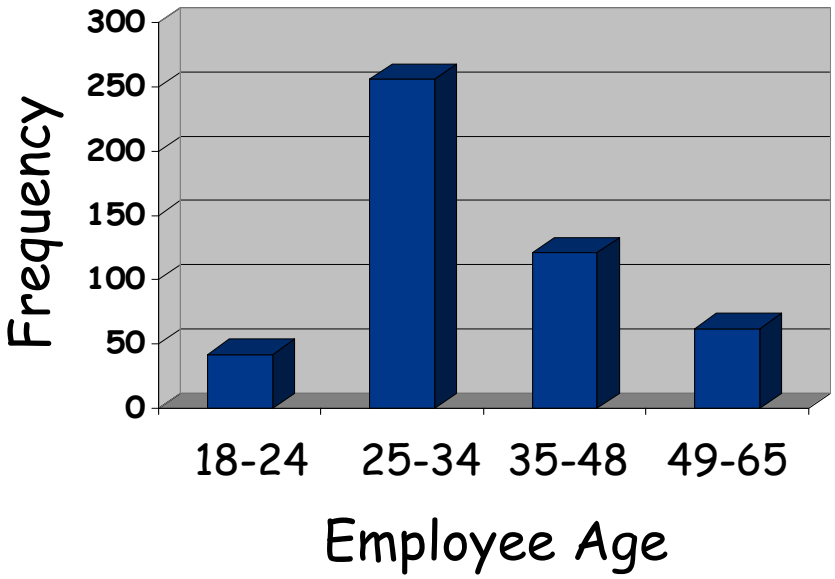
Ιστογράμματα

- Είδαμε πως υπολογίζουμε τον αριθμό των αποτελεσμάτων σε μία επιλογή ή σύζευξη μέσω απλών στατιστικών
- Πχ υπόθεση ομοιόμορφης κατανομής για την επιλογή:
  - $T(\sigma_{Z=\zeta}(W)) = T(W)/V(W,Z)$
- Στη πράξη τα συστήματα διατηρούν ποιο σύνθετα στατιστικά ώστε να επιτυγχάνουν μεγαλύτερη ακρίβεια στις εκτιμήσεις που παίρνουν
  - Παράδειγμα: ιστόγραμμα

Γενική ιδέα

- Partition attribute value(s) domain into a set of buckets (assume B buckets)
- Describe (compactly) data within a bucket
- Issues:
  - How to partition?
  - What to store for each bucket?
  - How to estimate an answer using the histogram?

Παράδειγμα



Bucket	Count
[18..24]	49
[25..34]	250
[35..48]	108
[49..65]	55

Q: select * from Employee
where Age = 30

T(Q)=?

Γιατί είναι χρήσιμο;

Q: select * from Employee
where Age = 30

Εκτίμηση T(Q)=25

Bucket	Count
[18..24]	49
[25..34]	250
[35..48]	108
[49..65]	55

Έστω μη clustering index στο Employee.Age
Αν το ευρετήριο βρίσκεται στη μνήμη και το κόστος του random I/O είναι 10msec, πόσο χρόνο θα πάρει η εκτέλεση του ερωτήματος με τη χρήση του ευρετηρίου;

Απάντηση: ?

Επίλυση

Q: select * from Employee T(Q)=?
where Age = 30

Bucket	Count
[18..24]	49
[25..34]	250
[35..48]	108
[49..65]	55

Χωρίς το ιστόγραμμα

T(R) = 462

V(R,age) = 34 (το γνωρίζω από τα απλά στατιστικά)

T(Q) = $\frac{462}{34}$ = 13.6

Επίλυση

Q: select * from Employee T(Q)=?
where Age = 30

Bucket	Count
[18..24]	49
[25..34]	250
[35..48]	108
[49..65]	55

Χωρίς το ιστόγραμμα

T(R) = 462

V(R,age) = 34

$T(Q) = \frac{462}{34} = 13.6$

Χρησιμοποιώντας το ιστόγραμμα

Η τιμή 30 ανήκει στο bucket [25..34]

Το bucket έχει 250 εγγραφές και εύρος 10 τιμές

$T(Q) = \frac{250}{10} = 25$

Υποθέτω ότι οι τιμές κατανέμονται **ομοιόμορφα** μέσα σε ένα bucket

Παρατήρηση

Q: select * from Employee
where Age = 30

Αν επιπλέον του αριθμού των εγγραφών (Count) σε κάθε bucket κρατούσα και τον αριθμό (Values) των διαφορετικών/distinct τιμών του age μέσα στο bucket

T(Q) = ?

Bucket	Count	Values
[18..24]	49	10
[25..34]	250	5
[35..48]	108	11
[49..65]	55	8

Απάντηση: Εκτίμηση T(Q) = $\frac{250}{5}$ = 50

(με την γνωστή υπόθεση ότι ψάχνω μία από τις τιμές που υπάρχουν και η κατανομή των εγγραφών **ως προς τις υπάρχουσες τιμές** είναι ομοιόμορφη)

Παρατήρηση

- Κρατώντας περισσότερα στατιστικά μέσα σε ένα bucket (όπως #distinctValues, most frequent value, κα)
  - Περιμένω καλύτερη ακρίβεια στις εκτιμήσεις που κάνω
  - Όμως ο χώρος που απαιτείται για την αποθήκευση του ιστογράμματος μεγαλώνει

Bucket	Count	Vd
[18..24]	49	10
[25..34]	250	5
[35..48]	108	11
[49..65]	55	8

ή

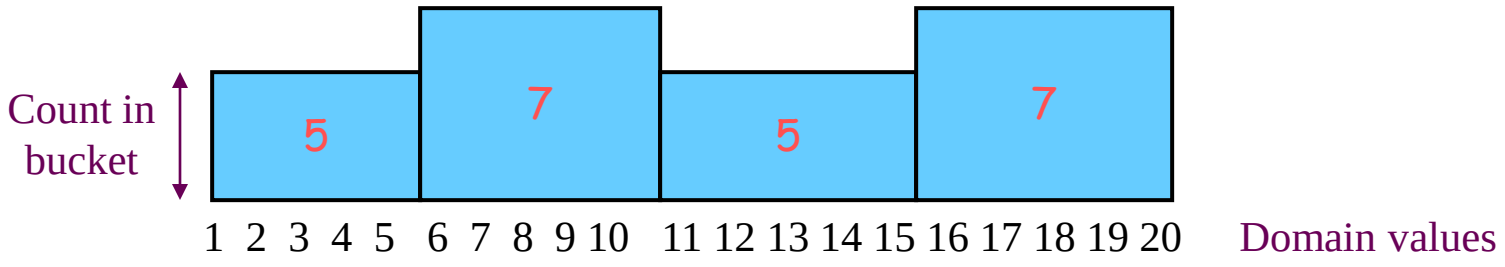
Bucket	Count
[18..24]	49
[25..34]	250
[35..48]	108
[49..65]	55

1-D Histograms: Equi-Width

- Goal: Split domain evenly
- Assume B=4 (Domain 1..20)

Data: 1 2 2 3 4 | 7 8 9 10 10 10 10 | 11 11 12 12 14 | 16 16 18 19 20 20 20

- Buckets?
  - [1..5] , [6..10], [11..15], [16..20]

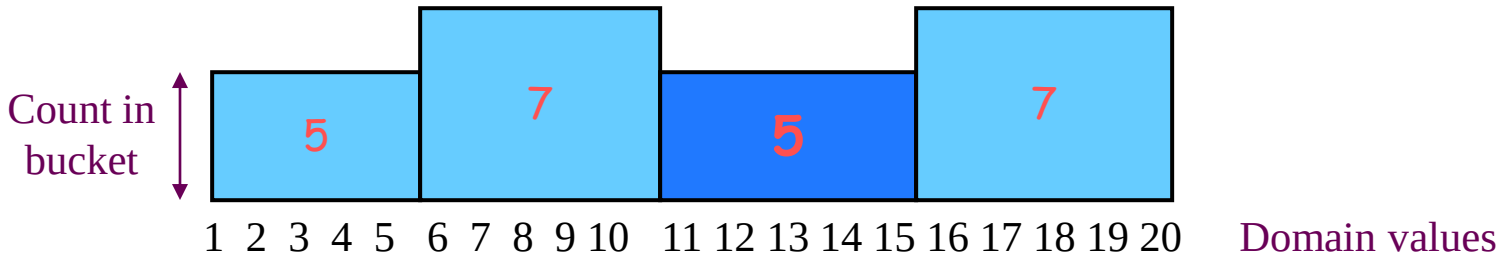


How many tuples with value=12?

Εκτίμηση (Point Queries)

- Έστω ότι η τιμή που αναζητάμε περιέχεται στο bucket b
  - Το b καλύπτει ένα διάστημα από το πεδίο τιμών του γνωρίσματος με πλάτος V_b
  - Η τιμή του counter για το bucket b είναι C_b

$$\text{Εκτίμηση} = \frac{C_b}{V_b}$$



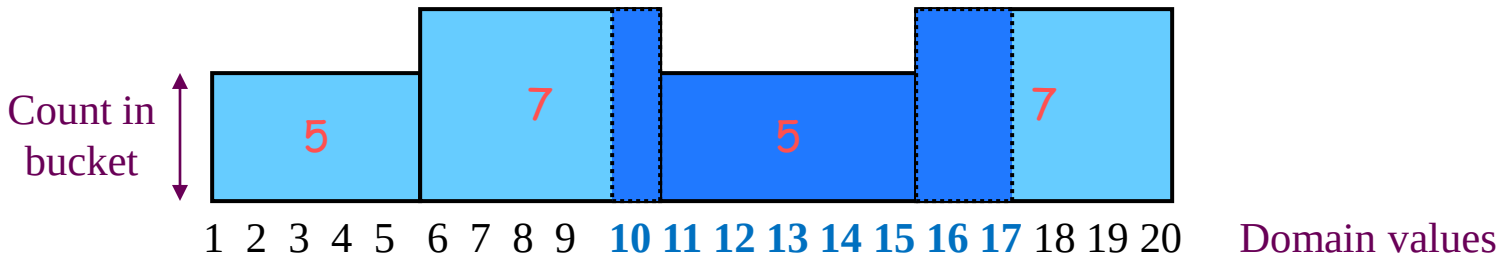
Πόσες εγγραφές με τιμή 12?

$$\text{Εκτίμηση} = \frac{5}{15-11+1} = 1$$

Εκτίμηση διαστήματος τιμών (Range Queries)

- Αν το διάστημα καλύπτει όλο το bucket, προσθέτουμε στην εκτίμηση το C_b
- Αν καλύπτει μέρος του bucket, προσθέτουμε την εκτίμηση για κάθε μία τιμή που περιέχεται στο διάστημα που αναζητούμε
- Εκτίμηση για το εύρος [10..17]?
- Απάντηση:
  - Από το bucket [6..10] → $+1*\frac{7}{5}$
  - Από το bucket [11..15] → +5
  - Από το bucket [16..20] → $+2*\frac{7}{5}$

→ Άρα η εκτίμηση για τον αριθμό των εγγραφών με τιμές [10..17] είναι $\frac{7}{5} + 5 + \frac{14}{5} = 9.2$



Problems with Equi-Width

- Data is often skewed
  - 1,1,1,1,1,2,2,3,5,20

- Assume B=4, domain=[1..20]
  - Buckets: [1..5] , [6..10], [11..15], [16..20]

- Histogram:

[1..5] : 9
[6..10] : 0
[11..15] : 0
[16..20] : 1

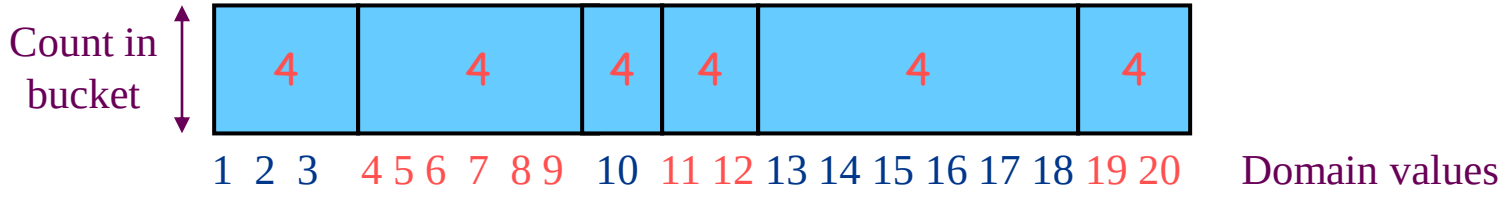
- Estimate for value=1? (most frequent)

(Often) a better Solution: Equi-Depth

- Goal: Equal number of rows per bucket
- Can **construct** by first sorting then taking B-1 equally-spaced splits

1 2 2 3 4 7 8 9 10 10 10 10 11 11 12 12 14 16 16 18 19 20 20 20

Example: B=6 (seek to create buckets with $\frac{24}{6} = 4$ records each)



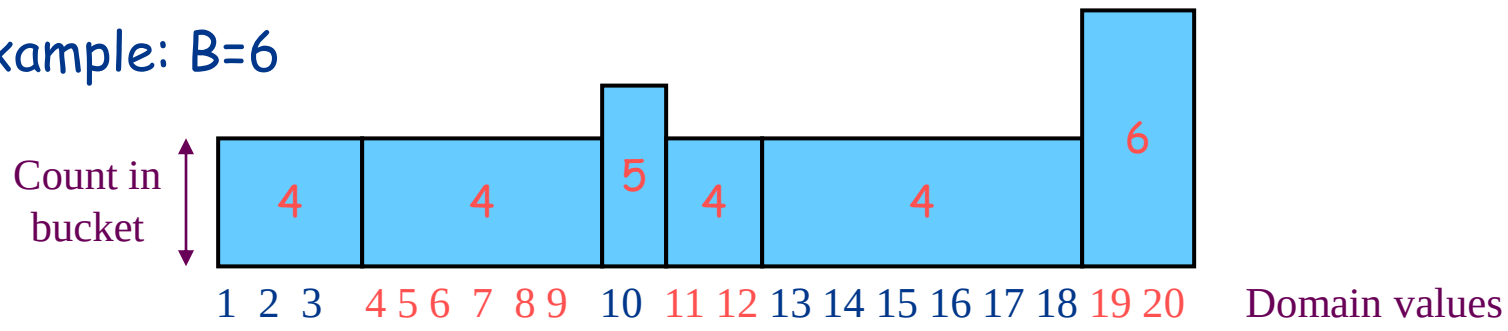
How many with value=12?

Equi-Depth

- Note: it is not always possible to split evenly

1 2 2 3 4 7 8 9 10 10 10 10 10 11 11 12 12 14 16 16 18 19 20 20 20 20 20

Example: B=6



Equi-Depth Histograms: Maintenance

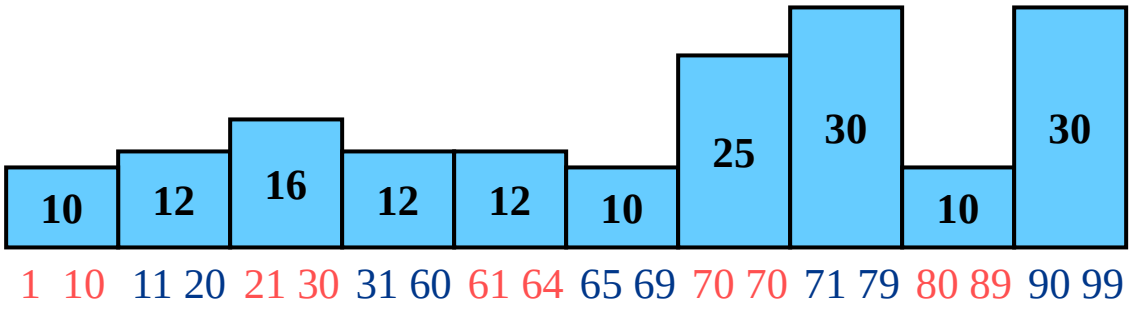
- **Choice 1: Re-compute periodically**
  - "optimize statistics"
- Choice 2: Use auxiliary data structures (makes more sense when data is remote or streaming)
  - Use a backing sample: Maintain a larger sample on disk in support of histogram maintenance
  - Use a sketch
- Choice 3: Learn as you go

Άλλοι τύποι ιστογραμμάτων

- Πλούσια βιβλιογραφία, πχ:
  - Self-Tuning Histograms (Microsoft Auto Admin Project)
  - Compressed Histograms (IBM DB2)
  - V-Opt Histograms (Y. Ioannidis et al)
  - Near-Optimal Histograms on Streaming Data (one-pass) (Gilbert et al)
- Πολυδιάστατα ιστογράμματα

Self-Tuning 1-D Histograms (Microsoft Research)

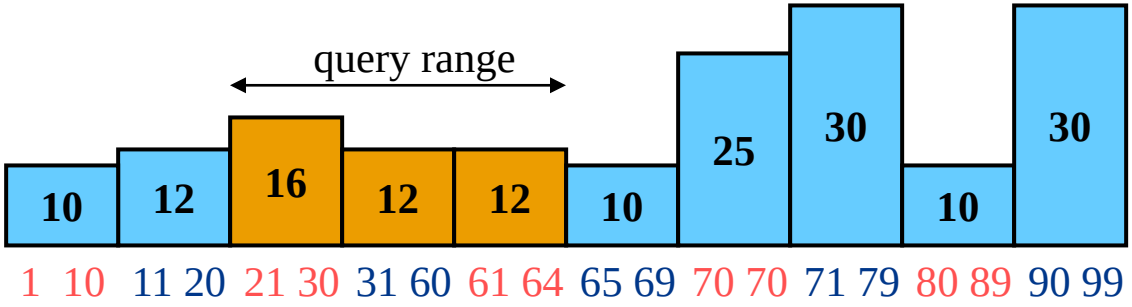
- Start with any histogram (equi-width, equi-dept, etc)



- Use results of user-queries to adjust the histogram
  - Update counters due to insert/delete/updates
  - Periodically adjust buckets (merge/slit buckets)

Self-Tuning 1-D Histograms

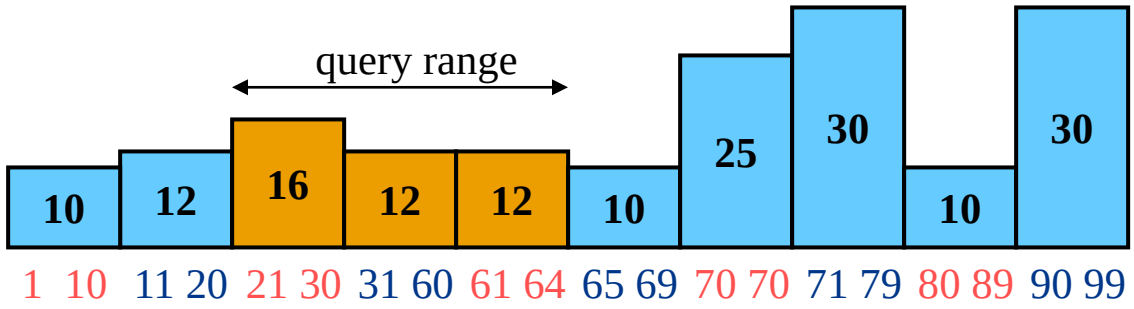
1. Update Bucket Frequencies:
   - Compare actual selectivity to histogram estimate
   - Use to adjust bucket frequencies



   - Assume query for range [21..64]
   - Based on this histogram we expect 16+12+12 = 40 records
   - Upon running the query, we find 60 records in the result
   - How to update the histogram ?

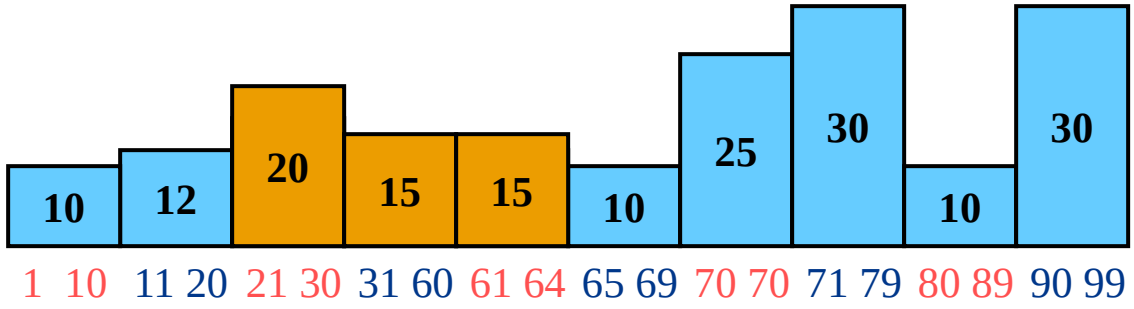
Self-Tuning 1-D Histograms

1. Tune Bucket Frequencies:
   - Compare actual selectivity to histogram estimate
   - Use to adjust bucket frequencies



Actual = 60
Estimate = 40
Error = +20

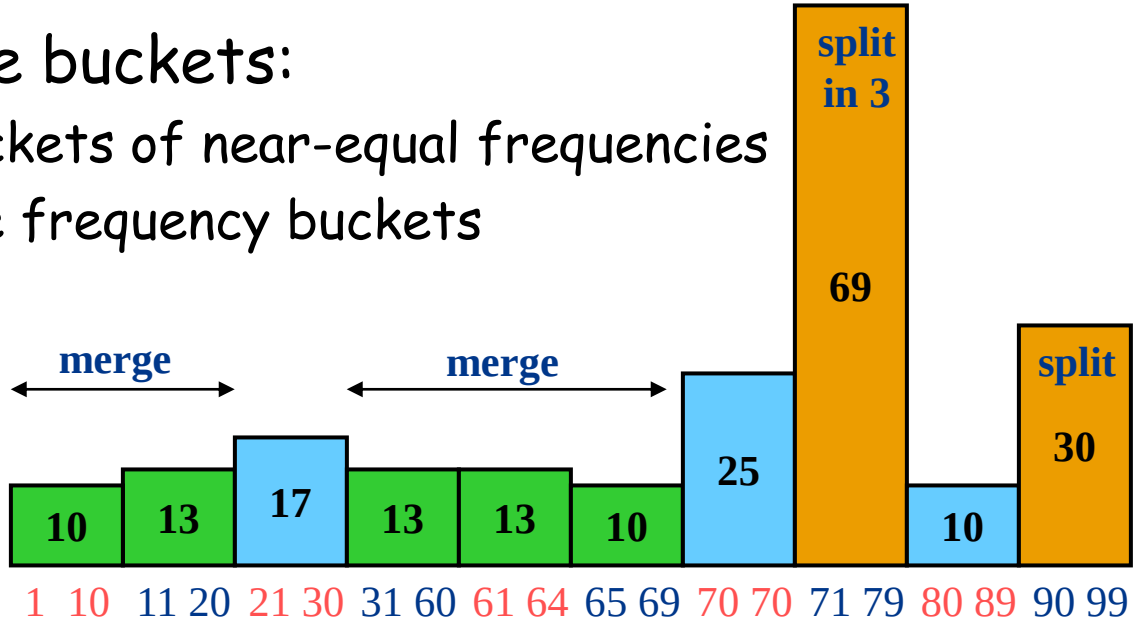
- Divide d*Error proportionately, d=dampening factor (≤1)



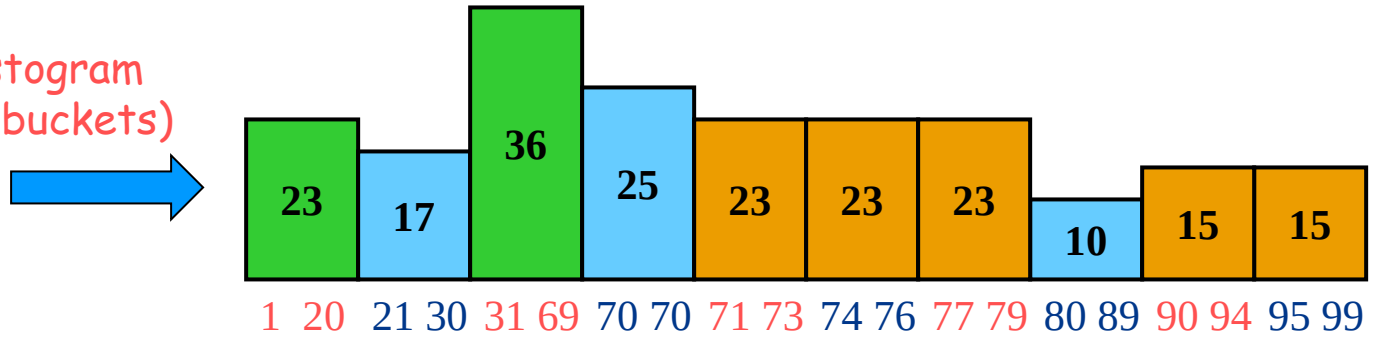
d=½ of Error
= +10
So divide
+4,+3,+3

Self-Tuning 1-D Histograms

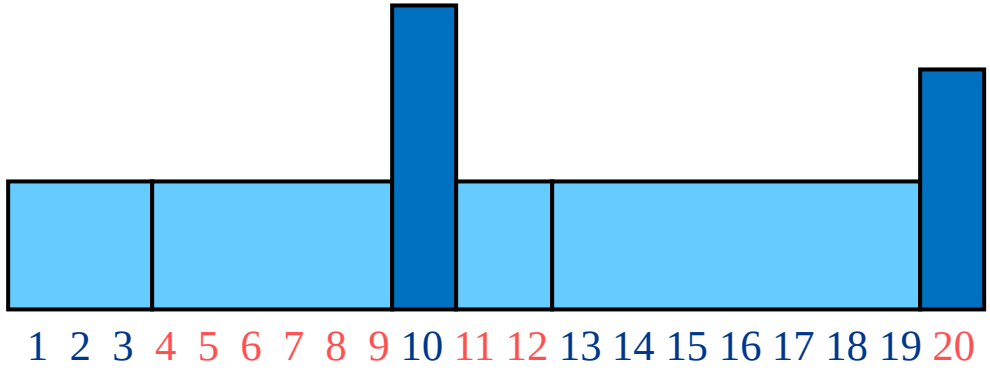
2. Restructure buckets:
   - Merge buckets of near-equal frequencies
   - Split large frequency buckets



Updated Histogram
(still using 10 buckets)



1-D Histograms: Compressed



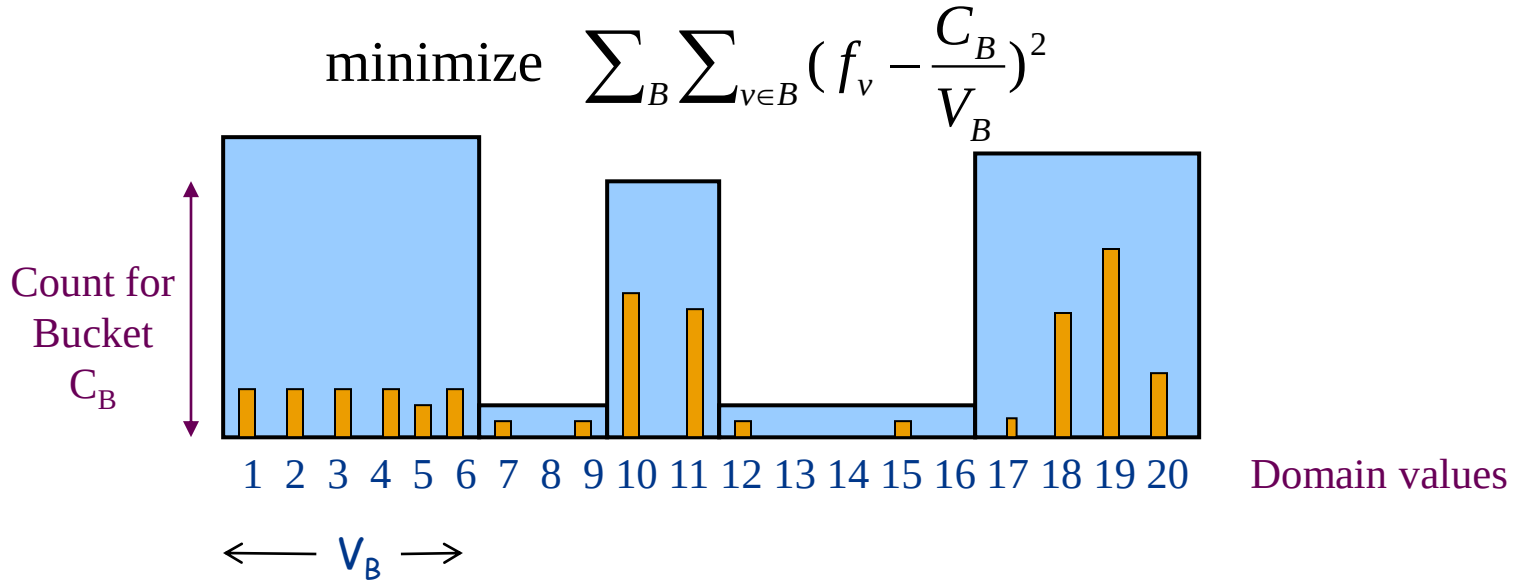
[PIH96]

- Create singleton buckets for largest values, equi-depth over the rest

- Improvement over equi-depth since get exact info on largest values, e.g., join estimation in DB2 compares largest values in the relations

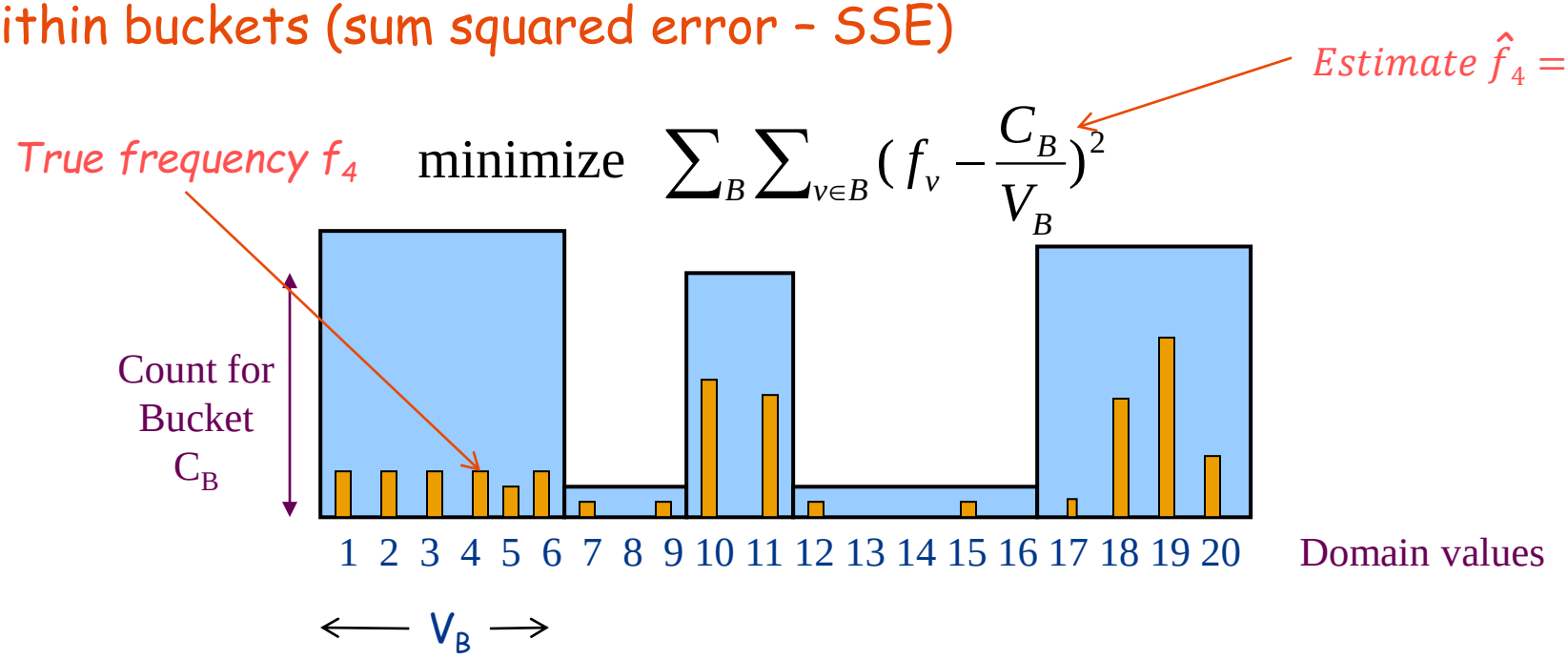
V-Optimal Histograms

- Idea: Select buckets to minimize frequency variance of approximation within buckets (sum squared error – SSE)



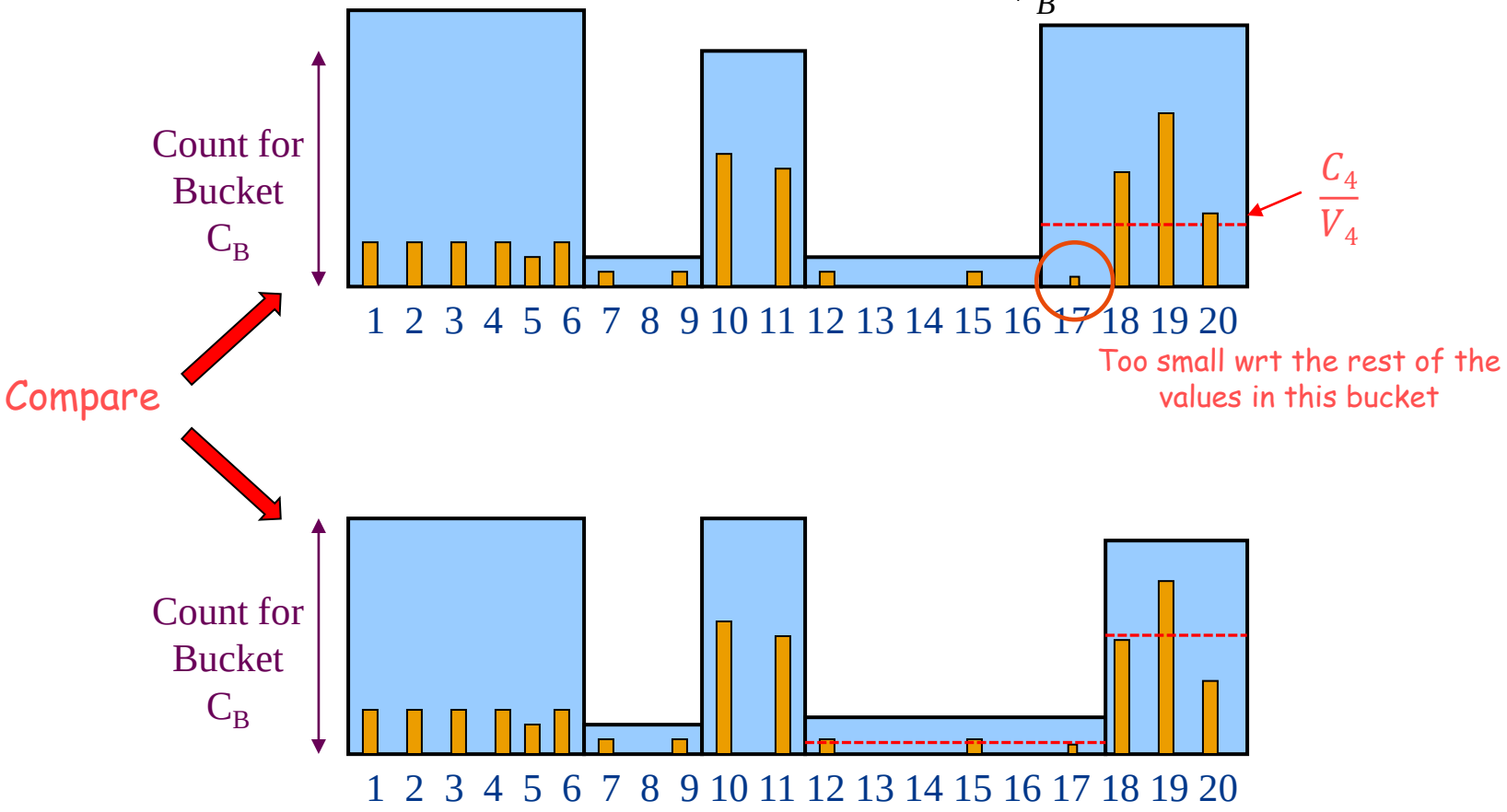
V-Optimal Histograms

- Idea: Select buckets to minimize frequency variance of approximation within buckets (sum squared error – SSE)



V-Optimal Histograms

minimize $\sum_B \sum_{v \in B} (f_v - \frac{C_B}{V_B})^2$

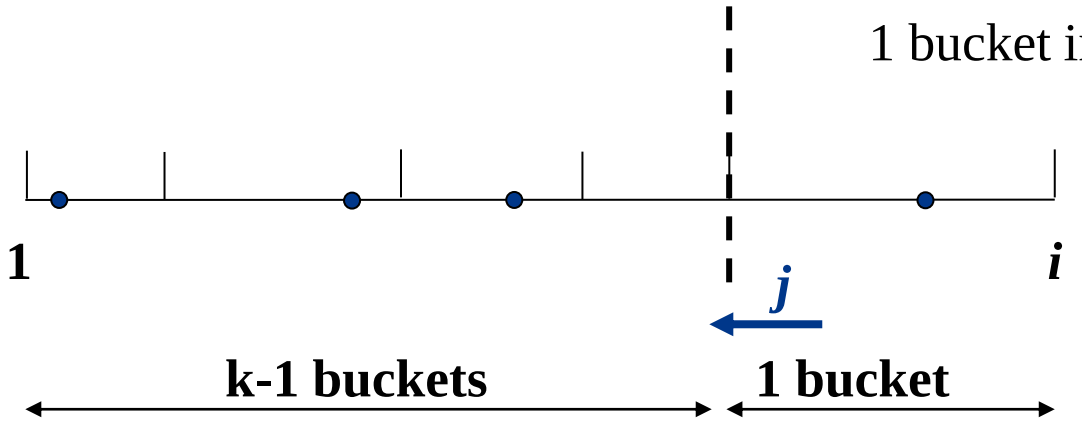


V-OPT Construction

$$SSEP(i,k) = \min_{j=1..i-1} (SSEP(j,k-1) + sse[j+1:i])$$

SSEP(i,k)=sse of using k buckets for range 1:j

sse when using
1 bucket in range j+1:i



Answering Queries (example 1)

- Assume histogram on age
  - [20..29]:25, [30..44]:8, [45..45]:5, [46..59]:11

- How many people with age = 28?
  - 25/(29-20+1)=2.5
  - Min possible = 0
  - Max possible = 25
- How many people with age in [30..44]
  - Answer is 8, error = ?
- How many people with age in [25..45]
  - Answer is 25/(10/5)+8+5 =12.5+8+5=25.5
  - Min possible = 13
  - Max possible = 38

Example 2

- Histogram on age:

  [18..21]:800, [22..24]:1200, [25..30]:6000, [31..65]:2000

- SQL Query:

```
SELECT name, age, city, street, number
FROM EMPLOYEE, ADDRESS
WHERE EMPLOYEE.empid = ADDRESS.empid
AND EMPLOYEE.age >= 30
```

- Estimate $T(\sigma_{age \geq 30}(EMPLOYEE))$?

Solution

- Histogram on age:

  [18..21]:800, [22..24]:1200, [25..30]:6000, [31..65]:2000

- Estimate $T(\sigma_{age \geq 30}(EMPLOYEE))$

- From Bucket [25..30] → + $1 * \frac{6000}{6}$ = 1000
- From Bucket [31..65] → + 2000
- $T(\sigma_{age \geq 30}(EMPLOYEE))$ = 1000+2000=3000 records