



15-826: Multimedia Databases and Data Mining

Lecture #4: Multi-key and
Spatial Access Methods - I

C. Faloutsos



Must-Read Material

- Textbook, Chapter 4
- [Bentley75] J.L. Bentley: *Multidimensional Binary Search Trees Used for Associative Searching*, CACM, 18,9, Sept. 1975.
- Ramakrishnan+Gehrke, Chapter 28.1-3

15-826

Copyright: C. Faloutsos (2011)

2



Outline

Goal: 'Find similar / interesting things'

- Intro to DB
- • Indexing - similarity search
- Data Mining

15-826

Copyright: C. Faloutsos (2011)

3



Indexing - Detailed outline

- primary key indexing
- • secondary key / multi-key indexing
- spatial access methods
- text
- ...

15-826

Copyright: C. Faloutsos (2011)

4



Sec. key indexing

- attributes w/ duplicates (eg., EMPLOYEES, with 'job-code')
- Query types:
 - exact match
 - partial match
 - 'job-code'='PGM' and 'dept'='R&D'
 - range queries
 - 'job-code'='ADMIN' and salary < 50K

15-826

Copyright: C. Faloutsos (2011)

5



Sec. key indexing

- Query types - cont'd
 - boolean
 - 'job-code'='ADMIN' or salary>20K
 - nn
 - salary ~ 30K

15-826

Copyright: C. Faloutsos (2011)

6



Solution?

15-826

Copyright: C. Faloutsos (2011)

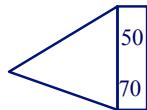
7



Solution?

- Inverted indices (usually, w/ B-trees)
- Q: how to handle duplicates?

salary-index



| Name | Job-code | Salary | Dept |
|--------|----------|--------|-------|
| Smith | PGM | 70 | R&D |
| Jones | ADMIN | 50 | R&D |
| | | | |
| Tomson | ENG | 50 | SALES |

15-826

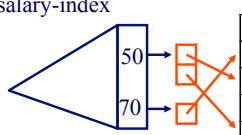
Copyright: C. Faloutsos (2011)

8



Solution

- A#1: eg., with postings lists

salary-index **postings lists**

| Name | Job-code | Salary | Dept |
|--------|----------|--------|-------|
| Smith | PGM | 70 | R&D |
| Jones | ADMIN | 50 | R&D |
| | | | |
| Tomson | ENG | 50 | SALES |

15-826

Copyright: C. Faloutsos (2011)

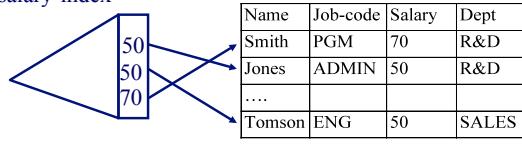
9

 CMU SCS

Solution

- A#2: modify B-tree code, to handle dup's

salary-index



| Name | Job-code | Salary | Dept |
|--------|----------|--------|-------|
| Smith | PGM | 70 | R&D |
| Jones | ADMIN | 50 | R&D |
| | | | |
| Tomson | ENG | 50 | SALES |

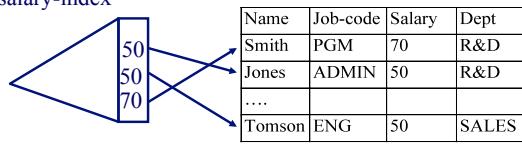
15-826 Copyright: C. Faloutsos (2011) 10

 CMU SCS

How to handle Boolean Queries?

- eg., 'sal=50 AND job-code=PGM'?

salary-index



| Name | Job-code | Salary | Dept |
|--------|----------|--------|-------|
| Smith | PGM | 70 | R&D |
| Jones | ADMIN | 50 | R&D |
| | | | |
| Tomson | ENG | 50 | SALES |

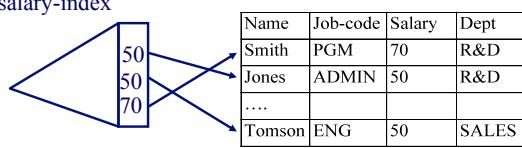
15-826 Copyright: C. Faloutsos (2011) 11

 CMU SCS

How to handle Boolean Queries?

- from indices, find lists of qual. record-ids
- merge lists (or check real records)

salary-index



| Name | Job-code | Salary | Dept |
|--------|----------|--------|-------|
| Smith | PGM | 70 | R&D |
| Jones | ADMIN | 50 | R&D |
| | | | |
| Tomson | ENG | 50 | SALES |

15-826 Copyright: C. Faloutsos (2011) 12



Sec. key indexing

- easily solved in commercial DBMS:

```
create index sal-index on
    EMPLOYEE (salary);
select * from EMPLOYEE
    where salary > 50 and
        job-code = 'ADMIN'
```

15-826

Copyright: C. Faloutsos (2011)

13



Sec. key indexing

- can create combined indices:

```
create index sj on EMPLOYEE
    ( salary, job-code);
```

15-826

Copyright: C. Faloutsos (2011)

14



Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
 - main memory: quad-trees
 - main memory: k-d-trees
- spatial access methods
- text
- ...

15-826

Copyright: C. Faloutsos (2011)

15



Quad-trees

- problem: find cities within 100mi from Pittsburgh
- assumption: all fit in main memory
- Q: how to answer such queries quickly?

15-826

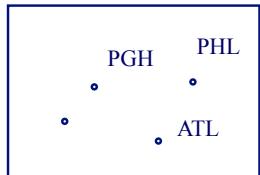
Copyright: C. Faloutsos (2011)

16



Quad-trees

- A: recursive decomposition of space, e.g.:



15-826

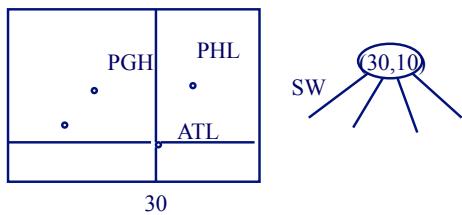
Copyright: C. Faloutsos (2011)

17



Quad-trees

- A: recursive decomposition of space, e.g.:



10

30

15-826

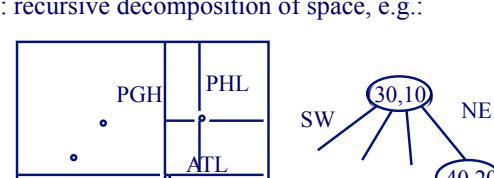
Copyright: C. Faloutsos (2011)

18


CMU SCS

Quad-trees

- A: recursive decomposition of space, e.g.:



The figure illustrates a quad-tree decomposition of a 2D space. The space is divided into four quadrants by two horizontal and two vertical lines, resulting in a 2x2 grid. The top-left quadrant is labeled "PGH", the top-right "PHL", the bottom-left "ATL", and the bottom-right "NE". The horizontal axis is marked at 30 and 40, and the vertical axis is marked at 10 and 20. Two points are plotted: one in the "PGH" region at approximately (32, 18) and another in the "ATL" region at approximately (38, 12). To the right, a hierarchical tree structure is shown with a root node at (30, 10). The root has four children labeled SW, NE, and two unlabeled circles. The SW child further branches into two unlabeled circles, representing a recursive decomposition of the space into smaller regions.

15-826

Copyright: C. Faloutsos (2011)

19


 CMU SCS

Quad-trees - search?

- find cities with ($35 < x < 45$, $15 < y < 25$):

15-826

Copyright: C. Faloutsos (2011)

20

CMU SCS

Quad-trees - search?

- find cities with ($35 < x < 45$, $15 < y < 25$):

20
10

30 40

PGH
PHL
ATL

SW X X X NE
30,10
40,20



Quad-trees - search?

- pseudocode:

```

range-query( tree_ptr, range)
  if (tree_ptr == NULL) exit;
  if (tree_ptr->point within range){
    print tree_ptr->point;
  for each quadrant {
    if ( range intersects quadrant ) {
      range-query( tree_ptr->quadrant_ptr, range);
    }
  }
}

```

15-826

Copyright: C. Faloutsos (2011)

22



Quad-trees - k-nn search?

- k-nearest neighbor algo - more complicated:
 - find ‘good’ neighbors and put them in a stack
 - go to the most promising quadrant, and update the stack of neighbors
 - until we hit the leaves

15-826

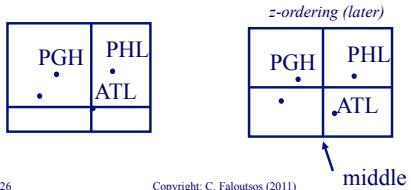
Copyright: C. Faloutsos (2011)

23



Quad-trees - discussion

- great for 2- and 3-d spaces
 - several variations, like fixed decomposition:
 - ‘adaptive’
 - ‘fixed’



15-826

Copyright: C. Faloutsos (2011)

24





Quad-trees - discussion

- but: unsuitable for higher-d spaces (why?)

15-826

Copyright: C. Faloutsos (2011)

25



Quad-trees - discussion

- but: unsuitable for higher-d spaces (why?)
- A: 2^d pointers, per node!
- Q: how to solve this problem?
- A: k-d-trees!

15-826

Copyright: C. Faloutsos (2011)

26



Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
 - main memory: quad-trees
 - main memory: k-d-trees
- spatial access methods
 - text
 - ...



15-826

Copyright: C. Faloutsos (2011)

27

CMUSCS

k-d-trees

- Binary trees, with alternating ‘discriminators’

A 2D plot showing four data points: PGH (top-left), PHL (top-right), ATL (bottom-right), and SW (bottom-left). The plot is divided into four quadrants by two perpendicular lines. The bottom-left quadrant contains the point SW. On the right, a quad-tree node is shown with a central oval labeled (30,10) and four branches pointing to the quadrants, representing the hierarchical partitioning of the space.

15-826

Copyright: C. Faloutsos (2011)

28

CMU SCS

k-d-trees

- Binary trees, with alternating ‘discriminators’

The diagram illustrates a k-d-tree structure. On the left, a 2D plot shows four points labeled PGH, PHL, ATL, and W. The plot is divided by a vertical line at x=30 and a horizontal line at y=10. The region above and to the left of these lines contains points PGH and W. The region below and to the right contains points PHL and ATL. On the right, a tree structure is shown with a root node containing the point (30, 10). This root node branches into two children, labeled W and E.

15-826

Copyright: C. Faloutsos (2011)

29


 CMU SCS

k-d-trees

- Binary trees, with alternating ‘discriminators’

A 2D plot showing a square region with points PGH, PHL, and ATL. The region is split vertically at $x=30$. The left half contains points PGH and PHL; the right half contains point ATL. To the right, a decision tree node is shown with a circle containing $(30, 10)$. It has two branches: one labeled $x \leq 30$ pointing to the left, and another labeled $x > 30$ pointing to the right.

15-826

Copyright: C. Faloutsos (2011)

30

 CMU SCS

k-d-trees

- Binary trees, with alternating ‘discriminators’

15-826 Copyright: C. Faloutsos (2011) 31

 CMU SCS

(Several demos/applets, e.g.)

- <http://donar.umiacs.umd.edu/quadtree/points/kdtree.html>

15-826 Copyright: C. Faloutsos (2011) 32

 CMU SCS

Indexing - Detailed outline

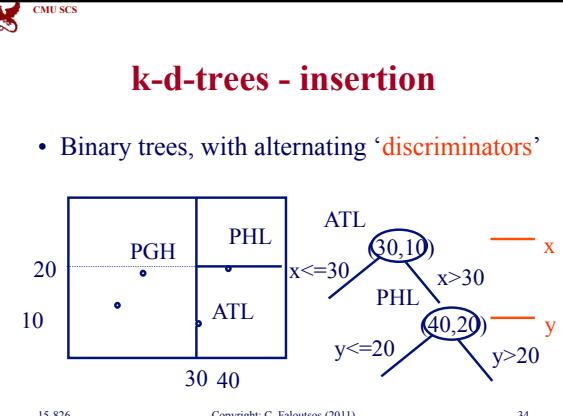
- primary key indexing
- secondary key / multi-key indexing
 - main memory: quad-trees
 - main memory: k-d-trees
 - insertion; deletion
 - range query; k-nn query
- spatial access methods
- text
- ...

15-826 Copyright: C. Faloutsos (2011) 33

 CMU SCS

k-d-trees - insertion

- Binary trees, with alternating ‘discriminators’



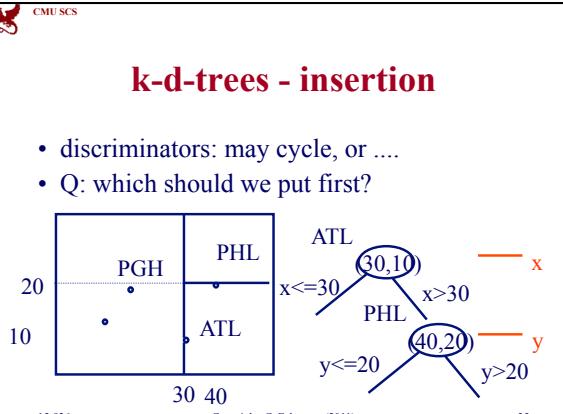
PGH PHL
20 10
x <= 30 x > 30
ATL PHL
y <= 20 y > 20
30 40

15-826 Copyright: C. Faloutsos (2011) 34

 CMU SCS

k-d-trees - insertion

- discriminators: may cycle, or
- Q: which should we put first?



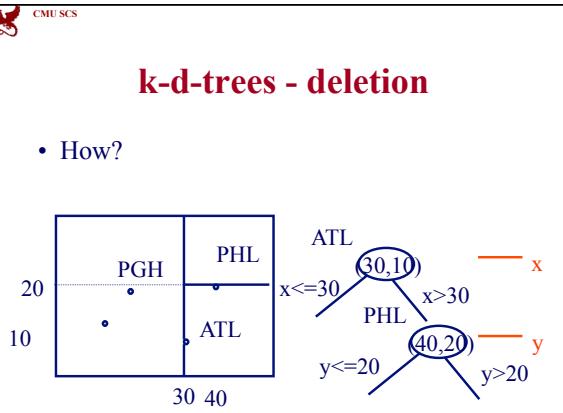
PGH PHL
20 10
x <= 30 x > 30
ATL PHL
y <= 20 y > 20
30 40

15-826 Copyright: C. Faloutsos (2011) 35

 CMU SCS

k-d-trees - deletion

- How?



PGH PHL
20 10
x <= 30 x > 30
ATL PHL
y <= 20 y > 20
30 40

15-826 Copyright: C. Faloutsos (2011) 36

 CMU SCS

k-d-trees - deletion

- Tricky! ‘delete-and-promote’ (or ‘mark as deleted’)

15-826 Copyright: C. Faloutsos (2011) 37

 CMU SCS

k-d-trees - range query

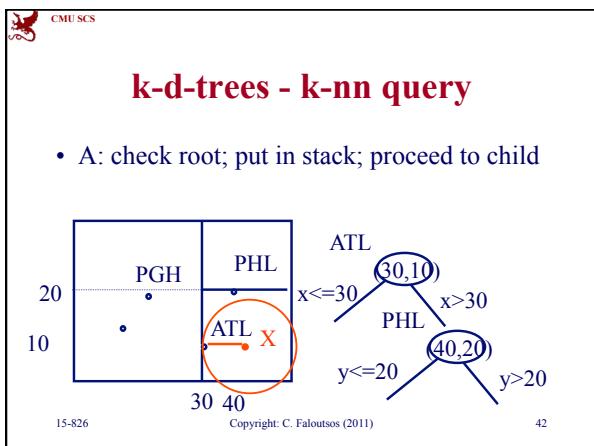
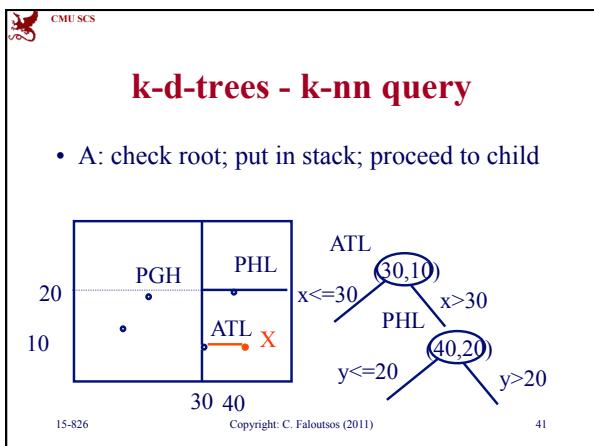
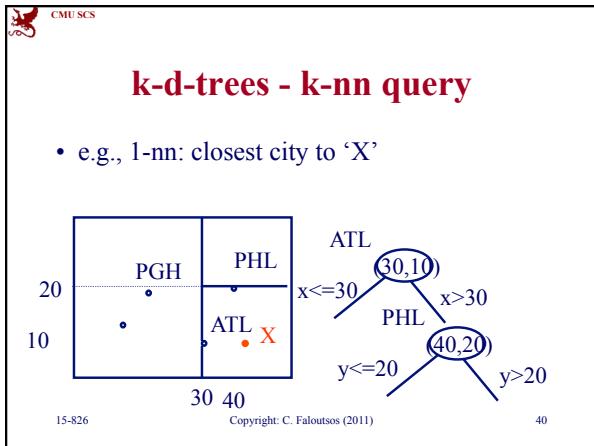
15-826 Copyright: C. Faloutsos (2011) 38

 CMU SCS

k-d-trees - range query

- similar to quad-trees: check the root; proceed to appropriate child(ren).

15-826 Copyright: C. Faloutsos (2011) 39





Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
 - main memory: quad-trees
 - main memory: k-d-trees
 - insertion; deletion
 - range query; k-nn query
 - discussion
- spatial access methods
- text

15-826

Copyright: C. Faloutsos (2011)

43



k-d trees - discussion

- great for main memory & low 'd' (~ 10)
- Q: what about high-d?
- A:
- Q: what about disk
- A:

15-826

Copyright: C. Faloutsos (2011)

44



k-d trees - discussion

- great for main memory & low 'd' (~ 10)
- Q: what about high-d?
- A: most attributes don't ever become discriminators
- Q: what about disk?
- A: Pagination problems, after ins./del.
(solutions: next!)

15-826

Copyright: C. Faloutsos (2011)

45



Conclusions

- sec. keys: B-tree indices (+ postings lists)
- multi-key, main memory methods:
 - quad-trees
 - k-d-trees

15-826

Copyright: C. Faloutsos (2011)

46



References

- [Bentley75] J.L. Bentley: *Multidimensional Binary Search Trees Used for Associative Searching*, CACM, 18,9, Sept. 1975.
- [Finkel74] R.A. Finkel, J.L. Bentley: *Quadtrees: A data structure for retrieval on composite keys*, ACTA Informatica, 4,1, 1974
- Applet: eg., <http://donar.umiacs.umd.edu/quadtree/points/kdtree.html>

15-826

Copyright: C. Faloutsos (2011)

47
