15-826: Multimedia Databases and Data Mining

Fractals - case studies - I
C. Faloutsos

Outline

Goal: ‘Find similar / interesting things’
- Intro to DB
- Indexing - similarity search
- Data Mining

Indexing - Detailed outline

- primary key indexing
- secondary key / multi-key indexing
- spatial access methods
  - 2-ordering
  - R-trees
  - misc
- fractals
  - intro
  - applications
- text

(Fractals mentioned before:)

- for performance analysis of R-trees
- fractals for dim. reduction

Case study#1: R-tree performance

Problem
- Given
  - N points in E-dim space

- Estimate # disk accesses for a range query
  \( q_1 \times \ldots \times q_n \)

(assume: ‘good’ R-tree, with tight, cube-like MBRs)
Case study#1: R-tree performance

Problem
- Given
  - N points in E-dim space
  - with fractal dimension D
- Estimate # disk accesses for a range query
  \((q_1 \times \ldots \times q_n)\)

(assume: ‘good’ R-tree, with tight, cube-like MBRs)

Typically, in DB Q-opt: uniformity + independence

Examples: World’s countries

- BUT: area vs population for ~200 countries

- neither uniform, nor independent!

Examples: TIGER files

- neither uniform, nor independent!

How to proceed?

- recall the [Pagel+] formula, for range queries of size \(q_1 \times q_2\)

\[
\#\text{DiskAccesses}(q_1, q_2) = \sum (x_{i,j} + q_1) * (x_{i,j} + q_2)
\]

But:
formula needs to know the \(x_{i,j}\) sizes of MBRs!

R-trees - performance analysis

I.e: for range queries - how many disk accesses,
if we just now that we have
- \(N\) points in \(E\)-d space?
A: can not tell! need to know distribution

Typically, in DB Q-opt: uniformity + independence
**Reminder:**

**Hausdorff or box-counting fd:**
- Box counting plot: $\log(N(r))$ vs $\log(r)$
- $r$: grid side
- $N(r)$: count of non-empty cells
- (Hausdorff) fractal dimension $D_0$:
  
  \[ D_0 = -\frac{\partial \log(N(r))}{\partial \log(r)} \]

**R-trees - performance analysis**

Q: OK - so we are told that the **Hausdorff** fractal dim. = $D_0$ - Next step?
(also know that there are at most $C$ points per page)

$D_0=1$

$D_0=2$

**Hausdorff fd:**

$N(r) \sim r^{D_0}$

# non-empty cells of side $r$

**Reminder**

- dfn of Hausdorff fd implies that

  \[ N(r) \sim r^{D_0} \]

**R-trees - performance analysis**

Hint: dfn of Hausdorff f.d.:

Felix Hausdorff (1868-1942)
R-trees - performance analysis

Q (rephrased): what is the side s1, s2, ... of parent nodes, given N data points, packed by C, with f.d. = D0

A: (educated guess)
- \( s = s_1 = s_2 = \ldots \) - square-like MBRs
- N/C non-empty cells = \( K \times s^{-D0} \)

log(#cells) vs log(s)

Q: does it make sense?

R-trees - performance analysis

Details of derivations: in [PODS 94].
Finally, expected side \( s \) of parent MBRs:
\[ s = (C/N)^{1/D0} \]

Q: sanity check: how does \( s \) change with \( D0 \)?
A:

R-trees - performance analysis

Q: Final-final formula (# disk accesses for range queries \( q1 \times q2 \times \ldots \)):
A:
R-trees - performance analysis

Q: Final-final formula (# disk accesses for range queries $q_1 \times q_2 \times ...$):
A: # of parent-node accesses:
$$\frac{N}{C} * (s + q_1) * (s + q_2) * ... * (s + q_E)$$
A: # of grand-parent node accesses
$$\frac{N}{(C^2)} * (s' + q_1) * (s' + q_2) * ... * (s' + q_E)$$

$N/(C^2)$, $s'$, and $q$ depend on the context:
- $s'$ = $\frac{C^2}{N}$ in the case of 2D-uniform distribution.
- $s'$ = IUE (x-y star coordinates).
- $s'$ = $\frac{C^2}{N} / D0$ for MG-county.

Results:
- IUE (x-y star coordinates)
- LB County
- MG-county

# leaf accesses

As an example, for 2D-uniform distribution:

- # leaf accesses vs. query size on both sides.

---

R-trees - performance analysis

Q: Final-final formula (# disk accesses for range queries $q_1 \times q_2 \times ...$):
A: # of parent-node accesses:
$$\frac{N}{C} * (s + q_1) * (s + q_2) * ... * (s + q_E)$$
A: # of grand-parent node accesses
$$\frac{N}{(C^2)} * (s' + q_1) * (s' + q_2) * ... * (s' + q_E)$$

$N/(C^2)$, $s'$, and $q$ depend on the context:
- $s'$ = $\frac{C^2}{N}$ in the case of 2D-uniform distribution.
- $s'$ = IUE (x-y star coordinates).
- $s'$ = $\frac{C^2}{N} / D0$ for MG-county.

Results:
- IUE (x-y star coordinates)
- LB County
- MG-county

# leaf accesses

As an example, for 2D-uniform distribution:

- # leaf accesses vs. query size on both sides.
R-trees - performance analysis
Conclusions: usually, <5% relative error, for range queries

Indexing - Detailed outline
• fractals
  – intro
  – applications
  • disk accesses for R-trees (range queries)
  • dimensionality reduction
  • selectivity in M-trees
  • dim. curse revisited
  • “fat fractals”
  • quad-tree analysis [Gaede+]
  • ...

Case study #2: Dim. reduction
Problem definition: ‘Feature selection’
• given \( N \) points, with \( E \) dimensions
• keep the \( k \) most ‘informative’ dimensions [Traina+,SBBD’00]

Dim. reduction - w/ fractals

Dim. reduction
Problem definition: ‘Feature selection’
• given \( N \) points, with \( E \) dimensions
• keep the \( k \) most ‘informative’ dimensions
Re-phrased: spot and drop attributes with strong (non-)linear correlations
Q: how do we do that?

Dim. reduction
A: Hint: correlated attributes do not affect the intrinsic/fractal dimension, e.g., if
\[
y = f(x,z,w)
\]
we can drop \( y \)
(hence: ‘partial fd’ (PFD) of a set of attributes = the fd of the dataset, when projected on those attributes)
Dim. reduction - w/ fractals

- keep the attribute with highest partial fd below the full f.d.

(b) Line

- add the one that causes the highest increase in pfd

- etc., until we are within \( \varepsilon \) from the full f.d.

Q: Algorithm?

- (backward elimination: ~ reverse)
  - drop the attribute with least impact on the p.f.d.
  - repeat
  - until we are \( \varepsilon \) below the full f.d.
Dim. reduction - w/ fractals

• Q: what is the smallest # of attributes we should keep?
• A: we should keep at least as many as the f.d. (and probably, a few more)

Dim. reduction - w/ fractals

E.g., on the ‘currency’ dataset

- Results: E.g., on the ‘currency’ dataset
- (daily exchange rates for USD, HKD, BP, FRF, DEM, JPY - i.e., 6-d vectors, one per day - base currency: CAD)
  - e.g.: FRF

E.g., on the ‘currency’ dataset

<table>
<thead>
<tr>
<th>Currency</th>
<th>log(#pairs(&lt;r))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Currency dataset</td>
<td>log(r)</td>
</tr>
<tr>
<td>USD</td>
<td>1.98</td>
</tr>
</tbody>
</table>

E.g., on the eigenface dataset

- 16-d vectors, one for each of ~1K faces
- Eigenfaces slp=4.2506
- German Mark
- French Franc
- British Pound
- Japanese Yen
- American Dollar
- Hong Kong Dollar

if unif + indep.

if unif + indep.
E.g., on the eigenface dataset

Dim. reduction - w/ fractals

Conclusion:
- can do non-linear dim. reduction

References