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
  - z-ordering
  - R-trees
  - misc
- fractals
  - intro
  - applications
- text

‘Fat’ fractals & R-tree performance on region data

- Problem [Proietti+,’99]
- Given
  - N (# of data regions)
- estimate how many of them will qualify for the average range query (q1 x q2 x ... qE)
Of course, we need more info
Q: what?

R-tree performance on region data

A: the distributions of their sizes

Q: do we also need some info about the locations?
R-tree performance on region data

A: the distributions of their sizes

Q: do we also need some info about the locations?
A: no (not for range queries)

R-tree performance on region data

A: the distributions of their sizes

Q: what exactly would we need?

A: for self-similar regions (~ ‘fat’ fractals), we just need the slope of the Korcak law!
(and the total area) [Proietti+]

More power laws: areas – Korcak’s law

Scandinavian lakes
Any pattern?

More power laws: areas – Korcak’s law

log(count(> area)) vs log(area)

Scandinavian lakes
area vs complementary cumulative count (log-log axes)

B (patchiness exponent)

Japan islands

More power laws: Korcak
More power laws: Korcak

Korcak’s law & “fat fractals”

R-tree performance on regions

- Once we know ‘B’ (and the total area)
- we can second-guess the individual sizes
- and then apply the [Page+93] formula
- Bottom line:

‘Fat’ fractals - observation

B = \frac{D_\text{H}}{d}

B: patchiness exp; d: dim, D_H: Hausdorff of periphery
‘Fat’ fractals - observation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>B</th>
<th>S</th>
<th>Dg - 2D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laser</td>
<td>1.94</td>
<td>0.53</td>
<td>0.61</td>
</tr>
<tr>
<td>Diameters</td>
<td>1.85</td>
<td>0.58</td>
<td>0.62</td>
</tr>
<tr>
<td>EX0</td>
<td>1.49</td>
<td>0.47</td>
<td>0.40</td>
</tr>
<tr>
<td>August, month</td>
<td>1.40</td>
<td>0.25</td>
<td>0.24</td>
</tr>
<tr>
<td>Fezang, month</td>
<td>1.19</td>
<td>0.15</td>
<td>0.11</td>
</tr>
<tr>
<td>Italy plants</td>
<td>1.30</td>
<td>0.15</td>
<td>0.11</td>
</tr>
<tr>
<td>Whole earth</td>
<td>1.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Cypress vegetation</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

‘Fat’ fractals

• intuition behind \( B = D_{H} / d \) ?

• A: consider ‘flooding’:

Conclusions

• ‘Fat’ fractals model regions well
• patchiness exp.: \( B = D_{H} / d \)
• can help us estimate selectivities

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+]
    • nn queries [Belussi+]
Fractals and Quadtrees

- Problem: how many quadtree nodes will we need, to store a region in some level of approximation? [Gaede+96]

Fractals and Quadtrees

- I.e.:

Fractals and Quadtrees

- Datasets:
  - Franconia
  - Brain Atlas

Fractals and Quadtrees

- Hint:
  - assume that the boundary is self-similar, with a given fd
  - how will the quad-tree (oct-tree) look like?
Fractals and Quadtrees

Let $p_g(i)$ the prob. to find a gray node at level $i$.
If self-similar, what can we say for $p_g(i)$?

A: $p_g(i) = p_g = \text{constant}$

Fractals and Quadtrees

Assume only ‘gray’ and ‘white’ nodes (ie., no volume’)
Assume that $p_g$ is given - how many gray nodes at level $i$?

A: 1 at level 0;
$4p_g$
$(4p_g)^2 (4p_g)^3$
$\ldots$
$(4p_g)^i$

Fractals and Quadtrees

I.e.:
\[
\begin{align*}
\text{# of quadtree 'blocks'} & \quad \sim \quad (4p_g)^i \\
\text{level of quadtree} & \quad \sim \quad \log(4p_g)^i
\end{align*}
\]
Fractals and Quadtrees

- Conclusion: Self-similarity leads to easy and accurate estimation.

\[
\log_2(\text{#blocks}) \begin{cases} \text{level} \end{cases}
\]

Fractals and Quadtrees

- Final observation: relationship between \( p_g \) and fractal dimension?

\[
(4^i p_g) = \text{# of gray nodes at level } i = \text{# of Hausdorff grid-cells of side } (1/2)^i = r
\]

Eventually: \( D_N = 2 + \log_2(p_g) \)

and, for E-d spaces: \( D_N = E + \log_2(p_g) \)
Fractals and Quadtrees

for E-d spaces: \( D_H = E + \log_2(p_f) \)

Sanity check:
- point: \( D_H = 0 \) \( p_f = ?? \)
- line in 2-d: \( D_H = 1 \) \( p_f = ?? \)
- plane in 2-d: \( D_H = 2 \) \( p_f = ?? \)

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+]
    - NN queries [Belussi+]

Fractals and Quadtrees

Final conclusions:
- self-similarity leads to estimates for # of z-values = # of quadtree/oct-tree blocks
- close dependence on the Hausdorff fractal dimension of the boundary

NN queries

- Q: in NN queries, what is the effect of the shape of the query region? [Belussi+95]

  \[ L_{inf} \quad \rightarrow \quad L_1 \quad \rightarrow \quad L_2 \]

NN queries

- Q: What about \( L_1, L_{inf} \)?

  \[ \log(\#\text{pairs-within}(\leq d)) \quad \rightarrow \quad D_2 \quad \rightarrow \quad \log(d) \]
NN queries

- Q: What about $L_1$, $L_{\text{inf}}$?
- A: Same slope, different intercept

$\log(\# \text{pairs-within}(\leq d))$

$D_2$

$\log(d)$

$N_2$ neighbors

$N_{\text{inf}}$ neighbors

volume: $V_2$

volume: $V_{\text{inf}}$

NN queries

- Consider sphere with volume $V_{\text{inf}}$ and $r'$ radius

$(t/r')^E = V_2 / V_{\text{inf}}$

$(t/r')^D_2 = N_2 / N_{\text{inf}}$

$N_2' = N_{\text{inf}}$ (since shape does not matter)

and finally:
NN queries
Conclusions: for self-similar datasets
• Avg # neighbors: grows like $(distance)^{D_x}$, regardless of query shape (circle, diamond, square, e.t.c.)

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+]
    • nn queries [Belussi+]
  – Conclusions

Fractals - overall conclusions
• self-similar datasets: appear often
• powerful tools: correlation integral, NCDF, rank-frequency plot
• intrinsic/fractal dimension helps in
  – estimations (selectivities, quadtrees, etc)
  – dim. reduction / dim. curse
• (later: can help in image compression...)

References
•Proietti, G. and C. Faloutsos (March 23-26, 1999). I/O complexity for range queries on region data stored using an R-tree. International Conference on Data Engineering (ICDE), Sydney, Australia.