15-826: Multimedia Databases and Data Mining

Lecture #12: Fractals - case studies Part III
(quadtrees, knn queries)

C. Faloutsos

Must-read Material

- Alberto Belussi and Christos Faloutsos,
  Estimating the Selectivity of Spatial Queries
  Using the 'Correlation' Fractal Dimension
  Proc. of VLDB, p. 299-310, 1995

Optional Material

Optional, but very useful: Manfred Schroeder
Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise

Outline

Goal: ‘Find similar / interesting things’

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

Optional = NOT in exam (but useful as mental drill!)
Indexing - Detailed outline

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

Optional

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.: # of quadtree 'blocks' (= # gray nodes)

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;

$$4^i p_g \cdot (4^i p_g) \cdot (4^i p_g) \cdot \ldots \cdot (4^i p_g)^i$$
Fractals and Quadtrees

- I.e.: $\log_2(\text{# of quadtree 'blocks')} = \log[(4*p_g)^i]

Fractals and Quadtrees

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

$\log_2(\text{#blocks})$ vs. level
Fractals and Quadtrees

- Final observation: relationship between $p_g$ and fractal dimension?

- A: very close:
  \[(4^i p_g) = \# \text{of gray nodes at level } i = \# \text{of Hausdorff grid-cells of side } (1/2)^i = r\]
  Eventually: $D_H = 2 + \log_2(p_g)$
  and, for E-d spaces $D_H = E + \log_2(p_g)$
Fractals and Quadtrees

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

Sanity check:
- point in 2-d: \( D_H = 0 \) \( p_g = ?? \)
- line in 2-d: \( D_H = 1 \) \( p_g = ?? \)
- plane in 2-d: \( D_H = 2 \) \( p_g = ?? \)
- point in 3-d: \( D_H = 0 \) \( p_g = ?? \)

Fractals and Quadtrees

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

Sanity check:
- point in 2-d: \( D_H = 0 \) \( p_g = 1/4 \)
- line in 2-d: \( D_H = 1 \) \( p_g = 1/2 \)
- plane in 2-d: \( D_H = 2 \) \( p_g = 1 \)
- point in 3-d: \( D_H = 0 \) \( p_g = 1/8 \)

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

Indexing - Detailed outline

- fractals
  - intro
  - applications
    - disk accesses for R-trees (range queries)
    - dimensionality reduction
    - dim. curse revisited
    - quad-tree analysis [Gaede+]
- \( \text{nn queries} \) [Belussi+]
NN queries

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

• Q: What about $L_1$, $L_{\infty}$?

• A: Same slope, different intercept
NN queries

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

log(#neighbors) vs. log(d)

NN queries

• Q: What about the intercept? I.e., what can we say about $N_{2}$ and $N_{\text{inf}}$?

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

$N_{2}$ neighbors

$N_{\text{inf}}$ neighbors

$r'$

$r$

volume: $V_{2}$

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

NN queries

• Consider sphere with volume $V_{\text{inf}}$ and $r'$ radius
  
  $(r/r')^E = V_{2} / V_{\text{inf}}$
  
  $(r/r')^D = N_{2} / N_{2}'$
  
  $N_{2}' = N_{\text{inf}}$ (since shape does not matter)
  
  and finally:
NN queries

\[(N_2 / N_{inf})^{1/D_2} = (V_2 / V_{inf})^{1/E}\]

Optional

NN queries

Conclusions: for self-similar datasets

- Avg # neighbors: grows like \[(distance)^{D_2}\]
- 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
    - dim. curse revisited
    - 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

