On the Average Performance of Fixed Partial Match Queries in Random Relaxed $K$-d Trees

Conrado Martínez
Univ. Politècnica Catalunya, Barcelona
AofA 2014, Paris, France

–Dedicated to Ph. Flajolet

Joint work with:

Amalia Duch  Gustavo Lau
The problem

Input:
- a collection of \( n \) multidimensional records, each with a \( K \)-dimensional key
  \[
  \mathbf{x} = (x_0, \ldots, x_{K-1}) \in \mathcal{D}_0 \times \cdots \times \mathcal{D}_{K-1}
  \]
  stored in a suitable multidimensional data structure, e.g., a \( K \)-d tree, a \( K \)-dimensional quadtree, a \( K \)-d trie, . . .
- a query
  \[
  \mathbf{q} = (q_0, \ldots, q_{K-1}), \quad q_i \in \mathcal{D}_i \cup \{\ast\}
  \]
  with \( s \) specified coordinates \((q_i \neq \ast)\), \( 0 < s < K \), for example
  \[
  \mathbf{q} = (0.1, \ast, \ast, 0.3, 0.07), \quad s = 3, K = 5
  \]

Output: the set of records such that \( \mathbf{x} \) satisfies the query \( \mathbf{q} \) (i.e., \( x_i = q_i \) whenever \( q_i \neq \ast \))
The problem

- **Input:**
  - a collection of \( n \) multidimensional records, each with a \( K \)-dimensional key
    
    \[
    \mathbf{x} = (x_0, \ldots, x_{K-1}) \in \mathcal{D}_0 \times \cdots \times \mathcal{D}_{K-1}
    \]

    stored in a suitable **multidimensional data structure**, e.g., a \( K \)-d tree, a \( K \)-dimensional quadtree, a \( K \)-d trie, \ldots
  - a query
    
    \[
    \mathbf{q} = (q_0, \ldots, q_{K-1}), \quad q_i \in \mathcal{D}_i \cup \{\ast\}
    \]

    with \( s \) specified coordinates \((q_i \neq \ast)\), \( 0 < s < K \), for example
    
    \[
    \mathbf{q} = (0.1, \ast, \ast, 0.3, 0.07), \quad s = 3, K = 5
    \]

- **Output:** the set of records such that \( \mathbf{x} \) satisfies the query \( \mathbf{q} \)
  (i.e., \( x_i = q_i \) whenever \( q_i \neq \ast \))
The problem

Input:
- a collection of \( n \) multidimensional records, each with a \( K \)-dimensional key

\[
\mathbf{x} = (x_0, \ldots, x_{K-1}) \in \mathcal{D}_0 \times \cdots \times \mathcal{D}_{K-1}
\]

stored in a suitable multidimensional data structure, e.g., a \( K \)-d tree, a \( K \)-dimensional quadtree, a \( K \)-d trie, \ldots

- a query

\[
\mathbf{q} = (q_0, \ldots, q_{K-1}), \quad q_i \in \mathcal{D}_i \cup \{\ast\}
\]

with \( s \) specified coordinates (\( q_i \neq \ast \)), \( 0 < s < K \), for example

\[
\mathbf{q} = (0.1, \ast, \ast, 0.3, 0.07), \quad s = 3, K = 5
\]

Output: the set of records such that \( \mathbf{x} \) satisfies the query \( \mathbf{q} \) (i.e., \( x_i = q_i \) whenever \( q_i \neq \ast \))
The problem

- **Input:**
  - a collection of $n$ multidimensional records, each with a $K$-dimensional key

\[
x = (x_0, \ldots, x_{K-1}) \in \mathcal{D}_0 \times \cdots \times \mathcal{D}_{K-1}
\]

stored in a suitable multidimensional data structure, e.g., a $K$-d tree, a $K$-dimensional quadtree, a $K$-d trie, …

- a query

\[
q = (q_0, \ldots, q_{K-1}), \quad q_i \in \mathcal{D}_i \cup \{\ast\}
\]

with $s$ specified coordinates ($q_i \neq \ast$), $0 < s < K$, for example

\[
q = (0.1, \ast, \ast, 0.3, 0.07), \quad s = 3, K = 5
\]

- **Output:** the set of records such that $x$ satisfies the query $q$ (i.e., $x_i = q_i$ whenever $q_i \neq \ast$)
Definition

A $K$-d tree for a set $X$ is either the empty tree if $X = \emptyset$ or a binary tree where:

- the root contains $y \in X$ and some value $i$ (the discriminant), $0 \leq i < K$
- the left subtree is a $K$-d tree for $X^- = \{x \in X \mid x_i < y_i\}$
- the right subtree is a $K$-d tree for $X^+ = \{x \in X \mid y_i < x_i\}$
$K$-d trees
K-d trees
$K$-d trees
$K$-d trees
K-d trees
K-d trees

- All discriminants at level $\ell$ equal to $\ell \mod K \Rightarrow$ standard $K$-d trees
- Discriminants independent and uniformly drawn from $\{0, \ldots, K - 1\} \Rightarrow$ relaxed $K$-d trees
- Discriminants chosen to cut along the longest side of the region where a new key is inserted $\Rightarrow$ squarish $K$-d trees
- \ldots
The algorithm

procedure PARTIAL_MATCH(T, q)
    if T = □ then return
    i ← T.dscr; x ← T.key
    if x satisfies q then
        Add x to the output
    if q_i = * then
        PARTIAL_MATCH(T.left, q)
        PARTIAL_MATCH(T.right, q)
    else
        if q_i < x_i then
            PARTIAL_MATCH(T.left, q)
        else
            PARTIAL_MATCH(T.right, q)
The probability model

- Without loss of generality: we assume $\mathcal{D}_i = [0, 1]$
- The standard probability model in this area: the $n$ keys are drawn i.i.d from some continuous distribution in $[0, 1]^K$ and inserted into an initially empty $K$-d tree $\Rightarrow$ random (standard/relaxed/$\ldots$) $K$-d tree $T_n$
- Equivalently: for any $i$, $0 \leq i < K$, and any $y$ in the random $K$-d tree $T_n$, $y$ is the $r$th smallest along the $i$th coordinate with
  $$
  \mathbb{P}\left\{ \#\{x \in T_n \mid x_i \leq y_i\} = r \right\} = \frac{1}{n}
  $$
  for $r = 1, 2, \ldots, n \Rightarrow$ shapes of $K$-d trees behave as binary search trees
The probability model

- Without loss of generality: we assume $\mathcal{D}_i = [0, 1]$
- The standard probability model in this area: the $n$ keys are drawn i.i.d from some continuous distribution in $[0, 1]^K$ and inserted into an initially empty $K$-d tree $\Rightarrow$ random (standard/relaxed/...) $K$-d tree $T_n$
- Equivalently: for any $i$, $0 \leq i < K$, and any $y$ in the random $K$-d tree $T_n$, $y$ is the $r$th smallest along the $i$th coordinate with
  \[
P\{\#\{x \in T_n \mid x_i \leq y_i\} = r\} = \frac{1}{n}\]
  for $r = 1, 2, \ldots, n \Rightarrow$ shapes of $K$-d trees behave as binary search trees
The probability model

Without loss of generality: we assume $\mathcal{D}_i = [0, 1]$

The standard probability model in this area: the $n$ keys are drawn i.i.d from some continuous distribution in $[0, 1]^K$ and inserted into an initially empty $K$-d tree $T_n$.

Equivalently: for any $i$, $0 \leq i < K$, and any $y$ in the random $K$-d tree $T_n$, $y$ is the $r$th smallest along the $i$th coordinate with

$$P \{ \# \{ x \in T_n \mid x_i \leq y_i \} = r \} = \frac{1}{n}$$

for $r = 1, 2, \ldots, n \Rightarrow$ shapes of $K$-d trees behave as binary search trees.
Previous work: Random queries

For a query \( q \) its pattern \( u(q) = (u_0, \ldots, u_{K-1}) \) is a bitvector with \( u_i = S \) if \( q_i \neq \ast \) and \( u_i = \ast \) if \( q_i = \ast \). For example, \( q = (0.1, \ast, \ast, 0.3, 0.07) \Rightarrow u = S \ast \ast SS \)

Let \( P_{n,u} \) denote the cost (number of visited nodes) of a random partial match query with pattern \( u \), and let \( \rho = s/K \). Then

\[
P_{n,u} := \mathbb{E}(P_{n,u}) = \beta_u n^\alpha + o(n^\alpha)
\]

where \( \alpha = \alpha(\rho) \) is \( 0 < \alpha < 1 \) for any \( \rho \in (0, 1) \) and \( \beta_u \) is a constant depending on the query pattern.
Previous work: Random queries

For a query $q$ its pattern $u(q) = (u_0, \ldots, u_{K-1})$ is a bitvector with $u_i = S$ if $q_i \neq \ast$ and $u_i = \ast$ if $q_i = \ast$. For example, $q = (0.1, \ast, \ast, 0.3, 0.07) \Rightarrow u = S \ast \ast SS$

Let $P_{n,u}$ denote the cost (number of visited nodes) of a random partial match query with pattern $u$, and let $\rho = s/K$. Then

$$P_{n,u} := \mathbb{E}(P_{n,u}) = \beta_u n^\alpha + o(n^\alpha)$$

where $\alpha = \alpha(\rho)$ is $0 < \alpha < 1$ for any $\rho \in (0, 1)$ and $\beta_u$ is a constant depending on the query pattern.
Previous work: Random queries

Different data structures have different “characteristic” exponents $\alpha$

Example ($s = 1, K = 2$)

- Standard $K$-d trees: $\alpha = 0.56 \ldots$
- Relaxed $K$-d trees: $\alpha = 0.61 \ldots$
- Squarish $K$-d trees: $\alpha = 0.5$
- $K$-d tries: $\alpha = 0.5$

In general, $\alpha(\rho) = 1 - \rho + \vartheta(\rho)$, with $\vartheta(\rho) \geq 0$ “small” in $(0, 1)$

Obs: the constant $\beta$ for relaxed $K$-d trees only depends on $s$ and $K$, not on the pattern, because of the randomly chosen discriminants
Previous work: Random queries

- Different data structures have different “characteristic” exponents $\alpha$

Example ($s = 1, K = 2$)

- Standard $K$-d trees: $\alpha = 0.56 \ldots$
- Relaxed $K$-d trees: $\alpha = 0.61 \ldots$
- Squarish $K$-d trees: $\alpha = 0.5$
- $K$-d tries: $\alpha = 0.5$

In general, $\alpha(p) = 1 - p + \theta(p)$, with $\theta(p) \geq 0$ “small” in $(0, 1)$

Obs: the constant $\beta$ for relaxed $K$-d trees only depends on $s$ and $K$, not on the pattern, because of the randomly chosen discriminants
Previous work: Random queries

- Different data structures have different “characteristic” exponents $\alpha$

Example ($s = 1, K = 2$)

- Standard $K$-d trees: $\alpha = 0.56 \ldots$
- Relaxed $K$-d trees: $\alpha = 0.61 \ldots$
- Squarish $K$-d trees: $\alpha = 0.5$
- $K$-d tries: $\alpha = 0.5$

In general, $\alpha(\rho) = 1 - \rho + \vartheta(\rho)$, with $\vartheta(\rho) \geq 0$ “small” in $(0, 1)$

Obs: the constant $\beta$ for relaxed $K$-d trees only depends on $s$ and $K$, not on the pattern, because of the randomly chosen discriminants.
Previous work: Random queries

- Different data structures have different “characteristic” exponents $\alpha$

Example ($s = 1, K = 2$)

- Standard $K$-d trees: $\alpha = 0.56 \ldots$
- Relaxed $K$-d trees: $\alpha = 0.61 \ldots$
- Squarish $K$-d trees: $\alpha = 0.5$
- $K$-d tries: $\alpha = 0.5$

In general, $\alpha(\rho) = 1 - \rho + \vartheta(\rho)$, with $\vartheta(\rho) \geq 0$ “small” in $(0, 1)$

Obs: the constant $\beta$ for relaxed $K$-d trees only depends on $s$ and $K$, not on the pattern, because of the randomly chosen discriminants
Different data structures have different “characteristic” exponents $\alpha$

Example ($s = 1, K = 2$)

- Standard $K$-d trees: $\alpha = 0.56 \ldots$
- Relaxed $K$-d trees: $\alpha = 0.61 \ldots$
- Squarish $K$-d trees: $\alpha = 0.5$
- $K$-d tries: $\alpha = 0.5$

In general, $\alpha(\rho) = 1 - \rho + \vartheta(\rho)$, with $\vartheta(\rho) \geq 0$ “small” in $(0, 1)$

Obs: the constant $\beta$ for relaxed $K$-d trees only depends on $s$ and $K$, not on the pattern, because of the randomly chosen discriminants
Previous work: Random queries

- Different data structures have different “characteristic” exponents $\alpha$

Example ($s = 1, K = 2$)

- Standard $K$-d trees: $\alpha = 0.56 \ldots$
- Relaxed $K$-d trees: $\alpha = 0.61 \ldots$
- Squarish $K$-d trees: $\alpha = 0.5$
- $K$-d tries: $\alpha = 0.5$

- In general, $\alpha(\rho) = 1 - \rho + \vartheta(\rho)$, with $\vartheta(\rho) \geq 0$ “small” in $(0, 1)$

- Obs: the constant $\beta$ for relaxed $K$-d trees only depends on $s$ and $K$, not on the pattern, because of the randomly chosen discriminants
Previous work: Random queries

- Different data structures have different “characteristic” exponents $\alpha$

Example ($s = 1, K = 2$)

- Standard $K$-d trees: $\alpha = 0.56 \ldots$
- Relaxed $K$-d trees: $\alpha = 0.61 \ldots$
- Squarish $K$-d trees: $\alpha = 0.5$
- $K$-d tries: $\alpha = 0.5$

In general, $\alpha(\rho) = 1 - \rho + \vartheta(\rho)$, with $\vartheta(\rho) \geq 0$ “small” in $(0, 1)$

Obs: the constant $\beta$ for relaxed $K$-d trees only depends on $s$ and $K$, not on the pattern, because of the randomly chosen discriminants
Previous work: Random queries

- Different data structures have different “characteristic” exponents $\alpha$

**Example ($s = 1, K = 2$)**

- Standard $K$-d trees: $\alpha = 0.56 \ldots$
- Relaxed $K$-d trees: $\alpha = 0.61 \ldots$
- Squarish $K$-d trees: $\alpha = 0.5$
- $K$-d tries: $\alpha = 0.5$

- In general, $\alpha(\rho) = 1 - \rho + \vartheta(\rho)$, with $\vartheta(\rho) \geq 0$ “small” in $(0, 1)$

- Obs: the constant $\beta$ for relaxed $K$-d trees only depends on $s$ and $K$, not on the pattern, because of the randomly chosen discriminants
Previous work: Random queries

• **Lots of results** for random partial match queries in the literature:
Our goal: get answers for the following question

What is the (expected) cost $P_{n,q}$ of a partial match query with a given fixed query $q$?

Remarks (valid for relaxed $K$-d trees!):

1. We can assume that the specified coordinates are the first $s$ coordinates: $q = (q_0, \ldots, q_{s-1}, \ast, \ast, \ldots)$

A query $q'$ which contains a permutation of the specified coordinates of $q$ will have the same cost as $P_{n,q}$.

If $q' = (q_0, \ldots, 1 - q_{i-1}, q_{i+1}, \ast, \ast, \ldots)$ then $P_{n,q} = P_{n,q'}$. 
Fixed queries

- Our goal: get answers for the following question

What is the (expected) cost $P_{n,q}$ of a partial match query with a given fixed query $q$?

Remarks (valid for relaxed $K$-d trees!):
- We can assume that the specified coordinates are the first $s$ coordinates: $q = (q_0, \ldots, q_{s-1}, *, *, \ldots)$
- A query $q'$ which contains a permutation of the specified coordinates of $q$ will have the same cost as $P_{n,q}$
- If $q' = (q_0, \ldots, 1 = q_0, \ldots, q_{s-1}, *, *, \ldots)$ then $P_{n,q} = P_{n,q'}$
Our goal: get answers for the following question

What is the (expected) cost $P_{n,q}$ of a partial match query with a given fixed query $q$?

Remarks (valid for relaxed $K$-d trees!):

- We can assume that the specified coordinates are the first $s$ coordinates: $q = (q_0, \ldots, q_{s-1}, *, *, \ldots)$
- A query $q'$ which contains a permutation of the specified coordinates of $q$ will have the same cost as $P_{n,q}$
- If $q' = (q_0, \ldots, 1 - q_i, \ldots, q_{s-1}, *, *, \ldots)$ then $P_{n,q} = P_{n,q'}$
Fixed queries

- Our goal: get answers for the following question

What is the (expected) cost $P_{n,q}$ of a partial match query with a given fixed query $q$?

- Remarks (valid for relaxed $K$-d trees!):
  - We can assume that the specified coordinates are the first $s$ coordinates: $q = (q_0, \ldots, q_{s-1}, *, *, \ldots)$
  - A query $q'$ which contains a permutation of the specified coordinates of $q$ will have the same cost as $P_{n,q}$
  - If $q' = (q_0, \ldots, 1 - q_i, \ldots, q_{s-1}, *, *, \ldots)$ then $P_{n,q} = P_{n,q'}$
Fixed queries

- Our goal: get answers for the following question

What is the (expected) cost $P_{n,q}$ of a partial match query with a given fixed query $q$?

- Remarks (valid for relaxed $K$-d trees!):
  - We can assume that the specified coordinates are the first $s$ coordinates: $q = (q_0, \ldots, q_{s-1}, *, *, \ldots)$
  - A query $q'$ which contains a permutation of the specified coordinates of $q$ will have the same cost as $P_{n,q}$
  - If $q' = (q_0, \ldots, 1 - q_i, \ldots, q_{s-1}, *, *, \ldots)$ then $P_{n,q} = P_{n,q'}$
Our goal: get answers for the following question

What is the (expected) cost $P_{n,q}$ of a partial match query with a given fixed query $q$?

Remarks (valid for relaxed $K$-d trees!):

- We can assume that the specified coordinates are the first $s$ coordinates: $q = (q_0, \ldots, q_{s-1}, *, *, \ldots)$
- A query $q'$ which contains a permutation of the specified coordinates of $q$ will have the same cost as $P_{n,q}$
- If $q' = (q_0, \ldots, 1 - q_i, \ldots, q_{s-1}, *, *, \ldots)$ then $P_{n,q} = P_{n,q'}$
Given a collection $X$ and a query $q$, the rank vector is 
$r(q) = (r_0, \ldots, r_{K-1})$, with $r_i = *$ if $q_i = *$ and 

$$r_i = \text{the number of } x \in X \text{ with } x_i \leq q_i$$

We will only write the ranks of specified coordinates 
$r = (r_0, \ldots, r_{s-1})$

We will concentrate in the analysis of the cost 
$P_{n,r} := \mathbb{E}(P_{n,r})$
Given a collection $X$ and a query $q$, the rank vector is $r(q) = (r_0, \ldots, r_{K-1})$, with $r_i = \ast$ if $q_i = \ast$ and $r_i$ is the number of $x \in X$ with $x_i \leq q_i$

We will only write the ranks of specified coordinates $r = (r_0, \ldots, r_{s-1})$

We will concentrate in the analysis of the cost $P_{n,r} := \mathbb{E}(P_{n,r})$
Given a collection $X$ and a query $q$, the rank vector is $r(q) = (r_0, \ldots, r_{K-1})$, with $r_i = \ast$ if $q_i = \ast$ and $r_i$ is the number of $x \in X$ with $x_i \leq q_i$

We will only write the ranks of specified coordinates $r = (r_0, \ldots, r_{s-1})$

We will concentrate in the analysis of the cost $P_{n,r} := \mathbb{E}(\mathcal{P}_{n,r})$
Ranks

Relating $P_{n,q}$ and $P_{n,r}$

\[ P_{n,q} = \sum_r P_{n,r} \cdot \mathbb{P}\{r(q) = r\} \]

\[ = P_{n,\bar{r}} + \text{l.o.t.} \]

with $\bar{r}_i = q_i \cdot n$ (the expected value of the rank of $q_i$)
Previous work: Fixed queries

Theorem (Curien & Joseph, 2011)

The cost of a partial match with \( q = (t, \ast) \) (equiv. \( q = (\ast, t) \)) and \( t \neq 0, 1 \) in a random 2-dimensional quadtree satisfies

\[
\lim_{n \to \infty} \frac{P_{n,q}}{n^{\alpha}} = \beta \cdot \frac{\Gamma(\alpha + 2)}{\Gamma^2 \left( \frac{\alpha}{2} + 1 \right)} \cdot (t(1 - t))^{\alpha/2}
\]

with \( \alpha = (\sqrt{17} - 3)/2 \approx 0.561 \ldots \) and
\( \beta = \text{constant factor for random partial match queries} \)
Previous work: Fixed queries

Theorem (Curien & Joseph, 2011)

The cost of a partial match with \( q = (t, *) \) (equiv. \( q = (*, t) \)) and \( t \neq 0, 1 \) in a random 2-dimensional quadtree satisfies

\[
\lim_{n \to \infty} \frac{P_{n,q}}{n^\alpha} = \beta \cdot \frac{\Gamma(\alpha + 2)}{\Gamma^2 \left( \frac{\alpha}{2} + 1 \right)} \cdot (t(1 - t))^{\alpha/2}
\]

with \( \alpha = (\sqrt{17} - 3)/2 \approx 0.561 \ldots \) and \( \beta = \text{constant factor for random partial match queries} \)

Broutin, Neininger, Sulzbach (2012, 2013): Distributional results (convergence in law to a random continuous function \( P(t) \)), variance; also for standard 2-d trees
Previous work: Fixed queries

Theorem (Duch, Jiménez & Martínez, 2012)

The cost of a partial match with rank vector \( r = (r, *, *, \ldots) \) in a random relaxed K-d tree

\[
P_{n,r} = \beta \frac{\Gamma(\alpha + 2)}{\Gamma^2 \left(\frac{\alpha}{2} + 1\right)} \cdot (r(n - r))^{\alpha/2} + o(n^{\alpha})
\]

with \( \alpha = \alpha(1/K) \), \( \alpha(\rho) = (\sqrt{9 - 8\rho} - 3)/2 \) and \( \beta = \beta(1/K) \), provided that

\[
0 < \lim_{n \to \infty} \frac{r}{n} < 1
\]
Previous work: Fixed queries

Theorem (Duch, Jiménez & Martínez, 2012)

The cost of a partial match with rank vector \( r = (r, *, *, \ldots) \) in a random relaxed \( K \)-d tree

\[
P_{n,r} = \beta \frac{\Gamma(\alpha + 2)}{\Gamma^2 \left( \frac{\alpha}{2} + 1 \right)} \cdot (r(n - r))^{\alpha/2} + o(n^\alpha)
\]

with \( \alpha = \alpha(1/K), \alpha(\rho) = (\sqrt{9 - 8\rho} - 3)/2 \) and \( \beta = \beta(1/K) \), provided that

\[
0 < \lim_{n \to \infty} \frac{r}{n} < 1
\]

A similar result holds for standard \( K \)-d trees
Results

Theorem (AofA 2014)

The expected cost of a partial match with rank vector $\mathbf{r} = (r_0, r_1, \ldots, r_{s-1}, *, *, \ldots)$ in a random relaxed $K$-d tree is

$$P_{n,\mathbf{r}} = \beta \frac{\Gamma^s(\alpha + 2)}{\Gamma^{2s}(\frac{\alpha}{2} + 1)} \cdot \prod_{i=0}^{s-1} \left( \frac{r_i}{n} \left(1 - \frac{r_i}{n}\right) \right)^{\alpha/2} \cdot n^\alpha + o(n^\alpha)$$

with $\alpha = \alpha(s/K)$, $\alpha(\rho) = (\sqrt{9 - 8\rho - 3})/2$ and $\beta = \beta(s/K)$, provided that

$$0 < \lim_{n \to \infty} \frac{r_i}{n} < 1, \quad 0 \leq i < s$$
Theorem (AofA 2014)

The expected cost of a partial match with rank vector 
\( r = (\ast, \ldots, r_0, \ast, \ldots, r_1, \ldots, r_{s-1}, \ast, \ldots) \) in a random standard K-d tree is

\[
P_{n,r} = \beta u(r) \frac{\Gamma^s(\alpha + 2)}{\Gamma^{2s}(\frac{\alpha}{2} + 1)} \cdot \prod_{i=0}^{s-1} \left( \frac{r_i}{n} \left(1 - \frac{r_i}{n}\right) \right)^{\alpha/2} \cdot n^\alpha + o(n^\alpha)
\]

with \( \alpha = \alpha(s/K), \alpha(\rho) \) the unique solution in \([0, 1]\) of

\[
(\alpha + 2)^\rho (\alpha + 1)^{1-\rho} = 2
\]

and provided that \( 0 < \lim_{n \to \infty} \frac{r_i}{n} < 1, \ 0 \leq i < s \)
Results

Corollary

The expected cost of a partial match with query $q = (q_0, q_1, \ldots, q_{s-1}, *, *, \ldots)$ in a random relaxed $K$-d tree is

$$P_{n,q} = \beta \frac{\Gamma^s(\alpha + 2)}{\Gamma^2s (\frac{\alpha}{2} + 1)} \cdot \prod_{i=0}^{s-1} (q_i (1 - q_i))^{\alpha/2} \cdot n^\alpha + o(n^\alpha)$$

with $\alpha = \alpha(s/K)$, $\alpha(\rho) = (\sqrt{9 - 8\rho} - 3)/2$ and $\beta = \beta(s/K)$, provided that $q_i \neq 0, 1$
Results

$K = 3, S = 2, n = 50000$

$X = 0$
$X = 0$
$X = 0.05$
$X = 0.05$
$X = 0.25$
$X = 0.25$
$X = 0.50$
$X = 0.50$
Overview of the proof

Let the rank vector be \( r = (r_0, \ldots, r_{s-1}) \). For \( n > 0 \)

\[
P_{n,r} = \frac{1}{K} (A_0 + \cdots + A_{s-1} + B_0 + \cdots + B_{K-s-1})
\]

with

\[
A_i = \mathbb{E} \left( \mathcal{P}_{n,r} \mid \text{root discr}=i \right)
\]

\[
B_i = \mathbb{E} \left( \mathcal{P}_{n,r} \mid \text{root discr}=i + s \right)
\]
Overview of the proof

\[ A_i = \frac{1}{n} \sum_{j=0}^{n-1} \mathbb{E} (\mathcal{P}_{n,r} | \text{root } x \text{ discr } = i \land \text{left subtree of size } j) \]

\[ = \frac{1}{n} \sum_{j=r_i}^{n-1} \mathbb{E} (\mathcal{P}_{n,r} | \text{root } x \text{ discr } = i \land \text{left subtree of size } j) \]

\[ + \frac{1}{n} \sum_{j=0}^{r_i-1} \mathbb{E} (\mathcal{P}_{n,r} | \text{root } x \text{ discr } = i \land \text{left subtree of size } j) \]

\[ = 1 + \frac{1}{n} \sum_{j=r_i}^{n-1} \sum_{r'} \pi_{r,r'} P_{j,r'} \frac{1}{n} \sum_{j=0}^{r_i-1} \sum_{r''} \pi_{r',r''} P_{n-1-j,r''} \]

with \( r' = (r'_0, \ldots, r'_{i-1}, r_i, r'_{i+1}, \ldots, r'_{s-1}) \), \( r'' = (r''_0, \ldots, r''_{i-1}, r_i - j - 1, r'_{i+1}, \ldots, r''_{s-1}) \) and \( \pi_{r,r'} \) (resp. \( \pi_{r',r''} \)) the probability that the rank vector in the left (resp. right) subtree is \( r' \) (resp. \( r'' \))
Overview of the proof
Overview of the proof

What’s the prob. there are exactly $r'_k$ elements here? (*)
Overview of the proof

\[
\binom{j}{r_k'} \binom{n-1-j}{r_k - r_k'} \frac{(n-1)!}{r_k}\]

\[
\binom{n-1}{r_k}
\]
Overview of the proof
Overview of the proof

What's the prob. there are exactly $r'_k$ elements here? (*)
Overview of the proof

\[
\binom{j}{r_k'} \cdot \left( \binom{n-1-j}{r_k - r_k' - 1} \right)
\]

\[
\binom{n-1}{r_k - 1}
\]
Overview of the proof

\[ A_i \sim 1 + \frac{1}{n} \sum_{j=r_i}^{n-1} P_{j, r^{(1)}} + \frac{1}{n} \sum_{j=0}^{r_i-1} P_{n-1-j, r^{(2)}} \]

with

\[ r^{(1)} = (r_0^{(1)}, \ldots, r_{i-1}^{(1)}, r_i, r_{i+1}, \ldots, r_{s-1}^{(1)}) \]
\[ r^{(2)} = (r_0^{(2)}, \ldots, r_{i-1}^{(2)}, r_i - j, r_{i+1}, \ldots, r_{s-1}^{(2)}) \]

and for \( k \neq i \)

\[ r_k^{(1)} = \frac{j}{n} r_k, \quad r_k^{(2)} = \frac{n-1-j}{n} r_k \]

Terms \( B_i \) are similarly handled.
Overview of the proof

Dividing by $n^{\alpha}$ ($\alpha := \alpha(s/K)$), and collecting all terms

$$\frac{P_{n,r}}{n^{\alpha}} \sim \frac{1}{n^{\alpha}} + \frac{1}{K} \sum_{i=0}^{s-1} \left( \frac{1}{n} \sum_{j=r_i}^{n-1} \frac{P_{j, \bar{r}^{(1)}}}{j^{\alpha}} \frac{j^{\alpha}}{n^{\alpha}} \right)$$

$$+ \frac{1}{n} \sum_{j=0}^{r_i-1} \frac{P_{n-1-j, \bar{r}^{(2)}}}{(n-1-j)^{\alpha}} \frac{(n-1-j)^{\alpha}}{n^{\alpha}}$$

$$+ \frac{K-s}{K} \frac{1}{n} \sum_{j=0}^{n-1} \frac{P_{j, \bar{r}^{(3)}}}{j^{\alpha}} \frac{j^{\alpha}}{n^{\alpha}} + \frac{P_{n-1-j, \bar{r}^{(4)}}}{(n-1-j)^{\alpha}} \frac{(n-1-j)^{\alpha}}{n^{\alpha}}$$
Overview of the proof

We anticipate $P_{n,r}/n^\alpha \sim f(z_0, \ldots, z_{s-1})$ with $z_i = r_i/n$ and thus, as $n \to \infty$

$$f(z) \sim \frac{1}{K} \sum_{i=0}^{s-1} \left\{ \frac{1}{n} \sum_{j=r_i}^{n-1} \left( \frac{j}{n} \right)^\alpha \cdot f \left( z_0, \ldots, z_{\frac{n}{j}}, \ldots \right) ight.$$  

$$+ \frac{1}{n} \sum_{j=0}^{r_i-1} \left( \frac{n-1-j}{n} \right)^\alpha \cdot f \left( z_0, \ldots, \left( z_i - \frac{j+1}{n} \right) \frac{n}{n-1-j}, \ldots \right) \right\}$$  

$$+ \frac{K-s}{K} \frac{1}{n} \sum_{j=0}^{n-1} \left( \left( \frac{j}{n} \right)^\alpha + \left( \frac{n-1-j}{n} \right)^\alpha \right) \cdot f(z_0, \ldots, z_{s-1})$$
Overview of the proof

Passing to the limit and substituting sums by integrals

\[ f(z_0, \ldots, z_{s-1}) \sim \frac{1}{K} \sum_{i=0}^{s-1} \left\{ \int_{z_i}^{1} x^\alpha f \left( z_0, \ldots, \frac{z_i}{x}, \ldots, z_{s-1} \right) \, dx \right. \]

\[ + \int_{0}^{z_i} (1 - x)^\alpha f \left( z_0, \ldots, \frac{z_i - x}{1 - x}, \ldots, z_{s-1} \right) \, dx \right\} \]

\[ + \frac{K - s}{K} \int_{0}^{1} f(z_0, \ldots, z_{s-1}) (x^\alpha + (1 - x)^\alpha) \, dx \]
Overview of the proof

After some “massaging”

\[
\alpha + 2 \sum_{i=0}^{s-1} \int_{z_i}^{1} x^\alpha f \left( z_0, \ldots, \frac{z_i}{x}, \ldots, z_{s-1} \right) \, dx
\]

\[
+ \int_{0}^{z_i} (1-x)^\alpha f \left( z_0, \ldots, \frac{z_i - x}{1-x}, \ldots, z_{s-1} \right) \, dx
\]
Overview of the proof

- In order to solve the integral equation, we assume
  \[ f(z_0, \ldots, z_{s-1}) = \vartheta_0(z_0) \cdots \vartheta_{s-1}(z_{s-1}) \]

- Because of the symmetry the problem we can safely assume \( \vartheta = \vartheta_0 = \vartheta_1 = \cdots = \vartheta_{s-1} \) and \( \vartheta(z) = \vartheta(1 - z) \)

- Furthermore, the analysis of extremal queries (when for some \( i, r_i = o(n) \) or \( r_i = n - o(n) \)), shows that \( P_{n,r} = o(n^\alpha) \) in those cases and thus

  \[ \lim_{z \to 0} \vartheta(z) = 0 \]
Overview of the proof

- The integral equation can be transformed into a second-order linear differential equation for $\vartheta$, which can be easily solved

$$\vartheta(z) = \kappa \cdot (z(1 - z))^{\alpha/2}$$

for some constant $\kappa$.

- Last but not least, the constant factor of $f$ can be determined using

$$\beta = \int_0^1 \int_0^1 \cdots \int_0^1 f(z_0, \ldots, z_{s-1}) \, dz_0 \, dz_1 \cdots \, dz_{s-1}$$
Concluding remarks

- We conjecture that the expected cost of partial matches follows the same pattern in many multidimensional data structures

\[
P_{n,q} = \beta \frac{\Gamma^s(\alpha + 2)}{\Gamma^{2s}(\frac{\alpha}{2} + 1)} \cdot \prod_{i=0}^{s-1} (q_i (1 - q_i))^{\alpha/2} \cdot n^\alpha + o(n^\alpha)
\]

with \(\beta\) and \(\alpha\) derived from the analysis of random queries

- Moreover we conjecture the convergence (at least in distribution), for the sequence of r.v.'s \(\{n^{-\alpha}P_{n,q}\}_{n \geq 0}\)

\[
\frac{P_{n,q}}{n^\alpha} \xrightarrow{(d)} P(q)
\]

for a continuous random function \(P(q)\) (in \(s\) variables)
Concluding remarks

- We conjecture that the expected cost of partial matches follows the same pattern in many multidimensional data structures

\[ P_{n,q} = \beta \frac{\Gamma^s(\alpha + 2)}{\Gamma^{2s}(\frac{\alpha}{2} + 1)} \cdot \prod_{i=0}^{s-1} (q_i (1 - q_i))^{\alpha/2} \cdot n^\alpha + o(n^\alpha) \]

with \( \beta \) and \( \alpha \) derived from the analysis of random queries

- Moreover we conjecture the convergence (at least in distribution), for the sequence of r.v.'s \( \{n^{-\alpha}P_{n,q}\}_{n \geq 0} \)

\[ \frac{P_{n,q}}{n^\alpha} \xrightarrow{(d)} P(q) \]

for a continuous random function \( P(q) \) (in s variables)
Concluding remarks

- Our on-going work is now focused in extending our analysis to other data structures and in finding a general argument which proves the “universality” of the factor

\[ \beta_u \cdot \frac{\Gamma^s(\alpha + 2)}{\Gamma^{2s}(\frac{\alpha}{2} + 1)} \cdot \prod_{i=0}^{s-1} (q_i (1 - q_i))^{\alpha/2} \]

- Partial match can be regarded as a generalization to higher dimensions of the selection of order statistics; we have heavily relied on techniques which have proven extremely useful in the analysis of classical quickselect and variants.
Concluding remarks

Our on-going work is now focused in extending our analysis to other data structures and in finding a general argument which proves the “universality” of the factor

$$\beta_u \cdot \frac{\Gamma^s(\alpha + 2)}{\Gamma^{2s}(\frac{\alpha}{2} + 1)} \cdot \prod_{i=0}^{s-1} (q_i (1 - q_i))^{\alpha/2}$$

Partial match can be regarded as a generalization to higher dimensions of the selection of order statistics; we have heavily relied on techniques which have proven extremely useful in the analysis of classical quickselect and variants.
Thanks for your attention!

See you in Strobl (Austria) next June 2015