Convergence Rates in the Probabilistic Analysis of Algorithms

Jasmin Straub
(joint work with Ralph Neininger)

Institute for Mathematics
Johann Wolfgang Goethe University
Frankfurt am Main

AofA 2020
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10
A toy example

Binary search tree built from a random permutation of \{1, \ldots, n\}.
6, 7, 2, 5, 3, 9, 8, 1, 4, 10

6
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10
A toy example

Binary search tree built from a random permutation of \{1, \ldots, n\}.

6, 7, 2, 5, 3, 9, 8, 1, 4, 10
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10

\begin{itemize}
  \item \( \mathcal{L}_0(n) \) number of nodes with no left descendant,
  \item \( \mathcal{L}_1(n) \) number of nodes with exactly one left descendant.
\end{itemize}
A toy example

Binary search tree built from a random permutation of \{1, \ldots, n\}.

6, 7, 2, 5, 3, 9, 8, 1, 4, 10
A toy example

Binary search tree built from a random permutation of \(\{1, \ldots, n\}\).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10
A toy example

Binary search tree built from a random permutation of \(\{1, \ldots, n\}\).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10
A toy example

Binary search tree built from a random permutation of \{1, \ldots, n\}.

6, 7, 2, 5, 3, 9, 8, 1, 4, 10

\[ L_0n = \text{number of nodes with no left descendant}, \]
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10

\( L_{0n} = \) number of nodes with no left descendant,
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10

\[ L_0n = \text{number of nodes with no left descendant}, \]
\[ L_{1n} = \text{number of nodes with exactly one left descendant}. \]
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10

\[ L_{0n} = \text{number of nodes with no left descendant}, \]
\[ L_{1n} = \text{number of nodes with exactly one left descendant}. \]
A toy example

Binary search tree built from a random permutation of \{1, \ldots, n\}.

6, 7, 2, 5, 3, 9, 8, 1, 4, 10

$L_{0n} = \text{number of nodes with no left descendant},$
$L_{1n} = \text{number of nodes with exactly one left descendant}.$
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).
6, 7, 2, 5, 3, 9, 8, 1, 4, 10

\( L_{0n} \) = number of nodes with no left descendant,
\( L_{1n} \) = number of nodes with exactly one left descendant.
A toy example

Binary search tree built from a random permutation of \( \{1, \ldots, n\} \).

6, 7, 2, 5, 3, 9, 8, 1, 4, 10

\( L_{0n} = \) number of nodes with no left descendant,
\( L_{1n} = \) number of nodes with exactly one left descendant.
A toy example

Binary search tree built from a random permutation of \{1, \ldots, n\}.

6, 7, 2, 5, 3, 9, 8, 1, 4, 10

$L_{0n} = \text{number of nodes with no left descendant},$
$L_{1n} = \text{number of nodes with exactly one left descendant}.$
A toy example

BST built from a random permutation of \{1, \ldots, n\}.

\(L_{0n} = \# \) nodes with no left descendant,
\(L_{1n} = \# \) nodes with exactly one left descendant.
A toy example

BST built from a random permutation of \( \{1, \ldots, n\} \).

\[ L_0^n = \# \text{ nodes with no left descendant}, \]
\[ L_1^n = \# \text{ nodes with exactly one left descendant}. \]

For \( Y_n := (L_0^n, L_1^n) \),
A toy example

BST built from a random permutation of \( \{1, \ldots, n\} \).

\[ L_{0n} = \# \text{ nodes with no left descendant}, \]
\[ L_{1n} = \# \text{ nodes with exactly one left descendant}. \]

For \( Y_n := (L_{0n}, L_{1n}) \), we have \( Y_0 = (0, 0) \) and

\[ Y_n \overset{d}{=} Y_{l_1^{(n)}}^{(1)} + Y_{l_2^{(n)}}^{(2)} + b_n, \quad n \geq 1, \]

with \( l_1^{(n)} \sim \text{unif}\{0, \ldots, n-1\} \), \( l_2^{(n)} = n - 1 - l_1^{(n)} \) and \( b_n = \begin{pmatrix} 1_{\{l_1^{(n)}=0\}} \\ 1_{\{l_1^{(n)}=1\}} \end{pmatrix} \).
A toy example

BST built from a random permutation of \{1, \ldots, n\}.

\[ L_{0n} = \# \text{ nodes with no left descendant,} \]

\[ L_{1n} = \# \text{ nodes with exactly one left descendant.} \]

For \( Y_n := (L_{0n}, L_{1n}) \), we have \( Y_0 = (0, 0) \) and

\[ Y_n \overset{d}{=} Y_{l_1^{(n)}} + Y_{l_2^{(n)}} + b_n, \quad n \geq 1, \]

with \( l_1^{(n)} \sim \text{unif}\{0, \ldots, n-1\} \), \( l_2^{(n)} = n - 1 - l_1^{(n)} \) and \( b_n = \begin{pmatrix} 1_{\{l_1^{(n)}=0\}} \\ 1_{\{l_1^{(n)}=1\}} \end{pmatrix} \).
A toy example

BST built from a random permutation of \( \{1, \ldots, n\} \).

\[ L_0^n = \# \text{ nodes with no left descendant}, \]
\[ L_1^n = \# \text{ nodes with exactly one left descendant}. \]

For \( Y_n := (L_0^n, L_1^n) \), we have \( Y_0 = (0, 0) \) and

\[ Y_n \overset{d}{=} Y_{l_1^{(n)}}^{(1)} + Y_{l_2^{(n)}}^{(2)} + b_n, \quad n \geq 1, \]

with \( l_1^{(n)} \sim \text{unif}\{0, \ldots, n-1\} \), \( l_2^{(n)} = n - 1 - l_1^{(n)} \) and

\[ b_n = \begin{pmatrix} 1_{\{l_1^{(n)}=0\}} \\ 1_{\{l_1^{(n)}=1\}} \end{pmatrix}. \]
A toy example

BST built from a random permutation of \{1, \ldots, n\}.

\[ L_{0n} = \# \text{ nodes with no left descendant}, \]
\[ L_{1n} = \# \text{ nodes with exactly one left descendant}. \]

For \( Y_n := (L_{0n}, L_{1n}) \), we have \( Y_0 = (0, 0) \) and
\[
Y_n \overset{d}{=} Y_{I_1^{(n)}}^{(1)} + Y_{I_2^{(n)}}^{(2)} + b_n, \quad n \geq 1,
\]
with \( I_1^{(n)} \sim \text{unif}\{0, \ldots, n-1\} \), \( I_2^{(n)} = n - 1 - I_1^{(n)} \) and \( b_n = \begin{pmatrix} 1_{\{I_1^{(n)}=0\}} \\ 1_{\{I_1^{(n)}=1\}} \end{pmatrix} \).
A toy example

BST built from a random permutation of \( \{1, \ldots, n\} \).

\( L_{0n} = \# \) nodes with no left descendant,
\( L_{1n} = \# \) nodes with exactly one left descendant.

For \( Y_n := (L_{0n}, L_{1n}) \), we have \( Y_0 = (0, 0) \) and

\[
Y_n \overset{d}{=} Y_{l_1^{(n)}}^{(1)} + Y_{l_2^{(n)}}^{(2)} + b_n, \quad n \geq 1,
\]

with \( l_1^{(n)} \sim \text{unif}\{0, \ldots, n-1\} \), \( l_2^{(n)} = n - 1 - l_1^{(n)} \) and \( b_n = \begin{pmatrix} 1_{\{l_1^{(n)}=0\}} \\ 1_{\{l_1^{(n)}=1\}} \end{pmatrix} \).
A toy example

BST built from a random permutation of \( \{1, \ldots, n\} \).

\( L_{0n} = \# \) nodes with no left descendant,

\( L_{1n} = \# \) nodes with exactly one left descendant.

For \( Y_n := (L_{0n}, L_{1n}) \), we have \( Y_0 = (0, 0) \) and

\[
Y_n \overset{d}{=} Y_{l_1^{(n)}}^{(1)} + Y_{l_2^{(n)}}^{(2)} + b_n, \quad n \geq 1,
\]

with \( l_1^{(n)} \sim \text{unif}\{0, \ldots, n-1\} \), \( l_2^{(n)} = n - 1 - l_1^{(n)} \) and \( b_n = \begin{pmatrix} 1 \{l_1^{(n)}=0\} \\ 1 \{l_1^{(n)}=1\} \end{pmatrix} \).

For \( n \geq 4 \) (cf. Devroye 1991),

\[
\mathbb{E}[Y_n] = (n + 1) \begin{pmatrix} 1/2 \\ 1/6 \end{pmatrix}, \quad \text{Cov}(Y_n) = (n + 1) \frac{1}{360} \begin{pmatrix} 30 & -15 \\ -15 & 28 \end{pmatrix}.
\]
A toy example

BST built from a random permutation of \( \{1, \ldots, n\} \).

\[ L_{0n} = \# \text{ nodes with no left descendant}, \]
\[ L_{1n} = \# \text{ nodes with exactly one left descendant}. \]

For \( Y_n := (L_{0n}, L_{1n}) \), we have \( Y_0 = (0, 0) \) and

\[ Y_n \overset{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \quad n \geq 1, \]

with \( I_1^{(n)} \sim \text{unif}\{0, \ldots, n-1\} \), \( I_2^{(n)} = n - 1 - I_1^{(n)} \) and \( b_n = \begin{pmatrix} 1 \{I_1^{(n)}=0\} \\ 1 \{I_1^{(n)}=1\} \end{pmatrix} \).

For \( n \geq 4 \) (cf. Devroye 1991),

\[ \mathbb{E}[Y_n] = (n + 1) \begin{pmatrix} 1/2 \\ 1/6 \end{pmatrix}, \quad \text{Cov}(Y_n) = (n + 1) \frac{1}{360} \begin{pmatrix} 30 & -15 \\ -15 & 28 \end{pmatrix}. \]

and we have asymptotic normality \( (n \to \infty) \):

\[ \text{Cov}(Y_n)^{-1/2}(Y_n - \mathbb{E}[Y_n]) \overset{d}{\to} \mathcal{N}(0, \text{Id}_2). \]
General recursion

recursion before:

\[ Y_n = d \ Y_{l_1^{(n)}} + Y_{l_2^{(n)}} + b_n, \quad n \geq 1. \]

general recursion:

\[ Y_n = \sum_{r=1}^{K} A_r(n) Y_{l_r^{(n)}} + b_n, \quad n \geq n_0, \]
General recursion

recursion before:

\[ Y_n \overset{d}{=} Y_{l_1}^{(1)} + Y_{l_2}^{(2)} + b_n, \quad n \geq 1. \]

general recursion:

\[ Y_n \overset{d}{=} \sum_{r=1}^{K} A_r(n) Y_{l_r}^{(r)} + b_n, \quad n \geq n_0, \]

▶ \( Y_n \) random vector in \( \mathbb{R}^d \),
General recursion

recursion before:

\[ Y_n = Y_{l_1}^{(1)} + Y_{l_2}^{(2)} + b_n, \quad n \geq 1. \]

general recursion:

\[ Y_n \overset{d}{=} \sum_{r=1}^{K} A_r(n) Y_{l_r}^{(r)} + b_n, \quad n \geq n_0, \]

- \( Y_n \) random vector in \( \mathbb{R}^d \),
- \( K \) number of subproblems,
General recursion

recursion before:

\[ Y_n = Y_{l_1}^{(1)} + Y_{l_2}^{(2)} + b_n, \quad n \geq 1. \]

general recursion:

\[ Y_n \equiv \sum_{r=1}^{K} A_r(n) Y_{l_r}^{(r)} + b_n, \quad n \geq n_0, \]

- \( Y_n \) random vector in \( \mathbb{R}^d \),
- \( K \) number of subproblems,
- \( l^{(n)} = (l_1^{(n)}, \ldots, l_K^{(n)}) \in \{0, \ldots, n\}^K \) sizes of subproblems,
General recursion

recursion before:

\[ Y_n \overset{d}{=} Y_{l_1(n)}^{(1)} + Y_{l_2(n)}^{(2)} + b_n, \quad n \geq 1. \]

general recursion:

\[ Y_n \overset{d}{=} \sum_{r=1}^{K} A_r(n) Y_{l_r(n)}^{(r)} + b_n, \quad n \geq n_0, \]

▷ \( Y_n \) random vector in \( \mathbb{R}^d \),
▷ \( K \) number of subproblems,
▷ \( l^{(n)} = (l_1^{(n)}, \ldots, l_K^{(n)}) \in \{0, \ldots, n\}^K \) sizes of subproblems,
▷ \( A_1(n), \ldots, A_K(n) \) random \((d \times d)\)-matrices,
General recursion

recursion before:

\[ Y_n \overset{d}{=} Y_{I_1^{(1)}}^{(1)} + Y_{I_2^{(1)}}^{(2)} + b_n, \quad n \geq 1. \]

general recursion:

\[ Y_n \overset{d}{=} \sum_{r=1}^{K} A_r(n) Y_{I_r^{(n)}}^{(r)} + b_n, \quad n \geq n_0, \]

- \( Y_n \) random vector in \( \mathbb{R}^d \),
- \( K \) number of subproblems,
- \( I^{(n)} = (I_1^{(n)}, \ldots, I_K^{(n)}) \in \{0, \ldots, n\}^K \) sizes of subproblems,
- \( A_1(n), \ldots, A_K(n) \) random \( (d \times d) \)-matrices,
- \( b_n \) random vector in \( \mathbb{R}^d \),
General recursion

Recursion before:

\[ Y_n \overset{d}{=} Y_{l_1^{(1)}}^{(1)} + Y_{l_2^{(2)}}^{(2)} + b_n, \quad n \geq 1. \]

general recursion:

\[ Y_n \overset{d}{=} \sum_{r=1}^{K} A_r(n) Y_{l_r^{(r)}}^{(r)} + b_n, \quad n \geq n_0, \]

- \( Y_n \) random vector in \( \mathbb{R}^d \),
- \( K \) number of subproblems,
- \( I^{(n)} = (I_1^{(n)}, \ldots, I_K^{(n)}) \in \{0, \ldots, n\}^K \) sizes of subproblems,
- \( A_1(n), \ldots, A_K(n) \) random \( (d \times d) \)-matrices,
- \( b_n \) random vector in \( \mathbb{R}^d \),
- \( (Y_{n}^{(r)})_{n \geq 0} \overset{d}{=} (Y_n)_{n \geq 0} \) for \( r = 1, \ldots, K \),
General recursion

recursion before:

\[ Y_n \overset{d}{=} Y_{l_1^{(n)}}^{(1)} + Y_{l_2^{(n)}}^{(2)} + b_n, \quad n \geq 1. \]

general recursion:

\[ Y_n \overset{d}{=} \sum_{r=1}^{K} A_r(n) Y_{l_r^{(n)}}^{(r)} + b_n, \quad n \geq n_0, \]

- \( Y_n \) random vector in \( \mathbb{R}^d \),
- \( K \) number of subproblems,
- \( l^{(n)} = (l_1^{(n)}, \ldots, l_K^{(n)}) \in \{0, \ldots, n\}^K \) sizes of subproblems,
- \( A_1(n), \ldots, A_K(n) \) random \((d \times d)\)-matrices,
- \( b_n \) random vector in \( \mathbb{R}^d \),
- \( (Y_{l_r^{(n)}}^{(r)})_{n \geq 0} \overset{d}{=} (Y_n)_{n \geq 0} \) for \( r = 1, \ldots, K \),
- \( (A_1(n), \ldots, A_K(n), b_n, l^{(n)}), (Y_{l_1^{(n)}}^{(1)})_{n \geq 0}, \ldots, (Y_{l_K^{(n)}}^{(K)})_{n \geq 0} \) are independent.
The contraction method

General recursion:

\[ Y_n \overset{d}{=} \sum_{r=1}^{K} A_r(n) Y_{I_r(n)}^{(r)} + b_n, \quad n \geq n_0. \]
The contraction method

General recursion:

\[ Y_n \overset{d}{=} \sum_{r=1}^{K} A_r(n) Y_{i_r}^{(r)} + b_n, \quad n \geq n_0. \]

We define the normalized sequence \((X_n)_{n \geq 0}\) by

\[ X_n := C_n^{-1/2}(Y_n - M_n), \quad n \geq 0, \]

where \(M_n\) is a \(d\)-dimensional vector and \(C_n\) a positive definite \((d \times d)\)-matrix.
The contraction method

General recursion:

\[ Y_n = \sum_{r=1}^{K} A_r(n) Y_{I_r(n)}^{(r)} + b_n, \quad n \geq n_0. \]

We define the normalized sequence \((X_n)_{n \geq 0}\) by

\[ X_n := C_n^{-1/2} (Y_n - M_n), \quad n \geq 0, \]

where \(M_n\) is a \(d\)-dimensional vector and \(C_n\) a positive definite \((d \times d)\)-matrix. The normalized quantities satisfy the following modified recursion:

\[ X_n = \sum_{r=1}^{K} A_r^{(n)} X_{I_r^{(n)}}^{(r)} + b^{(n)}, \quad n \geq n_0, \]

with

\[ A_r^{(n)} := C_n^{-1/2} A_r(n) C_{I_r^{(n)}}^{1/2}, \quad b^{(n)} := C_n^{-1/2} \left( b_n - M_n + \sum_{r=1}^{K} A_r(n) M_{I_r^{(n)}} \right) \]

and independence relations as before.
The contraction method

Modified recursion \((n \geq n_0)\):

\[
X_n^d = \sum_{r=1}^{K} A_r^{(n)} X_r^{(r)} + b^{(n)}.
\]
The contraction method

Modified recursion ($n \geq n_0$):

$$X_n = \sum_{r=1}^{K} A_r^{(n)} X_r^{(r)} + b^{(n)}.$$
The contraction method

Modified recursion \((n \geq n_0)\):

\[
X_n \overset{d}{=} \sum_{r=1}^{K} A_r^{(n)} X^{(r)}_{l_r^{(n)}} + b^{(n)}.
\]

\(\rightarrow A_r^* \quad \rightarrow b^*
\)
The contraction method

Modified recursion \((n \geq n_0)\):

\[
X_n \overset{d}{=} \sum_{r=1}^{K} \left( A_r^{(n)} \right) X^{(r)}_t + b^{(n)}.
\]

Limit equation \((n \to \infty)\):

\[
X \overset{d}{=} \sum_{r=1}^{K} A_r^* X^{(r)} + b^*
\]

with \((A_1^*, \ldots, A_K^*, b^*)\), \(X^{(1)}, \ldots, X^{(K)}\) independent and \(X^{(r)} \overset{d}{=} X\) for \(r = 1, \ldots, K\).
The contraction method

Modified recursion \((n \geq n_0)\):
\[
X_n \overset{d}{=} \sum_{r=1}^{K} \left[ A_r^{(n)} X_r^{(r)} + b^{(n)} \right] \rightarrow A_r^* \rightarrow b^*
\]

Limit equation \((n \rightarrow \infty)\):
\[
X \overset{d}{=} \sum_{r=1}^{K} A_r^* X^{(r)} + b^*
\]

with \((A_1^*, \ldots, A_K^*, b^*)\), \(X^{(1)}, \ldots, X^{(K)}\) independent and \(X^{(r)} \overset{d}{=} X\) for \(r = 1, \ldots, K\).

Idea:
\[
A_r^{(n)} \rightarrow A_r^* \\
b^{(n)} \rightarrow b^* \\
\implies X_n \rightarrow X
\]
The contraction method

Modified recursion \((n \geq n_0)\):

\[
X_n \overset{d}{=} \sum_{r=1}^{K} \left( A_r^{(n)} X^{(r)}_{l_r(n)} + b^{(n)} \right) \rightarrow A_r^* X^{(r)} + b^*
\]

Limit equation \((n \to \infty)\):

\[
X \overset{d}{=} \sum_{r=1}^{K} A_r^* X^{(r)} + b^*
\]

with \((A_1^*, \ldots, A_K^*, b^*)\), \(X^{(1)}, \ldots, X^{(K)}\) independent and \(X^{(r)} \overset{d}{=} X\) for \(r = 1, \ldots, K\).

Idea:

\[
A_r^{(n)} \to A_r^* \quad b^{(n)} \to b^* \quad \implies \quad X_n \to X
\]

Now:

\[
\sum_{r=1}^{K} \| A_r^{(n)} - A_r^* \|_s + \| b^{(n)} - b^* \|_s = O(R(n)) \quad \implies \quad \zeta_s(X_n, X) = O(R(n))
\]
Convergence result

A general transfer theorem

Let \((X_n)_{n \geq 0}\) be given as before. We assume that there exists some monotonically decreasing sequence \(R(n) \downarrow 0\) such that, as \(n \to \infty\),

\[
\|b^{(n)} - b^*\|_s + \sum_{r=1}^{K} \|A_r^{(n)} - A_r^*\|_s + \sum_{r=1}^{K} 1_{\{I_r^{(n)} < \ell\}} A_r^{(n)} = O(R(n))
\]

for all \(\ell \in \mathbb{N}\). If the technical condition \(\|1_{\{I_r^{(n)} = n\}} A_r^{(n)}\|_s \to 0\) is satisfied for \(r = 1, \ldots, K\) and if

\[
\limsup_{n \to \infty} E \sum_{r=1}^{K} \left( \frac{R(I_r^{(n)})}{R(n)} \|A_r^{(n)}\|_s^{op} \right) < 1,
\]

then we have, as \(n \to \infty\),

\[
\zeta_s(X_n, X) = O(R(n)),
\]

where \(X\) is the unique solution of the limit equation in \(\mathcal{M}_d^s(0, \text{Id}_d)\).
A refined version for the normal case

Let \((X_n)_{n \geq 0}\) be given as before. We assume that the limit coefficients satisfy \(b^* = 0\) and \(\sum_{r=1}^{K} A_r^* (A_r^*)^T = \text{Id}_d\) almost surely and that there exists some sequence \(R(n) \downarrow 0\) such that, as \(n \to \infty\),

\[
\|b^{(n)}\|_3^3 + \left\| \sum_{r=1}^{K} A_r^{(n)} (A_r^{(n)})^T - \text{Id}_d \right\|_{3/2}^{3/2} + \sum_{r=1}^{K} \|1_{\{I_r^{(n)}<\ell\}} A_r^{(n)}\|_3^3 = O(R(n))
\]

for all \(\ell \in \mathbb{N}\). If the technical condition \(\|1_{\{I_r^{(n)}=n\}} A_r^{(n)}\|_3 \to 0\) is satisfied for \(r = 1, \ldots, K\) and if

\[
\limsup_{n \to \infty} \mathbb{E} \sum_{r=1}^{K} \left( \frac{R(I_r^{(n)})}{R(n)} \|A_r^{(n)}\|_{op}^3 \right) < 1,
\]

then we have, as \(n \to \infty\),

\[
\zeta_3(X_n, \mathcal{N}(0, \text{Id}_d)) = O(R(n)).
\]
Applications

geometric problems
  ▶ maxima in right triangles

quantities of divide-and-conquer algorithms
  ▶ Quicksort
  ▶ Quickselect

parameters of random trees
  ▶ size of random $m$-ary search trees
  ▶ number of leaves in quad trees
  ▶ size of random tries
  ▶ path lengths in digital search trees, tries and PATRICIA tries
Applications

**geometric problems**
- maxima in right triangles

**quantities of divide-and-conquer algorithms**
- Quicksort
- Quickselect

**parameters of random trees**
- size of random $m$-ary search trees
- number of leaves in quad trees
- size of random tries
- path lengths in digital search trees, tries and PATRICIA tries
Applications

Maxima in right triangles

\( Y_n = \) number of maxima in a random sample of \( n \) points chosen uniformly and independently in the right triangle with corners \((0, 0), (0, 1) \& (1, 0)\)

Bai, Hwang, and Tsai (2003)

Denoting by \( \Phi \) the standard normal distribution function, we have, as \( n \to \infty \),

\[
\sup_{x \in \mathbb{R}} \left| \mathbb{P} \left( \frac{Y_n - \mathbb{E}[Y_n]}{\sqrt{\text{Var}(Y_n)}} < x \right) - \Phi(x) \right| = O \left( n^{-1/4} \right).
\]

With our theorem

We have, as \( n \to \infty \),

\[
\zeta_3 \left( \frac{Y_n - \mathbb{E}[Y_n]}{\sqrt{\text{Var}(Y_n)}}, \mathcal{N}(0, 1) \right) = O \left( n^{-1/4} \right).
\]
Applications

geometric problems
  ▶ maxima in right triangles

quantities of divide-and-conquer algorithms
  ▶ Quicksort
  ▶ Quickselect

parameters of random trees
  ▶ size of random $m$-ary search trees
  ▶ number of leaves in quad trees
  ▶ size of random tries
  ▶ path lengths in digital search trees, tries and PATRICIA tries
Applications

dedometric problems
  ▶ maxima in right triangles

quantities of divide-and-conquer algorithms
  ▶ Quicksort
  ▶ Quickselect

parameters of random trees
  ▶ size of random $m$-ary search trees
  ▶ number of leaves in quad trees
  ▶ size of random tries
  ▶ path lengths in digital search trees, tries and PATRICIA tries
Applications

Quicksort

\[ Y_n = \text{number of key comparisons needed by Quicksort to sort } n \text{ randomly permuted distinct numbers} \]

Neininger and Rüschendorf (2002)

We have, as \( n \to \infty \),

\[
\zeta_3 \left( \frac{Y_n - \mathbb{E}[Y_n]}{\sqrt{\text{Var}(Y_n)}}, X \right) = \Theta \left( \frac{\log n}{n} \right),
\]

where the limit \( X \) is given as the unique solution of a distributional fixed-point equation.

With our theorem

We have, for \( 2 < s \leq 3 \) and as \( n \to \infty \),

\[
\zeta_s \left( \frac{Y_n - \mathbb{E}[Y_n]}{\sqrt{\text{Var}(Y_n)}}, X \right) = O \left( \frac{\log n}{n} \right).
\]
Applications

geometric problems
  ▶ maxima in right triangles

quantities of divide-and-conquer algorithms
  ▶ Quicksort
  ▶ Quickselect

parameters of random trees
  ▶ size of random $m$-ary search trees
  ▶ number of leaves in quad trees
  ▶ size of random tries
  ▶ path lengths in digital search trees, tries and PATRICIA tries
Applications

geometric problems
  ▶ maxima in right triangles

quantities of divide-and-conquer algorithms
  ▶ Quicksort
  ▶ Quickselect

parameters of random trees
  ▶ size of random $m$-ary search trees
  ▶ number of leaves in quad trees
  ▶ size of random tries
  ▶ path lengths in digital search trees, tries and PATRICIA tries
Applications

Size of $m$-ary search trees

$Y_n = \text{number of (internal) nodes of an } m\text{-ary search tree constructed from a random permutation of } \{1, \ldots, n\}$

Hwang (2003)

As $n \to \infty$, we have

$$\sup_{x \in \mathbb{R}} \left| \mathbb{P} \left( \frac{Y_n - \mathbb{E}[Y_n]}{\sqrt{\text{Var}(Y_n)}} < x \right) - \Phi(x) \right| = \begin{cases} O(n^{-1/2}), & 3 \leq m \leq 19, \\ O(n^{-3(3/2-\alpha)}), & 20 \leq m \leq 26. \end{cases}$$

<table>
<thead>
<tr>
<th>$m$</th>
<th>$3 \ldots 19$</th>
<th>$20 \ldots 26$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3(3/2 - \alpha)$</td>
<td>$&gt; 1/2$</td>
<td>$0.45 \ldots 0.002$</td>
</tr>
</tbody>
</table>

With our theorem

For any $\varepsilon > 0$, we have as $n \to \infty$

$$\zeta_3 \left( \frac{Y_n - \mathbb{E}[Y_n]}{\sqrt{\text{Var}(Y_n)}}, \mathcal{N}(0, 1) \right) = \begin{cases} O(n^{-1/2+\varepsilon}), & 3 \leq m \leq 19, \\ O(n^{-3(3/2-\alpha)}), & 20 \leq m \leq 26. \end{cases}$$
Applications

BST built from a random permutation of \{1, \ldots, n\}.

\[ L_{0n} = \# \text{ nodes with no left descendant}, \]
\[ L_{1n} = \# \text{ nodes with exactly one left descendant}. \]

Theorem

Denoting by \( Y_n := (L_{0n}, L_{1n}) \) the vector of the numbers of nodes with no and with exactly one left descendant respectively in a random binary search tree with \( n \) nodes we have, as \( n \to \infty \), that

\[
\zeta_3 \left( \text{Cov}(Y_n)^{-1/2} (Y_n - \mathbb{E}[Y_n]), \mathcal{N}(0, \text{Id}_2) \right) = O(n^{-1/2}).
\]
Theorem

Let \((Y_n)_{n \geq 0}\) be 3-integrable and satisfy

\[ Y_n \overset{d}{=} Y^{(1)}_n + Y^{(2)}_n + b_n, \quad n \geq n_0, \]

with \(I^{(n)}_1 \sim \text{Bin}(n, \frac{1}{2})\), \(I^{(n)}_2 = n - I^{(n)}_1\) and \(\|b_n\|_3 = O(1)\). We assume that, as \(n \to \infty\), we have

\[ \mathbb{E}[Y_n] = nP_1(\log_2 n) + O(1), \]
\[ \text{Var}(Y_n) = nP_2(\log_2 n) + O(1), \]

for some smooth and 1-periodic functions \(P_1, P_2\) with \(P_2 > 0\). Then, for any \(\varepsilon > 0\) and \(n \to \infty\), we have

\[ \zeta_3 \left( \frac{Y_n - \mathbb{E}[Y_n]}{\sqrt{\text{Var}(Y_n)}}, \mathcal{N}(0, 1) \right) = O(n^{-1/2+\varepsilon}). \]
Thank you!