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

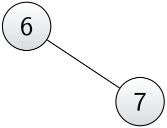
Binary search tree built from a random permutation of $\{1, \ldots, n\}$.

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

A toy example

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

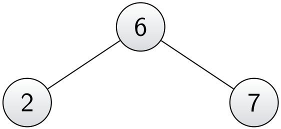
$6, 7, 2, 5, 3, 9, 8, 1, 4, 10$



A toy example

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

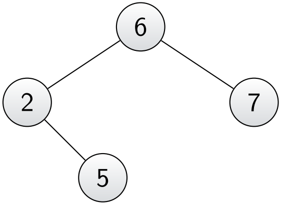
$6, 7, 2, 5, 3, 9, 8, 1, 4, 10$



A toy example

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

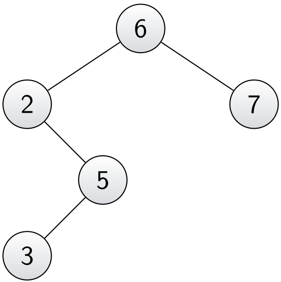
$6, 7, 2, 5, 3, 9, 8, 1, 4, 10$



A toy example

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

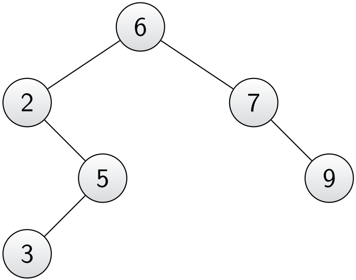
$6, 7, 2, 5, 3, 9, 8, 1, 4, 10$



A toy example

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

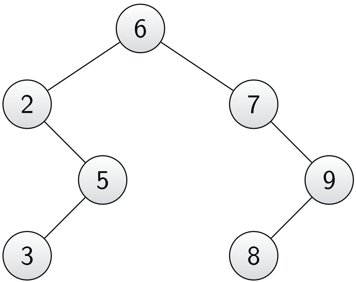
$6, 7, 2, 5, 3, 9, 8, 1, 4, 10$



A toy example

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

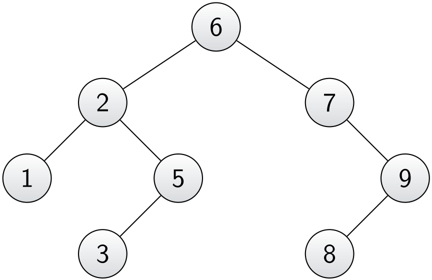
$6, 7, 2, 5, 3, 9, 8, 1, 4, 10$



A toy example

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

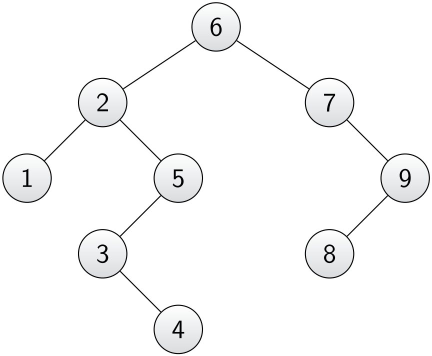
$6, 7, 2, 5, 3, 9, 8, 1, 4, 10$



A toy example

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

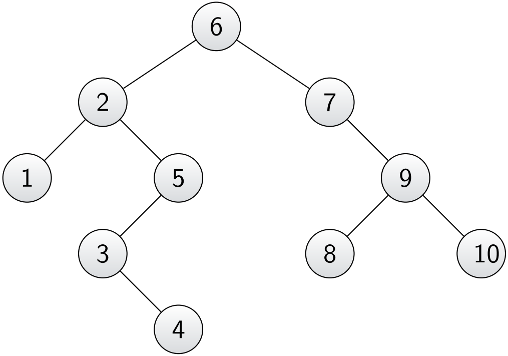
$6, 7, 2, 5, 3, 9, 8, 1, 4, 10$



A toy example

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

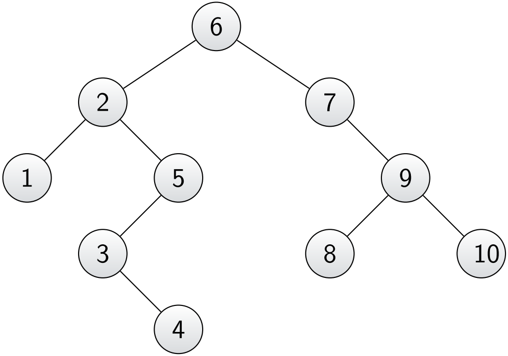
$6, 7, 2, 5, 3, 9, 8, 1, 4, 10$



A toy example

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

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

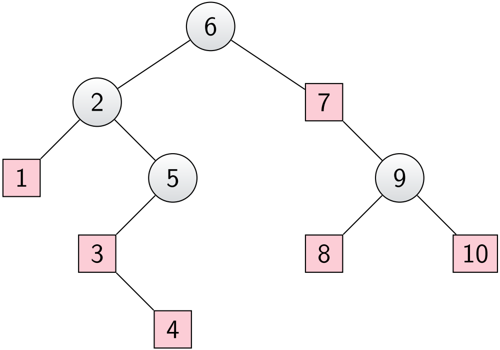


L_{0n} = number of nodes with no left descendant,

A toy example

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

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

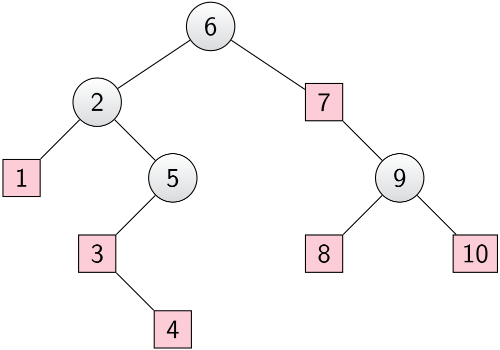


L_{0n} = number of nodes with no left descendant,

A toy example

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

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



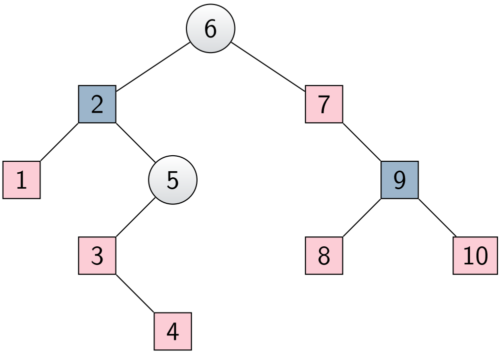
L_{0n} = number of nodes with no left descendant,

L_{1n} = number of nodes with exactly one left descendant.

A toy example

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

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



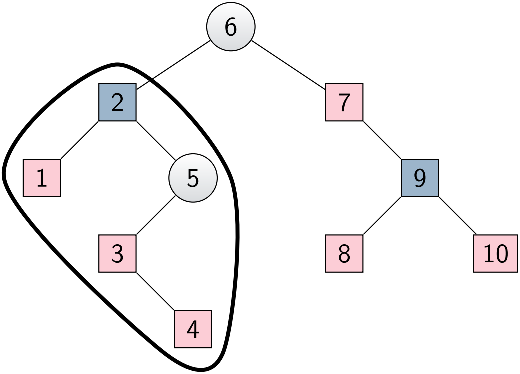
L_{0n} = number of nodes with no left descendant,

L_{1n} = number of nodes with exactly one left descendant.

A toy example

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

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



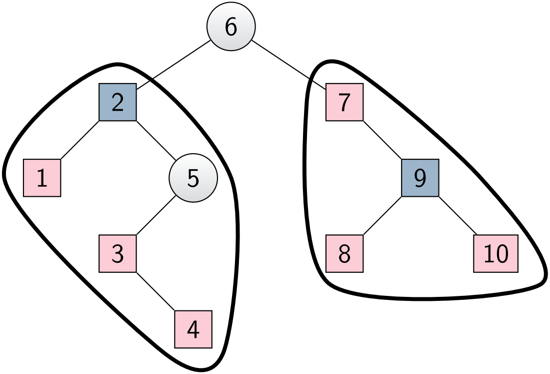
L_{0n} = number of nodes with no left descendant,

L_{1n} = number of nodes with exactly one left descendant.

A toy example

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

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



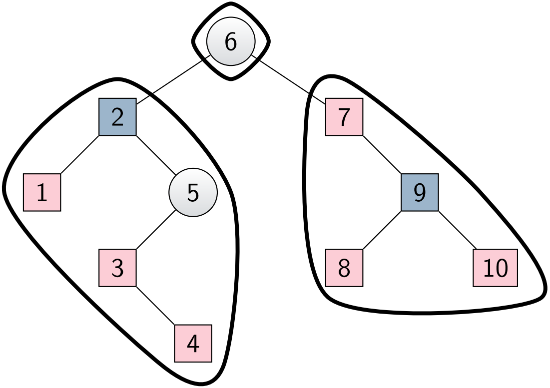
L_{0n} = number of nodes with no left descendant,

L_{1n} = number of nodes with exactly one left descendant.

A toy example

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

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



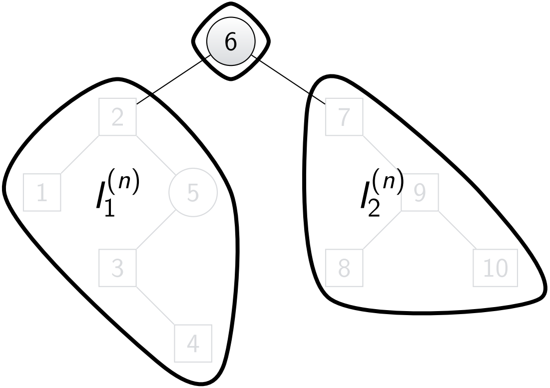
L_{0n} = number of nodes with no left descendant,

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,

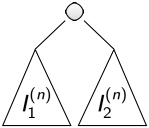
L_{1n} = number of nodes with exactly one left descendant.

A toy example

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

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

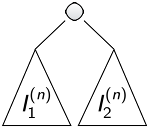
$L_{1n} = \#$ nodes with exactly one left descendant.



A toy example

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

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



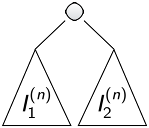
For $Y_n := (L_{0n}, L_{1n})$,

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.



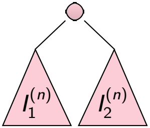
For $Y_n := (L_{0n}, L_{1n})$, we have $Y_0 = (0,0)$ and

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \qquad n \geq 1,$$

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

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

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \qquad n \geq 1,$$

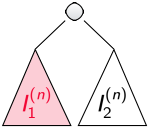
with $I_1^{(n)} \sim \text{unif}\{0, \ldots, n-1\}$, $I_2^{(n)} = n - 1 - I_1^{(n)}$ and $b_n = \begin{pmatrix} \mathbf{1}_{\{I_1^{(n)}=0\}} \\ \mathbf{1}_{\{I_1^{(n)}=1\}} \end{pmatrix}$.

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

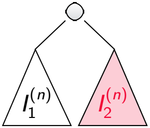
$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \qquad n \geq 1,$$

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

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

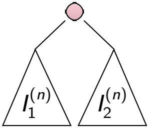
$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \qquad n \geq 1,$$

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

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

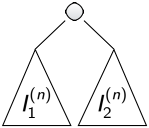
$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \qquad n \geq 1,$$

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

A toy example

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

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



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

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \qquad n \geq 1,$$

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

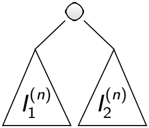
For $n \geq 4$ (cf. Devroye 1991),

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

A toy example

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

$L_{0n} = \#$ 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

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \qquad n \geq 1,$$

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

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

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

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

$$\mathrm{Cov}(Y_n)^{-1/2}(Y_n - \mathbb{E}[Y_n]) \xrightarrow{d} \mathcal{N}(0, \mathrm{Id}_2).$$

General recursion

recursion before:

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \quad n \geq 1.$$

general recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0,$$

General recursion

recursion before:

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \quad n \geq 1.$$

general recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0,$$

- Y_n random vector in $\mathbb{R}^d$,

General recursion

recursion before:

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \quad n \geq 1.$$

general recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0,$$

- Y_n random vector in $\mathbb{R}^d$,
- K number of subproblems,

General recursion

recursion before:

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \quad n \geq 1.$$

general recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0,$$

- Y_n random vector in $\mathbb{R}^d$,
- K number of subproblems,
- $I^{(n)} = (I_1^{(n)}, \dots, I_K^{(n)}) \in \{0, \dots, n\}^K$ sizes of subproblems,

General recursion

recursion before:

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \quad n \geq 1.$$

general recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0,$$

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

General recursion

recursion before:

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \quad n \geq 1.$$

general recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0,$$

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

General recursion

recursion before:

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \quad n \geq 1.$$

general recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0,$$

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

General recursion

recursion before:

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \quad n \geq 1.$$

general recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0,$$

- ▷ 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 $\mathbb{R}^d$,
- ▷ $(Y_n^{(r)})_{n \geq 0} \stackrel{d}{=} (Y_n)_{n \geq 0}$ for $r = 1, \ldots, K$,
- ▷ $(A_1(n), \ldots, A_K(n), b_n, I^{(n)}), (Y_n^{(1)})_{n \geq 0}, \ldots, (Y_n^{(K)})_{n \geq 0}$ are independent.

The contraction method

General recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0.$$

The contraction method

General recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0.$$

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

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

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

The contraction method

General recursion:

$$Y_n \stackrel{d}{=} \sum_{r=1}^{K} A_r(n) Y^{(r)}_{I_r^{(n)}} + b_n, \qquad n \geq n_0.$$

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

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

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:

$$X_n \stackrel{d}{=} \sum_{r=1}^{K} A_r^{(n)} X^{(r)}_{I_r^{(n)}} + b^{(n)}, \qquad n \geq n_0,$$

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 ($n \geq n_0$):

$$X_n \stackrel{d}{=} \sum_{r=1}^{K} A_r^{(n)} X_{I_r^{(n)}}^{(r)} + b^{(n)} .$$

The contraction method

Modified recursion ($n \geq n_0$):

$$X_n \stackrel{d}{=} \sum_{r=1}^{K} \underbrace{\boxed{A_r^{(n)}}}_{\to A_r^*} X_{I_r^{(n)}}^{(r)} + b^{(n)} .$$

The contraction method

Modified recursion ($n \geq n_0$):

$$X_n \stackrel{d}{=} \sum_{r=1}^{K} \underbrace{\boxed{A_r^{(n)}}}_{\to A_r^*} X_{I_r^{(n)}}^{(r)} + \underbrace{\boxed{b^{(n)}}}_{\to b^*}.$$

The contraction method

Modified recursion ($n \geq n_0$):

$$X_n \stackrel{d}{=} \sum_{r=1}^{K} \underbrace{\boxed{A_r^{(n)}}}_{\to A_r^*} X_{I_r^{(n)}}^{(r)} + \underbrace{\boxed{b^{(n)}}}_{\to b^*}.$$

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

$$X \stackrel{d}{=} \sum_{r=1}^{K} A_r^* X^{(r)} + b^*$$

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

The contraction method

Modified recursion ($n \geq n_0$):

$$X_n \stackrel{d}{=} \sum_{r=1}^{K} \underbrace{\boxed{A_r^{(n)}}}_{\to A_r^*} X_{I_r^{(n)}}^{(r)} + \underbrace{\boxed{b^{(n)}}}_{\to b^*}.$$

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

$$X \stackrel{d}{=} \sum_{r=1}^{K} A_r^* X^{(r)} + b^*$$

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

Idea: $\begin{array}{c} A_r^{(n)} \to A_r^* \\ b^{(n)} \to b^* \end{array} \implies X_n \to X$

The contraction method

Modified recursion ($n \geq n_0$):

$$X_n \stackrel{d}{=} \sum_{r=1}^{K} \underbrace{\boxed{A_r^{(n)}}}_{\to A_r^*} X_{I_r^{(n)}}^{(r)} + \underbrace{\boxed{b^{(n)}}}_{\to b^*}.$$

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

$$X \stackrel{d}{=} \sum_{r=1}^{K} A_r^* X^{(r)} + b^*$$

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

Idea: $\begin{array}{l} A_r^{(n)} \to A_r^* \\ b^{(n)} \to b^* \end{array} \implies X_n \to X$

Now:

$$\sum_{r=1}^{K} \left\| A_r^{(n)} - A_r^* \right\|_s + \left\| b^{(n)} - b^* \right\|_s = \mathrm{O}(R(n)) \implies \zeta_s(X_n, X) = \mathrm{O}(R(n))$$

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 $n \to \infty$,

$$\left\|b^{(n)} - b^*\right\|_s + \sum_{r=1}^{K} \left\|A_r^{(n)} - A_r^*\right\|_s + \sum_{r=1}^{K} \left\|\mathbf{1}_{\{I_r^{(n)} < \ell\}} A_r^{(n)}\right\|_s = \mathrm{O}(R(n))$$

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

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

then we have, as $n \to \infty$,

$$\zeta_s(X_n, X) = \mathrm{O}(R(n)),$$

where X is the unique solution of the limit equation in $\mathcal{M}_s^d(0, \mathrm{Id}_d)$.

Convergence result II

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 $\sum_{r=1}^K A_r^*(A_r^*)^T = \mathrm{Id}_d$ almost surely and that there exists some sequence $R(n) \downarrow 0$ such that, as $n \to \infty$,

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

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

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

then we have, as $n \to \infty$,

$$\zeta_3(X_n, \mathcal{N}(0, \mathrm{Id}_d)) = \mathrm{O}(R(n)).$$

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 Φ the standard normal distribution function, we have, as $n \to \infty$,

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

With our theorem

We have, as $n \to \infty$,

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

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

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

Quicksort

$$\begin{aligned} Y_n = &\text{ number of key comparisons needed by Quicksort} \\ &\text{ to sort } n \text{ randomly permuted distinct numbers} \end{aligned}$$

Neininger and Rüschendorf (2002)

We have, as $n \to \infty$,

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

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 $n \to \infty$,

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

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

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

Hwang (2003)

As $n \to \infty$, we have

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

m	$3 \ldots 19$	$20 \ldots 26$
$3(3/2 - \alpha)$	$> 1/2$	$0.45 \ldots 0.002$

With our theorem

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

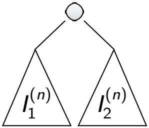
$$\zeta_3\left(\frac{Y_n - \mathbb{E}[Y_n]}{\sqrt{\mathrm{Var}(Y_n)}}, \mathcal{N}(0,1) \right) = \begin{cases} \mathrm{O}\left(n^{-1/2+\varepsilon}\right), & 3 \le m \le 19, \\ \mathrm{O}\left(n^{-3(3/2-\alpha)}\right), & 20 \le m \le 26. \end{cases}$$

Applications

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

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

$L_{1n} = \#$ 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

$$\zeta_3\big(\mathrm{Cov}(Y_n)^{-1/2}(Y_n - \mathbb{E}[Y_n]), \mathcal{N}(0, \mathrm{Id}_2)\big) = \mathrm{O}(n^{-1/2}).$$

Applications: periodic functions in mean & variance

Theorem

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

$$Y_n \stackrel{d}{=} Y^{(1)}_{I_1^{(n)}} + Y^{(2)}_{I_2^{(n)}} + b_n, \quad n \geq n_0,$$

with $I_1^{(n)} \sim \mathrm{Bin}(n, \frac{1}{2})$, $I_2^{(n)} = n - I_1(n)$ and $\|b_n\|_3 = \mathrm{O}(1)$. We assume that, as $n \to \infty$, we have

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

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

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

Thank you!