Inversions in split trees and conditional Galton–Watson trees

Xing Shi Cai, Cecilia Holmgren, Svante Janson, Tony Johansson, Fiona Skerman

Analysis of Algorithms 2018

Department of Mathematics, Uppsala University, Sweden

My coauthors

Table of contents

The definition

Inversions in a permutation

- Let $\sigma_1, \dots, \sigma_n$ be a permutation of $\{1, \dots, n\}$.
- If $i < j$ and $\sigma_i > \sigma_j$, then the pair (σ_i, σ_j) is called an *inversion*.

$$\boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} \quad \boxed{5}$$

Inversions in a permutation

- Let $\sigma_1, \dots, \sigma_n$ be a permutation of $\{1, \dots, n\}$.
- If $i < j$ and $\sigma_i > \sigma_j$, then the pair (σ_i, σ_j) is called an *inversion*.

$$\boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} \quad \boxed{5}$$

$$\boxed{5} \quad \boxed{4} \quad \boxed{3} \quad \boxed{2} \quad \boxed{1}$$

Inversions in a permutation

- Let $\sigma_1, \ldots, \sigma_n$ be a permutation of $\{1, \ldots, n\}$.
- If $i < j$ and $\sigma_i > \sigma_j$, then the pair (σ_i, σ_j) is called an *inversion*.

$$\boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} \quad \boxed{5}$$

$$\boxed{5} \quad \boxed{4} \quad \boxed{3} \quad \boxed{2} \quad \boxed{1}$$

$$\boxed{4} \quad \boxed{1} \quad \boxed{3} \quad \boxed{5} \quad \boxed{2}$$

Inversions in a permutation

- Let $\sigma_1, \ldots, \sigma_n$ be a permutation of $\{1, \ldots, n\}$.
- If $i < j$ and $\sigma_i > \sigma_j$, then the pair (σ_i, σ_j) is called an *inversion*.

$$\boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} \quad \boxed{5}$$

$$\boxed{5} \quad \boxed{4} \quad \boxed{3} \quad \boxed{2} \quad \boxed{1}$$

$$\boxed{4} \quad \boxed{1} \quad \boxed{3} \quad \boxed{5} \quad \boxed{2}$$

Inversions in a permutation

- Let $\sigma_1, \ldots, \sigma_n$ be a permutation of $\{1, \ldots, n\}$.
- If $i < j$ and $\sigma_i > \sigma_j$, then the pair (σ_i, σ_j) is called an *inversion*.

$$\boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} \quad \boxed{5}$$

$$\boxed{5} \quad \boxed{4} \quad \boxed{3} \quad \boxed{2} \quad \boxed{1}$$

$$\boxed{4} \quad \boxed{1} \quad \boxed{3} \quad \boxed{5} \quad \boxed{2}$$

Inversions in a permutation

- Let $\sigma_1, \ldots, \sigma_n$ be a permutation of $\{1, \ldots, n\}$.
- If $i < j$ and $\sigma_i > \sigma_j$, then the pair (σ_i, σ_j) is called an *inversion*.

$$\boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} \quad \boxed{5}$$

$$\boxed{5} \quad \boxed{4} \quad \boxed{3} \quad \boxed{2} \quad \boxed{1}$$

$$\boxed{4} \quad \boxed{1} \quad \boxed{3} \quad \boxed{5} \quad \boxed{2}$$

Inversions in a permutation

- Let $\sigma_1, \dots, \sigma_n$ be a permutation of $\{1, \dots, n\}$.
- If $i < j$ and $\sigma_i > \sigma_j$, then the pair (σ_i, σ_j) is called an *inversion*.

$$\boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} \quad \boxed{5}$$

$$\boxed{5} \quad \boxed{4} \quad \boxed{3} \quad \boxed{2} \quad \boxed{1}$$

$$\boxed{4} \quad \boxed{1} \quad \boxed{3} \quad \boxed{5} \quad \boxed{2}$$

Inversions in a permutation

- Let $\sigma_1, \ldots, \sigma_n$ be a permutation of $\{1, \ldots, n\}$.
- If $i < j$ and $\sigma_i > \sigma_j$, then the pair (σ_i, σ_j) is called an *inversion*.

$$\boxed{1} \quad \boxed{2} \quad \boxed{3} \quad \boxed{4} \quad \boxed{5}$$

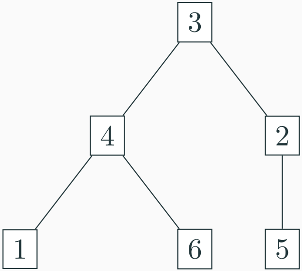
$$\boxed{5} \quad \boxed{4} \quad \boxed{3} \quad \boxed{2} \quad \boxed{1}$$

$$\boxed{4} \quad \boxed{1} \quad \boxed{3} \quad \boxed{5} \quad \boxed{2}$$

Inversions in a fixed tree

- Let T be a tree with node set V.
- Let λ be node labeling $\lambda : V \to \{1, \dots, |V|\}$.
- Define the number of *inversions*

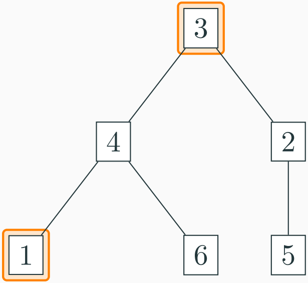
$$I(T, \lambda) \stackrel{\text{def}}{=} \sum_{u<v} \mathbf{1}_{\lambda(u)>\lambda(v)}.$$



Inversions in a fixed tree

- Let T be a tree with node set V.
- Let λ be node labeling $\lambda : V \to \{1, \dots, |V|\}$.
- Define the number of *inversions*

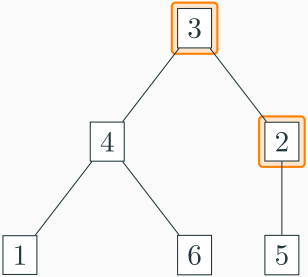
$$I(T, \lambda) \stackrel{\text{def}}{=} \sum_{u<v} \mathbf{1}_{\lambda(u)>\lambda(v)}.$$



Inversions in a fixed tree

- Let T be a tree with node set V.
- Let λ be node labeling $\lambda : V \to \{1, \dots, |V|\}$.
- Define the number of *inversions*

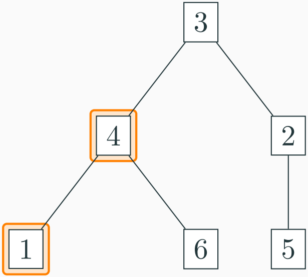
$$I(T, \lambda) \stackrel{\text{def}}{=} \sum_{u<v} \mathbf{1}_{\lambda(u)>\lambda(v)}.$$



Inversions in a fixed tree

- Let T be a tree with node set V.
- Let λ be node labeling $\lambda : V \to \{1, \dots, |V|\}$.
- Define the number of *inversions*

$$I(T, \lambda) \stackrel{\text{def}}{=} \sum_{u<v} \mathbf{1}_{\lambda(u)>\lambda(v)}.$$



Inversions in fixed trees

Inversions in a randomly labeled tree

- Fix the tree T.

Inversions in a randomly labeled tree

- Fix the tree T.
- Choose a uniform random labeling λ of T.

Inversions in a randomly labeled tree

- Fix the tree T.
- Choose a uniform random labeling λ of T.
- We study the random variable $I(T) = I(T, \lambda)$.

Inversions in a randomly labeled tree

- Fix the tree T.
- Choose a uniform random labeling λ of T.
- We study the random variable $I(T) = I(T, \lambda)$.
- Flajolet, Poblete, and Viola (1998) showed that this random variable for Cayley trees converges to an Airy distribution.

Inversions in a randomly labeled tree

- Fix the tree T.
- Choose a uniform random labeling λ of T.
- We study the random variable $I(T) = I(T, \lambda)$.
- Flajolet, Poblete, and Viola (1998) showed that this random variable for Cayley trees converges to an Airy distribution.
- Panholzer and Seitz (2012) generalized this to conditional Galton–Watson trees.

Expectations

- Note that

$$\mathbb{E}\left[I(T)\right] = \sum_{u<v} \mathbb{E}\left[\mathbf{1}_{\lambda(u)>\lambda(v)}\right] = \frac{1}{2}\sum_{u<v} 1 \stackrel{\text{def}}{=} \frac{1}{2}\Upsilon(T).$$

- $\Upsilon(T)$ is also known as the *total path length*, since

$$\Upsilon(T) \stackrel{\text{def}}{=} \sum_{v} d(v),$$

where $d(v)$ is the depth of v.

Inversions in a sequence of trees

- Let T_n be a sequence of trees of size n.

Inversions in a sequence of trees

- Let T_n be a sequence of trees of size n.
- For $T_n = P_n$ (a path of length n) [Feller (1968)]

$$\frac{I(P_n) - \mathbb{E}\left[I(P_n)\right]}{n} \xrightarrow{\mathrm{d}} N(0, \sigma^2).$$

Inversions in a sequence of trees

- Let T_n be a sequence of trees of size n.
- For $T_n = P_n$ (a path of length n) [Feller (1968)]

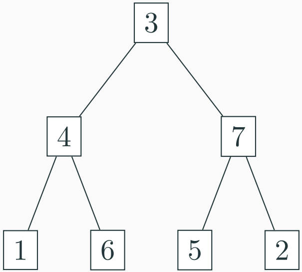
$$\frac{I(P_n) - \mathbb{E}\left[I(P_n)\right]}{n} \xrightarrow{\mathrm{d}} N(0, \sigma^2).$$

- What if T_n is a complete binary tree?

- Let T_n be a sequence of trees of size n.
- For $T_n = P_n$ (a path of length n) [Feller (1968)]

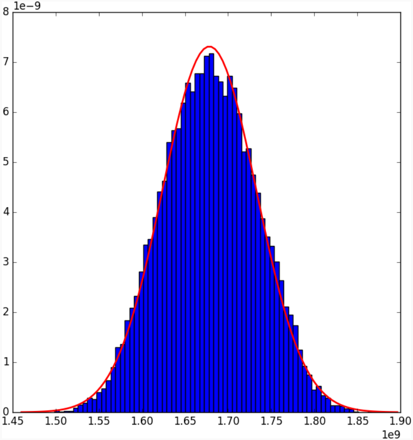
$$\frac{I(P_n) - \mathbb{E}\left[I(P_n)\right]}{n} \xrightarrow{\mathrm{d}} N(0, \sigma^2).$$

- What if T_n is a complete binary tree?



Simulations

- We did simulation on the complete binary tree of height 26
- Does the result suggest a central limit theorem?



The method of moments

- It is easy to see

$$\mathbb{E}\left[I(T_n)\right] = \Upsilon(T_n) \sim \frac{1}{2}\frac{n\log_2 n}{2}.$$

The method of moments

- It is easy to see

$$\mathbb{E}\left[I(T_n)\right] = \Upsilon(T_n) \sim \frac{1}{2}\frac{n\log_2 n}{2}.$$

- The second moment

$$\mathbb{E}\left[(I(T_n) - \Upsilon(T_n))^2\right] \sim \frac{1}{6}n^2.$$

The method of moments

- It is easy to see

$$\mathbb{E}\left[I(T_n)\right] = \Upsilon(T_n) \sim \frac{1}{2}\frac{n\log_2 n}{2}.$$

- The second moment

$$\mathbb{E}\left[(I(T_n) - \Upsilon(T_n))^2\right] \sim \frac{1}{6}n^2.$$

- The third moment

$$\mathbb{E}\left[(I(T_n) - \Upsilon(T_n))^3\right] = o\left(n^3\right).$$

The method of moments

- It is easy to see

$$\mathbb{E}\left[I(T_n)\right] = \Upsilon(T_n) \sim \frac{1}{2}\frac{n\log_2 n}{2}.$$

- The second moment

$$\mathbb{E}\left[(I(T_n) - \Upsilon(T_n))^2\right] \sim \frac{1}{6}n^2.$$

- The third moment

$$\mathbb{E}\left[(I(T_n) - \Upsilon(T_n))^3\right] = o\left(n^3\right).$$

- The fourth moment

$$\mathbb{E}\left[(I(T_n) - \Upsilon(T_n))^4\right] \sim \frac{31}{405}n^4.$$

The method of moments

- It is easy to see

$$\mathbb{E}\left[I(T_n)\right] = \Upsilon(T_n) \sim \frac{1}{2}\frac{n\log_2 n}{2}.$$

- The second moment

$$\mathbb{E}\left[(I(T_n) - \Upsilon(T_n))^2\right] \sim \frac{1}{6}n^2.$$

- The third moment

$$\mathbb{E}\left[(I(T_n) - \Upsilon(T_n))^3\right] = o\left(n^3\right).$$

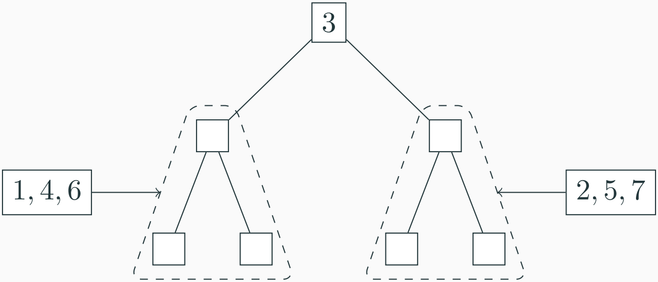
- The fourth moment

$$\mathbb{E}\left[(I(T_n) - \Upsilon(T_n))^4\right] \sim \frac{31}{405}n^4.$$

- So it cannot be a normal distribution!

A key observation

- Let Z_o be the number of inversions involving the root.
- Then Z_o and the numbers of inversions in the left subtree and right subtree are independent.
- Proof by conditioning on the labels that go the left and the right.



A key lemma

- Let z_v be the size of the subtree at v.
- Let Z_v be the number of inversions involving v and one of its descendants.

Lemma 1

Let T be a fixed tree. Then

$$I(T) \stackrel{\mathrm{d}}{=} \sum_{v \in V} Z_v,$$

where $\{Z_v\}_{v \in V}$ are independent random variables, and $Z_v \sim \mathrm{Unif}\{0, 1, \dots, z_v - 1\}$.

Cumulants

- The cumulant-generating function of a r.v. X is

$$K_X(t) = \log \mathbb{E}\left[e^{tX}\right].$$

- The cumulants $\varkappa_k(X)$ are defined by

$$K_X(t) = \sum_{k \geq 1} \varkappa_k(X) \frac{t^k}{k!}.$$

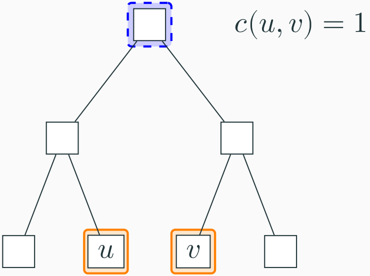
- If X is independent of Y, then

$$\varkappa_k(X+Y) = \varkappa_k(X) + \varkappa_k(Y).$$

- We can compute centralized-moments from cumulants.

k-total common ancestors

- For k nodes $v_1, \dots, v_k$, let $c(v_1, \dots, v_k)$ be the number of ancestors that they share.
- We define
$$\Upsilon_k(T) \stackrel{\text{def}}{=} \sum_{v_1,\dots,v_k} c(v_1, \dots, v_k).$$
- Note that $\Upsilon(T) = \Upsilon_1(T) - |V|$.

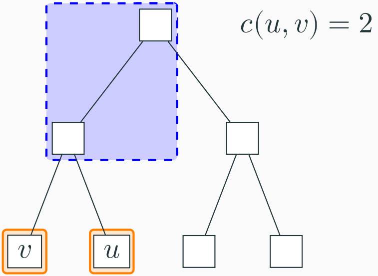


k-total common ancestors

- For k nodes $v_1, \dots, v_k$, let $c(v_1, \dots, v_k)$ be the number of ancestors that they share.
- We define

$$\Upsilon_k(T) \stackrel{\text{def}}{=} \sum_{v_1,\dots,v_k} c(v_1, \dots, v_k).$$

- Note that $\Upsilon(T) = \Upsilon_1(T) - |V|$.



Cumulants for $I(T)$

Theorem 2

We have

$$\mathbb{E}\left[I(T)\right] = \varkappa_1(I(T)) = \frac{1}{2}(\Upsilon_1(T) - |V|),$$
$$\mathrm{Var}\left(I(T)\right) = \varkappa_2(I(T)) = \frac{1}{12}(\Upsilon_2(T) - |V|).$$

More generally, for $k \geq 1$,

$$\varkappa_{2k+1}(I(T)) = 0, \qquad \varkappa_{2k}(I(T)) = \frac{B_{2k}}{2k}(\Upsilon_{2k}(T) - |V|),$$

where B_k denotes the k-th Bernoulli number.

The condition for convergence

Theorem 3

Let T_n be a sequence of fixed trees on n nodes. Let

$$X_n = \frac{I(T_n) - \mathbb{E}\left[I(T_n)\right]}{\sqrt{\Upsilon_2(T_n)}}.$$

Assume that for all $k \geq 1$,

$$\frac{\Upsilon_{2k}(T_n)}{\Upsilon_2(T_n)^k} \to \zeta_{2k},$$

for some sequence (ζ_{2k}). Then there exists a unique X with

$$\varkappa_{2k-1}(X) = 0, \qquad \varkappa_{2k}(X) = \frac{B_{2k}}{2k}\zeta_{2k}, \qquad k \geq 1,$$

such that $X_n \xrightarrow{\mathrm{d}} X$.

Back to the Path

- Let $T_n = P_n$ (a path of length n),
$$\Upsilon_k(T_n) \sim \frac{1}{k+1} n^{k+1}.$$
- Thus
$$\frac{\Upsilon_{2k}(T_n)}{\Upsilon_2(T_n)^k} \to 0 \qquad (k \geq 2).$$
- So $(I(T_n) - \mathbb{E}[I(T_n)])/n$ converges to X with
$$\varkappa_k(X) = 0, \qquad (k \geq 3).$$
- Then X must be a normal distribution.

Back to complete binary trees

Theorem 4

Let $b \geq 2$ and let T_n be the complete b-ary tree of height m with n nodes. Then

$$X_n = \frac{I(T_n) - \mathbb{E}\left[I(T_n)\right]}{n} \xrightarrow{\mathrm{d}} \sum_{d \geq 0} \sum_{j=1}^{b^d} \frac{U_{d,j}}{b^d},$$

where $(U_{d,j})_{d \geq 0, j \geq 1}$ are independent $\mathrm{Unif}[-1/2, 1/2]$.

Inversions in split trees

Binary search trees (BST)

- BST is a computer data structure for storing "item" according to the order of their "keys".
- BST can be defined with a bijection to permutations.
- The average height of a BST of size n is $\alpha \ln n - \beta lnlnn$ [Reed, 2003].

$$\boxed{4} \quad \boxed{2} \quad \boxed{3} \quad \boxed{5} \quad \boxed{1}$$

Binary search trees (BST)

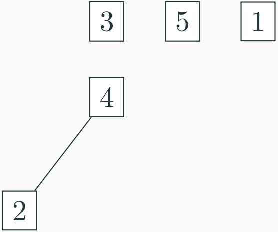
- BST is a computer data structure for storing “item” according to the order of their “keys”.
- BST can be defined with a bijection to permutations.
- The average height of a BST of size n is $\alpha \ln n - \beta lnlnn$ [Reed, 2003].

$\boxed{2}$ $\boxed{3}$ $\boxed{5}$ $\boxed{1}$

$\boxed{4}$

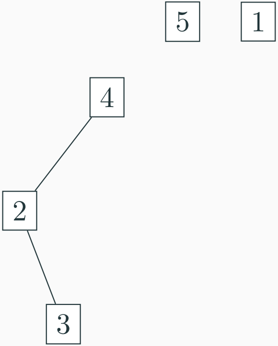
Binary search trees (BST)

- BST is a computer data structure for storing "item" according to the order of their "keys".
- BST can be defined with a bijection to permutations.
- The average height of a BST of size n is $\alpha \ln n - \beta lnlnn$ [Reed, 2003].



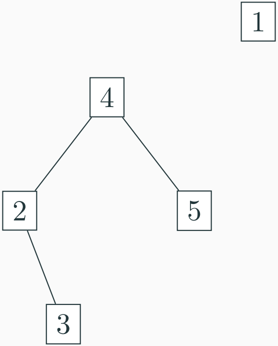
Binary search trees (BST)

- BST is a computer data structure for storing "item" according to the order of their "keys".
- BST can be defined with a bijection to permutations.
- The average height of a BST of size n is $\alpha \ln n - \beta lnlnn$ [Reed, 2003].



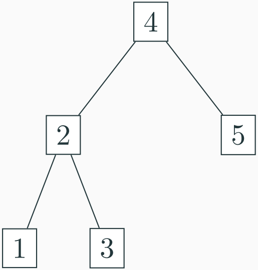
Binary search trees (BST)

- BST is a computer data structure for storing "item" according to the order of their "keys".
- BST can be defined with a bijection to permutations.
- The average height of a BST of size n is $\alpha \ln n - \beta lnlnn$ [Reed, 2003].



Binary search trees (BST)

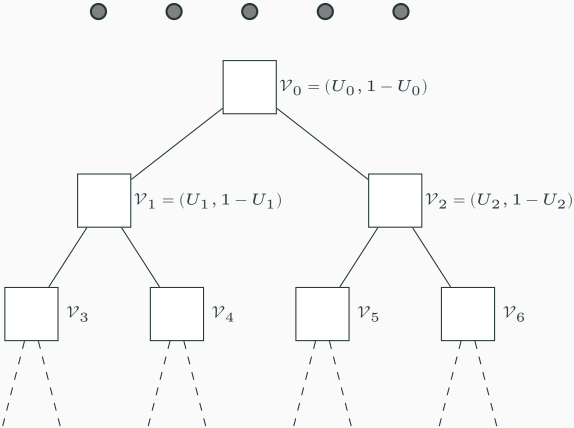
- BST is a computer data structure for storing "item" according to the order of their "keys".
- BST can be defined with a bijection to permutations.
- The average height of a BST of size n is $\alpha \ln n - \beta lnlnn$ [Reed, 2003].



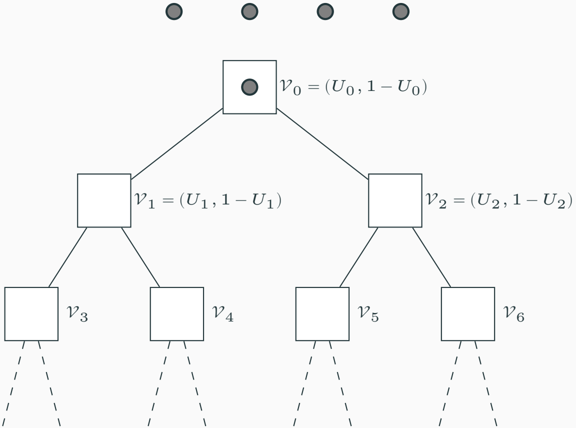
BST as a split tree i

- We can construct BST in another way.
- Consider an infinite binary tree.
- Each node is a "bucket" of size one.
- Each node is given a split vector $\mathcal{V} = (U, 1 - U)$ chosen independently.
- n balls come into the root one by one.
- When a bucket has more than one node, the extra goes to child nodes chosen at random according to $\mathcal{V}$.
- All empty buckets are removed in the end.

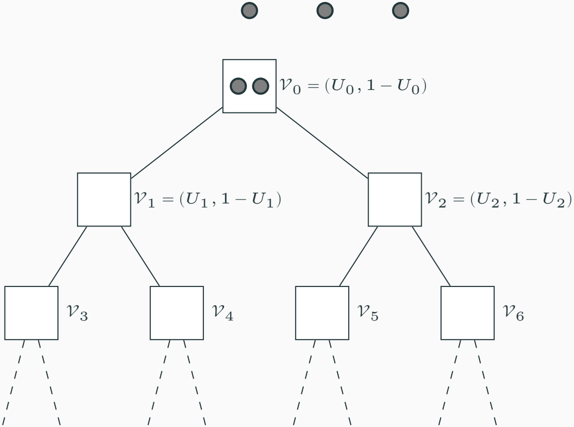
BST as a split tree ii



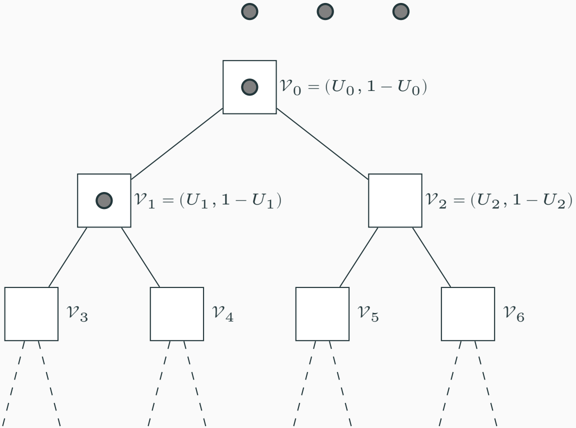
BST as a split tree ii



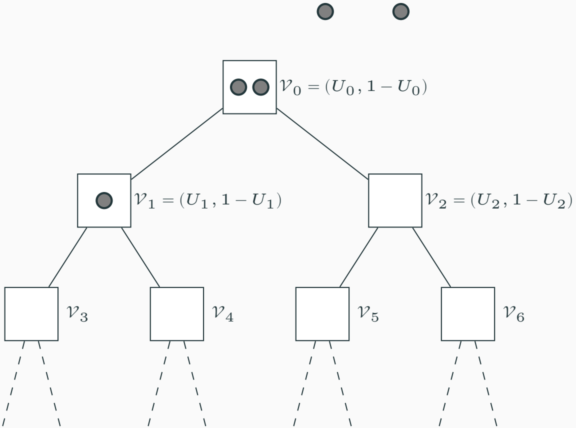
BST as a split tree ii



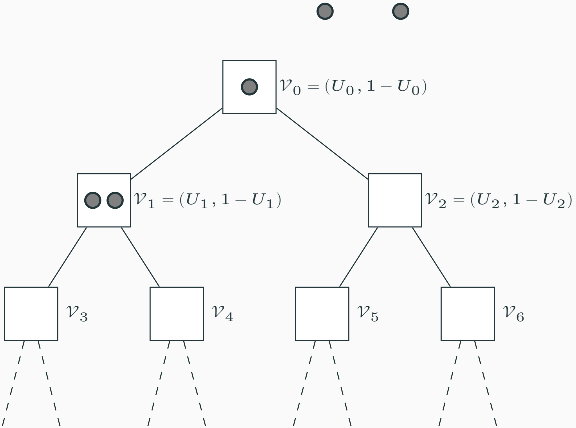
BST as a split tree ii



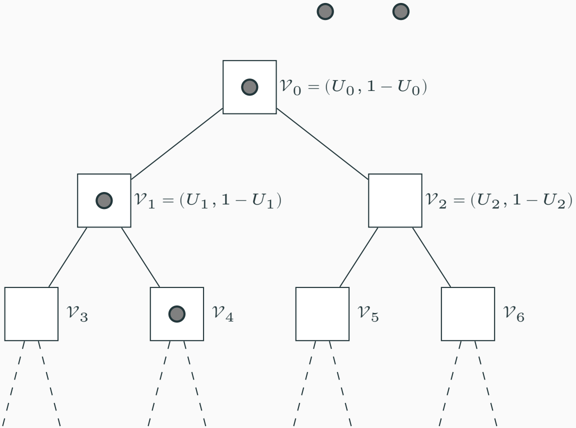
BST as a split tree ii



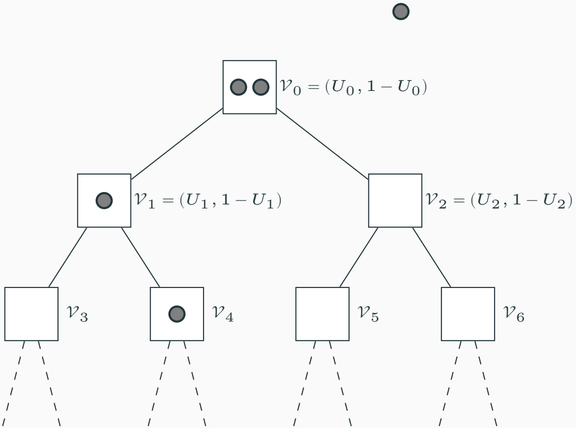
BST as a split tree ii



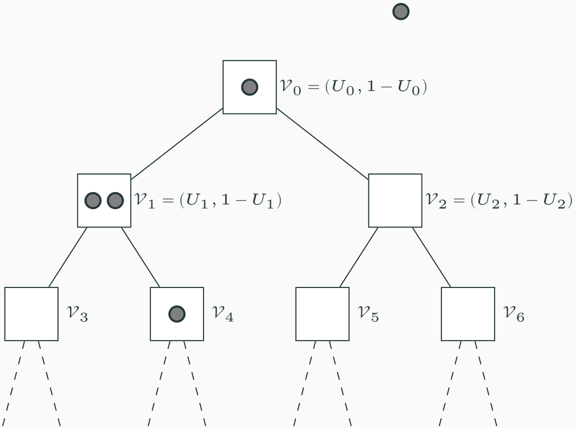
BST as a split tree ii



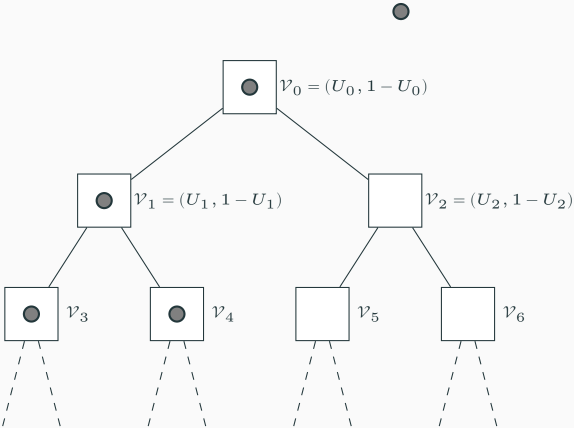
BST as a split tree ii



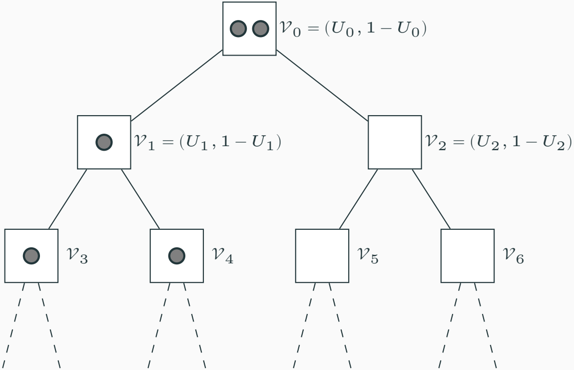
BST as a split tree ii



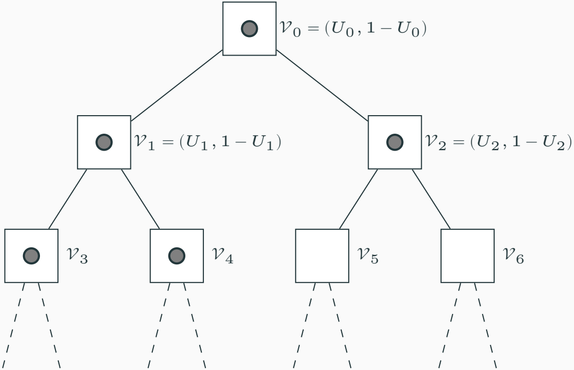
BST as a split tree ii



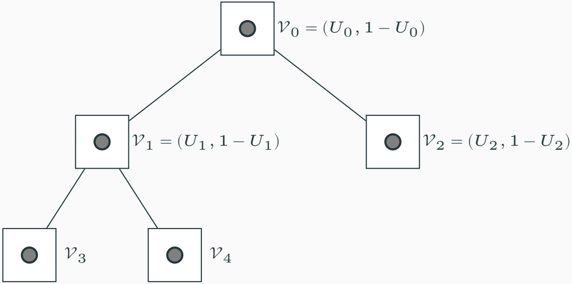
BST as a split tree ii



BST as a split tree ii



BST as a split tree ii



Split trees

- By choosing different:
  - inifnite trees
  - bucket sizes
  - distributions of split vector
- split trees encompasses:
  - binary search trees
  - b-ary search trees
  - digital search trees
  - tries, etc.
- Split trees are introduced by Devroye (1999).

Split trees

- By choosing different:
    - inifnite trees
    - bucket sizes
    - distributions of split vector
- split trees encompasses:
    - binary search trees
    - b-ary search trees
    - digital search trees
    - tries, etc.
- Split trees are introduced by Devroye (1999).

Inversions in split trees

- We first choose the split tree T_n with n balls.
- Then we randomly label the balls.
- We define $\widehat{I}(T_n)$ as the number of inversions for balls.
- We study

$$\hat{X}_n = \frac{\widehat{I}(T_n) - \mathbb{E}\left[\widehat{I}(T_n)\right]}{n}.$$

Results for split trees

Theorem 5

Let T_n be a b-ary split tree with bucket size s_0. Let $\mathcal{V} = (V_1, \dots, V_b)$ be a split vector. Let $\hat{X}$ be the unique solution for the fixed-point equation

$$\hat{X} \overset{\mathrm{d}}{=} \sum_{i=1}^{b} V_i \hat{X}^{(i)} + \sum_{j=1}^{s_0} U_j + \frac{s_0}{2} D(\mathcal{V}).$$

Then $\hat{X}_n \xrightarrow{\mathrm{d}} \hat{X}$.

- Proof by the contraction method.
- A similar result holds for labeling nodes instead of balls.

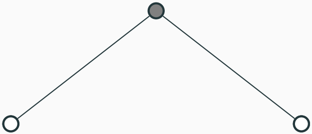
Inversions in conditional Galton-Watson trees

Galton-Watson trees

- A Galton–Watson tree starts with a root node.
- Each node in the tree is given a random number of child nodes.
- The numbers of children are independent with distribution ξ.

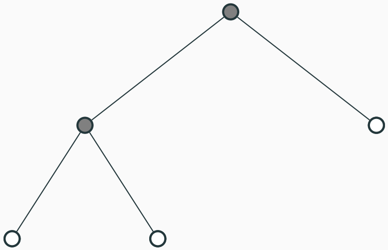
Galton-Watson trees

- A Galton–Watson tree starts with a root node.
- Each node in the tree is given a random number of child nodes.
- The numbers of children are independent with distribution ξ.



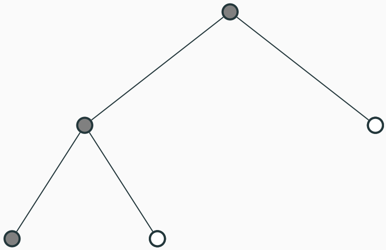
Galton-Watson trees

- A Galton–Watson tree starts with a root node.
- Each node in the tree is given a random number of child nodes.
- The numbers of children are independent with distribution ξ.



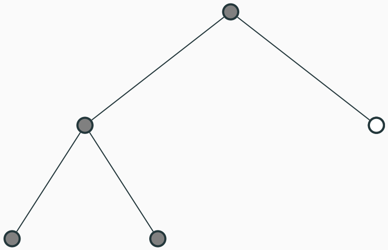
Galton-Watson trees

- A Galton–Watson tree starts with a root node.
- Each node in the tree is given a random number of child nodes.
- The numbers of children are independent with distribution ξ.



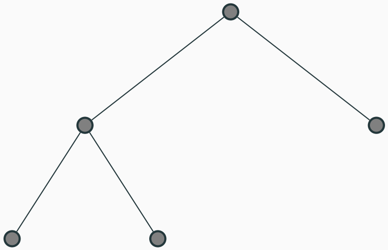
Galton-Watson trees

- A Galton–Watson tree starts with a root node.
- Each node in the tree is given a random number of child nodes.
- The numbers of children are independent with distribution ξ.



Galton-Watson trees

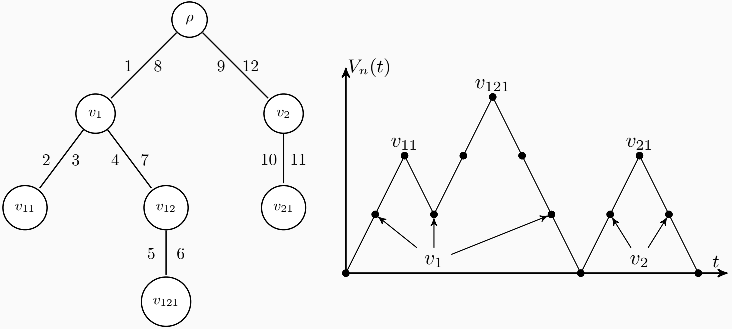
- A Galton–Watson tree starts with a root node.
- Each node in the tree is given a random number of child nodes.
- The numbers of children are independent with distribution ξ.



Conditional Galton–Watson

- A conditional Galton–Watson tree T_n is a Galton–Watson tree conditioned on having n nodes.
- It encompasses (uniform random)
  - plane trees
  - binary trees
  - b-ary trees
  - Cayley trees
- Very well studied, see, e.g., Janson (2012).

The depth-first walk on a conditional Galton–Watson tree



It is well-known that the depth-first walk on conditional GW trees converges to Brownian excursions [Aldous (1991a), Aldous (1991b), Aldous (1993), and Le Gall (2005)].

Let $e(t)$ be a Brownian excursion. Let

$$\eta \stackrel{\text{def}}{=} 4 \int_{0 \le s \le t \le 1} \min_{s \le u \le t} e(u) \mathrm{d}s \mathrm{d}t.$$

Theorem 6

Assume that $\mathbb{E}\left[\xi\right] = 1$, $\mathrm{Var}\left(\xi\right) = \sigma^2 \in (0, \infty)$, *and* $\mathbb{E}\left[e^{\alpha\xi}\right] < \infty$ *for some* $\alpha > 0$. *Then*

$$\frac{I(T_n) - \frac{1}{2}\Upsilon(T_n)}{n^{5/4}} \xrightarrow{\mathrm{d}} \frac{1}{\sqrt{12\sigma}} \sqrt{\eta}\, N(0,1).$$

Results for conditional GW trees ii

- Let
$$X_n = \frac{I(T_n) - \mathbb{E}[I(T_n)]}{n^{3/2}}.$$

- Then we can decompose
$$X_n = \frac{I(T_n) - \frac{1}{2}\Upsilon(T_n)}{n^{3/2}} + \frac{\Upsilon(T_n) - \mathbb{E}[\Upsilon(T_n)]}{2n^{3/2}}.$$

- Our result shows that the first term goes to zero.
- Aldous (1991b) showed that $\Upsilon(T_n)/n^{-3/2}$ converges to an Airy distribution.
- So X_n also converges to an Airy distribution.
- We recover result from Panholzer and Seitz, 2012.

Patterns on trees

- Let σ be a permutation of $\{1, \ldots, k\}$. Let

$$R_\sigma(T, \lambda) = \sum_{u_1 < \cdots < u_k} 1_{[\lambda(u_1, \ldots, u_k) = \sigma]}.$$

- Then $R_{21}(T, \lambda) = I(T, \lambda)$.
- Recently Albert, Holmgren, Johansson, and Skerman, 2018 studied $R_\sigma(T, \lambda)$ for complete binary trees and split trees.
- The spirit – most questions about permutations can be asked for trees.

Questions?

References i

- M. Albert et al. "Permutations in binary trees and split trees". To appear in AofA'18 proceedings. 2018.
- D. Aldous. "The continuum random tree. I". In: *Ann. Probab.* 19.1 (1991), pp. 1–28.
- D. Aldous. "The continuum random tree. II. An overview". In: *Stochastic analysis (Durham, 1990)*. Vol. 167. London Math. Soc. Lecture Note Ser. Cambridge Univ. Press, Cambridge, 1991, pp. 23–70.
- D. Aldous. "The continuum random tree. III". In: *Ann. Probab.* 21.1 (1993), pp. 248–289.

- L. Devroye. "Universal limit laws for depths in random trees". In: *SIAM J. Comput.* 28.2 (1999), pp. 409–432.
- W. Feller. *An introduction to probability theory and its applications. Vol. I*. 3rd. John Wiley & Sons, Inc., New York-London-Sydney, 1968, pp. xviii+509.
- P. Flajolet, P. Poblete, and A. Viola. "On the analysis of linear probing hashing". In: *Algorithmica* 22.4 (1998), pp. 490–515.

- S. Janson. "Simply generated trees, conditioned Galton-Watson trees, random allocations and condensation: extended abstract". In: *23rd Intern. Meeting on Probabilistic, Combinatorial, and Asymptotic Methods for the Analysis of Algorithms (AofA'12)*. Discrete Math. Theor. Comput. Sci. Proc., AQ. Assoc. Discrete Math. Theor. Comput. Sci., Nancy, 2012, pp. 479–490.
- J.-F. Le Gall. "Random trees and applications". In: *Probab. Surveys* 2 (2005), pp. 245–311.
- A. Panholzer and G. Seitz. "Limiting distributions for the number of inversions in labelled tree families". In: *Ann. Comb.* 16.4 (2012), pp. 847–870.

B. Reed. “The Height of a Random Binary Search Tree”. In: *J. ACM* 50.3 (May 2003), pp. 306–332.