The time-marginalized coalescent prior for hierarchical clustering

Boyles, L.; Welling, M.

Publication date
2013

Document Version
Submitted manuscript

Published in
26th Annual Conference on Neural Information Processing Systems 2012

Citation for published version (APA):
https://papers.nips.cc/paper/4786-the-time-marginalized-coalescent-prior-for-hierarchical-clustering

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
1 Proofs

**Lemma 1.** A tree \( \psi_n \) has \( T(\psi_n) = \frac{(n-1)!}{\prod_{i=1}^{n-1} m_i} \) possible orderings on its internal nodes, where \( m_i \) is the number of internal nodes in the subtree rooted at node \( i \).

**Proof.** This can be done with a recursion relation. Consider an unordered tree \( \psi_n \) with \( n_l \) leaves in the left subtree, \( n_r \) in the right, and define \( m_i = n_i - 1 \) to be the number of internal nodes of subtree \( i \). Additionally, there are \( k_l \) possible orderings of the left subtree’s internal nodes, and \( k_r \) possible orderings of the right subtree’s internal nodes. Consider the case where \( k_l = k_r = 1 \), where we need to count the number of ways in which the two sequences of length \( m_r \) and \( m_l \) can be ”interleaved”. There are \( (m_r + m_l)! \) possible orderings of the \( m_r + m_l \) nodes if we have no constraints on the orderings of the right nodes with respect to each other, and no such constraints for the left nodes. Thus, the total number of ways to interleave the two sequences is \( \frac{(m_r + m_l)!}{m_r! m_l!} = \binom{m_r + m_l}{m_r} \). See Figure 1. For \( k_r, k_l > 1 \), we simply multiply these in, giving the total number of orderings as:

\[
T(\psi_n) = \prod_{i=1}^{n-1} \binom{m_r i + m_l i}{m_r i}
\]

Note we can apply the same logic to the right and left subtrees to determine \( k_r \) and \( k_l \). Given the children of node \( i \) are \( l_i \) and \( r_i \), the number of orderings \( k_i \) consistent with a particular subtree \( i \) is:

\[
k_i = \binom{m_r i + m_l i}{m_r i} k_r k_l
\]

So we can write out the total number of orderings as a product over all internal nodes of the tree:

\[
T(\psi_n) = \prod_{i=1}^{n-1} \binom{m_r i + m_l i}{m_r i}
\]

Furthermore, note that \( m_i = m_l i + m_r i + 1 \) (the number of internal nodes in a tree is the sum of those in the subtrees, plus the root), so the denominator of the binomial coefficients of a parent node will cancel the numerators of the coefficients of its children, however leaving a term proportional to \( \frac{1}{m_i} \), and the numerator of the root will not be canceled. Thus we have:

\[
T(\psi_n) = \frac{(n-1)!}{\prod_{i=1}^{n-1} m_i}
\]

\( \square \)

**Theorem 1.** \( p(\psi_n) \) defines an exchangeable and consistent prior over \( \Psi_n \).
Figure 1: A tree with two unbalanced subtrees. The internal nodes for each of the subtrees can only be ordered in one way with respect to each other, however, for the overall tree there are \( \frac{6!}{3!3!} = \binom{6}{3} = 20 \) orderings of the internal nodes.

Proof. Since \( p(\psi_n) \) doesn’t depend on the order of data seen, it is exchangeable. Thus for consistency all we need to show is that we have a well defined conditional prior; ie that when we marginalize out a particular point from \( p(\psi_n) \) we get back \( p(\psi_{n-1}) \), if \( \psi_{n-1} \) is a tree structure obtained by removing a leaf from \( \psi_n \). We can do this by showing 

\[
\sum_{\psi_{n+1} \in C(\psi_n)} T(\psi_{n+1}) = \binom{n+1}{2} T(\psi_n)
\]

which can be proven by induction, and we take this as our inductive hypothesis for all \( l < n \).

The base case for \( n = 1 \):

\[
T(\psi_2) = \frac{(2 - 1)!}{1!!} = 1 = \binom{2}{2} T(\psi_1)
\]

Consider the tree \( \psi_n \) and its two subtrees \( \psi_{nl} \) and \( \psi_{nr} \). As listed before, we have \( T(\psi_n) = \binom{m_l + m_r}{m_l} T(\psi_{nl}) T(\psi_{nr}) \). Depending on where we add the \( n+1 \)st point, (denoted \( x_{n+1} \)) we have:

\[
T(\psi_{n+1}) = \begin{cases} 
\binom{m_l + m_r + 1}{m_l + 1} T(\psi_{nl+1}) T(\psi_{nr}) & \text{if } x_{n+1} \text{ is added to the left subtree} \\
\binom{m_l + m_r + 1}{m_r + 1} T(\psi_{nl}) T(\psi_{nr+1}) & \text{if } x_{n+1} \text{ is added to the right subtree} \\
\binom{m_l + m_r + 1}{m_l} \binom{m_l + m_r}{m_r} T(\psi_{nl}) T(\psi_{nr}) & \text{if } x_{n+1} \text{ is added above the root}
\end{cases}
\]

Summing over all such \( \psi_{n+1} \), and invoking the inductive hypothesis, we have:

\[
\sum_{\psi_{n+1} \in C(\psi_n)} T(\psi_{n+1}) = \binom{m_l + m_r + 1}{m_l + 1} \binom{n_l + 1}{2} T(\psi_{nl}) T(\psi_{nr}) + \binom{m_l + m_r + 1}{m_r + 1} \binom{n_r + 1}{2} T(\psi_{nl}) T(\psi_{nr}) + \binom{m_l + m_r + 1}{m_l} \binom{m_l + m_r}{m_r} T(\psi_{nl}) T(\psi_{nr})
\]

2
\[
\begin{align*}
&= \left( \frac{m_l + m_r}{m_l} \right) T(\psi_{n_l}) T(\psi_{n_r}) \left( \frac{m_l + m_r + 1}{m_l + 1} \frac{(n_l + 1)n_l}{2} + \frac{m_l + m_r + 1}{m_r + 1} \frac{(n_r + 1)n_r}{2} + 1 \right) \\
&= T(\psi_{n_l}) \left( \frac{(n_l - 1)(n_l + n_r + 2)}{2} + 1 \right) \\
&= T(\psi_{n_r}) \left( \frac{(n_l - 1)(n + 2)}{2} + \frac{2}{2} \right) \\
&= \left( \frac{n + 1}{2} \right) T(\psi_{n_l})
\end{align*}
\]
Figure 2: Posterior sample from our model applied to the leukemia dataset. Best viewed in color. Each pure subtree is painted a color unique to the class associated with it. The OTHERS class is a set of datapoints to which no diagnostic label was assigned.