APPENDIX

MATHEMATICAL DEFINITIONS

The trees we use are members of a normed vector space: we can add them, find out how far apart they are and interpolate between them. In principle, we can run clustering algorithms to find configurations that are similar, and identify when a model has left one cluster and entered another.

Definition 1. *Let $\mathcal{S}$ be a set of labels, and let $\mathbb{F}$ be a field like the real or complex numbers. Then we define a node $\boldsymbol{n}$ as a triplet of the form $(s, v, \boldsymbol{E})$, with $v \in \mathbb{F}$, $s \subset \mathcal{S}$, and the set $\boldsymbol{E}$ is a (possibly empty) set of nodes of the same form with the restriction that no two nodes in $\boldsymbol{E}$ may have the same label in their first ordinate. We also define the triple $\mathsf{O} = (\emptyset, 0, \emptyset)$, which we will call the null tree, and define $\boldsymbol{T}$ to be the union of the set of all trees composed of a finite number of these nodes.*

The ordinates of $\boldsymbol{u} = (\boldsymbol{u}_{\mathcal{S}}, \boldsymbol{u}_v, \boldsymbol{u}_{\boldsymbol{E}})$ in $\boldsymbol{T}$ correspond to its label, value, *and* extension set. *An element of $\boldsymbol{u}_{\boldsymbol{E}}$ will be called an* extension.

In our discussion, it will help to have a few more descriptive terms.

A node with an empty extension set is called a *simple* node, if this node happens to be a root node, then it is a simple tree.

Two trees, $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{T}$ are called *compatible* if either $\boldsymbol{u}_{\mathcal{S}} = \boldsymbol{v}_{\mathcal{S}}$ or at least one of $\boldsymbol{u}$ and $\boldsymbol{v}$ is O.

We define the *depth* of a tree with:

$$\text{depth}(\boldsymbol{u}) = \begin{cases} 0 & \text{if } \boldsymbol{u} = (\emptyset, 0, \emptyset) = \mathsf{O} \\ 1 & \text{if } \boldsymbol{u} \text{ is a simple node} \\ 1 + \max(\{\text{depth}(\boldsymbol{v}) : \forall \boldsymbol{v} \in \boldsymbol{u}_{\boldsymbol{E}}\}) & \text{otherwise.} \end{cases}$$

We will also define for $\boldsymbol{u} \in \boldsymbol{T}$,

$$\text{trim}(\boldsymbol{u}) = \begin{cases} \mathsf{O} & \text{if } \boldsymbol{u} = \mathsf{O} \\ \mathsf{O} & \text{if } \boldsymbol{u} \text{ is simple} \\ (\boldsymbol{u}_{\mathcal{S}}, \boldsymbol{u}_v, \{\text{trim}(\boldsymbol{e}) : & \\ \qquad\qquad \forall \boldsymbol{e} \in \boldsymbol{u}_{\boldsymbol{E}}\} \setminus \{\mathsf{O}\}) & \text{otherwise.} \end{cases}$$

The cardinality of a tree is the number of nodes it contains. Specifically,

$$\|\boldsymbol{u}\|_{\mathsf{T}} = \begin{cases} 0 & \text{if } \boldsymbol{u} = \mathsf{O} \\ 1 + \sum_{e \in \boldsymbol{u}_{\boldsymbol{E}}} \|\boldsymbol{e}\|_{\mathsf{T}} & \text{otherwise.} \end{cases}$$

Simple nodes are the only nodes that have a cardinality of one, and O is the only node or tree with a cardinality of zero.

The support of a tree is the number of nodes which have a non-zero value.

$$\operatorname{supp}(\boldsymbol{u}) = \begin{cases} 0 + \sum_{\boldsymbol{e}\in\boldsymbol{u}_{\boldsymbol{E}}} \operatorname{supp}(\boldsymbol{e}) & \text{if } \boldsymbol{u}_v = 0 \\ 1 + \sum_{\boldsymbol{e}\in\boldsymbol{u}_{\boldsymbol{E}}} \operatorname{supp}(\boldsymbol{e}) & \text{otherwise} \end{cases}.$$

Related is the fundamental support of a tree, which only counts nodes with no zero valued nodes in their connection to the root node

$$\operatorname{fund}(\boldsymbol{u}) = \begin{cases} 0 & \text{if } \boldsymbol{u}_v = 0 \\ 1 + \sum_{\boldsymbol{e}\in\boldsymbol{u}_{\boldsymbol{E}}} \operatorname{fund}(\boldsymbol{e}) & \text{otherwise} \end{cases}.$$

Clearly the support of a tree, $\operatorname{supp}\boldsymbol{u}$, must lie in the domain $[0, \|\boldsymbol{u}\|_{\mathsf{T}}]$ and $\operatorname{fund}\boldsymbol{u} \le \operatorname{supp}(\boldsymbol{u})$.

The *overlap* between two trees is defined

$$\operatorname{overlap}(\boldsymbol{u},\boldsymbol{v}) = \begin{cases} 0 & \text{if } \boldsymbol{u} = \mathbf{0} \text{ or } \boldsymbol{v} = \mathbf{0} \text{ or } \boldsymbol{u}_s \neq \boldsymbol{v}_s \\ 1 + \sum\limits_{\substack{\boldsymbol{e}\in\boldsymbol{u}_{\boldsymbol{E}} \\ \boldsymbol{f}\in\boldsymbol{v}_{\boldsymbol{E}}}} \operatorname{overlap}(\boldsymbol{e},\boldsymbol{f}) & \text{otherwise} \end{cases}$$

Clearly two trees, $\boldsymbol{u}$ and $\boldsymbol{v}$, are compatible if and only if $\operatorname{overlap}(\boldsymbol{u},\boldsymbol{v}) \neq 0$; they will be said to *completely overlap* if $\|\boldsymbol{u}\|_{\mathsf{T}} = \|\boldsymbol{v}\|_{\mathsf{T}} = \operatorname{overlap}(\boldsymbol{u},\boldsymbol{v})$.

We can now define scalar multiplication, and tree addition.

DEFINITION 2. *Take $a \in \mathbb{F}$ and $\boldsymbol{u} \in \boldsymbol{T}$, then*

$$a\boldsymbol{u} = \begin{cases} \mathbf{0} & \text{if } \boldsymbol{u} = \mathbf{0} \\ \left(\boldsymbol{u}_s, a\boldsymbol{u}_v, \{a\boldsymbol{f} : \boldsymbol{f} \in \boldsymbol{u}_{\boldsymbol{E}}\} \setminus \{\mathbf{0}\}\right) & \text{otherwise.} \end{cases} \tag{1}$$

DEFINITION 3. *Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be compatible elements of $\boldsymbol{T}$. Then taking the symbol $+$ to be addition in the field $\mathbb{F}$, we extend it to addition in $\boldsymbol{T}$ so that for nodes $\boldsymbol{u}$ and $\boldsymbol{v}$,*

$$\boldsymbol{u}+\boldsymbol{v} = \begin{cases} \mathbf{0} & \text{if } \boldsymbol{u} = \boldsymbol{v} = \mathbf{0} \\ \boldsymbol{u} & \text{if } \boldsymbol{u} \neq \mathbf{0} \text{ and } \boldsymbol{v} = \mathbf{0} \\ \boldsymbol{v} & \text{if } \boldsymbol{u} = \mathbf{0} \text{ and } \boldsymbol{v} \neq \mathbf{0} \\ (\boldsymbol{u}_s, \boldsymbol{u}_v + \boldsymbol{v}_v, \emptyset) & \text{if } \boldsymbol{u}_{\boldsymbol{E}}, \boldsymbol{v}_{\boldsymbol{E}} = \emptyset \\ \Big(\boldsymbol{u}_s, \boldsymbol{u}_v + \boldsymbol{v}_v, \big(\{\boldsymbol{f}+\boldsymbol{g} : \boldsymbol{f} \in \boldsymbol{u}_{\boldsymbol{E}} \text{ and } \boldsymbol{g} \in \boldsymbol{v}_{\boldsymbol{E}} \text{ and } \boldsymbol{f}_s = \boldsymbol{g}_s\} & \\ \quad \cup \{\boldsymbol{f} : \boldsymbol{f} \in \boldsymbol{u}_{\boldsymbol{E}} \text{ and } \boldsymbol{f}_s \neq \boldsymbol{g}_s \forall \boldsymbol{g} \in \boldsymbol{v}\} & \\ \quad \cup \{\boldsymbol{g} : \boldsymbol{g} \in \boldsymbol{v}_{\boldsymbol{E}} \text{ and } \boldsymbol{g}_s \neq \boldsymbol{f}_s \forall \boldsymbol{f} \in \boldsymbol{u}\}\big) \setminus \{\mathbf{0}\}\Big) & \text{otherwise.} \end{cases} \tag{2}$$

DEFINITION 4. *Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be compatible elements of $\boldsymbol{T}$. Then we define inner-multiplication between the two nodes*

$$\boldsymbol{u}\cdot\boldsymbol{v} = \left(\boldsymbol{u}_s, \boldsymbol{u}_v\boldsymbol{v}_v, \{\boldsymbol{f}\cdot\boldsymbol{g} : \boldsymbol{f} \in \boldsymbol{u}_{\boldsymbol{E}}, \boldsymbol{g} \in \boldsymbol{v}_{\boldsymbol{E}} \text{ and } \boldsymbol{f}_s = \boldsymbol{g}_s\} \setminus \{\mathbf{0}\}\right) \tag{3}$$

It can be shown that $\boldsymbol{T}$ with scalar multiplication and tree addition forms a vector space. This isn't quite enough to give us distances between trees, however, so we define a semi-norm

Definition 5. *Let $\boldsymbol{u}$ be an element of $\boldsymbol{T}$. Then we can define a semi-norm over $\boldsymbol{T}$*

$$\langle \boldsymbol{u} \rangle = \begin{cases} 0 & \text{if } \boldsymbol{u} = \mathbf{O} \\ |\boldsymbol{u}_v| & \text{if } \boldsymbol{u}_{\boldsymbol{E}} = \emptyset \\ |\boldsymbol{u}_v| + \sum_{\boldsymbol{e} \in \boldsymbol{u}_{\boldsymbol{E}}} \langle \boldsymbol{e} \rangle & \text{otherwise.} \end{cases}$$

It is clear that $\langle \boldsymbol{u} \rangle$ will always be non-negative, and the only shortcoming is that we can have a node $\boldsymbol{u}$ with $\langle \boldsymbol{u} \rangle = 0$, but $\boldsymbol{u} \neq \mathbf{O}$. In order to turn this into a normed vector space we take the set $\mathfrak{O} = \{\boldsymbol{u} : u \in \boldsymbol{T} \text{ and } \langle \boldsymbol{u} \rangle = 0\}$ and we construct an equivalence relation on $\boldsymbol{T}$ by the rule $[\boldsymbol{u}] \equiv [\boldsymbol{v}]$ if and only if there exist $\boldsymbol{z_u}, \boldsymbol{z_v} \in \mathfrak{O}$ such that $\boldsymbol{u} + \boldsymbol{z_u} = \boldsymbol{v} + \boldsymbol{z_v}$. It can be shown that scalar multiplication, tree addition, and the semi-norm behave appropriately with respect to the equivalence classes. This means that if we identify elements of $\boldsymbol{T}$ with their equivalence class, then we can take $\langle \boldsymbol{u} \rangle$ to be a norm and that it induces a distance function

$$\mathrm{d}(\boldsymbol{u}, \boldsymbol{v}) = \langle \boldsymbol{u} - \boldsymbol{v} \rangle$$

.

Definition 6. *We define the functions* mask *and* $\overline{\text{mask}}$ *that set the values associated with particular nodes in a tree to $v \in \mathbb{F}$. Specifically, if $\mathcal{L} \subset \mathcal{T}$, then*

$$\text{mask}(\boldsymbol{u}, \mathcal{L}, v) = \begin{cases} (\boldsymbol{u}_s, v, \{\text{mask}(\boldsymbol{f}, \mathcal{L}, v) : \boldsymbol{f} \in \boldsymbol{u}_{\boldsymbol{E}}\}) & \text{if } \boldsymbol{u}_s \in \mathcal{L} \\ (\boldsymbol{u}_s, \boldsymbol{u}_v, \{\text{mask}(\boldsymbol{f}, \mathcal{L}, v) : \boldsymbol{f} \in \boldsymbol{u}_{\boldsymbol{E}}\}) & \text{otherwise} \end{cases} \tag{4}$$

and

$$\overline{\text{mask}}(\boldsymbol{u}, \mathcal{L}, v) = \begin{cases} (\boldsymbol{u}_s, v, \{\overline{\text{mask}}(\boldsymbol{f}, \mathcal{L}, v) : \boldsymbol{f} \in \boldsymbol{u}_{\boldsymbol{E}}\}) & \text{if } \boldsymbol{u}_s \notin \mathcal{L} \\ (\boldsymbol{u}_s, \boldsymbol{u}_v, \{\overline{\text{mask}}(\boldsymbol{f}, \mathcal{L}, v) : \boldsymbol{f} \in \boldsymbol{u}_{\boldsymbol{E}}\}) & \text{otherwise.} \end{cases} \tag{5}$$

$\text{mask}(\boldsymbol{u}, \mathcal{L}, 0)$ *would return a tree similar to $\boldsymbol{u}$, but all its nodes that have labels in $\mathcal{L}$ would have values of zero.*

We also define several functions which prune or select a child from a tree's extension set. This function returns only the part of $\boldsymbol{u}$ which overlaps $\boldsymbol{p}$,

$$\text{excise}(\boldsymbol{u}, \boldsymbol{p}) = \begin{cases} (\boldsymbol{u}_s, \boldsymbol{u}_v, \{\text{excise}(\boldsymbol{f}, \boldsymbol{g}) : \boldsymbol{f} \in \boldsymbol{u}_{\boldsymbol{E}} \text{ and } \boldsymbol{g} \in \boldsymbol{p} \text{ and } \boldsymbol{f}_s = \boldsymbol{g}_s\} \setminus \{\mathbf{O}\}) & \text{if } \boldsymbol{u}_s = \mathcal{L} \\ \mathbf{O} & \text{if } \boldsymbol{u}_s \neq \boldsymbol{p}_s \\ (\boldsymbol{u}_s, \boldsymbol{u}_v, \emptyset) & \text{otherwise,} \end{cases} \tag{6}$$

and this one either returns an appropriately labelled child from the extension set (a branch), or $\mathbf{O}$.

$$\text{child}(\boldsymbol{u}, \ell) = \begin{cases} \boldsymbol{f} & \text{if } \boldsymbol{f}_s = \ell \text{ and } \boldsymbol{f} \in \boldsymbol{u}_{\boldsymbol{E}} \\ \mathbf{O} & \text{otherwise.} \end{cases} \tag{7}$$

We now finish with a definition of a function, $\Delta(,)$, that gives us a measure of the degree of divergence between two trees.

Definition 7. *The degree of deviation between two trees, $\boldsymbol{u}$ and $\boldsymbol{v}$ is given by the expression*

$$\Delta(\boldsymbol{u}, \boldsymbol{v}) = \begin{cases} \|u\|_{\mathsf{T}} + \|v\|_{\mathsf{T}} - 2\,\text{overlap}(\boldsymbol{u}, \boldsymbol{v}) + \langle \boldsymbol{u} - \boldsymbol{v} \rangle & \text{if } \boldsymbol{u} \text{ and } \boldsymbol{v} \text{ are compatible} \\ \|u\|_{\mathsf{T}} + \|v\|_{\mathsf{T}} & \text{otherwise.} \end{cases} \tag{8}$$

The rationale behind this definition is that if trees $\boldsymbol{u}$ and $\boldsymbol{v}$ are identical, then $\Delta(\boldsymbol{u}, \boldsymbol{v})$ will be zero. We also want nodes that aren't common to both trees to count as differences.