1 Binary Tree Tensor Networks

We are interested in the model space of binary tree tensor networks as shown in figure 1. We assume that every network is non-negative and normalized, and we set $f \equiv \text{id}$.

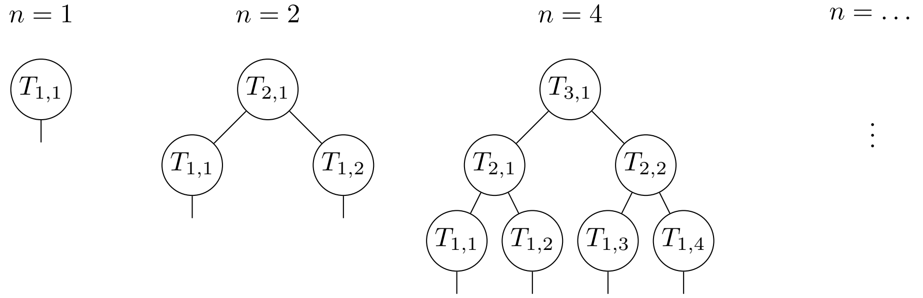

Figure 1: Binary tree model space for sequences of length $n = 2^k$.

1.1 Bulk Marginal Property

In definition **??** we saw how to construct a model with the desired bulk marginal property based on the base model. However, we might not always have a base model for every $n \in \mathbb{N}$ like discussed. Luckily, it turns out that this is not an issue, as there are many ways we can build a new model with the bulk marginal property from a base model even if it is only defined on a subset of $\mathbb{N}$. Without a proof, we might do the same procedure as in definition **??** but with bigger steps (instead of taking always the consecutive model), and induce the in-between models by marginalizing the bigger ones.

Alternatively, if we wanted a model with bulk marginal property that itself is also an element of our specified model space, we might ask ourselves, how we can construct a bigger tensor network while preserving the distribution in its leading random variables.

Let's analyze the following example: Say we wanted to integrate the tree tensor network for $n = 2$ in figure 1 into a bigger tree tensor network with $n = 4$. Note that by assumptions the tensor networks are non-negative and normalized, and $f \equiv \text{id}$. Thus, in order for our new tensor network to have the bulk marginal property, contracting the smaller network must be equivalent to contracting the bigger one, where the new nodes (in this case $T'_{1,3}$ and $T'_{1,4}$) are contracted with all-ones vectors, see figure 2.

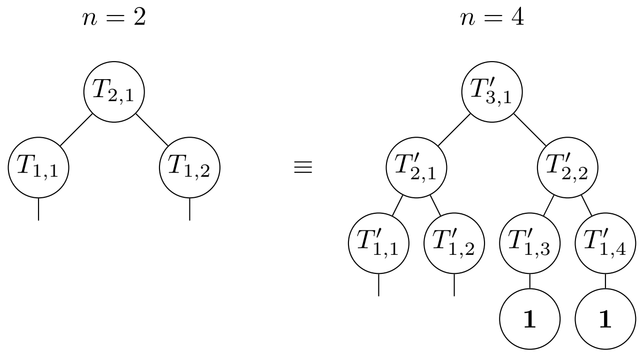

Figure 2: Bulk marginal property enforces the equivalence of these models.

Note that if we indeed had equivalence of these models, this would imply that the bigger model is now normalized as well based on lemma **??**.

Let's now analyze the vector $\boldsymbol{v}$ depicted in figure 3.

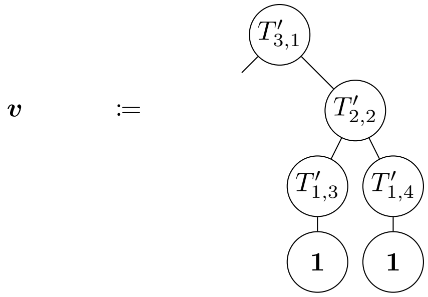

Figure 3: Contracting this sub-network with all-ones vectors yields vector $\boldsymbol{v}$.

What can we say about $\boldsymbol{v}$? Well, we can assume that it has at least one non-zero entry. This is always possible, and if $\boldsymbol{v}$ consisted of only zeros, then the network wouldn't be normalized.

But now we can initialize the remaining tensors: The leaf tensors $T'_{1,1}$ and $T'_{2,1}$ can be taking over from the smaller model, and the new tensor $T'_{2,1}$, which is now a vector of matrices like the old $T_{2,1}$, can initialized as the matrix $T_{2,1}$ divided by the non-zero entry of $\boldsymbol{v}$ at the same position. All other matrices in this vector will just be set to zero-matrices.

It is apparent that this initialization ensures the equivalence of the models as depicted in figure 2. It is also clear that this method works when transitioning from any $n = 2^k$ to $n' = 2^{k+1}$. Note that we also didn't increase the sizes of the tensors (except for $T'_{2,1}$, as it got another axis). Furthermore, when assuming all tensor entries are non-negative, then the new tensor network $\mathcal{T}'$ is also non-negative. This leads to the following observation:

Corollary 1.1. *Let $\mathcal{T}$ be a binary tree tensor network over $\Sigma^n, n = 2^k$. Then, there exists a binary tree tensor network $\mathcal{T}'$ over $\Sigma^{n'}, n' = 2^{k+1}$ s.t. the transition from $\mathcal{T}$ to $\mathcal{T}'$ complies with the bulk marginal property. Furthermore, $\mathcal{T}'$ complies with axes-sizes constrains (under the assumption of a maximum axis-size constrain which is increasing in n).*

1.2 Binary Tree Tensor Networks are Universal Approximators

Now, we want to analyze the properties of these binary tree tensor networks further. It may not bother us how we construct increasingly bigger models that satisfy the bulk marginal property, we know that the model space of binary tree tensor networks is capable of producing such families.

One question we might ask is whether such a model space restricts the space of possible probability distributions, and if so by how much. As it turns out, in the most general case when allowing very large tensors in the networks, we can model *any* probability distribution:

Proposition 1.1. *Given any probability distribution $p : \Sigma^{2^k} \mapsto [0, 1]$, we can always construct a binary tree tensor network $\mathcal{T}$ over Σ^{2^k} s.t. $p \equiv S_{2^k,\mathcal{T}}$. (Where $\mathcal{T}$ has the properties mentioned in the beginning and has no constrains on the tensor sizes.)*

Proof. For clarity reasons, we only show how to construct $\mathcal{T}$ for $n = 2^k = 4$. The procedure can easily be extended to the general case.

Our model structure is depicted in figure 4.

Now, we initialize the leaf matrices as identity matrices $\delta_2 \in \mathbb{R}^{|\Sigma| \times |\Sigma|}$. Thus, when contracting a leaf tensor with a one-hot encoded input vector at position i, we get the vector $\boldsymbol{v}^{(i)}$ with $\boldsymbol{v}^{(i)}_j = \mathbf{1}[X_i = c_j], c_j \in \Sigma$.

Now, the tensors in layer two are of the following form:

$$T_{2,j} : |\Sigma| \times |\Sigma| \mapsto \mathbb{R}^{|\Sigma|^2} \quad .$$

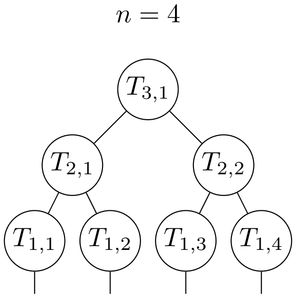

Figure 4: Model structure of binary tree tensor networks for $n = 2^k = 4$.

The outgoing axis may be index by $(X'_i, X'_{i+1}) \in \Sigma^2$. The map is then defined by

$$T_{2,j}(X_i, X_{i+1}) = \mathbf{1}[(X'_i, X'_{i+1}) = (X_i, X_{i+1})] \quad ,$$

i.e. $T_{2,j}$ is a three dimensional tensor with $|\Sigma| \times |\Sigma|$ many vectors of size $|\Sigma|^2$ which are one-hot encoded vectors of 2-tuples of Σ^2.

Finally, $T_{3,1}$ stores the entire probability distribution:

$$T_{3,1} : |\Sigma|^2 \times |\Sigma|^2 \mapsto [0,1], ((X'_1, X'_2), (X'_3, X'_4)) \mapsto p(X'_1, X'_2, X'_3, X'_4) \quad .$$

Thus, based on the construction we see that upon contracting the network with an initialization defined by $w \in \Sigma^4$, we get $S_{4,\mathcal{T}}(w) = p(w)$ as desired.

Note that this construction can easily be extended to arbitrary $n = 2^k$. □

As one might expect, we see that our general construction needs $\Omega(|\Sigma|^n)$ many parameters, as the root tensor stores all the $|\Sigma|^n$ many probabilities for $w \in \Sigma^n$.

1.3 Restricting Parameters

One natural question is what happens if we restrict the number of parameters. Obviously, if we don't have exponentially many parameters with respect to the word length, we won't be able to construct *every* probability distribution.

However, when modelling natural language for example, we really aren't interested in the most general case of probability distributions. For example, for a fixed word length n, we might want behavior similar to large scale time invariance (see definition **??**). Note that for a fixed n, the constrain of the bulk marginal property has no restrictive effect on the possible distributions.

Most decisively, we are interested in models capable of power-law behavior. Our goal is to show that binary tree tensor networks are incapable of this when we cap the number of parameters (i.e. the tensor sizes).

To formalize this, we first specify what it means to cap the parameters. There are two approaches that come to my mind: Either cap the total number of parameters (the entries of *all* tensors), or cap the axes-sizes and hence the size of each tensor individually. Of course, the latter approach implies that the total number of parameters are capped by $2n$ times the maximal number of parameters per tensor, as there are $2n-1$ many tensors in the network. Thus, it seems natural to cap the individual tensor sizes to ensure a *good* distribution of parameters over the tensors (which means that there shouldn't be one very large tensor and many small tensors). We might do this by capping the axes-sizes, as this allows the tensors to be more "cube-like" and to not have one big axis and two smaller ones for example (note that most tensors have three axes).

Thus, let us assume we cap the axes-sizes. We could define an upper bound for every tensor individually in the network, or, for every layer, but for simplicity's sake we define an upper bound on the axes-sizes in the entire network. As it turns out, it also doesn't really matter which approach we choose when arguing with complexity bounds.

So, we want to cap the axes-sizes. To this end, let d denote the biggest axis-size allowed. Now, there are multiple options again: First, d could stay constant for every network of size $n = 2^k$. In this case, the parameters grow linearly in n (since the number of tensors grows linearly). This, however, is probably too restrictive. The second approach is to let d grow with n. Of course, if $d := |\Sigma|^n$ (or even $d := |\Sigma|^{\frac{n}{2}} = \left(\sqrt{|\Sigma|}\right)^n$), we won't have any restrictions and way too many parameters. Alternatively, we could try to find a smaller base b for $d = b^n$. Another approach is to define $d(n) \in \mathcal{O}(n^p)$ for some $p \in \mathbb{N}$. This means that axes-sizes grow polynomially with respect to the word length n. Of course, this implies that every tensor grows polynomial in n with $\mathcal{O}((n^p)^3) = \mathcal{O}(n^{3p})$, and hence the parameter complexity of the entire network grows with $\mathcal{O}(n^{3p+1})$.

We see that polynomially growing parameters of the entire network with respect to n es equivalent to polynomially growing axes-sizes. Good. Let us now turn to the power-law constrain.

As we now know, there are different definitions for power-law behavior. The strongest proof would include to disprove weak power-law behavior (see definition **??**), as this would also disprove strong power law behavior (see definition **??**, proposition **??**). Alternatively, under the assumption that our model complies with the bulk marginal property, we can use the contraposition of theorem **??** to show the weaker fact that there is no model satisfying the bulk marginal property and having weak power-law behavior (and hence also strong power-law behavior).

Let us look at the latter approach of employing the contraposition of theorem **??**. What we would need to show is that there exists a character at position t, and no matter what tensor network we apply, we can bound the mutual information by $I(X_t, X_{t+\tau}) \in \mathcal{O}(e^{-\lambda\tau})$ for some fixed $\lambda \in \mathbb{R}_{>0}$, as this of course implies that there is no $\alpha \in \mathbb{R}_{>0}$ s.t. $I(X_t, X_{t+\tau}) \in \Omega(\tau^{-\alpha})$. Note that there obviously exist families satisfying the bulk marginal property. Thus, a potential proof of this claim has meaning concerning power-law behavior of binary tree tensor networks.

Theorem 1.1 (No Power-Law in BTTN with BMP for Polynomially Capped Parameters)**.** *Let $d(n)$ denote the maximum axis-size in a binary tree tensor network over Σ^n. If $d(n) \in \mathcal{O}(n^p)$ for some $p \in \mathbb{N}$, then there exists no family of binary tree tensor networks $(\mathcal{T}_{2^k})_{k\in\mathbb{N}}$ with associated model $S_n(w)$ s.t. S complies with the bulk marginal property and has weak power-law behavior.*