	Learning		Classification	
	Time	*Space*	*Time*	*Space*
TK	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(n^2)$
FTK	$A(n)$	$O(n^2)$	$A(n)$	$O(n^2)$
FTK+FS	$A(n)$	$O(n^2)$	k	k
ATK	$O(\frac{n^2}{q_\omega})$	$O(n^2)$	$O(\frac{n^2}{q_\omega})$	$O(n^2)$
DTK	d	d	d	d

Table 1. Computational time and space complexities for several tree kernel techniques: n is the tree dimension, q_ω is a speed-up factor, k is the size of the selected feature set, d is the dimension of space R^d, $O(\cdot)$ is the worst-case complexity, and $A(\cdot)$ is the average case complexity.

ateness with respect to the ideal properties, we evaluate whether these concrete basic composition functions yield to effective DTKs, and, finally, we evaluate the computation efficiency by comparing average computational execution times of TKs and DTKs. For the following experiments, we focus on a reduced space $\mathbb{R}^d$ with $d = 8192$.

5.1. Approximating Ideal Basic Composition Function

5.1.1. Concrete composition functions

We consider two possible approximations for the ideal composition function $\diamond$: the *shuffled γ-product* $\boxtimes$ and *shuffled circular convolution* $\boxdot$. These functions are defined as follows:

$$\overrightarrow{a} \boxtimes \overrightarrow{b} = \gamma \cdot p_1(\overrightarrow{a}) \otimes p_2(\overrightarrow{b})$$
$$\overrightarrow{a} \boxdot \overrightarrow{b} = p_1(\overrightarrow{a}) \odot p_2(\overrightarrow{b})$$

where: $\otimes$ is the element-wise product between vectors and $\odot$ is the circular convolution (as for distributed representations in (Plate, 1995)) between vectors; p_1 and p_2 are two different permutations of the vector elements; and γ is a normalization scalar parameter, computed as the average norm of the element-wise product of two vectors.

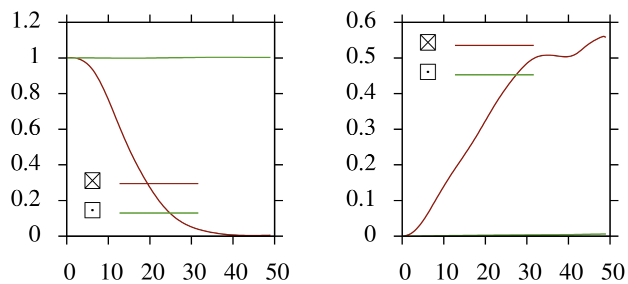
5.1.2. Empirical Evaluations of Properties

Properties 2.1, 2.2, and 2.3 hold by construction. The two permutation functions, p_1 and p_2, guarantee Prop. 2.1, for a high degree k, and Prop. 2.2. Property 2.3 is inherited from element-wise product $\otimes$ and circular convolution $\odot$.

Properties 2.4, 2.5 and 2.6 can only be approximated. Thus, we performed tests to evaluate the appropriateness of the two considered functions.

Property 2.4 approximately holds for $\boxdot$ since approximate norm preservation already holds for circular convolution, whereas $\boxtimes$ uses factor γ to preserve norm. We empirically evaluated this property. Figure 2(a) shows the average norm for the composition of an increasing number of basic vectors (i.e. vectors with unitary norm) with the two basic composition functions. Function $\boxdot$ behaves much better than $\boxtimes$.

Properties 2.5 and 2.6 were tested by measuring similarities between some combinations of vectors. The first experiment compared a single vector $\overrightarrow{a}$ to a combination $\overrightarrow{t}$ of several other vectors, as in property 2.5. Both functions resulted in average similarities below 1%, independently of the number of vectors in $\overrightarrow{t}$, satisfying property 2.5. To test property 2.6 we compared two compositions of vectors $\overrightarrow{a} \diamond \overrightarrow{t}$ and $\overrightarrow{b} \diamond \overrightarrow{t}$, where all the vectors are in common except for the first one. The average similarity fluctuates around 0, with $\boxdot$ performing better than $\boxtimes$; this is mostly notable observing that the variance grows with the number of vectors in $\overrightarrow{t}$ as shown in Fig. 2(b). A similar test was performed, with all the vectors in common except for the last one, yielding to similar results.



(a) Average norm of the vector obtained as combination of different numbers of basic random vectors

(b) Variance of the dot product between two combinations of basic random vectors with one common vector

Figure 2. Statistical properties for vectors on 100 samples ($d = 8192$).

In light of these results, $\boxdot$ seems to be a better choice than $\boxtimes$, although it should be noted that, for vectors of dimension d, $\boxtimes$ is computed in $O(d)$ time, while $\boxdot$ takes $O(d \log d)$ time.

5.2. Evaluating Distributed Tree Kernels: Direct and Task-based Comparison

In this section, we evaluate whether DTKs with the two concrete composition functions, $DTK_{\boxtimes}$ and $DTK_{\boxdot}$, approximate the original TK (as in Equation 4).We perform two sets of experiments: (1) a *direct comparison* where we directly investigate the correlation between DTK and TK values; and, (2) a *task based comparison* where we compare the performance of DTK against that of TK on two natural language processing tasks, i.e., question classification (QC) and textual entailment recognition (RTE).

5.2.1. Experimental Set-up

For the experiments, we used standard datasets for the two NLP tasks of QC and RTE.