# Supervised Tree-Wasserstein Distance

Yuki Takezawa[1 2], Ryoma Sato[1 2], Makoto Yamada[1 2]

[1]Kyoto University, [2]RIKEN AIP

## 1. Background

### Wasserstein Distance

- Powerful tool to measure the distance between distributions.
- High computational cost. (e.g., linear programing, Sinkhorn algorithm)

$$W(\mu_i, \mu_j) = \inf_{\gamma \in \Pi(\mu_i, \mu_j)} \int d(x_i, x_j) \gamma(dx, dy)$$

### Tree-Wasserstein Distance

- The Wasserstein distance on a tree.
- Closed form solution, which can be computed in linear time.

$$W_{d_T}(\mu_i, \mu_j) = \sum_{v \in V} w_v \left| \sum_{x \in \Gamma(v)} \mu_i(x) - \mu_j(x) \right|_1$$

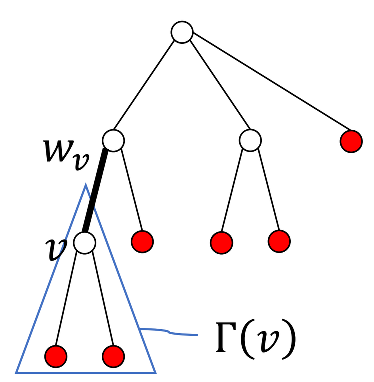


Fig.1

- $w_v$ : the edge length between $v$ and its parent node.
- $\Gamma(v)$ : a set of nodes contained in the subtree rooted at $v$ .

### Methods to Construct a Tree.

- Quadtree [Indyk & Thaper 03]
- Clustering based method [Le+ 19]

These methods are unsupervised.

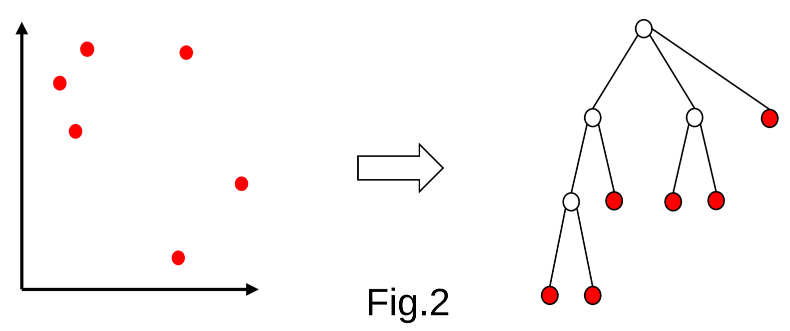

Fig.2

## 2. Contribution Summary

- We propose the **Supervised tree-Wasserstein (STW) distance** to construct a tree that can represent task-specific distances using the label information of documents.
- We propose the **Soft variant of the tree-Wasserstein distance** that is differential w.r.t. parent-child relationships in a tree.
- The STW distance outperforms other unsupervised tree-based methods in document classification tasks.
- Since the STW distance is GPU suitable, it can compute the tree-Wasserstein distance more efficiently.

## 3. Problem Setting

**Input :**
- $Z = \{z_1, z_2, \ldots, z_{N_{\text{leaf}}}\}$ : a set of words.
- $\mu_i$ : probability measure of a document $i$. (i.e., normalized bag-of-words)
- $y_i \in \mathbf{N}$ :a label of document $i$.
- $D = \{(\mu_i, y_i)\}_{i=1}^{M}$ : a training dataset.
- $N_{\text{in}}$ : the number of internal nodes. (hyper parameter)

Then, we denote as follows :

- $V = \{v_1, v_2, \ldots, v_{N_{\text{in}} + N_{\text{leaf}}}\}$ : a set of nodes ($v_1$ is a root).

**Goal :**

To obtain the tree metric that can represent task specific distance.

## 4. Soft Tree-Wasserstein Distance

### Difficulty :

- Optimization w.r.t. a tree structure is discrete optimization. (e.g., $\Gamma(v)$)

### Soft Tree-Wasserstein Distance :

- Differential w.r.t. parent-child relationships.
- $P_{\text{sub}}(x \mid v)$ : probability that $x \in \Gamma(v)$. (i.e.,$x$ is contained the subtree rooted at $v$.)

$$W_{d_T}^{\text{soft}}(\mu_i, \mu_j) = \sum_{v \in V} w_v \left| \sum_{x \in V} P_{\text{sub}}(x \mid v) \; (\mu_i(x) - \mu_j(x)) \right|_{\alpha}$$

**Theorem 2 :**

If the tree metric is given and $|\cdot|_{\alpha}$ approaches $|\cdot|_1$, then the soft tree-Wasserstein distance converges to the tree-Wasserstein distance.

### Parent-Child Relationships :

- These relationships can be represented by a directed tree.
  - i.e., an adjacency matrix $\mathbf{D}_{\text{par}}$ .

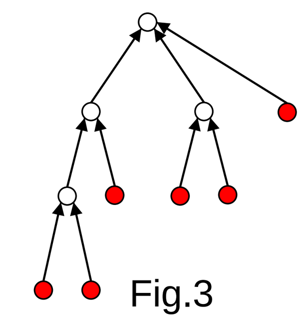

Fig.3

**Theorem 1 : conditions of an adjacency matrix**

Let $\mathbf{D}_{\text{par}} \in \{0,1\}^{|V| \times |V|}$ be an adjacency matrix of the directed graph $G$.
If $\mathbf{D}_{\text{par}}$ satisfies the followings:

*i.* $\mathbf{D}_{\text{par}}$ is a strictly upper triangular matrix.
*ii.* $\mathbf{D}_{\text{par}}^{\top} \mathbf{1} = (0, 1, \ldots, 1)^{\top}$ .

Then $G$ is a directed tree.

### Formulation of $P_{\text{sub}}(x \mid v)$ :

- $[\mathbf{D}_{\text{par}}^{k}]_{i,j}$ : the probability that there exists the path from $v_j$ to $v_i$ with exactly $k$ steps.

Example

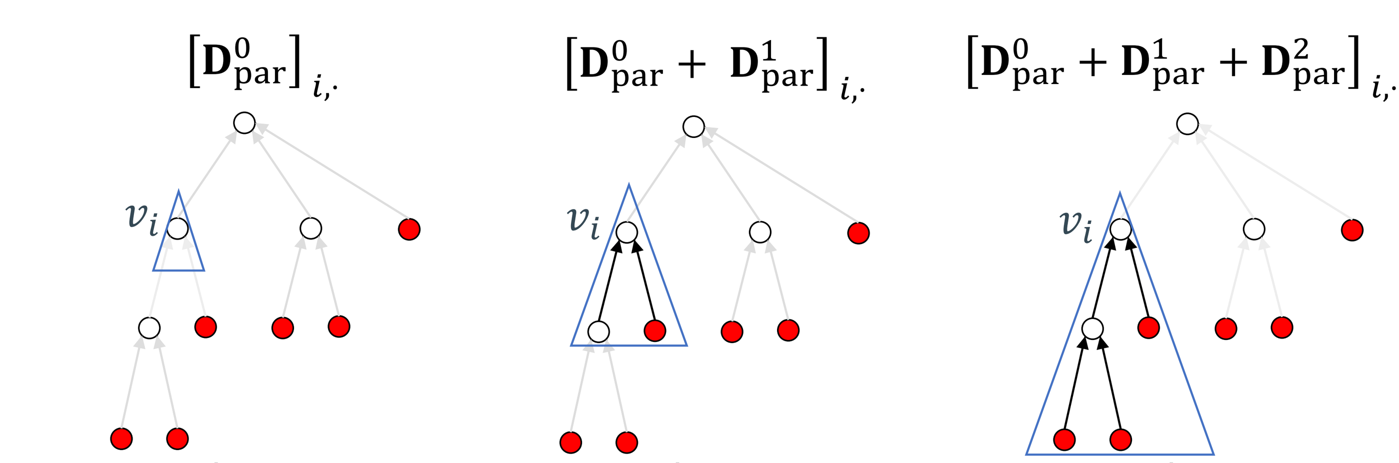


Fig.4-1 Fig.4-2 Fig.4-3

$$P_{\text{sub}}(v_j \mid v_i) = \left[ \sum_{k=0}^{\infty} \mathbf{D}_{\text{par}}^{k} \right]_{i,j} = \left[ (\mathbf{I} - \mathbf{D}_{\text{par}})^{-1} \right]_{i,j}$$

### Matrix Form Formulation :

- $\mathbf{a}_i$ and $\mathbf{a}_j$ are normalized bag-of-words.
- Batch processing

$$W_{d_T}(\mathbf{a}_i, \mathbf{a}_j) = \left| \mathbf{w}_v \circ (\mathbf{I} - \mathbf{D}_{\text{par}})^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_i - \mathbf{a}_j \end{pmatrix} \right|_1$$

## 5. Supervised Tree-Wasserstein (STW) Distance

### Loss Function :

$$L(\mathbf{D}_{\text{par}}, \boldsymbol{w}_v) = \frac{1}{|D_p|} \sum_{(i,j) \in D_p} W_{d_T}^{\text{soft}}(\mu_i, \mu_j) - \frac{1}{|D_n|} \sum_{(i,j) \in D_n} \min\{W_{d_T}^{\text{soft}}(\mu_i, \mu_j),\ m\}$$

where $D_p = \{(i,j) \mid y_i = y_j\}$ , $D_n = \{(i,j) \mid y_i \neq y_j\}$ and $\boldsymbol{w}_v = (w_{v_1}, \ldots, w_{v_N})$.

- We can minimize the loss by using the stochastic gradient descent.
- After the optimization, we select the most probable parent node for each node, and then construct a tree.

### Techniques :

- We fix the edge length $w_v$ to 1.
- We fix the tree structure of internal nodes.

## 6. Experimental Results

### Comparison Methods :

- Word Mover's Distance (WMD)
- Supervised WMD
- Quadtree
- TSW
- Flowtree

(Quadtree, TSW, Flowtree:) Unsupervised tree-Wasserstein distance based methods

### Document Classification Accuracy :

- On four datasets, STW outperforms unsupervised tree-based methods.

Table 1 : kNN test error rate.

| | TWITTER | AMAZON | CLASSIC | BBCSPORT | OHSUMED | REUTERS |
|---|---|---|---|---|---|---|
| WMD | $28.7 \pm 0.6$ | $7.4 \pm 0.3$ | $\mathbf{2.8 \pm 0.1}$ | $4.6 \pm 0.7$ | 44.5 | 3.5 |
| S-WMD | $\mathbf{27.5 \pm 0.5}$ | $\mathbf{5.8 \pm 0.1}$ | $3.2 \pm 0.2$ | $\mathbf{2.1 \pm 0.5}$ | **34.3** | **3.2** |
| Quadtree | $30.4 \pm 0.8$ | $10.7 \pm 0.3$ | $4.1 \pm 0.4$ | $4.5 \pm 0.5$ | 44.0 | 5.2 |
| Flowtree | $29.8 \pm 0.9$ | $9.9 \pm 0.3$ | $5.6 \pm 0.6$ | $4.7 \pm 1.1$ | 44.4 | 4.7 |
| TSW-1 | $30.2 \pm 1.3$ | $14.5 \pm 0.6$ | $5.5 \pm 0.5$ | $12.4 \pm 1.9$ | 58.4 | 7.5 |
| TSW-5 | $29.5 \pm 1.1$ | $9.2 \pm 0.1$ | $4.1 \pm 0.4$ | $11.9 \pm 1.3$ | 51.7 | 5.8 |
| TSW-10 | $29.3 \pm 1.0$ | $8.9 \pm 0.5$ | $4.1 \pm 0.6$ | $11.4 \pm 0.9$ | 51.1 | 5.4 |
| STW | $28.9 \pm 0.7$ | $10.1 \pm 0.7$ | $4.4 \pm 0.7$ | $3.4 \pm 0.8$ | 40.2 | 4.4 |

### Running Time :

- The Tree-Wasserstein distance is faster than the Wasserstein distance.

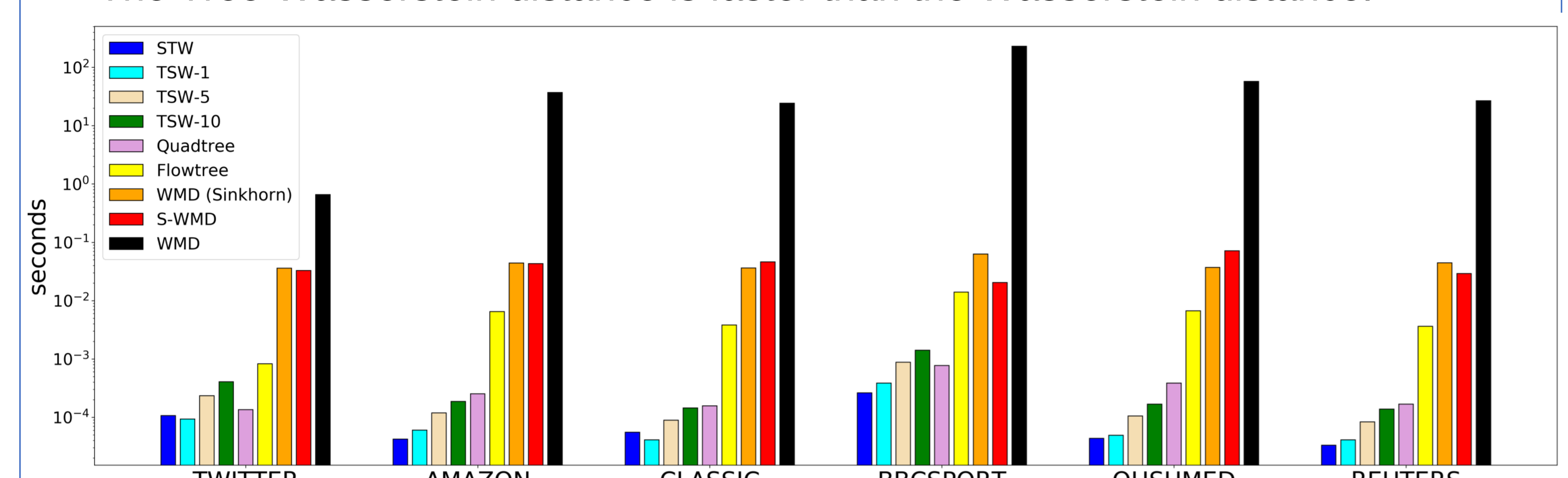


Fig.5 : Average time consumption for comparing 500 documents with one document.

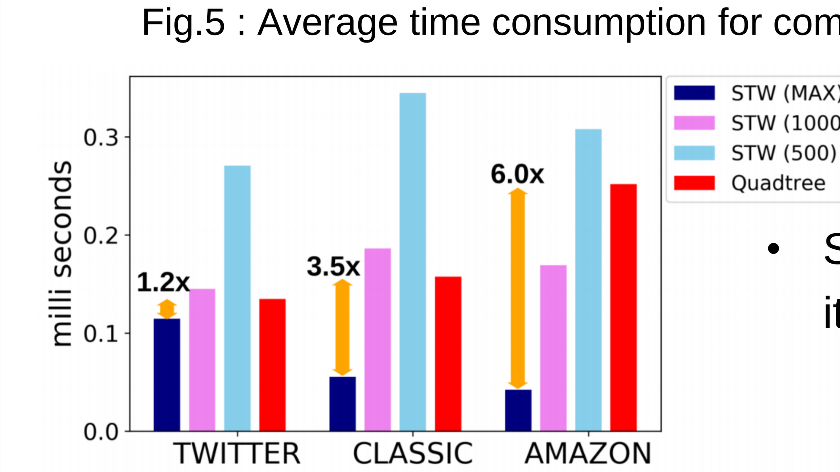


Fig. 6 : Running time varying the batch size.

- Since STW is suitable for batch processing, it can be computed faster.

Our code is available :
https://github.com/yukiTakezawa/STW