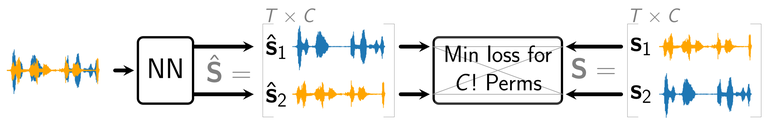
Speeding Up Permutation Invariant Training for Source Separation

Thilo von Neumann, Christoph Boeddeker, Keisuke Kinoshita, Marc Delcroix and Reinhold Haeb-Umbach
Paderborn University, Germany NTT Corporation, Japan

1 Introduction

- Permutation Invariant Training is widely used for source separation
- **Problem**: Naive PIT has factorial runtime
- **Goal**: Speed up PIT for utterance- and meeting-level separation
- **Proposed**
  - uPIT: Improve score matrix computation for Hungarian algorithm
  - Graph-PIT: Propose new algorithms to find the optimal assignment
- **Separation System**
  - Estimate C separated output signals $\hat{\mathbf{S}} \in \mathbb{R}^{T \times C}$ from mixture with neural network
  - Targets: U utterance signals $\mathbf{S} \in \mathbb{R}^{T \times U}$ (T: #time frames)
  - Utterance-level separation: number of outputs = number of utterances ($C = U$)
  - Meeting-level separation: number of outputs < number of utterances ($C < U$)

3 Utterance-level PIT (uPIT)



Problem: Find the best matching permutation (permutation matrix $\mathbf{P} \in \{0,1\}^{C \times C}$) between target utterances (speakers) $\mathbf{S}$ and outputs $\hat{\mathbf{S}}$

Naive: $\mathcal{O}(C!)$

$$\mathcal{J}^{(\text{uPIT})}(\hat{\mathbf{S}}, \mathbf{S}) = \min_{\mathbf{P} \in \mathcal{P}_C} \mathcal{L}(\hat{\mathbf{S}}, \mathbf{SP})$$

- The full loss has to be computed for each permutation ($C!$ times)

Decompose + Hungarian algorithm: $\mathcal{O}(C^3)$

The permutation problem can be solved with the Hungarian algorithm if it can be formulated with

$$\mathcal{J}^{(\text{uPIT})}(\hat{\mathbf{S}}, \mathbf{S}) = f\Big(\underbrace{\min_{\mathbf{P} \in \mathcal{P}_C} \operatorname{Tr}(\mathbf{MP})}_{\text{Solve with Hungarian Algorithm}}, \hat{\mathbf{S}}, \mathbf{S} \Big)$$

where f is strictly increasing in its first argument and $\mathbf{M} \in \mathbb{R}^{C \times C}$

- All relevant objectives can be decomposed like this

Example for $\mathcal{L}^{(\text{sa-SDR})}$ (proposed: Hungarian dot)

$$\begin{aligned}\mathcal{J}^{(\text{uPIT})} &= \min_{\mathbf{P} \in \mathcal{P}_C} -10 \log_{10} \frac{\operatorname{Tr}(\mathbf{P}^\mathsf{T} \mathbf{S}^\mathsf{T} \mathbf{S} \mathbf{P})}{\operatorname{Tr}\left((\hat{\mathbf{S}} - \mathbf{SP})^\mathsf{T} (\hat{\mathbf{S}} - \mathbf{SP})\right)} \\ &= -10 \log_{10} \frac{\operatorname{Tr}(\mathbf{S}^\mathsf{T} \mathbf{S})}{\operatorname{Tr}(\mathbf{S}^\mathsf{T} \mathbf{S}) + \operatorname{Tr}(\hat{\mathbf{S}}^\mathsf{T} \hat{\mathbf{S}}) + 2 \underbrace{\min_{\mathbf{P} \in \mathcal{P}_C} \operatorname{Tr}\left(\underbrace{-\hat{\mathbf{S}}^\mathsf{T} \mathbf{S}}_{\mathbf{M}} \mathbf{P} \right)}_{\text{solve with Hungarian alg.}}}\end{aligned}$$

2 Signal-to-Distortion-Ratio (SDR) loss

- SDR-based objectives are commonly used for source separation

Here written without PIT

Averaged SDR: average the SDRs of each output *(conventional)*

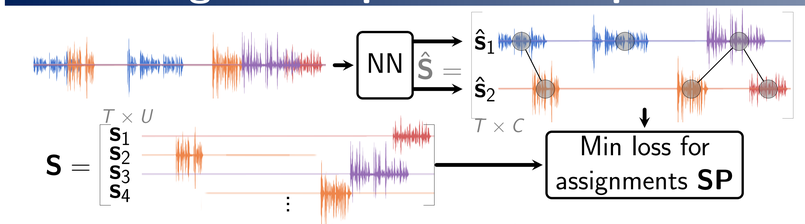
$$\mathcal{L}^{(\text{a-SDR})}(\hat{\mathbf{S}}, \mathbf{S}) = -\frac{10}{C} \sum_{c=1}^{C} \log_{10} \frac{\|\mathbf{s}_c\|^2}{\|\mathbf{s}_c - \hat{\mathbf{s}}_c\|^2}$$

Source-aggregated SDR: treat all outputs as one

$$\mathcal{L}^{(\text{sa-SDR})}(\hat{\mathbf{S}}, \mathbf{S}) = -10 \log_{10} \frac{\sum_{c=1}^{C} \|\mathbf{s}_c\|^2}{\sum_{c=1}^{C} \|\mathbf{s}_c - \hat{\mathbf{s}}_c\|^2}$$

- Required for Graph-PIT
- Neglectible improvement over $\mathcal{L}^{(\text{a-SDR})}$ in separation performance

4 Meeting-level separation: Graph-PIT



Problem: Find the best valid assignment of utterances to output channels (assignment matrix $\mathbf{P} \in \{0,1\}^{U \times C}$)

$\Rightarrow$ Color the overlap graph (vertices = utterances, edges = overlaps)

Naive: $\mathcal{O}(C(C-1)^{U-1})$ (exponential)

$$\mathcal{J}^{(\text{Graph-PIT})} = \min_{\mathbf{P} \in \mathcal{B}_{\mathcal{G},C}} \mathcal{L}(\hat{\mathbf{S}}, \mathbf{SP})$$

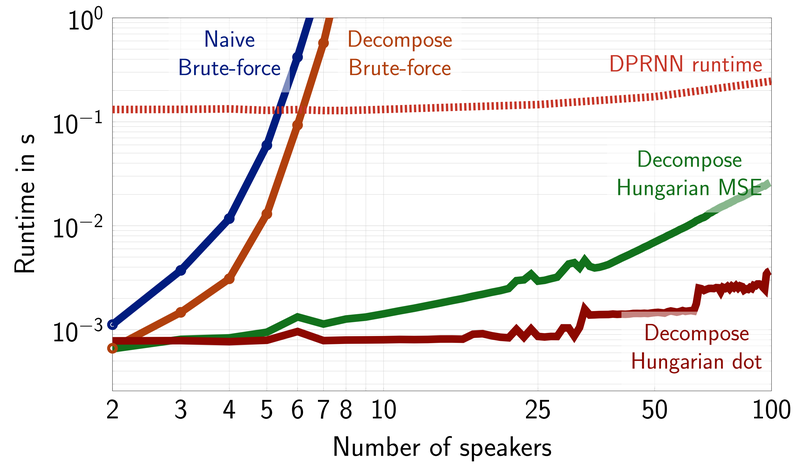
This formulation looks similar to uPIT, but

- $\mathcal{B}_{\mathcal{G},C}$ is the set of all valid colorings of the overlap graph
- $\mathbf{SP}$ no longer only represents a permutation, but can sum non-overlapping utterances

Decompose + Assignment algorithm: $\mathcal{O}(UC^{C-1})$ (linear)

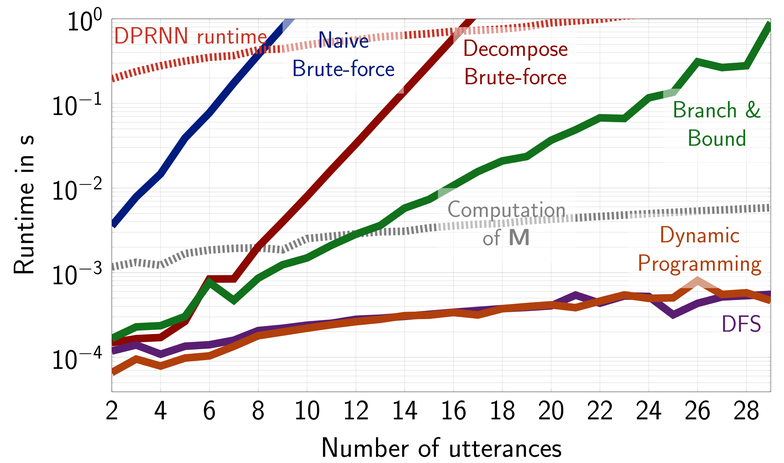
- uPIT's decomposition of $\mathcal{L}^{(\text{sa-SDR})}$ can be used (and is required!)
- Different shapes: $\mathbf{M} \in \mathbb{R}^{C \times U}$, $\mathbf{P} \in \{0,1\}^{U \times C}$
- Proposed assignment algorithms
  - **Brute-force**: Try all assignments ($\mathcal{O}(C(C-1)^{U-1})$)
  - **Greedy DFS**: Use Depth-first search in the solution space to find one (not necessarily the best) solution (best case $\mathcal{O}(CU)$, but often much slower)
  - **Branch-and-bound**: Branch-and-bound finds the best solution
  - **Dynamic Programming**: Use Dynamic Programming to elegantly traverse the solution space to find the optimal solution ($\mathcal{O}(UC^{C-1})$)

5 Runtime uPIT



- **Brute-force**: Impractical already for small numbers of speakers
- **Hungarian**: Negligible runtime compared to DPRNN separator

6 Runtime Graph-PIT Assignment



- **Naive**: Impractical already for small numbers of utterances
- **Dynamic Programming**: Optimal and as fast as a greedy approach

7 Conclusions

- General framework to speed up PIT without approximations
- Proposed new algorithms to find the best assignment for Graph-PIT
- Runtimes of optimized algorithms are negligible compared to separator

ITG 2021, Kiel

GitHub: https://github.com/fgnt/graph_pit

Computational resources were provided by the Paderborn Center for Parallel Computing.