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

3 Utterance-level PIT (uPIT)

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

Naive: $O(N!)$

- The full loss has to be computed for each permutation ($C!$ times)
- Decompose: Hungarian algorithm: $O(C^3)$

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

$$J^{(uPIT)}(\hat{S}, S) = f\left( \min_{P \in \mathcal{P} \subset \mathbb{R}^{C \times C}} \text{Tr}(MP) \right), \hat{S}, S)$$

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

- All relevant objectives can be decomposed like this
- Example for $L^{(sa-SDR)}$ (proposed: Hungarian dot)

$$J^{(uPIT)}(\hat{S}, S) = \min_{P \in \mathcal{P} \subset \mathbb{R}^{C \times C}} \text{Tr}(P S \hat{S}) - 10 \log_{10} \left( \text{Tr}(\hat{S}S - SP) / \text{Tr}(\hat{S}S) \right)$$

$$J^{(Graph-PIT)}(\hat{S}, S) = \min_{P \in \mathcal{P} \subset \mathbb{R}^{C \times C}} \text{Tr}(P S \hat{S}) - 10 \log_{10} \left( \text{Tr}(\hat{S}S - SP) / \text{Tr}(\hat{S}S) \right)$$

4 Meeting-level separation: Graph-PIT

- Required for Graph-PIT
- Neglectable improvement over $L^{(sa-SDR)}$ in separation performance

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