SDR —Medium Rare with Fast Computations

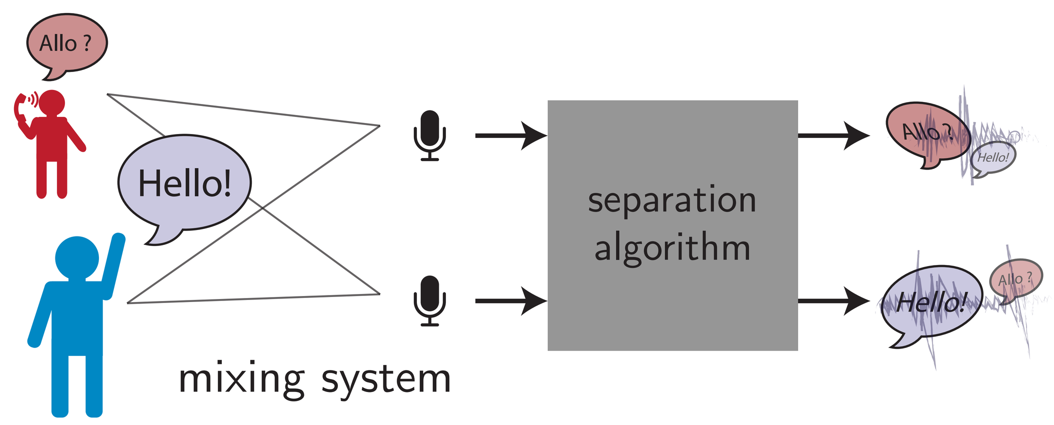
Robin Scheibler

LINE

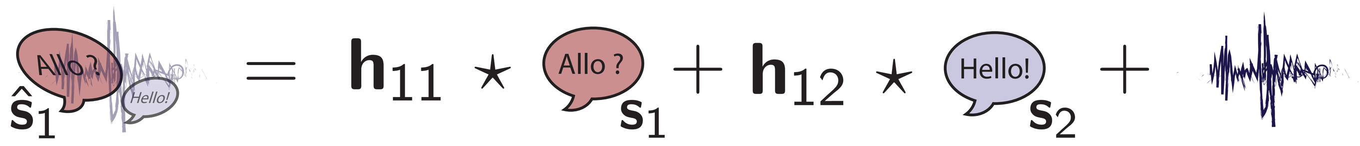
Evaluation of BSS Algorithms

Abstract —We revisit the widely used **bss eval metrics** [1] for source separation. We propose a **fast** algorithm for BSS Eval. In experiments, we assess speed and numerical accuracy. The speed-up is up to two orders of magnitude in some cases.

Signals Model

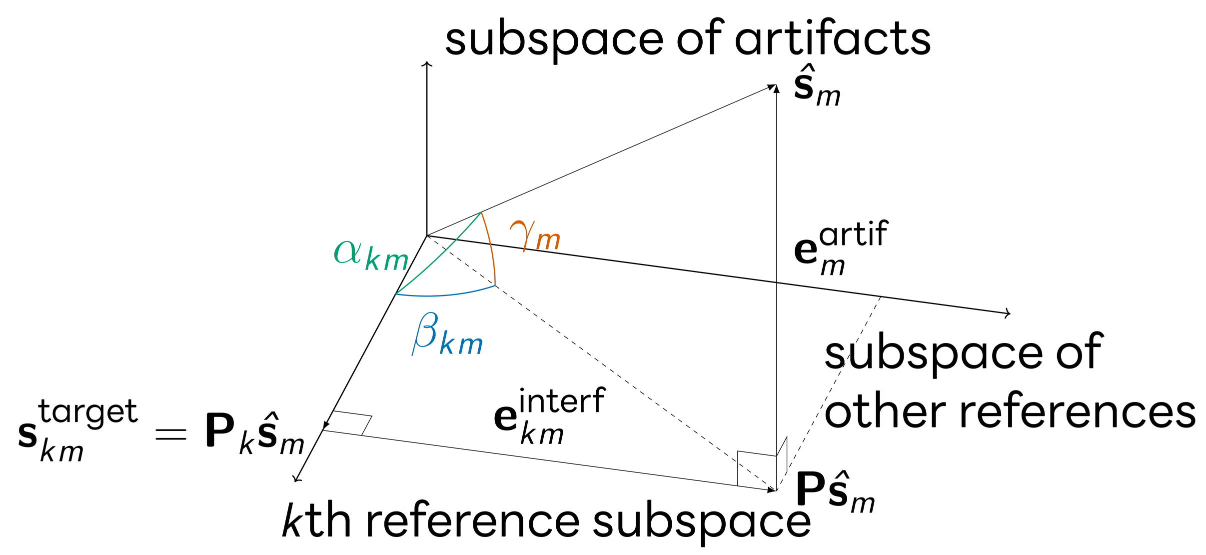


The **estimated signals** $\hat{\mathbf{s}}_m$ are convolutive mixtures of reference signals $\mathbf{s}_k$, plus an artifact term



BSS Eval Metrics

Decomposes the estimated signals in three **orthognal parts**



- $\mathbf{s}^{\text{target}}_k$: contribution of reference k
- $\mathbf{e}^{\text{interf}}_{km}$: contribution of other sources
- $\mathbf{e}^{\text{artif}}_m$: contribution of artifacts

Signal-to-Distortion Ratio (SDR)

$$\text{SDR}_{km} = 10\log_{10}\frac{\|\mathbf{s}^{\text{target}}_{km}\|^2}{\|\mathbf{e}^{\text{interf}}_{km} + \mathbf{e}^{\text{artif}}_{km}\|^2}$$

Conventional Algorithm

1. Compute statistics of ref./est. $O(M^2T\log T)$
2. Solve large linear systems $O(ML^3)/O((ML)^3)$ (by Gaussian elimination)
3. Filter signals $O(M^2T\log T)$

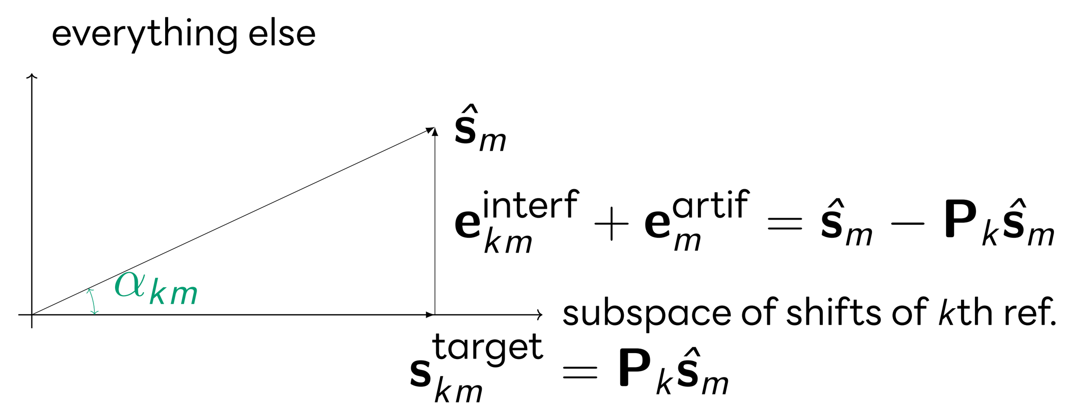
Fast BSS Eval

Key New Insight

The metrics are functions of the **subspace angles**!

$$\text{SDR}_{km} = -10\log_{10}\tan^2\alpha_{km}$$
$$\text{SIR}_{km} = -10\log_{10}\tan^2\beta_{km}$$
$$\text{SAR}_{m} = -10\log_{10}\tan^2\gamma_{km}$$

Proof (SDR)



1. Definition of cosine: $\|\mathbf{P}_k\hat{\mathbf{s}}_m\|^2 = \cos^2\alpha_{km}$
2. Pythagor: $\|\hat{\mathbf{s}}_m - \mathbf{P}_k\hat{\mathbf{s}}_m\|^2 = \|\hat{\mathbf{s}}_m\|^2 - \|\mathbf{P}_k\hat{\mathbf{s}}_m\|^2 = 1 - \cos^2\alpha_{km}$

Norm of Projection onto Shifts of $\mathbf{s}_k$

Matrix $\mathbf{A}_k$ contains shifts of $\mathbf{s}_k$ in its columns, and the matrix $\mathbf{P}_k = \mathbf{A}_k(\mathbf{A}_k^{-1}\mathbf{A}_k)^{-1}\mathbf{A}_k^\top$ projects onto the subspace they span. Then,

$$\|\mathbf{P}_k\hat{\mathbf{s}}_m\|^2 = (\mathbf{A}_k^\top\hat{\mathbf{s}}_m)^\top(\mathbf{A}_k^\top\mathbf{A}_k)^{-1}(\mathbf{A}_k^\top\hat{\mathbf{s}}_m)$$

Proposed Fast Algorithm

1. Compute $\mathbf{R}_k = \mathbf{A}_k^\top\mathbf{A}_k$ and $\mathbf{x}_{km} = \mathbf{A}_k^\top\hat{\mathbf{s}}_m$
2. Solve $\mathbf{R}_k\mathbf{h} = \mathbf{x}_{km}$, this is a **Toeplitz** system
3. Compute $\cos^2\alpha_{km} = \mathbf{x}_{km}^\top\mathbf{h}$
4. $\text{SDR}_{km} = 10\log_{10}\frac{\cos^2\alpha_{km}}{1-\cos^2\alpha_{km}}$

Fast Toeplitz Solver

The system matrix $\mathbf{R}_k$ is **Toeplitz** and can be solved quickly [3]

- Conjugate Gradient Algorithm
- Multiplication by $\mathbf{R}_k$ in $O(L\log L)$ via FFT
- Circulant pre-conditioner, also $O(L\log L)$ via FFT
- Eigenvalues cluster around 1, and converges in few iterations [3]

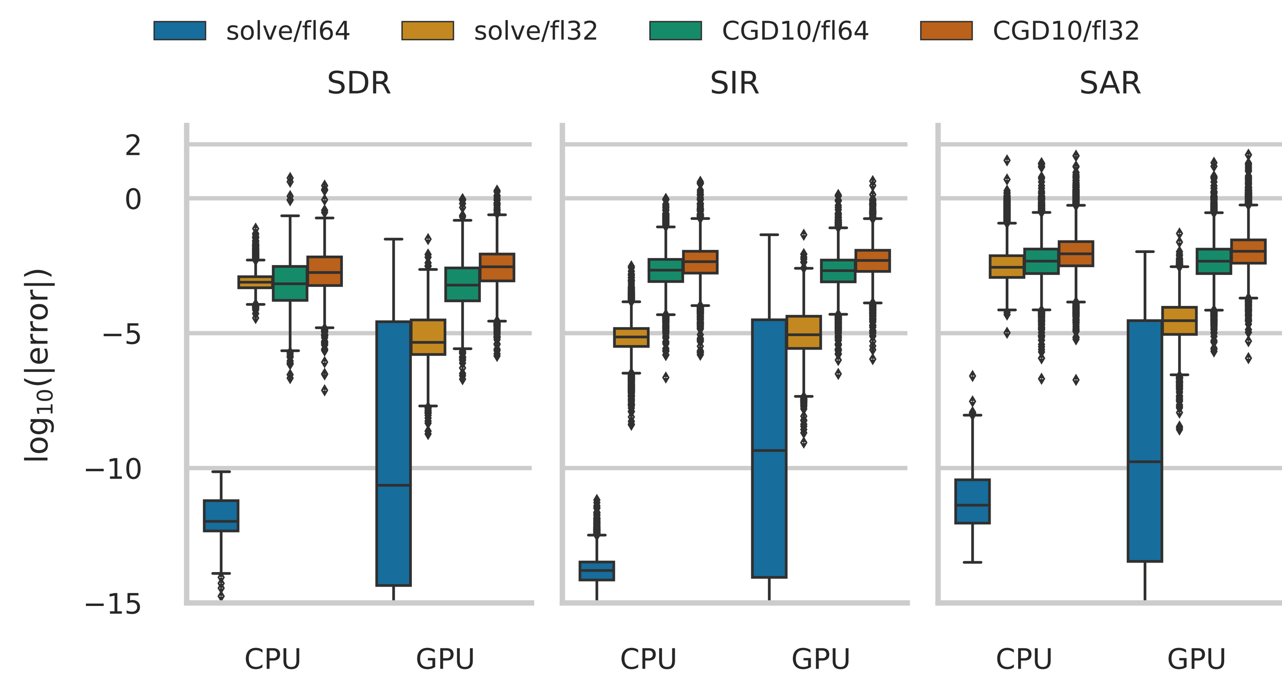
Experimental Validation

Python implementation in `fast-bss-eval` package

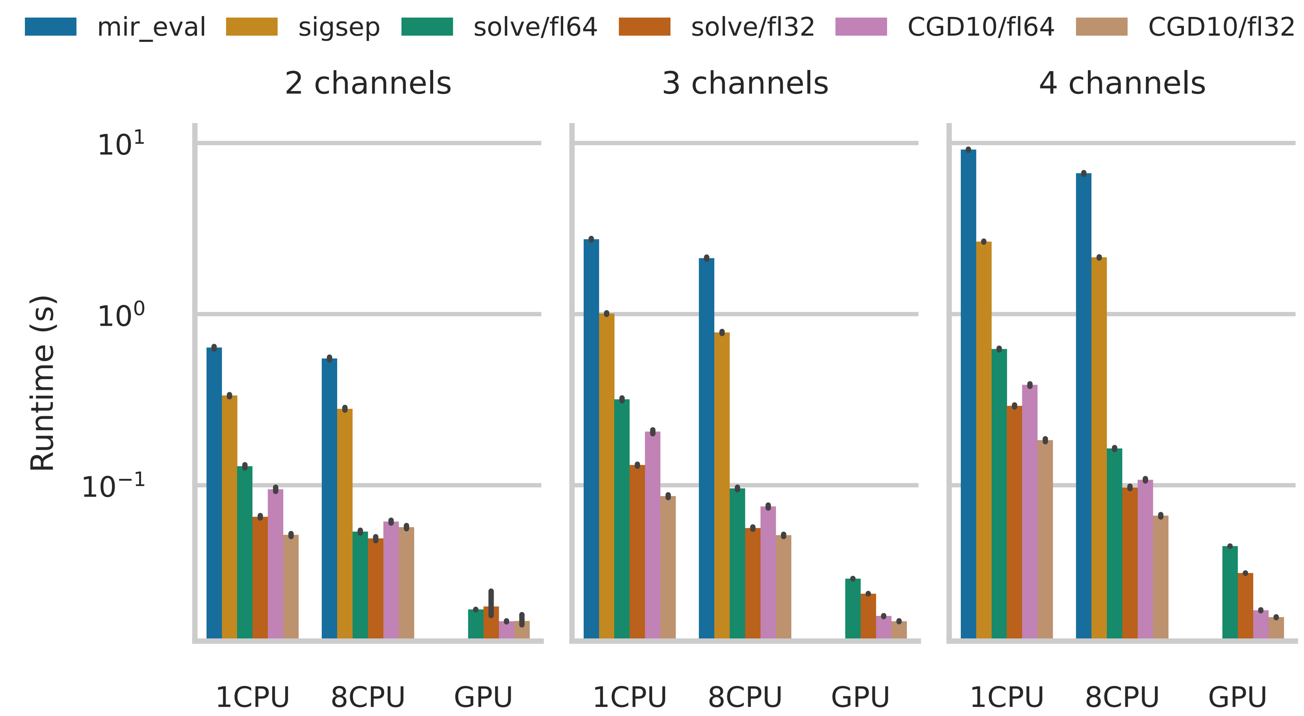
```
pip install fast-bss-eval
```

package	metrics	backend
mir_eval [4]	SDR/SIR/SAR	numpy
sigsep [5]	SDR/SIR/SAR	numpy
ci_sdr [6]	SDR	torch
fast-bss-eval	SDR/SIR/SAR	numpy/torch

Accuracy



Speed



References

[1] Vincent et al., IEEE TASLP, Jun. 2006, pp. 1464–1469.
[2] Le Roux et al., Proc. ICASSP, May 2019.
[3] Chan and Ng, SIAM Review, Sep. 1996, pp. 427–482.
[4] Raffel, Proc. ISMIR, Oct. 2014.
[5] `https://github.com/sigsep`
[6] Boeddeker et al., Proc. ICASSP, Jun. 2021.