BSS_ORACLE Toolbox User Guide

Emmanuel Vincent

Rémi Gribonval

29th August 2005, *Version* 1.0

Contents

Chapter 1

Getting started

This document is meant to help you use the `BSS_ORACLE` toolbox, which implements criteria to benchmark the best performance achievable by various classes of source separation algorithms as described in the paper [1] (in French).

1.1 License - no warranty

The toolbox is distributed under the terms of the GNU GENERAL PUBLIC LICENSE as a set of MATLAB® [1] routines so you should first get familiar with MATLAB® to use it.

1.2 Cite this as:

Within the limits of the GNU GENERAL PUBLIC LICENSE, you can use the toolbox as you please. If you use the toolbox in a work of your own that you wish to publish, please cite it as: E. Vincent and R. Gribonval, *BSS_ORACLE Toolbox User Guide Version 1.0*, 2005, URL: `http://www.irisa.fr/metiss/bss_oracle/`

1.3 Download and install

The latest version of the toolbox can be downloaded at the address

`http://www.irisa.fr/metiss/bss_oracle/bss_oracle.zip`

Once you have downloaded and uncompressed the toolbox you should get a directory called BSS_ORACLE containing the following files

- LICENSE.txt
- Contents.m
- bss_oracle_filter.m
- bss_oracle_mask.m

[1] MATLAB® is a registered trademark of The MathWorks, Inc.

- sdr.m
- mdct.m
- imdct.m

as well as the present documentation in LaTeX, PostScript and PDF

`user_guide.bib, user_guide.tex, user_guide.ps, user_guide.pdf`

1.4 Getting help

Within MATLAB® , from the directory where the toolbox files are located, you can get basic online help on the various functions of the toolbox by typing

```
help Contents
```

If you have added the toolbox directory to the MATLAB® path you can simply type

```
help BSS_ORACLE
```

1.5 Reading guide

In chapter 2 you will learn how to use the various functions of the toolbox to compute oracle estimators and performance measures for source separation. Chapter 3 gives a detailed documentation for each function of the toolbox.

Chapter 2

User guide

This document is meant to help you use the `BSS_ORACLE` toolbox, which implements criteria to benchmark the best performance achievable by various classes of source separation algorithms as described in the paper [1] (in French). The toolbox is distributed under the terms of the GNU GENERAL PUBLIC LICENSE as a set of MATLAB® [1] routines so you should first get familiar with MATLAB® to use it.

2.1 Context

The purpose of this toolbox is to evaluate the best performance achievable by a class of source separation algorithms in an evaluation framework where the original sources (or their images on the sensors) are available. In no way it provides a method to separate the sources blindly.

2.2 Principle

Let us suppose we observe a set of mixture signals (x_i) and we want to estimate the jth underlying source s_j. Most blind source separation methods belong to two classes: stationary filtering (as in ICA) and time-frequency masking (as in DUET). Within a given class, all algorithms estimate s_j as

$$\widehat{s_j} = f_{\theta_j}(x_1, \dots, x_n), \tag{2.1}$$

where θ_j represents the coefficients of the demixing filters or the coefficients of the time-frequency mask used. Algorithms differ in that they use different methods to estimate these coefficients.

Separation performance is usually limited by several factors including badly designed source models or local maxima of the function to be optimized. But also, performance may be limited by constraints on the estimate, such as the length of the demixing filters or the number of frequency bins of the time-frequency masks. The best possible source that can be estimated under these constraints (in the ideal case where source models and optimization algorithms are perfect) is called an oracle estimator of the source.

[1] MATLAB® is a registered trademark of The MathWorks, Inc.

This estimator may be computed only in an evaluation framework where the reference source s_j is available and it is defined by

$$\widetilde{s_j}(\Theta) = \arg\max_{\theta_j \in \Theta} q(f_{\theta_j}(x_1, \dots, x_n), s_j), \tag{2.2}$$

where $q(\widehat{s_j}, s_j)$ is a performance measure and Θ summarizes the constraints above. The performance of this estimator equals

$$\tilde{q}(\Theta) = q(\widetilde{s_j}(\Theta), s_j) = \max_{\theta_j \in \Theta} q(f_{\theta_j}(x_1, \dots, x_n), s_j). \tag{2.3}$$

It is a performance threshold for all algorithms of the corresponding class, and can be used to determine whether this class is suitable to separate the mixture.

This toolbox computes oracle source estimators for two classes of methods: stationary filtering of multichannel mixtures (constraint: length of the demixing filters) and time-frequency masking of single-channel mixtures using a MDCT representation with a sine window (constraint: length of the MDCT elements). The performance measure used is the Source to Distortion Ratio (SDR) $q(\widehat{s_j}, s_j) = -10 \log_{10}(\|\widehat{s_j} - s_j\|^2 / \|s_j\|^2)$. Details about the computation of oracle estimators are given in [1] (in French).

As an example, you would type

```
sej=bss_oracle_filter(X,sj,L)
```

to compute the oracle estimator `sej` of the source `sj` from the mixture signals in the rows of matrix `X` using `L`-tap stationary demixing filters. Then you would type

```
SDR=sdr(sej,sj)
```

to evaluate the performance of this estimator.

The toolbox can also readily be used to compute oracle estimates of other quantities, such as the image of the jth source on a given sensor which is a better reference when deconvolution is not required.

Bibliography

[1] E. Vincent and R. Gribonval, “Construction d’estimateurs oracles pour la séparation de sources,” in Proc. 20th GRETSI Symposium on Signal and Image Processing, 2005.

Chapter 3

Reference manual

```
bss_oracle_filter
```

Oracle estimator for source separation by stationary filtering.

Synopsis:

```
[Se,W]=bss_oracle_filter(X,S,L)
```

Inputs:

Name	Description
`X`	$n \times T$ matrix containing the mixture signals
`S`	$m \times T$ matrix containing the reference signals (*i.e.* original sources or source images on the sensors)
`L`	length of the demixing filters (in samples)

Outputs:

Name	Description
`Se`	$m \times T$ matrix containing the oracle estimates of the reference signals
`W`	$m \times n \times L$ table containing the coefficients of the corresponding demixing filters

bss_oracle_mask

Oracle estimator for source separation by time-frequency masking.

Synopsis:

```
[Se,W]=bss_oracle_mask(x,S,L)
```

Inputs:

Name	Description
`x`	$1 \times T$ vector containing the mixture signal
`S`	$m \times T$ matrix containing the reference signals (*i.e.* source images on the sensor)
`L`	size of the MDCT elements (in samples)

Outputs:

Name	Description
`Se`	$m \times T$ matrix containing the oracle estimates of the reference signals
`W`	$m \times L/2 \times nfram$ table containing the corresponding time-frequency masking coefficients

sdr

Source to Distortion Ratio.

Synopsis:

```
SDR=sdr(Se,S)
```

Inputs:

Name	Description
`Se`	$m \times T$ matrix containing the estimated signals
`S`	$m \times T$ matrix containing the reference signals

Output:

Name	Description
SDR	$m \times 1$ vector containing the SDR for each signal in deciBels (dB)

`mdct`

Modified Discrete Cosine Transform using a sine window.

Synopsis:

```
X=mdct(x,L)
```

Inputs:

Name	Description
`x`	$1 \times T$ vector containing a waveform signal
`L`	size of the MDCT elements (in samples, must be a multiple of 4)

Output:

Name	Description
`X`	$L/2 \times nfram$ matrix containing the MDCT coefficients

`imdct`

Inverse Modified Discrete Cosine Transform using a sine window.

Synopsis:

```
x=imdct(X)
```

Input:

Name	Description
`X`	$L/2 \times nfram$ matrix containing the MDCT coefficients

Output:

Name	Description
`x`	$1 \times T$ vector containing the corresponding waveform signal