User Manual for the SPRIGHT Code Package

BASiCS

April 24, 2017

1 Installation

1.1 Required software and packages

- C++ compiler supporting ISO C++ 2011 standard

1.2 Building SPRIGHT

Makefile is provided with the distribution and SPRIGHT should compile without a problem by calling `make` from the command line in the same directory of the Makefile.

2 Using the code

The SPRIGHT engine takes an input object and an output object that describes the input/output operations. The implementation of these two can be tailored to application without changing the SPRIGHT engine.

The Makefile compiles an example program `spright`. It implements an *experiment* and a *customized* mode.

The experiment mode generates random input signals according to the specifications given from the command line and tests the performance of the SPRIGHT algorithm.

Customized mode takes a text file from the command line that stores the signal and outputs the frequency contents to a desired file. The text file should have the values of the signal at each time index in tuple form (a, b), where a denotes the real part and b denotes the complex part. For example, the text file with content

```
(1,0)
(0,1)
```

```
(-1,0)
(0,-1)
```

defines a length 4 input x where $x[0] = 1$, $x[1] = i$, $x[2] = -1$, and $x[3] = -i$.

2.1 Arguments

-a
: Run experiment

-c
: Do not count the number of samples used by SPRIGHT to speed up the algorithm

-n NUM
: Signal length

-i NUM
: Number of iterations

-f FNAME
: Input file name

-z FNAME
: Output file name

-k NUM
: Number of non-zero frequencies, sparsity

-s NUM
: SNR in dB

-d NUM
: Number of delays per chain

-e NUM
: Number of chains

-l
: Use ML decoding, slow search

2.2 Examples

Below we list some example calls for the executable file.

./spright –help
Display help.

./spright -a -n 262144 -k 20 -i 30
Run 30 SPRIGHT experiments on randomly generated signals of length 262144 (2^{18}) having 20 sparse Walsh Hadamard spectrum.

./spright -a -n 16384 -s 30 -k 10 -i 30 -e 5
Run 30 SPRIGHT experiments on randomly generated signals of length 16384 (2^{14}) having an SNR of 30 dB and 10 sparse Fourier spectrum, using 5 chains of delays.

./spright -f inFile.txt -k 40 -z outFile.txt
Run SPRIGHT on input data given in `inFile.txt` to recover 40 sparse spectrum and write the recovered signal in `outFile.txt`.

./spright -a -i 100 -b "7 56 960" -n 1024 -e 5 -k 6 -s 30
Run 100 SPRIGHT experiments on randomly generated signals of length 1024 (2^10) having an SNR of 30 dB and 6 sparse Fourier spectrum. For the algorithm, use 3 stages with bins for the first 3 indices, indices 4-6, and indices 7-10, using 5 chains of delays (cf. [1]).

3 Code structure

config.cpp
This class gets input arguments from the command line and stores the parameters of the algorithm.

input.cpp
This is a template class. Instances of this class should implement a function that will return the value of the signal at a chosen time index. Two such instances of this class are the `experimentinput.cpp` and `customizedinput.cpp`.

frontend.cpp
This class is where sub-sampling and short Walsh Hadamard transforms are done.

backend.cpp
: The peeling engine is implemented here.

output.cpp
: This is a template class. Instances of this class should implement a function that will get the results from the backend and output. Two instances are `experimentoutput.cpp` and `customizedoutput.cpp`.

spright.cpp
: The class that runs the SPRIGHT engine.

main.cpp
: The implementation for the executable file with the experiment and customized modes.

References

[1] X. Li, J. K. Bradley, S. Pawar, and K. Ramchandran, "SPRIGHT: A fast and robust framework for sparse walsh-hadamard transform," *CoRR*, vol. abs/1508.06336, 2015. [Online]. Available: http://arxiv.org/abs/1508.06336