SFAmix: the sparse factor analysis model with a mixture of sparse and dense components

Chuan Gao

December 2013

1 Compilation

SFAmix is written in C++. Both the source code and a compiled binary executable are provided. SFAmix used the scientific library GSL, Eigen and Boost, all three libraries need to be installed before compilation. To compile, edit the Makefile file by changing the -L/ and -I/ line to your linker and header file location, then in the terminal,

```
$ make
```

this generates the binary executable file: SFAmix. This binary only support Linux for the moment.

2 Run SFAmix

SFAmix takes a tab or space delimited file of all numbers (strictly numbers, no headers) as its input, and writes its estimates in a specified directory. It uses the default value of $a = b = c = d = e = f = 0.5$ to recapitulate the horseshoe, and a starting value of $\alpha = \beta = 1$ for the mixing proportion of the sparse and dense components. SFAmix infer the factor numbers non-parametrically by shrinking a big starting value down. It evaluates convergence by checking the number of nonzero values for the loading matrix, and assume convergence if the number fixes for 200 iterations. It also writes parameter values to a specified directory for every 200 iterations, so that when the algorithms takes unbearably long, some intermediate values are available.

To run SFAmix, issue the following command in terminal:

```
$ SFAmix --nf 20 --y your_gene_expression_file --out dir_result --sep tab
```

where each argument is specified by a flag, more details about the flags are:

- `--nf` specifies the starting factor number, users should make it reasonably big but not too big to burden the program.

- `--y` specifies the input file.
- `--out` specifies the output directory.
- `--sep` specifies the delimiter of the file, takes two values, "space" or "tab".

The files that are written into the specified directory are:

- command.txt: a file that records the command that is given.
- LAM, THETA, DELTA, PHI, TAU, GAMMA, ETA: the MAP estimates that correspond to the parameters in the model.
- Z: a $2 \times K$ matrix indicating whether a factor is dense or sparse. For a factor, a top row value of 0 and bottom row value of 1 indicate that it is sparse, vice versa. Because the expected value of this hidden variable is used, a value in the range of [0, 1] is sometimes observed.
- EXX: the value of $\langle X^T X \rangle = \langle X \rangle^T \langle X \rangle + p * \Sigma_X$
- PSI: the diagonal values for the variance matrix $\mathbf{\Psi}$
- itr: the iteration number the program is currently at.
- final: shows if the algorithm has converged, has value of 'done' if it is converged.

3 Simulations

A toy data has been provided, where a gene expression file of $n = 200$ samples and $p = 500$ genes is simulated. Files in the data directory are:

- Y: The gene expression file
- LAM_sparse: Sparse loading matrix
- LAM_dense: Dense loading matrix
- EX_sparse: Sparse factor matrix
- EX_dense: Dense factor matrix

To run SFAmix on the data

```
mkdir result
./SFAmix --nf 50 --y ./data/Y.txt --out result --sep tab
```

4 Speed

We have implemented a few tricks to speed up the algorithm. For the provided toy data with a dimension of 200 samples and 500 genes ($\mathbf{\Psi}$ in this case is 500×500), starting from a factor number of 50, on a 3 processor Intel Xeon CPU 2.90GHz workstation, it takes $\sim$600 iterations to converge in a matter of $\sim$8 seconds. If the statically compiled binary is used directly on a different machie, the time may be slightly longer.

The time taken for the first few iterations is not in proportion to the time taken for the later iterations, as the algorithm run, it shrunk the factor numbers, so the algorithm gradually run faster.