SOFTWARE SUPPLEMENT TO 'GENERATING SPIKE TRAINS WITH SPECIFIED CORRELATION-COEFFICIENTS'

JAKOB MACKE, PHILIPP BERENS, ALEXANDER ECKER, ANDREAS TOLIAS, MATTHIAS BETHGE

The software in this package complements the paper Generating Spike Trains with Specified Correlation-Coefficients accepted for publication in Neural Computation. It contains all functions neccessary to use the techniques described in the paper. You can use them to generate samples from correlated binary random variables as you need for modeling and generating spike trains. You can use them to generate correlated spike counts with Poisson and other marginal statistics. All functions contain inline commentary and documentation.

Paper. Macke, J. H., P. Berens, A. S. Ecker, A. S. Tolias and M. Bethge: Generating Spike Trains with Specified Correlation Coefficients. Neural Computation 21(2), 397-423 (02 2009)

You can download the paper from our website at
`http://www.kyb.tuebingen.mpg.de/publication.html?publ=5157`.
If you use any function of this toolbox for you research, please cite this paper.

Before you start. Run the m-file setPath to add the relevant subdirectories to your Matlab search path. Currently, the functions require the statistics and optimization toolboxes. As you can see in the code, this can be easily remedied, should you not have them at hand.

Demo. To get familiar with the functions in this package, we suggest running the demo file included in the package. It takes you through the main functionality while reproducing some of the results and simulations shown in the paper. Inside the m-file, we describe each step in detail.

Important functions. While the software package contains a variety of functions that work "behind the scenes" there is only a limited number of important ones. For all other functions, look inside the m-files.

- sampleDichGauss01: Generate samples from a multivariate binary distribution with given mean and covariances. In the downloaded code, the resulting correlation matrix does not get computed, if the algorithm determines that it leads to a nonpermissable correlation matrix of the latent variable. You can disable this behaviour by uncommenting line 34 and putting line 33 in comments. In this case, a version of findLatentGaussian is used, which produces a correlation matrix in any case. If it is not permissable, we use an algorithm by Higham to find the closest correlation matrix.
- findLatentGaussian: Compute parameters for the hidden Gaussian variable.

- sampleCovPoisson: Generate samples from a multivariate Poisson distribution with a given mean and (positive) covariance. Uses the method described in section 3.1. of the paper.
- DGPoisson: Generate samples from a multivariate Poisson distribution with a given mean and covariance. Uses the method described in section 3.3. of the paper. Resulting distributions will have Poisson marginal statistics and the desired covariance.
- DGAnyMarginal: Generate samples from a multivariate distribution with arbitrary marginals and given covariance. Uses the method described in section 3.3. of the paper.

Disclaimer. While we hope and did our best to ensure that all the provided functions are errorfree, we of course cannot guarantee this. Therefore, we do not take responsibility for mistakes you might make in the consequence of using these functions.

Contact. You can reach us via email
berens@tuebingen.mpg.de
jakob@tuebingen.mpg.de
if you have any questions. In addition, we will post updates and news on our website www.kyb.mpg.de/bethgegroup/code/efficientsampling.

Software license. All functions are freely avaible for download under the BSD license. This means you are free to use, copy, modify and redistribute our code (see the license file on MATLAB central). This explicitly excludes all proprietary code used in the functions, which might be needed for proper functionality (Matlab toolboxes by The Mathworks). For the bivariate normal, we used an implementation by William Moranvil.