

RESPONSE TO ‘FITTING A FOLDED NORMAL DISTRIBUTION WITHOUT EM’

BY SUNGKYU JUNG¹, MARK FOSKEY² AND J. S. MARRON³

¹*Department of Statistics, Seoul National University, sungkyu@snu.ac.kr*

²*Accuray Incorporated, mark.foskey@gmail.com*

³*Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, marron@unc.edu*

We thank the Editor-in-Chief for providing us the opportunity to respond to the letter by Professor Iain MacDonald.

Professor MacDonald pointed out that MLEs of parameters for the univariate folded normal distribution can be efficiently obtained via a ready-made general-purpose optimizer, rather than deriving EM algorithms as we originally proposed. We welcome his comment and take this opportunity to numerically compare these two solutions and to raise a couple of related questions.

We have compared the folded normal MLEs obtained by our EM algorithm (implemented in Matlab) and by the `nlm` function (of R) for the 15 sets of data used in Jung, Foskey and Marron (2011). While the EM estimates are nearly identical to the `nlm`-estimates (up to the fifth digit), the computation time needed for the EM algorithm is nearly one tenth of `nlm`: 0.01 seconds for the EM and 0.13 seconds for the optimizer. This might be due to the fact that most of the data points are far from the boundary, that is, there was almost no folding with $\hat{\mu}/\hat{\sigma} \approx 15$. To mimic the situation where some folding must occur, we have translated the data toward the origin (and taken the absolute values) so that $\hat{\mu}/\hat{\sigma} \approx 1$. The results were similar; both estimates are nearly identical, but the EM algorithm took 0.01 seconds while the optimizer needed 0.11 seconds. (The data and codes for this experiment can be found in the Supplementary Material (Jung, Foskey and Marron (2020)) and also at <https://github.com/sungkyujung/EMvsNLM>.) While directly applying general-purpose optimizers can be considered as an efficient solution, carefully designed *statistical algorithms* (such as the EM algorithm) can be even more efficient.

We want to point out that a similar problem of fitting a folded normal was also considered in Eltzner, Huckemann and Mardia (2018). There, the authors considered fitting a transformed folded normal distribution; writing $f_{\mu,\sigma}(x)$ for the folded normal density, the problem of Eltzner, Huckemann and Mardia is to obtain MLE of (μ, σ) from the sample $X_i \sim g(x) = c(\mu, \sigma) \sin^d(x) f_{\mu,\sigma}(x)$. Here, d is a known natural number and $c(\mu, \sigma)$ is the normalizing constant (with no closed form expression available). Eltzner et al. did not attempt to construct an EM algorithm. A potential reason for that is the fact that an M-step does not have a closed-form solution, at least not without an ingenious formulation. For such problems, a numerical optimizer indeed provides satisfactory solutions for the likelihood problem, as Professor MacDonald suggested.

The problems considered in Jung, Foskey and Marron (2011) and Eltzner, Huckemann and Mardia (2018) are motivated by preventing overfitting of the least-square circles and spheres to directional data on the unit hypersphere. The circles and spheres on the hypersphere correspond to certain classes of curves and surfaces in \mathbb{R}^d , including straight lines and hyperplanes. The folded normal distribution was used to decide whether nonlinear dimension reduction is more advantageous than linear dimension reduction. The use of the folded normal distribution for such purposes is superseded by other proposals in Jung, Dryden and Marron (2012)