NOTES ABOUT MNPP

Each step of the likelihood estimation are done exactly like in the book from Kenneth E. Train “Discrete Choice Methods with Simulation”. The covariance matrix is normalized with each diagonal element in the “L” matrix (see book for details) equal to 1. Same thing for the element at second row and first column.

I don’t exactly know why, but almost all optimizers fail to optimize the mean-likelihood. The only one doing the job is pybrain’s Separable NES. Note you need to let it iterate long time before stopping, but you can stop the optimization when you want by keyboard abort.

The model actually works, but in practice coefficients are still noisy with around 0,01 standard deviation. Logit coefficients are almost the same and much more stable. Estimating the hessian matrix (with precision) for a stochastic likelihood function asks for huge amount of computations, so it is not implemented yet. For those reasons, I actually paused the project and looking for other models.

To get the coefficients of you regression, use the get_params method.

Note you need a lot of cores on your computer to estimate the model in a correct time, because the likelihood is evaluated in parallel.

If you have any idea about any other way to estimate the hessian matrix or anything else, please contact me. Note the first interest of this project is definitely not production but a pedagogic challenge.