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 element in the “L” matrix (see book for details) equal to 1.

This project is currently paused due to trouble with optimizers available on Python. I am not a stochastic optimization specialist but I tried almost every optimizers available and the only one doing the job is the Separable NES in the pybrain package.

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.

To get the coefficients of you regression, check your_reg.es.bestEvaluable The (J-1)*K are the beta coefficients for alternatives 1...J, with beta = 0 for the alternative 0, and the coefficients above are about the covariance matrix.

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

If you have any idea about any other way to estimate the hessian matrix or anything else, please contact me.