

Scalable Estimation and Regularization for the Logistic Normal Multinomial Model

The code is developed for the LNM and LNM+ methods. The empirical Bayes estimates of the multinomial probabilities are also computed.

Simulation

- **analysis.r**
An example for running programs in the Simulation folder.
- **plotcond.r**
Draws density plots of condition numbers for the LNM method based on 100 simulations.
- **LNM**
Given the reference category (*base*) and the count data (*x*), this function estimates μ and Σ in the LNM model.
Imports: MASS
Usage: LNM(*base*, *x*, *Iter1* = 5e3, *samplenum* = 5, *HMC_burn* = 1)
- **Prediction.LNM**
This function computes the predictive multinomial probabilities in the LNM model.
Imports: MASS
Usage: Prediction.LNM(*p*, *test*, *samplenum* = 1e3, *gap* = 2e1, *HMC_burn* = 5e2)
- **LNMP**
Given the reference category (*base*), the count data (*x*) and the threshold of the condition number (*kappa*), this function estimates ν and D in the LNM+ model.
Imports: MASS
Usage: LNMP(*base*, *x*, *kappa*, *Iter1* = 5e3, *samplenum* = 5, *HMC_burn* = 1)
- **LNMP.CV**
This function estimates ν and D in the LNM+ model with κ obtained by cross-validation.
Usage: LNMP.CV(*p*, *test*, *Iter1* = 5e3, *samplenum* = 5, *HMC_burn* = 1)
- **Prediction.LNMP**
This function computes the predictive multinomial probabilities in the LNM+ model.
Imports: MASS
Usage: Prediction.LNMP(*p*, *test*, *samplenum* = 1e3, *gap* = 2e1, *HMC_burn* = 5e2)
- **HMCsampling**
The function generates samples from the posterior density by the HMC method in the LNM model.
Usage: HMCsampling(*mu*, *sigma*, *y*, *x*, *Iter*)
- **HMCsampling.cond**
The function generates samples from the posterior density by the HMC method in the LNM+ model.
Usage: HMCsampling.cond(*mu*, *sigma*, *y*, *x*, *Iter*)
- **Accept_hmc**
Gives the probability of accepting the proposed state in the LNM model.
Usage: Accept_hmc(*x*, *M*, *ay*, *ap*, *by*, *bp*, *mu*, *sigma*)
- **Accept_hmc.cond**
Gives the probability of accepting the proposed state in the LNM+ model.
Usage: Accept_hmc.cond(*x*, *M*, *ay*, *ap*, *by*, *bp*, *nu*, *D*, *h1*)
- **norm.vec.2**
Computes the L_2 norm of a vector (*x*).
Usage: norm.vec.2(*x*)

- **cond**
Computes the condition number of a matrix (x).
Usage: `cond(x)`
- **repmat**
Returns a matrix containing $m1$ copies of Mat in the row and $n1$ copies of Mat in the column dimensions.
Usage: `repmat(Mat, m1, n1)`
- **Regcond**
Returns the result of the condition number regularized problem.
Usage: `Regcond(S, kappa)`
- **CVlist**
Gives a list for cross-validation.
Usage: `CVlist(Runs, D, Lk)`
- **CV.PAR**
Gives the estimate obtained with a certain tuning parameter κ while leaving out one group of the data.
Usage: `CV.PAR(p, testlisti, Iter1 = 5e3, samplenum = 5, HMC_burn = 1)`
- **loglikeli**
Returns the log-likelihood evaluated on one group of the data with the estimators (nu and D).
Imports: `Rmpfr, MASS`
Usage: `loglikeli(x, nu, D)`
- **Simu_data**
Given the dimension (p), sample sizes (n) and the number of the simulations ($Runs$), this function generates $Runs$ sparse count datasets and saves them separately.
Imports: `MASS, stats`
Usage: `Simu_data(p, n, Runs)`

Realdata

- **realdata.RData**
The file contains the count data and body mass index (BMI) for the real application.
- **analysis_real.r**
An example for running programs in the Realdata folder.
- **plotheatmap.r**
Produces heatmaps of estimated compositions for the gut microbiome data.
- **plotdiversity.r**
Produces boxplots of estimated diversity indices for the lean and obese groups in the gut microbiome data.
- **plotpower.r**
Draws empirical power curves of the two-sample tests after estimation using different methods for the gut microbiome data.
- **rewritedata.r**
Summarizes the estimated compositions in a file.
- **LNM**
Given the reference category ($base$) and the count data (x), this function estimates μ and Σ in the LNM model.
Imports: `MASS`
Usage: `LNM(base, x, Iter1 = 5e3, samplenum = 5, HMC_burn = 1)`

- **Prediction.LNM**
This function computes the predictive multinomial probabilities in the LNM model.
Imports: MASS
Usage: Prediction.LNM(realdata_file, samplenum = 1e3, gap = 2e1, HMC_burn = 5e2)
- **LNMP**
Given the reference category (*base*), the count data (*x*) and the threshold of the condition number (*kappa*), this function estimates ν and D in the LNM+ model.
Imports: MASS
Usage: LNMP(base, x, kappa, Iter1 = 5e3, samplenum = 5, HMC_burn = 1)
- **LNMP.CV**
This function estimates ν and D in the LNM+ model with κ obtained by cross-validation.
Usage: LNMP.CV(base, countdata, lean, obese, Iter1 = 5e3, samplenum = 5, HMC_burn = 1)
- **Prediction.LNMP**
This function computes the predictive multinomial probabilities in the LNM+ model.
Imports: MASS
Usage: Prediction.LNMP(realdata_file, samplenum = 1e3, gap = 2e1, HMC_burn = 5e2)
- **HMCsampling**
The function generates samples from the posterior density by the HMC method in the LNM model.
Usage: HMCsampling(mu, sigma, y, x, Iter)
- **HMCsampling.cond**
The function generates samples from the posterior density by the HMC method in the LNM+ model.
Usage: HMCsampling.cond(mu, sigma, y, x, Iter)
- **Accept_hmc**
Gives the probability of accepting the proposed state in the LNM model.
Usage: Accept_hmc(x, M, ay, ap, by, bp, mu, sigma)
- **Accept_hmc.cond**
Gives the probability of accepting the proposed state in the LNM+ model.
Usage: Accept_hmc.cond(x, M, ay, ap, by, bp, nu, D, h1)
- **norm.vec.2**
Computes the L_2 norm of a vector (*x*).
Usage: norm.vec.2(x)
- **cond**
Computes the condition number of a matrix (*x*).
Usage: cond(x)
- **repmat**
Returns a matrix containing *m1* copies of *Mat* in the row and *n1* copies of *Mat* in the column dimensions.
Usage: repmat(Mat, m1, n1)
- **Regcond**
Returns the result of the condition number regularized problem.
Usage: Regcond(S, kappa)
- **CVlist**
Gives a list for cross-validation.
Usage: CVlist(Runs, D, Lk)
- **CV.PAR**
Gives the estimate obtained with a certain tuning parameter κ while leaving out one group of the data.

Usage: CV.PAR(base, countdata, lean, obese, testlisti, Iter1 = 5e3, samplenum = 5, HMC_burn = 1)

- **loglikeli**

Returns the log-likelihood evaluated on one group of the data with the estimators (nu and D).

Imports: Rmpfr, MASS

Usage: loglikeli(x, nu, D)

- **Two_sample_test**

Given two multinomial probabilities (x and y), this function calculates the asymptotic p -value by the method in Cao, Lin, and Li (2018).

Usage: Two_sample_test(x, y)

- **Swap**

Given a certain percentage (*percent*) for perturbing the dataset and the number of resamples (*Runs*), this function generates *Runs* perturbed datasets and saves them separately.

Usage: Swap(realdata_file, percent, Runs)

* References

Cao, Yuanpei, Wei Lin, and Hongzhe Li. 2018. "Two-Sample Tests of High-Dimensional Means for Compositional Data." *Biometrika* 105: 115–32.