SPORM Vignette

Hua Yun Chen
Division of Epidemiology and Biostatistics,
School of Public health, University of Illinois Chicago.

1 Introduction

`SPORM` (SemiParametric Odds Ratio Model) is an R package developed by Hua Yun Chen for estimating and inferring the model parameters in the semiparametric odds ratio models (Chen, 2007, 2010, 2015, 2022). The package can be used to model univariate and multivariate outcome data, test independence and conditional independence, and fit a network model to multivariate data.

There are three basic form of the semiparametric odds ratio models. The first is for univariate outcome which is a generalization of traditional generalized linear model. Let $p(y \mid x)$ denote the density of Y given X with respect to a known measure μ. The odds ratio function of the conditional density with respect to a reference point (y_0, x_0) (Chen, 2003, 2004, 2015, 2022) is

$$\eta\{(y, y_0); (x, x_0)\} = \frac{p(y \mid x)p(y_0 \mid x_0)}{p(y \mid x_0)p(y_0 \mid x)}. \tag{1}$$

The density can be represented as

$$p(y \mid x) = \frac{\eta\{(y, y_0); (x, x_0)\}p(y \mid x_0)}{\int \eta\{(y, y_0); (x, x_0)\}p(y \mid x_0)d\mu(y)}.$$

The semiparametric odds ratio model has a parametric form for the odds ratio function $\eta\{(y, y_0); (x, x_0)\}$ and a nonparametric form for the baseline density $p(y \mid x_0)$.

The second is for the joint conditional density $p(y_1, y_2 \mid x)$, where y_1, y_2 are two vectors of outcomes and x is the covariates. The conditional odds ratio function with respective a fixed point (y_{10}, y_{20}) is defined as

$$\eta\{(y_1, y_{10}); (y_2, y_{20}) \mid x\} = \frac{p(y_1, y_2 \mid x)p(y_{10}, y_{20} \mid x)}{p(y_{10}, y_2 \mid x)p(y_1, y_{20} \mid x)}.$$

The joint conditional density can be expressed as (Chen, 2007)

$$p(y_1, y_2 \mid x) = \frac{\eta\{(y_1, y_{10}); (y_2, y_{20}) \mid x\}p(y_1 \mid y_{20}, x)p(y_2 \mid y_{10}, x)}{\int\int \eta\{(y_1, y_{10}); (y_2, y_{20}) \mid x\}p(y_1 \mid y_{20}, x)p(y_2 \mid y_{10}, x)d\mu_1(y_1)d\mu_2(y_2)}.$$

By modeling the conditional odds ratio function parametrically and the baseline functions nonparametrically, we have a semiparametric odds ratio model for the joint conditional density.

The third is for multivariate outcomes. Let the variables in a data set be divided into G groups, $y_1, \cdots, y_G$. The joint density $p(y_1, \cdots, y_G)$ can be represented as (Chen, 2010, Chen et al, 2015)

$$\frac{\prod_{k=1}^{G-1} \eta_k\{y_k; (y_{(k+1)}, \cdots, y_G) \mid y_{(k-1)0}, \cdots, y_{10}\} \prod_{k=1}^{G} dP_k(y_k \mid y_{-k0})}{\int \cdots \int \prod_{k=1}^{G-1} \eta_k\{y_k; (y_{(k+1)}, \cdots, y_G) \mid y_{(k-1)0}, \cdots, y_{10}\} \prod_{k=1}^{G} dP_k(y_k \mid y_{-k0})}, \tag{2}$$

where $\eta_k\{y_k; (y_{k+1}, \cdots, y_G) \mid (y_{10}, \cdots, y_{(k-1)0}\}, k = 1, \cdots, G-1$ are odds ratio functions with reference point $(y_{10}, \cdots, y_{G0})$, and $P_k, k = 1, \cdots, G$ are baseline conditional distribution functions. We model the odds ratio functions parametrically and the baseline functions nonparametrically.

Although more general forms of the parametric odds ratio models can be easily fit into the framework (Chen, 2022), the `SPORM` package implemented a restricted version focusing on the bilinear log-odds ratio function,

$$\log \eta\{(y, y_0); (x, x_0)\} = \sum_{j=1}^{p} \sum_{k=1}^{q} \theta_{jk}(y_j - y_{j0})(x_k - x_{k0}).$$

The `SPORM` package fits the models using three different likelihood approaches. The first is the pairwise pseudo-likelihood approach, the second is the semiparametric likelihood approach, and the third is the permutation likelihood approach.

2 Installation

If you do not have the package "devtools" installed in R, you can run the following command in R to install it,

```
install.packages("devtools")
```

Once the package "devtools" is installed or you had installed the package `devtools` in your computer previously, run

```
library(devtools)
```

to include the package in your current R session. You can now install the development version of `SPORM` from github with:

```
devtools::install_github("hychen-uic/SPORM")
```

3 Example data

The package has an internal data set called `"example.rda"`. This is a simulated data set having 400 records, 9 variables. We use this data set to show the usage of the package here.

4 Data analysis

4.1 Univariate outcome

To fit a semiparametric odds ratio model to the data with the first variable as the outcome, and the rest as covariates, we can use the pseudo-likelihood approach as follows

```
pwlkh(y=example[,1],x=example[,2:5])
[[1]]

Estimates for the odds ratio parameters.

[1] 1.14278527 0.64590193 0.01072011 1.36236201

[[2]]

Estimates of the covariance matrix of the odds ratio parameter estimator (Note
that the column and row names are not part of the output and are added manually
to clarify the meaning of the output),
```

	y vs x1	y vs x2	y vs x3	y vs x4
y vs x1	0.0099140218	0.0044773861	0.0003637713	0.008718301
y vs x2	0.0044773861	0.0036590509	-0.0002861348	0.004067566
y vs x3	0.0003637713	-0.0002861348	0.0020407331	0.002211230
y vs x4	0.0087183014	0.0040675657	0.0022112305	0.014969354

We can also use the maximum semiparametric likelihood approach to fit the model as

```
splkh(y=example[,1],x=example[,2:5])
[[1]]

Estimates for the odds ratio parameters.

[1] 1.18458388 0.66001310 0.02479448 1.40983836

[[2]]

Estimates of the covariance matrix of the odds ratio parameter estimator
```

	y vs x1	y vs x2	y vs x3	y vs x4
y vs x1	0.0091439599	0.0044717977	0.0003418976	0.008465714
y vs x2	0.0044717977	0.0038391992	-0.0003261434	0.004283063
y vs x3	0.0003418976	-0.0003261434	0.0016337451	0.001120679
y vs x4	0.0084657137	0.0042830629	0.0011206789	0.012209916

We can also use the maximum permutation likelihood approach to fit the model. This method takes more time to fit as the Monte Carlo simulation is involved in drawing the permutations.

```
pmlkh(dat=example[,1:5], group=c(1,4))
```

```
[1] "Convergence criterion is not met"
```

This method also output a draw to show the progress of the Monte Carlo estimates of the odds ratio parameters.

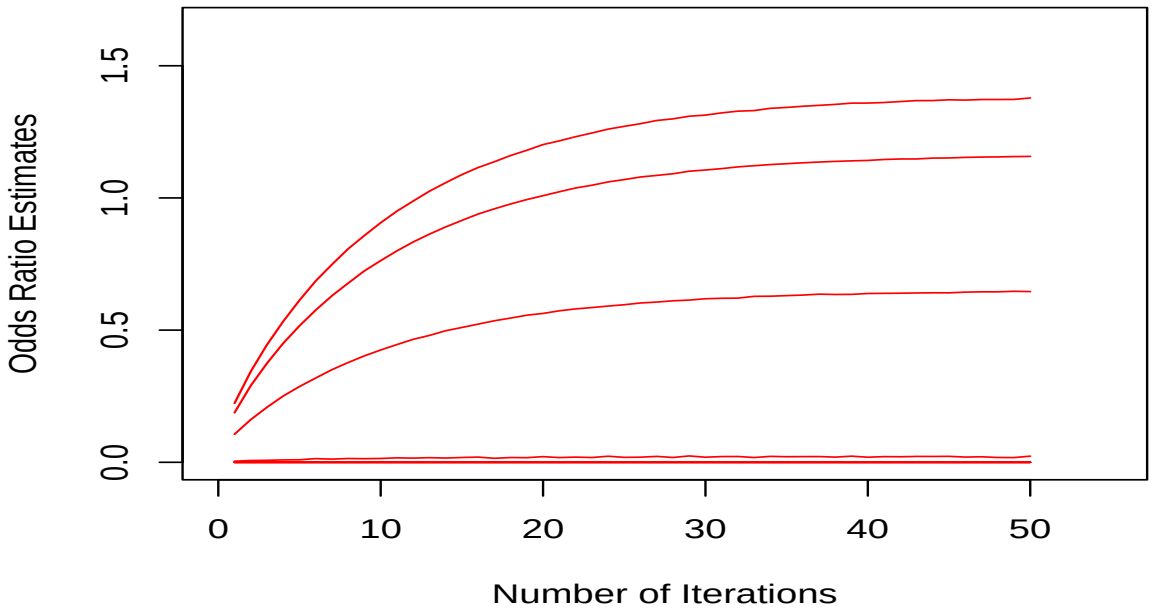

Figure 1: Convergence plot

```
[[1]]

Estimates of the odds ratio parameters.

[1] 1.14158894 0.63710827 0.02200461 1.35720951
```

```
[[2]]

Estimates of the covariance matrix of the odds ratio parameter estimator
```

	y vs x1	y vs x2	y vs x3	y vs x4
y vs x1	0.0020328782	0.0005045837	0.0002029093	0.0001033892
y vs x2	0.0005045837	0.0015735405	-0.0003931447	-0.0003831406
y vs x3	0.0002029093	-0.0003931447	0.0015936536	0.0009223340
y vs x4	0.0001033892	-0.0003831406	0.0009223340	0.0020976229

The foregoing call did not general convergent results. To solve the issue, we can increase the number of iterations in the Monte Carlo sample. The following call increases the number of iterations to 100 from the default 50.

```
pmlkh(dat=example[,1:5], group=c(1,4),niter=100)
```

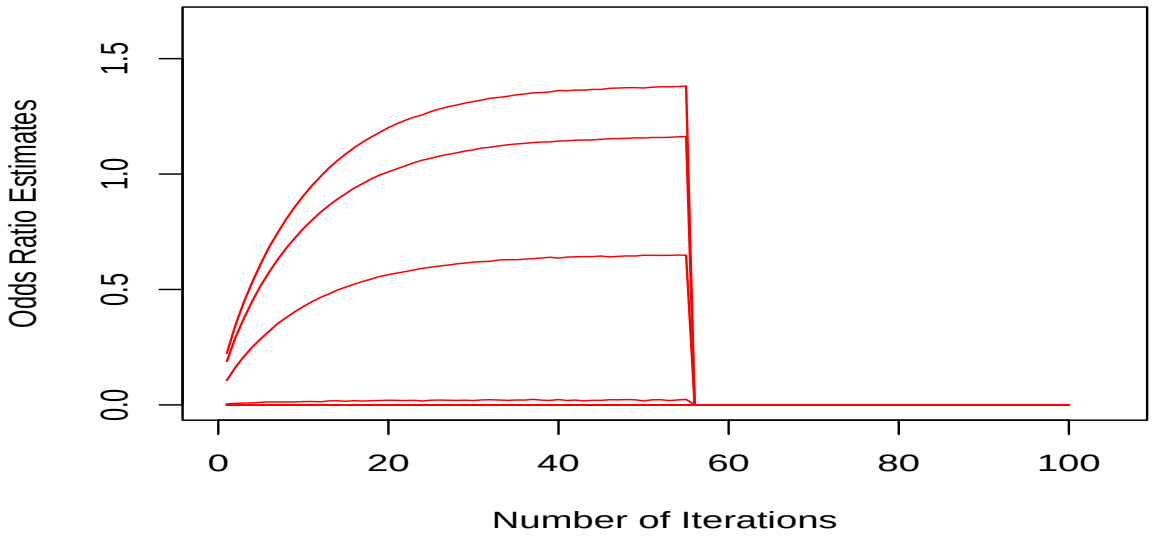

Figure 2: Convergence plot

The convergence plot shows the Monte Carlo simulation was stopped due to the convergence achieved before reaching the maximum 100 iterations.

```
[[1]]

[1] 1.15348231 0.64368657 0.02211383 1.37098804

[[2]]
```

```
   Estimates of the covariance matrix of the odds ratio parameter estimator
```

	y vs x1	y vs x2	y vs x3	y vs x4
y vs x1	2.049591e-03	0.0005100815	0.0002021779	9.977075e-05
y vs x2	5.100815e-04	0.0015900105	-0.0003984558	-3.875663e-04
y vs x3	2.021779e-04	-0.0003984558	0.0016103619	9.295271e-04
y vs x4	9.977075e-05	-0.0003875663	0.0009295271	2.114616e-03

4.2 Multivariate outcomes as a group

SPORM can also be used to fit models with multiple outcomes and multiple covariates.

```
   pwlkh(y=example[,1:3],x=example[,4:5])

   [[1]]

   Estimates for the odds ratio parameters.  The default order is columnwise
vectorization

   [1] 0.01231718 -0.07356452 0.10646882 1.39979685 -1.17124491 -0.54092314

   [[2]]

   Estimates of the covariance matrix of the odds ratio parameter estimator
```

OR	y1 vs x1	y2 vs x1	y3 vs x1	y1 vs x2	y2 vs x2	y3 vs x2
y1 vs x1	0.0017772	-0.0015223	-0.0008612	0.0013347572	-0.0011865612	-0.0001652511
y2 vs x1	-0.0015223	0.0021238	0.0008705	-0.0009367974	0.0014296050	0.0001225414
y3 vs x1	-0.0008612	0.0008705	0.0010479	-0.0005172496	0.0004943969	0.0003755057
y1 vs x2	0.0013347	-0.0009367	-0.0005172	0.0113265483	-0.0090962273	-0.0047517828
y2 vs x2	-0.0011865	0.0014296	0.0004943	-0.0090962273	0.0109320821	0.0047124454
y3 vs x2	-0.0001652	0.0001225	0.0003755	-0.0047517828	0.0047124454	0.0045266325

The same model can be fit using the semiparametric likelihood approach.

```
   splkh(y=example[,1:3],x=example[,4:5])
   [[1]]

   [1] 0.03102648 -0.08390129 0.09701887 1.40844156 -1.16948682 -0.52921929

   [[2]]
```

```
Estimates of the covariance matrix of the odds ratio parameter estimator
```

OR	y1 vs x1	y2 vs x1	y3 vs x1	y1 vs x2	y2 vs x2	y3 vs x2
y1 vs x1	0.001648	-0.00134	-0.00073	0.00119	-0.00117	-0.00014
y2 vs x1	-0.001343	0.00174	0.00074	-0.00097	0.00135	0.00020
y3 vs x1	-0.00073	0.000741	0.00086	-0.00052	0.00059	0.00040
y1 vs x2	0.00119	-0.00097	-0.00052	0.01216	-0.00997	-0.00511
y2 vs x2	-0.00117	0.00134	0.00059	-0.00997	0.01089	0.00486
y3 vs x2	-0.00014	0.00020	0.00040	-0.00511	0.00486	0.00436

The same model can be fit using the permutation likelihood approach.

```
pmlkh(dat=example[,1:5], group=c(3,2))
```

The plot monitoring the convergence of the Monte Carlo estimates is

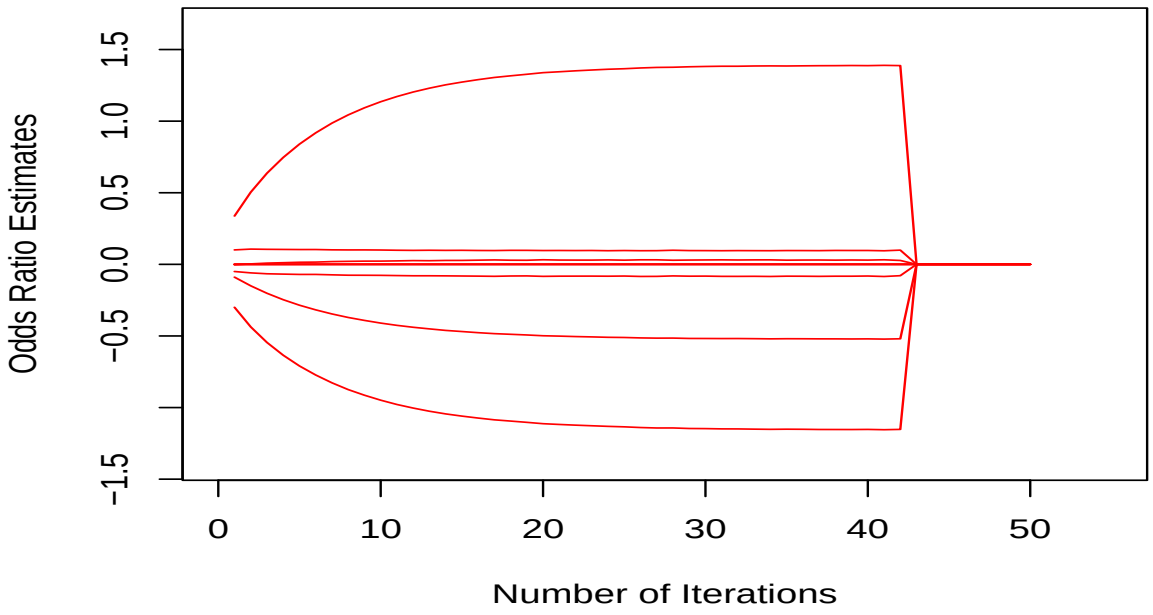

Figure 3: Convergence plot

```
[[1]]

[1] 0.03076026 -0.08299769 0.09620124 1.38122780 -1.14700781 -0.51735903

[[2]]
```

OR	y1 vs x1	y2 vs x1	y3 vs x1	y1 vs x2	y2 vs x2	y3 vs x2
y1 vs x1	0.0014706	-0.0011897	-0.0006695	0.0008131	-0.000660	-0.0003571
y2 vs x1	-0.0011897	0.0015268	0.0006952	-0.0006585	0.0008318	0.0003919
y3 vs x1	-0.0006695	0.0006952	0.0007708	-0.0003587	0.0003939	0.0004070
y1 vs x2	0.0008131	-0.0006585	-0.0003587	0.0020023 -	0.0016270	-0.0008678
y2 vs x2	-0.0006607	0.0008318	0.0003939	-0.0016270	0.0020847	0.0009151
y3 vs x2	-0.0003571	0.0003919	0.0004070	-0.0008671	0.0009157	0.0010136

4.3 Fit a network model with grouped variables

SPORM can be used to fit a network model to multivariate data with grouped variables. For example, we may divide the five variables used previously into three groups with 2, 1, 2 variables respectively. We can fit the network model by usingthe following call

```
pmlkh(dat=example[,1:5], group=c(2,1,2), niter=100)
```

The plot monitoring the convergence of the Monte Carlo estimates is

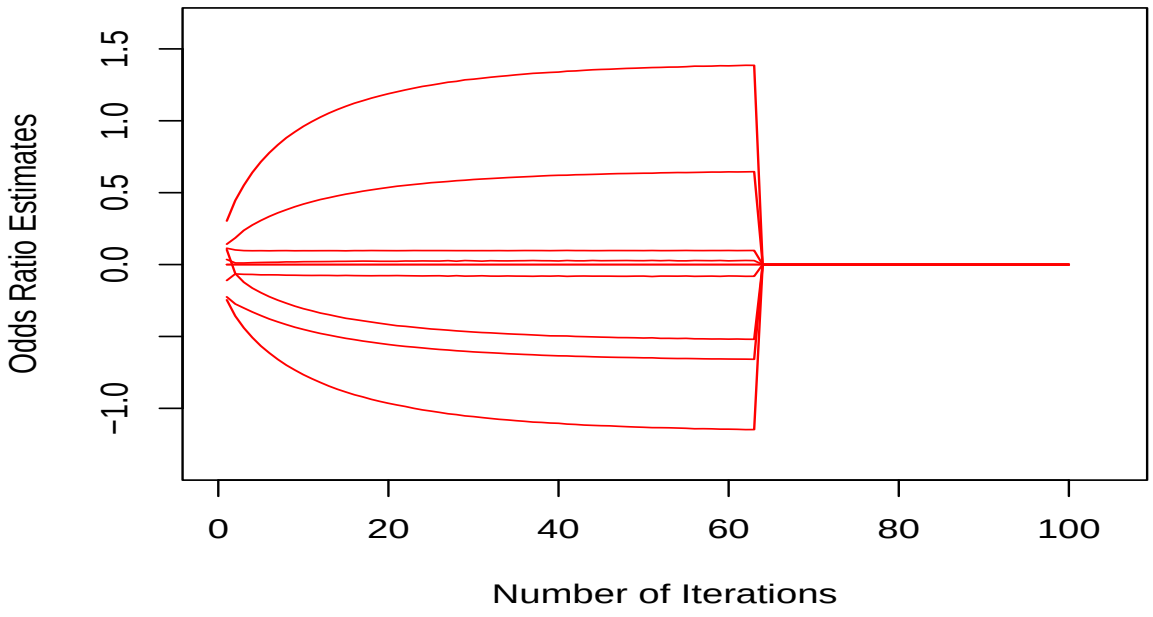

Figure 4: Convergence plot

```
[[1]]

[1] 0.638 -0.651 0.027 -0.081 1.373 -1.136 0.098 -0.513
```

```
[[2]]
```

OR	y1 vs x1	y2 vs x1	y1 vs z1	y2 vs z1	y1 vs z2	y2 vs z2	x1 vs z1	x1 vs z2
y1 vs x1	0.000695	-0.000399	-0.000210	0.000123	-0.000178	0.000106	-0.000061	-0.000728
y2 vs x1	-0.000399	0.000718	0.000126	-0.000229	0.000114	-0.000222	0.000112	0.000467
y1 vs z1	-0.000210	0.000126	0.001513	-0.001206	0.000856	-0.000674	-0.000632	-0.000136
y2 vs z1	0.000123	-0.000229	-0.001206	0.001584	-0.000675	0.000871	0.000630	0.000220
y1 vs z2	-0.000178	0.000114	0.000856	-0.000675	0.002074	-0.001664	-0.000339	-0.000704
y2 vs z2	0.000106	-0.000222	-0.000674	0.000871	-0.001664	0.002113	0.000329	0.000786
x1 vs z1	-0.000061	0.000112	-0.000632	0.000630	-0.000339	0.000329	0.000770	0.000484
x1 vs z2	-0.000728	0.000467	-0.000136	0.000220	-0.000704	0.000786	0.000484	0.001787

The functions `pwlkh, splkh` can also be used to fit the same models. But multiple calls are needed to fit network models.

5 Perform test of independence

SPORM can be used to test conditional independence of y and x given z, where y, x, z can all be vectors. Let the data matrix with the sequence of variables in the order $(y_1, y_2, x_1, x_2, x_3, z_1, z_2, z_3, z_4)$. To make the tests more flexible, we assume some of the odds ratio parameters can be preset to 0. The joint model have the fixed structure matrix as

$$M = \begin{pmatrix} \mathbf{0} & \mathbf{0} & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 1 & 1 & 0 \\ 1 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & 0 & 0 & 0 \\ 0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 1 & 1 & 0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & 1 & 1 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

where the bold face blacks show the variable groups, and 1 means the corresponding odds ratio is not preset to 0. Let the null model have the fixed structure matrix as

$$M_0 = \begin{pmatrix} \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 1 & 1 & 0 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 1 & 1 & 0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & 1 & 1 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

For the prospective likelihood ratio test using the semiparametric likelihoods, first compute the loglikelihood for y given x, z under the alternative hypothesis by

```
fit1=splkhfix(y=example[, 1:2],x=example[,3:9],fixstruct=M[1:2,3:9]).
```

The maximum log-likelihood value in the output list is `fit1[[3]]`. Next, compute the loglikelihood for $y \mid x, z$ under the null hypothesis by

```
fit0=splkhfix(y=example[, 1:2],x=example[,3:9],fixstruct=M0[1:2,3:9]).
```

The maximum log-likelihood value is `fit0[[3]]`. The likelihood ratio test statistic can be computed as

```
2(fit1[[3]]-fit0[[3]])
```

which asymptotically follows a χ^2 with degree of freedom 4, the non-zero cells in the matrix $M[1:2, 3:5]$.

For the retrospective likelihood ratio test using the semiparametric likelihoods, first compute the loglikelihood for $x \mid y, z$ under the alternative hypothesis by

```
fit1=splkhfix(y=example[, 3:5],x=cbind(example[,1:2], example[,6:9]),
              fixstruct=cbind(M[3:5,1:2], M[3:5, 6:9])).
```

The maximum log-likelihood value in the output is `fit1[[3]]`. Next, compute the loglikelihood for $y \mid x, z$ under the null hypothesis by

```
fit0=splkhfix(y=example[, 3:5],x=cbind(example[,1:2], example[,6:9]),
              fixstruct=cbind(M[3:5,1:2], M[3:5, 6:9])).
```

The maximum log-likelihood value is `fit0[[3]]`. The likelihood ratio test statistic can be computed as

```
2(fit1[[3]]-fit0[[3]])
```

which asymptotically follows a χ^2 with degree of freedom 4, the non-zero cells in the matrix $M[3:5, 1:2]$.

For the joint likelihood ratio test, since the current `SPORM` package does not support the joint model estimated by maximizing the joint semiparametric likelihood due to its computation burden, the test cannot be performed using the semiparametric likelihood. However, the joint likehodd ratio test can be performed in `SPORM` using the permutation likelihood approach. The joint model under the alternative hypothesis can be fit by

```
fit1=pmlkhfix(dat=example,group=c(2,3,4), fixstruct=M).
```

The joint model under the null hypothesis can be fit by

```
fit1=pmlkhfix(dat=example,group=c(2,3,4), fixstruct=M0).
```

The likelihood ratio test statistic can be computed as

```
2(fit1[[3]]-fit0[[3]])
```

which asymptotically follows a χ^2 with degree of freedom 4, the non-zero cells in the matrix $M[3:5, 1:2]$. In addition, the prospective likelihood ratio test and the retrospective likelihood ratio test using the permutation likelihood approach can be obtained similarly as in the case of semiparametric likelihood approach.

6 Network detection with SPORM

The syntax for the maximum penalized pairwise pseudo-likelihood approach for node-wise detection of network connections is

```
pwpenlkh(example, group, lambda, niter, eps),
```

where `dat` denotes the data matrix, `group` denote the variable groups in the model, `lambda` denotes the penalty parameter values. One network is determined for each value of `lambda`. `niter` sets the maximum number of iterations and `eps` is the convergence criterion. The output of the function is primarily a series of matrices representing structures of the network detected, each corresponding to one penalty value given in `lambda`. Possible status of the network connections are coded as follow: 0 means not connected; 1 means weakly connected, i.e., in two different node-wise models, one detected as connected and the other detected as unconnected; 2 denotes strongly connected, i.e., both node-wise models detected as connected; A `NA` status means the connection is not included in the consideration of the detection algorithm. Under the semiparametric odds ratio model, only connections between groups of variables are modeled and detected. Connections within a group are not modeled or detected. If all the connections between pairs of nodes are considered, the group is a vector of length equal to the number of variables and the number of variables in each group is 1. Only the diagonal elements are `NA`s. The output of a set of selected network is the second object in the output list. The total number of networks detected is the same as the length of the vector of the penalty parameter `lambda`.

The syntax for the maximum penalized semiparametric-likelihood approach for node-wise detection of network connections is

```
sppenlkh(example, group, lambda, niter, eps),
```

The argument of this function are the same as those in `pwpenlkh`. The output is also the same.

The syntax for the maximum penalized permutation likelihood approach for joint selection of network connections is

```
pmpenlkh(example, group, lambda, niter, eps, nburnin, nsamp, nintv,
                                    maxcyc).
```

In addition to the arguments similar to the previous two functions, four other arguments are used for controlling the Monte Carlo sample. `nburnin` specifies the length of the sampling sequence to be run before the samples are taken. `nsamp` is the sample size to be taken. `nintv` is the length of interval between two samples, and `maxcyc` is the cycle length of the candidate permutation move taken in the Metropolis sampling algorithm. Default values are given so that a user does not have to choose them.

For a series of networks corresponding to different penalty values, which network is chosen can be done by the R function `networkselect`. It can be used to fit a pre-determined set of networks using a particular model-fitting method and to select a particular network from the set of networks based on a given criterion. The function has the following form

```
networkselect(example, group, network, lambda, method, criterion),
```

where `dat` denotes the data used for the network selection, `group` specifies the number of clusters and the cluster sizes, `network` contains the network structures corresponding to `lambda` values determined by one of the previous three penalized network selection approaches specified in `method`. Those methods are the pairwise likleihood (`pw`), the semi-parametric likelihood (`sp`), or the permutation likelihood (`pm`). The `criterion` specifies the model selection approach such as the BIC. The output of the function is the selected network structure.

To illustrate the use of the R functions for network selection, let the variable clusters be `group=c(2,3,4)`, and `lambda=c(10, 50, 200, 500, 800)`. Using the penalized pairwise pseudo-likelihood approach to select the network, apply the following statement

```
fit1=pwpenlkh(dat=example,group=group,lambda=lambda).
```

The selected networks can be found in

```
network=fit1[[4]].
```

The five selected networks corresponding respectively to the five penalty values are

$$\texttt{network[,,1]} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ \mathbf{0} & \mathbf{0} & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2 & 2 & 2 & 2 \\ 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2 & 2 & 2 & 2 \\ 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 2 & 2 & 2 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 2 & 2 & 2 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 2 & 2 & 2 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

$$\texttt{network[,,2]} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ \mathbf{0} & \mathbf{0} & 2 & 2 & 0 & 0 & 2 & 2 & 2 \\ 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2 & 0 & 2 & 2 \\ 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2 & 2 & 2 & 2 \\ 2 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2 & 2 & 2 & 2 \\ 2 & 0 & 2 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 2 & 2 & 0 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 2 & 2 & 2 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 2 & 2 & 2 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

$$\texttt{network[,,3]} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & 2 & 2 & 1 & 1 & 2 & 2 & 2 \\ \mathbf{0} & \mathbf{0} & 2 & 0 & 0 & 0 & 2 & 2 & 2 \\ 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2 & 0 & 2 & 2 \\ 2 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2 & 2 & 2 & 2 \\ 1 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 2 & 2 & 2 \\ 1 & 0 & 2 & 2 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 2 & 2 & 0 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 2 & 2 & 2 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 2 & 2 & 2 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

$$\texttt{network[,,4]} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2 & 0 & 0 & 0 \\ 1 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 2 & 2 & 0 & 2 \\ 1 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 2 \\ 1 & 0 & 2 & 2 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & 0 & 0 & 2 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & 0 & 0 & 2 & 2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

and

$$\texttt{network[,,5]} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 2 \\ 1 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & 2 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

Use the network detected as input to call the function `networkselect` as follows

```
fit2=networkselect(dat=example,group=group,lambda=lambda,network=network,
                   method='pw',criterion='BIC').
```

The output provides the selected network as `fit2[[2]]` is the same as `network[,,1]`. The BIC vector ouput is $(15275.89, 20130.66, 26808.80, 59871.69, 171961.83)$.

To select the network using the penalized semiparametric likelihood, similar statements can be applied.

```
fit1=sppenlkh(dat=example,group=group,lambda=lambda).
                   network=fit1[[4]].
fit2=networkselect(dat=example,group=group,lambda=lambda,network=network,
                   method='sp',criterion='BIC').
```

For selection using the permutation likelihood approach, the following statements are applied

```
fit1=pmpenlkh(dat=example,group=group,lambda=lambda,nsamp=2e4,niter=50),
                   network=fit1[[4]],
fit2=networkselect(dat=example,group=group,lambda=lambda,network=network,
                   method='pm',criterion='BIC').
```

The difference of the penalized permutation likelihood approach from the other two approaches is that the permutation likelihood approach uses the joint likelihood and is not a node-wise (or block-wise) selection. The selected network coded as either 0 or 1 only.

7 Baseline function estimation

The functions `pwlkh, pmlkh, splkh, pwlkhfix, pmlkhfix, splkhfix` all produce odds ratio parameter estimates. Once estimates of the odds ratio parameters are obtained, we may need to find the baseline function estimates, this can be done using the `baseline` function. Two approaches may be used to obtain the baseline function estimates. One is the odds ratio function weighted estimator (Chen, 2022), i. e.,

$$p(Y = Y_i \mid X = x_0) = \frac{\eta^{-1}(Y_i, X_i \mid \hat{\theta})}{\sum_{i=1}^{n} \eta^{-1}(Y_i, X_i \mid \hat{\theta})}.$$

The other is the iterative maximum profile likelihood estimator (Chen, 2015), i.e.,

$$p(Y = Y_i \mid X = x_0) = \frac{\eta^{-1}(Y_i, X_i \mid \hat{\theta}) \int \eta(y, X_i \mid \hat{\theta}) p(y \mid x_0) dy}{\sum_{i=1}^{n} \eta^{-1}(Y_i, X_i \mid \hat{\theta}) \int \eta(y, X_i \mid \hat{\theta}) p(y \mid x_0) dy}.$$

The function `baseline` is used to compute the estimates. We can first compute the odds ratio parameter estimator. For example,

```
fit=pwlkh(example[,1:2],example[,3:5])
```

The odds ratio parameters estimate is in `fit[[1]]` in the columnwise vectorized form as

```
fit[[1]]
[1] 0.681193318 -0.672439199 0.004218654 -0.075546032 1.420299289
[6] -1.206979077
```

To make it in the correct matrix form, we may set `parm=matrix(fit[[1]]. nrow=2)` as

```
parm=matrix(fit[[1]], nrow=2)
parm
        [,1]       [,2]       [,3]
[1,] 0.6811933 0.004218654 1.420299
[2,] -0.6724392 -0.075546032 -1.206979
```

We can now call the baseline function to get the baseline function estimates as

```
fit1=baseline(example[,1:2],example[,3:5],parm)
```

The output includes `fit1[[1]]` containing the baseline function estimate using the weighting approach and `fit[[2]]` containing the Y values corresponding to the baseline estimates. To use the iterative likelihood approach to estimating the baseline function, we need to specify the method in the call as

```
fit1=baseline(example[,1:2],example[,3:5],parm,method="iterate")
```

We can also fit a model with some of the odds ratio parameter value fixed using `pwlkhfix`. For example,

```
structure=cbind(c(0,0),matrix(rep(1,4),ncol=2))
```

```
structure
[,1] [,2] [,3]
[1,] 0 1 1
[2,] 0 1 1
parm=cbind(c(0.7, -0.7), matrix(rep(0,4), ncol=2))
parm
[,1] [,2] [,3]
[1,] 0.7 0 0
[2,] -0.7 0 0
```

They set up the fixed structure of the model and the parameter values. Note that the fixed parameter values $(0.7, -0.7)$ are different from 0.

```
fit2=pwlkhfix(example[,1:2],example[,3:5], fixstruct=structure,
                            theta=parm)
```

```
fit2[[1]]
[1] 0.700000000 -0.700000000 0.003441385 -0.074155927 1.442560717
[6] -1.236040496
parma=matrix(fit2[[1]], nrow=2)
parma
      [,1]      [,2]      [,3]
[1,] 0.7 0.003441385 1.442561
[2,] -0.7 -0.074155927 -1.236040
```

We can then fit a model to estimate the baseline function with some odds ratio parameter values fixed as

```
fit3=baseline(example[,1:2],example[,3:5],parm=parma)
```

The baseline function estimate is in `fit3[[1]]`. In the foregoing demonstration, we can change the pairwise pseudo-likelihood method into semiparametric likelihood approach, i.e., use `splkh, splkhfix` in place of `pwlkh, pwlkhfix`.

References

1. Chen, H. Y. (2003). A note on prospective analysis of outcome-dependent samples. *Journal of the Royal Statistical Society, Ser. B*, 575-584.

2. Chen, H. Y. (2004). Nonparametric and semiparametric models for missing covariates in parametric regression. *Journal of the American Statistical Association,* 99, 1176-1189.

3. Chen, H. Y. (2007). A semiparametric odds ratio model for measuring association. *Biometrics*, **63**, 413-421.

4. Chen, H. Y. (2010). Compatibility of conditionally specified models. *Statistics & Probability Letters*, **80**, 670-677.

5. Chen, H. Y. (2015). A note on convergence of an iterative algorithm for the semiparametric odds ratio model. *Biometrika*, **102**, 747-751.

6. Chen, H. Y. (2022). *Semiparametric Odds Ratio Model and its Applications.* Chapman and Hall/CRC Press, Boca Raton, FL.

7. Chen, H. Y., Rader, D.E., Li, M. (2015). Likelihood inferences on semiparametric odds ratio model. *Journal of the American Statistical Association*, **110**, 1125-1135.