USING THE CORRBIN PACKAGE FOR NONPARAMETRIC ANALYSIS OF CORRELATED BINARY DATA

ANIKO SZABO

1. Overview

2. Data input

All the analysis functions in the package work on `CBData` objects, so we start by setting up the data in the format needed for analysis. The Shell toxicology data set is available in the CBData format in the package, however we will load it from a text file using the `read.CBData` function to show more typical usage. The "ShellTox.txt" file contains four space-delimited columns (other delimiters can also be used). The first column contains an integer 1 – 4 giving the treatment group, the second column gives the size of the cluster, the third the number of responses in the cluster, and the last gives the number of times the given combination occured in the data.

```
> sh <- read.CBData("ShellTox.txt", with.freq = TRUE)
> levels(sh$Trt) <- c("Control", "Low", "Medium", "High")
> str(sh)
Classes ŚCBDataŠ and 'data.frame':        67 obs. of  4 variables:
 $ Trt        : Factor w/ 4 levels "Control","Low",..: 2 3 1 2 3 4 2 4 1 2 ...
 $ ClusterSize: num  1 3 4 4 4 4 5 5 6 6 ...
 $ NResp      : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Freq       : int  1 1 1 1 1 1 1 1 2 3 ...
```

Alternatively, if the data is already in a data frame, the `CBData` function can be used to define the roles of the variables.

3. Marginal compatibility

A basic assumption of all the following analyses is that of *marginal compatibility* (MC), which states that the size of the cluster has no effect on either the marginal probability of response, or the correlation (any order) within the cluster. We can test for marginal compatibility:

```
> mc.test.chisq(sh)
$overall.chi
[1] 4.017923

$overall.p
[1] 0.4035857

$individual
$individual$chi.sq
cbdata$Trt
   Control        Low     Medium       High
0.46055742 2.04650267 0.04703645 1.46382641

$individual$p
cbdata$Trt
  Control       Low    Medium      High
0.4973636 0.1525563 0.8283027 0.2263223
```

Neither the overall p-value of 0.4, or the individual treatment group p-values show evidence of deviation from marginal compatibility.

Now we can obtain non-parametric estimates of the distribution of the number of responses in the cluster under MC:

```
> require(lattice)
> f1 <- xyplot(Prob ~ NResp | factor(ClusterSize), groups = Trt, data = sh.mc,
+     subset = ClusterSize > 0 & ClusterSize < 13, type = "l", as.table = TRUE,
+     auto.key = list(columns = 4, lines = TRUE, points = FALSE), xlab = "Number of responses",
+     ylab = "P(R=r|N=n)")
> print(f1)
```

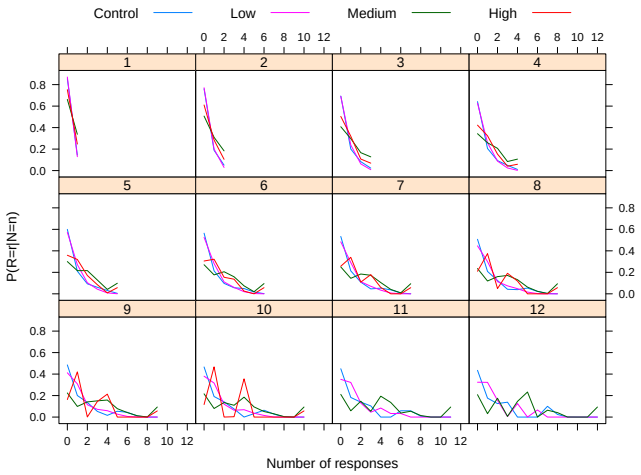

FIGURE 1. Density function of number of responses under MC by cluster-size estimated separately for each treatment group

```
> sh.mc <- mc.est(sh)
```

The `mc.est` functions gives estimates for all cluster-sizes, due to the marginal compatibility assumption the estimates $\pi_{r,M}$ for the largest cluster-size M determine all the other estimates:

$$\pi_{r,n} = \sum_{t=0}^{M} h(r,t,n)\pi_{t,M}, \tag{1}$$

where $h(r,t,n) = \binom{t}{r}\binom{M-t}{n-r}/\binom{M}{n}$ is the hypergeometric density function. Figure 1 shows the estimates.

The density functions in Figure 1 are difficult to compare (there is no obvious shift); distribution functions often provide a cleaner comparison, so they are plotted in Figure 2.

```
> panel.cumsum <- function(x, y, ...) {
+     x.ord <- order(x)
+     panel.xyplot(x[x.ord], cumsum(y[x.ord]), ...)
+ }
> f2 <- xyplot(Prob ~ NResp | factor(ClusterSize), groups = Trt, data = sh.mc,
+     subset = ClusterSize > 0 & ClusterSize < 13, type = "s", panel = panel.superpose,
+     panel.groups = panel.cumsum, as.table = T, auto.key = list(columns = 4,
+         lines = T, points = F), xlab = "Number of responses", ylab = "Cumulative Probability R(R>=r|N=n)",
+     ylim = c(0, 1.1))
> print(f2)
```

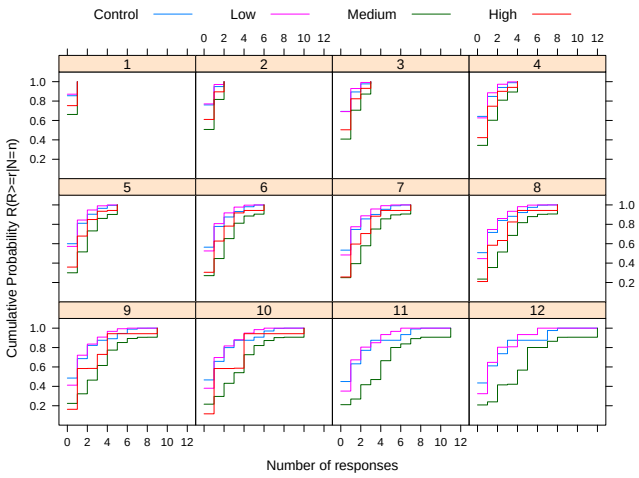

FIGURE 2. Distribution function of number of responses under MC by cluster-size estimated separately for each treatment group