Notes based on Fast HHG Supplemental

minP K-sample univariate test

We have N independent realizations $(x_1, y_1), \ldots, (x_N, y_N)$

Formally, for N observations, there are

$\binom{N+1}{2}$ possible cells, and $\binom{N-1}{m-1}$ possible partitions of the observations into m cells.

Perform the test statistics on the ranked observations for ease:

$rank(Y) \in \{1, \ldots, N\}$

Let Π_m denote the set of partitions into m cells.

Within each partition L, for a cell C in the set of m cells defined by the partition,

Oc(g) = observed counts for distribution g within {1….K}

Ec(g) = expected counts for distribution g within {1….K}

Ec(g) can be calculated by:

$$Ec(g) = width\ of\ cell\ C * \frac{Ng}{N}$$

$$e_{[i_l, i_{l+1}]}(g) = (i_{l+1} - i_l) \times N_g/N, \text{ where } l \in \{0, \ldots, m-1\}$$

where Ng is the total number of observations from distribution g.

From this, you can then derive the likelihood ratio score for a cell:

$$tc = \sum_{g=1}^{K} Oc(g) * \log\left(\frac{Oc(g)}{Ec(g)}\right)$$

Then for that partition L, you can obtain the likelihood ratio test statistic TL:

$$TL = \sum\nolimits_{all\ C} tc$$

Then given all TL for every partition L, you can obtain the test statistic for a given partition size by summation or maximization (**summation is used in paper**):

$$Sm = \sum\nolimits_{all\ L} L$$

To obtain the p-value for a given Sm for a given partition size m, if the N is large,

We will need <u>large scale Monte Carlo simulations</u> to obtain the null distribution.

Given sample sizes $N_1, N_2, \ldots, N_K$, randomly reassign ranks {1…..N} to K groups of sizes $N_1, \ldots, N_K$ and compute test statistic for each reassignment. P-value is the fraction of reassignments at least as large as the one observed, computed out of the B+1 assignments that include the B reassignments made at random and the original observed assignment (see Chapter 5 in Testing Statistical Hypotheses, 3rd Edition).

minP – the final univariate test statistic – is the minimum of the p-values from all Sm. Its null distribution "can be easily obtained from the null distributions of the test statistics for fixed ms". (Personally not sure about this)

Benefits:

- Partitioning and calculating can be done in O(N^2)
- Shows visibly better performance in paper experiments, especially in data that is clustered/scattered.
-

Python Implementation

We want to partition into a given number of m cells.

OR we can perform all possible partitions and then filter down into partitions of given size m.

Partitioning for all can be done through a recursive function (link: .

Uncertain about Monte Carlo simulation to obtain p-value, as well as the final p-value for .

Math/Partitioning Lingo:

A partition of a set *X* is a set of non-empty subsets of *X* such that every element *x* in *X* is in exactly one of these subsets

Cells = a set within a family of sets P – the partition

Having 2 numbers above one another in a curved bracket is always **a binomial coefficient**.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$