Transmission, an indicator of synergy reimplemented

Matthijs den Besten*

September 15, 2014

The code below implements Leydesdorff’s \( T \) (Leydesdorff et al. [2014]) derived from Shannon’s \( H \) in R (R Core Team [2012]). Leydesdorff’s own implementation in another language is available at http://www.leydesdorff.net/software/th4.

Transmission (\( T \)) or mutual information among components is defined as the sum of each component’s entropy (Shannon’s \( H \)) minus the joint entropy of each pair of components plus the joint entropy of each triplet of components, etcetera.

In order to compute the transmission value of a set of variables we first need to list all combinations of these variables.

```r
> list.combinations <- function(variable.names) {
+ require utils;
+ n <- length(variable.names);
+ return(lapply(1:n, function(m) combn(variable.names, m)));
+ }
```

The joint entropy for combinations of variables is based on the number of observation in each contingency. The function joint entropy extends the function \( H \) [Hausser and Strimmer [2013]].

```r
> # use default arguments of entropy function
> joint.entropy.vanilla <- function(...) {
+ require(entropy);
+ entropy(summary(factor(paste(list(...))), maxsum=Inf));
+ }
> # separately specify entropy arguments
> joint.entropy <- function(var.list, ...) {
+ require(entropy);
+ if(is.data.frame(var.list)) {
+ counts <- summary(as.factor(apply(apply(var.list, 2, as.character),
+ 1, paste, collapse="")), maxsum=Inf);
+ }
```
The joint entropy is computed for each combination of variables in the set.

```
> apply.combn <- function(input, ...) {
+   return(lapply(list.combinations(names(input)),
+                 function(el) {
+                   apply(el, 2,
+                         function(col) {
+                         joint.entropy(input[,col], ...);
+                       })
+                 }))
+ }
```

Transmission is defined as the sum all entropies derived from an odd number of variables minus the sum of all entropies derived from an even number of variables.

```
> transmission <- function(...) {
+   entropies <- apply.combn(...);
+   return(sum(sapply(entropies, sum)*ifelse(1:length(entropies)%%2, 1, -1)));
+ }
```

Leydesdorff measures entropy in bits and presumably uses maximum likelihood.

```
> T.leydesdorff <- function(...) {
+   transmission(unit="log2", method="ML", ...);
+ }
```

References

