

How to match on a Mahalanobis distance

Jake Bowers and Ben Hansen

October 30, 2006

In order to match on a Mahalanobis distance, or on a Mahalanobis distance within calipers, one has to first combine covariates into a matrix of Mahalanobis distances (or list of such matrices). R has some functions for creating Mahalanobis distances, but they seem to be oriented to applications other than Mahalanobis matching. This How To illustrates how those functions are adapted to this purpose.

First, here is an adaptation of the R function `mahalanobis`. It is specifically designed to be combined with `outer` or `makedist`, and may behave unexpectedly if used in isolation. Its arguments are `data`, a data frame containing all covariates to be combined in the distance; `inv.cov`, an inverted covariance for the k covariates, where $k \geq 2$; and character vectors `Tnms`, `Cnms` containing subsets of the row names of `data` that correspond to treatment and control groups, respectively.

```
> myMH <- function(Tnms, Cnms, inv.cov, data) {  
+   stopifnot(!is.null(dimnames(inv.cov)[[1]]), dim(inv.cov)[1] >  
+     1, all.equal(dimnames(inv.cov)[[1]], dimnames(inv.cov)[[2]]),  
+     all(dimnames(inv.cov)[[1]] %in% names(data)))  
+   covars <- dimnames(inv.cov)[[1]]  
+   xdiffs <- as.matrix(data[Tnms, covars])  
+   xdiffs <- xdiffs - as.matrix(data[Cnms, covars])  
+   rowSums((xdiffs %*% inv.cov) * xdiffs)  
+ }
```

Before using it, one has to select the covariates, invert their covariance matrix, and isolate names of treated and control subjects.

You're loading `optmatch`, by Ben Hansen, a package for flexible and optimal matching. Important license information:
 The `optmatch` package makes essential use of D. P. Bertsekas and P. Tseng's RELAX-IV algorithm and code, as well as Bertsekas' AUCTION algorithm and code.
 Bertsekas and Tseng freely permit their software to be used for research purposes, but non-research uses, including the use of it to 'satisfy in any part commercial delivery requirements to government or industry,' require a special agreement with them.
 By extension, this requirement applies to any use of the `fullmatch()` function. (If you are using another package that has loaded `optmatch`, then you will probably be using `fullmatch` indirectly.)
 For more information, enter `relaxinfo()` at the command line

```
> icv <- solve(cov(nuclear.nopt[, c("cap", "date")]))
> trtnms <- row.names(nuclear.nopt)[as.logical(nuclear.nopt$pr)]
> ctlnms <- row.names(nuclear.nopt)[!as.logical(nuclear.nopt$pr)]
> mdist <- outer(trtnms, ctlnms, FUN = myMH, inv.cov = icv, data = nuclear.nopt)
> dimnames(mdist) <- list(trtnms, ctlnms)
> round(mdist, 2)
```

	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
A	5.00	0.00	0.48	7.38	2.13	9.83	2.81	7.59	3.94	7.90	2.73	3.81	3.20	6.23	3.66
B	3.66	0.48	0.00	6.98	1.39	8.71	1.84	6.25	2.36	6.79	0.92	1.99	2.09	3.56	1.49
C	0.45	2.66	1.56	2.44	0.07	3.06	0.03	1.61	0.14	1.96	1.46	0.29	0.04	1.27	2.24
D	0.59	7.90	6.79	0.27	2.03	0.12	1.59	0.05	1.67	0.00	6.47	2.61	1.41	3.51	7.78
E	3.14	2.73	0.92	7.64	1.79	8.33	1.89	5.63	1.58	6.47	0.00	0.92	1.96	1.34	0.10
F	6.34	15.25	10.54	11.90	8.15	9.83	7.26	7.51	5.10	8.73	5.48	4.33	6.79	2.18	4.91
G	2.82	9.57	6.07	7.15	3.93	5.97	3.32	3.96	1.93	4.88	2.68	1.47	3.01	0.38	2.57
	W	X	Y	Z											
A	2.72	7.74	16.80	15.24											
B	1.15	4.83	12.79	10.31											
C	0.31	1.72	6.21	9.75											
D	3.48	3.17	4.31	15.79											
E	0.46	2.22	8.51	5.17											
F	5.26	1.72	2.19	2.12											
G	2.11	0.21	1.97	3.25											

```
> fullmatch(mdist)
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
m.1	m.2	m.3	m.4	m.5	m.6	m.7	m.3	m.1	m.2	m.4	m.3	m.4	m.3	m.4	m.3	m.4	m.5	m.3	m.3
U	V	W	X	Y	Z														
m.7	m.5	m.3	m.7	m.7	m.6														

A good way to use this construction is in combination with `makedist`, a function designed to handle a number of contingencies.

```
> mdd <- function(trtvar, dat, inverse.cov) {
+   ans <- outer(names(trtvar)[trtvar], names(trtvar)[!trtvar],
+     FUN = myMH, inv.cov = inverse.cov, data = dat)
+   dim(ans) <- c(sum(trtvar), sum(!trtvar))
+   dimnames(ans) <- list(names(trtvar)[trtvar], names(trtvar)[!trtvar])
+   ans
+ }
> altmdist <- makedist(pr ~ 0, nuclear.nopt, mdd, inverse.cov = icv)
> fullmatch(altmdist)
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
m.1	m.2	m.3	m.4	m.5	m.6	m.7	m.3	m.1	m.2	m.4	m.3	m.4	m.3	m.4	m.3	m.4	m.5	m.3	m.3
U	V	W	X	Y	Z														
m.7	m.5	m.3	m.7	m.7	m.6														

Building on this, you could modify the function `mdd` to include calipers, variations on the Mahalanobis distance, or whatever.