MatchingFrontier: Computing the Balance-Sample Size Frontier in Matching Methods for Causal Inference[*]

Gary King
Harvard University

Christopher Lucas
Harvard University

Richard Nielsen
MIT

Abstract

MatchingFrontier is an R package that implements the methods described in King, Lucas, and Nielsen (n.d.) for optimizing both balance and sample size in matching methods for causal inference. **MatchingFrontier** supports the computation of frontiers for both continuous and discrete metrics and also provides functions for visualizing the frontier and exporting matched data sets for further analysis.

Keywords: R, matching, frontier, Mahalanobis, L1.

1. Getting Started

MatchingFrontier is a software package written in the R language (Team *et al.* 2012) for the computation of the balance - sample size frontier. **MatchingFrontier** is currently hosted on Github and is not in the CRAN repositories. You can install the current development release of **MatchingFrontier** with the **devtools** package (Wickham and Chang 2013), as follows.

```
library(devtools)
install_github('ChristopherLucas/MatchingFrontier/package')
```

2. Using MatchingFrontier

MatchingFrontier supports the computation of the balance - sample size frontier, the estimation of effects along it, plotting functions, and support for exporting optimal data sets that are on the balance - sample size frontier. We will illustrate the use of **MatchingFrontier** with the LaLonde data (LaLonde 1986; Dehejia and Wahba 1999), which is included in the package.

The user must first create the frontier. To do so, use the `makeFrontier()` function, as follows.

```
# Load the package and the data
library(MatchingFrontier)
```

---

[*]The current release of **MatchingFrontier** is in active development and will continue to grow over the coming months. Comments and suggestions are greatly appreciated.

```
data('lalonde')

# Create a vector of column names to indicate which variables we
# want to match on. We will match on everything except the treatment
# and the outcome.
match.on <- colnames(lalonde)[!(colnames(lalonde) %in% c('re78', 'treat'))]
my.frontier <- makeFrontier(dataset = lalonde,
                            treatment = 'treat',
                            outcome = 're78',
                            match.on = match.on)
Calculating Mahalanobis distances...
Calculating theoretical frontier...
Calculating information for plotting the frontier...
> my.frontier
An imbalance frontier with 13494 points.
```

`match.on` is a vector holding the variable names that the user wishes to match on. In the LaLonde data, `re78` is the outcome and `treat` is the treatment, so we exclude those variable names from the vector.

By default, `makeFrontier()` calculates the frontier for the Average Mahalanobis Distance metric (AMD). The quantity of interest is the *feasible sample average treatment effect on the treated* or FSATT (King *et al.* n.d.). Weights are computed in the estimation stage and are bound to the data when the user exports a data set. The full function is as follows. For complete information, see the help file in the package with `?makeFrontier`.

```
makeFrontier(dataset,
             treatment,
             outcome,
             match.on,
             QOI = 'FSATT',
             metric = 'Mahal',
             ratio = 'variable',
             breaks = NULL)
```

Continuing with the Lalonde example, next we'll estimate the effects along the frontier with the `estimateEffects()` function, which takes the output from `makeFrontier()` to estimate the effect of the treatment along all values of the frontier. With the Lalonde example, the code is as follows.

```
my.estimates <- estimateEffects(my.frontier,
                                're78 ~ treat')
```

The first argument is the output from `makeFrontier()` and the second is the formula passed to the `lm()` function. `estimateEffects` stores the estimates and the 95% confidence interval for each point it estimates.

Next, we can plot the frontier and the estimates with the plotting functions, as follows.

```
# Plot frontier
plotFrontier(my.frontier)

# Plot estimates
plotEstimates(my.frontier, my.estimates)
```

Both plotting functions use the ellipses feature to pass arguments to the base `plot()` function, so any argument that can be used with `plot()` can also be used with `plotFrontier()` and `plotEstimates()`, such as 'main' for the title, 'xlab' and 'ylab' for the axis labels, etc.

Lastly, users may wish to export a data set on the frontier for additional analysis. To do so, use the `generateDataset()` function, as follows.

```
n = 1000 # Identify the point from which to select the data
matched.data <- generateDataset(my.frontier, N = n)
```

If the estimand is variable ratio, as it is by default, the exported data set will include the appropriate weights necessary for estimating the FSATT.

References

Dehejia RH, Wahba S (1999). "Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs." *Journal of the American statistical Association*, **94**(448), 1053–1062.

King G, Lucas C, Nielsen R (n.d.). "Optimizing Balance and Sample Size in Matching Methods for Causal Inference." *Working Paper*.

LaLonde RJ (1986). "Evaluating the econometric evaluations of training programs with experimental data." *The American Economic Review*, pp. 604–620.

Team RC, *et al.* (2012). "R: A language and environment for statistical computing."

Wickham H, Chang W (2013). "devtools: Tools to make developing R code easier." *R package version*, **1**(1).

Affiliation:

Gary King
Department of Government
Harvard University
1737 Cambridge St, Cambridge, MA, USA
E-mail: king@harvard.edu
URL: http://gking.harvard.edu/

Christopher Lucas
Department of Government
Harvard University
1737 Cambridge St, Cambridge, MA, USA
E-mail: clucas@fas.harvard.edu
URL: christopherlucas.org

Richard Nielsen
Department of Political Science
Massachusetts Institute of Technology
77 Massachusetts Avenue, Cambridge, MA, USA
E-mail: rnielsen@mit.edu
URL: http://www.mit.edu/ rnielsen/index.htm