Entropy Balancing Performance Considerations

Apoorva Lal

```
library(ebal); library(knitr)
```

```
## Loading required package: torch
```

```
## ##
## ## ebal Package: Implements Entropy Balancing.
```

```
## ## See http://www.stanford.edu/~jhain/ for additional information.
```

```
# modified dummy function from documentation
trial_fn = function(k) {
  n0 = 5 * k; n1 = 3 * k
  treatment = c(rep(0, n0), rep(1, n1))
  X = rbind(replicate(3, rnorm(n0, 0)), replicate(3, rnorm(n1, .5)))
  status = try({
    old_time = system.time({
      ebout1 = ebalance(Treatment = treatment, X = X, method = "GaussNewton")
    })[3]
    }, silent = TRUE)
  # gauss newton fails often
  if(inherits(status, "try-error")) {
      old_time = NA
      cat("GN did not converge", k, "\n");
      ebout1 = list(w = rep(1, n0))
    }
  new_time = system.time({
    ebout2 = ebalance(Treatment = treatment, X = X, method = "AutoDiff")
  })[3]
  list(
    problem_size = n0 + n1,
    # means in reweighted control group data
    max_imbalance_old = max(abs(
                colMeans(X[treatment == 1, ]) -
                apply(X[treatment == 0, ], 2, weighted.mean, w = ebout1$w)
              ))  %>% round(3),
            # estimates from new ebal
    max_imbalance_new = max(abs(
            colMeans(X[treatment == 1, ]) -
            apply(X[treatment == 0, ], 2, weighted.mean, w = ebout2$w)
          )) %>% round(3),
    old_time = round(unname(old_time),3),
    new_time = round(unname(new_time),3)
  )
}
```

Summarize for growing problem size.

```r
sapply(10^(1:7), trial_fn) %>% t %>% as.data.frame %>% kable()
```

problem_size	max_imbalance_old	max_imbalance_new	old_time	new_time
80	0	0	0.002	0.012
800	0.001	0	0.001	0.02
8000	0	0	0.003	0.046
80000	0	0	0.042	0.281
800000	0	0	0.415	0.162
8000000	0	0	3.754	1.048
80000000	0	0	29.09	7.841