Performance of Big Match for Varying Sample Sizes

Rachael Caelie (Rocky) Aikens

1/28/2019

This rmarkdown is meant for testing the performance of the current implementation of big_match on sample data.

Import Data

This sample cohort data was provided by Justin Lee at the Quantitative Sciences Unit. It contains ~900,000 observations of 112 variables.

```
dat <- read_sas("../sample_data/justincohort_june2017.sas7bdat")

# dimensions: ~900,000 x 112
dim(dat)
```

```
## [1] 893498    112
```

We include only hospitalizations with `totalct` > 1 and `arteryCt` < 3. A patient is considered to have recieved treatment if `arteryCt` is greater than 1. The outcome of this analysis is mortality.

```
# filter and add treatment column
dat <- filter(dat, totalct > 1 & arteryCt < 3) %>%
  mutate(treat = ifelse(arteryCt > 1, 1, 0))

# dimensions: ~900,000 x 112
dim(dat)
```

```
## [1] 833657    113
```

```
dat <- filter(dat, hosp_state != "Virgin Islands")
```

User Time

```
run_big_match <- function(n_samples){
  n_dat <- sample_n(dat, n_samples)
  t1 <- proc.time()
  a.strat <- auto_stratify(data = n_dat, treat = "treat",
                           outcome = "dead",
                           covariates = c("totalct", "AMI_7", "COPD_7",
                                          "ISCHEMICHEART_7","STROKE_TIA_7", "ATRIAL_FIB_7",
                                          "CHRONICKIDNEY_7", "DIABETES_7","ALZH_DEMEN_7",
                                          "Male"),
                           size = 2500)
  print("Stratification complete.")
  strat_time <- proc.time() - t1
  t2 <- proc.time()
  big_match(a.strat, propensity_formula = treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_
  strata_match_time <- proc.time() - t2
  t3 <- proc.time()
  big_match_slow(a.strat, propensity_formula = treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICH
  full_match_time = proc.time() - t3
  return(data.frame(rbind(strat_time, strata_match_time, full_match_time)))
}
```

```
options("optmatch_max_problem_size" = Inf)
n_samples <- c(5000, 10000, 15000, 20000, 25000, 30000, 35000)
time_list <- lapply(n_samples, run_big_match)
```

```
## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: dead~totalct+AMI_7+COPD_7+ISCHEMICHEART_7+STROKE_TIA_7+ATRIAL_FIB_7+CH
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
## [1] "Stratification complete."
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race + strata(stratum)
## <environment: 0x7fb70c5abd20>
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race
## <environment: 0x7fb7121e8190>
```

```
## Warning in value[[3L]](cond): Error gathering complete data. If the data
## has missing cases, imputation will not be performed. (Sometimes this can
## be fixed by supplying a `data` argument when fitting the model that's to
## be passed to `scores`. Alternatively, just take care of (impute) NAs before
## you fit that model.)
```

```
## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: dead~totalct+AMI_7+COPD_7+ISCHEMICHEART_7+STROKE_TIA_7+ATRIAL_FIB_7+CH
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
## [1] "Stratification complete."
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race + strata(stratum)
```

```
## <environment: 0x7fb712035000>
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race
## <environment: 0x7fb6e8468d70>

## Warning in value[[3L]](cond): Error gathering complete data. If the data
## has missing cases, imputation will not be performed. (Sometimes this can
## be fixed by supplying a `data` argument when fitting the model that's to
## be passed to `scores`. Alternatively, just take care of (impute) NAs before
## you fit that model.)

## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: dead~totalct+AMI_7+COPD_7+ISCHEMICHEART_7+STROKE_TIA_7+ATRIAL_FIB_7+CH
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
## [1] "Stratification complete."
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race + strata(stratum)
## <environment: 0x7fb6eb635230>
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race
## <environment: 0x7fb7085ba070>

## Warning in value[[3L]](cond): Error gathering complete data. If the data
## has missing cases, imputation will not be performed. (Sometimes this can
## be fixed by supplying a `data` argument when fitting the model that's to
## be passed to `scores`. Alternatively, just take care of (impute) NAs before
## you fit that model.)

## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: dead~totalct+AMI_7+COPD_7+ISCHEMICHEART_7+STROKE_TIA_7+ATRIAL_FIB_7+CH
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
## [1] "Stratification complete."
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race + strata(stratum)
## <environment: 0x7fb6f6cc91f8>
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race
## <environment: 0x7fb6fc18c4a0>

## Warning in value[[3L]](cond): Error gathering complete data. If the data
## has missing cases, imputation will not be performed. (Sometimes this can
## be fixed by supplying a `data` argument when fitting the model that's to
## be passed to `scores`. Alternatively, just take care of (impute) NAs before
## you fit that model.)

## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: dead~totalct+AMI_7+COPD_7+ISCHEMICHEART_7+STROKE_TIA_7+ATRIAL_FIB_7+CH
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
## [1] "Stratification complete."
```

```
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race + strata(stratum)
## <environment: 0x7fb70a2c0918>
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race
## <environment: 0x7fb6f79e77e0>
```

```
## Warning in value[[3L]](cond): Error gathering complete data. If the data
## has missing cases, imputation will not be performed. (Sometimes this can
## be fixed by supplying a `data` argument when fitting the model that's to
## be passed to `scores`. Alternatively, just take care of (impute) NAs before
## you fit that model.)
```

```
## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: dead~totalct+AMI_7+COPD_7+ISCHEMICHEART_7+STROKE_TIA_7+ATRIAL_FIB_7+CH
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
## [1] "Stratification complete."
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race + strata(stratum)
## <environment: 0x7fb70875b9e0>
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race
## <environment: 0x7fb6f693b2d8>
```

```
## Warning in value[[3L]](cond): Error gathering complete data. If the data
## has missing cases, imputation will not be performed. (Sometimes this can
## be fixed by supplying a `data` argument when fitting the model that's to
## be passed to `scores`. Alternatively, just take care of (impute) NAs before
## you fit that model.)
```

```
## [1] "Constructing a model set via subsampling."
## [1] "Fitting prognostic model: dead~totalct+AMI_7+COPD_7+ISCHEMICHEART_7+STROKE_TIA_7+ATRIAL_FIB_7+CH
## [1] "Generating strata assignments based on prognostic score."
## [1] "Completing strata diagnostics."
## [1] "Stratification complete."
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race + strata(stratum)
## <environment: 0x7fb6bdcd03f8>
## treat ~ totalct + hosp_state + AMI_7 + COPD_7 + ISCHEMICHEART_7 +
##     STROKE_TIA_7 + ATRIAL_FIB_7 + CHRONICKIDNEY_7 + DIABETES_7 +
##     ALZH_DEMEN_7 + Male + race
## <environment: 0x7fb70bd99818>
```

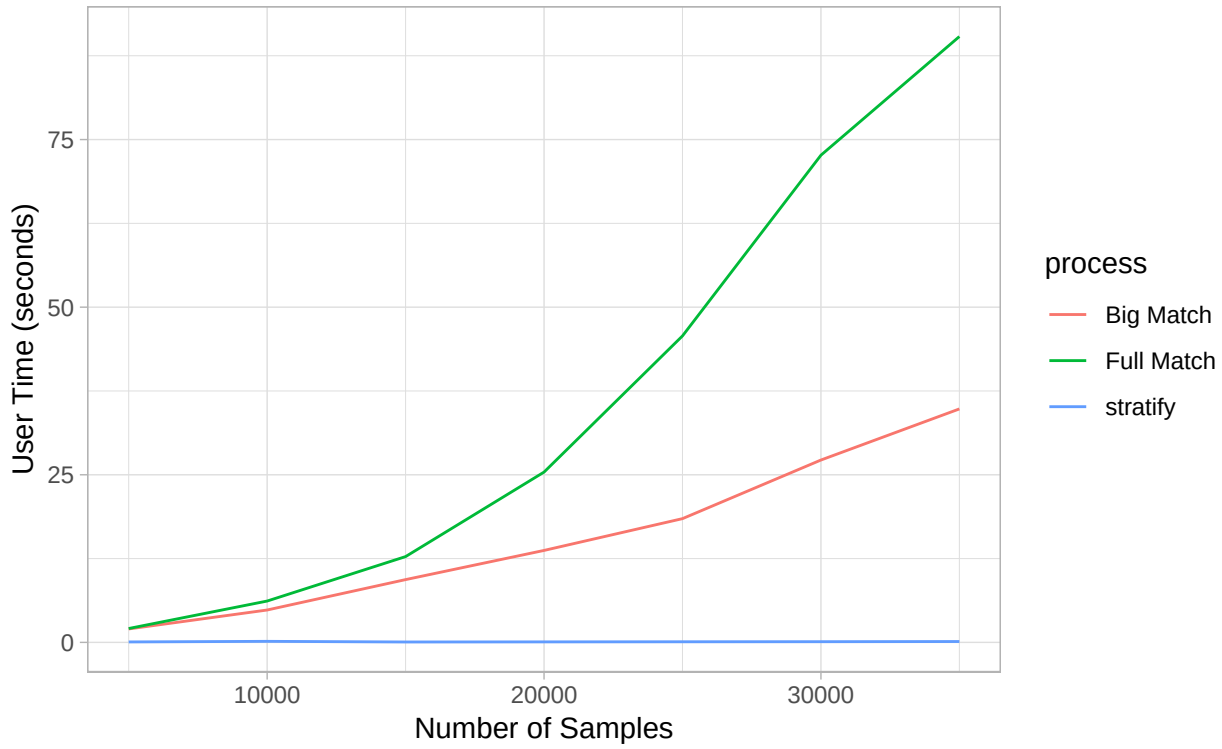
```
## Warning in value[[3L]](cond): Error gathering complete data. If the data
## has missing cases, imputation will not be performed. (Sometimes this can
## be fixed by supplying a `data` argument when fitting the model that's to
## be passed to `scores`. Alternatively, just take care of (impute) NAs before
## you fit that model.)
```

```
bind_rows(time_list, .id = "column_label")
```

```
##    column_label user.self sys.self elapsed user.child sys.child
## 1             1     0.067    0.008   0.081          0         0
## 2             1     2.004    0.451   2.456          0         0
## 3             1     2.066    0.291   2.358          0         0
## 4             2     0.154    0.229   0.383          0         0
## 5             2     4.827    1.012   5.843          0         0
## 6             2     6.163    1.317   7.482          0         0
## 7             3     0.059    0.010   0.069          0         0
## 8             3     9.362    2.213  11.581          0         0
## 9             3    12.796    1.905  14.708          0         0
## 10            4     0.074    0.016   0.090          0         0
## 11            4    13.712    3.144  16.863          0         0
## 12            4    25.402    5.229  30.853          0         0
## 13            5     0.091    0.020   0.110          0         0
## 14            5    18.455    4.890  23.373          0         0
## 15            5    45.712    9.713  55.853          0         0
## 16            6     0.102    0.021   0.123          0         0
## 17            6    27.200    6.887  34.114          0         0
## 18            6    72.667   16.076  90.089          0         0
## 19            7     0.124    0.036   0.160          0         0
## 20            7    34.819    8.647  43.517          0         0
## 21            7    90.365   22.601 115.792          0         0
```

```
trials <- length(n_samples)
time_df <- bind_rows(time_list, .id = "column_label") %>%
  mutate(n_samples = rep(n_samples, each = 3)) %>%
  mutate(process = rep(c("stratify", "Big Match", "Full Match"), trials)) %>%
  select(c(process, n_samples, user.self, sys.self, elapsed))
```

```
a <- ggplot(time_df, aes(x = n_samples, y = user.self, group = process, color = process)) +
  geom_line() +
  labs(x = "Number of Samples", y = "User Time (seconds)")
a
```


User Time (seconds)
75
50
25
0
10000
20000
30000
Number of Samples
process
Big Match
Full Match
stratify