Scalar on Function-Regression: PASAT scores of MS patients with `FDboost`

Model Fitting:

```
mb_formula <-
  pasat ~ bols(sex) +
    bbs(visit.time, df=2) +
    bbs(visit.time, df=2) %Xc% bols(sex) +
    brandom(subject, by = visit.time, df=2) +
    brandom(subject, df=2) +
    bsignal(rcst, s=segm.s, df=2) + bsignal(cca, s=segm.t, df=2) +
    bsignal(rcst, s=segm.s, df=2) %X% bolsc(sex) +
    bsignal(cca, s=segm.t, df=2)  %X% bolsc(sex) +
    bsignal(rcstds, s=segm.ds, df=2) + bsignal(ccadt, s=segm.dt, df=2) +
    bsignal(rcstds, s=segm.ds, df=2) %X% bolsc(sex) +
    bsignal(ccadt, s=segm.dt, df=2) %X% bolsc(sex)
   # everything with 2 df for unbiased selection; interactions centered around main effects
mb_formula_rel <-
  pasat_rel_be ~ bols(sex) +
    bbs(visit.time, df=2) +
    bbs(visit.time, df=2) %Xc% bols(sex) +
    brandom(subject, by = visit.time, df=2) +
    brandom(subject, df=2) +
    bsignal(rcst, s=segm.s, df=2) + bsignal(cca, s=segm.t, df=2) +
    bsignal(rcst, s=segm.s, df=2) %X% bolsc(sex) +
    bsignal(cca, s=segm.t, df=2)  %X% bolsc(sex) +
    bsignal(rcstds, s=segm.ds, df=2) + bsignal(ccadt, s=segm.dt, df=2) +
    bsignal(rcstds, s=segm.ds, df=2) %X% bolsc(sex) +
    bsignal(ccadt, s=segm.dt, df=2) %X% bolsc(sex)
```

Median Regression:

Careful, the code chunks will start 25 processes on your machine...

```
set.seed(20160621)
system.time(mb_q50 <- FDboost(mb_formula,
  timeformula = NULL, family=QuantReg(tau = 0.5, qoffset = 0.5),
  data = c(as.list(dti), segm.t = 1:length.t, segm.s = 1:length.s,
    segm.dt = 1:(length.t-1), segm.ds = 1:(length.s-1)),
  control = boost_control(mstop = 1000, nu = 0.5)))
```

```
## Note: method with signature 'ddenseMatrix#dsparseMatrix' chosen for function 'crossprod',
##  target signature 'Cholesky#dgTMatrix'.
##  "Matrix#TsparseMatrix" would also be valid
```

```
##    user  system elapsed
##   3.028   0.024   3.055
```

```r
cvr_q50 <- cvrisk(mb_q50, folds = cv(model.weights(mb_q50), B = 100),
  mc.cores = 25)
```

```
## For bolsc, the transformation matrix Z is fixed over all folds.
```

```r
ss_q50 <- stabsel(mb_q50[mstop(cvr_q50)], PFER=1, cutoff=.8, mc.cores = 25)
```

```
## For bolsc, the transformation matrix Z is fixed over all folds.
```

```r
saveRDS(list(mb_q50, cvr_q50, ss_q50), "models/fdboost_q50.rds")
print(ss_q50)
```

```
##  Stability Selection with unimodality assumption
##
## Selected variables:
## bsignal(x = cca, s = segm.t, df = 2)
##                                    7
##
## Selection probabilities:
##                 brandom(subject, by = visit.time, df = 2)
##                                                      0.00
##                                                 bols(sex)
##                                                      0.01
##                  "bbs(visit.time, df = 2) %Xc% bols(sex)"
##                                                      0.03
##  "bsignal(x = ccadt, s = segm.dt, df = 2) %X% bolsc(sex)"
##                                                      0.06
##                   bsignal(x = ccadt, s = segm.dt, df = 2)
##                                                      0.11
##     "bsignal(x = cca, s = segm.t, df = 2) %X% bolsc(sex)"
##                                                      0.12
##    "bsignal(x = rcst, s = segm.s, df = 2) %X% bolsc(sex)"
##                                                      0.13
##                                  brandom(subject, df = 2)
##                                                      0.15
## "bsignal(x = rcstds, s = segm.ds, df = 2) %X% bolsc(sex)"
##                                                      0.21
##                                   bbs(visit.time, df = 2)
##                                                      0.22
##                     bsignal(x = rcst, s = segm.s, df = 2)
##                                                      0.35
##                  bsignal(x = rcstds, s = segm.ds, df = 2)
##                                                      0.61
##                      bsignal(x = cca, s = segm.t, df = 2)
##                                                      1.00
##
## ---
## Cutoff: 0.8; q: 3; PFER (*):  0.57
##    (*) or expected number of low selection probability variables
## PFER corresponds to signif. level 0.0438 (without multiplicity adjustment)
```

Binomial Model:

Needs some extra work for binomial as opposed to binary data:

```
Bin <- as.families(fname="BI")
#-------------------------------------------------------------------------------
# ugly, hacky fix: set bd-defaults to 60 wherever it's needed
# see https://github.com/boost-R/gamboostLSS/issues/12;
#     https://github.com/boost-R/mboost/issues/34

Bin60 <- Bin
add_bd_60 <- function(f){
  frmls <- formals(f)
  frmls$bd <- 60
  formals(f) <- frmls
  f
}
with(environment(Bin60@ngradient), {
  #all slots share same env, so enough to do this for one slot
  FAM$dldm <- add_bd_60(FAM$dldm)
  FAM$d2ldm2  <- add_bd_60(FAM$d2ldm2)
  FAM$d2ldm2  <- add_bd_60(FAM$d2ldm2)
  FAM$G.dev.incr <- add_bd_60(FAM$G.dev.incr)
  FAM$rqres <- quote(rqres(pfun = "pBI", type = "Discrete", ymin = 0, y = y,
    mu = mu, bd = 60))
  FAM$mu.initial <- quote(mu <- (y + 0.5)/(60 + 1))
  pdf <- add_bd_60(pdf)
})

#-------------------------------------------------------------------------------
set.seed(20160621)
system.time(mb_bi <- FDboost(mb_formula,
  timeformula = NULL, family=Bin60,
  data = c(as.list(dti), segm.t = 1:length.t, segm.s = 1:length.s,
    segm.dt = 1:(length.t-1), segm.ds = 1:(length.s-1)),
  control = boost_control(mstop = 5000, nu = 0.05)))
```

```
##    user  system elapsed
##  13.524   0.135  13.667
```

```
cvr_bi <- cvrisk(mb_bi, folds = cv(model.weights(mb_bi), B = 100),
  mc.cores = 25)
```

```
## For bolsc, the transformation matrix Z is fixed over all folds.
```

```
ss_bi <- stabsel(mb_bi[mstop(cvr_bi)], PFER=1, cutoff=.8, mc.cores = 25)
```

```
## For bolsc, the transformation matrix Z is fixed over all folds.
```

```
saveRDS(list(mb_bi, cvr_bi, ss_bi), "models/fdboost_bi.rds")
print(ss_bi)
```

```
##  Stability Selection with unimodality assumption
##
## Selected variables:
## bsignal(x = cca, s = segm.t, df = 2)
##                                   7
##
## Selection probabilities:
##                   "bbs(visit.time, df = 2) %Xc% bols(sex)"
##                                                       0.00
##                 brandom(subject, by = visit.time, df = 2)
##                                                       0.00
##                   bsignal(x = ccadt, s = segm.dt, df = 2)
##                                                       0.00
##                                                 bols(sex)
##                                                       0.01
## "bsignal(x = rcstds, s = segm.ds, df = 2) %X% bolsc(sex)"
##                                                       0.10
##  "bsignal(x = ccadt, s = segm.dt, df = 2) %X% bolsc(sex)"
##                                                       0.13
##                     bsignal(x = rcst, s = segm.s, df = 2)
##                                                       0.19
##     "bsignal(x = cca, s = segm.t, df = 2) %X% bolsc(sex)"
##                                                       0.25
##                                  brandom(subject, df = 2)
##                                                       0.27
##    "bsignal(x = rcst, s = segm.s, df = 2) %X% bolsc(sex)"
##                                                       0.27
##                   bsignal(x = rcstds, s = segm.ds, df = 2)
##                                                       0.35
##                                   bbs(visit.time, df = 2)
##                                                       0.43
##                      bsignal(x = cca, s = segm.t, df = 2)
##                                                       1.00
##
## ---
## Cutoff: 0.8; q: 3; PFER (*):  0.57
##    (*) or expected number of low selection probability variables
## PFER corresponds to signif. level 0.0438 (without multiplicity adjustment)
```

Beta Model:

```r
Beta <- as.families(fname="BE")

set.seed(20160621)
# No covariate effects on scale parameter:
system.time(mb_be <- FDboostLSS(list(
  mu = mb_formula_rel,
  sigma = pasat_rel_be ~ bols(ones, intercept = FALSE)),
  timeformula = NULL, families=Beta, method = "inner",
  data = c(as.list(dti), segm.t = 1:length.t, segm.s = 1:length.s,
    segm.dt = 1:(length.t-1), segm.ds = 1:(length.s-1)),
  control = boost_control(mstop = 500, nu = 0.2)))
```

```
##    user  system elapsed
##   8.920   0.010   8.934
```

```
cvr_be <- cvrisk(mb_be, folds = cv(model.weights(mb_be), B = 100), mc.cores = 25)
```

```
## Starting cross-validation...
## [fold]   [current mstop]
```

```
attr(mb_be, "data") <- dti
ss_be <- stabsel(mb_be[mstop(cvr_be)], PFER=1, cutoff=.8, mc.cores = 25)
```

```
## Run stabsel
```

```
saveRDS(list(mb_be, cvr_be, ss_be), "models/fdboost_be.rds")
print(ss_be)
```

```
##  Stability Selection with unimodality assumption
##
## Selected variables:
##     cca.mu ones.sigma
##          7         14
##
## Selection probabilities:
##     visit.time, sex.mu                 sex.mu subject, visit.time.mu
##                   0.00                   0.01                   0.06
##               ccadt.mu         rcstds, sex.mu             subject.mu
##                   0.06                   0.07                   0.14
##           rcst, sex.mu            cca, sex.mu          ccadt, sex.mu
##                   0.14                   0.15                   0.20
##                rcst.mu              rcstds.mu          visit.time.mu
##                   0.23                   0.40                   0.54
##                 cca.mu             ones.sigma
##                   1.00                   1.00
##
## ---
## Cutoff: 0.8; q: 4; PFER (*):  0.941
##    (*) or expected number of low selection probability variables
## PFER (specified upper bound):  1
## PFER corresponds to signif. level 0.0672 (without multiplicity adjustment)
```

```
#------------------------------------------------------------------------------

set.seed(20160621)
# With covariate effects on scale parameter:
system.time(mb_be2 <- FDboostLSS(mb_formula_rel,
  timeformula = NULL, families=Beta, method = "inner",
  data = c(as.list(dti), segm.t = 1:length.t, segm.s = 1:length.s,
    segm.dt = 1:(length.t-1), segm.ds = 1:(length.s-1)),
  control = boost_control(mstop = 500, nu = 0.2)))
```

```
##    user  system elapsed
##  11.303   0.209  11.524
```

```r
cvr_be2 <- cvrisk(mb_be2, folds = cv(model.weights(mb_be2), B = 100),
  mc.cores = 25)
```

```
## Starting cross-validation...
## [fold]   [current mstop]
```

```r
attr(mb_be2, "data") <- dti
ss_be2 <- stabsel(mb_be2[mstop(cvr_be2)], PFER=1, cutoff=.8, mc.cores = 25)
```

```
## Run stabsel
```

```r
saveRDS(list(mb_be2, cvr_be2, ss_be2), "models/fdboost_be2.rds")
print(ss_be2)
```

```
##  Stability Selection with unimodality assumption
##
## Selected variables:
##      cca.mu ccadt.sigma
##           7          24
##
## Selection probabilities:
##                    sex.mu    subject, visit.time.mu
##                      0.00                      0.00
##     visit.time, sex.sigma        visit.time, sex.mu
##                      0.00                      0.01
##                  ccadt.mu             rcst, sex.mu
##                      0.01                      0.03
##            rcstds, sex.mu             rcstds.sigma
##                      0.03                      0.03
##                subject.mu subject, visit.time.sigma
##                      0.04                      0.06
##                 cca.sigma          rcst, sex.sigma
##                      0.06                      0.06
##         rcstds, sex.sigma         ccadt, sex.sigma
##                      0.13                      0.13
##             ccadt, sex.mu              cca, sex.mu
##                      0.14                      0.15
##           cca, sex.sigma                 rcst.mu
##                      0.15                      0.20
##          visit.time.sigma               rcst.sigma
##                      0.22                      0.27
##                rcstds.mu            subject.sigma
##                      0.31                      0.31
##             visit.time.mu                sex.sigma
##                      0.42                      0.44
##               ccadt.sigma                   cca.mu
##                      0.80                      1.00
##
## ---
## Cutoff: 0.8; q: 5; PFER (*):  0.792
##    (*) or expected number of low selection probability variables
## PFER (specified upper bound):  1
## PFER corresponds to signif. level 0.0305 (without multiplicity adjustment)
```

Beta-Binomial Model:

Needs some extra work for binomial as opposed to binary data:

```
Betabin <- as.families(fname="BB")
#------------------------------------------------------------------------------
# ugly, hacky fix: set bd-defaults to 60 wherever it's needed
Betabin60 <- Betabin
add_bd_60 <- function(f){
  frmls <- formals(f)
  frmls$bd <- 60
  formals(f) <- frmls
  f
}
with(environment(Betabin60[[1]]@ngradient), {
  #all slots share same env, so enough to do this for one slot
  FAM$dldm <- add_bd_60(FAM$dldm)
  FAM$d2ldm2  <- add_bd_60(FAM$d2ldm2)
  FAM$d2ldm2  <- add_bd_60(FAM$d2ldm2)
  FAM$dldd  <- add_bd_60(FAM$dldd)
  FAM$d2ldd2  <- add_bd_60(FAM$d2ldd2)
  FAM$G.dev.incr <- add_bd_60(FAM$G.dev.incr)
  FAM$mu.initial <- quote(mu <- (y + 0.5)/(60 + 1))
  pdf <- add_bd_60(pdf)
})
with(environment(Betabin60[[2]]@ngradient), {
  FAM$dldm <- add_bd_60(FAM$dldm)
  FAM$d2ldm2  <- add_bd_60(FAM$d2ldm2)
  FAM$d2ldm2  <- add_bd_60(FAM$d2ldm2)
  FAM$dldd  <- add_bd_60(FAM$dldd)
  FAM$d2ldd2  <- add_bd_60(FAM$d2ldd2)
  FAM$G.dev.incr <- add_bd_60(FAM$G.dev.incr)
  FAM$mu.initial <- quote(mu <- (y + 0.5)/(60 + 1))
  pdf <- add_bd_60(pdf)
})

#------------------------------------------------------------------------------

set.seed(20160621)
# No covariate effects on scale parameter:
system.time(mb_bb <- FDboostLSS(list(mu = mb_formula,
  sigma = pasat ~ bols(ones, intercept = FALSE)),
  timeformula = NULL, families=Betabin60, method = "inner",
  data = c(as.list(dti), segm.t = 1:length.t, segm.s = 1:length.s,
    segm.dt = 1:(length.t-1), segm.ds = 1:(length.s-1)),
  control = boost_control(mstop = 500, nu = 0.1)))
```

```
##    user  system elapsed
##   9.342   0.143   9.492
```

```
cvr_bb <- cvrisk(mb_bb, folds = cv(model.weights(mb_bb), B = 100),
  mc.cores = 25)
```

```
## Starting cross-validation...
```

```
## [fold]   [current mstop]
```

```
attr(mb_bb, "data") <- dti
ss_bb <- stabsel(mb_bb[mstop(cvr_bb)], PFER=1, cutoff=.8, mc.cores = 25)
```

```
## Run stabsel
```

```
saveRDS(list(mb_bb, cvr_bb, ss_bb), "models/fdboost_bb.rds")
print(ss_bb)
```

```
##  Stability Selection with unimodality assumption
##
## Selected variables:
## visit.time.mu        cca.mu    ones.sigma
##             2             7            14
##
## Selection probabilities:
##     visit.time, sex.mu subject, visit.time.mu                 sex.mu
##                   0.00                   0.01                   0.02
##             subject.mu              ccadt.mu           rcst, sex.mu
##                   0.03                   0.03                   0.06
##         rcstds, sex.mu          ccadt, sex.mu            cca, sex.mu
##                   0.06                   0.09                   0.11
##                rcst.mu             rcstds.mu         visit.time.mu
##                   0.21                   0.40                   0.98
##                 cca.mu            ones.sigma
##                   1.00                   1.00
##
## ---
## Cutoff: 0.8; q: 4; PFER (*):  0.941
##    (*) or expected number of low selection probability variables
## PFER (specified upper bound):  1
## PFER corresponds to signif. level 0.0672 (without multiplicity adjustment)
```

```
#-------------------------------------------------------------------------------

set.seed(20160621)
# With covariate effects on scale parameter:
system.time(mb_bb2 <- FDboostLSS(mb_formula,
  timeformula = NULL, families=Betabin60, method = "inner",
  data = c(as.list(dti), segm.t = 1:length.t, segm.s = 1:length.s,
    segm.dt = 1:(length.t-1), segm.ds = 1:(length.s-1)),
  control = boost_control(mstop = 500, nu = 0.1)))
```

```
##    user  system elapsed
##  12.366   0.435  12.811
```

```
cvr_bb2 <- cvrisk(mb_bb2, folds = cv(model.weights(mb_bb2), B = 100),
  mc.cores = 25)
```

```
## Starting cross-validation...
## [fold]   [current mstop]
```

```r
attr(mb_bb2, "data") <- dti
ss_bb2 <- stabsel(mb_bb2[mstop(cvr_bb2)], PFER=1, cutoff=.8, mc.cores = 25)
```

```
## Run stabsel
```

```r
saveRDS(list(mb_bb2, cvr_bb2, ss_bb2), "models/fdboost_bb2.rds")
print(ss_bb2)
```

```
##  Stability Selection with unimodality assumption
##
## Selected variables:
##    visit.time.mu           cca.mu        sex.sigma visit.time.sigma
##                2                7               14               15
##
## Selection probabilities:
##        visit.time, sex.mu    subject, visit.time.mu
##                      0.00                      0.00
##     visit.time, sex.sigma subject, visit.time.sigma
##                      0.00                      0.00
##             subject.sigma              rcstds.sigma
##                      0.00                      0.00
##          ccadt, sex.sigma                 cca.sigma
##                      0.00                      0.01
##           rcst, sex.sigma            cca, sex.sigma
##                      0.01                      0.01
##         rcstds, sex.sigma                    sex.mu
##                      0.01                      0.02
##                  ccadt.mu                rcst.sigma
##                      0.02                      0.02
##              rcst, sex.mu            rcstds, sex.mu
##                      0.03                      0.06
##                subject.mu               cca, sex.mu
##                      0.07                      0.07
##             ccadt, sex.mu                   rcst.mu
##                      0.08                      0.10
##               ccadt.sigma                 rcstds.mu
##                      0.30                      0.31
##                 sex.sigma          visit.time.sigma
##                      0.94                      0.96
##             visit.time.mu                    cca.mu
##                      0.98                      1.00
##
## ---
## Cutoff: 0.8; q: 5; PFER (*):  0.792
##    (*) or expected number of low selection probability variables
## PFER (specified upper bound):  1
## PFER corresponds to signif. level 0.0305 (without multiplicity adjustment)
```

Model Comparison:

```
# Selected baselearners:
(mb_selected <- list(q50 =  names(selected(ss_q50)),
  bi =  names(selected(ss_bi)), be = names(selected(ss_be)$mu),
  be2 = c(names(selected(ss_be2)$mu), names(selected(ss_be2)$sigma)),
  bb = names(selected(ss_be)$mu),
  bb2 = c(names(selected(ss_bb2)$mu), names(selected(ss_bb2)$sigma))))
```

```
## $q50
## [1] "bsignal(x = cca, s = segm.t, df = 2)"
##
## $bi
## [1] "bsignal(x = cca, s = segm.t, df = 2)"
##
## $be
## [1] "cca.mu"
##
## $be2
## [1] "cca.mu"      "ccadt.sigma"
##
## $bb
## [1] "cca.mu"
##
## $bb2
## [1] "visit.time.mu"    "cca.mu"           "sex.sigma"
## [4] "visit.time.sigma"
```

```
# Cross-validated Loss:
min_cvr <- function(cvr) min(rowMeans(cvr))
(cvrisk_min <- sapply(list(q50=cvr_q50, bi=cvr_bi, be=cvr_be, bb=cvr_bb,
  be2=cvr_be2, bb2=cvr_bb2), min_cvr))
```

```
##       q50        bi        be        bb       be2       bb2
##  3.700228  3.917353 -1.043764  3.325176 -1.083713  3.316811
```

Mostly small differences, ranking of models not stable for different CV folds.

Computational Details:

```
sessionInfo()
```

```
## R version 3.3.0 (2016-05-03)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.4 LTS
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
```

```
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] gamlss.dist_4.3-6 MASS_7.3-44       gamboostLSS_1.3-0 FDboost_0.2-0
## [5] mboost_2.6-0      stabs_0.5-1       zoo_1.7-13
##
## loaded via a namespace (and not attached):
##  [1] modeltools_0.2-21  coin_1.1-2         splines_3.3.0
##  [4] lattice_0.20-33    colorspace_1.2-6   gamm4_0.2-3
##  [7] htmltools_0.3.5    stats4_3.3.0       yaml_2.1.13
## [10] mgcv_1.8-12        MCMCpack_1.3-6     survival_2.39-4
## [13] nloptr_1.0.4       multcomp_1.4-5     plyr_1.8.3
## [16] stringr_1.0.0      MatrixModels_0.4-1 munsell_0.4.3
## [19] gtable_0.2.0       mvtnorm_1.0-5      codetools_0.2-14
## [22] coda_0.18-1        evaluate_0.9       magic_1.5-6
## [25] knitr_1.13         strucchange_1.5-1  SparseM_1.7
## [28] RLRsim_3.1-2       quantreg_5.24      TH.data_1.0-7
## [31] Rcpp_0.12.5        pbs_1.1            party_1.0-25
## [34] scales_0.4.0       formatR_1.4        lme4_1.1-12
## [37] grpreg_3.0-0       mcmc_0.9-4         ggplot2_2.1.0
## [40] digest_0.6.9       stringi_1.1.1      grid_3.3.0
## [43] quadprog_1.5-5     tools_3.3.0        sandwich_2.3-4
## [46] magrittr_1.5       fda_2.4.4          Matrix_1.2-6
## [49] nnls_1.4           minqa_1.2.4        rmarkdown_0.9.6
## [52] boot_1.3-17        refund_0.1-14      nlme_3.1-128
```