Cervical data

A. Inglis and C. Hurley

2021-07-05

Packages

```
# devtools::install_github("AlanInglis/vivid")
# devtools::install_github("cbhurley/condvis2")
library(mlr)
library(vivid)
library(dplyr)
RNGkind(kind="Mersenne-Twister", normal.kind="Inversion", sample.kind="Rejection" ) # defaults
```

Read in and setup data.

```
cervical <- read.csv("https://raw.githubusercontent.com/AlanInglis/vivid/master/paperCodeData/cervicalC
                     sep=",",  na.strings = c('?'), stringsAsFactors = FALSE)
# Remove cancer tests, similar and constant variables:
# take logs of skewed variables
# Turn dummy variables into factors:
cervical <- dplyr::select(cervical, -Citology, -Schiller, -Hinselmann,
                          -Dx.CIN, -Dx, -Horm_Cont,
                          -Smokes, -IUD, -STDs, -STDs_No_diag, -STDs_AIDS,
                          -STDs.Hep_B, -STDs_cerv_condy, -STDs_Time_first_diag,
                          -STDs_Time_last_diag, -STDs_pel_inf,
                          -STDs_gen_h, -STDs_m_c,
                          -STDs.HPV, -STDs_vag_condy, -STDs_vp_condy) %>%
   mutate(across(Age:IUD_yrs, ~ log(.x+ 1))) %>%
   mutate(Biopsy = factor(Biopsy, levels=0:1,labels=c('Healthy', 'Cancer') )) %>%
   mutate(across(.cols=c("STDs_condy", "STDs_syph", "STDs_HIV",
                         "Dx.HPV", "Dx.Cancer"),   factor))
# set up training and testing, making sure Biopsy is proportionally sampled
set.seed(1701)
split <- rsample::initial_split(cervical, prop=.7, strata = Biopsy)
cTrain <- rsample::training(split)
cTest <- rsample::testing(split)
```

Model fitting

```
cTask <- makeClassifTask(data = cTrain, target = "Biopsy")
cLrn <- makeLearner("classif.gbm",
                    predict.type = "prob",
                    par.vals = list(
                       interaction.depth = 2,
                       n.trees = 100,
                       shrinkage = 0.15,
                       n.minobsinnode=5
                    ))
set.seed(1701)
cfit <- train(cLrn, cTask)
```

```
## Distribution not specified, assuming bernoulli ...
```

```
# # Test predictions
pred1 <- predict(cfit, newdata = cTest)
# # Evaluate performance accuracy, area under curve and mean misclassification error
performance(pred1, measures = list(acc, auc))
```

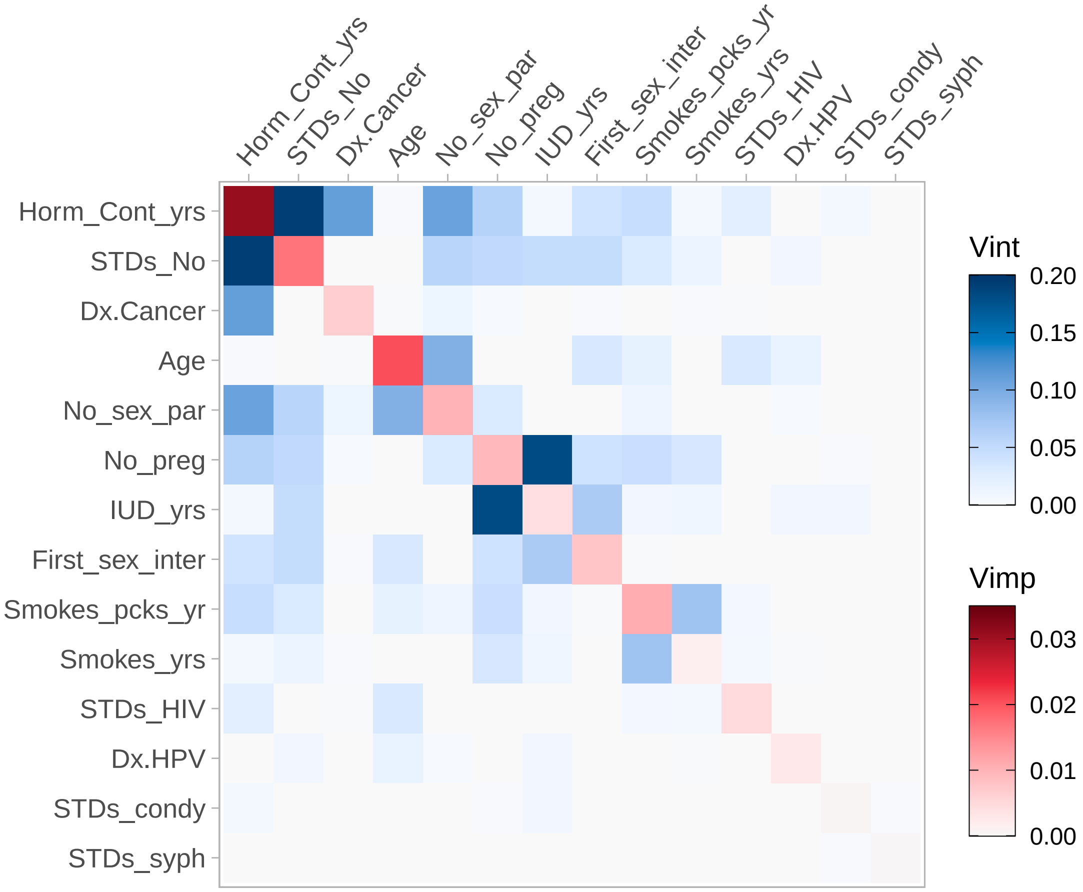
```
##       acc       auc
## 0.9066148 0.7717300
```

Create vivid matrix

```
set.seed(1701)
viv <- vivi(cTrain, cfit, response = "Biopsy", class = "Cancer",
                gridSize = 30, importanceType = "agnostic")
```

Heatmap

```
viviHeatmap(viv, angle = 50)
```

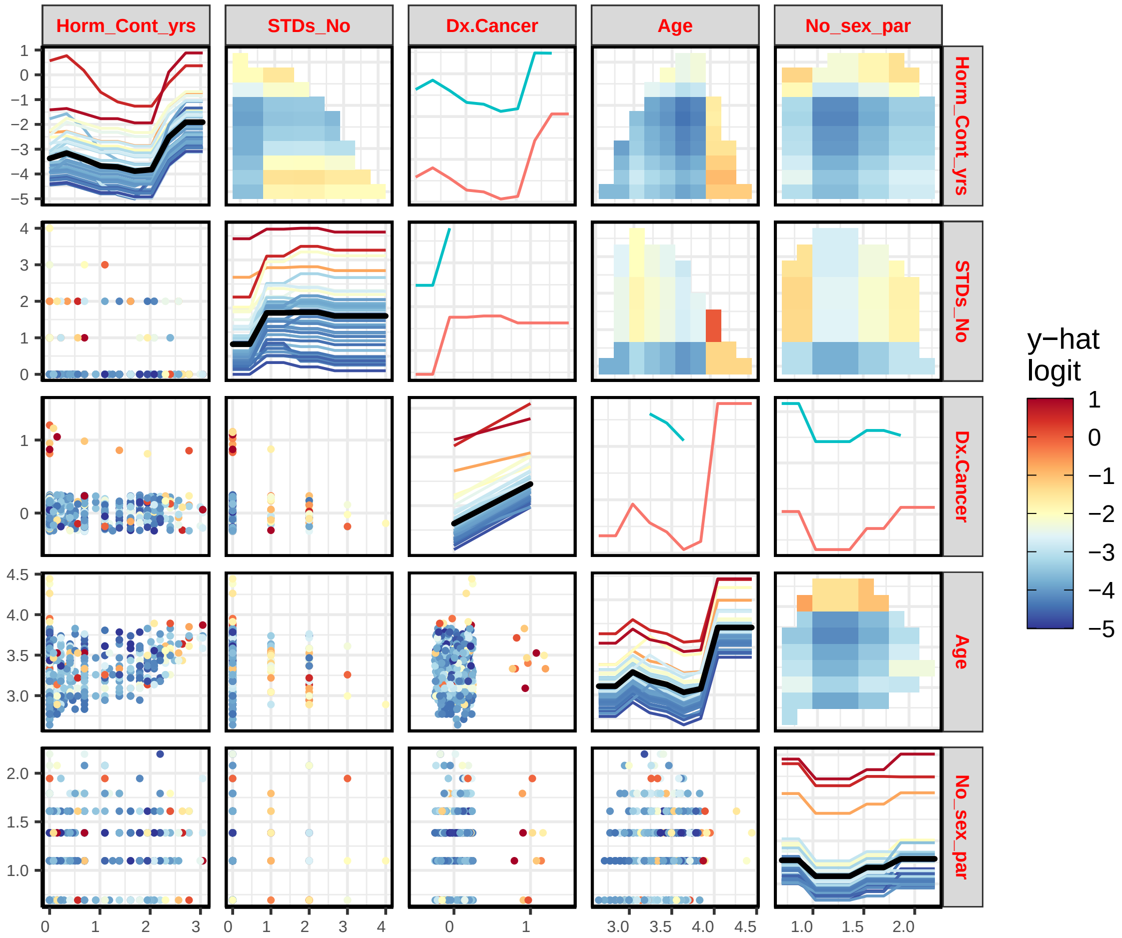


- The first 7 variables have the highest vimp/vint scores.

- Overall Age has the highest importance and interaction score (with No_preg)
- There are a few variables with high interaction but not high importance eg the two variables STDs_No: No_sex_par.
- No_preg is not highly important but has a highly important interaction (with Age)

pdp

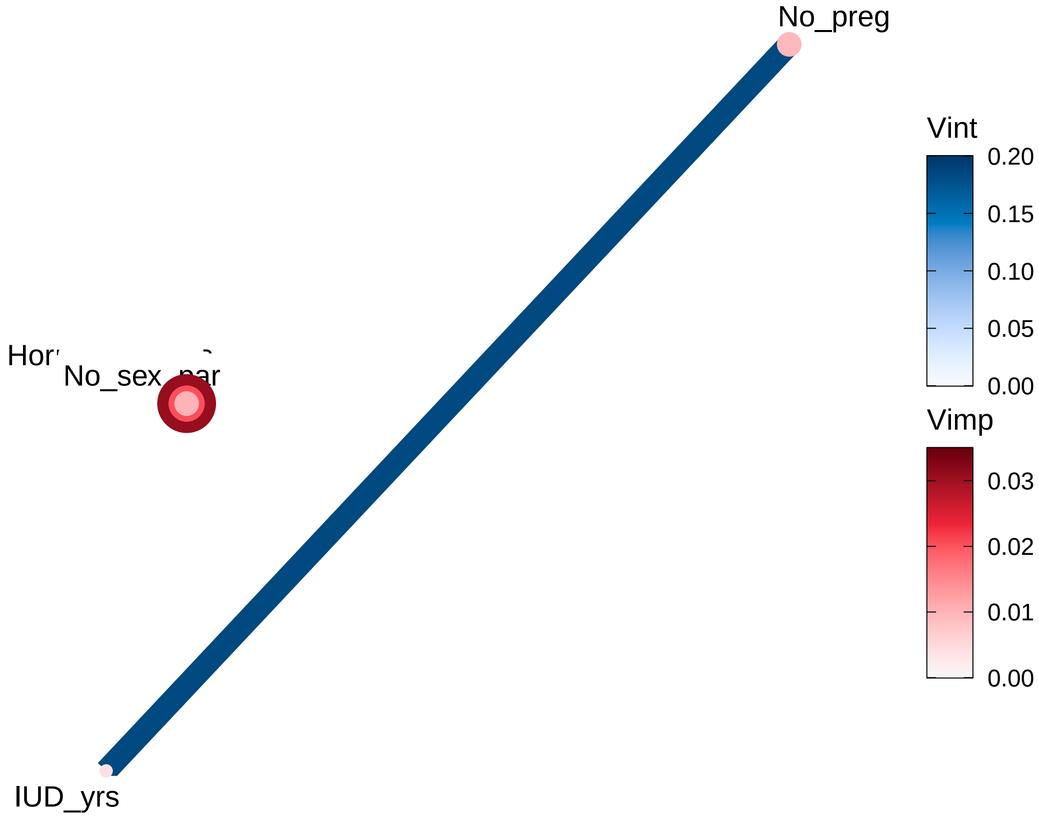
```
# Subsetting top 5 variables:
# sample 50 ice curves - 25 from positive class, 25 from negative class:
set.seed(1701)
yesRows <- sample(which(cTrain$Biopsy == "Cancer"), 25)
noRows <- sample(which(cTrain$Biopsy == "Healthy"), 25)
pdpPairs(data = cTrain,
         fit = cfit,
         response = "Biopsy",
         class = "Cancer",
         nmax = nrow(cTrain),
         nIce = c(yesRows, noRows),
         vars = colnames(viv)[1:5],
         convexHull = TRUE,
         probability = FALSE,fitlims = c(-5,1))
```



- First we look at the Age pdp, as it is the most important predictor. The age pdp curve has lower prob in middle age, the steep incline after that is based on just a few obs with high age, so thissteep incline curve may not be reliable
- Next we investigate No_preg and Age as this pair has the highest h-index. In the bivariate No_preg Age plot, high No_preg is associated with low Cancer prob for middle age groups, but it associated with high Cancer prob for younger ages. This is an interesting interaction.??
- In the bivariate pdps the highest cancer prob occurs for this with high Horm_Cont_yrs and high No_preg.
- Note that in plots of one numeric and one categorical predictor, the numeric variable is always drawn in the x-axis, not withstanding the label is on the y-axis. This is to make the plot easy to read, eg the plot for STD condy and Age (or No_preg stc.)
- In the plot of STD condy and Age, the bivariate pdp is the same as two pdps for each level of STD_condy (the green curve is for STD_condy=1) . It does not look in this plot like an interaction is present, though it has a moderately high H-index.

Network plot

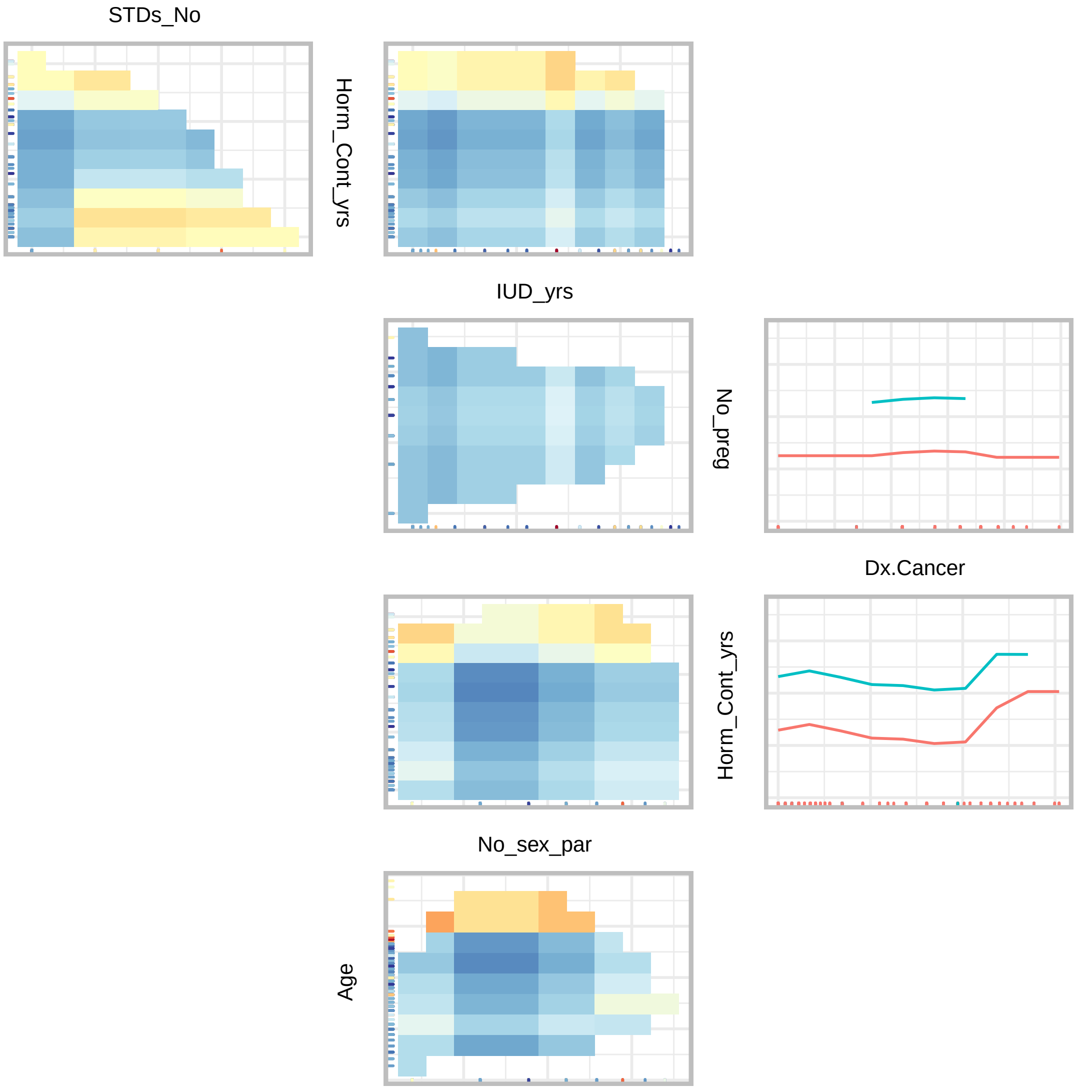
```
viviNetwork(viv, intThreshold = 0.08, removeNode = T,
            # cluster = cutree(h,4),
            layout = igraph::layout_with_lgl
            )
```



- In this plot we check out pairs of variables with h-index over 0.08
- Clustering does not help here so it is omitted
- Not all of these vars are in the top 5 shown in pdp. The extra vars are No_sex_par, STD_No, IUD_yrs

Zen plot

```
zpath <- zPath(viv, cutoff = 0.08) # same as network
set.seed(1701)
pdpZen(data = cTrain,
       fit = cfit,
       response = "Biopsy",
       class = "Cancer",
       zpath = zpath,
       convexHull = TRUE,
       probability = FALSE, fitlims = c(-5,1)
)
```



- There are 8 interactions identified in the network plot, involving 8 variables.
- This would need an 8 by 8 pdp to display byt the zen version is more compact and just shows the selected pairs
- This plot has the same color scale as the pdp plot
- The STD_no:No_sex_par plot is a flat surface and no evidence of interaction
- The IUD_yrs: No_preg plot has prob increasing with IUD_yrs with a steeper gradient for moderately high No_preg. Does this make sense?