Example: step function

Jasper Velthoen

2/20/2020

As a first example we compare the behaviour of gbex with the behaviour of grf based on the simple example that was given in the paper that introduces quantile forests with quantile splitting. The simulation model is given by $Y \sim N(0,(1+I(X_1>0)))$. With covariates $X_1, \dots X_{40}$ all uniformly distributed on the interval $[-1,1]$.

To visualize the comparison we set some the simulation parameters and the model parameters.

```
set.seed(8)
n = 1000
tau_thresh = 0.85
tau_extreme = c(0.95,0.99,0.995)
d = 40
depth = c(1,0)
B = 150
min_leaf_size = c(10,10)
sf = 0.75
```

Next we can simulate the data and estimate the quantile random forest,

```
X = data.frame(X = matrix(runif(n*d,-1,1),ncol=d))
y = rnorm(n)*(1+(X[,1]>0))
fit_grf = grf::quantile_forest(X,y,mtry=d)
```

Given the forest the threshold is estimated, the exceedances are calculated and the gbex model is estimated. Here we use the Out-of-Bag estimate of the quantiles. The reason for this is that the quantiles at exactly the data points has quite bad performance. This is probably related to the fact that the split points occur at exactly the observed covariates themselves.

```
threshold = predict(fit_grf,quantiles=tau_thresh)

Z = y-threshold
y_gbex = Z[Z>0]
X_gbex = X[Z>0,]

fit_gbex = gbex(y_gbex,X_gbex,B = B,depth=depth,sf=sf,min_leaf_size = min_leaf_size,silent=T)
```

Now we have our two estimated models we need a proper comparison. As the only variable that is really of interest is X_1 we fix a grid of 100 points for this variable and take an average over the quantile estimates of 25 randomly sampled sets of covariates for $X_2, \dots X_4$0.

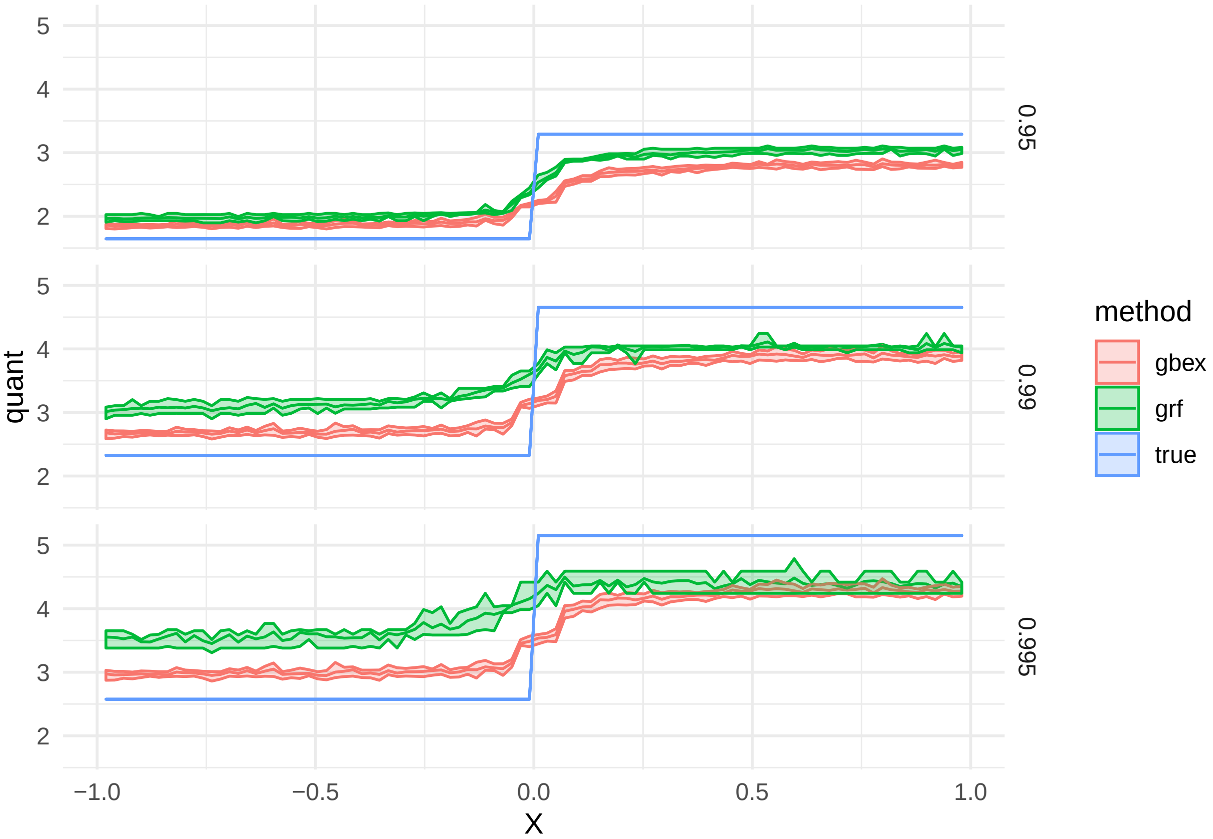
```
Xtest = lapply(seq(-1,1,length.out=100)[-c(1,100)],function(x1){
  res = data.frame(X = matrix(runif(25*d,-1,1),ncol=d))
  res[,1] = x1
  return(res)
}) %>% bind_rows()
colnames(Xtest) = colnames(X)
```

Now we are ready to make the prediction, where for gbex we first estimate the threshold. Note that the probability level for gbex needs the be scaled by the probability of the threshold.

```
threshold_test = predict(fit_grf,newdata=Xtest,quantiles=tau_thresh)
quantiles_gbex = apply(predict(fit_gbex,newdata=Xtest,probs =1-(1-tau_extreme)/(1-tau_thresh),what="quar
                    function(q){
                      q + threshold_test
                    })

quantiles_grf = predict(fit_grf,newdata = Xtest ,quantiles=tau_extreme)

quantiles_true = sapply(tau_extreme,function(tau){qnorm(tau,mean=0,sd=ifelse(Xtest[,1]>0,2,1))})
```

Finally we plot the estimated quantiles and the corresponding 50 percent empirical confidence intervals generated by the 25 randomly sampled other covariates.

```
plot_data = data.frame(X=Xtest[,1],gbex=quantiles_gbex,grf = quantiles_grf,true = quantiles_true)  %>%
  pivot_longer(2:(length(tau_extreme)*3+1),names_to="method",values_to="quantile") %>%
  mutate(probs = tau_extreme[as.numeric(substr(method,nchar(method),nchar(method)))],
         method = substr(method,1,nchar(method)-2)) %>%
  group_by(X,probs,method) %>%
  summarize(quant= mean(quantile),
            quant_min= quantile(quantile,probs=0.25),
            quant_max= quantile(quantile,probs=0.75))

## Make a figure
g= ggplot(plot_data,aes(x=X,y=quant,col=method)) +
  geom_line() +
  geom_ribbon(data=plot_data,aes(x=X,ymin=quant_min,ymax=quant_max,fill=method),alpha=0.25)+
  facet_grid(vars(probs)) +
  theme_minimal()

print(g)
```



For large probability levels the quantile forest essentially forecasts the largest observation. Because there are 39 noise variables. The largest observations ofr $X_1 < 0$ is often still the global maximum. Leading to a constant quantile estimate over all values of X_1.

The probability level of the threshold is very important. First of all the threshold should be high enough such that the gpd approximation holds. Secondly, there should be enough observations such that we can estimate the tail for all covariate levels. Finally it is important to consider errors in the threshold itself. As it is estimated a too high threshold will create a lot of noise in the exceedances which makes the eventual tail quantile estimate worse.

In order to understand a bit better how the model behaves and comapres for different thresholds we set up a small simulation study. For this model. Considering sample sizes of $n = 1000$ and $n = 2000$ we simulate different quantiles and compute mean squared error and bias for two points $X_1 = -0.5$ and $X_1 = 0.5$. With 500 replications we get the following results,

Table 1: Simulation results for step function simulation for $n = 1000$ with normal errors. For two different thresholds 0.8 and 0.9.

		$x = -0.5$				$x = 0.5$			
τ_c	τ	gbex		grf		gbex		grf	
		MSE	$bias$	MSE	$bias$	MSE	$bias$	MSE	$bias$
0.8	0.99	0.165	0.371	0.224	0.317	0.167	-0.259	0.129	-0.062
	0.995	0.28	0.486	0.474	0.484	0.206	-0.206	0.219	-0.071
	0.999	0.706	0.761	1.584	0.961	0.453	-0.04	0.538	-0.226
0.9	0.99	0.100	0.284	0.039	0.104	0.115	-0.235	0.094	-0.015
	0.995	0.188	0.397	0.082	0.166	0.138	-0.196	0.166	-0.025
	0.999	0.558	0.681	0.568	0.51	0.296	-0.039	0.485	-0.114

One observation that can be made from these results is the gbex definately improves on quantile forests

Table 2: Simulation results for step function simulation for $n = 2000$ with normal errors. For two different thresholds 0.8 and 0.9.

		$x = -0.5$				$x = 0.5$			
τ_c	τ	gbex		grf		gbex		grf	
		MSE	$bias$	MSE	$bias$	MSE	$bias$	MSE	$bias$
0.8	0.99	0.166	0.36	0.332	0.434	0.184	-0.322	0.13	-0.083
	0.995	0.283	0.477	0.684	0.644	0.21	-0.281	0.234	-0.079
	0.999	0.737	0.766	2.133	1.207	0.413	-0.13	0.594	-0.223
0.9	0.99	0.221	0.409	0.540	0.658	0.331	-0.466	0.166	-0.100
	0.995	0.36	0.526	1.095	0.961	0.4	-0.471	0.27	-0.105
	0.999	0.913	0.822	3.185	1.678	0.68	-0.405	0.686	-0.31

especially for $X_1 = -0.5$. But the results are not very consistent. As it can be observed that random forests has different behaviour for different thresholds, which is not possible as this thershold is not used for random forests. Secondly gbex does not improve when adding 1000 more observations and the method becomes even worse for $\tau_c = 0.9$. Though this can be contributed to the worse behaviour for grf and hence the threshold quanlity might be very bad.