1 Background

Given data $(y_i, x_i, z_i), i = 1, \ldots, n$ with outcome $y_i \in \mathbb{R}$, covariates $x_i \in \mathbb{R}^p$ and effect modifiers $z_i \in \mathbb{R}^q$ one assumes the varying coefficient model

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j(z_i) + \varepsilon_i = x_i^T\beta(z_i) + \varepsilon_i, \tag{1}$$

[2]. The coefficients $\beta(\cdot) = (\beta_1(\cdot), \ldots, \beta_p(\cdot))^T$ determine a (rather simple) functional, relationship between outcome y and (likely low dimensional) covariate x. The coefficients themselves are considered to be (potentially complex) functions of the (potentially high dimensional) effect modifier z.
Each varying coefficient mapping $\beta_j(\cdot)$ is estimated using an ensemble of gradient boosted decision trees. This is done by iteratively minimizing a loss function

$$L(y, \beta) = \sum_{i=1}^{n} l(y_i, x_i^T\beta(z_i)). \tag{2}$$

The generic gradient boosting algorithm described in [1] is adapted for the varying coefficient scenario. The resulting procedure is sketched in Algorithm 1 below. A similar algorithm has already been described by [3].

Algorithm 1 VCBoost

1: $\beta_j^{(0)}(z) = 0,\ j = 1, \ldots, p$
2: **for** $m = 1, \ldots,$ maxiter **do**
3:     **for** $j = 1, \ldots, p$ **do**
4:         $\tilde{y_i} = -\frac{\partial l(y_i, x_i^T\beta^{(m-1)}(z_i))}{\partial\beta_j^{(m-1)}(z_i)}$ ▷ negative gradient wrt. β_j
5:         $h =$ regression tree fit on $\{\tilde{y_i}, z_i\}_{i=1,\ldots,n}$
6:         $\rho = \text{argmin}_\rho \left[\sum_{i=1}^{n} l(y, x_i^T\beta^{(m-1)}(z_i) + \rho h(z_i)x_{ij})\right]$ ▷ line search
7:         $\beta_j^{(m)}(\cdot) = \beta_j^{(m-1)}(\cdot) + \rho h(\cdot)$

The pseudoresponses $\tilde{y_i}$ in Line 4 are obtained by noting that

$$\tilde{y_i} = -\frac{\partial l(y_i, x_i^T\beta^{(m-1)}(z_i))}{\partial\beta_j^{(m-1)}(z_i)} = -\frac{\partial l(y_i, x_i^T\beta^{(m-1)}(z_i))}{\partial x_i^T\beta^{(m-1)}(z_i)}x_{ij} \tag{3}$$

and the derivative remaining on the right hand side is the usual gradient based on the previous fitted responses $x_i^T\beta^{(m-1)}(z_i)$. Hence, this is easily determined for common loss functions. Also this part of the pseudo response is the same, regardless of the coordinate ($j = 1, \ldots, p$) currently handled in the inner loop. A variation of the algorithm only computes $\frac{\partial l(y_i, x_i^T\beta^{(m-1)}(z_i))}{\partial x_i^T\beta^{(m-1)}(z_i)}$ once during the outer loop and then updates all coefficient estimates before recomputing.
Also the line search determining the step size ρ can be done globally, e.g. determining one ρ based on all observations $i = 1, \ldots n$ as shown above. Alternatively,

the line search can be done separately for each leaf in the regression tree h (see Eq. (18) in [1]).
As usual one can also introduce a learning rate < 1 to shrink the step size for the update in row 7 of the algorithm.
Finally there would also be many hyper-parameters controlling the tree building process in line 5.

1.1 Details for specific loss functions

...

References

[1] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics* (2001), pp. 1189–1232.

[2] Trevor Hastie and Robert Tibshirani. "Varying-coefficient models". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 55.4 (1993), pp. 757–779.

[3] Yichen Zhou and Giles Hooker. "Tree boosted varying coefficient models". In: *arXiv preprint arXiv:1904.01058* (2019).