MARS, PRIM and HME

Wei Li

Syracuse University

Spring 2024

OVERVIEW

Multivariate Adapive Regression Splines (MARS)

Bump hunting PRIM

Hierarchical Mixtures of Experts (HME)

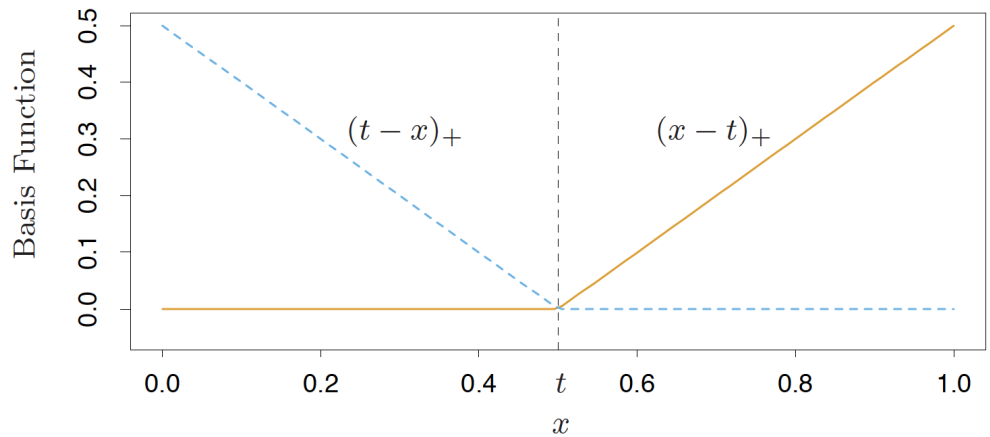
Multivariate Adapive Regression Splines (MARS)

Multivariate Adapive Regression Splines (MARS) (Friedman 19991)

Consider the following two functions (called a *reflected pair* of **hinge functions**)

$$(x-t)_+ = \begin{cases} x-t, & \text{if } x > t \\ 0, & \text{otherwise} \end{cases}$$

$$(t-x)_+ = \begin{cases} t-x, & \text{if } x < t \\ 0, & \text{otherwise} \end{cases}$$



ESL. Fig. 9.9.

For a regression problem $Y = f(X) + \epsilon$:

Training data $(x_1, y_1), \ldots, (x_n, y_n)$ with $y_i \in \mathbb{R}$ and

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^\top \in \mathbb{R}^p$$

For an input vector $X \in \mathbb{R}^p$ define the pairs of **hinge functions**

$$I_0(X, j, i) = (X_j - x_{ij})_+ \quad \text{and} \quad I_1(X, j, i) = (x_{ij} - X_j)_+$$

- Each hinge function depends on a chosen predictor and the knot position.
- $I_0(X, j, i)$ and $I_1(X, j, i)$ are viewed as functions of X, not only of X_j.

The collection of basis functions (linear splines) is given by

$$\mathcal{C} = \{I_0(X, j, i), I_1(X, j, i)\}_{j=1,\ldots,p, i=1,\ldots,n}$$

Now the model for f is given by

$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m\left(X, \boldsymbol{\alpha}_m\right)$$

where each $\boldsymbol{\alpha}_m = (b_{m,1}, j_{m,1}, i_{m,1}; \ldots; b_{m,p_m}, j_{m,p_m}, i_{m,p_m})$ with $b_{m,k} \in \{0, 1\}$ such that

$$h_m\left(X, \boldsymbol{\alpha}_m\right) = \prod_{k=1}^{p_m} I_{b_{m,k}}\left(X, j_{m,k}, i_{m,k}\right)$$

- Intuitively, each $h_m\left(X, \boldsymbol{\alpha}_m\right)$ is a hinge function from $\mathcal{C}$ or a product of two or more hinge functions from $\mathcal{C}$.

$$h_m\left(X, \boldsymbol{\alpha}_m\right) = \prod_{k=1}^{p_m} I_{b_{m,k}}\left(X, j_{m,k}, i_{m,k}\right)$$

For example:

$$h(X, \boldsymbol{\alpha}) = I_0(X, 3, 9) = \left(X_3 - x_{93}\right)_+$$

$$h(X, \boldsymbol{\alpha}) = I_0(X, 1, 5) I_1(X, 2, 7) = \left(X_1 - x_{51}\right)_+ \left(x_{72} - X_2\right)_+$$

Once we have $\mathcal{M} = \{h_m : m = 1, \ldots, M\}$, the coefficients β_m can be estimated by standard linear regression.

The MARS model procedure automatically selects which variables to use, the positions of the kinks/knots in the hinge functions, and how the hinge functions are combined.

There are two-stage involved:

- forward pass
- backward pass

Forward pass

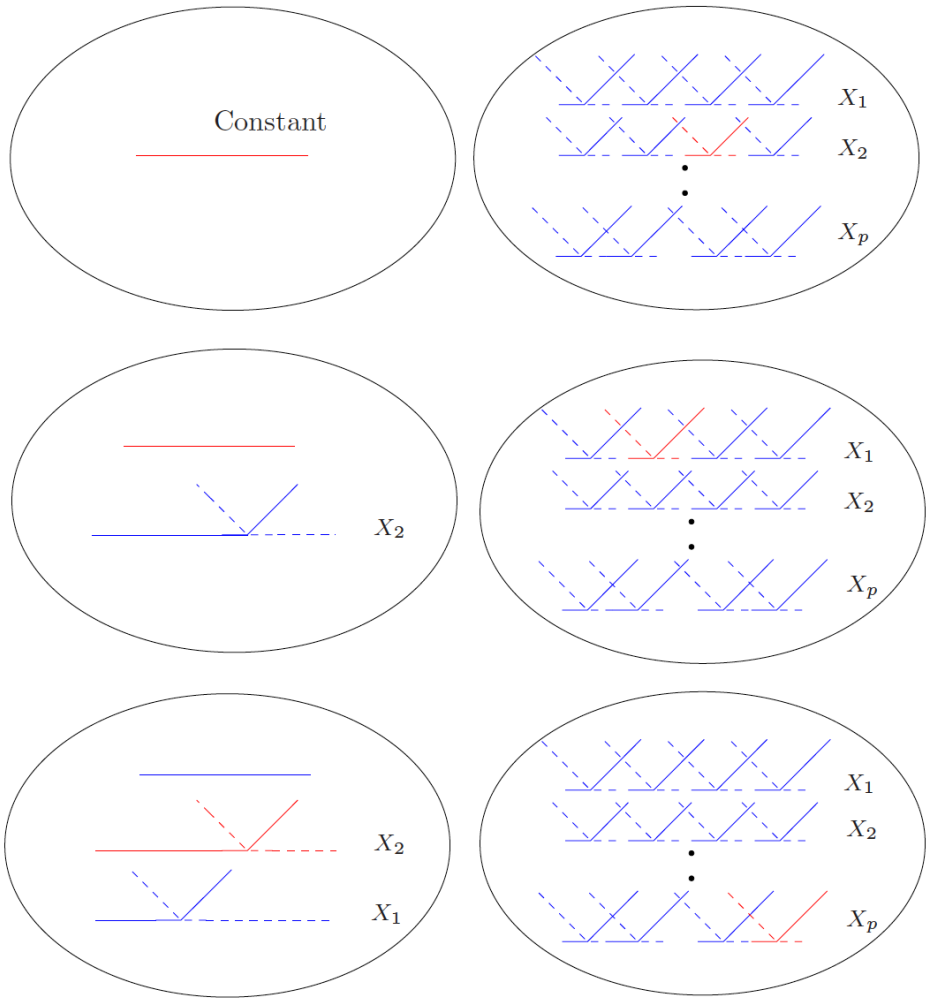
To construct the set $\mathcal{M}$, like a forward stepwise linear regression based on above basis expansion.

- start with $\mathcal{M} = \{h_0\}$ where $h_0 = 1$.
- given current model set $\mathcal{M}$, consider as a new basis function pair all products of a function $h \in \mathcal{M}$ with one of the reflected pairs in $\mathcal{C}$.
- add to the model $\mathcal{M}$ the pair of terms

$$\{h(X)I_0\left(X, j, i\right), h(X)I_1\left(X, j, i\right)\},$$

where $h \in \mathcal{M}, (j, i) \in \{1, \ldots, p\} \times \{1, \ldots, n\}$ that produces the largest decrease in training error.

Forward pass



ESL Fig 9.10. At each stage we consider all products of a candidate pair with a basis function in the model. The product that decreases the residual error the most is added into the current model. The selected functions shown in red.

Algorithm (Forward process):

- $h_0(X, \boldsymbol{\alpha}_0) \equiv 1$, $\mathcal{M} = \{h_0(X, \boldsymbol{\alpha}_0)\}$, M_{max}: pre-set model size, $m = 0$.
- while $m < M_{max}$ perform the following:
  - for each $(k, j, i) \in \{0, \ldots, m\} \times \{1, \ldots, p\} \times \{1, \ldots, n\}$
    1. Consider attempting to augment $\mathcal{M} = \{h_0(X, \alpha_0), \ldots, h_m(X, \alpha_m)\}$ - with a pair of functions
    $$h_k(X, \boldsymbol{\alpha}_k) I_0(X, j, i) \text{ and } h_k(X, \boldsymbol{\alpha}_k) I_1(X, j, i)$$
    2. Use standard linear regression to estimate the $\hat{\beta}$'s
    $$f_{\text{try},k,j,i}(X) = \sum_{l=0}^{m} \hat{\beta}_l h_l(X, \boldsymbol{\alpha}_l) + \hat{\beta}_{m,1} h_k(X, \boldsymbol{\alpha}_k) I_0(X, j, i) + \hat{\beta}_{m,2} h_k(X, \boldsymbol{\alpha}_k) I_1(X, j, i)$$
    3. Compute and record training error of $f_{\text{try},k,j,i}(X)$
  - Let (k^*, j^*, i^*) be triplet producing lowest training error.
  - Add to $\mathcal{M}$ the pair
  $$\{h_{k^*}(X, \boldsymbol{\alpha}_{k^*}) I_0(X, j^*, i^*), h_{k^*}(X, \boldsymbol{\alpha}_{k^*}) I_1(X, j^*, i^*)\},$$
  call it $\{h_{m+1}(X, \boldsymbol{\alpha}_{m+1}), h_{m+2}(X, \boldsymbol{\alpha}_{m+2})\}$;
  $\boldsymbol{\alpha}_{m+1} = \boldsymbol{\alpha}_{k^*} \cup \{0, j^*, i^*\}$, $\boldsymbol{\alpha}_{m+2} = \boldsymbol{\alpha}_{k^*} \cup \{1, j^*, i^*\}$.
  - Update $m \leftarrow m + 2$.

Backward pass

If $\lambda := |\mathcal{M}| = M + 1$ is too large, the model tends to overfit. We may apply **backward deletion procedure**. The term whose removal causes the smallest increase in residual squared error is deleted from the model at each stage.

- To choose λ, one may use CV.
- A more convenient approach is GCV:

$$\mathrm{GCV}(\lambda) = \frac{\sum_{i=1}^{n} \left(y_i - \hat{f}_\lambda (x_i) \right)^2}{(1 - \nu(\lambda)/n)^2}$$

  The value $\nu(\lambda)$ is the effective number of parameters in the model.
- r linearly independent basis functions in the model,
- $K = (r-1)/2$ hinge knots were selected in the forward process
- The formula is $\nu(\lambda) = r + cK$, with $c = 3$
  - When the model is restricted to be additive a penalty of $c = 2$ is used.

Remarks

In the forward process, some useful restrictions.

- There is one restriction put on the formation of model terms: each input can appear at most once in a product.
  - This prevents the formation of higher-order powers of an input.
- A useful option in the MARS procedure is to set an upper limit on the order of interaction.
- Another option is to specify that interactions are allowed only for certain input variables.

Remarks

- The MARS method and algorithm can be extended to handle classification problems.
- For two classes, one can code the output as 0/1 and treat the problem as a regression
- For more than two classes, one can use the indicator response approach. One codes the K response classes via 0/1 indicator variables, and then performs a multi-response MARS regression. There are, however, potential **masking** problems with this approach, just as using multi-response linear regression model for classification.
- Mars can handle "mixed" predictors—quantitative and qualitative. MARS considers *all possible* binary partitions of the categories for a qualitative predictor into two groups. Each such partition generates a pair of piecewise constant basis functions— indicator functions for the two sets of categories.

Remarks

- MARS (like recursive partitioning) does automatic variable selection (meaning it includes important variables in the model and excludes unimportant ones). However, there can be some arbitrariness in the selection, especially when there are correlated predictors, and this can affect interpretability.
- With MARS models, as with any non-parametric regression, parameter confidence intervals and other checks on the model cannot be calculated directly (unlike linear regression models)
- Missing values in MARS can be handled in a similar fashion as the trees by surrogate splits.
- MARS models do not give as good fits as boosted trees, but can be built much more quickly and are more interpretable.

Relationship between MARS and CART

If the MARS procedure is amended so that

1. Set $I_0(X, j, i) = 1\,(X_j - x_{ij} > 0)$ step function
2. Set $I_1(X, j, i) = 1\,(X_j - x_{ij} \leq 0)$ step function
3. When a term $h \in \mathcal{M}$ is chosen at one iteration, e.g.,

$$\mathcal{M} \leftarrow \mathcal{M} \cup \{h(X)I_0(X, j, i)\} \cup \{h(X)I_1(X, j, i)\}$$

remove h from $\mathcal{M}$.

Bump hunting PRIM

Bump hunting PRIM (Friedman and Fisher 1999)

The patient rule induction method (**PRIM**) finds boxes in the feature space, but seeks boxes in which the response average is high. Hence it looks for *maxima* (or a plateau of high points) in the target function, an exercise known as bump hunting.

Its goal is to produce a box B within which the target mean

$$\bar{f}_B = \int_{x \in B} f(x)p(x)dx / \int_{x \in B} p(x)dx$$

is as large as possible (with certain restriction that the size of B not too small).

The mean response on a box is

$$\bar{y}_B = \frac{\sum_{x_i \in B} y_i}{\sum_{x_i \in B} 1}$$

This procedure is coined as a patient method due to its slow, stepwise process

Algorithm

1. Start with a maximal box containing all of the data.
2. *Peeling*: shrinking the box by compressing one face, peeling off the proportion α of observations having either the highest or lowestt values of a predictor X_j,, choose the peeling that produces the highest response mean in the remaining box. ($\alpha = 0.05$ or 0.10.)
3. Repeat step 2 until the proportion of data within the current box β_B falls below some threshold value β_0 (small).

$$\beta_B := \frac{1}{n} \sum_{i=1}^{n} 1\left(\mathbf{x}_i \in B\right) \leq \beta_0$$

   This process produces a decreasing sequence of nested subboxes.
4. *Pasting*: expand the box along any face, as long as the resulting box mean increases, producing an increasing sequence of nested subboxes.
5. Steps $1-4$ give a sequence of boxes, with different numbers of observations in each box. Choose the box with the highest $\bar{y}_B$. Call the box B_1.
6. Remove the data in box B_1 from the dataset and repeat steps $2-5$ to obtain a second box, and continue to get as many boxes.

Formally

Box B is defined by the set of inequalities

$$a_j \leq X_j \leq b_j \quad \text{for } j = 1, \ldots, p$$

where p is the dimension of the feature vectors.

$B' = \text{NewBox}(B, k, 0, a)$ is defined by the inequalities

$$\begin{aligned} a_j \leq X_j \leq b_j &\text{ for } j = 1, \ldots, k-1 \\ a \leq X_k \leq b_k & \\ a_j \leq X_j \leq b_j &\text{ for } j = k+1, \ldots, p \end{aligned}$$

$B' = \text{NewBox}(B, k, 1, b)$ is defined by the inequalities

$$\begin{aligned} a_j \leq X_j \leq b_j &\quad \text{for } j = 1, \ldots, k-1 \\ a_k \leq X_k \leq b & \\ a_j \leq X_j \leq b_j &\quad \text{for } j = k+1, \ldots, p \end{aligned}$$

Let $n_B = \#$ of training observations in box B.

Peeling

- Define $B' = \text{Peel}(B, k, 0, \alpha)$ to be the box

$$B' = \text{NewBox}(B, k, 0, a)$$

where a is the smallest scalar for $\alpha \in (0, 1)$ s.t.

$$a > a_k \text{ and } n_{B'} \leq (1 - \alpha) n_B$$

- Define $B' = \text{Peel}(B, k, 1, \alpha)$ to be the box

$$B' = \text{NewBox}(B, k, 1, b)$$

where b is the largest scalar for $\alpha \in (0, 1)$ s.t.

$$b < b_k \text{ and } n_{B'} \leq (1 - \alpha) n_B$$

Peeling step 2-3 works as in the following:

- Let C_i denote the current box.
- Compute the proposed trimmed boxes $C_{0,k} = \text{Peel}\,(C_i, k, 0, \alpha)$ and $C_{1,k} = \text{Peel}\,(C_i, k, 1, \alpha)$ for $k = 1, \ldots, p$
- Choose the C_* from $\{C_{0,k}, C_{1,k}\}$ that gives highest response mean.
- Set $C_{i+1} = C_*$.

pasting

- Define $B' = \text{ExpandBox}\ (B, k, 0, \alpha)$ to be the box

$$B' = \text{NewBox}(B, k, 0, a)$$

where a is the largest scalar for $\alpha \in (0, 1)$ s.t.

$$a < a_k \text{ and } n_{B'} \geq (1 + \alpha) n_B$$

- Define $B' = \text{ExpandBox}\ (B, k, 1, \alpha)$ to be the box

$$B' = \text{NewBox}(B, k, 1, b)$$

where b is the smallest scalar for $\alpha \in (0, 1)$ s.t.

$$b > b_k \text{ and } n_{B'} \geq (1 + \alpha) n_B$$

Pasting step 4 works as in the following:

- Let C_i denote the current box.
- $C_{0,k} = \text{ExpandBox}\left(C_i, k, 0, \alpha\right), C_{1,k} = \text{ExpandBox}\left(C_i, k, 1, \alpha\right)$ for $k = 1, \ldots, p$
- Choose the C_* from $\{C_{0,k}, C_{1,k}\}$ that gives highest response mean.
- Set $C_{i+1} = C_*$ if $\bar{y}_{C^*} > \bar{y}_{C_i}$.

Tuning

The meta parameter β_0 is the proportion of observations in the box.

- a small value of β_0 tends to be consistent with accurate estimation of a maximum of $f(\boldsymbol{x})$.
- yet too small value is counter productive to locate the bump, due to the noise
- the peeling process yields values of $\bar{y}_B$ that may increase whereas the value of $\bar{f}_B$ actually decreases.
  - "overfitting"

In step 5, a cross validation is often used to choose the optimal box.

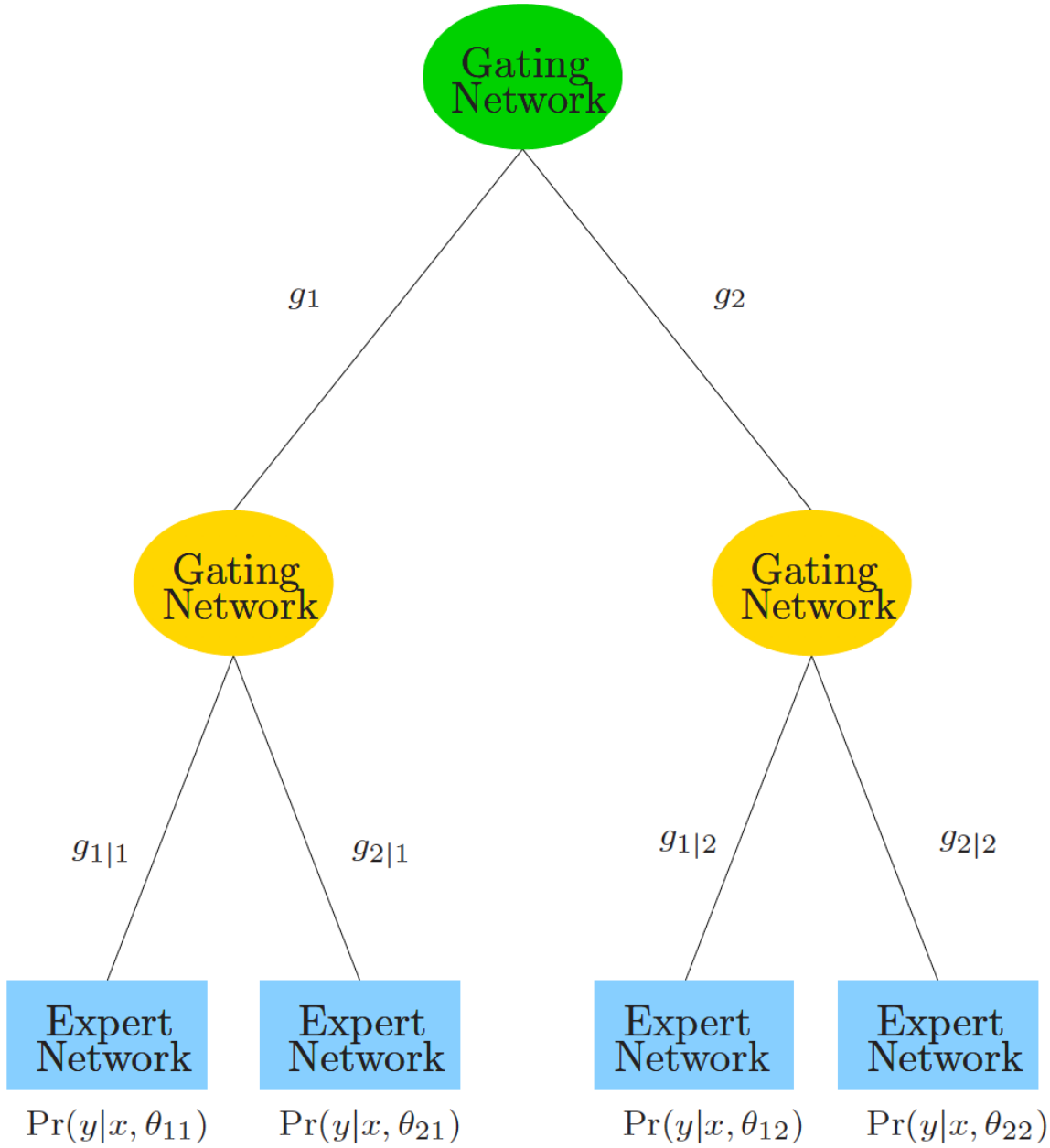
Remarks

1. PRIM is designed for regression (quantitative response)
2. For $K = 2$ class, can be handled using 0/1 coding.
3. For $K > 2$, no simple generalization. But can run PRIM separately for each class versus the baseline class.
   - Alternative, can find a clump of points of maximal purity.

Hierarchical Mixtures of Experts (HME)

Hierarchical Mixtures of Experts (HME) (Jordan and Jacobs 1994)

- Variant of tree-base methods.
- Tree splits are soft probabilistic ones as opposed to hard ones.
  - parameter estimation - optimize a smooth cost function
  - prediction accuracy - avoids discontinuities in the response function
- non-terminal nodes are called gating networks.
- Splits can be multi-way.
- Splits are probabilistic functions of a linear combination of inputs.
- Terminal nodes called experts where a linear (or logistic regression) model is fit


Gating Network
g_1
g_2
Gating Network
Gating Network
$g_{1|1}$
$g_{2|1}$
$g_{1|2}$
$g_{2|2}$
Expert Network
Expert Network
Expert Network
Expert Network
$\Pr(y|x, \theta_{11})$
$\Pr(y|x, \theta_{21})$
$\Pr(y|x, \theta_{12})$
$\Pr(y|x, \theta_{22})$

ESL Fig 9.13.

Top gate network output is a soft K-way split:

$g_j\left(x, \gamma_j\right) = \frac{e^{\gamma_j^\top x}}{\sum_{k=1}^{K} e^{\gamma_k^\top x}}$, for $j = 1, \ldots, K$ the prob of assigning x to the j th branch.

The second level has a similar split:

$g_{l|j}\left(x, \gamma_{jl}\right) = \frac{e^{\gamma_{jl}^\top x}}{\sum_{k=1}^{K} e^{\gamma_{jk}^\top x}}$, for $l = 1, \ldots, K$, the prob of assigning to l th branch given previous assignment to j th branch.

The expert node model the response $Y \sim P\left(y \mid x, \theta_{jl}\right)$:

- Regression: Gaussian linear regression model

$$P\left(Y \mid x, \theta_{jl}\right) = \mathcal{N}\left(\beta_{jl}^{\top} x, \sigma_{jl}^{2}\right) \qquad \text{where } \theta_{jl} = \left(\beta_{jl}, \sigma_{jl}^{2}\right)$$

- Classification: linear logistic regression model

$$P\left(Y = 1 \mid x, \theta_{jl}\right) = \frac{1}{1 + e^{-\theta_{jl}^{\top} x}}$$

let $\Psi = \{\gamma_j, \gamma_{jl}, \theta_{jl}\}$,

$$P(y \mid x, \Psi) = \sum_{j=1}^{K} g_j\left(x, \gamma_j\right) \sum_{l=1}^{K} g_{l|j}\left(x, \gamma_{lj}\right) P\left(y \mid x, \theta_{jl}\right)$$

Estimate Ψ by maximizing the log-likelihood:

$$\max_{\Psi} \sum_{i=1}^{n} \log P\left(y_i \mid x_i, \Psi\right)$$

Advantages of HMEs over CART:

- Smooth final regression function. Soft splits allow for smooth transitions from high to low responses.
- Easier to optimize for parameters. The log-likelihood is a smooth function and is amenable to numerical optimization.

Disadvantage of HMEs over CART:

- Tree topology? No good way to find it for HME.
- Harder to interpret the model. Not so clear cut which factors cause which effects.