Non-linear Least Squares

unknow true model: $Y^* = f_{\beta^*}(X) + \varepsilon$ $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

$f_{\beta^*}(X)$: non-linear fct with true parameters β^*

[special case: $f_{\beta^*}(X) = X \cdot \beta^*$ (linear fct.) $\Rightarrow$ ordinary least squares]

goal: $\hat{\beta} = \underset{\beta}{\arg\min} \, \|Y - f_\beta(X)\|^2$, in general no analytic solution

many approximations:

FML
- regression trees, regression forests
- Levenberg-Marquardt algorithm
- change of variables to transform non-linear probl. into a linear one (approximately)
- kernel methods

Advanced Machine Learning
- Gaussian processes
- neural networks
- ...

Regression Trees

work like density and decision trees: partition the feature space into bins using a tree, have a constant response for every bin:

$$\hat{Y}_i = \hat{f}(X_i) = \underbrace{\sum_{l=1}^{L}}_{\text{sum over all bins}} \underbrace{f_l}_{\text{constant bin response}} \underbrace{\mathbb{1}[X_i \in \text{bin}_l]}_{\text{tree to find the bin for } X_i}$$

find optimal response in bin_l via least squares loss (assume bins are already fixed):

$$\hat{f}_l = \int_{x \in \text{bin}_l} (f_l - f(x))^2 \, dx \qquad \hat{f}(x_i) = \sum_{l=1}^{L} f_l \, \mathbb{1}[x_i \in \text{bin}_l]$$

$$\{\hat{f}_l\} = \arg\min_{\{f_l\}} \int (\hat{f}(x) - f(x))^2 \, p(x) \, dx$$

$$= \arg\min_{\{f_l\}} \underbrace{\int f(x)^2 p(x) \, dx}_{\substack{\text{independent of model} \\ \Rightarrow \text{drop}}} - 2 \int \hat{f}(x) f(x) \, p(x) \, dx + \int \hat{f}(x)^2 p(x) \, dx$$

$$= \arg\min_{\{f_l\}} -2 \sum_{l=1}^{L} f_l \int_{x \in \text{bin}_l} f(x) \, p(x) \, dx + \sum_l f_l^2 \int_{x \in \text{bin}_l} p(x) \, dx$$

$$\frac{\partial \text{Loss}}{\partial f_l} = -2 \underbrace{\int_{x \in \text{bin}_l} f(x) \, p(x) \, dx}_{\substack{\mathbb{E}_{\text{bin}_l}[f(X)] \\ \downarrow}} + 2 f_l \underbrace{\int_{x \in \text{bin}_l} p(x) \, dx}_{\text{total density in bin}_l} \overset{!}{=} 0$$

$$\approx \left(\frac{1}{N_l} \sum_{i: x_i \in \text{bin}_l} f(x_i) \right) \cdot \frac{N_l}{N} + f_l \frac{N_l}{N} \overset{!}{=} 0 \qquad \hat{f}_l = \frac{1}{N_l} \sum_{i: x_i \in \text{bin}_l} \overbrace{f(x_i)}^{= y_i}$$

$$\boxed{\hat{f}_l = \frac{1}{N_l} \sum_{i \in \text{bin}_l} y_i}$$

regression ~~forest~~ tree: piece-wise constant approximation of a non-linear function

truth: $Y = f(X) + \varepsilon$ approx: $Y = \hat{f}(X) = \sum_{l=1}^{L} \hat{f}_l \, \mathbb{1}[X \in \text{bin}_l]$

minimize the squared loss

$$\{\hat{f}_l\}_{l=1}^{L} = \underset{\{f_l\}}{\arg\min}\; \mathbb{E}_{p(x)}\left[\left(f(x) - \hat{f}(x)\right)^2\right] = \underset{\{f_l\}}{\arg\min} \int \left(f(x) - \hat{f}(x)\right)^2 p(x)\,dx$$

$$= \underset{\{f_l\}}{\arg\min}\; -2 \sum_{l=1}^{L} f_l \underbrace{\int_{x \in \text{bin}_l} f(x)\, p(x)\,dx}_{\sum_{i: x_i \in \text{bin}_l} Y_i (= f(x_i)) \frac{1}{N}} + \sum_{l=1}^{L} f_l^2 \underbrace{\int_{x \in \text{bin}_l} 1 \cdot p(x)\,dx}_{\sum_{i: x_i \in \text{bin}_l} 1 \cdot \frac{1}{N}}$$

$$= \underset{\{f_l\}}{\arg\min}\; -2 \sum_{l=1}^{L} f_l \cdot \frac{1}{N} \sum_{i: x_i \in \text{bin}_l} Y_i + \sum_{l=1}^{L} f_l^2 \frac{N_l}{N}$$

$$\frac{\partial \text{Loss}}{\partial f_l} = -2 \frac{1}{N} \sum_{i: x_i \in \text{bin}_l} Y_i + 2 f_l \frac{N_l}{N} \overset{!}{=} 0$$

$$\boxed{\hat{f}_l = \frac{1}{N_l} \sum_{i: x_i \in \text{bin}_l} Y_i}$$

training alg.: exactly like decision & density tree, just with different criterion for selecting the optimal split: in each node to be split, iterate over all candidate splits and compute the squared error: $f_{\text{left}} = \frac{1}{N_{\text{left}}} \sum_{i: x_i \in \text{left}} Y_i$ $f_{\text{right}} = \frac{1}{N_{\text{right}}} \sum_{i: x_i \in \text{right}} Y_i$

execute split with minimal R: $R = \sum_{i: x_i \in \text{left}} (Y_i - f_{\text{left}})^2 + \sum_{i: x_i \in \text{right}} (Y_i - f_{\text{right}})^2$

regression forest: to avoid overfitting, train an ensemble of regression trees with the usual tricks (each tree is trained with a bootstrap sample of the TS, in each node a random subset of the features is considered for splitting) and return the average response of all trees

Levenberg - Marquardt - Algorithm

setting here: $Y = f_\beta(X) + \epsilon$ with $f_\beta(X)$ a non-linear function with ~~params~~ parameters β, functional form (formula) of f_β is known, only the true parameters β^* are unknown

[notice: linear models ~~$X\beta$~~ $f_\beta(X) = X \cdot \beta$ are a special case]

difficulty: - for $f_\beta(x)$ non-linear, there is usually no analytic solution for β

$\Rightarrow$ reduce to a sequence of linear problems via Taylor series, iteratively solve non-linear optimization by solving a least-squares probl. in every iteration

least-squares objective: $\hat\beta = \arg\min\limits_{\beta} \sum\limits_{i=1}^{N} (Y_i - f_\beta(X_i))^2$

- iterative Gauss-Newton alg. want to minimize $\mathcal{L}(\beta)$
  - given current guess $\beta^{(t-1)}$, expand $\mathcal{L}(\beta)$ into Taylor series at $\beta^{(t-1)}$

$$\mathcal{L}(\underbrace{\beta^{(t-1)} + \Delta}_{=\beta^{(t)}}) \approx \mathcal{L}(\beta^{(t-1)}) + \underbrace{\frac{\partial \mathcal{L}}{\partial \beta}\bigg|_{\beta^{(t-1)}}}_{\text{"Jacobian of the loss"}} \cdot \Delta \quad + \text{higher orders (ignored)}$$

- least-squares minimization ~~of~~ w.r.t. improvement Δ

$$\hat{\Delta} = \arg\min_{\Delta} \Big(\underbrace{L(\beta^{(t-1)})}_{\tilde{Y}} + \underbrace{\frac{\partial L}{\partial \beta}\Big|_{\beta^{(t-1)}}}_{-\tilde{X}} \cdot \Delta \Big)^2$$

auxilliary variables: (vector of length N) (matrix of derivatives N × D)

$$\hat{\Delta} = \arg\min_{\Delta} \left(\tilde{Y} - \tilde{X} \cdot \Delta \right)^2 \qquad \hat{\Delta} = (\tilde{X}^T \tilde{X})^{-1} \cdot \tilde{X}^T \cdot \tilde{Y}$$

solution via pseudo-inverse

not yet optimal, because it tends to overfit => L_2 regularization

$$\hat{\Delta} = \arg\min_{\Delta} \left(\tilde{Y} - \tilde{X} \cdot \Delta \right)^2 + \tau \|\Delta\|^2$$

- in case of non-linear least squares: ~~$L(x)$~~

$$L(\beta) = Y - f_\beta(X)$$

$$L(\beta^{(t-1)} + \Delta) = \underbrace{L(\beta^{(t-1)})}_{= Y - f_{\beta^{(t-1)}}(X)} - \underbrace{\frac{\partial f_\beta(X)}{\partial \beta}\Big|_{\beta^{(t-1)}}}_{\tilde{X}} \cdot \Delta$$

$$= \tilde{Y} - \tilde{X} \cdot \Delta$$

optimize via ridge-regression: $\hat{\Delta} = (\tilde{X}^T \tilde{X} + \tau \mathbb{I})^{-1} \tilde{X}^T \tilde{Y}$

$$\beta^{(t)} = \beta^{(t-1)} + \Delta$$

- why is it good to use ridge regression?

case 1: $\tilde{X}^T\tilde{X}$ has good condition ⇒ $\tau \approx 0$ (no regularization needed)

⇒ $\hat{\Delta}$ is unbiased least-squares solution ⇒ very fast convergence (few iterations)

case 2: $\tilde{X}^T\tilde{X}$ has bad condition ⇒ $\tau \gg 0$,

in the extreme, $\tilde{X}^T\tilde{X}$ is dominated by $\tau\mathbb{I}$

$$(\tilde{X}^T\tilde{X} + \tau\mathbb{I})^{-1}\tilde{X}^T\tilde{Y} \Longrightarrow \frac{1}{\tau}\tilde{X}^T\tilde{Y}$$

⇒ we get a gradient descent alg. with learning rate $\frac{1}{2\tau}$

$$\left.\frac{\partial(\tilde{Y} - \tilde{X}\cdot\Delta)^2}{\partial\Delta}\right|_{\Delta=0} = -2\tilde{X}^T\tilde{Y}$$

⇒ τ allows us to interpolate between fast Newton iterations and slower, but numerically stable, gradient descent

- Questions: - how to choose good τ? ⇒ self-regulating algorithm
  - how to ensure that τ has the same effect on all features

[recall: in ridge regression, we standardized the features $\tilde{X}_j = \frac{X_j - \bar{X}_j}{\mathrm{std}(X_j)}$]

standardization is now impossible, because $\tilde{X}$ is the Jacobian matrix $\left(\frac{\partial f}{\partial\beta}\right)$, which would become incorrect by scaling

⇒ Marquardt's trick: weight τ with $\|\tilde{X}_j\|^2 = \mathrm{diag}(\tilde{X}^T\tilde{X})$

$$\hat{\Delta} = (\tilde{X}^T\tilde{X} + \tau\,\mathrm{diag}(\tilde{X}^T\tilde{X}))^{-1}\cdot\tilde{X}^T\tilde{Y}$$

(in practice: compute $S = \tilde{X}^T\tilde{X}$, then scale diagonal elements $S_{jj} \Leftarrow S_{jj}(1+\tau)$

Full Levenberg - Marquardt algorithm

```
(0) define initial guess $\beta^{(0)}$ (this is usually critical, bad guess => bad optimum)
        initial regularization strength $\tau^{(0)}$
(1) for $t = 1, \dots, T$ (or until convergence):
    (a) compute $\tilde{Y} = Y - f_{\beta^{(t-1)}}(X)$ , $\tilde{X} = \left.\frac{\partial f_\beta(X)}{\partial \beta}\right|_{\beta^{(t-1)}}$
    (b) solve $(\tilde{X}^T \tilde{X} + \tau^{(t-1)} \cdot \operatorname{diag}(\tilde{X}^T \tilde{X})) \cdot \Delta = \tilde{X}^T \cdot \tilde{Y}$ by some
            linear solver => $\hat{\Delta}$
    (c) compute $\hat{\beta} = \beta^{(t-1)} + \hat{\Delta}$ and residuals $(Y - f_{\hat{\beta}}(X))^2$
    (d) if residual (squared loss) of $\hat{\beta}$ is smaller than of $\beta^{(t-1)}$
            - accept $\hat{\beta}$ as our new guess $\beta^{(t)} = \hat{\beta}$
            - reduce regularization strength $\tau^{(t)} = \tau^{(t-1)} / \overset{\text{hyper param}}{\sqrt{2}}$
        otherwise
            - reject the guess $\hat{\beta}$ : $\beta^{(t)} = \beta^{(t-1)}$
            - increase the regularization $\tau^{(t)} = \sqrt{2}\, \tau^{(t-1)}$
```

$$\operatorname{diag}(A) = \begin{bmatrix} A_{11} & & 0 \\ & A_{22} \ddots & \\ 0 & & A_{ND} \end{bmatrix}$$