Lecture January 29

Boosting (XGBoost)
not in
scikit-learn

Basic idea is to use a weak learner/simple approximation and iterate based upon this;

Example: Regression

$$C(f) = \frac{1}{n} \sum_i \left(y_i - f(x_i)\right)^2$$

Data set $\{(x_0, y_0), (x_1, y_1) \ldots (x_{n-1}, y_{n-1})\}$

- Define a function

$$f(x) = f_M(x)$$

simple

$$f_M(x) = \sum_{m=0}^{M} \beta_m \, b(x; \gamma_m)$$

often $f_0(x) = 0$

- want to minimize

$$\hat{\beta}, \hat{\gamma} \quad \arg\min \, C(f_M)$$

$\beta \gamma$

- Example

$$b(x;\gamma) = 1 + \gamma x$$

$M = 1$

$$f_1(x) = f_0(x) + \underline{\beta_1 b(x;\gamma_1)}$$

$$f_m(x) = f_{m-1}(x) + \beta_m b(x;\gamma_m)$$

$$(\hat{\beta}_m, \hat{\gamma}_m) = \underset{\beta,\gamma}{\arg\min} \sum_{i=0}^{n-1} \left(y_i - f_{m-1}(x_i) - \beta b(x_i;\gamma)\right)^2$$

Take derivatives wrt γ, β

$$\frac{\partial c}{\partial \beta} = -2 \sum_i (1+\gamma x_i)(y_i - \beta(1+\gamma x_i)) = 0$$

$$\frac{\partial c}{\partial \gamma} = -2 \sum_i \beta x_i (y_i - \beta(1+\gamma x_i)) = 0$$

Solve with SGD in general.

<u>algorithm for regression</u>

- initialize $f_0(x;\gamma)$
- have a model for $\underline{b(x;\gamma)}$

- for $m = 1 : M$

  a) optimize and compute

  $$(\hat{\beta}_m, \hat{\gamma}_m) = \underset{\beta\,\gamma}{\arg\min} \sum_{i=0}^{n-1} \mathcal{L}\left(y_i, f_{m-1}(x_i;\gamma_{m-1}) + \beta\, b(x_i;\gamma)\right)$$

  b) set $f_m(x;\gamma) = f_{m-1}(x;\gamma_{m-1}) + \beta_m \times b(x;\gamma_m)$

End for, continue til M

$$f(x) = f_M(x) = \sum_{m=0}^{M} \beta_m\, b(x;\gamma_m)$$

This type of additive expansion is at the heart of many learning techniques.

Neural network:

$$b(x;\gamma) = \sigma(\gamma_0 + \gamma_1 x)$$

Sigmoid

$$\sigma(z) = \frac{1}{1+e^{-z}}$$