Lecture November 12

Boosting methods

model $f(x) = \sum_{m=0}^{M} \beta_m b_m(x)$

$b_m(x)$ = weak learner.

- Restrictive model

$$f(x_i) = \sum_{j=0}^{p-1} f_j(x_i)$$
$$= \sum_{j=0}^{p-1} \sum_{m=0}^{M} \beta_{jm} b_{jm}(x_i)$$

linear regression and polynomial expansion.

- Selection models

include only those basis functions b_m that contribute significantly. Random forests, Bagging & boosting models belong here.

- Regularization models

Ridge is a simple example. Screen away selected features.

0

Boosting :

$$f(x) = f_0(x) + \sum_{m=1}^{M} b_m(x)\beta_m$$

weak learner.

Put emphasis in the iteration on misclassified data.

$$C(f) = \sum_{i=0}^{n-1} \mathcal{L}(y_i, f(x_i))$$

$$\hat{f} = \underset{f}{\arg\min}\ C(f)$$

In linear regression

$$MSE = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - f(x_i))^2$$

L2 boosting for regression

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

(i) Establish a cost function

$$C(f) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - f_m(x_i))^2$$

algo:

a) initialize $f_0(x)$

b) for $m = 1 : M$

$$(\hat{\beta}_m, \hat{\gamma}_m) = \underset{\gamma\, \beta}{\operatorname{argmin}}$$

$$\frac{1}{n} \sum_i \left(y_i - f_{m-1}(x_i) - \beta \underbrace{b(x;\gamma)}_{\text{decision tree}}\right)^2$$

c) determine

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

end for

Return $f(x) = \sum_{m=1}^{M} \beta_m b(x;\gamma_m)$

Adaboost for Regression

Define $\displaystyle Err_m = \frac{\sum_{i=0}^{n-1} w_{im} \, \mathbb{I}(y_i \neq f_m(x_i))}{\sum_{i=0}^{n-1} w_{im}}$

$$f_m(x) = f_{m-1}(x) + \beta_m b_m(x)$$

Cost function

$$C(f) = \sum_{i=0}^{n-1} \mathcal{L}_i$$

$$\mathcal{L}_i = e^{-y_i f_m(x_i)}$$

$$y_i = \{-1, 1\}$$

$$f_m(x_i) = \{-1, 1\}$$

algo

a) initialize $w_i = 1/n$

b) for $m = 1 : M$ DO

- Fit a classifier $b_m(x_i)$ to the training data using w_{im}
- compute Err_m
- compute $\beta_m = \frac{1}{2} \ln\left(\frac{1 - \mathrm{Err}_m}{\mathrm{Err}_m}\right)$
- Find new weights w_{im+1}
- update $f_m(x) = f_{m-1}(x) + \beta_m b_m(x)$

end for

return $f(x) = \mathrm{sign}\left\{\sum_{m=0}^{M} \beta_m b_m(x)\right\}$

→ parametrize through ~~mima~~ minimization γ_m, β_m

Gradient boosting

(i) fit f to individual points

$$f : \{ f(x_0), f(x_1) \ldots f(x_{n-1}) \}$$

$$g_{im} = \left[\partial L(y_i, f(x_i)) \right]$$

$$\partial f(x_i) \Big]_{f = f_{m-1}(x_i)}$$

update

$$f_m = f_{m-1} - \underbrace{\rho_m}_{\text{step length to be optimized}} g_m$$

$$\hat{\rho}_m = \arg\min_{\rho} C(f_{m-1} - \rho g_m)$$

steepest descent, less use.

(ii) Gradient boosting:

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

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

$$\boxed{f_m - f_{m-1} = -g_m \beta_m}$$

$$g_m = b(x; \gamma_m) \Rightarrow$$

$$-g_m - b(x; \gamma_m) = 0$$

$$\boxed{\gamma_m = \arg\min_{\gamma} \sum_{i=0}^{m-1} \left(-g_{im} - b(x_i; \gamma)\right)^2}$$

algo

(i) initialize $f_0(x) = \underset{\gamma}{\arg\min} \sum_{i=0}^{n-1} \mathcal{L}(y_i, \gamma)$

(ii) for $m = 1:M$ DO

- compute
$$g_{im} = \left. \frac{\partial \mathcal{L}(y_i, f(x_i))}{\partial f(x_i)} \right|_{f = f_{m-1}(x_i)}$$

- use weak learner to compute γ_m

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

end for

Return $f(x) = \sum_{m=0}^{M} f_m(x)$