Thursday, 10/29/2020

EE 660

MACHINE LEARNING FROM SIGNALS: FOUNDATIONS AND METHODS

Prof. B. Keith Jenkins

Lecture 20

Announcements

- Homework 7 is due Friday
- Homework 8 will be posted
- My office hours for today will be: 3:30 - 4:30 PM

Today's topics

- Boosting
  - Introduction and notation
  - Loss functions
  - Forward stagewise additive modeling: base classifiers and their importance
  - Adaboost
    - Minimization of L
    - Weight of each data point
    - err_m
    - Adaboost.M1 algorithm
    - Example

next lecture

Boosting [Murphy 16.4.0 - 16.4.4] [also Hastie et al., Ch. 10]

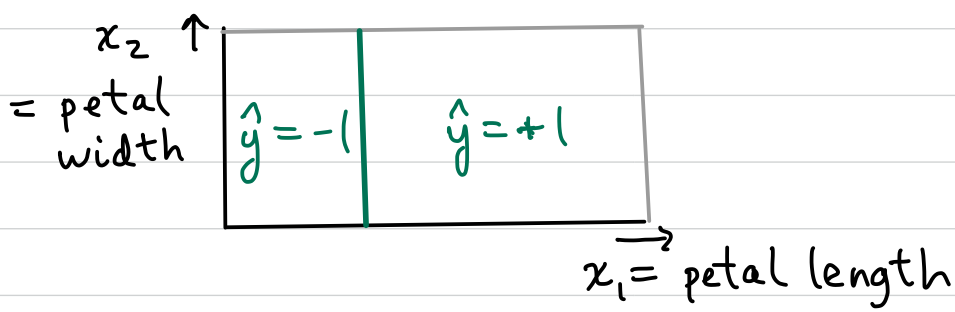
Is also an adaptive basis fcn. model. (ABM)

(1) $\hat{f}(\underline{x}) = w_0 + \sum_{m=1}^{M} w_m \, \phi(\underline{x}; \underline{\gamma}_m)$

$w_m \triangleq \beta_m$; $\underline{\gamma}_m$: parameters of $\phi_m(\underline{x})$, learned from data

Each $\phi(\underline{x}, \underline{\gamma}_m) = \phi_m(\underline{x})$

- is a simple classifier that can classify the entire feature space
- is a "weak learner" that is only required to do better than chance
- is typically a "decision stump"
  - a 1-stage CART resulting in 1 node and 2 leaves (or variants with > 2 leaves)



The simple classifiers $\phi(\underline{x}; \underline{\gamma}_m)$ are found sequentially: $m = 1, 2, \cdots, M$.

Notation

For 2-class problem, let labels be ± 1.

(2) $$\hat{f}(\underline{x}) = f_0 + \sum_{m=1}^{M} \beta_m \, \underbrace{\phi(\underline{x}; \underline{\gamma}_m)}$$

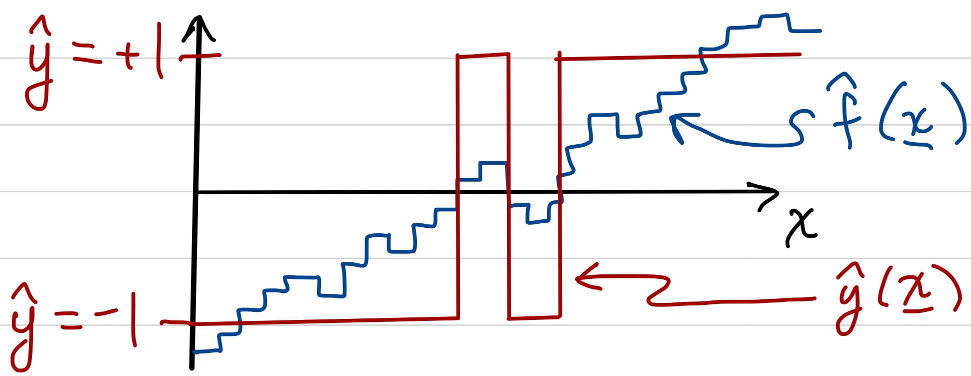
"weight" or importance of $m^{\underline{th}}$ base classifier

$\beta_m \in \mathbb{R}$ (will have $\beta_m \geq 0$)

$\phi(\underline{x}; \underline{\gamma}_m) = \phi_m(\underline{x})$ is the $m^{\underline{th}}$ simple "base" classifier

$\phi_m \in \{-1, +1\}$

Final classifier: $\hat{y}(\underline{x}) = \text{sign}\{\hat{f}(\underline{x})\}$, $\hat{y} \in \{-1, +1\}$



Measure of confidence in output prediction $\hat{y}$?

$\rightarrow |\hat{f}(\underline{x})|$

(3) Intermediate expressions for f ($m^{\underline{th}}$ iteration):

$$\hat{f}_m(\underline{x}) = f_0 + \sum_{m'=1}^{m} \beta_{m'} \, \phi(\underline{x}, \underline{\gamma}_{m'})$$

Loss function

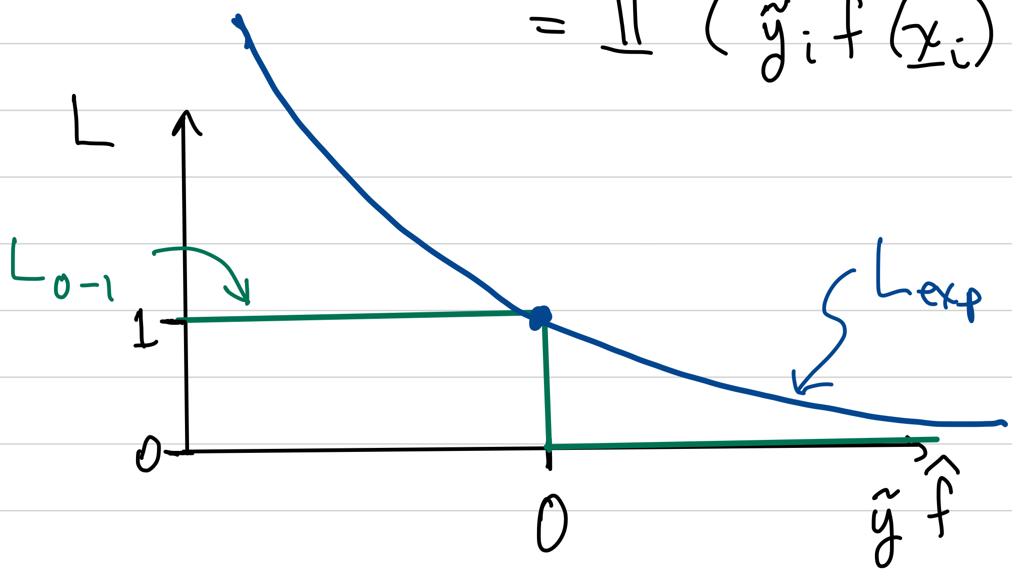
Given $\mathcal{D} = \{ (\underline{x}_i, \tilde{y}_i) \mid \tilde{y}_i \in \{-1, +1\} \}_{i=1}^{N}$

$$f_{obj}\left[\{\beta_m,\ m=1,2,\cdots,M\}, \{\underline{\gamma}_m,\ m=1,2,\cdots,M\} \right] = \frac{1}{N}\sum_{i=1}^{N} L\left(\tilde{y}_i, \hat{f}(\underline{x}_i)\right)$$

Loss function.

0-1 Loss : $L_{0-1} = \mathbb{I}\left(\tilde{y}_i \neq \mathrm{sign}\{\hat{f}(\underline{x}_i)\}\right)$

$= \mathbb{I}\left(\tilde{y}_i \hat{f}(\underline{x}_i) < 0\right)$



(4) Exponential loss : $L_{exp} = \exp\left[-\tilde{y}_i \hat{f}(\underline{x}_i)\right]$

$\exists$ other Loss functions [e.g., Murphy]

Why use L_{exp} instead of L_{0-1}?

- smooth, differentiable
- provides confidence measure
- convex

Want to find: $$\hat{f}(\underline{x}) = \underset{f(\underline{x})}{\operatorname{argmin}} \sum_{i=1}^{N} L_{exp}[\tilde{y}_i, f(\underline{x}_i)]$$

$$= \underset{f_0,\ \beta_m,\ \underline{\gamma}_m,\ \forall m=1,2,\cdots,M}{\operatorname{argmin}} \sum_{i=1}^{N} L_{exp}\left[\tilde{y}_i, f_0 + \sum_{m=1}^{M} \beta_m \phi_m(\underline{x}_i, \underline{\gamma}_m)\right]$$

Easier to min. each term, sequentially:
at m^{th} iteration:

$$\underset{\beta_m, \underline{\gamma}_m}{\operatorname{argmin}} \sum_{i=1}^{N} L_{exp}\left[\tilde{y}_i, \hat{f}_{m-1}(\underline{x}_i) + \beta_m \phi(\underline{x}_i, \underline{\gamma}_m)\right]$$

$$\text{e.g.: } \underline{\gamma}_m = \begin{bmatrix} j' \\ t_{j'} \\ \ell \end{bmatrix} = \begin{bmatrix} \text{Feature to split} \\ \text{threshold value} \\ \text{region labels} \end{bmatrix}$$

Forward Stagewise Additive Modelling

1. Initialize $f_0 = 0$

2. For $m=1$ to M:

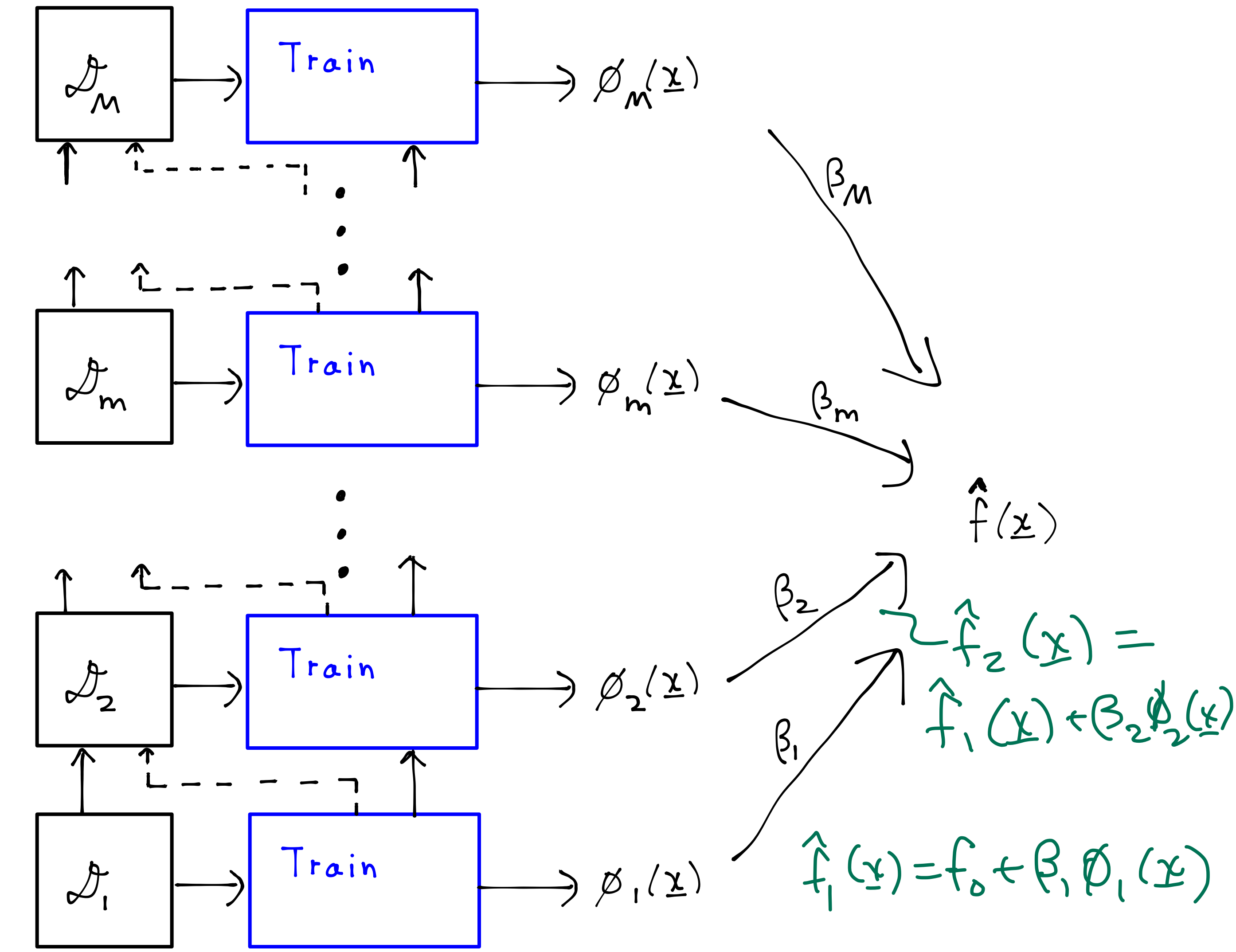
   (i) Find $(\beta_m, \underline{\gamma}_m) = \underset{\beta_m', \underline{\gamma}_m'}{\operatorname{argmin}} \sum_{i=1}^{N} L\left[\tilde{y}_i, \hat{f}_{m-1}(\underline{x}_i) + \beta_m' \phi(\underline{x}_i, \underline{\gamma}_m')\right]$

   (ii) $\hat{f}_m(\underline{x}) = \hat{f}_{m-1}(\underline{x}) + \beta_m \phi(\underline{x}, \underline{\gamma}_m)$

3. Final classifier is: $\hat{y}(\underline{x}) = \operatorname{sign}\{\hat{f}_M(\underline{x})\}$

(For L could use L_{exp} or other loss fcn.)

How to do 2 (i) is left unspecified.



$\mathcal{D}_1 = \mathcal{D}_{Tr}$

$$\hat{y}(\underline{x}) = \operatorname{sign}\{\hat{f}(\underline{x})\} = \operatorname{sign}\left\{ \sum_{m=1}^{M} \beta_m \phi_m(\underline{x}) \right\}$$

Adaboost

→ Use $L = L_{exp}$

From above 2(i):

$$\text{Find } (\beta_m, \underline{\gamma}_m) = \underset{\beta_m', \underline{\gamma}_m'}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{N} L_{exp}\left[\tilde{y}_i, \hat{f}_{m-1}(\underline{x}) + \beta_m' \phi(\underline{x}_i, \underline{\gamma}_m')\right]}_{L_m}$$

(5) $$L_m = \sum_{i=1}^{N} \exp\left\{ \underbrace{-\tilde{y}_i \left[\hat{f}_{m-1}(\underline{x}_i)\right.}_{w_{i,m} \;=\, \text{const. of } \beta_m', \underline{\gamma}_m'} \left. + \beta_m' \phi(\underline{x}_i, \underline{\gamma}_m')\right] \right\}$$

(6) Let $w_{i,m} \triangleq \exp\left\{-\tilde{y}_i \hat{f}_{m-1}(\underline{x}_i)\right\}$

data pt. index

iteration (base classifier) index

$$L_m = \sum_{i=1}^{N} \underbrace{w_{i,m}}_{\substack{\text{weight on each}\\ \text{data pt. } i \text{ at } m^{th}\\ \text{iteration.}}} \underbrace{\exp\left[-\tilde{y}_i \beta_m' \phi(\underline{x}_i, \underline{\gamma}_m')\right]}_{\substack{\text{interpret as } L_{exp} \text{ for data pt. } i \text{ using base}\\ \text{classifier } \phi_m.}}, \qquad \beta_m' > 0.$$

L_m can be minimized algebraically. [See Murphy].

Can re-arrange L_m above to get eqns. for Adaboost algorithm [Murphy]:

(*) $\phi(\underline{x}_i, \underline{\gamma}_m)$ is chosen to minimize:

$$\phi_m = \underset{\phi}{\operatorname{argmin}} \left\{ \underbrace{\sum_{i=1}^{N} w_{i,m} \mathbb{I}\left[\tilde{y}_i \neq \phi(\underline{x}_i)\right]}_{\text{sum of weights of misclassified data pts.}} \right\}$$

(decision stump optimization)

☆ Your Murphy may have this omitted (error) in Eq. (16.40).