Gradient boosting:

- I wish to construct a function approximating some data (x_i, y_i) in some sense.

- So must minimize

$$\int \text{LOSS}(f(x), y(x))\, P(x)\, dx$$

(which I can't evaluate — must substitute a sum & eventually.)

But look at the problem its a variational problem.

I have

$$\min_{f} \mathcal{J}(f(x))$$

Attack by

$$f^{(i)}(x) = \sum_{j=1}^{i} \alpha_j \phi_j(x, \theta_j)$$

these are parameters

where these are model functions of some sort - eg trees, etc

Assume I have

$f^{(i)}$ - how to get next?

Take functional gradient of $\mathcal{J}\big|_{f^{(i)}}$

- this is a derivative, wrt f
- linear function of a small change δf.

$(\nabla_f \mathcal{F})(\phi)$ is best approx to $F(f+\phi)$

- $(\nabla_f \mathcal{F})(\phi)$ is linear, so it must have the form

$$\int H \cdot \phi \, dx$$

- Notice that if F depends on f, but NOT derivatives of f.

$$F(f+\phi) = \int \text{loss}(f+\phi, y) P(x) \, dx$$

$$\approx \int \text{loss}(f, y) P(x) \, dx + \int \frac{\partial \text{loss}}{\partial u_1}(f, y) \cdot \phi \cdot P(x) \, dx$$

This means that the functional gradient is

$$\langle \delta \mathcal{F}, \phi \rangle = \int \frac{\partial \text{loss}}{\partial u_1}(f, y) \cdot \phi \; P(x)\, dx$$

↑ first argument

(for the case where loss depends only on f not f_x etc)

Now we can do Functional gradient descent.

Notice similarity/analogy with gradient

Gradient	
directional derivative in dir a	dir deriv in dir ϕ
$= \nabla f \cdot a$	$= \int \frac{\partial l}{\partial u_1} P(x)\, \phi \, dx$

Gradient Descent:

- assume we have $f^{(n)}$
- functional gradient in dir φ is

$$\int \left.\frac{\partial L}{\partial u_1}\right|_{u_1 = f^{(n)}} \cdot \varphi \cdot P(x)\, dx.$$

- we cannot represent this accurately, its at least as complex as a function
- Option choose an update from a finite dimensional space of fn's

$$\Delta f^{(n)} \in S$$

↑ this could be trees, etc, etc.

Now, find the element of S that best approximates the gradient

Want to minimize

$$\int \left\| \left. \frac{\partial L}{\partial u_1} \right|_{u_1 = f^{(u)}} - \Delta f^{(u)} \right\|^2 P(x)\, dx$$

By choice of function Δf in S

Notice that, if S consists of functions with bounded norm (some trees, etc).

this is equivalent to

$$\max \int \left. \frac{\partial L}{\partial u_1} \right|_{u_1 = f^{(u)}} \cdot \Delta f^{(u)} \cdot P(x)\, dx$$

analogy: $\left. \frac{\partial L}{\partial u_1} \right|_{u_1 = f^{(u)}} \equiv \nabla f$

$\int (\alpha, \beta)\, P(x)\, dx \equiv$ dot product

So we are looking for the Δf that has largest dot product w/ "gradient"

- for many types of function model, this is fairly easy.

e.g. S = Decision stumps

S = Decision trees

- Now once we have $\Delta f^{(n)}$, we can choose α_n by linesearch

$$f^{(n+1)} = f^{(n)} - \alpha_n \Delta f^{(n)}$$

~~linesearch~~

~~$g(\alpha_n) = \int loss(f^{(n)}, y)$~~

Linesearch:

$$g(\alpha_n) = \mathcal{F}(f^{(n)} \bar{+} \alpha_n \Delta f^{(n)})$$

$$= \int loss(f^{(n)} \bar{+} \alpha_n \Delta f^{(n)}, y(x)) \, P(x) \, dx$$

$$\approx \sum_{i \in \text{examples}} loss(f^{(n)}_{(x_i)} \bar{+} \alpha_n \Delta f^{(n)}_{(x_i)}, y_i)$$

Notice that we can iterate ad nauseam

~~- particularly if the~~

- the cost is applied to an α-dim obj.
- we are taking fd steps.

Example

- Classification with stumps.
- $f(x) = \sum_{i \in \text{stumps}} \alpha_i \{\text{value of leaf at } x\}$

where ~~stumps~~ leaves have values $-1, 1$.

- $\text{loss} = \max(0, 1 - y_{\text{known}} \cdot f_{\text{pred}})$

Procedure:

- have $f^{(n)}$
- $\left.\frac{\partial l}{\partial u_i}\right|_{f^{(n)}} = \begin{cases} 0 & \text{if } y_k \cdot f_{\text{pred}} > 1 \\ -y_{\text{known}} & \text{otherwise} \end{cases}$

$\Rightarrow$ new stump tries to have items too close to margin on either side dep. sgn y_{known}

and ignores ~~correct~~ answers to the far side of margin.

NUMEROUS other examples in Friedmen paper, q.v.