How to Boost *Any* Loss Function

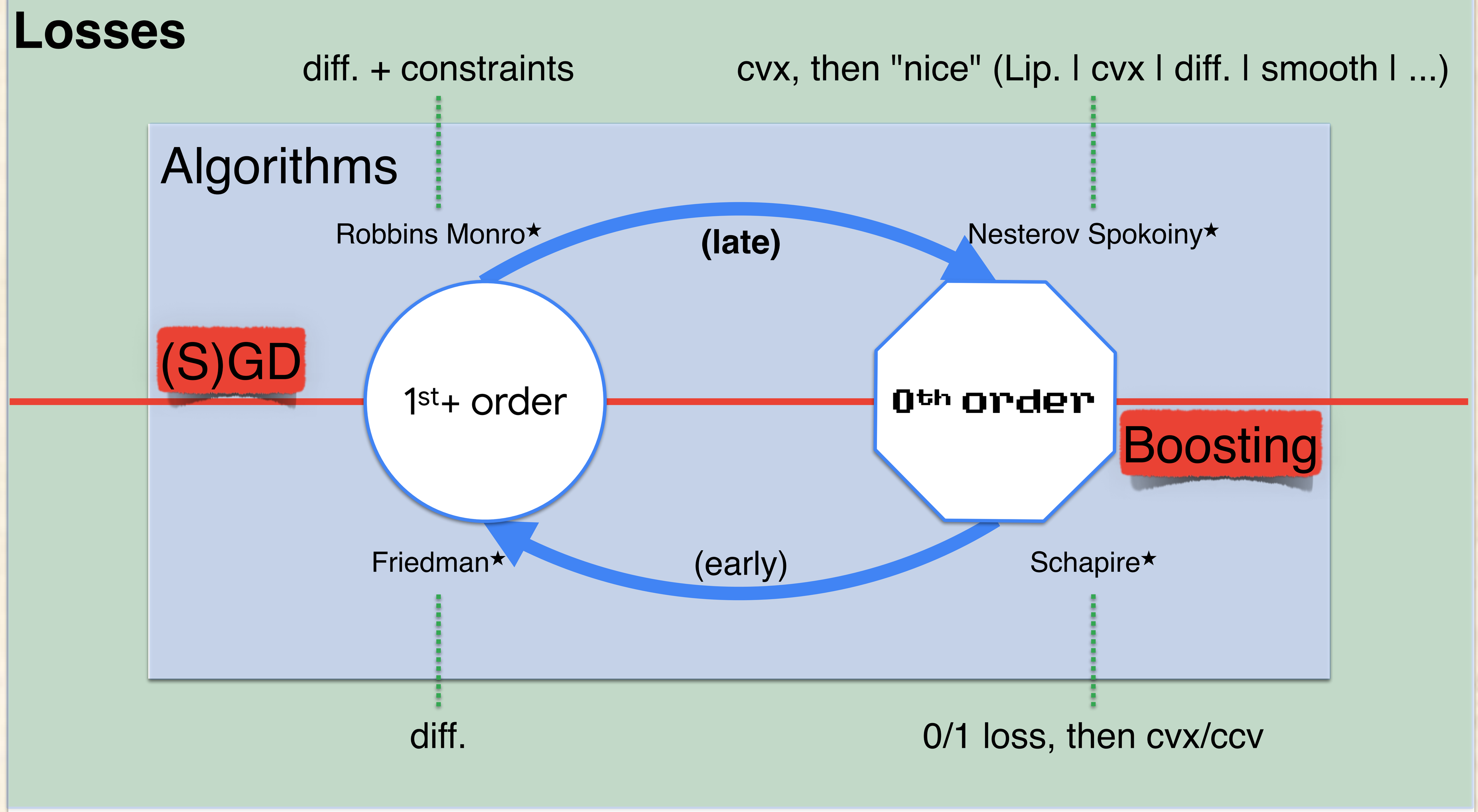
Richard Nock Yishay Mansour

Google Research

TEL AVIV UNIVERSITY

Summary

- Recent evolution of **(S)GD** → *0^{th} order opt.*: gradient-free, only loss queries (not any loss: they need to be somehow nice)



*=and then *many* others

- **Boosting** is gradient-free by design (Kearns/Valiant) w/ "nasty" 0/1 loss and then evolved → *1^{st} order opt.* w/ differentiable losses
  - We question the power of the original 0^{th} order framework*: *what losses can it directly optimize under the weak learning assumption* ?
  - Answer: any loss whose set of discontinuities has 0 Lebesgue measure - computer-wise, this means **any loss** + our technique is constructive: *we give an algorithm*

*=analysis of boosting-compliant convergence on training, since generalization entails restrictions on losses w/ SOTA toolbox (no different from (S)GD → 0^{th} order's mainstream analysis)

⚠
- (S)GD → 0^{th} order "natively" operates on (m)any architectures
- Boosting implies finding the architecture (how "blocks" from weak learner are assembled), so (still) restricted from this standpoint

Algorithm☆

☆simplified, see paper for full presentation

Algorithm 1 SecBoost($\mathcal{S}, T$)

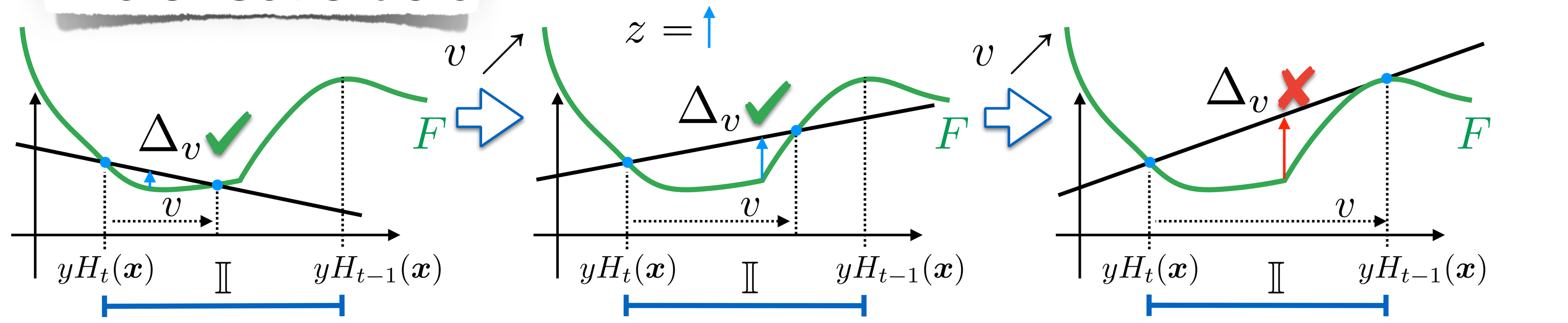
Input sample $\mathcal{S} = \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$, number of iterations T, initial (h_0, v_0) (constant classification and offset).
Step 1 : let $H_0 \leftarrow 1 \cdot h_0$ and $\boldsymbol{w}_1 = -\delta_{v_0} F(h_0) \cdot \boldsymbol{1}$;
Step 2 : **for** $t \in [T]$
  Step 2.1 : let $h_t \leftarrow$ Weak_Learner($\mathcal{S}_t, |\boldsymbol{w}_t|$);
  Step 2.2 : compute leveraging coefficient α_t, params $\varepsilon_t > 0, \overline{w}_{2,t} > 0$;
  Step 2.3 : let $H_t \leftarrow H_{t-1} + \alpha_t \cdot h_t$;
  Step 2.4 : **for** $i \in [m]$, let $v_{ti} \leftarrow$ Offset_Oracle($t, i, \varepsilon_t \cdot \alpha_t^2 M_t^2 \overline{w}_{2,t}$);
  Step 2.5 : **for** $i \in [m]$, let $w_{(t+1)i} \leftarrow -\delta_{v_{ti}} F(y_i H_t(\boldsymbol{x}_i))$;
  Step 2.6 : **if** $\boldsymbol{w}_{t+1} = \boldsymbol{0}$ **then** break;
Return H_T.

$M_t \doteq \max_i |h_t(\boldsymbol{x}_i)|$

Notable generalizations with respect to "boosting-à-la-Valiant"

- Examples for Weak_Learner can be *label flipped*: $\mathcal{S}_t \doteq \{(\boldsymbol{x}_i, y_i \cdot \mathrm{sign}(w_{ti}))\}$
- Need an "oracle" giving *offsets* (implementation generic or loss dependent)

The offset oracle



- For the ease of exposure, $yH_t(\boldsymbol{x}) < yH_{t-1}(\boldsymbol{x})$ (see paper for general case)
  → Let $\mathbb{I} \doteq [yH_t(\boldsymbol{x}), yH_{t-1}(\boldsymbol{x})]$
  → Let Δ_v be the secant through $(yH_t(\boldsymbol{x}), F(yH_t(\boldsymbol{x})) \,\&\, (yH_t(\boldsymbol{x}) + v, F(yH_t(\boldsymbol{x}) + v)$, for $v > 0$. Compute the maximum difference $\delta_v \doteq \max_{\mathbb{I}} \Delta_v - F$ (it is ≥ 0)
  → Offset_Oracle($., ., z$) returns any $v \neq 0$ such that $\delta_v \leqslant z$

Step 2.2

- Two possibilities to get $\alpha_t, \varepsilon_t, \overline{w}_{2,t}$, where $\overline{w}_{2,t}$ is any > 0 real s.t.

$$\mathbb{E}_{i\sim[m]}\left[\delta_{\{\alpha_t y_i h_t(\boldsymbol{x}_i), v_{(t-1)i}\}} F(y_i H_{t-1}(\boldsymbol{x}_i)) \cdot \left(\frac{h_t(\boldsymbol{x}_i)}{M_t}\right)^2\right] \leqslant \overline{w}_{2,t}$$

tricky bit !

If offsets were → 0, this would be a second-order derivative

⇨ **Possibility 1**: F is "nice" ⇒ easy bound: we just have to pick $\varepsilon_t > 0, \pi_t \in (0,1)$ and α_t as:
(example: F β-smooth ⇒ $\overline{w}_{2,t} = 2\beta$)

$$\alpha_t \in \frac{\eta_t}{2(1+\varepsilon_t) M_t^2 \overline{w}_{2,t}} \cdot [1 - \pi_t, 1 + \pi_t]$$

$\eta_t \doteq (1/m) \cdot \sum_i w_{ti} y_i h_t(\boldsymbol{x}_i)$ important for boosting rate

⇨ **Possibility 2**: no niceness ⇒ *Cf* paper for efficient algorithm providing all params ($\alpha_t, \varepsilon_t, \overline{w}_{2,t} \,\&\, \pi_t$)

Toolbox

- Generalization of quantum calculus' (≠ quantum computation) v-derivative:

$$\delta_{\mathcal{V}} F(z) \doteq \begin{cases} F(z) & \text{if} \quad \mathcal{V} = \varnothing \\ \delta_v F(z) & \text{if} \quad \mathcal{V} = \{v\} \\ \delta_{\{v\}}(\delta_{\mathcal{V}\backslash\{v\}} F)(z) & \text{otherwise} \quad \mathcal{V} = \{v, w, ...\} \text{ (eventually multiset)} \end{cases}$$

- Singleton $\mathcal{V} = \{v\}$ ⇒ classical secant's slope

$$\delta_v F(z) \doteq \frac{F(z+v) - F(z)}{v}$$

v ← offset

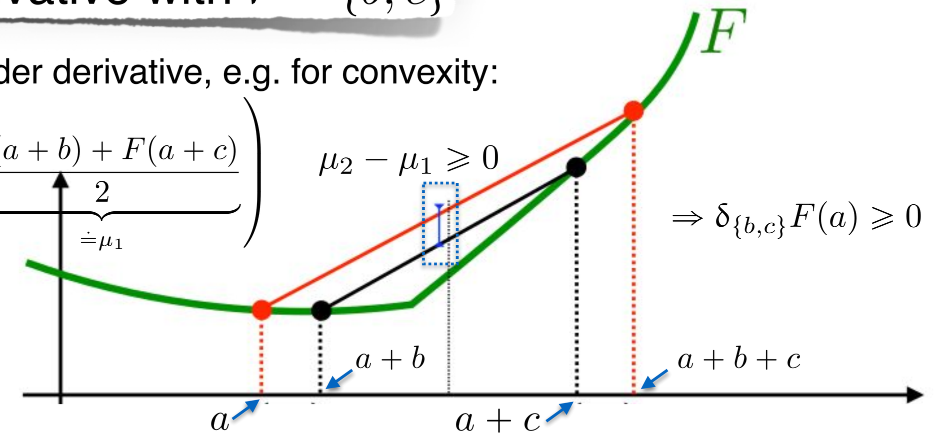
called h-derivative, with $\mathcal{V} = \{v, v, ...\}$ in "Quantum calculus", Kac & Cheung, 2002

Example: second-order v-derivative with $\mathcal{V} = \{b, c\}$

generalizes some properties of second-order derivative, e.g. for convexity:

$$\delta_{\{b,c\}} F(a) = \frac{2}{b} \cdot \frac{1}{c} \cdot \left(\underbrace{\frac{F(a+b+c)+F(a)}{2}}_{\doteq \mu_2} - \underbrace{\frac{F(a+b)+F(a+c)}{2}}_{\doteq \mu_1}\right)$$

(wlog c > b > 0, see Lemma 5.2 in paper for more & general case(s))

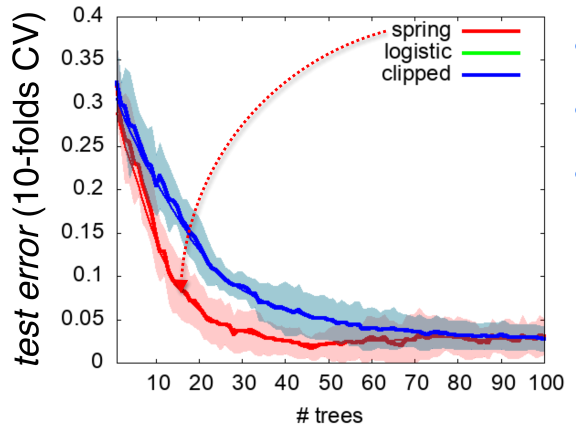


Boosting !

- Weak Learning Assumption: $\left|\mathbb{E}_{\tilde{\boldsymbol{w}}_t}\left[\tilde{y}_{ti} \cdot \frac{h_t(\boldsymbol{x}_i)}{M_t}\right]\right| \geqslant \gamma > 0$
  ⇨ $\tilde{y}_{ti} \doteq y_i \cdot \mathrm{sign}(w_{ti})$ label eventually *flipped*
  ⇨ $\tilde{\boldsymbol{w}}_t = |\boldsymbol{w}_t|$ *normalized* to unit
- Weak Convergence Regime: $\frac{|\mathbb{E}_{i\sim[m]}[w_{ti}]|^2}{\overline{w}_{2,t}} \geqslant \rho > 0$
  ⇨ numerator ← 1^{st} order *v*-derivative, expected *signed* weights
  ⇨ denominator ← 2^{nd}-order *v*-derivative, loss "jiggling"
- Theorem. Let the expected empirical loss of classifier H be $F(\mathcal{S}, H) \doteq \mathbb{E}_{i\sim[m]}[F(y_i H(\boldsymbol{x}_i))]$ and its initial value $F_0 \doteq F(\mathcal{S}, h_0)$. Suppose WLA+WCR hold. Then for any $z \in \mathbb{R}$ such that $F(z) \leqslant F_0$, if SecBoost is run for a number of iterations

$$T \geqslant \frac{4(F_0 - F(z))}{\gamma^2 \rho} \cdot \frac{1 + \max_t \varepsilon_t}{1 - \max_t \pi_t^2}$$

then $F(\mathcal{S}, H_T) \leqslant F(z)$.



- Example implementation (details: Cf paper)
- Weak Classifiers = size-20 DTs
- Losses: logistic & two variations: clipped logistic and a non-[cvx,Lip,diff] loss ("spring loss")