# Random Classification Noise does not defeat All Convex Potential Boosters Irrespective of Model Choice

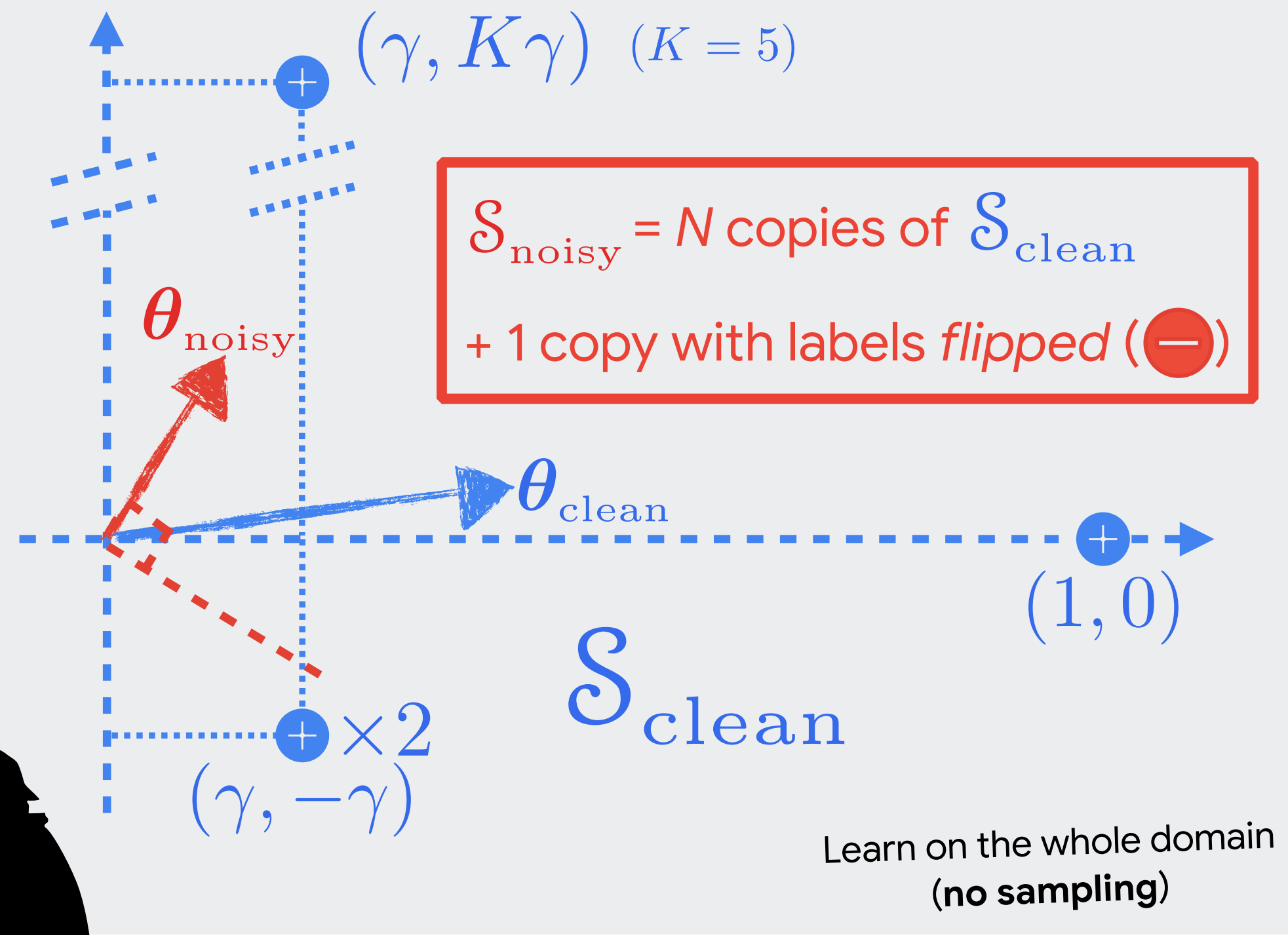
Yishay Mansour
Tel Aviv U. & Google Research

Richard Nock
Google Research

Robert C. Williamson
U. of Tübingen & Tübingen AI center

## Why this work

- "Random Classification Noise Defeats All Convex Potential Boosters" -- Long and Servedio (ICML'08 & MLJ'10)
- 1-class trivial data, noisify it with symmetric label noise
- Compute the respective minimizers of any* **convex loss** or of any* convex **boosting** algorithm ($\boldsymbol{\theta}_{\text{clean}}, \boldsymbol{\theta}_{\text{noisy}}$)
- On $\mathcal{S}_{\text{clean}}$, $\boldsymbol{\theta}_{\text{clean}}$ is (predictably) perfect **but** $\boldsymbol{\theta}_{\text{noisy}}$ =
- Sizeable impact on design of boosting algorithms,
- Recurrent stress of key culprit: <u>convex boosting</u>. **Or is it ?**



## The Setting: *properness*

Shufford *et al.*, 1966; Savage, 1971

- Loss for Class Probability Estimation (CPE)

$$\ell(\underset{\in[0,1]}{y^*, \overset{\text{prediction / guess}}{u}}) \doteq \left\{\begin{array}{c} [\![y^* = 1]\!] \cdot \ell_1(u) \\ + \\ [\![\overset{\text{label / class}}{y^*} = -1]\!] \cdot \ell_{-1}(u) \end{array}\right.$$

- Functions $\ell_1, \ell_{-1}$ = partial losses
- Conditional risk of $u$ wrt ground truth $v \in [0,1]$:

$$L(u, v) \doteq v \cdot \ell_1(u) + (1 - v) \cdot \ell_{-1}(u)$$

- Bayes risk: $\underline{L}(v) \doteq \inf_u L(u, v)$

- A CPE loss is
  - *symmetric* iff $\ell_1(u) = \ell_{-1}(1-u), \forall u \in [0,1]$

    No class-dependent misclassification cost
  - differentiable iff $\ell_1, \ell_{-1}$ differentiable
  - lowerbounded iff $\ell_1, \ell_{-1}$ lowerbounded
  - <u>***proper***</u> iff:

    $$\forall v \in [0,1], L(v, v) = \inf_u L(u, v)$$

    Encourages eliciting ground truth
  - ***strictly proper*** iff $v$ = sole minimizer

$\forall$ proper loss, $\underline{L}(v) = v \cdot \ell_1(v) + (1 - v) \cdot \ell_{-1}(v)$

Ex. of **S**trictly **P**roper **D**ifferentiable - SPD - (and *symmetric*) losses: square, log / crossentropy, Matusita

ML: given sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y^*_i)\}_{i=1}^m$, learn *posterior* $\tilde{\eta} : \mathcal{X} \rightarrow [0,1]$ min. $\Phi(\tilde{\eta}, \mathcal{S}) \doteq \mathbb{E}_{i \sim [m]}\left[\ell(y^*_i, \tilde{\eta}(\boldsymbol{x}_i))\right]$

## A "proper paradox"

- In Long and Servedio's setting, does not learn a posterior directly but real-valued *classifier* $h : \mathcal{X} \rightarrow \mathbb{R}$

Theorem (Nock & Menon, 2020): any SPD loss admits a dual form for real-valued classification,

$$\ell(y, h(\boldsymbol{x})) = -\underline{L}(\underset{[\![y^*=1]\!]}{y}) + \phi_\ell(-h(\boldsymbol{x})) - yh(\boldsymbol{x})$$

$$\phi_\ell(z) \doteq \overset{\text{convex conjugate}}{(-\underline{L})^\star}(-z)$$

& if +*symmetric*,

$$\ell(y^*, h(\boldsymbol{x})) = -\underline{L}\left(\frac{1+y^*}{2}\right) + \overset{\text{"margin form"}}{\phi_\ell(y^* h(\boldsymbol{x}))}$$

- **Lemma**: for any SPD+symmetric loss, $\phi_\ell$ meets the blueprint to Long and Servedio's convex losses

On Long and Servedio's data, [boosted] optimum ends up eliciting **not** Bayes rule, but a fair coin !

... and there is "worse":

**Theorem**: Long and Servedio's results survive to dropping the *symmetry* constraint on SPD loss...

⇢ class-dependent partial losses cannot solve problem

⇢ negative result holds for losses "without margin form" as well

Properness virtually *useless* to learn, while it should lead to maximal accuracy on noise-free data...

## The culprit & how to address the negative result

- Long and Servedio's setting relates to [**boosted**] optimum of a (margin) **convex** loss... but supervised ML involves training **a model** in the pipeline (Long and Servedio: linear)... so what if we *just* **replace the model**?
- +make it more general: any* **S**trictly **P**roper **D**ifferentiable loss (not necessarily *symmetric*, no margin form)

```
Algorithm 1 ModaBoost($\mathcal{S}, \ell$, WL, AEO, $T$)
  Input: Dataset $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, SPD loss $\ell$, weak learner WL, architecture
  emulation oracle AEO, iteration number $T \geqslant 1$;
  Output: PLM $H_T$;
  Step 1 : $\forall i \in [m], w_{i,1} \doteq w((\boldsymbol{x}_i, y_i), H_0)$          initial guess (constant, e.g. 0)
  Step 2 : for $t = 1, 2, ..., T$
      Step 2.1 : $\mathcal{X}_t \leftarrow$ AEO$(\mathcal{X}, H_{t-1})$;
      Step 2.2 : $h_t \leftarrow$ WL$(\boldsymbol{w}^*_t, \mathcal{S} \cap \mathcal{X}_t)$;   subset of weights in $\mathcal{S} \cap \mathcal{X}_t$
      Step 2.3 : compute $\alpha_t$ as the solution to:
                                                                  in here
          $\sum_{i \in [m]_t} w((\boldsymbol{x}_i, y_i), H_t) \cdot y^*_i h_t(\boldsymbol{x}_i) = 0$;
                    Indexes of $\mathcal{S}$ in $\mathcal{X}_t$
      Step 2.4 : $\forall i \in [m]_t, w_{t+1,i} \doteq w((\boldsymbol{x}_i, y_i), H_t)$
  return $H_T(\boldsymbol{x}) \doteq \sum_{t=1}^T [\![\boldsymbol{x} \in \mathcal{X}_t]\!] \cdot \alpha_t h_t(\boldsymbol{x})$;
```

- PLM = Partition Linear Model:

$$H_t(\boldsymbol{x}) \doteq \sum_{t=1}^T [\![\boldsymbol{x} \in \mathcal{X}_t]\!] \cdot \alpha_t h_t(\boldsymbol{x})$$

  - $h_t$ returned by classical Weak Learner
  - $\mathcal{X}_t$ by <u>*Architecture Emulation Oracle*</u>
- AEO [+WL] chosen so that $H_T$ *emulates* a specific model architecture (if AEO returns $\mathcal{X}$, learns linear model)
- Weight function uses both class notations:

$$w((\boldsymbol{x}, y), H) \doteq y - y^* \cdot (-\underline{L}')^{-1}(H(\boldsymbol{x}))$$

Weak learning assumption (WLA): any $h_t$ satisfies

$$\left|\sum_{i \in [m]_t} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y^*_i \cdot \frac{h_t(\boldsymbol{x}_i)}{\max_{j \in [m]_t} |h_t(\boldsymbol{x}_j)|}\right| \geqslant \gamma_{\text{WL}}$$

for some $\gamma_{\text{WL}} > 0$

AEO Compliance (AEOC): any $\mathcal{X}_t$ satisfies

$$J([m]_t, t) \geqslant u_t \cdot J([m], t)$$

for some $u_t > 0$

$\underline{J(\mathcal{W}, t) \doteq \text{Card}(\mathcal{W}) \cdot (\mathbb{E}_{i \sim \mathcal{W}}[w_{t,i}])^2}$

"sufficiently many significant weights chosen"

**Theorem** : Suppose WLA + AEOC hold & the SPD loss has $\ell_{-1}(0) \geqslant C, \ell_1(1) \geqslant C, \inf\{\ell'_{-1} - \ell'_1\} \geqslant \kappa$ . $C \in \mathbb{R}, \kappa > 0$
Then for any $\theta \geqslant 0, \varepsilon > 0$, the following holds for ModaBoost's output $H_T$ :

$$\left(T \geqslant U^{-1}\left(\frac{2\left(\Phi(H_0, \mathcal{S}) - C\right)}{\kappa \cdot \varepsilon^2 \underline{w}(\theta)^2 \gamma^2_{\text{WL}}}\right)\right) \Rightarrow \left(\mathbb{P}_{i \sim [m]}[y^*_i H_T(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon\right)$$

large margin guarantees

$$U(T) \doteq \sum_{t=1}^T u_t \qquad \underline{w}(\theta) \doteq \min\{1 - (-\underline{L}')^{-1}(\theta), (-\underline{L}')^{-1}(-\theta)\}$$

**ML models "boostable"** by emulation from ModaBoost (see the paper for translated boosting rates) include: Linear separators, Decision trees, Alternating decision trees, Nearest neighbors, Labeled branching programs

- ModaBoost stands an independent contribution to our work, so what does it bring in the context of our paper ?

**Lemmata**: for ***any*** values of the triple $(N, K, \gamma) \in \mathbb{N}^2_{>0} \times \mathbb{R}_{>0}$ in Long and Servedio's data, **ModaBoost, trained on $\mathcal{S}_{\text{noisy}}$ , is Bayes optimal in 1 iteration <u>*if*</u>** it emulates any of [alternating] decision trees, nearest neighbors or labeled branching programs... but it can so blatantly fail with linear separators that it can hit fair coin prediction on $\mathcal{S}_{\text{clean}}$ in just 2 iterations (depending on loss)

**Conclusion: mind the parameterization !**

## Toy Experiment

- ModaBoost + linear separators on Long and Servedio's data, loss = Matusita