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

Yishay Mansour

Tel Aviv U.
& Google Research

Richard Nock

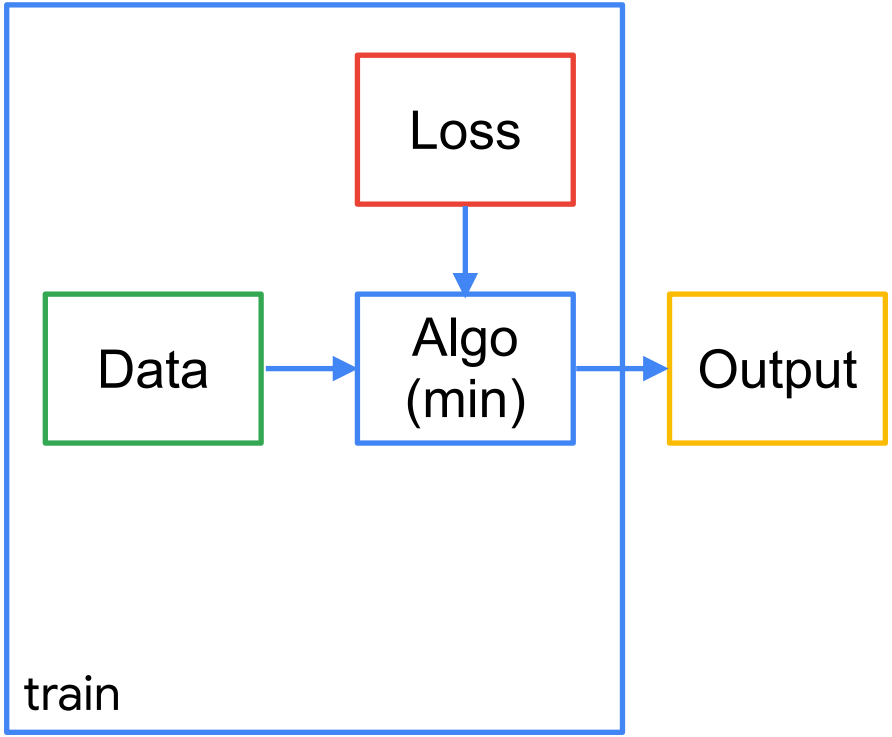
Google Research

Robert C. Williamson

Tübingen U.
& Tübingen AI Center

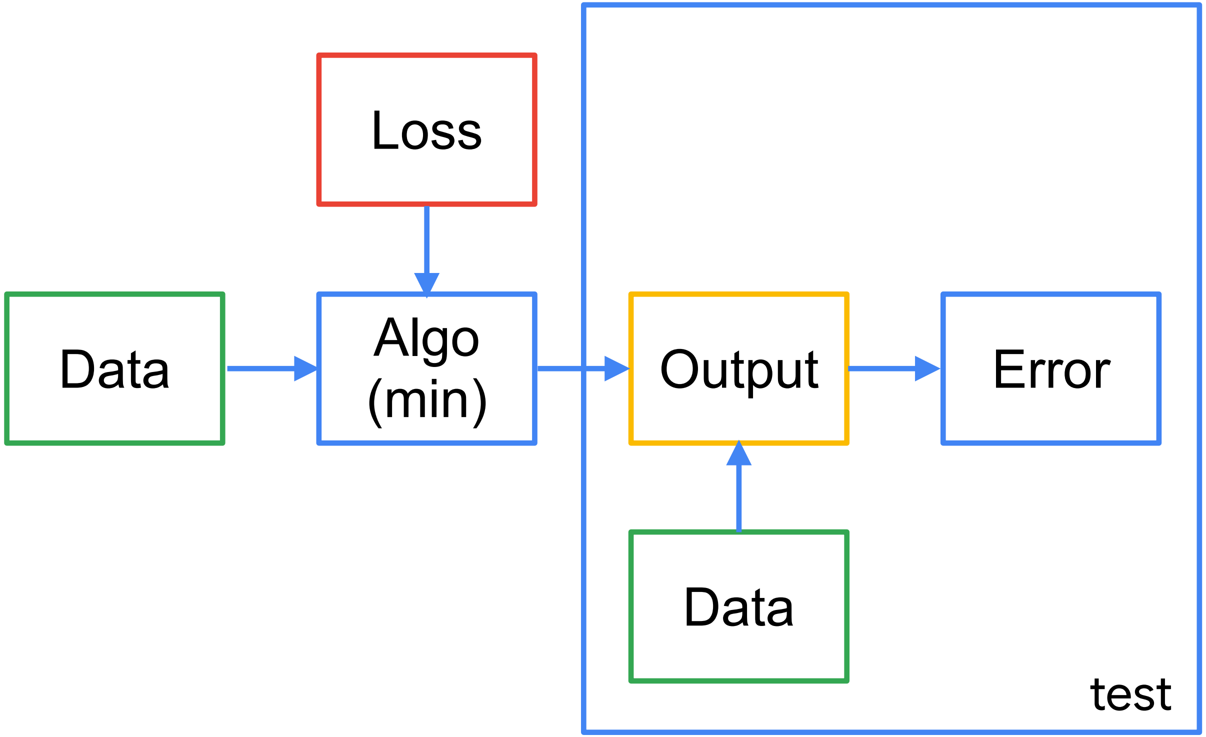
Why this work ?

Long & Servedio (L&S) - Setting II

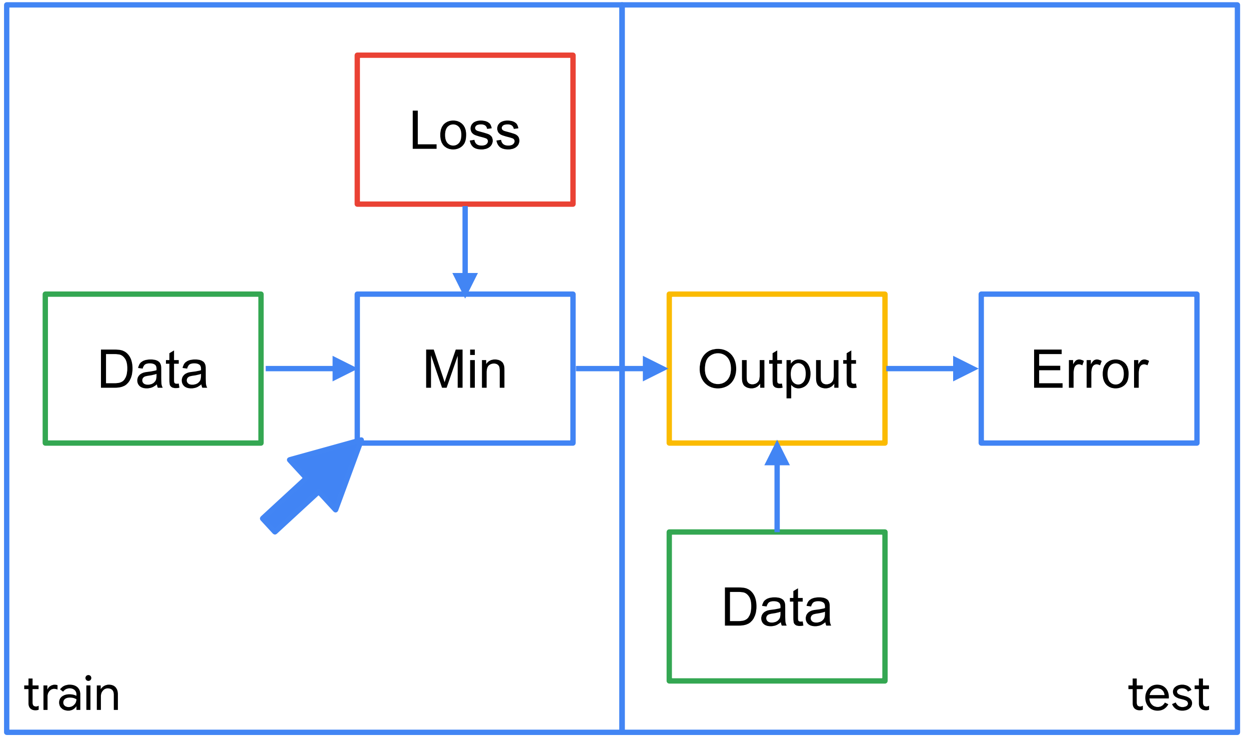


Long, P.-M. and Servedio, R.-A. Random classification noise defeats all convex potential boosters. In 25^{th} *ICML*, pp. 608–615, 2008b.

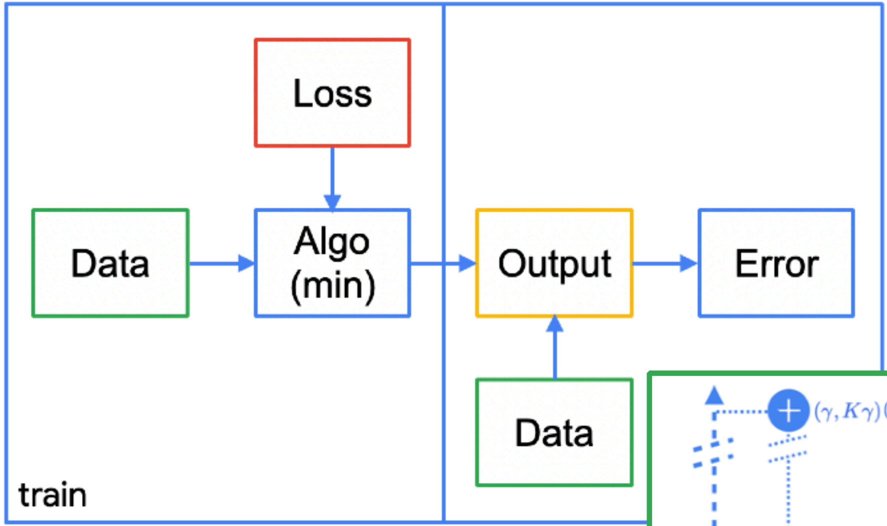
Setting II

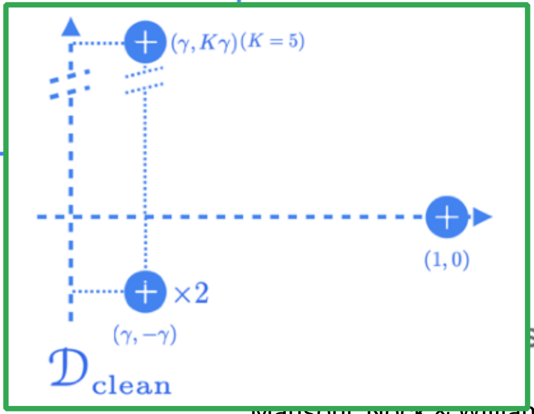


Setting I

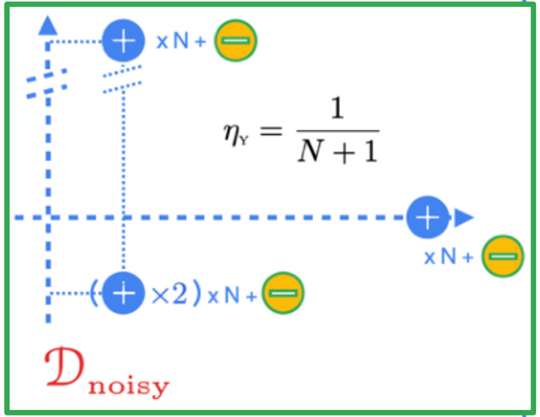


Setting II - test data

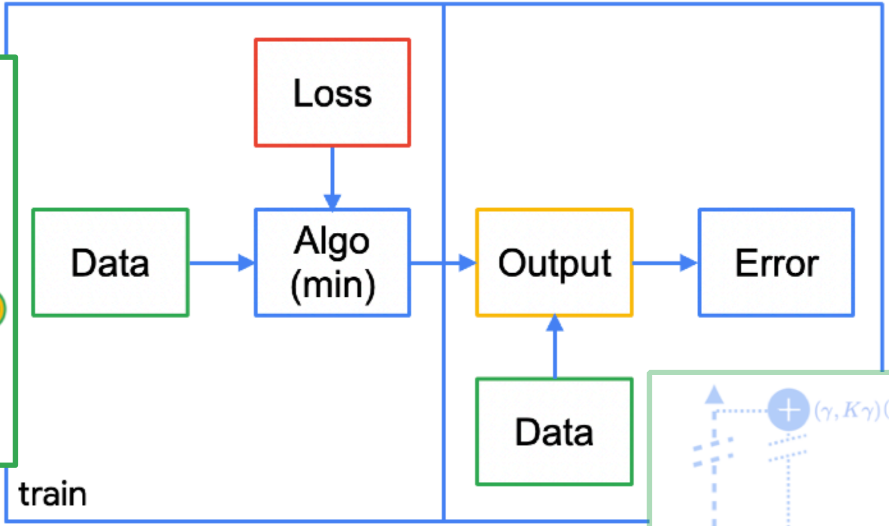


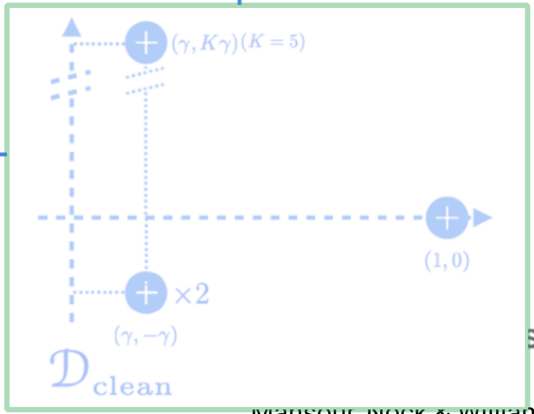


Setting II - training data



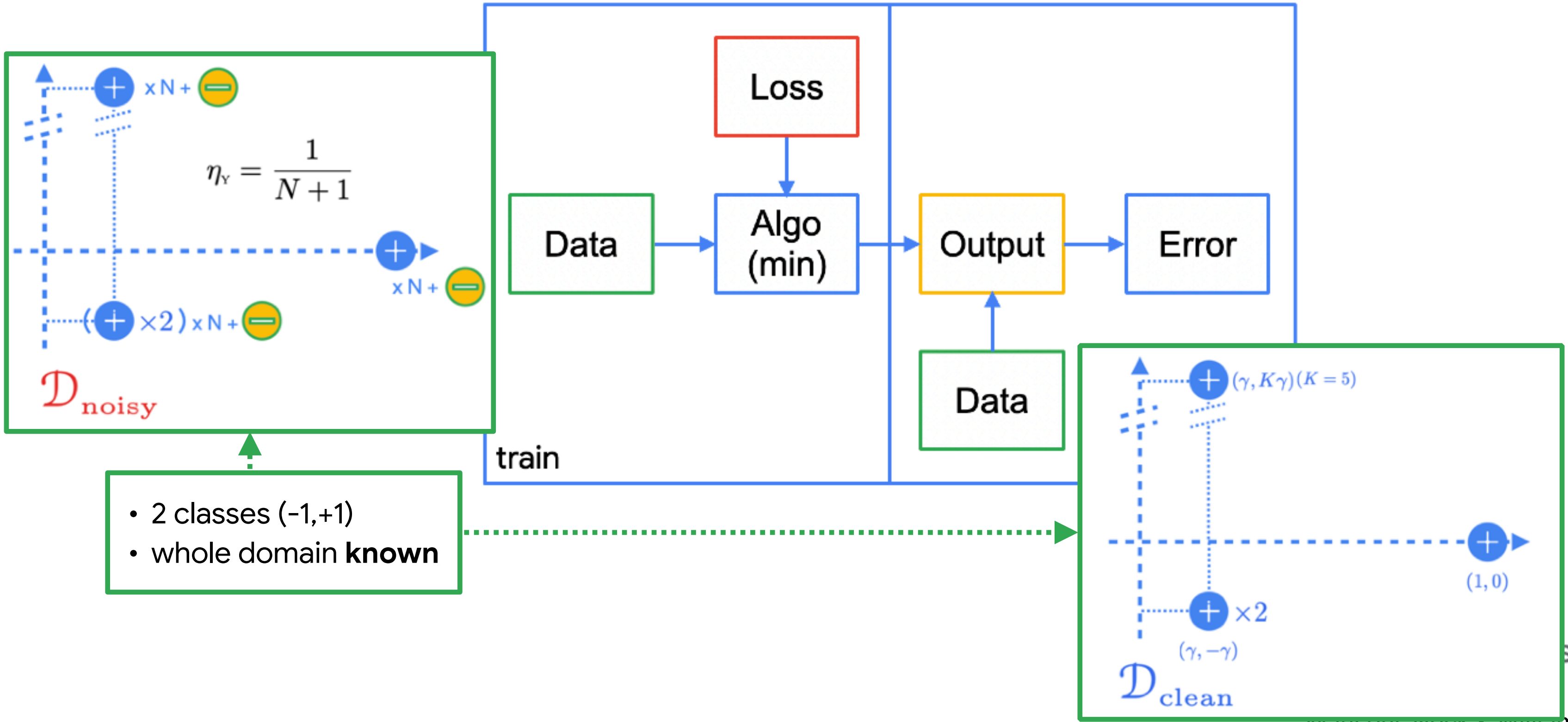
$(N > 1)$



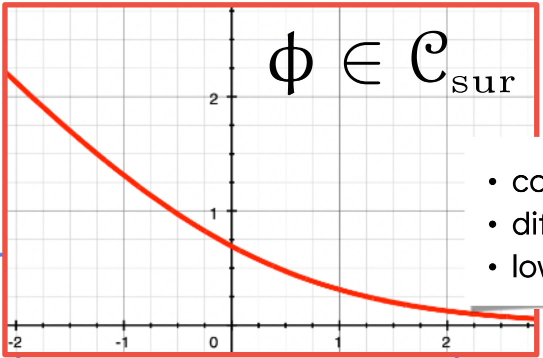


search

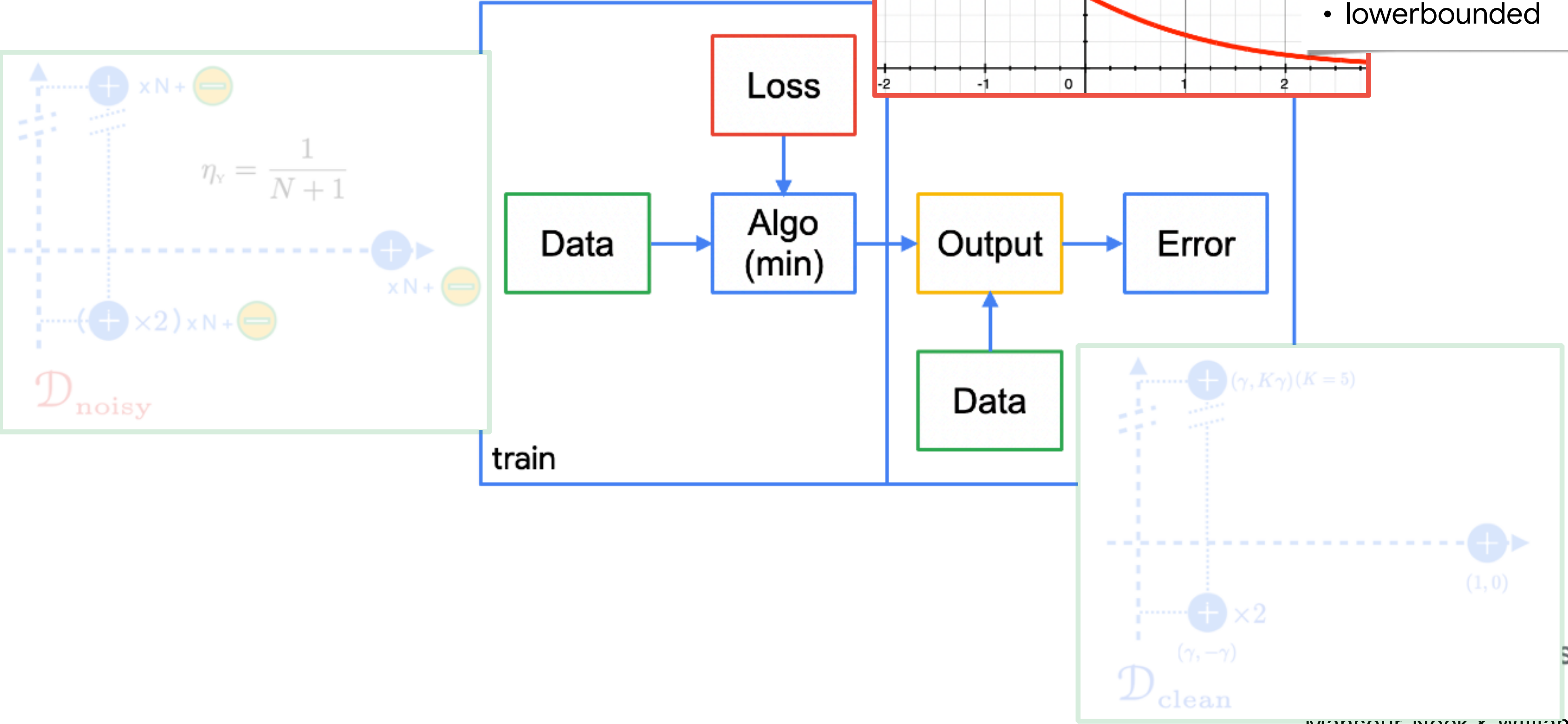
Setting II - data summary



Setting II - loss

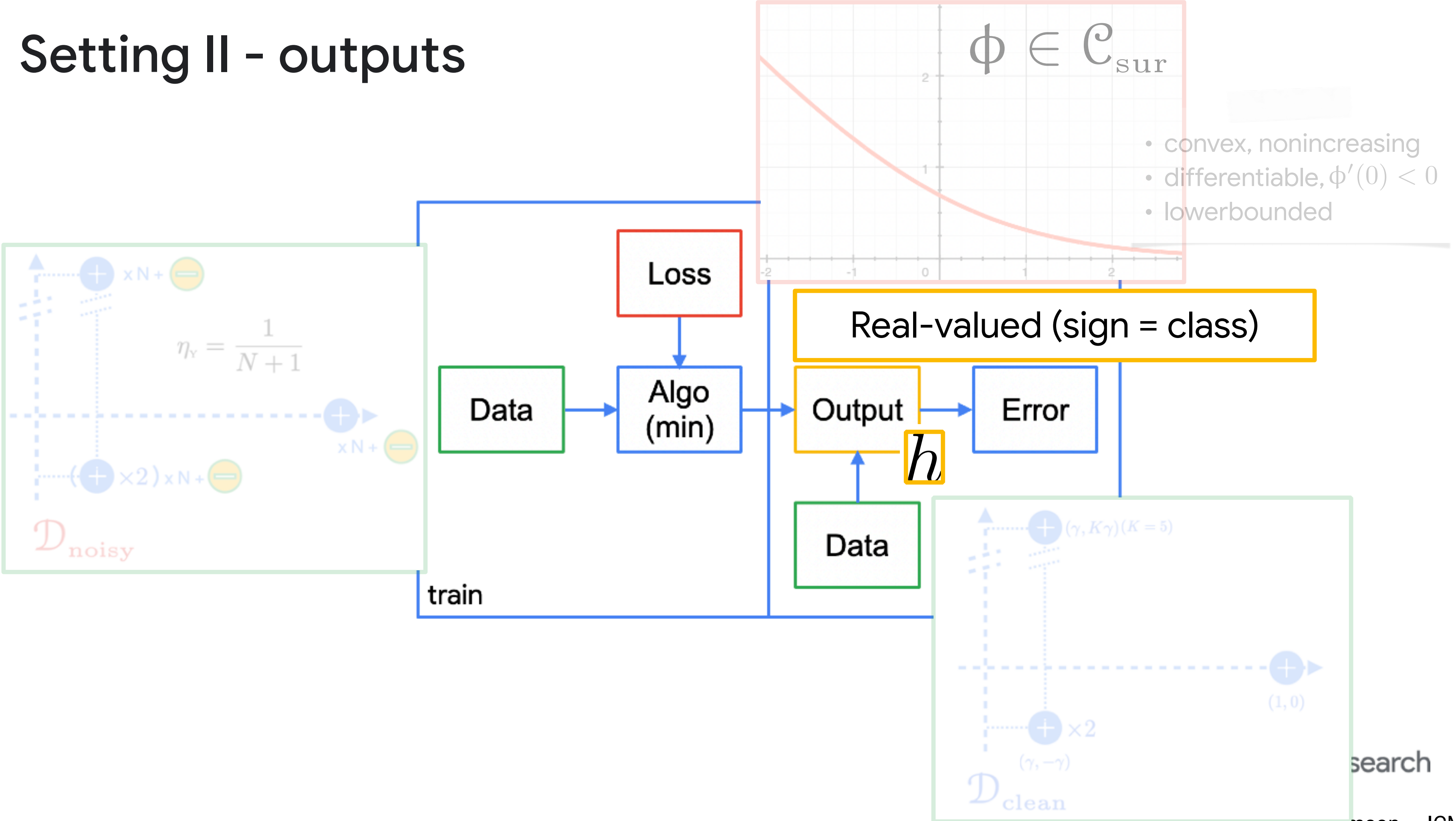


- convex, nonincreasing
- differentiable, $\phi'(0) < 0$
- lowerbounded

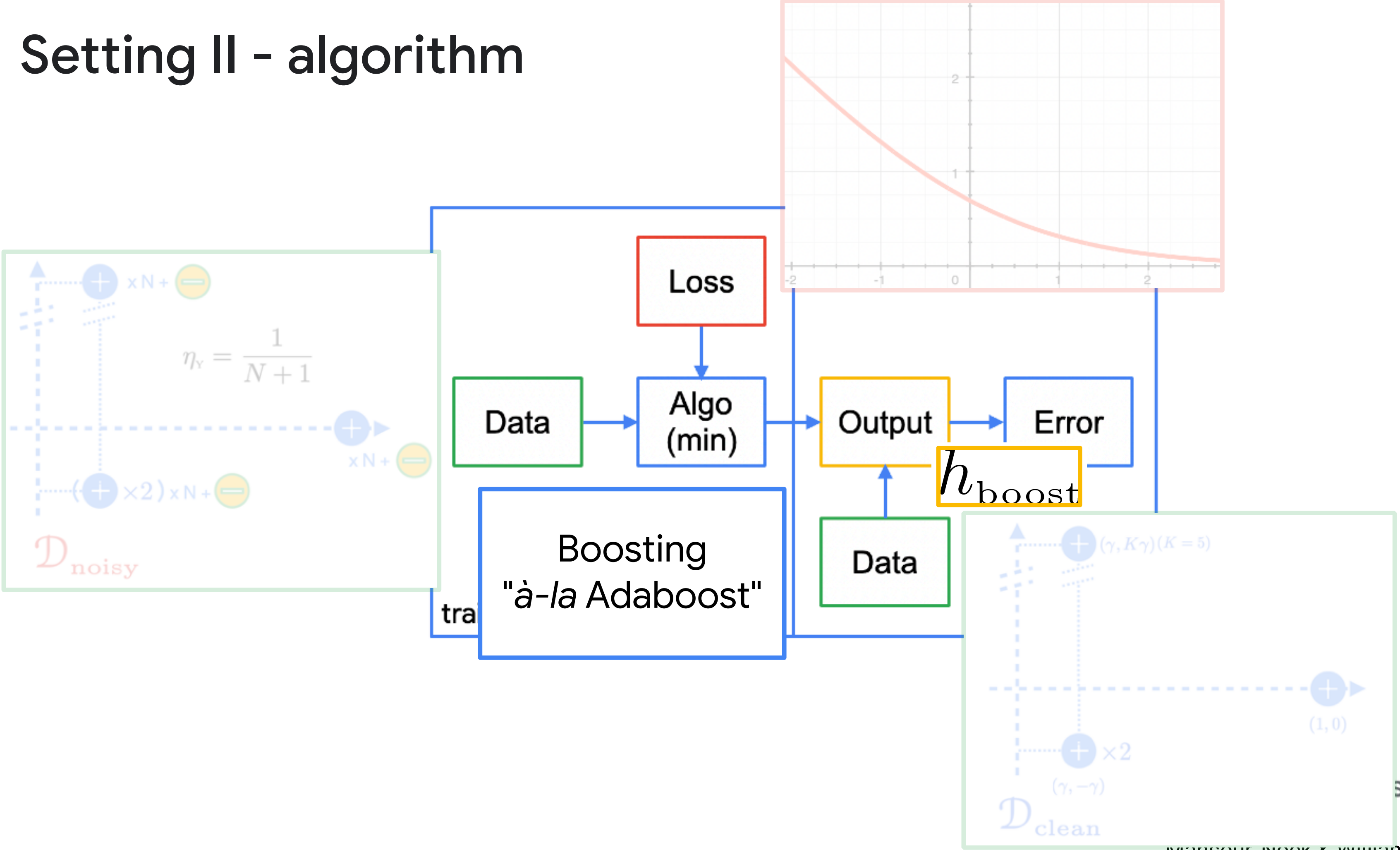


search

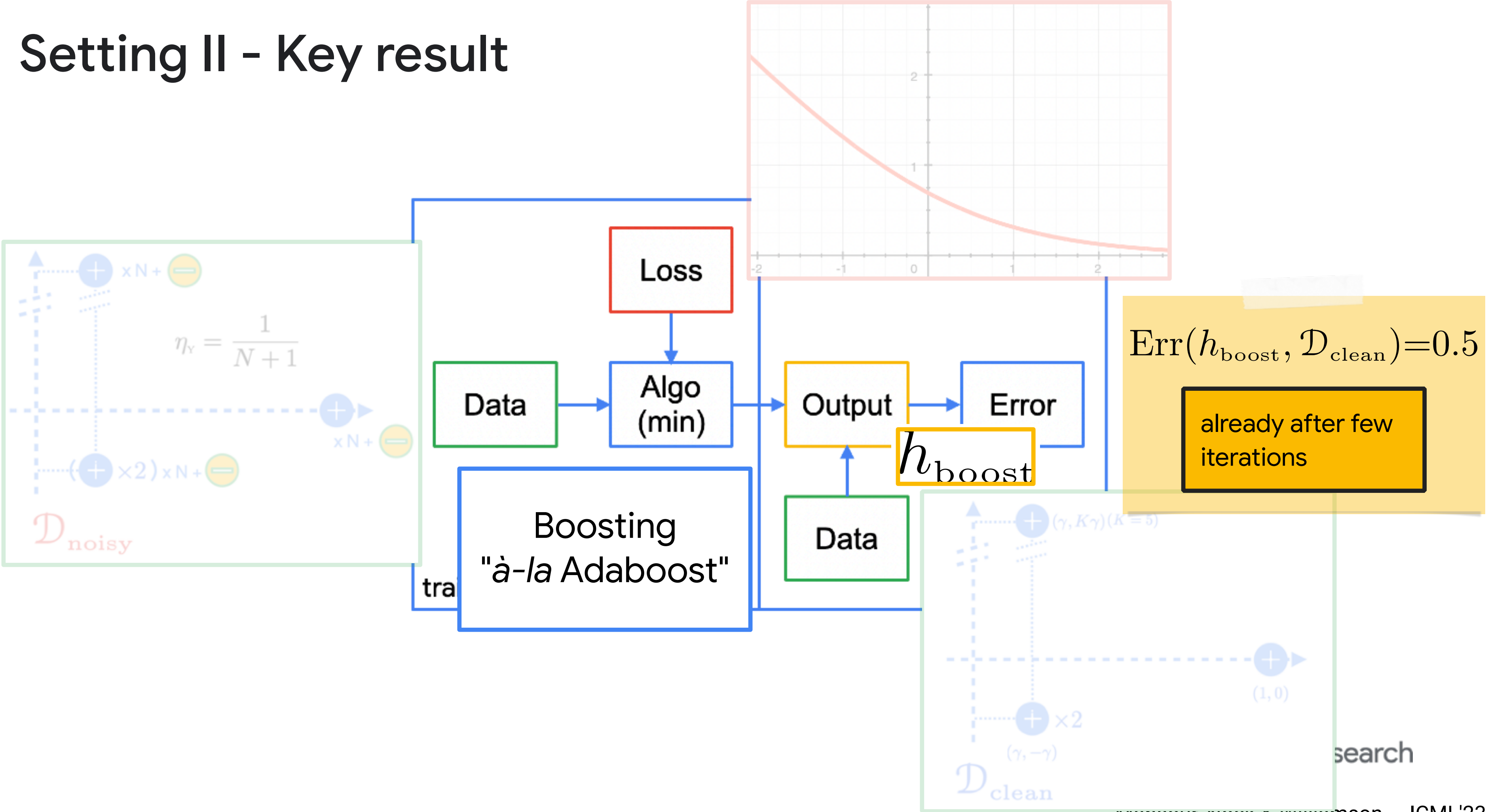
Setting II - outputs



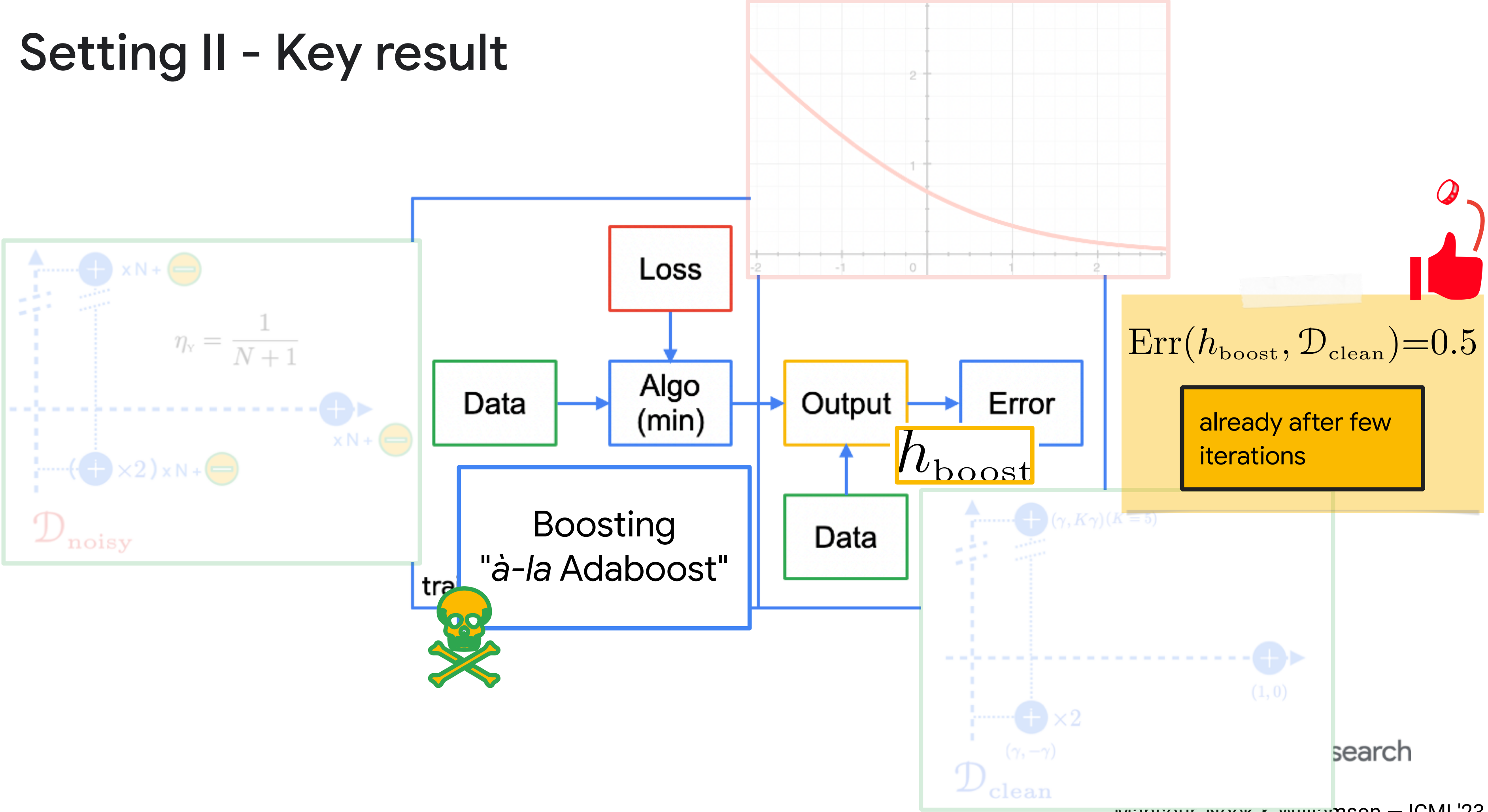
Setting II - algorithm



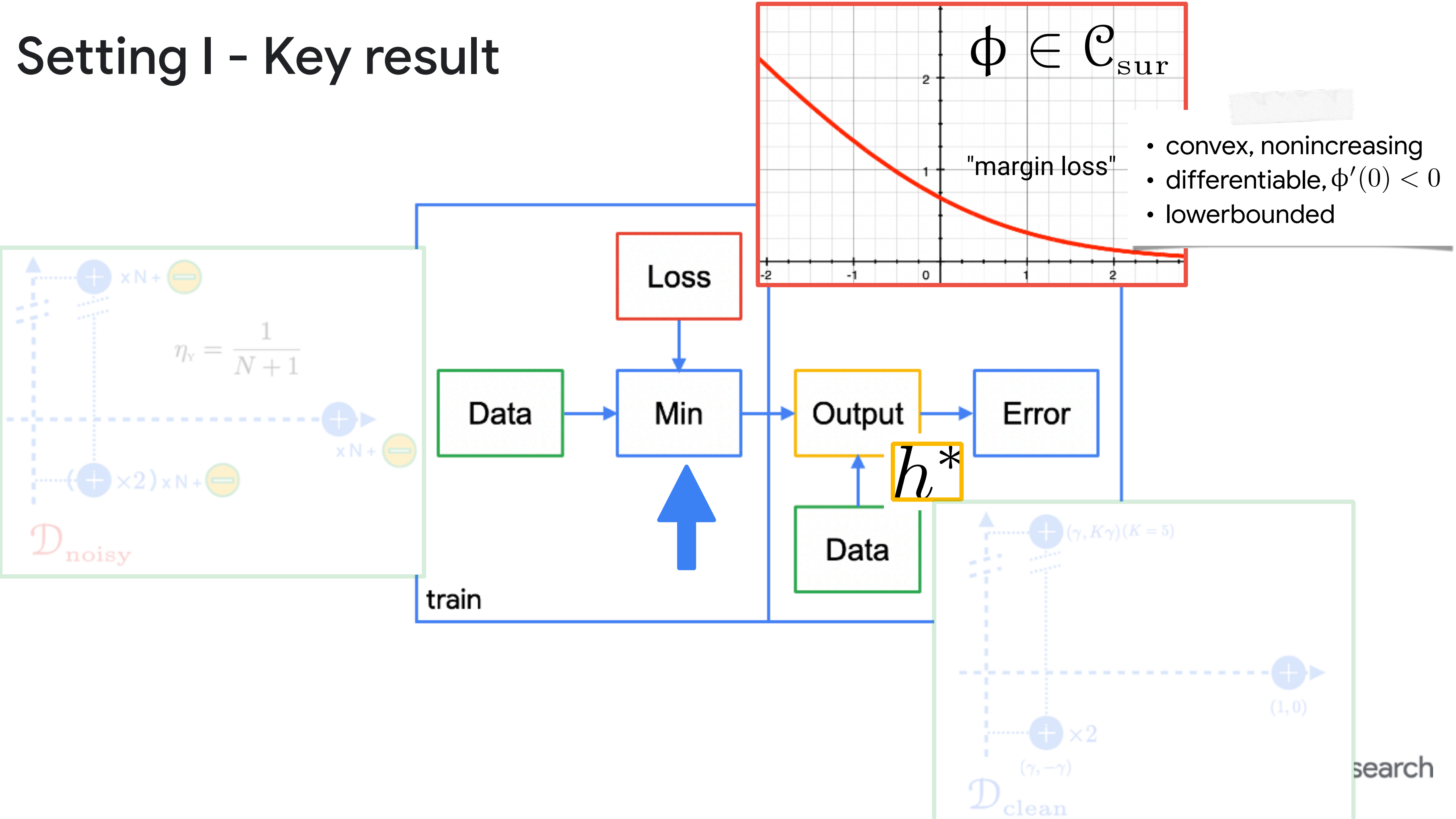
Setting II - Key result



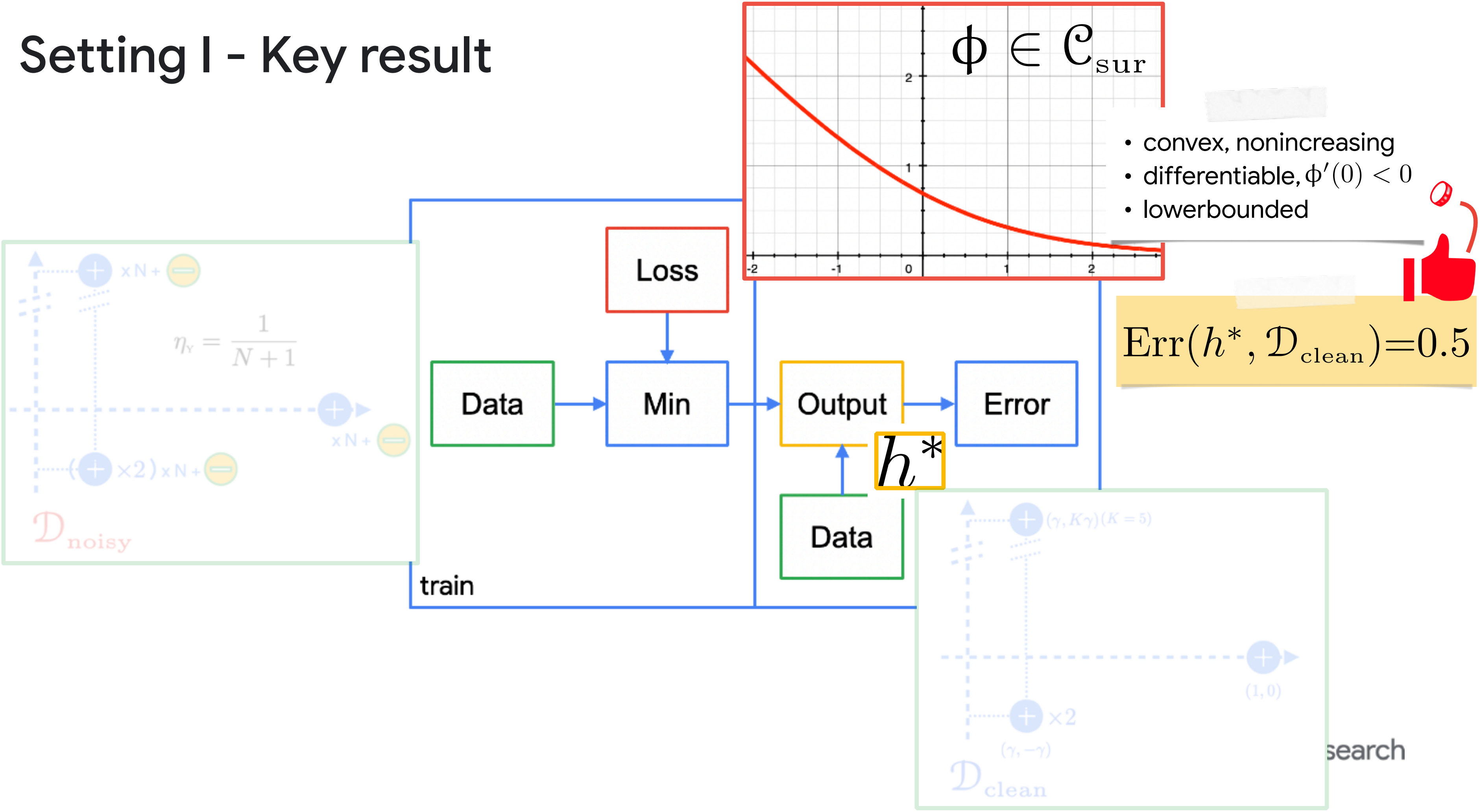
Setting II - Key result



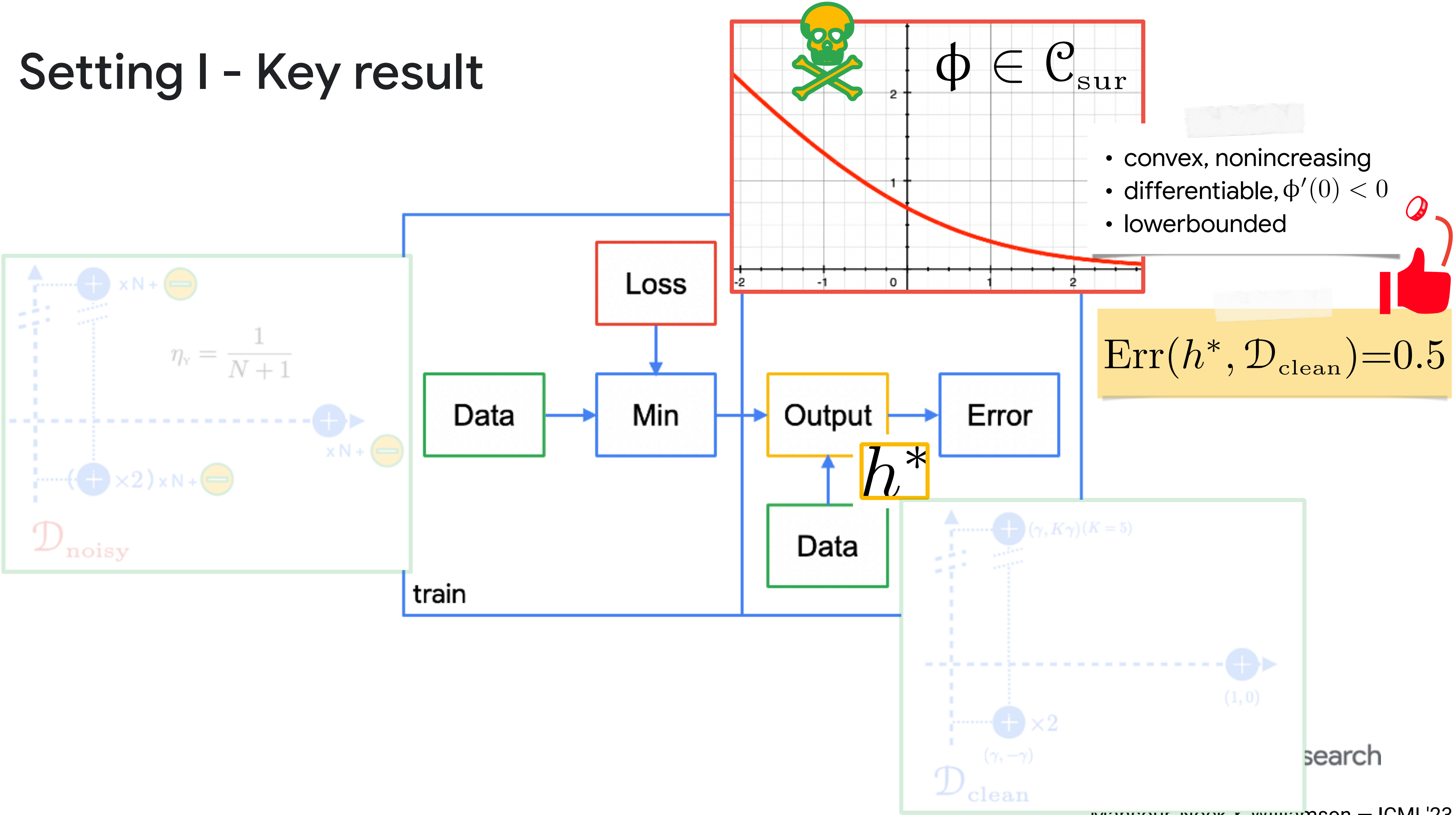
Setting I - Key result



Setting I - Key result



Setting I - Key result



the "simplest" form of corruption defeats two praised ML components: convex [losses | boosters]...

(x100s)

the "simplest" form of corruption
defeats two praised ML components:
convex [losses | boosters]...

or does it ?

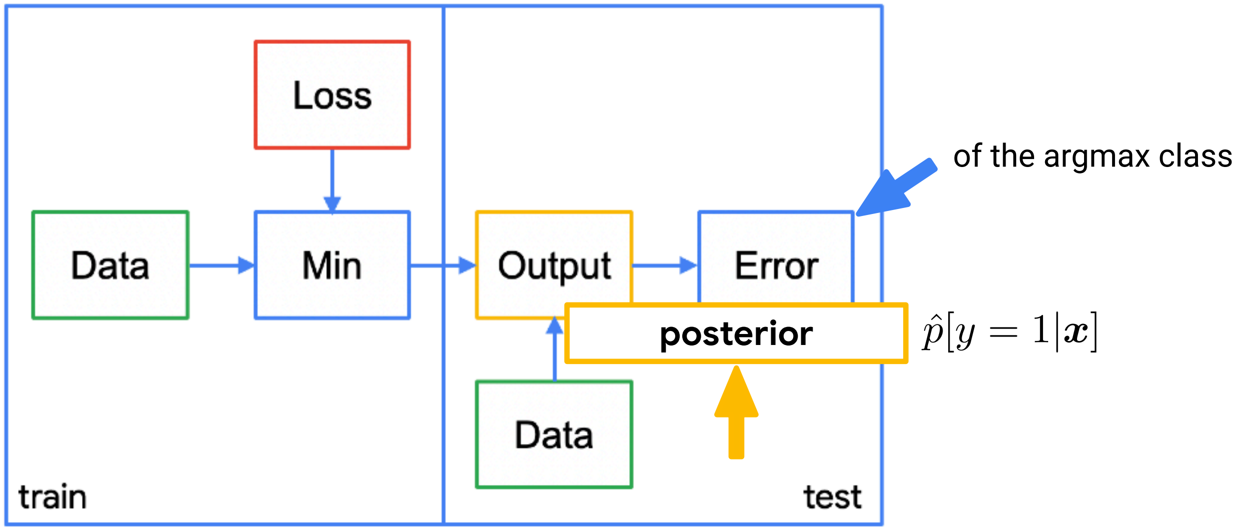
why this work

Enters Savage

Savage, L.-J. Elicitation of personal probabilities and expectations. *J. of the Am. Stat. Assoc.*, pp. 783–801, 1971.

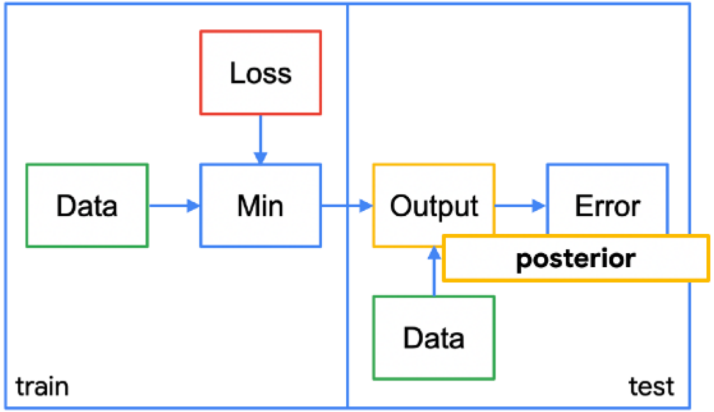
Google Research

Setting I tweak (temporary)

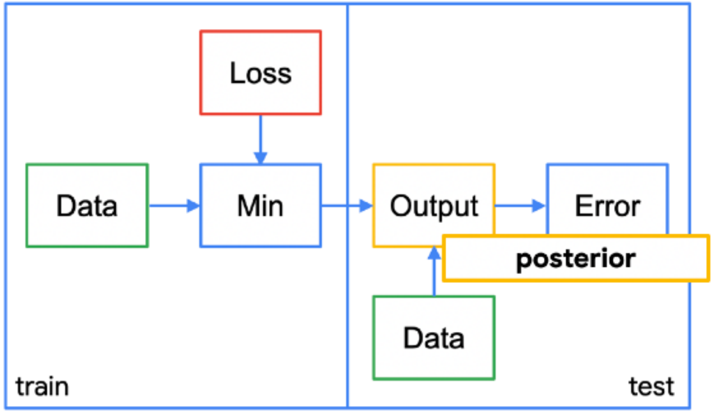


Class Probability Estimation

Class prediction → posterior prediction ($\hat{p}[y = 1|\boldsymbol{x}]$)



Class Probability Estimation



Class prediction → posterior prediction ($\hat{p}[y = 1|\boldsymbol{x}]$)

CPE loss (pointwise)

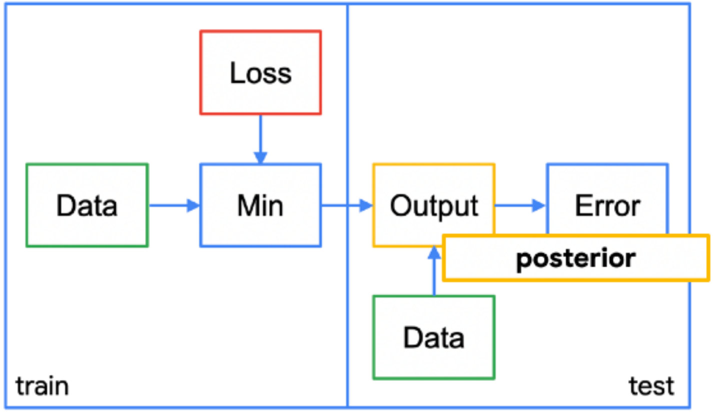
partial losses

$$\ell(y, u) \doteq [\![y = 1]\!] \cdot \ell_1(u) + [\![y = -1]\!] \cdot \ell_{-1}(u)$$

estimated posterior in [0,1]

true label / class in {-1,1}

Class Probability Estimation



Class prediction → posterior prediction ($\hat{p}[y = 1|\boldsymbol{x}]$)

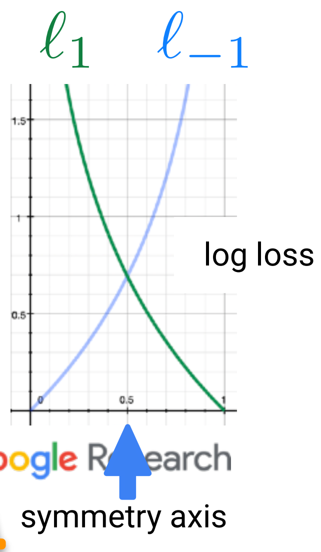
CPE loss (pointwise)

partial losses

$$\ell(y, u) \doteq [\![y = 1]\!] \cdot \ell_1(u) + [\![y = -1]\!] \cdot \ell_{-1}(u)$$

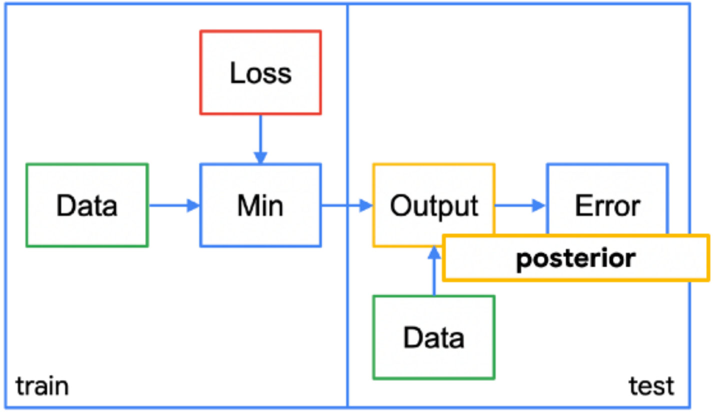
u: estimated posterior in [0,1]

y: true label / class in {-1,1}



Google Research

Class Probability Estimation



Class prediction → posterior prediction ($\hat{p}[y=1|\boldsymbol{x}]$)

CPE loss (pointwise)

partial losses

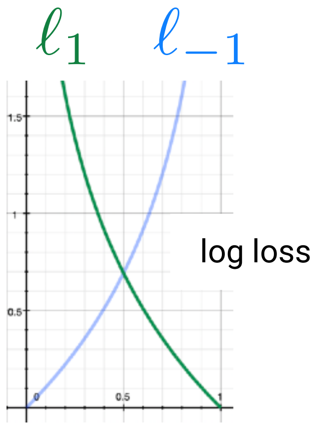
$$\ell(y,u) \doteq [\![y=1]\!] \cdot \ell_1(u) + [\![y=-1]\!] \cdot \ell_{-1}(u)$$

u: estimated posterior in [0,1]

y: true label / class in {-1,1}

CPE loss (population)

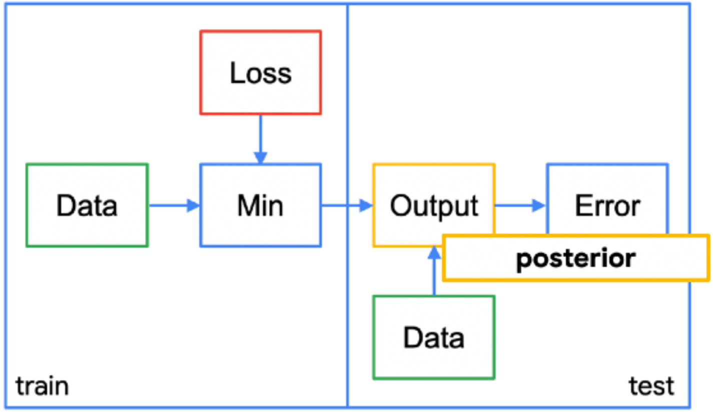
$$\Phi(\eta, \mathcal{D}) \doteq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\ell(y,\eta(\boldsymbol{x}))\right]$$



Google Research

Mansour, Nock & Williamson — ICML'23

Properness



Class prediction → posterior prediction ($\hat{p}[y = 1|\boldsymbol{x}]$)

CPE loss (pointwise)

partial losses

$$\ell(y, u) \doteq [\![y = 1]\!] \cdot \ell_1(u) + [\![y = -1]\!] \cdot \ell_{-1}(u)$$

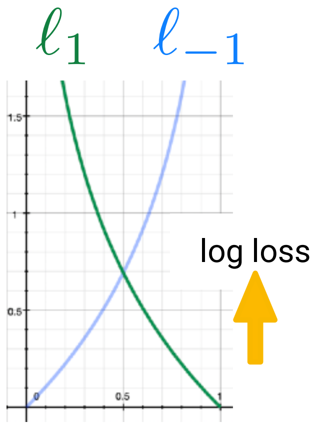
estimated posterior in [0,1]

true label / class in {-1,1}

CPE loss (population)

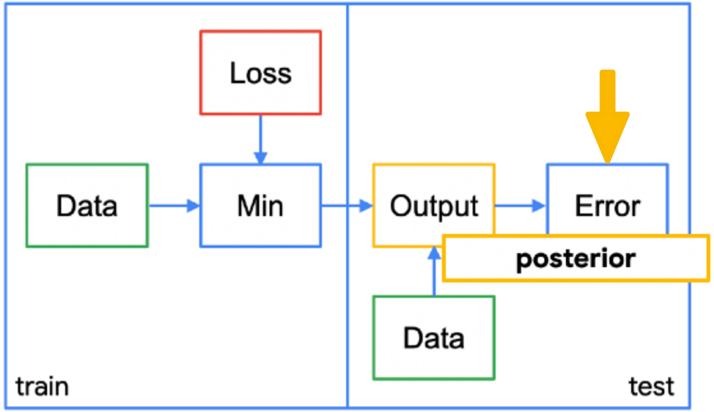
$$\Phi(\eta, \mathcal{D}) \doteq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\ell(y, \eta(\boldsymbol{x}))\right]$$

Quality: **strict properness** (strict optimum = Bayes prediction)



$$\eta_{\text{Bayes}} = \arg\min_{\eta} \Phi(\eta, \mathcal{D})$$

Properness



Class prediction → posterior prediction ($\hat{p}[y = 1|\boldsymbol{x}]$)

CPE loss (pointwise)

partial losses

$$\ell(y, u) \doteq [\![y = 1]\!] \cdot \ell_1(u) + [\![y = -1]\!] \cdot \ell_{-1}(u)$$

estimated posterior in [0,1]

true label / class in {-1,1}

CPE loss (population)

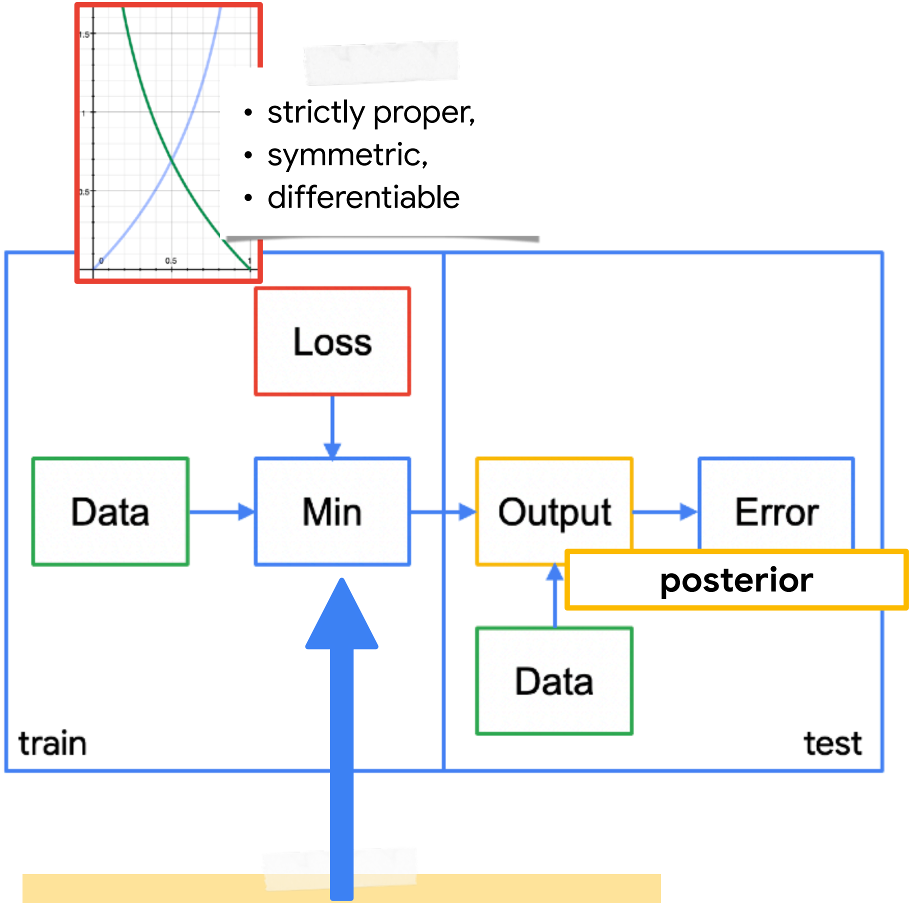
$$\Phi(\eta, \mathcal{D}) \doteq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} [\ell(y, \eta(\boldsymbol{x}))]$$

Quality: **properness** (optima $\ni$ Bayes prediction)

$$\eta_{\text{Bayes}} \in \arg\min_{\eta} \Phi(\eta, \mathcal{D})$$

Back to L&S (Setting I)

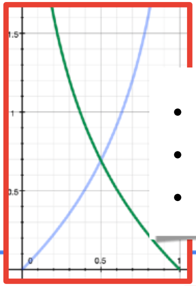
Savage on L&S (Setting I)



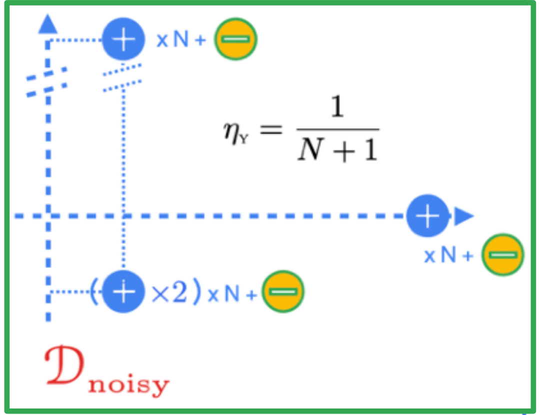
Because we train on the full domain, min sought = 

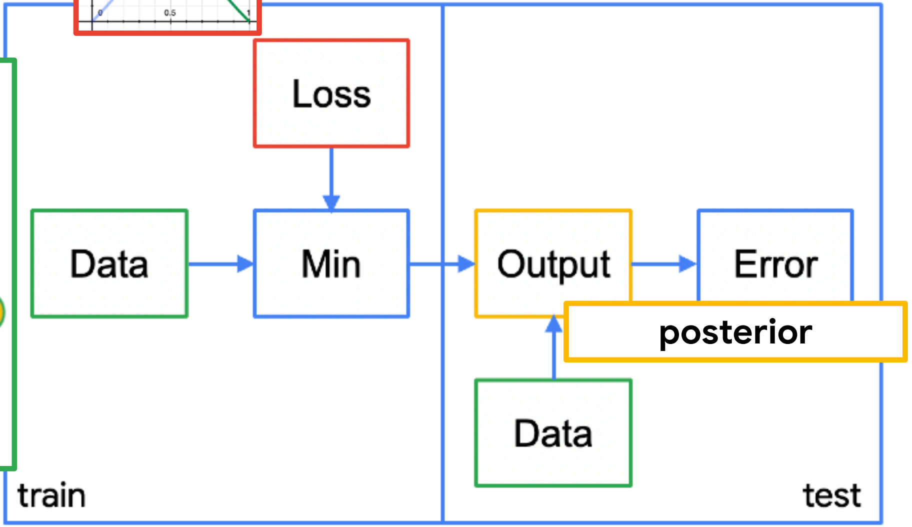
Google Research

Savage on L&S (Setting I)



- strictly proper,
- symmetric,
- differentiable

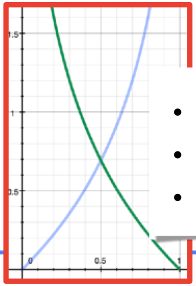




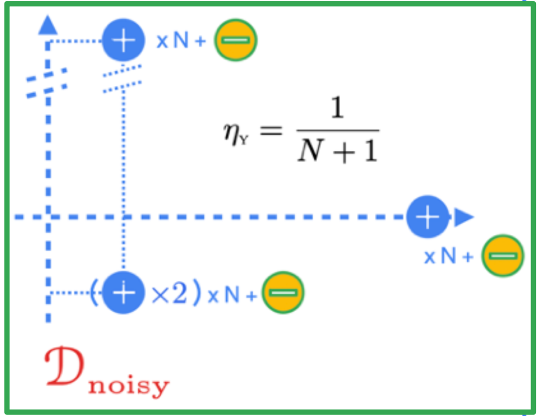
$$\eta_{\text{Bayes}} = \frac{N}{N+1}$$

$(\forall x\,!)$

Savage on L&S (Setting I)

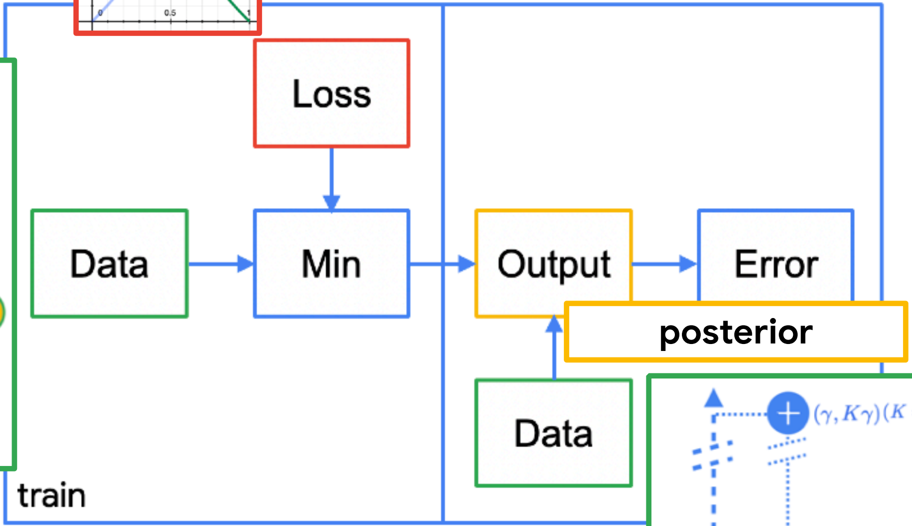


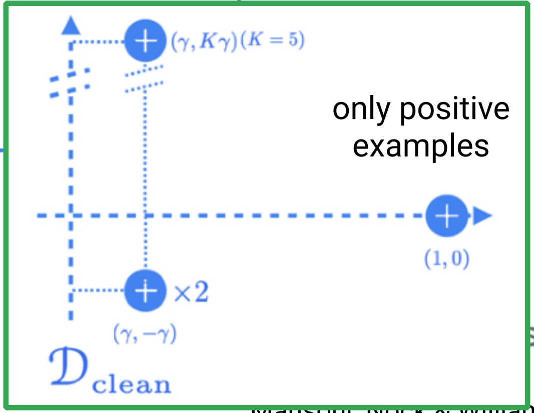
- strictly proper,
- symmetric,
- differentiable



$$\eta_{\text{Bayes}} = \frac{N}{N+1}$$

$(\forall \boldsymbol{x}\,!)$

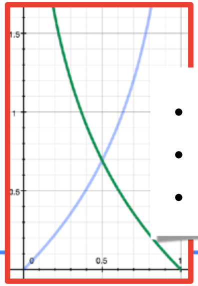




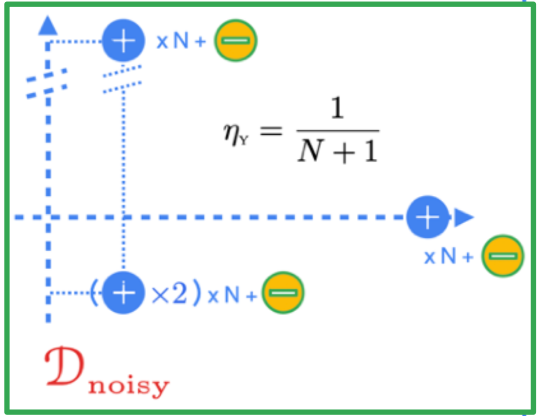
& $\eta_{\text{Bayes}} > \frac{1}{2}$

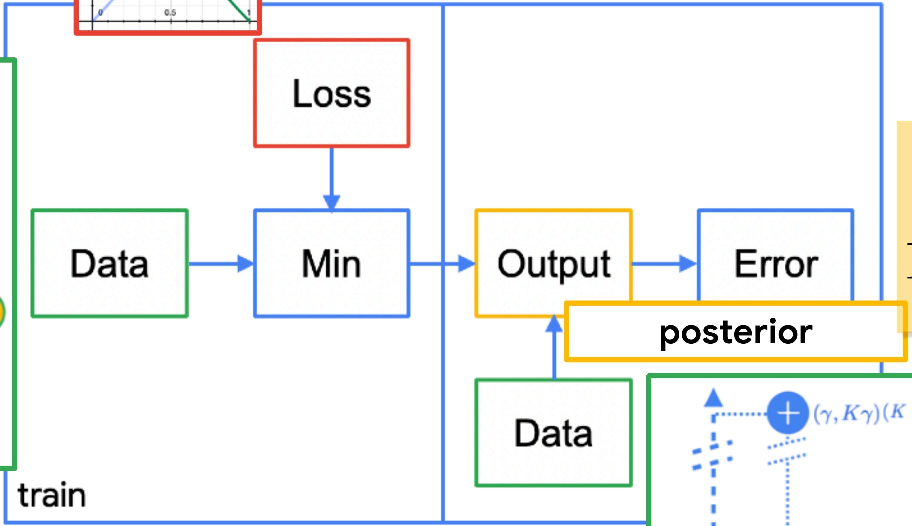
search

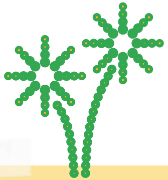
Savage on L&S (Setting I)



- strictly proper,
- symmetric,
- differentiable

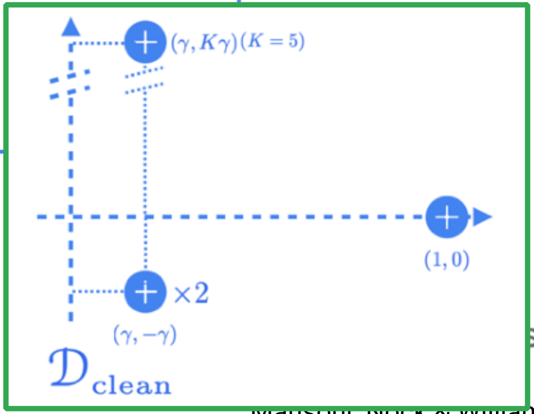






$$\mathrm{Err}(\eta_{\text{Bayes}}, \mathcal{D}_{\text{clean}}) = 0$$

$$\eta_{\text{Bayes}} > \frac{1}{2}$$



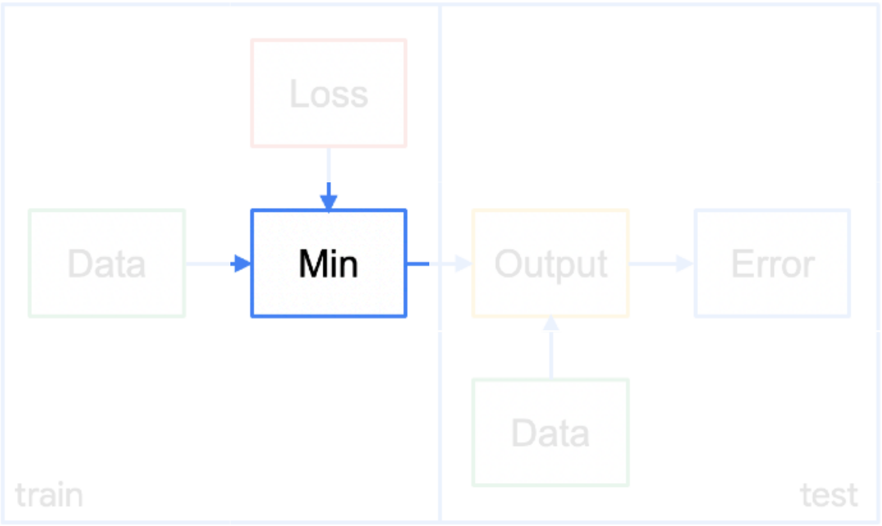
$$\eta_{\text{Bayes}} = \frac{N}{N+1}$$

$(\forall x!)$

search

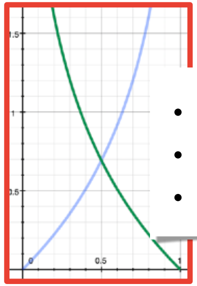
Savage on Setting I

L&S on Setting I



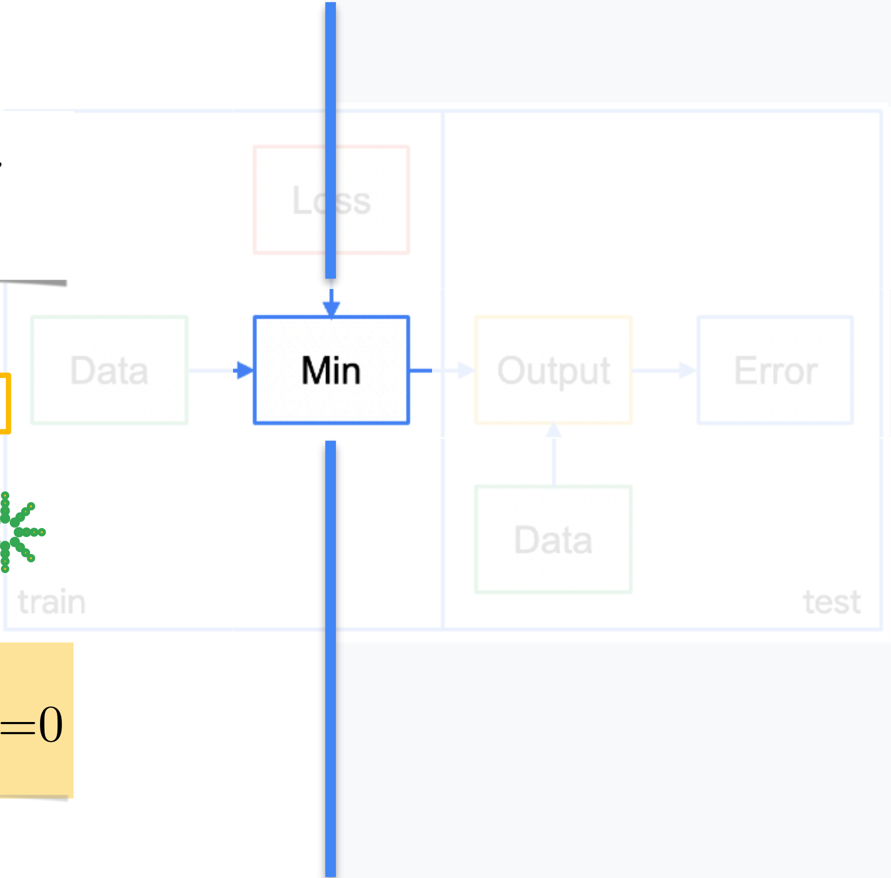
Google Research

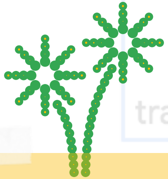
Savage on Setting I



- strictly proper,
- symmetric,
- differentiable

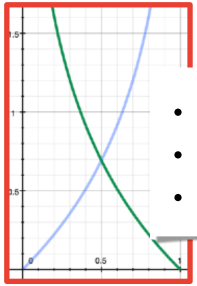
posterior η





$$\mathrm{Err}(\eta_{\mathrm{Bayes}}, \mathcal{D}_{\mathrm{clean}}) = 0$$

Savage on Setting I



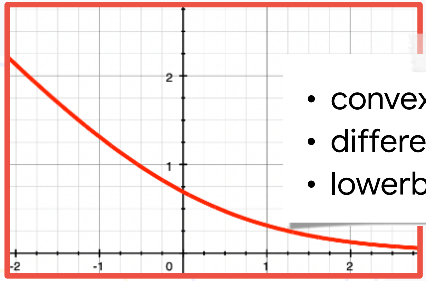
- strictly proper,
- symmetric,
- differentiable

posterior η

$$\mathrm{Err}(\eta_{\mathrm{Bayes}}, \mathcal{D}_{\mathrm{clean}}) = 0$$

Min

L&S on Setting I

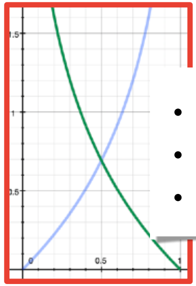


- convex, nonincreasing
- differentiable, $\phi'(0) < 0$
- lowerbounded

real-valued h

$$\mathrm{Err}(h^*, \mathcal{D}_{\mathrm{clean}}) = 0.5$$

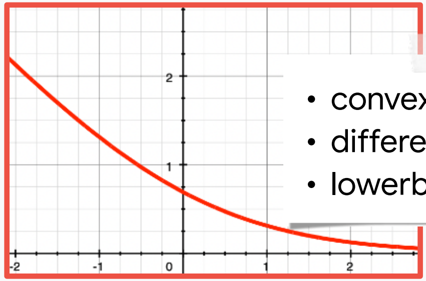
But...



- strictly proper,
- symmetric,
- differentiable

posterior η

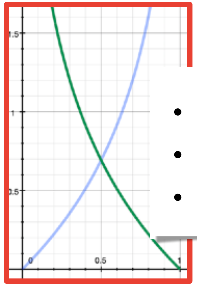
?



- convex, nonincreasing
- differentiable, $\phi'(0) < 0$
- lowerbounded

real-valued h

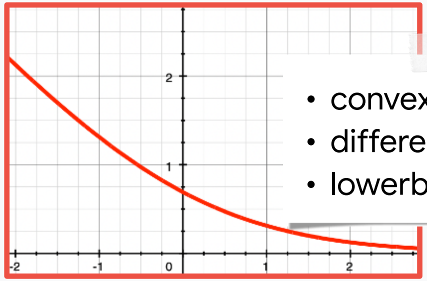
Google Research



- strictly proper,
- symmetric,
- differentiable

posterior η

$\subset$

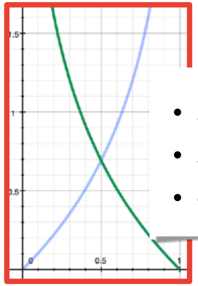


- convex, nonincreasing
- differentiable, $\phi'(0) < 0$
- lowerbounded

real-valued h

↳ Minimization of any* strictly proper, symmetric, differentiable CPE loss can be formulated as a convex surrogate minimization for a real valued classifier with a correspondence via the (canonical) link of the loss:

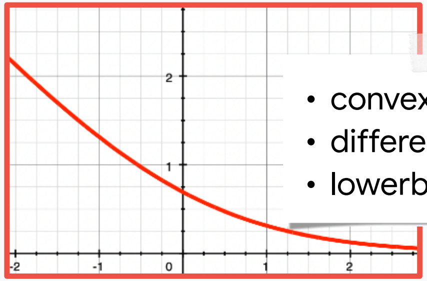
$$\eta \doteq (\ell_{-1} - \ell_1)^{-1}(h)$$



- strictly proper,
- symmetric,
- differentiable

posterior η

$\subset$



- convex, nonincreasing
- differentiable, $\phi'(0) < 0$
- lowerbounded

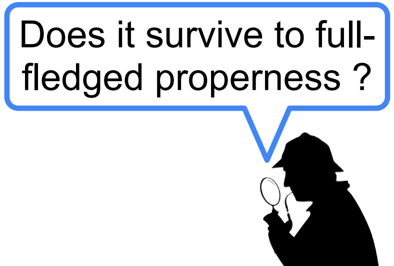
real-valued h

↳ Minimization of any* strictly proper, symmetric, differentiable CPE loss can be formulated as a convex surrogate minimization for a real valued classifier with a correspondence via the (canonical) link of the loss:

$$\eta \doteq (\ell_{-1} - \ell_1)^{-1}(h)$$

paradox ?

Google Research

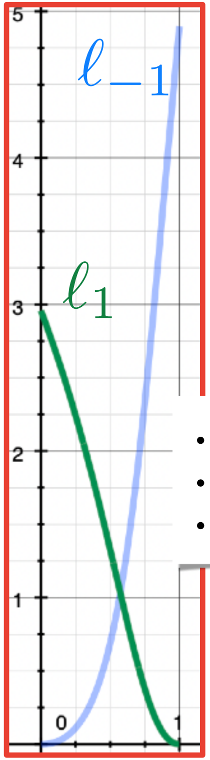


Google Research

What about properness without symmetry ?

Strict properness **without** symmetry assumption:

↳ asymmetry brings much more freedom to fine-tune costs



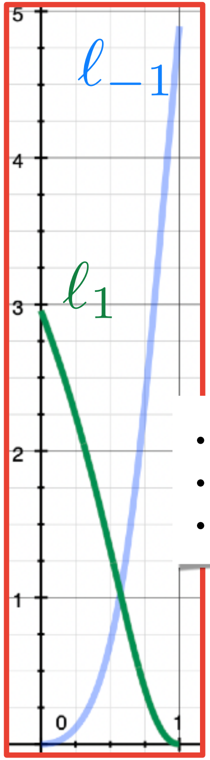
- strictly proper,
- symmetric,
- differentiable

What about properness without symmetry ?

Strict properness **without** symmetry assumption:

↳ asymmetry brings much more freedom to fine-tune costs

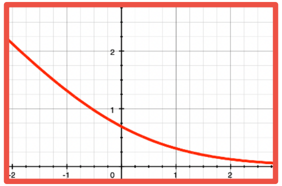
↳ no "classical" margin formulation anymore -- "escapes" Long & Servedio's setting



$\not\subset$

- strictly proper,
- symmetric,
- differentiable

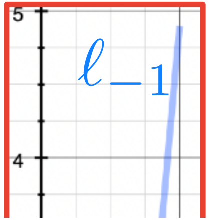
L&S



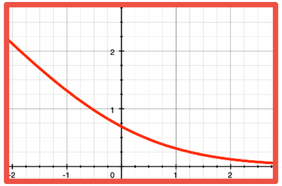
Google Research

What about properness without symmetry ?

Strict properness **without** symmetry assumption:



L&S



↳ asymmetry b
fine-tune co

↳ no "classical"
-- "escapes"

Setting II

$$\mathrm{Err}(h_{\mathrm{boost}}, \mathcal{D}_{\mathrm{clean}}) = 0.5$$

already after #iterations as small as **2**

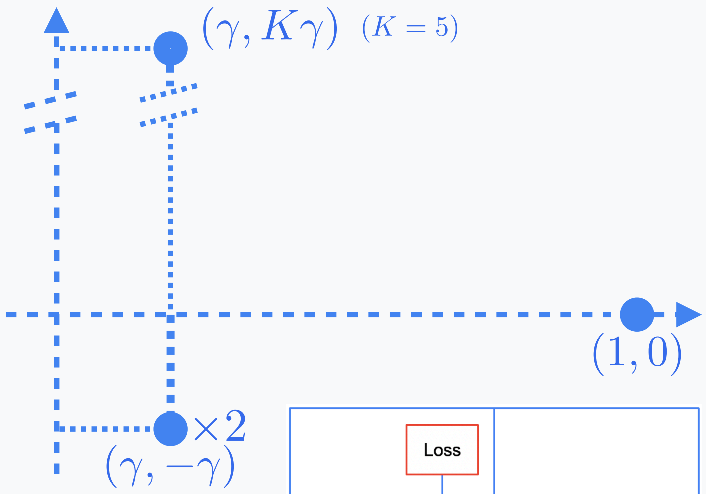
Setting I

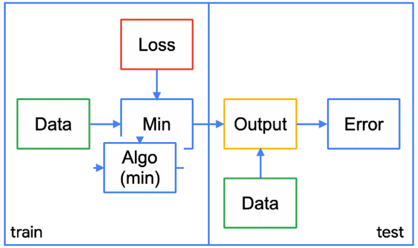
$$\mathrm{Err}(h^*, \mathcal{D}_{\mathrm{clean}}) = 0.5$$

properness as a whole "useless" !

Let us cut to the chase...

In-context, hardness has nothing to do with
↳ the convexity of the loss
↳ nor the fact that algorithm = boosting

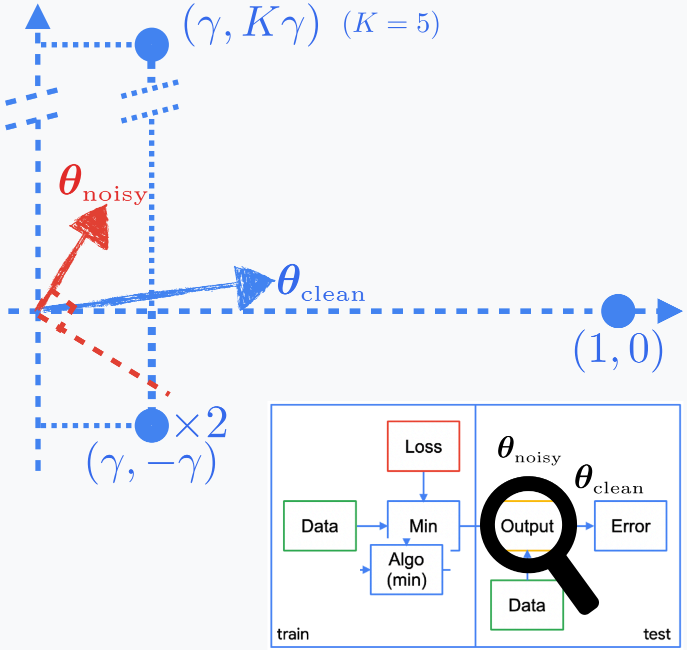




Google Research

Let us cut to the chase...

In-context, hardness has nothing to do with
↳ the convexity of the loss
↳ nor the fact that algorithm = boosting



Culprit = model class

Linear Separators "break" the guarantee of properness under the "simplest" noise model

... and ...

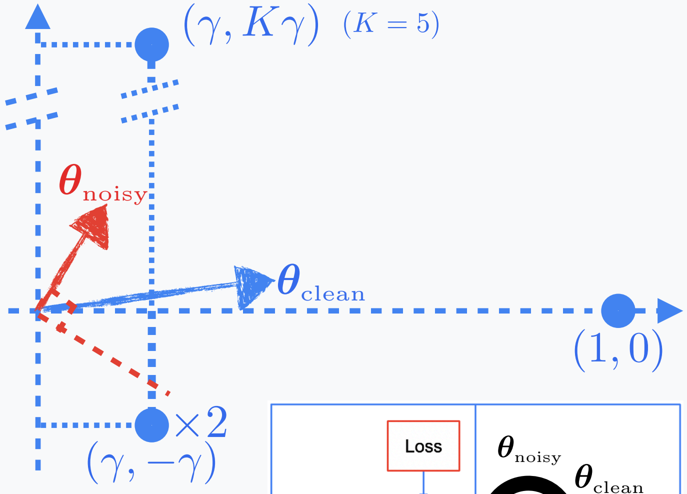
Google Research

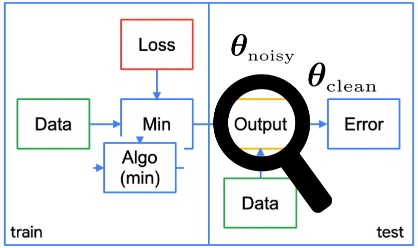
Let us cut to the chase...

In-context, hardness has nothing to do with

↳ the convexity of the loss

↳ nor the fact that algorithm = boosting





Culprit = model class

Linear Separators

"break" the guarantee of properness

under the "simplest" noise model

... and <u>we are also going to show it **constructively**</u>

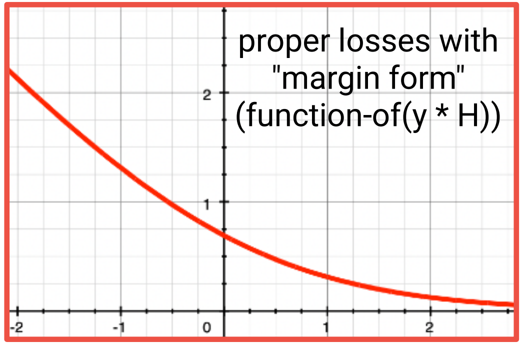
requires a new convex booster...

Google Research

Convex boosting, model-adaptive

ModaBoost (Model-Adaptive Boosting)

↳ **Start**: Adaboost-style boosting for
strictly proper, symmetric, differentiable losses



Convex boosting, model-adaptive

ModaBoost (Model-Adaptive Boosting)

↳ **Start:** Adaboost-style boosting for
strictly proper, symmetric, differentiable losses

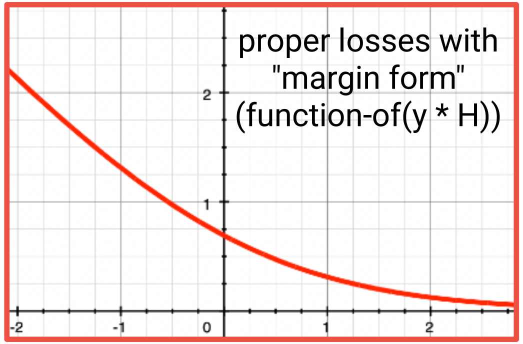
- Weights $\boldsymbol{w}$ = record of past performances
- ...
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random

$$|\mathbb{E}_{\boldsymbol{w}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

Weak Learning Assumption (WLA)

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns a linear model $H \doteq \sum_t \alpha_t \cdot h_t$



Google Research

Convex boosting, model-adaptive

ModaBoost (Model-Adaptive Boosting)

↳ **Step 1:** lift the applicable losses to all
strictly proper, symmetric, differentiable loss

- Weights $\boldsymbol{w}$ = record of past performances
- ...
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random

$$|\mathbb{E}_{\boldsymbol{w}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns a linear model $H \doteq \sum_t \alpha_t \cdot h_t$

no more "margin form"

two convex surrogates instead of 1

Convex boosting, model-adaptive

ModaBoost (Model-Adaptive Boosting)

↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$

Convex boosting, model-adaptive

ModaBoost (Model-Adaptive Boosting)

↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

Convex boosting, model-adaptive

ModaBoost (Model-Adaptive Boosting)

↳ **Step 2:** introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$

Boosting !

Theorem 1. *Suppose the following assumptions are satisfied on the loss and weak learner:*

LOSS *the loss is strictly proper differentiable; its partial losses are such that* $\exists \kappa > 0, C \in \mathbb{R}$,

$$\ell_{-1}(0), \ell_1(1) \geqslant C, \quad (18)$$

$$\inf\{\ell'_{-1} - \ell'_1\} \geqslant \kappa. \quad (19)$$

WLA *There exists a constant* $\gamma_{\mathrm{WL}} > 0$ *such that at each iteration* $t \in [T]$, *the weak hypothesis* h_t *returned by* WL *satisfies*

$$\left| \sum_{i \in [m]_t} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y_i \cdot \frac{h_t(\boldsymbol{x}_i)}{\max_{j \in [m]_t} |h_t(\boldsymbol{x}_j)|} \right| \geqslant \gamma_{\mathrm{WL}}. \quad (20)$$

AEOC *there exists a sequence* $\{u_t\}_{t \in \mathbb{N}_{>0}}$ *of strictly positive reals such that the choice of* $\mathcal{X}_t$ *in Step 2.1 is* u_t *compliant.*

Then for any $\theta \geqslant 0, \varepsilon > 0$, *letting* $\underline{w}(\theta) \doteq \min\{1 - (-\underline{L}')^{-1}(\theta), (-\underline{L}')^{-1}(-\theta)\}$, *if* ModaBoost *is run for at least*

$$T \geqslant U^{-1}\left(\frac{2\left(\Phi(H_0, \mathcal{S}) - C\right)}{\kappa \cdot \varepsilon^2 \underline{w}(\theta)^2 \gamma_{\mathrm{WL}}^2}\right) \quad (21)$$

iterations, then we are guaranteed

$$\mathbb{P}_{i \sim [m]}[y_i H_T(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon. \quad (22)$$

Here, U *is crafted as in* (17).

Theorem 1. *Suppose the following assumptions are satisfied on the loss and weak learner:*

LOSS *the loss is strictly proper differentiable; its partial losses are such that* $\exists \kappa > 0, C \in \mathbb{R}$,

$$\begin{aligned} \ell_{-1}(0), \ell_1(1) &\geqslant C, & (18)\\ \inf\{\ell'_{-1} - \ell'_1\} &\geqslant \kappa. & (19) \end{aligned}$$

Assumptions on loss,
"necessary"

WLA *There exists a constant* $\gamma_{\mathrm{WL}} > 0$ *such that at each iteration* $t \in [T]$*, the weak hypothesis* h_t *returned by* WL *satisfies*

$$\left| \sum_{i \in [m]_t} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y_i \cdot \frac{h_t(\boldsymbol{x}_i)}{\max_{j \in [m]_t} |h_t(\boldsymbol{x}_j)|} \right| \geqslant \gamma_{\mathrm{WL}}. \quad (20)$$

AEOC *there exists a sequence* $\{u_t\}_{t \in \mathbb{N}_{>0}}$ *of strictly positive reals such that the choice of* $\mathcal{X}_t$ *in Step 2.1 is* u_t *compliant.*

Then for any $\theta \geqslant 0, \varepsilon > 0$*, letting* $\underline{w}(\theta) \doteq \min\{1 - (-\underline{L}')^{-1}(\theta), (-\underline{L}')^{-1}(-\theta)\}$*, if* MODABOOST *is run for at least*

$$T \geqslant U^{-1}\left(\frac{2\,(\Phi(H_0, \mathcal{S}) - C)}{\kappa \cdot \varepsilon^2 \underline{w}(\theta)^2 \gamma_{\mathrm{WL}}^2}\right) \quad (21)$$

iterations, then we are guaranteed

$$\mathbb{P}_{i \sim [m]}[y_i\, H_T(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon. \quad (22)$$

Here, U *is crafted as in* (17).

Theorem 1. *Suppose the following assumptions are satisfied on the loss and weak learner:*

LOSS *the loss is strictly proper differentiable; its partial losses are such that* $\exists \kappa > 0, C \in \mathbb{R}$,

$$\ell_{-1}(0), \ell_1(1) \geqslant C, \quad (18)$$
$$\inf\{\ell'_{-1} - \ell'_1\} \geqslant \kappa. \quad (19)$$

WLA *There exists a constant* $\gamma_{\mathrm{WL}} > 0$ *such that at each iteration* $t \in [T]$, *the weak hypothesis* h_t *returned by* WL *satisfies*

$$\left| \sum_{i \in [m]_t} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y_i \cdot \frac{h_t(\boldsymbol{x}_i)}{\max_{j \in [m]_t} |h_t(\boldsymbol{x}_j)|} \right| \geqslant \gamma_{\mathrm{WL}}. \quad (20)$$

AEOC *there exists a sequence* $\{u_t\}_{t \in \mathbb{N}_{>0}}$ *of strictly positive reals such that the choice of* $\mathcal{X}_t$ *in Step 2.1 is* u_t *compliant.*

Then for any $\theta \geqslant 0, \varepsilon > 0$, *letting* $\underline{w}(\theta) \doteq \min\{1 - (-\underline{L}')^{-1}(\theta), (-\underline{L}')^{-1}(-\theta)\}$, *if* ModaBoost *is run for at least*

$$T \geqslant U^{-1}\left(\frac{2\left(\Phi(H_0, \mathcal{S}) - C\right)}{\kappa \cdot \varepsilon^2 \underline{w}(\theta)^2 \gamma_{\mathrm{WL}}^2} \right) \quad (21)$$

iterations, then we are guaranteed

$$\mathbb{P}_{i \sim [m]}[y_i^* H_T(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon. \quad (22)$$

Here, U *is crafted as in* (17).

Theorem 1. *Suppose the following assumptions are satisfied on the loss and weak learner:*

LOSS *the loss is strictly proper differentiable; its partial losses are such that* $\exists \kappa > 0, C \in \mathbb{R}$,

$$\begin{aligned} \ell_{-1}(0), \ell_1(1) &\geqslant C, & (18)\\ \inf\{\ell'_{-1} - \ell'_1\} &\geqslant \kappa. & (19) \end{aligned}$$

WLA *There exists a constant* $\gamma_{\text{WL}} > 0$ *such that at each iteration* $t \in [T]$, *the weak hypothesis* h_t *returned by* WL *satisfies*

$$\left| \sum_{i \in [m]_t} \frac{w_{t,i}}{\sum_{j \in [m]_t} w_{t,j}} \cdot y_i \cdot \frac{h_t(\boldsymbol{x}_i)}{\max_{j \in [m]_t} |h_t(\boldsymbol{x}_j)|} \right| \geqslant \gamma_{\text{WL}}. \quad (20)$$

AEOC *there exists a sequence* $\{u_t\}_{t \in \mathbb{N}_{>0}}$ *of strictly positive reals such that the choice of* $\mathcal{X}_t$ *in Step 2.1 is* u_t *compliant.*

Then for any $\theta \geqslant 0, \varepsilon > 0$, *letting* $\underline{w}(\theta) \doteq \min\{1 - (-\underline{L}')^{-1}(\theta), (-\underline{L}')^{-1}(-\theta)\}$, *if* ModaBoost *is run for at least*

$$T \geqslant U^{-1}\left(\frac{2\left(\Phi(H_0, \mathcal{S}) - C\right)}{\kappa \cdot \varepsilon^2 \underline{w}(\theta)^2 \gamma^2_{\text{WL}}}\right) \quad (21)$$

iterations, then we are guaranteed

$$\mathbb{P}_{i \sim [m]}[y_i H_T(\boldsymbol{x}_i) \leqslant \theta] < \varepsilon. \quad (22)$$

guarantee on edges / margins

Here, U *is crafted as in* (17).

Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

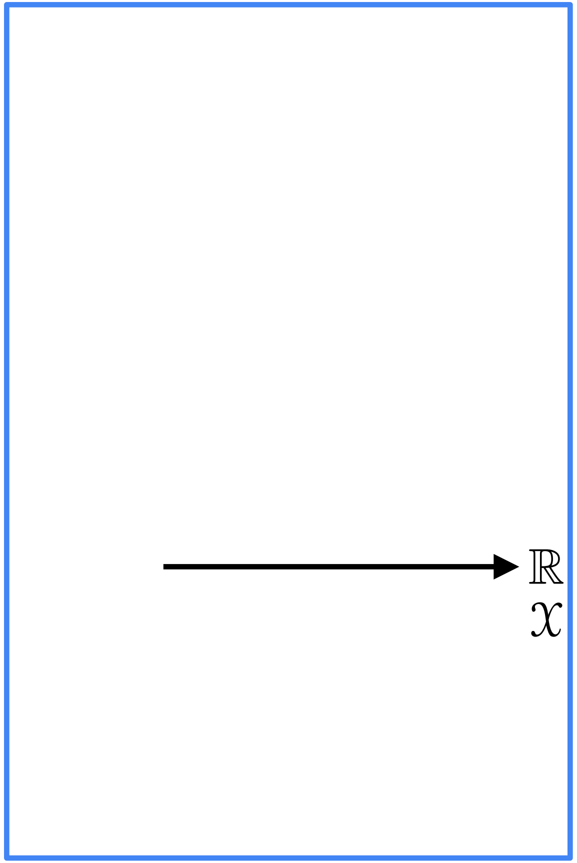
- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$

AEO → Models != Linear Models



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

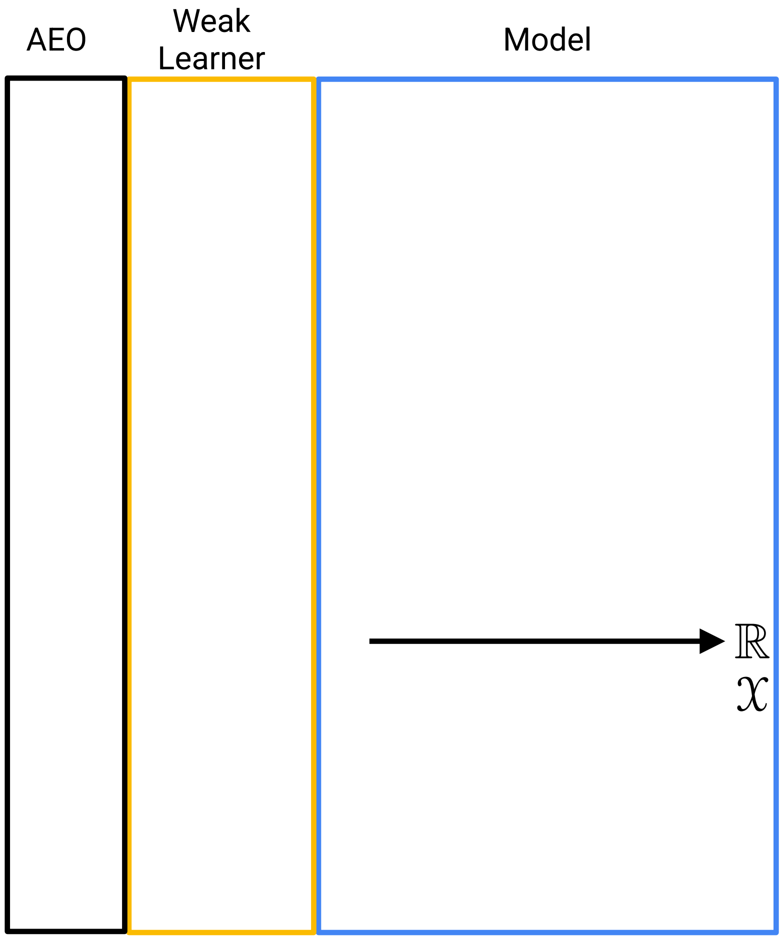
↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

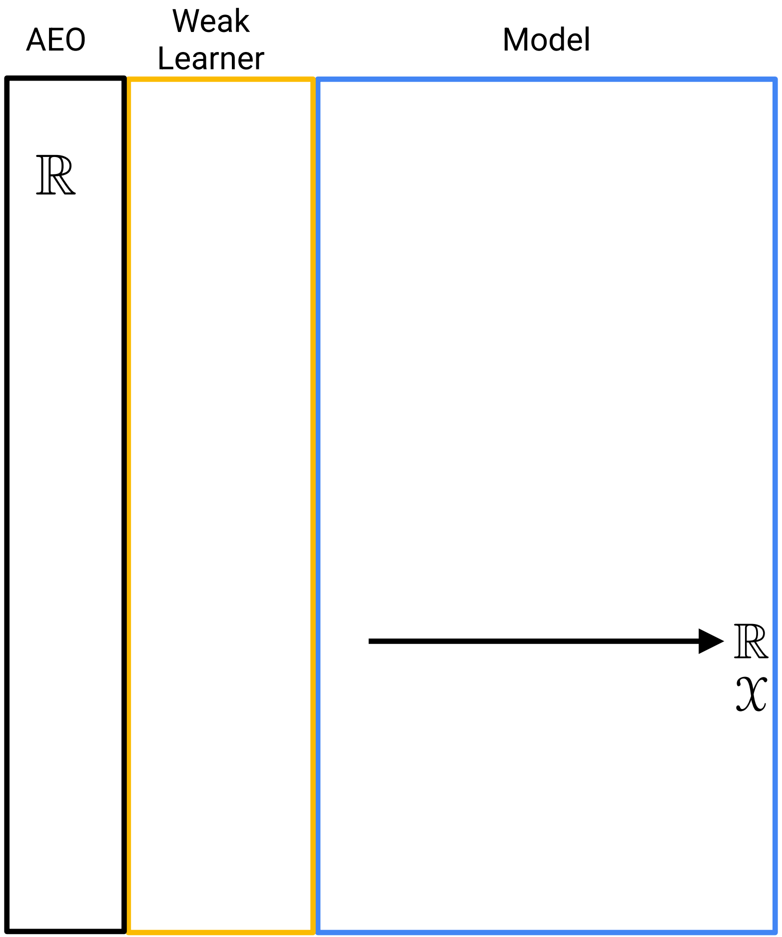
↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

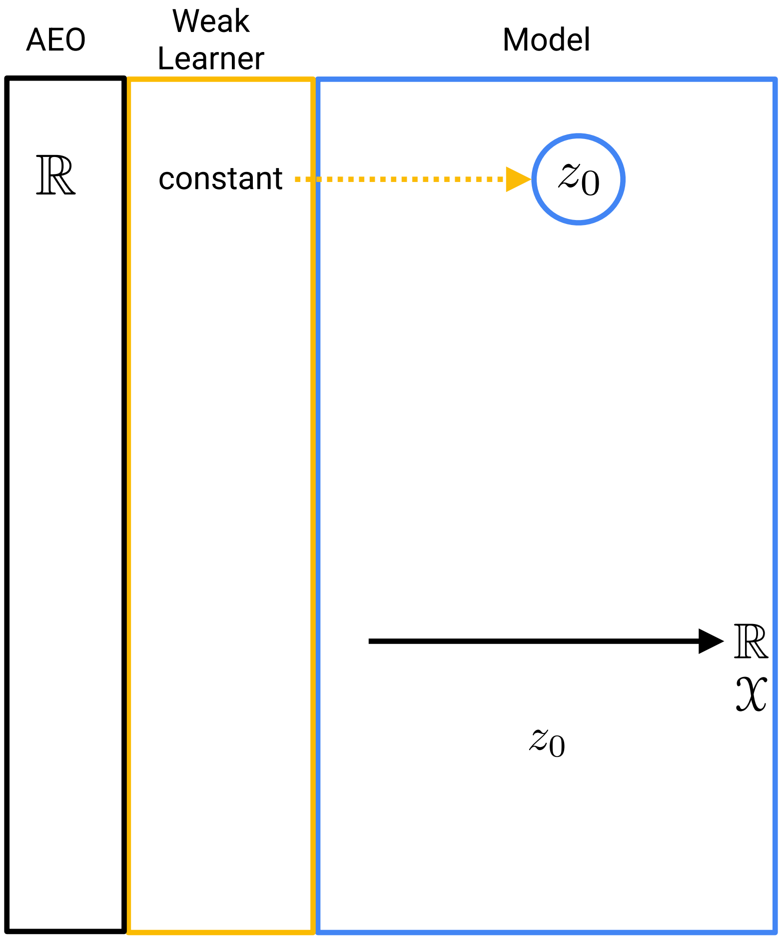
↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

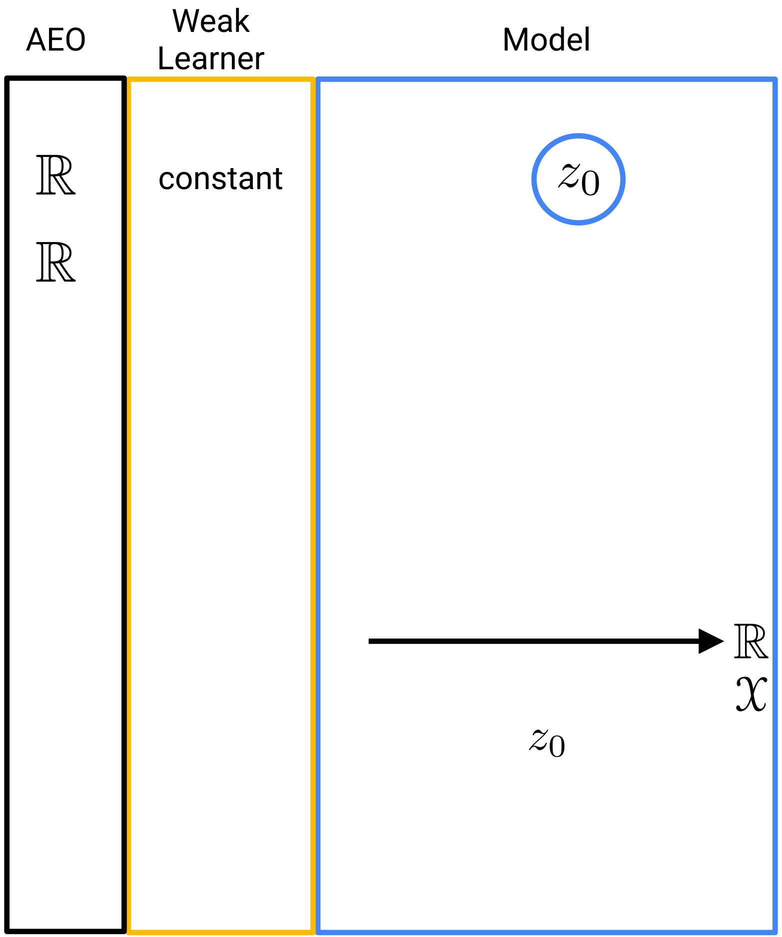
↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

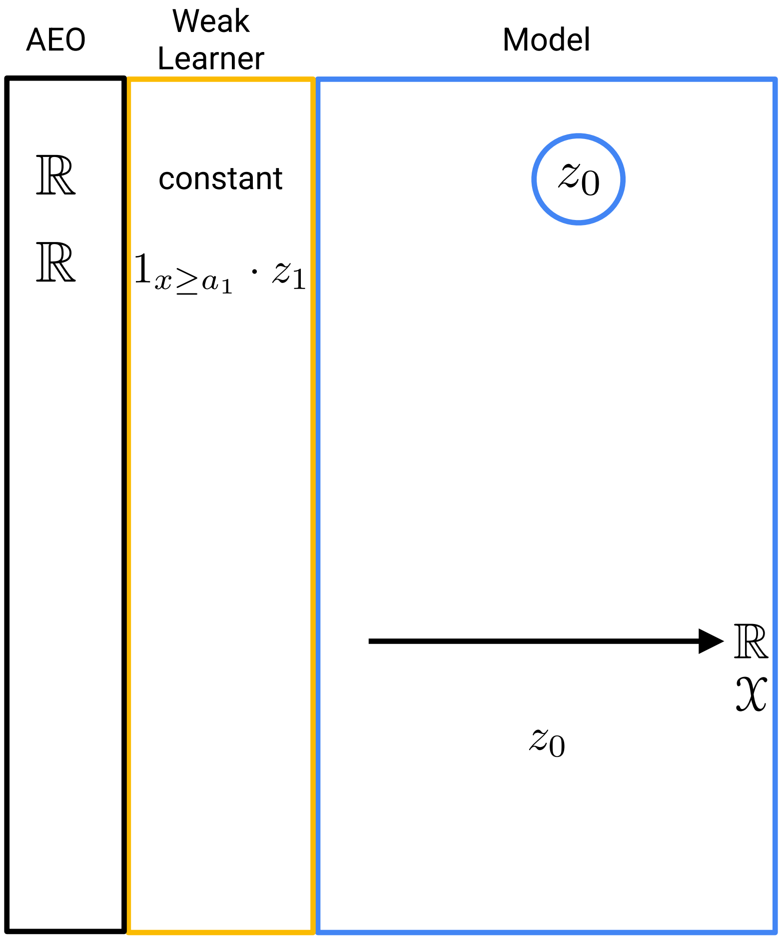
↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

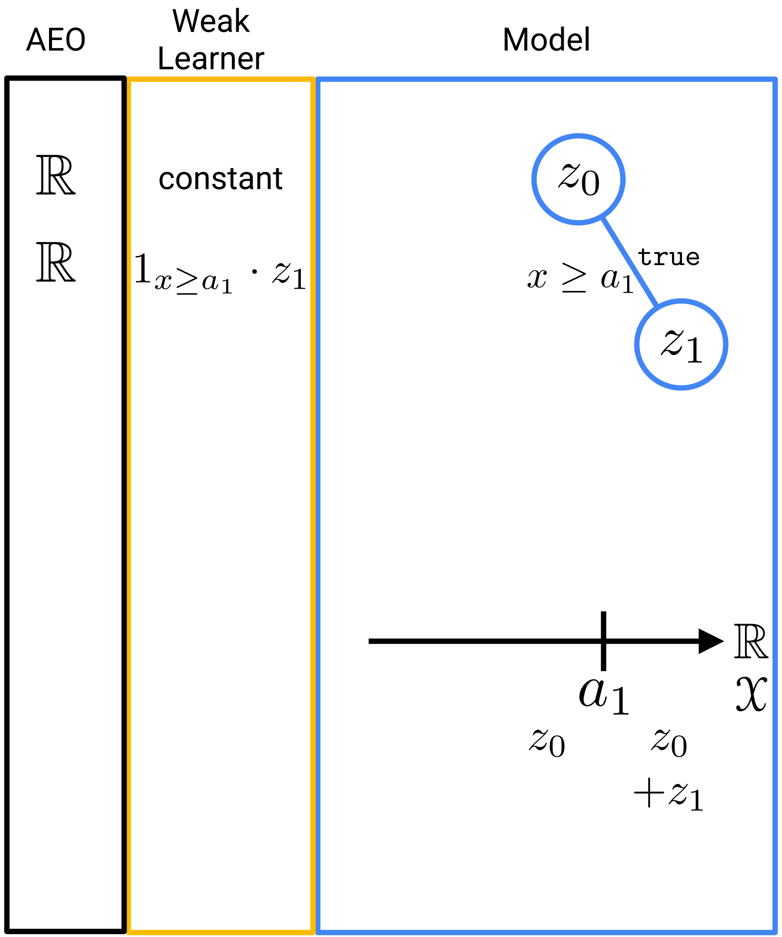
↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

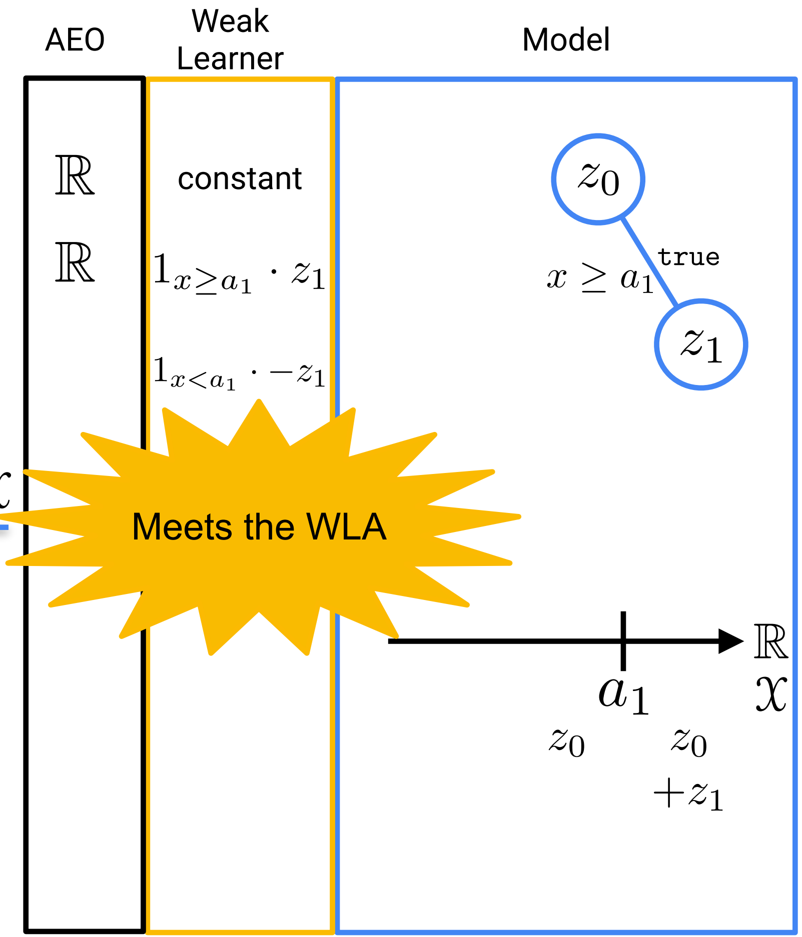
↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

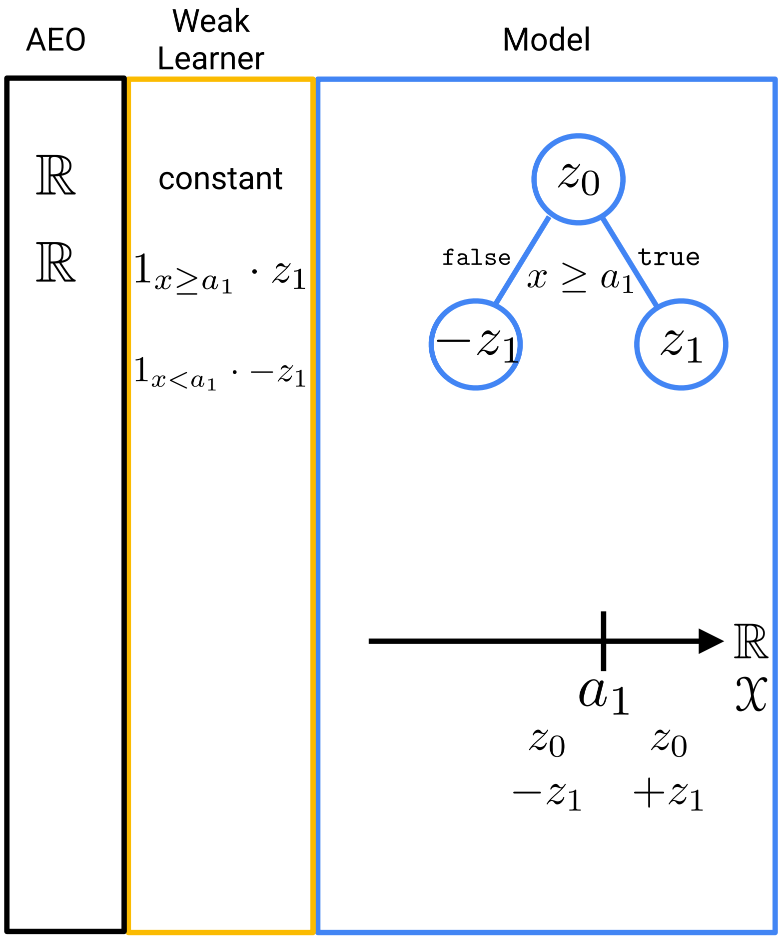
↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

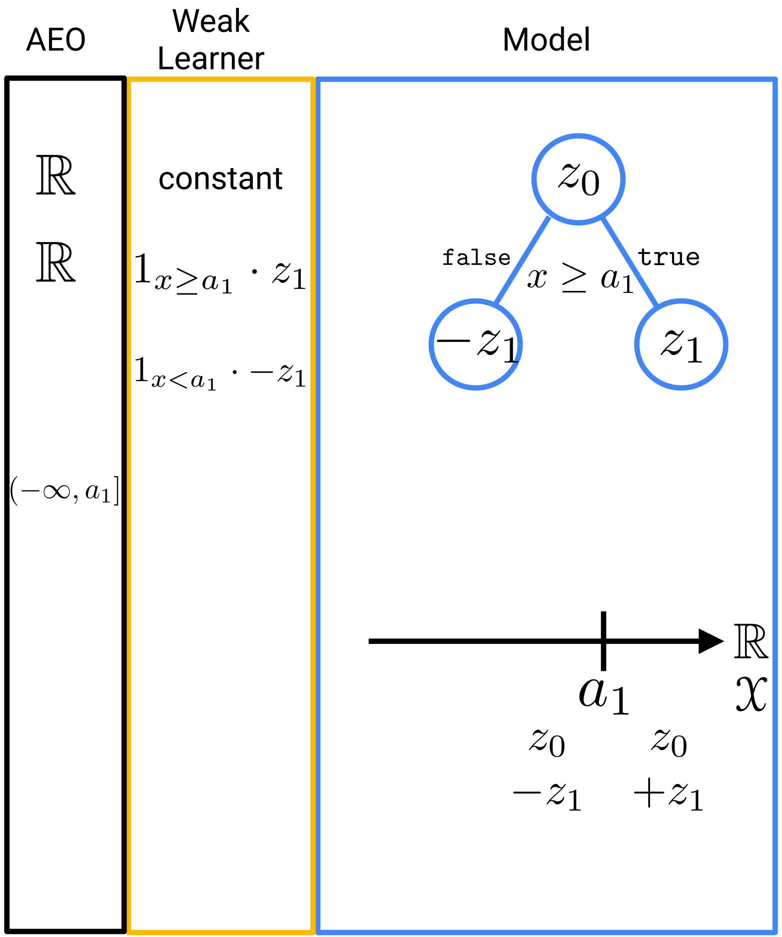
↳ **Step 2:** introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x}\in\mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

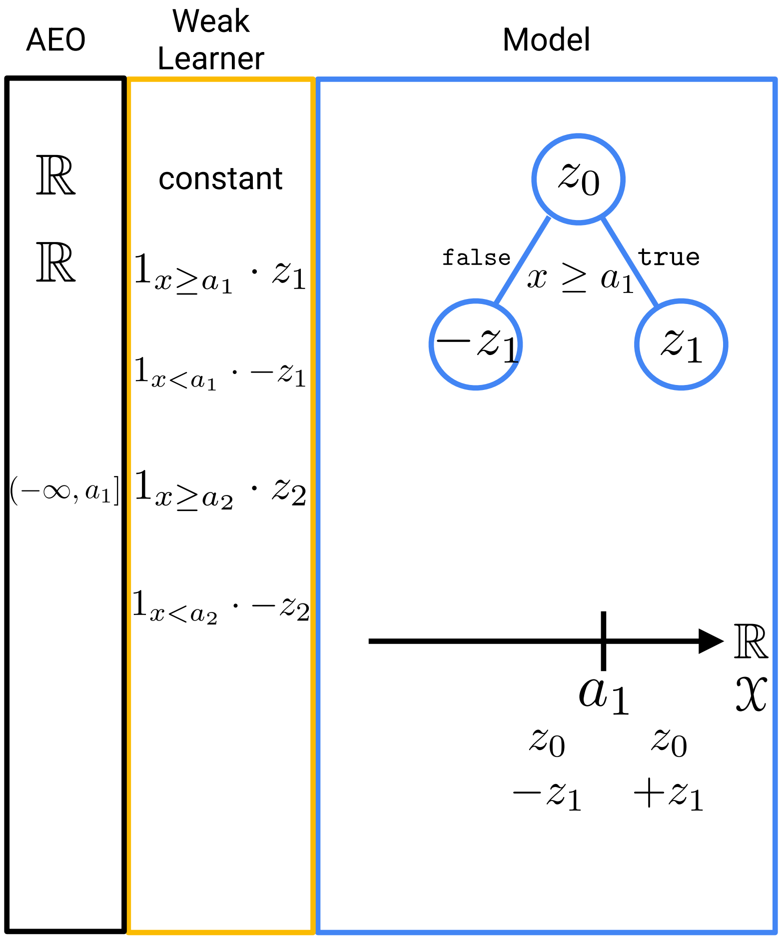
↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

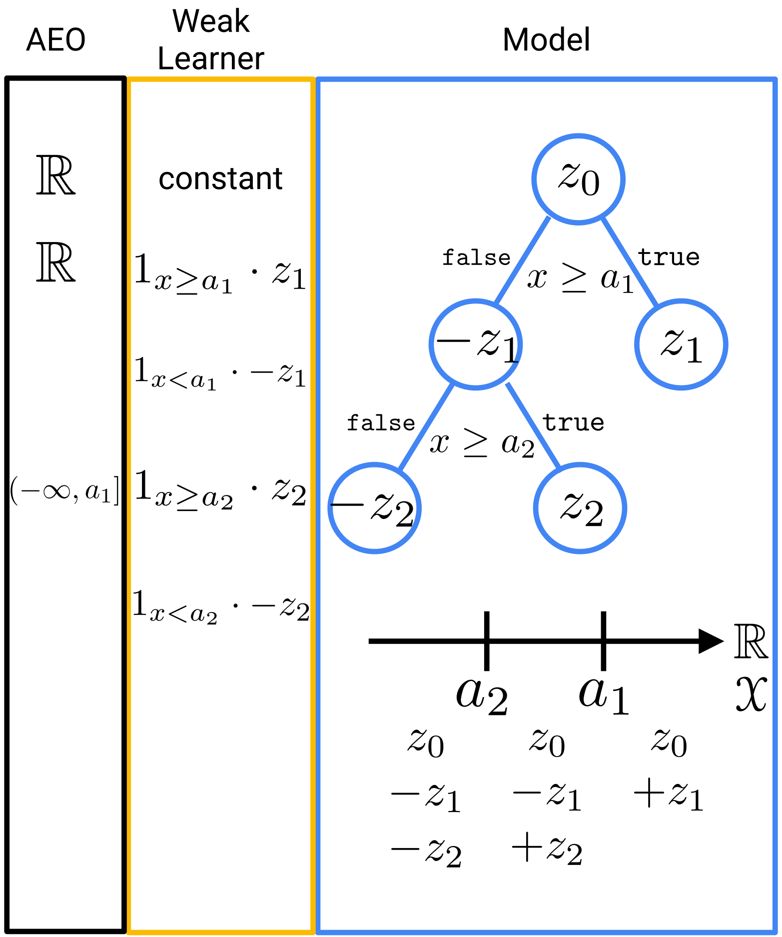
↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

↳ **Step 2:** introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

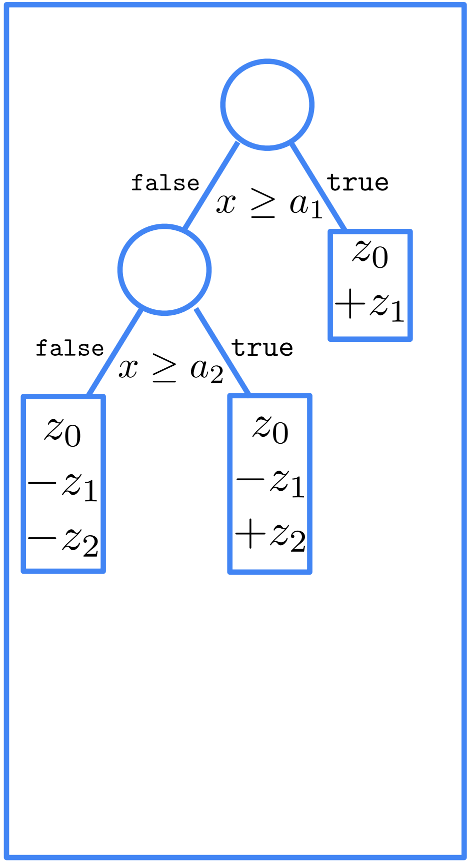
- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$

Equivalent representation



Convex boosting: which models ?

ModaBoost (Model-Adaptive Boosting)

↳ **Step 2**: introduce a new oracle ensuring the final emulates (is $\Leftrightarrow$ to) a specific *model architecture*

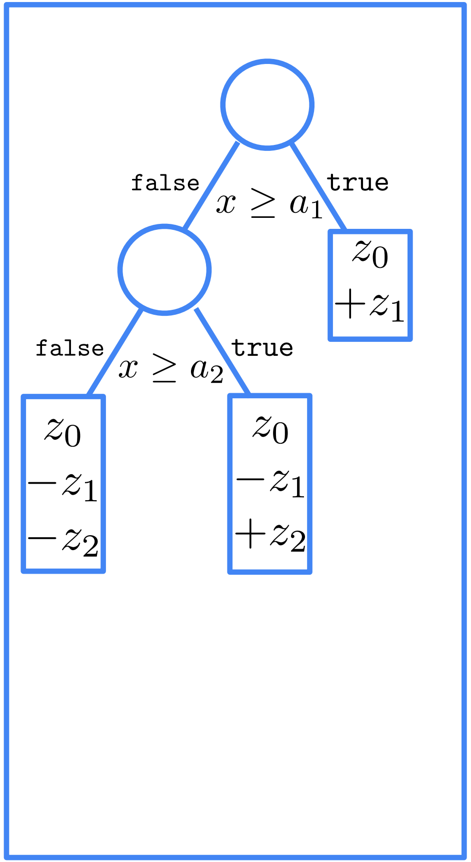
- Weights $\boldsymbol{w}$ = record of past performances
- Architecture Emulation Oracle : outputs $\mathcal{S} \subseteq \mathcal{X}$
- Weak learner : outputs hypotheses $h \in \mathbb{R}^{\mathcal{X}}$ at least $(\gamma > 0)$ different from random on $\mathcal{S}$

$$|\mathbb{E}_{\boldsymbol{w}_{|\mathcal{S}}}[y \cdot h(\boldsymbol{x})]| \geq \gamma$$

- Fits leveraging coefficients $\alpha \in \mathbb{R}$

↳ Returns model $H(\boldsymbol{x}) \doteq \sum_t 1_{\boldsymbol{x} \in \mathcal{S}_t} \alpha_t \cdot h_t(\boldsymbol{x})$

Decision Tree



Convex boosting with ModaBoost: which models ?

Decision Trees

Convex boosting with ModaBoost: which models ?

Decision Trees

Linear Separators

Convex boosting with ModaBoost: which models ?

Decision Trees

Linear Separators

Alternating Decision Trees

Convex boosting with ModaBoost: which models ?

Decision Trees

Linear Separators

Alternating Decision Trees

Nearest Neighbors

Convex boosting with ModaBoost: which models ?

Decision Trees

Linear Separators

Alternating Decision Trees

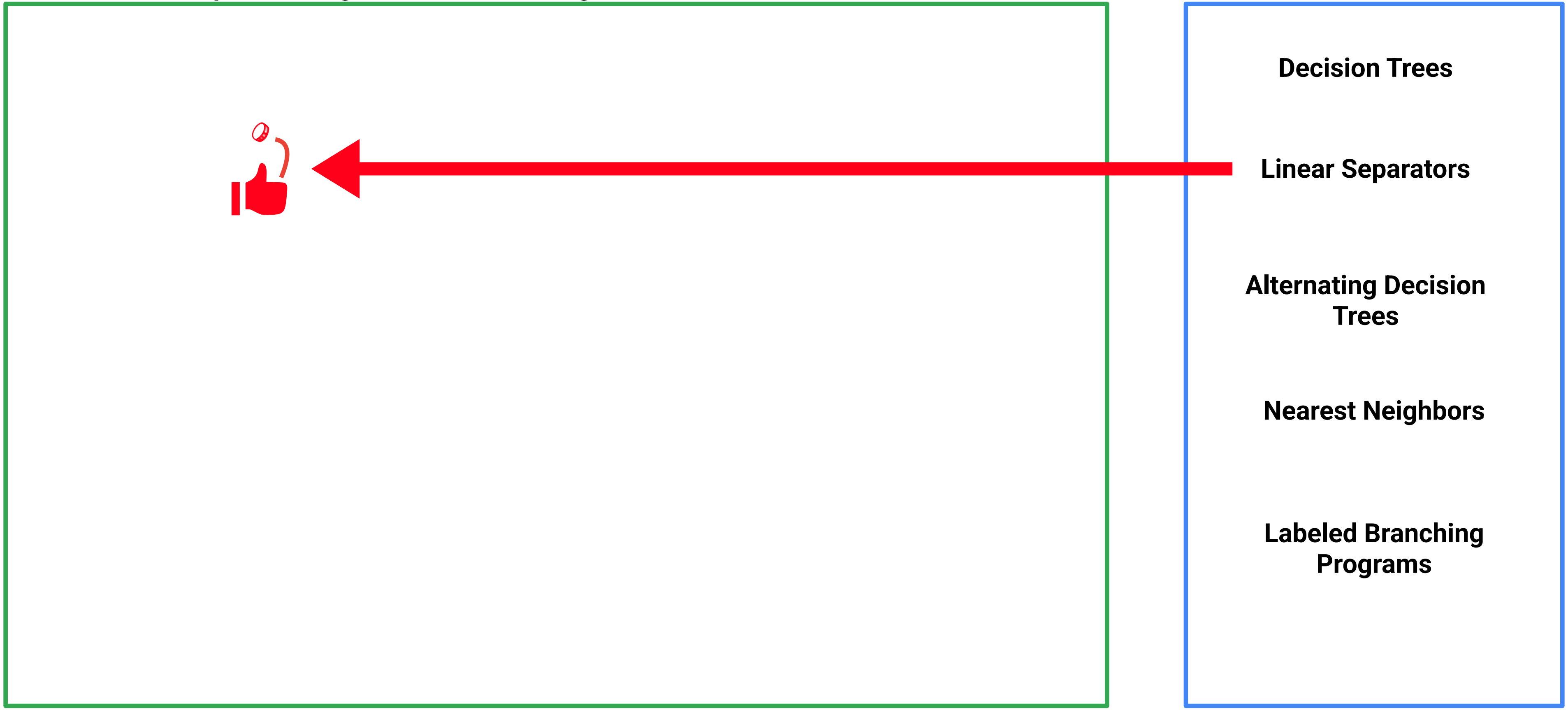
Nearest Neighbors

Labeled Branching Programs

...

Convex boosting with ModaBoost vs Long & Servedio

ModaBoost's output on Long & Servedio's setting



Convex boosting with ModaBoost vs Long & Servedio

ModaBoost's output on Long & Servedio's setting

Decision Trees

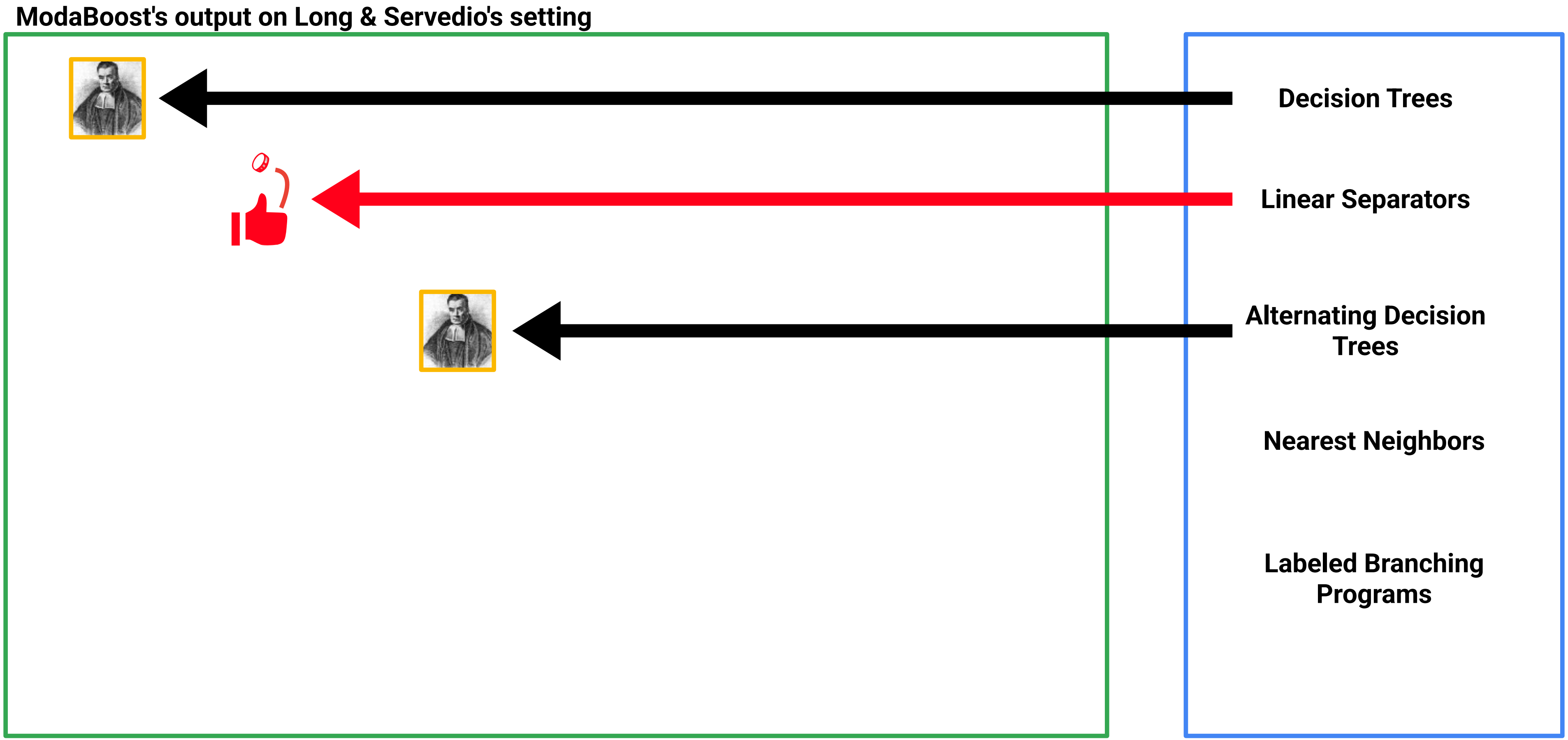
Linear Separators

Alternating Decision Trees

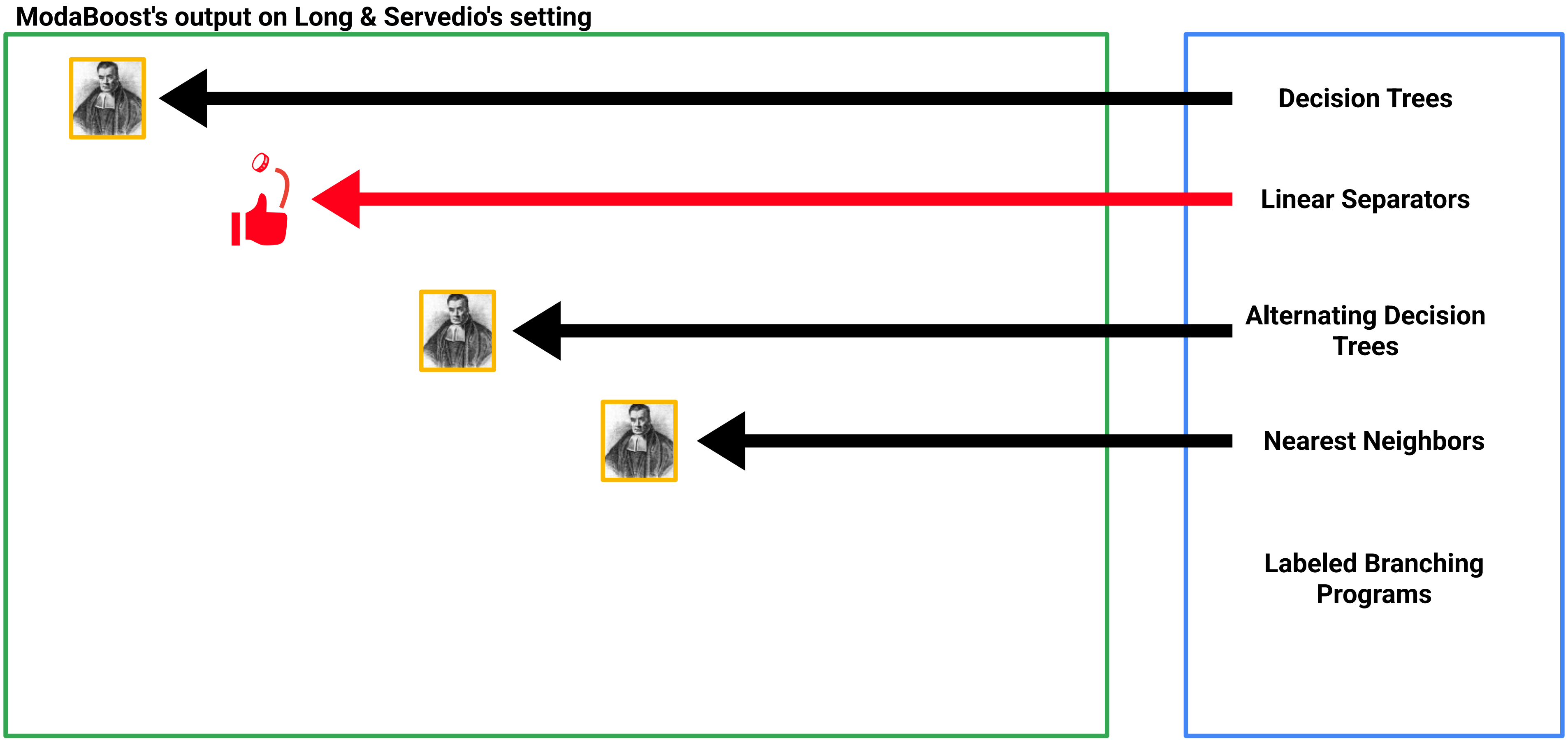
Nearest Neighbors

Labeled Branching Programs

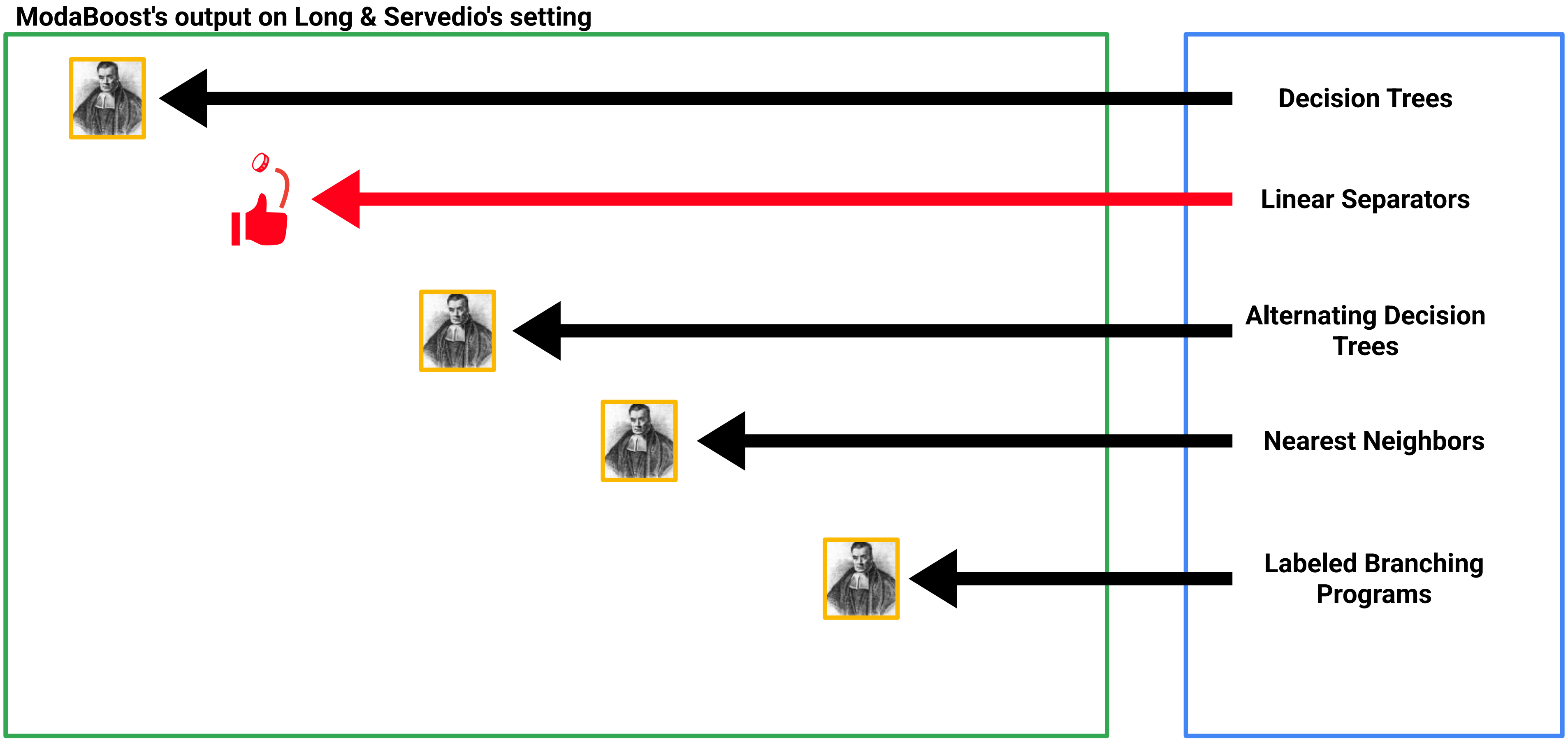
Convex boosting with ModaBoost vs Long & Servedio



Convex boosting with ModaBoost vs Long & Servedio

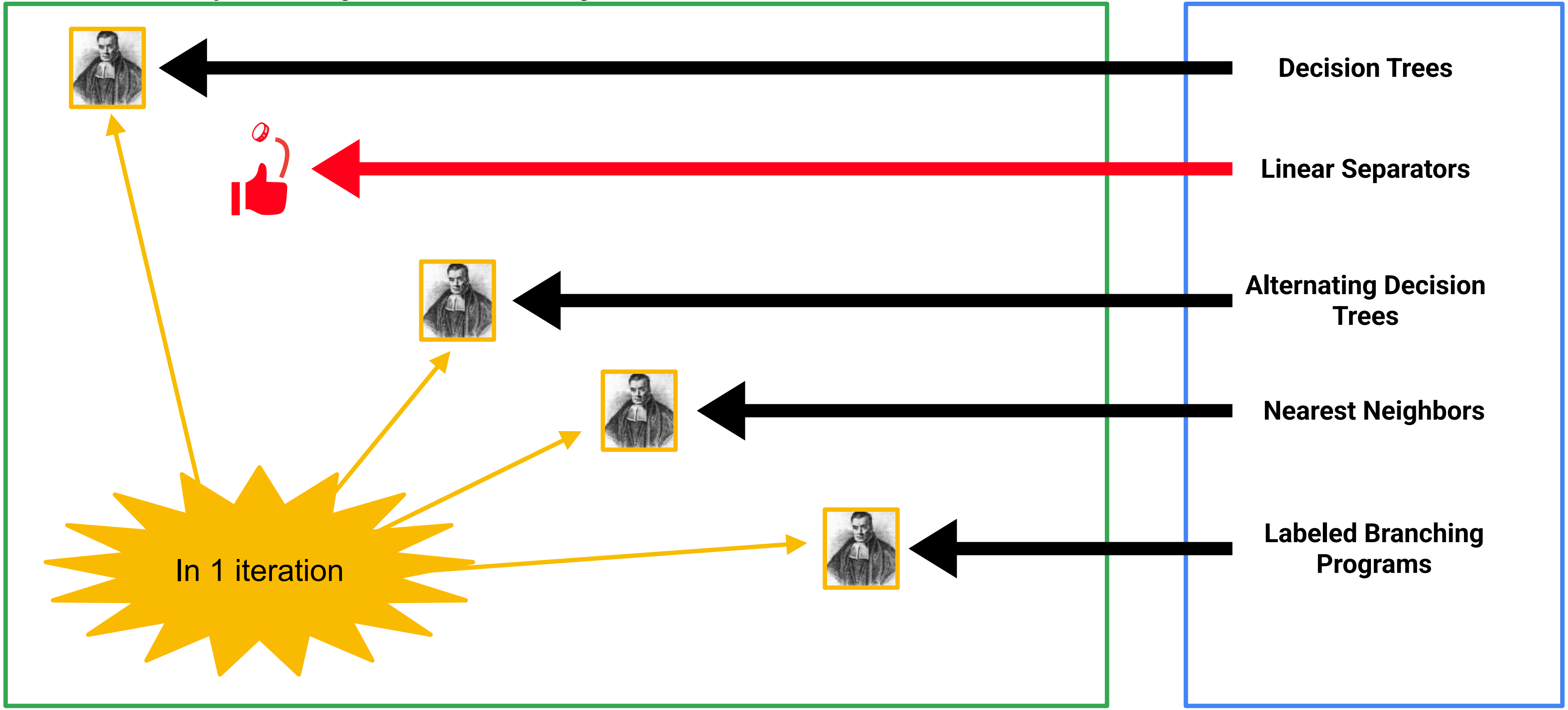


Convex boosting with ModaBoost vs Long & Servedio



Convex boosting with ModaBoost vs Long & Servedio

ModaBoost's output on Long & Servedio's setting



Conclusion

↳ Long and Servedio's paper has has a lasting impact on boosting / optimization

↳ Its impact should broaden on / shift to **models**, because it shows that

Linear Models can derail a whole ML pipeline otherwise optimal as soon as the "simplest" form of noise affects training data

↳ Suggests a broader question: given a class of models (more complex ?), what is its simplest "nemesis" noise model ?

Thank you !

Google Research