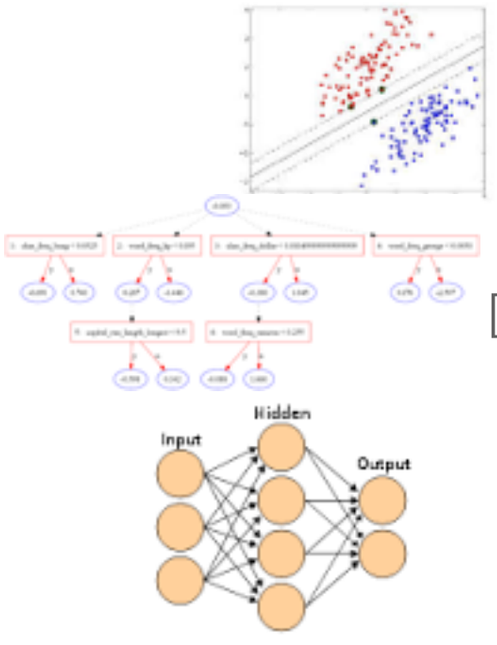
Aggregating Binary Classifiers Optimally with General Losses

Akshay Balsubramani and Yoav Freund, UC San Diego

Unlabeled data is **easier** to obtain than **labeled data**,
and contains useful information about labels that can be efficiently extracted
even when using many **"general" nonconvex losses**.

Ensemble aggregation scenario: we know
approximate classifier errors and predictions on unlabeled test set



SVM **err(SVM) ≈ 0.2**

Decision tree **err(DT) ≈ 0.25**

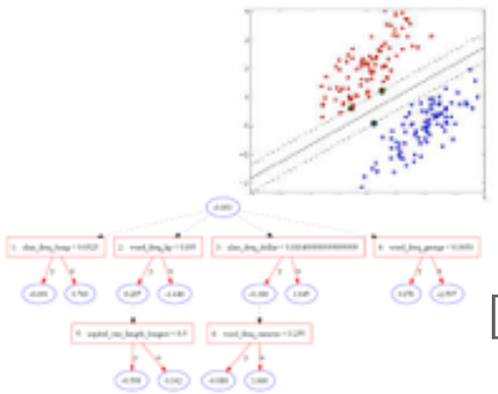
Deep net **err(NN) ≈ 0.1**

. . . . . .

x_1	x_2	x_3	…	x_n
+	-	+	. . .	-
+	-	-	. . .	+
-	-	+	. . .	-
…	…	…	…	…

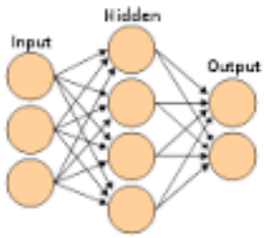
Aggregating Binary Classifiers Optimally with General Losses

Akshay Balsubramani and Yoav Freund, UC San Diego



SVM

Decision tree



Deep net

	x_1	x_2	x_3	…	x_n
err(SVM)	+	-	+	…	-
err(DT)	+	-	-	…	+
err(NN)	-	-	+	…	-

Ensemble Aggregator

g_1	g_2	g_3	…	g_n
-	+	+	…	-

randomized in [-1,1]

z_1	z_2	z_3	…	z_n
?	?	?	…	?

Goal: Minimize test loss $\frac{1}{n}\sum_{j=1}^{n}\ell(z_j, g_j)$

Formalization

(extends [Balsubramani and Freund, COLT 2015])

We predict **g*** to minimize worst-case loss,
given ensemble errors on unlabeled data.

$$\mathbf{g}^* = \arg \min_{\mathbf{g} \in [-1,1]^n} \left[\max_{\substack{\mathbf{z} \in [-1,1]^n, \\ \frac{1}{n}\mathbf{F}\mathbf{z} \geq \mathbf{b}}} \frac{1}{n} \sum_{j=1}^{n} \ell(z_j, g_j) \right]$$

By definition, this **g*** has an optimality property.

Minimax Guarantee

No predictor whatsoever has a better worst-case loss bound than **g***, using given info (**ensemble errors** and **unlabeled data**)

The Optimal Decision Rule

LEARNING

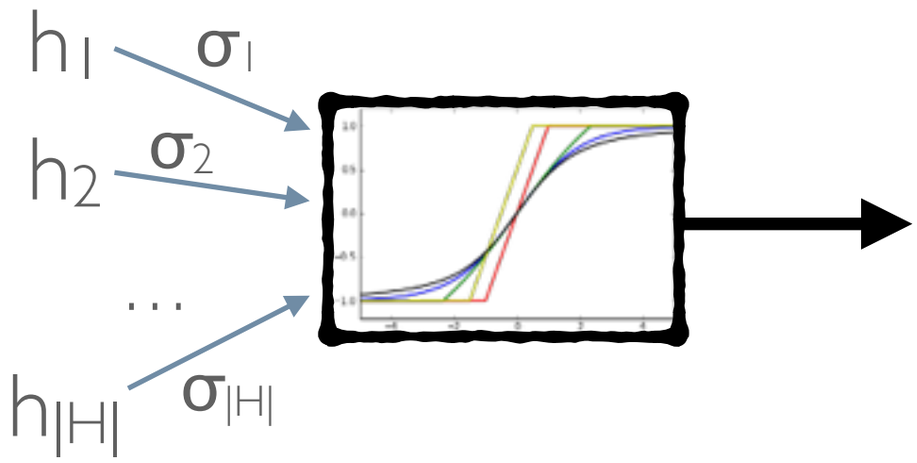
Optimize "slack function" (1-Lipschitz, **often convex**) to get weighting $\sigma \geq 0$ over hypotheses

$$\arg \min_{\sigma \geq 0^{|\mathcal{H}|}} \left[-\mathbf{b}^\top \sigma + \frac{1}{n} \sum_{j=1}^{n} \Psi(\mathbf{x}_j^\top \sigma) \right]$$

PREDICTION

On each test example, predict a **sigmoid (depends on loss)** over ensemble with weights σ

("neuron")

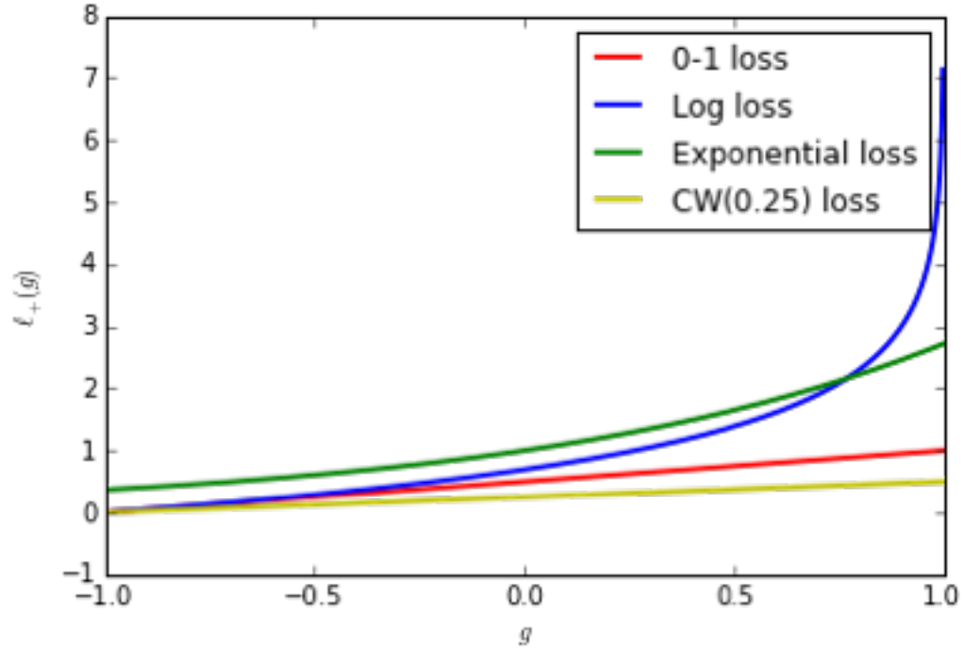


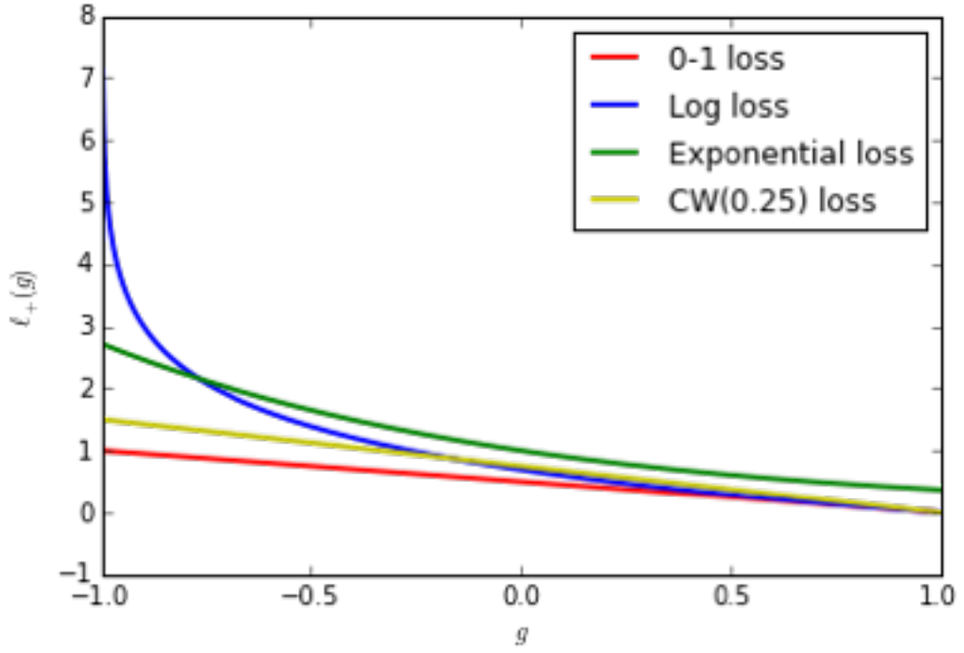
“General Losses”

Define partial losses $\ell_+(g) = \ell(1, g)$, $\ell_-(g) = \ell(-1, g)$ so that

$$\ell(z, g) = \left(\frac{1+z}{2}\right)\ell_+(g) + \left(\frac{1-z}{2}\right)\ell_-(g)$$

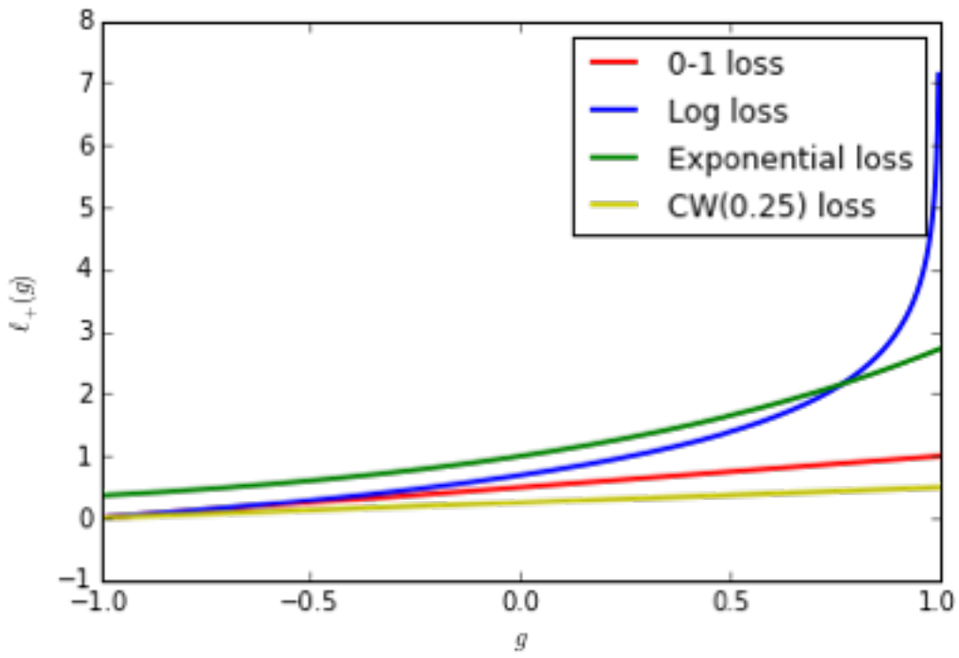
Partial losses are typically monotonic.

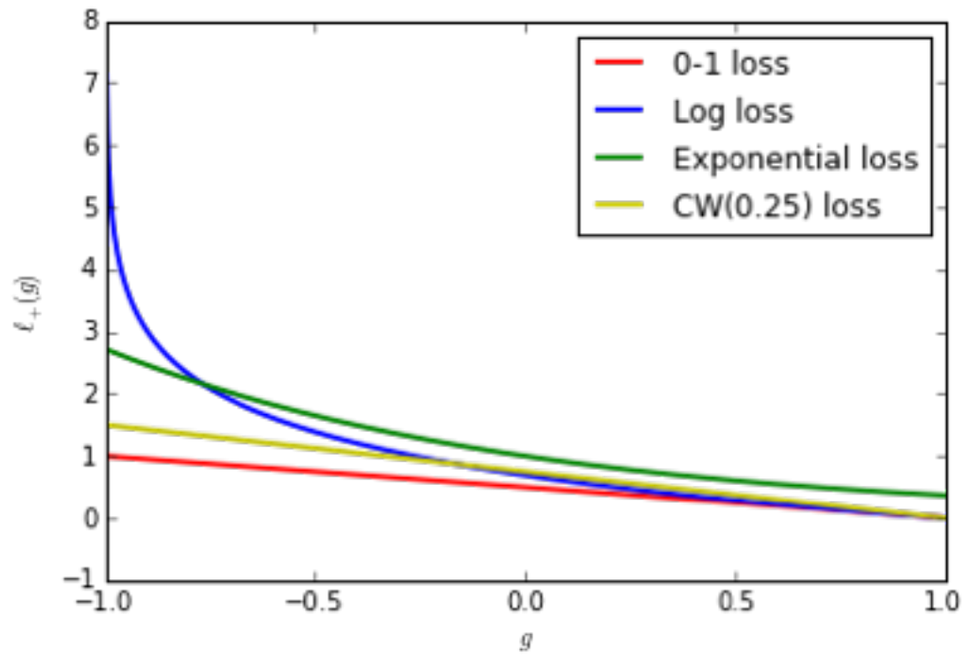




“General Losses”

Partial losses are typically monotonic.





Our predictor is minimax optimal whenever the partial losses are monotonic, as above.

Poster/paper: form of sigmoid + other computational/statistical results + extensions

Backup Slide: When is Learning Efficient?

LEARNING

Optimize "slack function" (1-Lipschitz, **often convex**) to get weighting $\sigma \geq 0$ over hypotheses

$$\arg \min_{\sigma \geq 0^{|\mathcal{H}|}} \left[-\mathbf{b}^\top \sigma + \frac{1}{n} \sum_{j=1}^{n} \Psi(\mathbf{x}_j^\top \sigma) \right]$$

Lemma 2. *The potential well $\Psi(m)$ is continuous and 1-Lipschitz. It is also convex under* any *of the following conditions:*

(A) The partial losses $\ell_\pm(\cdot)$ are convex over $(-1, 1)$.

(B) The loss function $\ell(\cdot, \cdot)$ is a proper loss.

(C) $\ell'_-(x)\ell''_+(x) \geq \ell''_-(x)\ell'_+(x)$ for all $x \in (-1, 1)$.

necessary/sufficient

$$\Psi(m) = \begin{cases} -m + 2\ell_-(-1) & \text{if} \quad m \leq \Gamma(-1) \\ \ell_+(\Gamma^{-1}(m)) + \ell_-(\Gamma^{-1}(m)) & \text{if} \quad m \in (\Gamma(-1), \Gamma(1)) \\ m + 2\ell_+(1) & \text{if} \quad m \geq \Gamma(1) \end{cases}$$