COS234:

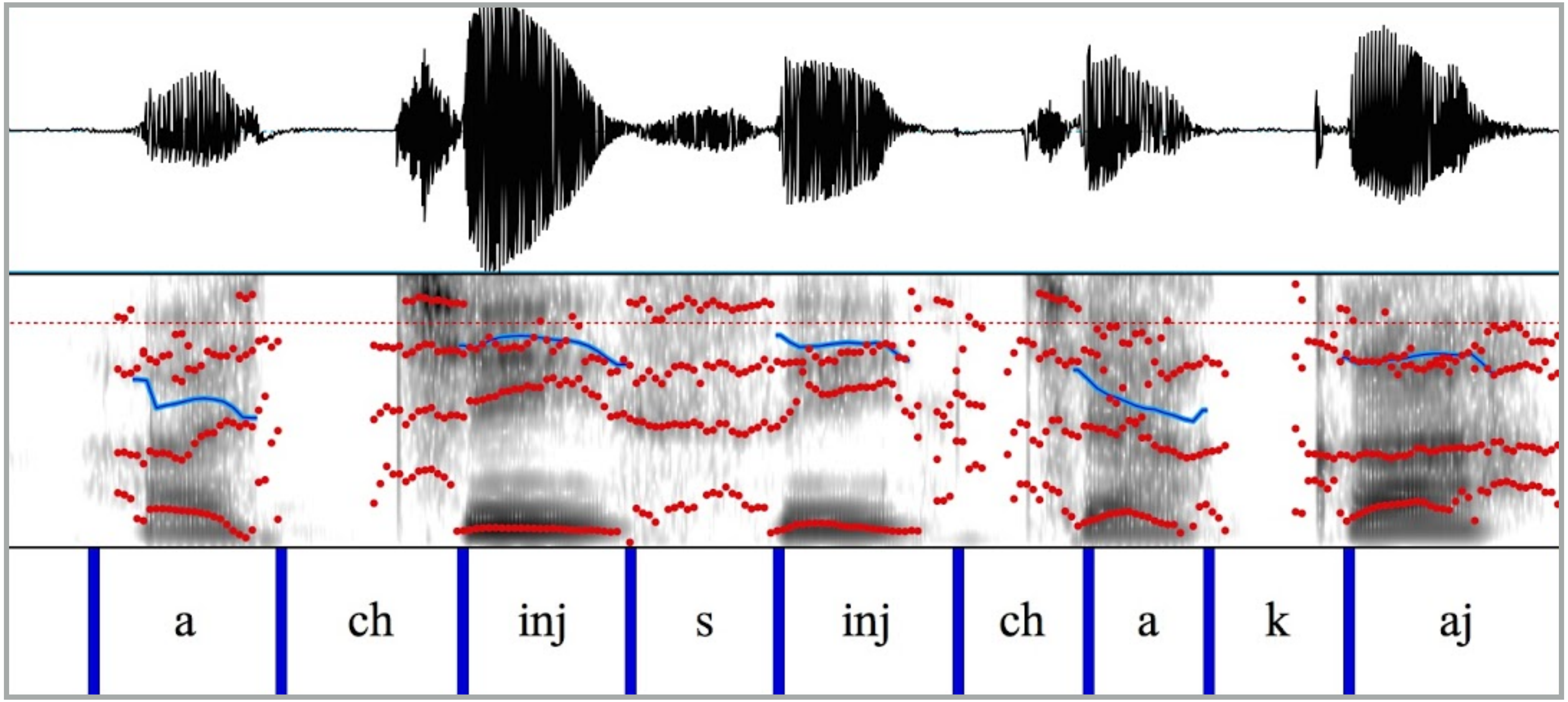
Introduction To Machine Learning

Prof. Yoram Singer

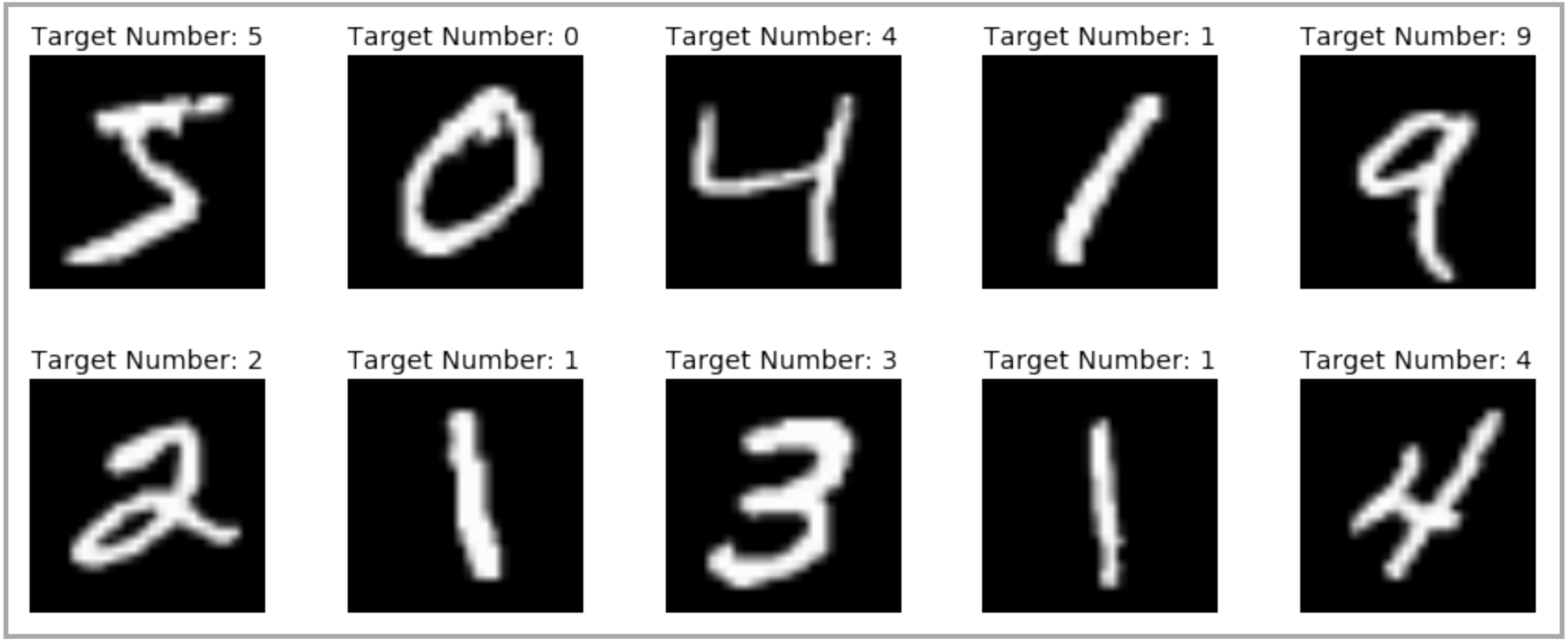
PRINCETON UNIVERSITY

Topic: Multiclass Learning

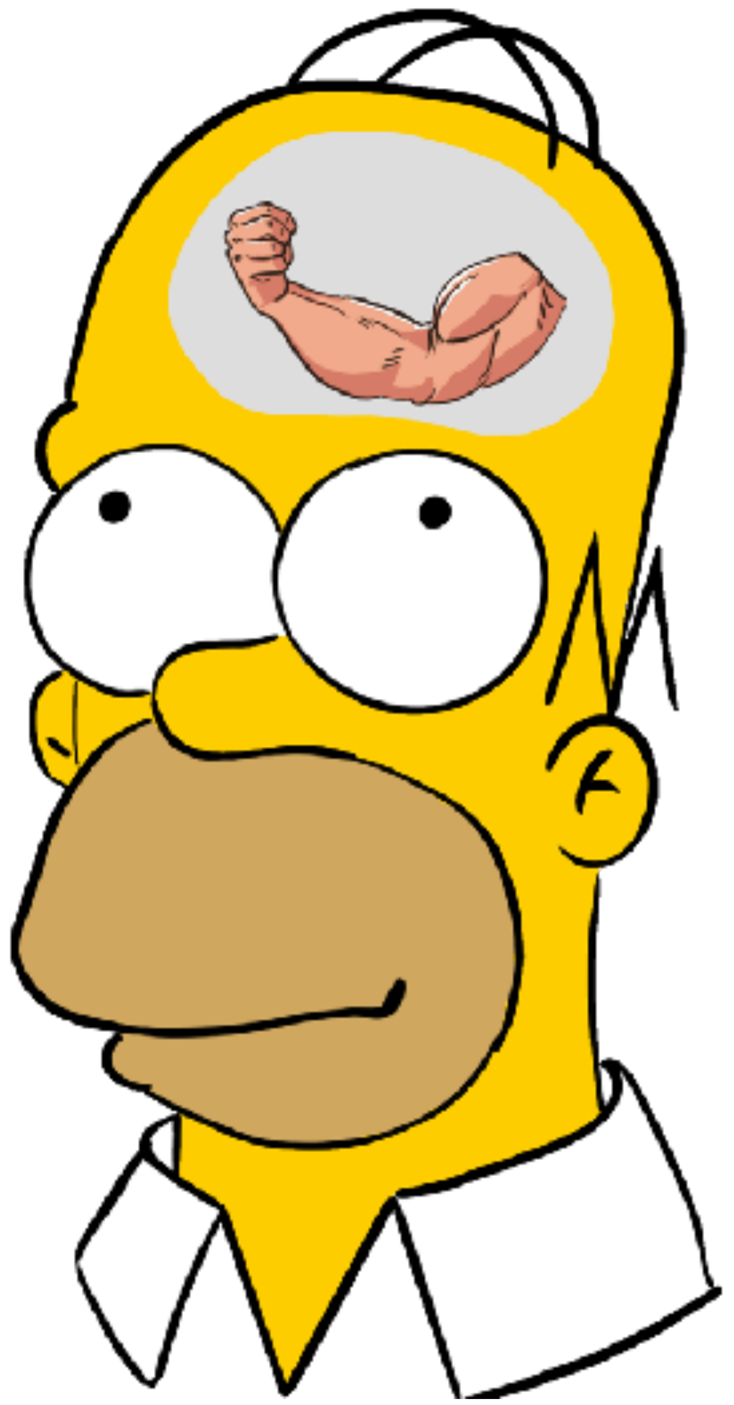
Phoneme Classification



Optical Character Recognition

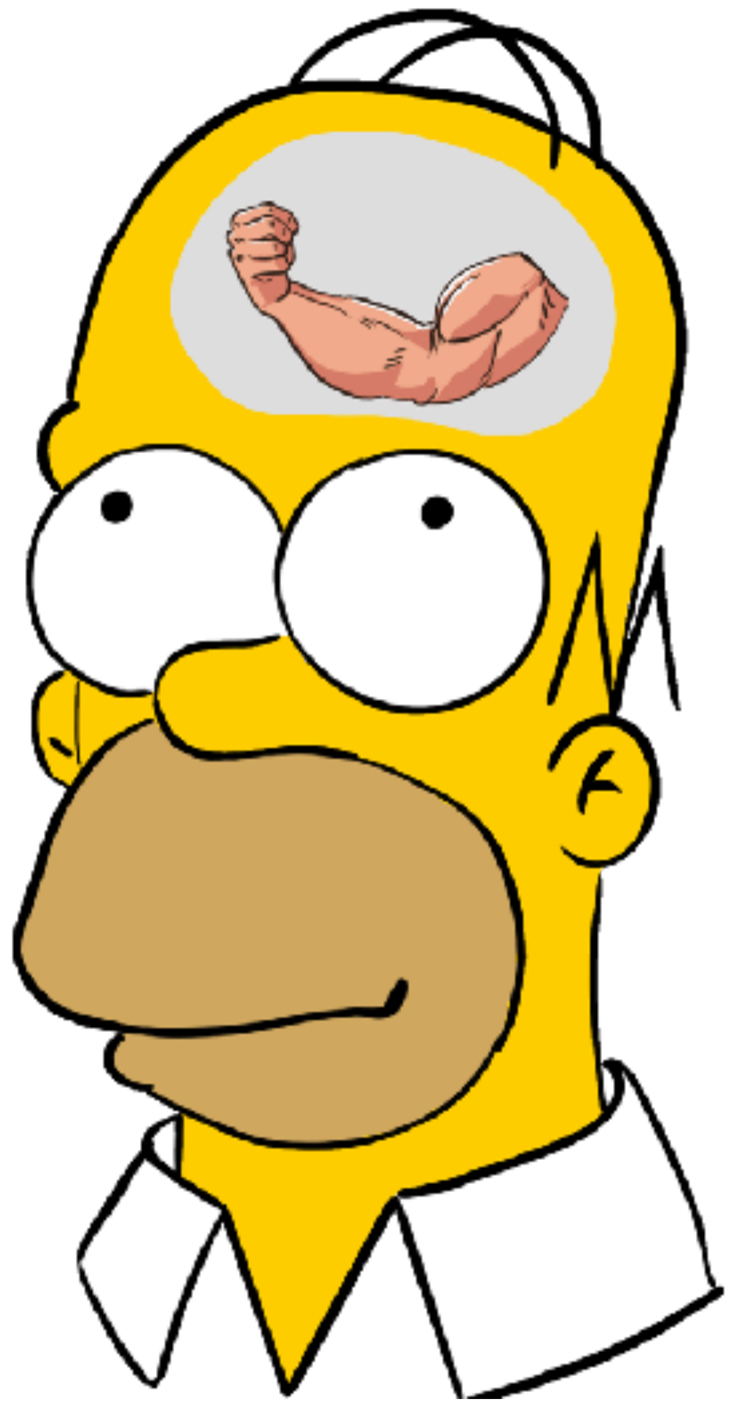


Multiclass: Problem Setting



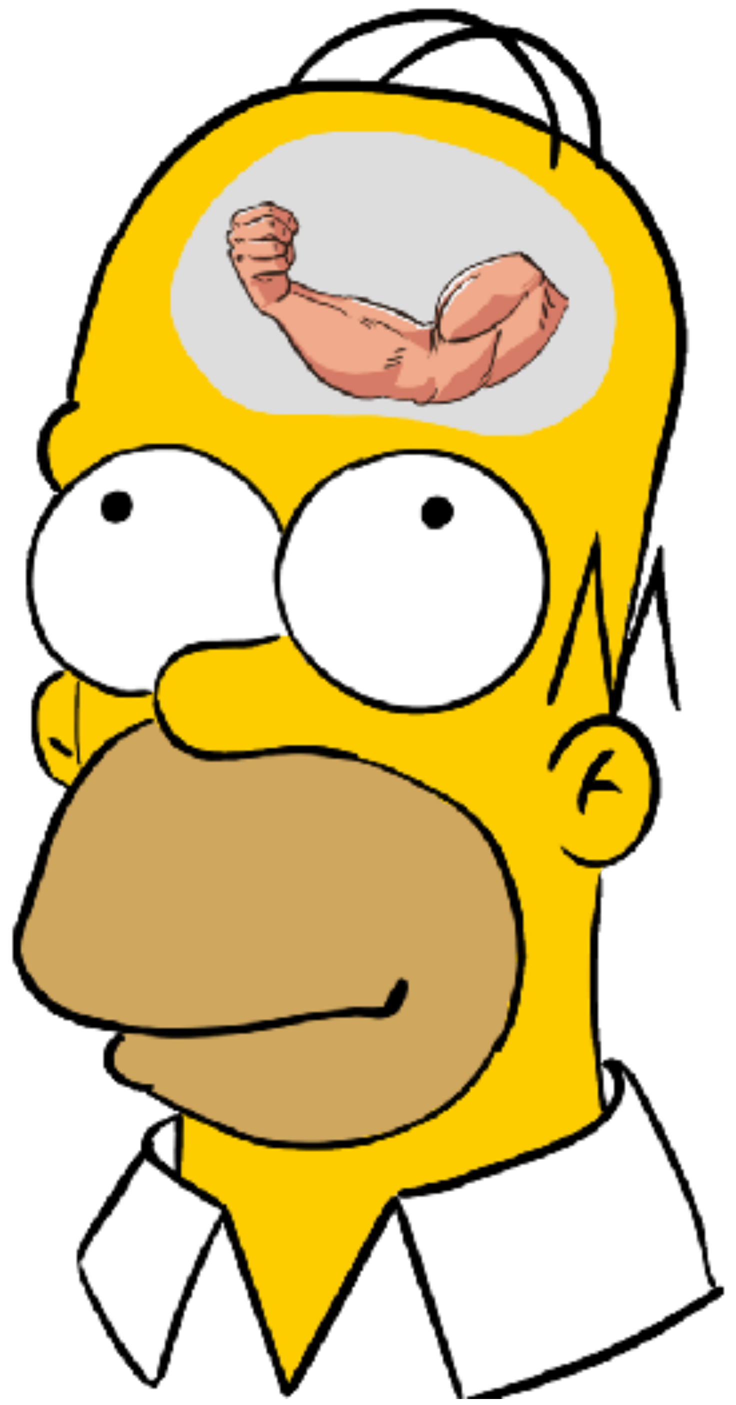
Multiclass: Problem Setting

- Instances: $\mathbf{x} \in \mathbf{R}^d$



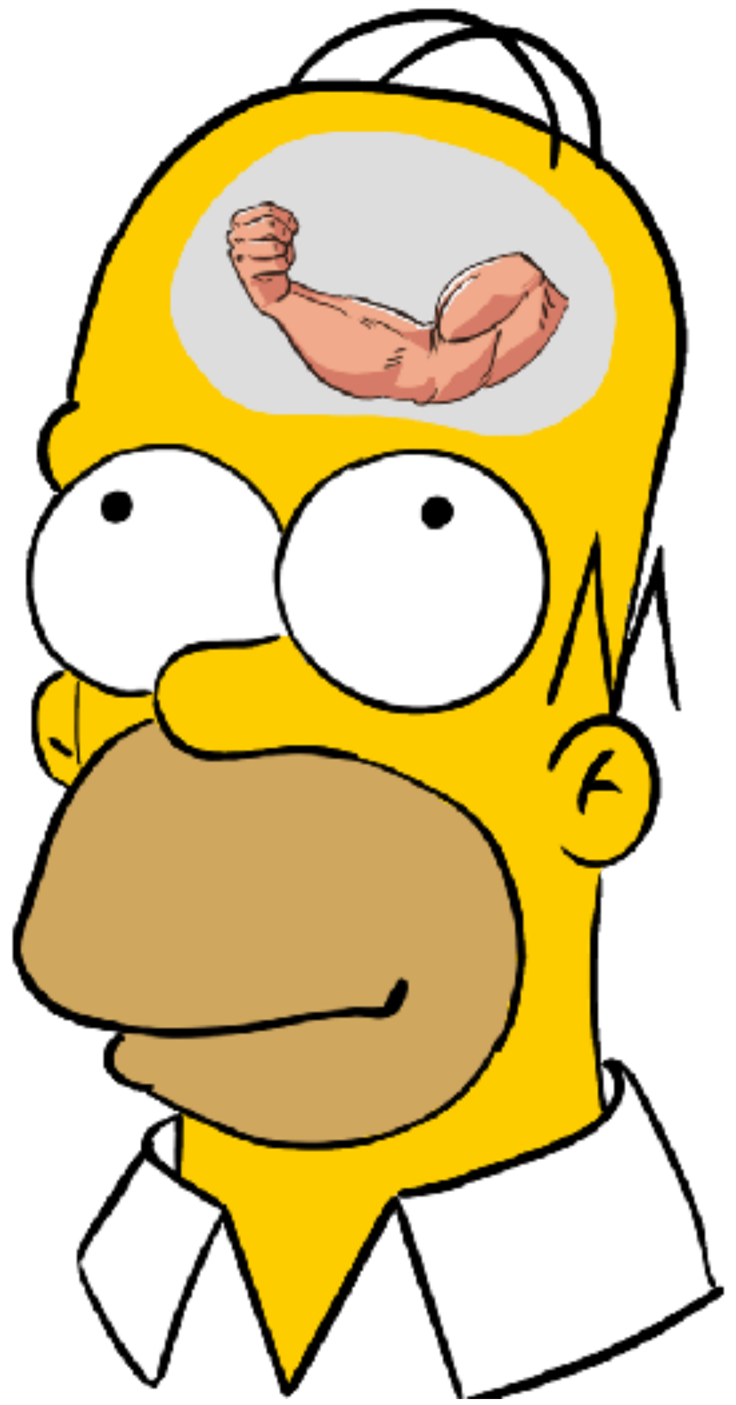
Multiclass: Problem Setting

- Instances: $\mathbf{x} \in \mathbf{R}^d$
- Labels: $y \in [k] = \{1, 2, \ldots, k\}$



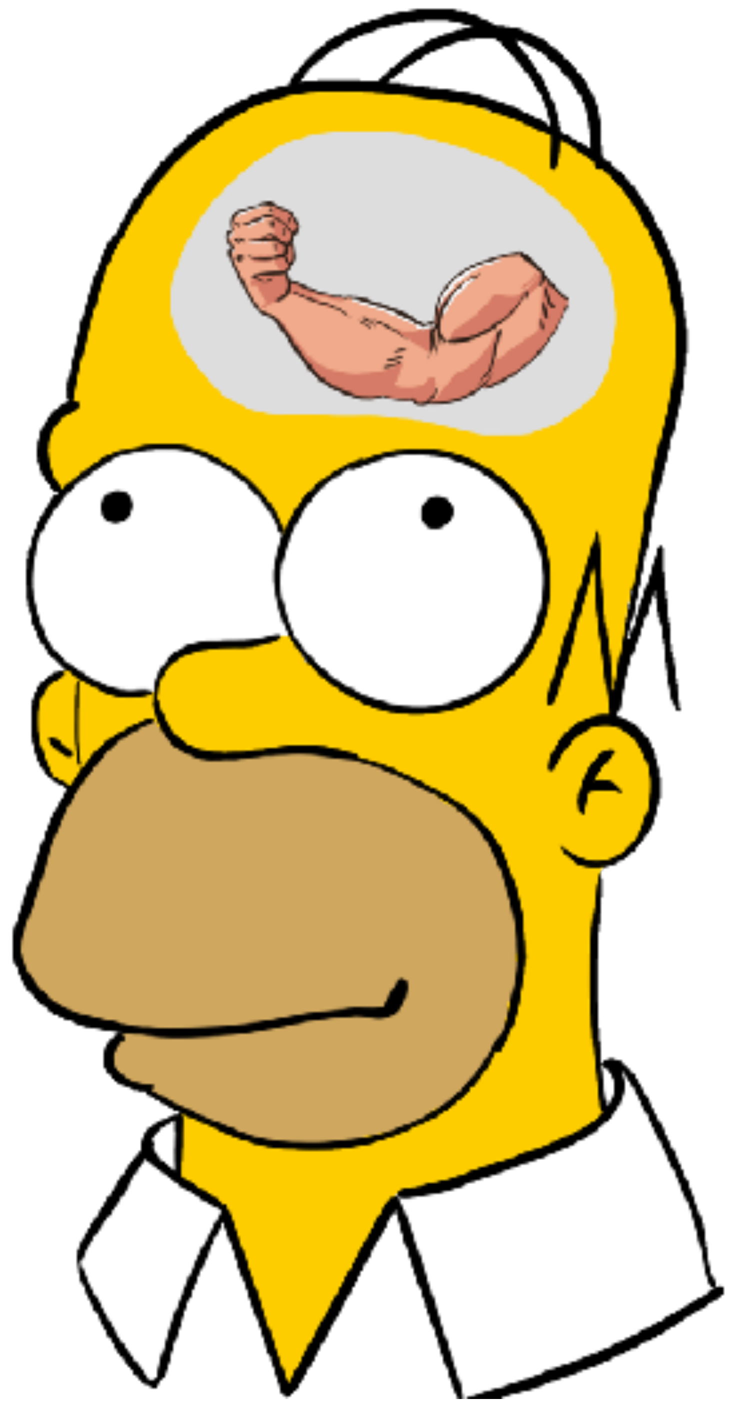
Multiclass: Problem Setting

- Instances: $\mathbf{x} \in \mathbf{R}^{\mathrm{d}}$
- Labels: $\mathrm{y} \in [\mathrm{k}] = \{1, 2, \ldots, \mathrm{k}\}$
- Multiclass predictor $\mathrm{h} : \mathbf{R}^{\mathrm{d}} \to [\mathrm{k}]$



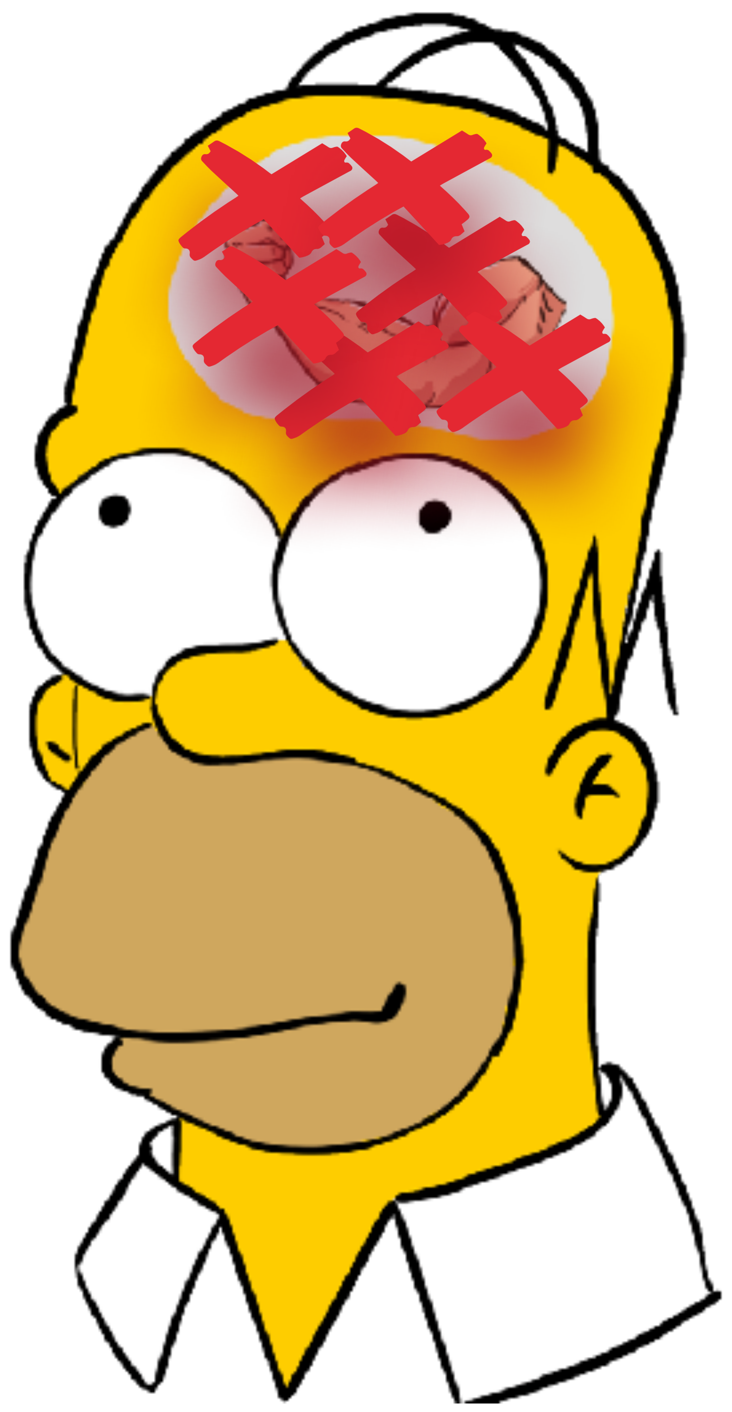
Multiclass: Problem Setting

- Instances: $\mathbf{x} \in \mathbf{R}^d$
- Labels: $y \in [k] = \{1, 2, \ldots, k\}$
- Multiclass predictor $h : \mathbf{R}^d \to [k]$
- Classification error / mistake: $\mathbf{1}\left[h(\mathbf{x}) \neq y\right]$



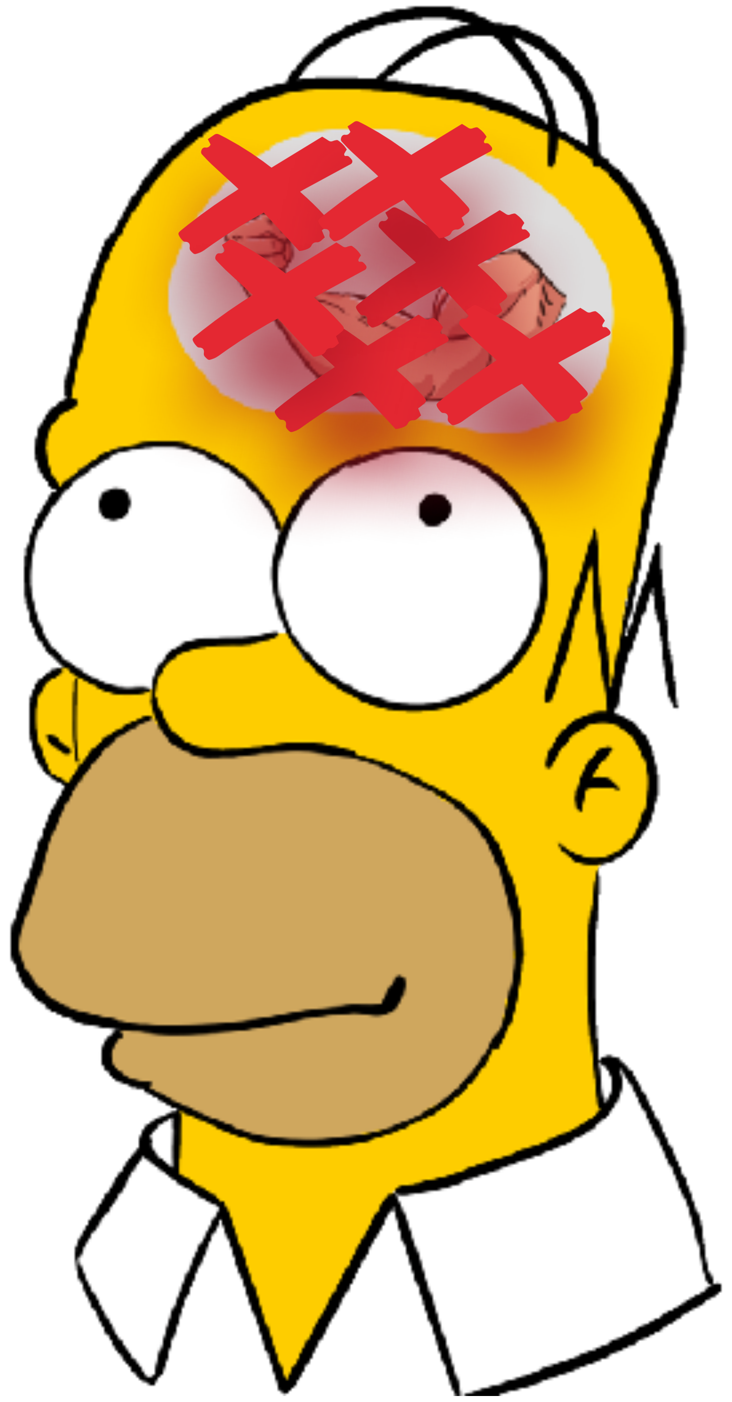
Multiclass: Problem Setting

- Instances: $\mathbf{x} \in \mathbf{R}^d$
- Labels: $y \in [k] = \{1, 2, \ldots, k\}$
- Multiclass predictor $h : \mathbf{R}^d \rightarrow [k]$
- Classification error / mistake: $\mathbf{1}\left[h(\mathbf{x}) \neq y\right]$



Multiclass: Problem Setting

- Instances: $\mathbf{x} \in \mathbf{R}^d$
- Labels: $y \in [k] = \{1, 2, \ldots, k\}$
- Multiclass predictor $h : \mathbf{R}^d \to [k]$
- Classification error / mistake: $\mathbf{1}\left[h(\mathbf{x}) \neq y\right]$
- As in binary case: minimizing prediction mistakes is NP-hard



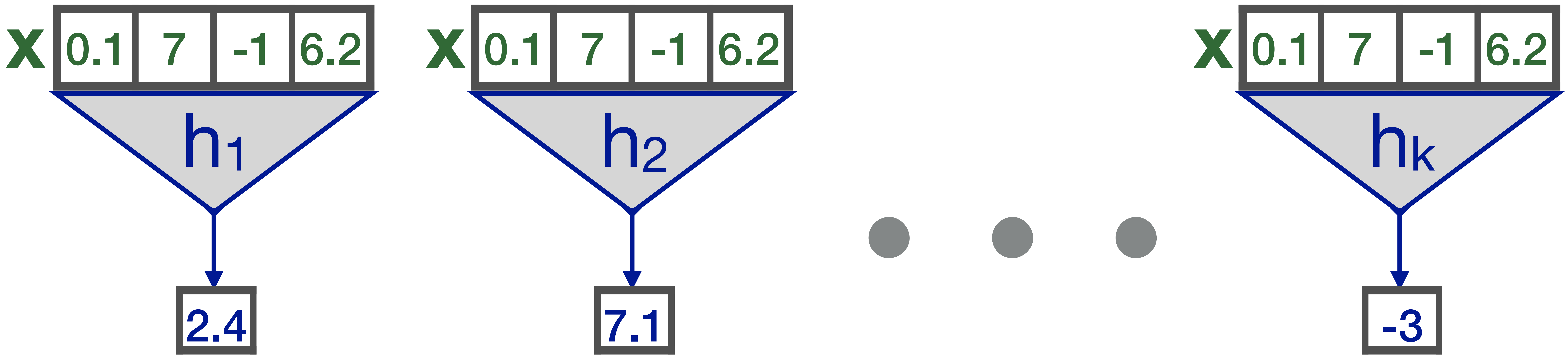
Prediction

I. Predictor h(**x**) can be a general function

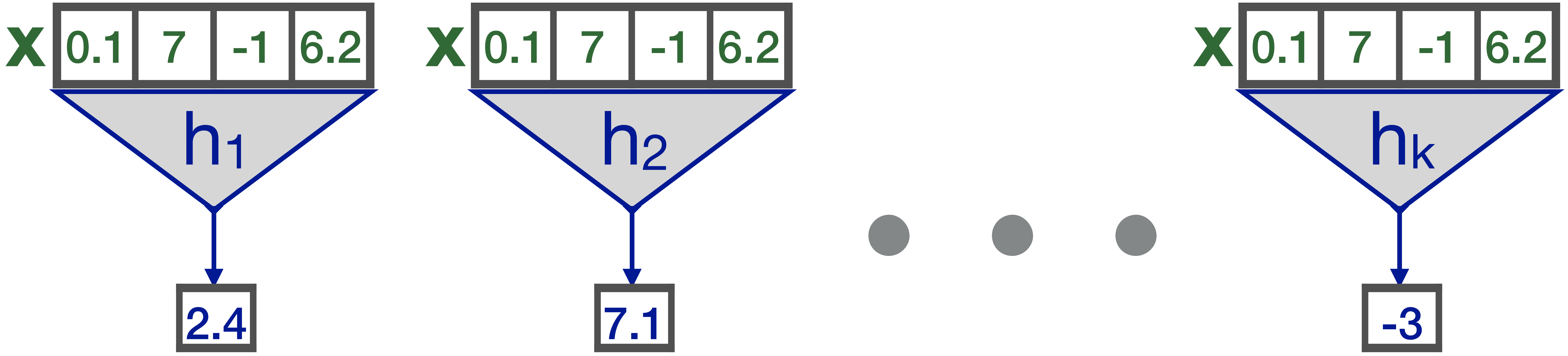
II. Need to express confidence in predicted class

Instead of $\mathbf{h} : \mathbf{R}^d \rightarrow [k]$ use $\mathbf{h} : \mathbf{R}^d \times [k] \rightarrow \mathbf{R}$:

where h(**x**,*c*) confidence that label of **x** is *c*

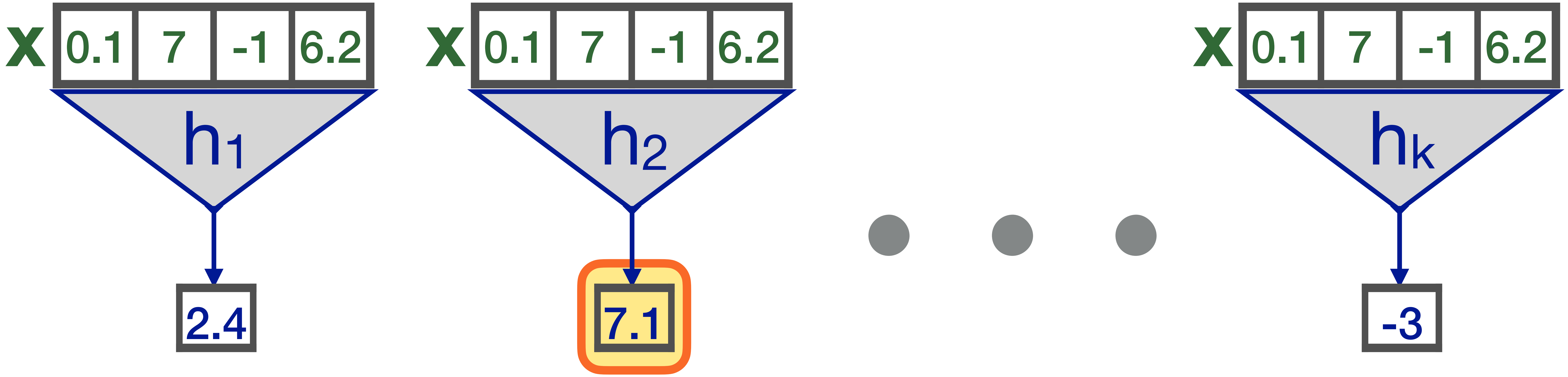


Winner Takes All



Predicted class: $\hat{y} = \arg\max_j \mathrm{h}_j(\mathbf{x})$

Winner Takes All



Predicted class: $\hat{y} = \arg\max_{j} h_j(\mathbf{x})$

Linear Multiclass Predictors

Linear predictor for class j : $h_j(\mathbf{x})$ is $\mathbf{w}_j \cdot \mathbf{x}$

Linear Multiclass Predictors

Linear predictor for class j : $h_j(\mathbf{x})$ is $\mathbf{w}_j \cdot \mathbf{x}$

Construct matrix W of size k × d whose j'th row is $\mathbf{w}_j$

$$W = \begin{bmatrix} -\mathbf{w}_1- \\ -\mathbf{w}_2- \\ \cdots \\ \cdots \\ \cdots \\ -\mathbf{w}_k- \end{bmatrix}$$

Linear Multiclass Predictors

Linear predictor for class j : $h_j(\mathbf{x})$ is $\mathbf{w}_j \cdot \mathbf{x}$

Construct matrix W of size k × d whose j'th row is $\mathbf{w}_j$

$$W = \begin{bmatrix} -\mathbf{w}_1- \\ -\mathbf{w}_2- \\ \cdots \\ \cdots \\ \cdots \\ -\mathbf{w}_k- \end{bmatrix}$$

Predicted scores: $\mathbf{z} = W\mathbf{x}$

$$\mathbf{z} = \begin{bmatrix} W_{11} & \cdots & W_{1d} \\ \vdots & \cdots & \vdots \\ W_{k1} & \cdots & W_{kd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

Linear Multiclass Predictors

Linear predictor for class j : $h_j(\mathbf{x})$ is $\mathbf{w}_j \cdot \mathbf{x}$

Construct matrix W of size k × d whose j'th row is $\mathbf{w}_j$

$$W = \begin{bmatrix} -\mathbf{w}_1- \\ -\mathbf{w}_2- \\ \cdots \\ \cdots \\ \cdots \\ -\mathbf{w}_k- \end{bmatrix}$$

Predicted scores: $\mathbf{z} = W\mathbf{x}$

$$\mathbf{z} = \begin{bmatrix} W_{11} & \cdots & W_{1d} \\ \vdots & \cdots & \vdots \\ W_{k1} & \cdots & W_{kd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

Predicted label: $\hat{y} = \arg\max_{j=1}^{k} z_j$

One vs. Rest (One vs. All)

- Learn **k** binary linear predictors
- **j**'th predictor distinguishes **j**'th class from the rest
- Learning scheme:

I. Transform $S \mapsto S_1, S_2, \ldots, S_k$ where $S_j = \left\{ \left(\mathbf{x}_i, (-1)^{\mathbf{1}[y_i \neq j]} \right) \right\}_{i=1}^{m}$

II. For j = 1,...,k learn a linear classifier $\mathbf{w}_j$ from S_j

- Inference: $\hat{y} = \arg\max_{j=1}^{k} z_j = \arg\max_{j=1}^{k} \mathbf{w}_j \cdot \mathbf{x}$

Example

Original training set: $S = \{(\mathbf{x}_1, 2), (\mathbf{x}_2, 4), (\mathbf{x}_3, 2), (\mathbf{x}_4, 3), (\mathbf{x}_5, 1)\}$

Results in four binary-labeled datasets:

S_1	S_2	S_3	S_4
$(\mathbf{x}_1, -)$	$(\mathbf{x}_1, +)$	$(\mathbf{x}_1, -)$	$(\mathbf{x}_1, -)$
$(\mathbf{x}_2, -)$	$(\mathbf{x}_2, -)$	$(\mathbf{x}_2, -)$	$(\mathbf{x}_2, +)$
$(\mathbf{x}_3, -)$	$(\mathbf{x}_3, +)$	$(\mathbf{x}_3, -)$	$(\mathbf{x}_3, -)$
$(\mathbf{x}_4, -)$	$(\mathbf{x}_4, -)$	$(\mathbf{x}_4, +)$	$(\mathbf{x}_4, -)$
$(\mathbf{x}_5, +)$	$(\mathbf{x}_5, -)$	$(\mathbf{x}_5, -)$	$(\mathbf{x}_5, -)$

Deficiencies of OvA

Deficiencies of OvA

Predictors are trained independently and then used jointly

Deficiencies of OvA

Predictors are trained independently and then used jointly

Albeit trained independently “compete” during inference

Deficiencies of OvA

Predictors are trained independently and then used jointly

Albeit trained independently “compete” during inference

Resulting binary problems might be overly difficult

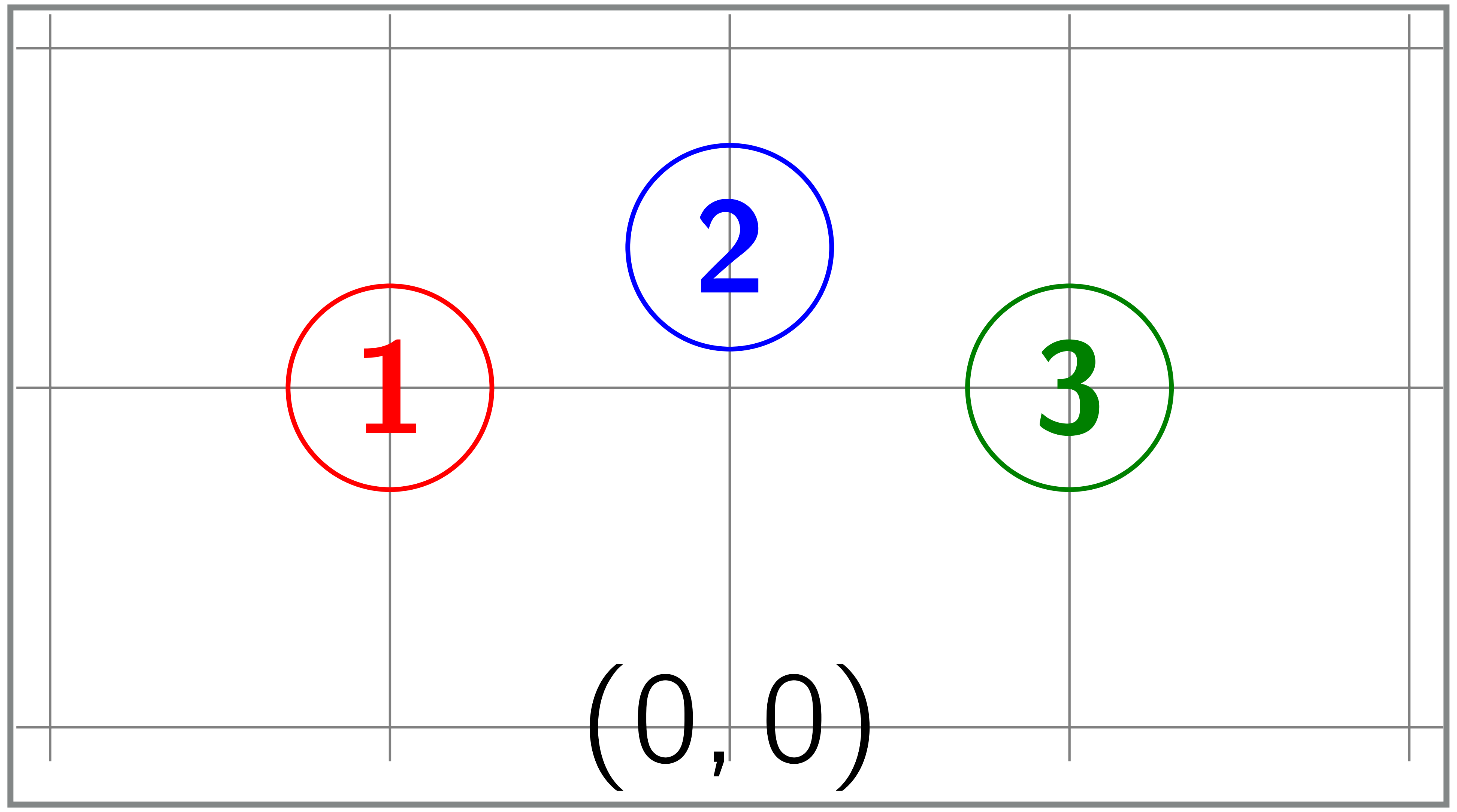
Deficiencies of OvA

Predictors are trained independently and then used jointly

Albeit trained independently “compete” during inference

Resulting binary problems might be overly difficult

OvA would fail for setting:



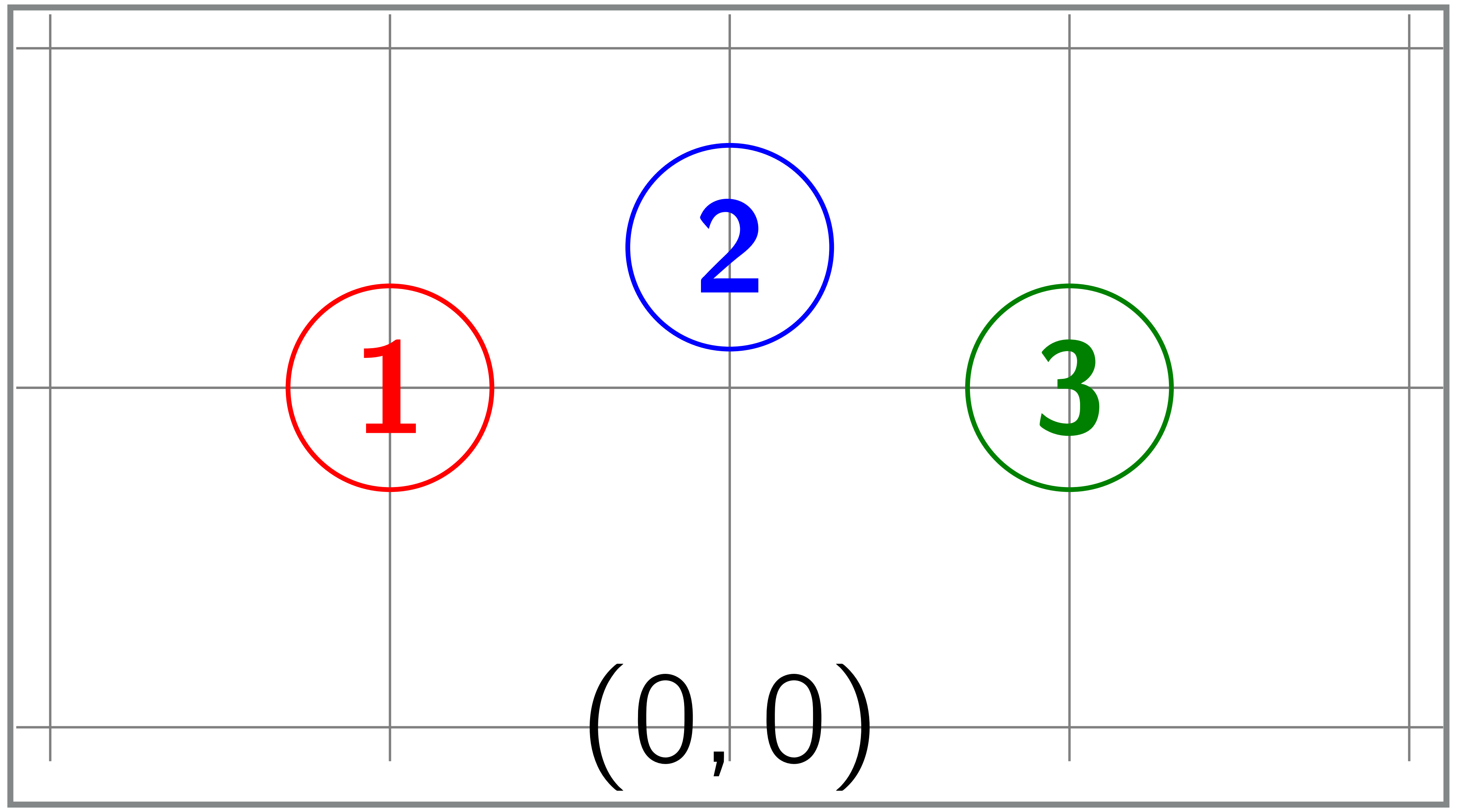
Deficiencies of OvA

Predictors are trained independently and then used jointly

Albeit trained independently “compete” during inference

Resulting binary problems might be overly difficult

OvA would fail for setting:



While data is linearly separable using:

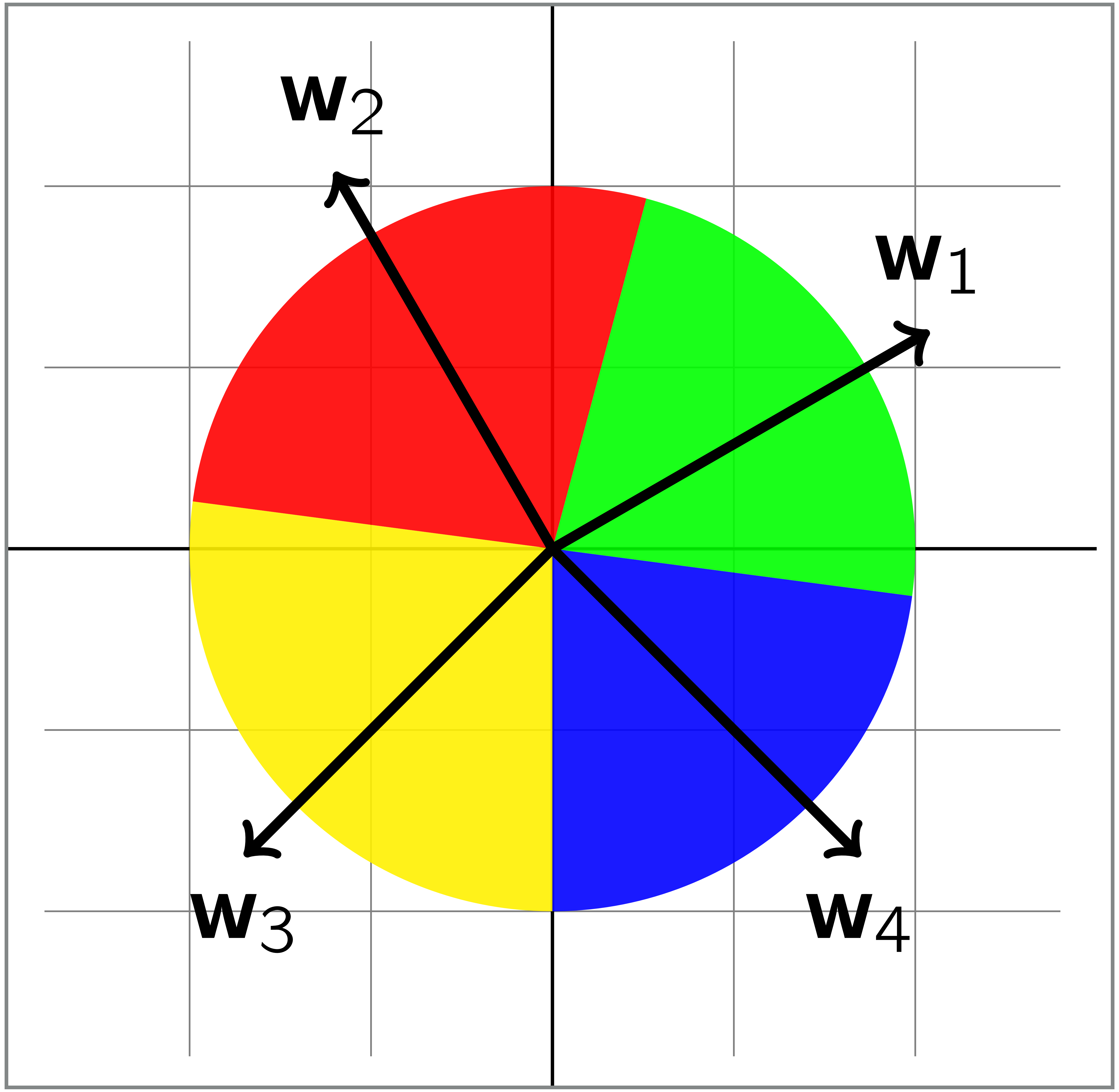
$$W = \begin{bmatrix} -1 & 1 \\ 0 & \sqrt{2} \\ 1 & 1 \end{bmatrix}$$

Multiclass Margin

$$W = \begin{bmatrix} -\mathbf{w}_1- \\ -\mathbf{w}_2- \\ -\mathbf{w}_3- \\ -\mathbf{w}_4- \end{bmatrix} \in \mathbf{R}^{4x2}$$

Assume : $\|\mathbf{w}_j\| = 1 \quad \|\mathbf{x}\| = 1$

$$\measuredangle(\mathbf{w}, \mathbf{x}) = \cos^{-1}(\mathbf{w} \cdot \mathbf{x})$$

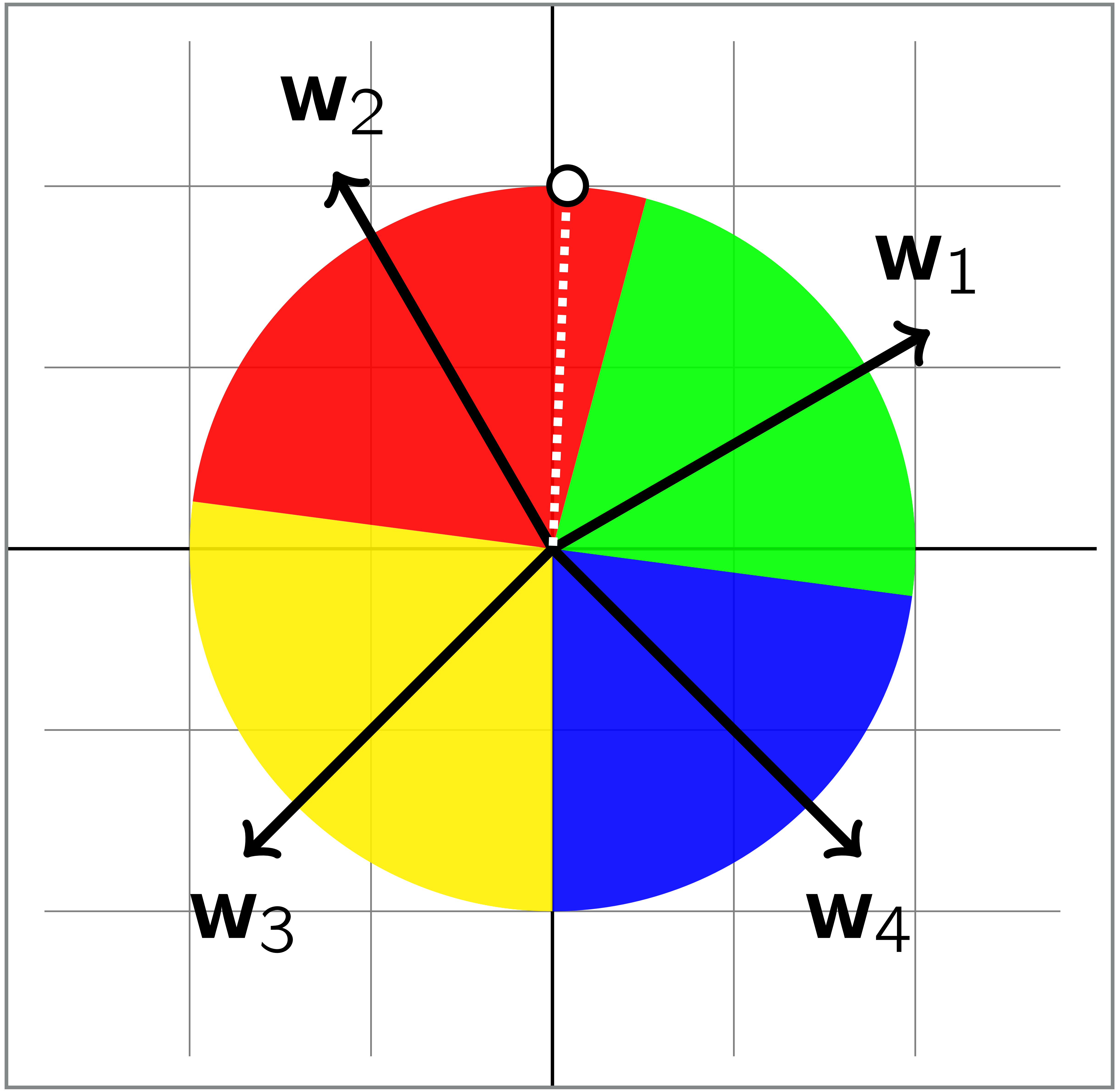


Multiclass Margin

$$W = \begin{bmatrix} -\mathbf{w}_1- \\ -\mathbf{w}_2- \\ -\mathbf{w}_3- \\ -\mathbf{w}_4- \end{bmatrix} \in \mathbf{R}^{4x2}$$

Assume : $\|\mathbf{w}_j\| = 1 \quad \|\mathbf{x}\| = 1$

$$\measuredangle(\mathbf{w}, \mathbf{x}) = \cos^{-1}(\mathbf{w} \cdot \mathbf{x})$$



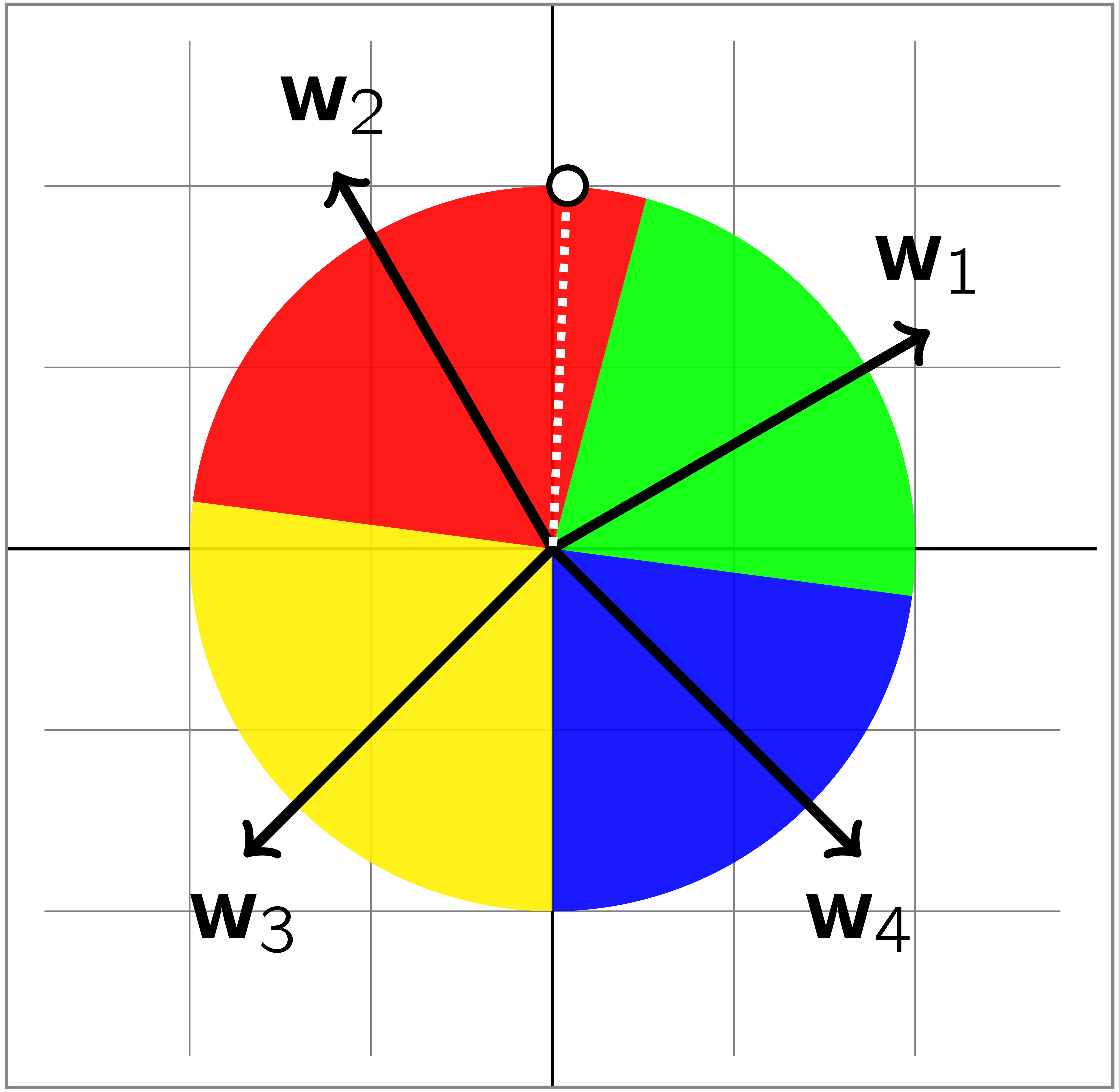
Multiclass Margin

$$W = \begin{bmatrix} -\mathbf{w}_1- \\ -\mathbf{w}_2- \\ -\mathbf{w}_3- \\ -\mathbf{w}_4- \end{bmatrix} \in \mathbf{R}^{4x2}$$

Assume : $\|\mathbf{w}_\mathbf{j}\| = 1 \quad \|\mathbf{x}\| = 1$

$$\measuredangle(\mathbf{w}, \mathbf{x}) = \cos^{-1}(\mathbf{w} \cdot \mathbf{x})$$

$$\measuredangle(\mathbf{w_2}, \mathbf{x}) < \measuredangle(\mathbf{w_j}, \mathbf{x})$$



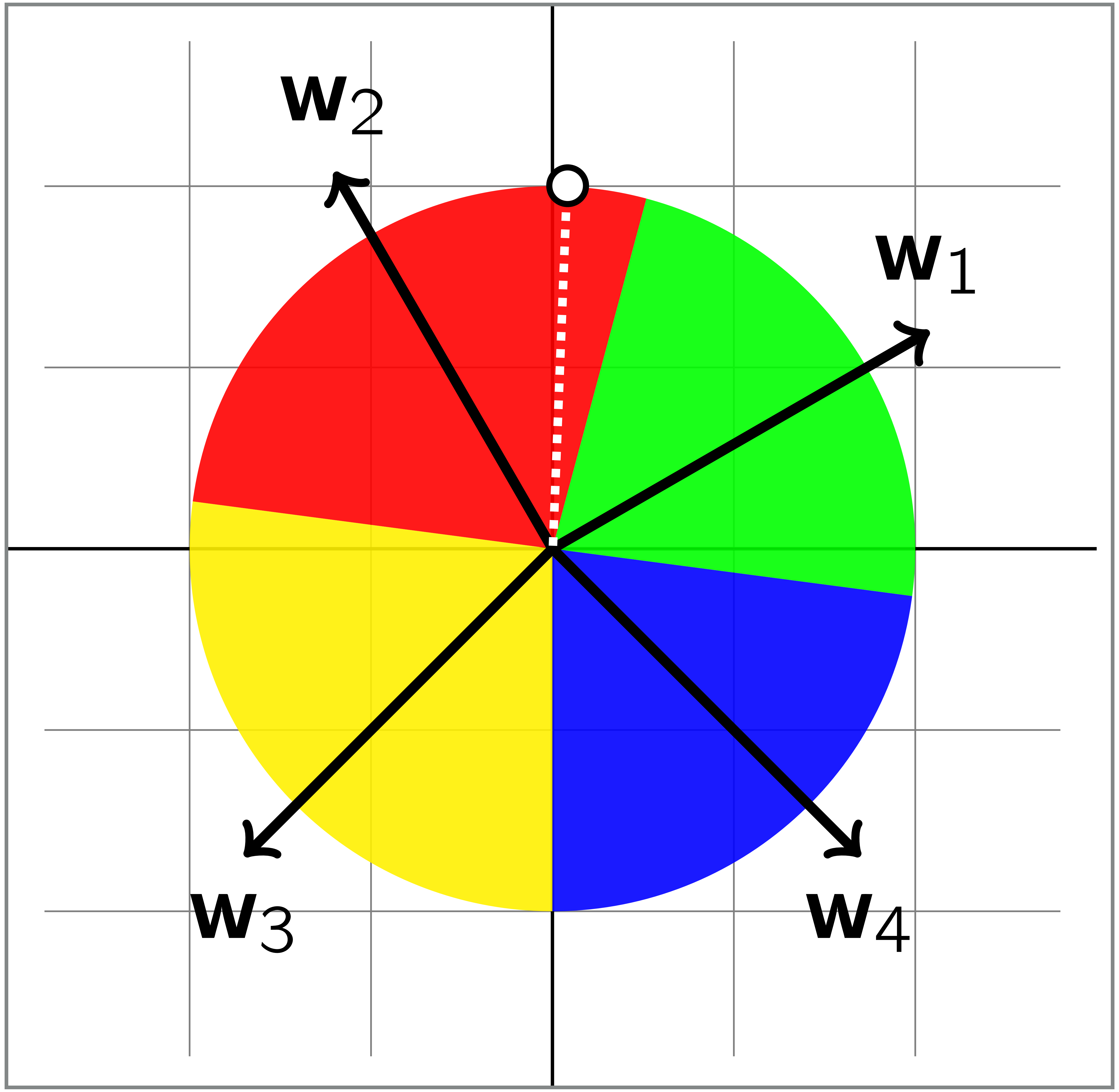
Multiclass Margin

$$W = \begin{bmatrix} -\mathbf{w}_1- \\ -\mathbf{w}_2- \\ -\mathbf{w}_3- \\ -\mathbf{w}_4- \end{bmatrix} \in \mathbf{R}^{4x2}$$

Assume : $\|\mathbf{w}_j\| = 1 \quad \|\mathbf{x}\| = 1$

$$\measuredangle(\mathbf{w}, \mathbf{x}) = \cos^{-1}(\mathbf{w} \cdot \mathbf{x})$$

$$\measuredangle(\mathbf{w}_2, \mathbf{x}) < \measuredangle(\mathbf{w}_j, \mathbf{x}) \quad \Rightarrow \quad \hat{y} = 2$$



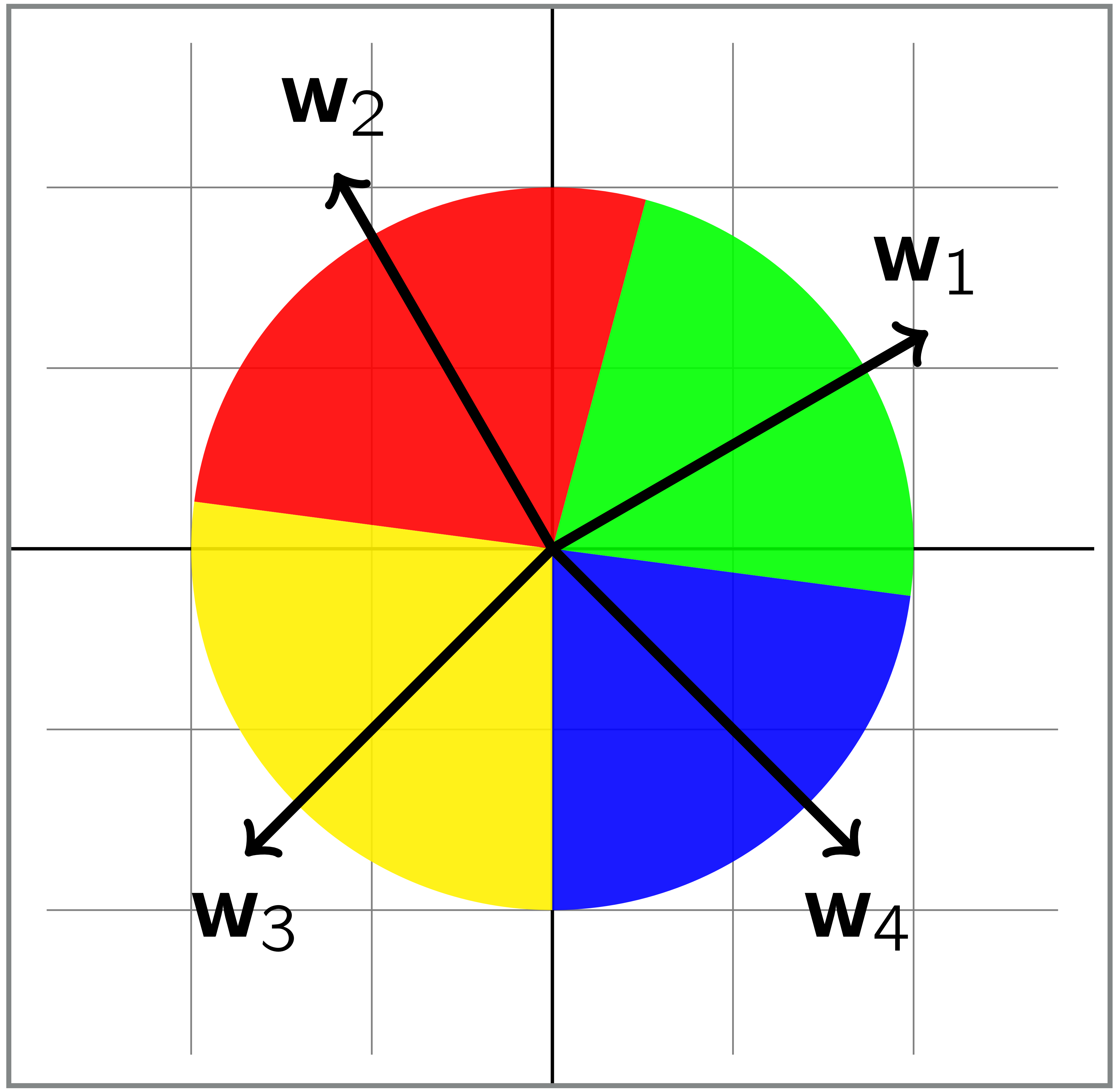
Multiclass Margin

For general vectors impose:

$$(\mathbf{x}, \mathrm{y}) \quad \Rightarrow \quad \forall \mathrm{j} \neq \mathrm{y} : \ \mathbf{w}_{\mathrm{y}} \cdot \mathbf{x} > \mathbf{w}_{\mathrm{j}} \cdot \mathbf{x}$$

In matrix-vector format:

$$(\mathbf{x}, \mathrm{y}) \quad \Rightarrow \quad \forall \mathrm{j} \neq \mathrm{y} : \ [\mathrm{W}\mathbf{x}]_{\mathrm{y}} > [\mathrm{W}\mathbf{x}]_{\mathrm{j}}$$



Margin Loss

Predicted class: $\hat{y}(\mathbf{z}) = \arg\max_{j=1}^{k} z_j$

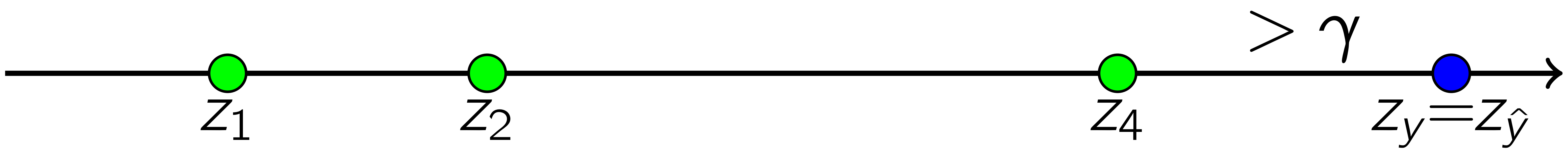
Classification error:

$$\ell^{MC}(\mathbf{z}) = \mathbf{1}\left[\hat{y}(\mathbf{z}) \neq y\right]$$

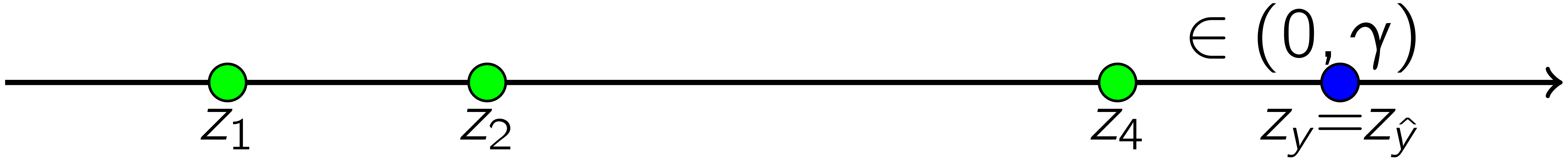
Max-Margin Loss is difference in scores + penalty γ:

$$\ell^{MM}(\mathbf{z}) = \left[\gamma + \max_{j \neq y} z_j - z_y\right]_+ \qquad \text{where} \quad [z]_+ = \max\{0, z\}$$

Margin great than $\gamma \;\Rightarrow\; \ell^{\mathrm{MC}} = \ell^{\mathrm{MM}} = 0$



Margin $\in (0, \gamma) \;\Rightarrow\; \ell^{\mathrm{MC}} = 0$ but $\ell^{\mathrm{MM}} \geq 0$



Margin $< 0 \;\Rightarrow\; \ell^{\mathrm{MC}} = 1$ and $\ell^{\mathrm{MM}} \geq \gamma$

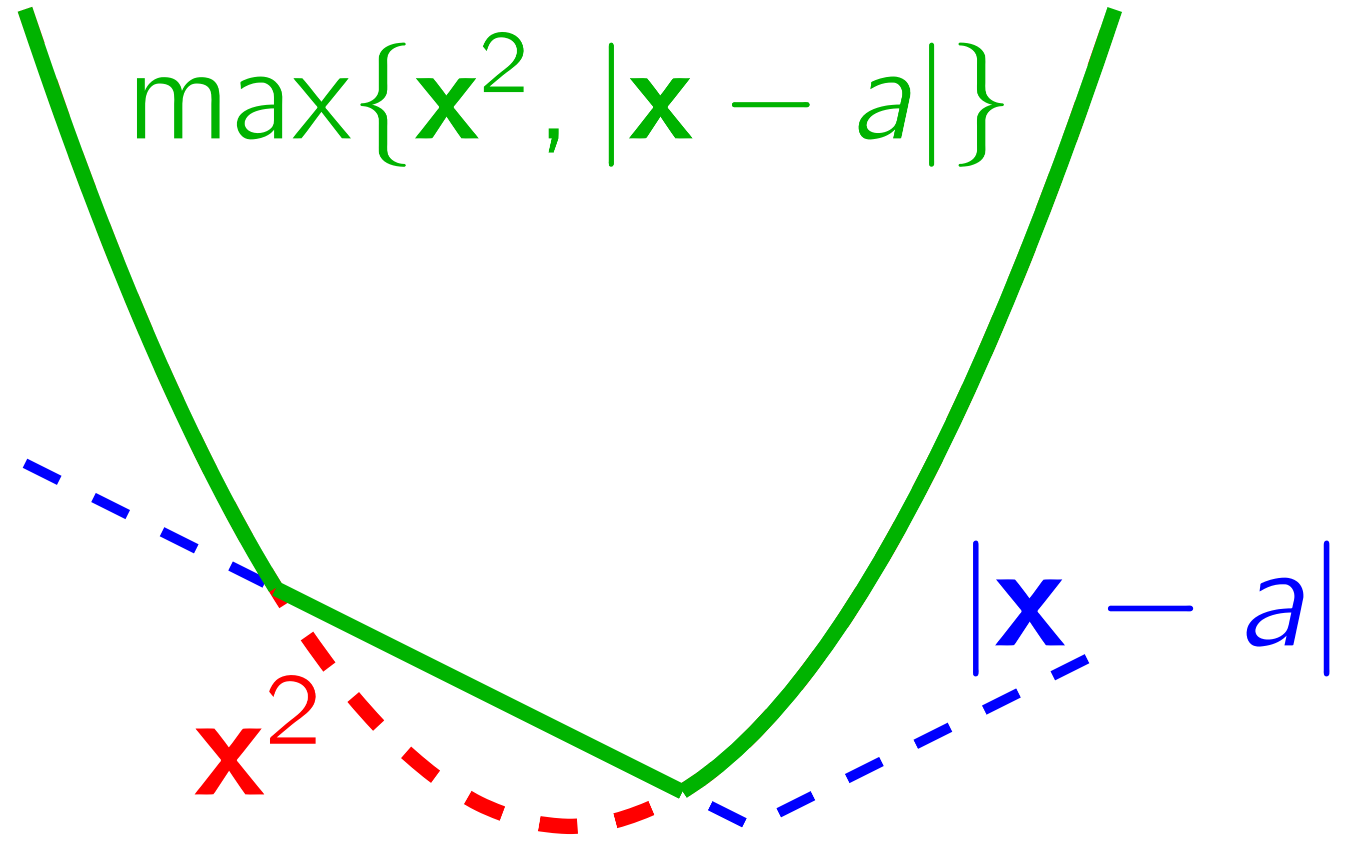
Convexity of Max-Margin Loss*

Inner product $\mathbf{w_j} \cdot \mathbf{x}$ is linear in $\mathbf{w_j}$ $\Rightarrow$ $\mathbf{w_j} \cdot \mathbf{x}$ convex in $\mathbf{w_j}$ (and concave)

Convexity of Max-Margin Loss*

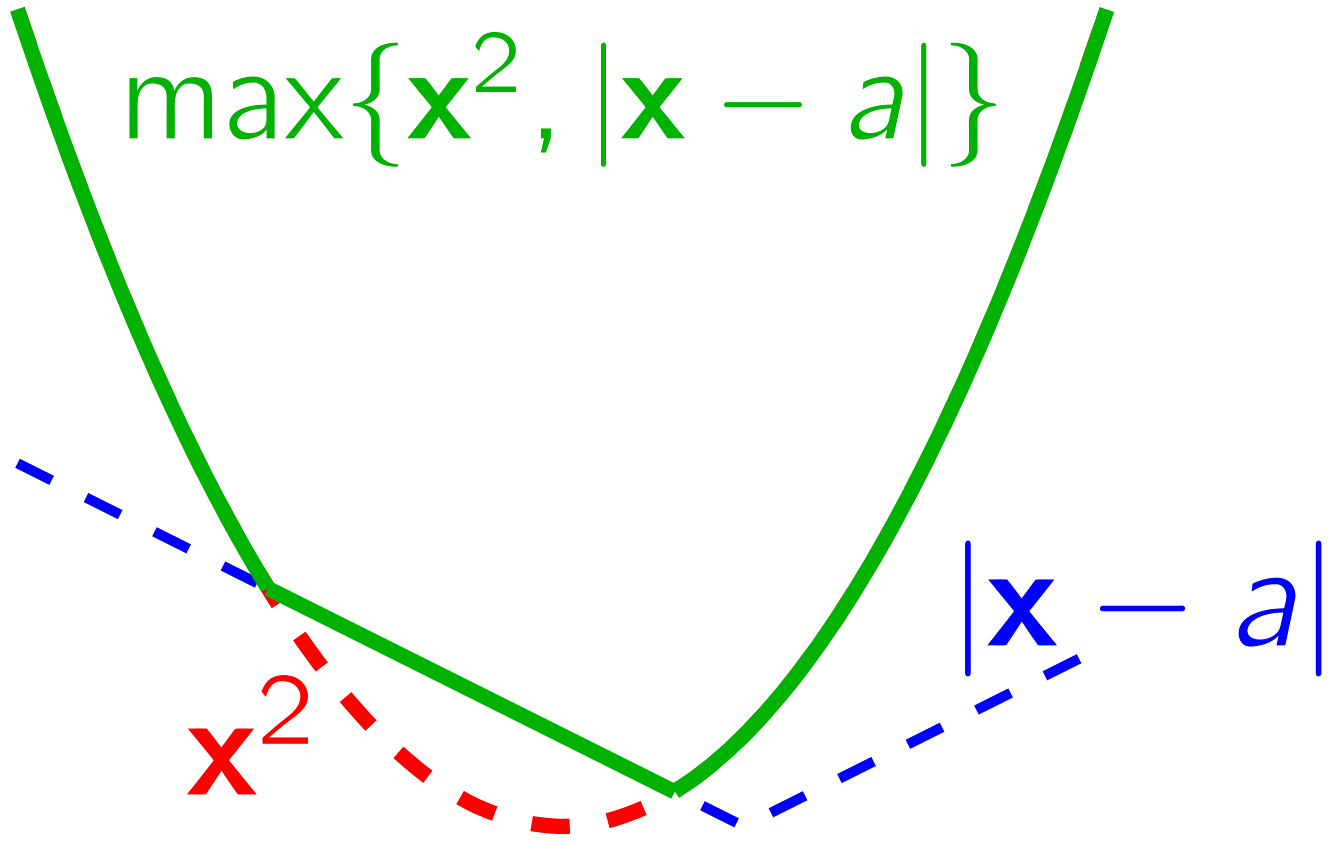
Inner product $\mathbf{w_j} \cdot \mathbf{x}$ is linear in $\mathbf{w_j}$ $\Rightarrow$ $\mathbf{w_j} \cdot \mathbf{x}$ convex in $\mathbf{w_j}$ (and concave)

Maximum of convex function is convex



Convexity of Max-Margin Loss*

Inner product $\mathbf{w_j} \cdot \mathbf{x}$ is linear in $\mathbf{w_j}$ $\Rightarrow$ $\mathbf{w_j} \cdot \mathbf{x}$ convex in $\mathbf{w_j}$ (and concave)



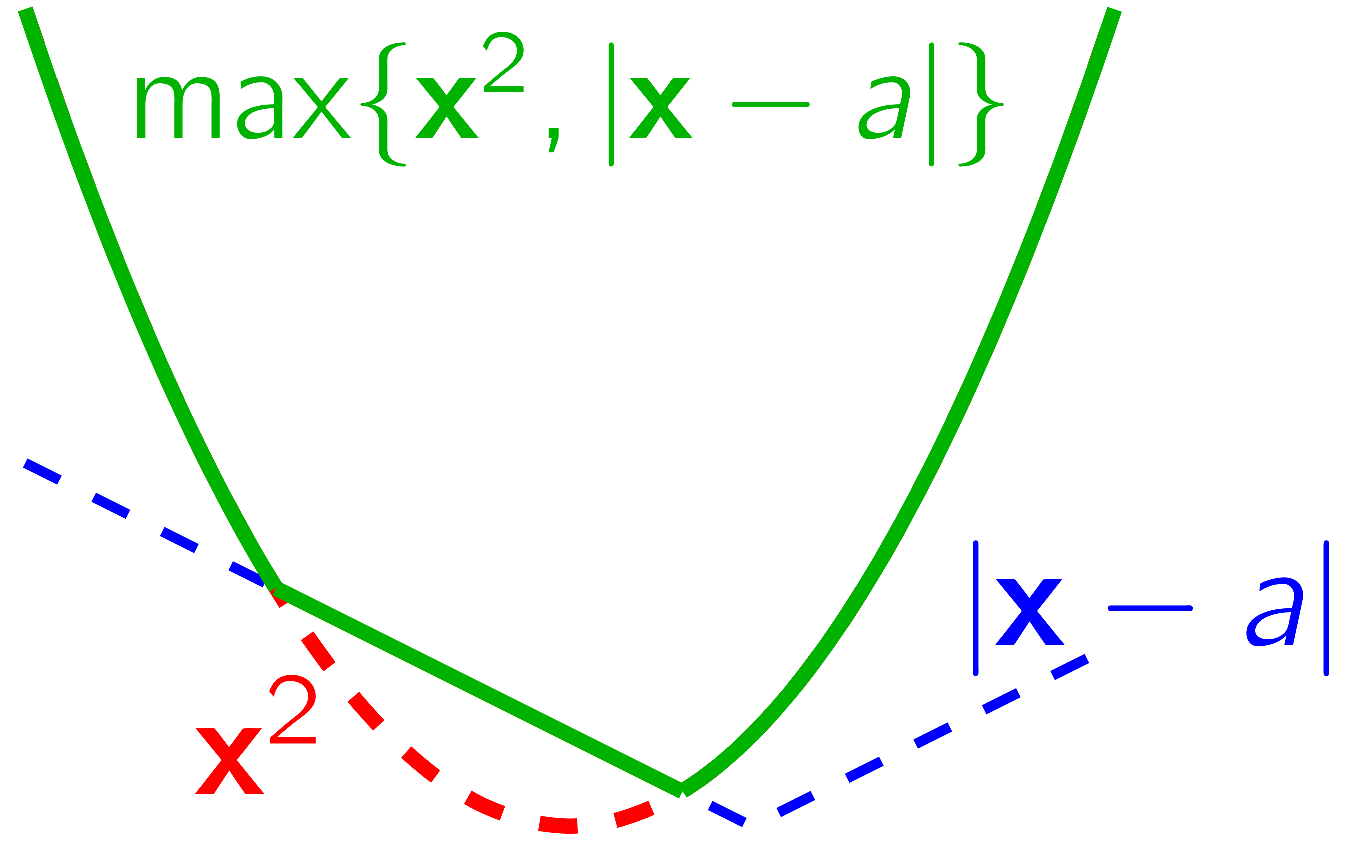
Maximum of convex function is convex

Therefore $\max\limits_{j \neq y} \mathbf{w}_j \cdot \mathbf{x}$ is convex in W

Convexity of Max-Margin Loss*

Inner product $\mathbf{w_j} \cdot \mathbf{x}$ is linear in $\mathbf{w_j}$ $\Rightarrow$ $\mathbf{w_j} \cdot \mathbf{x}$ convex in $\mathbf{w_j}$ (and concave)

Maximum of convex function is convex



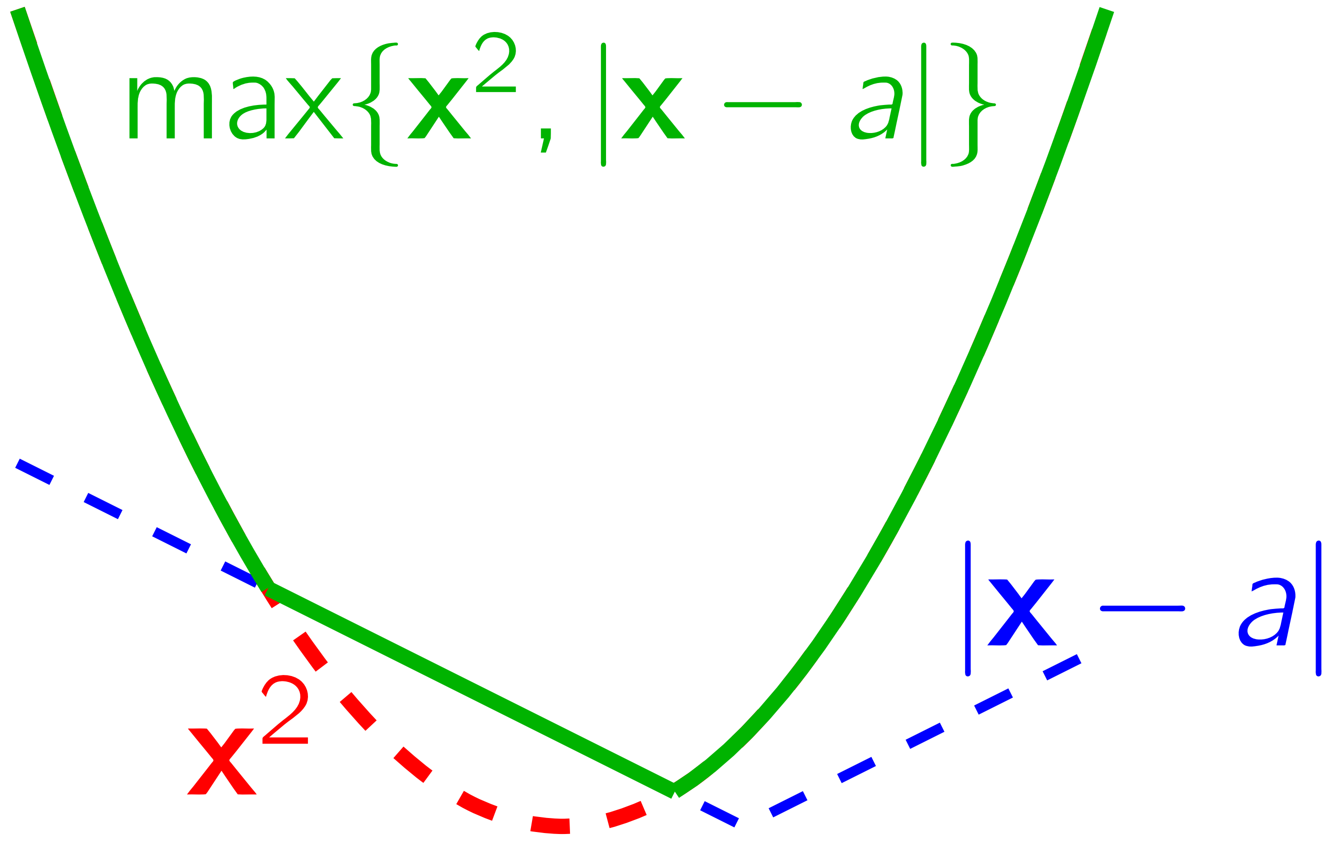
Therefore $\max_{j \neq y} \mathbf{w}_j \cdot \mathbf{x}$ is convex in W

Sum of convex functions is convex $\Rightarrow$ $\gamma + \max_{j} \mathbf{w}_j \cdot \mathbf{x} - \mathbf{w}_y \cdot \mathbf{x}$ is convex in W

Convexity of Max-Margin Loss*

Inner product $\mathbf{w_j} \cdot \mathbf{x}$ is linear in $\mathbf{w_j}$ $\Rightarrow$ $\mathbf{w_j} \cdot \mathbf{x}$ convex in $\mathbf{w_j}$ (and concave)

Maximum of convex function is convex



Therefore $\max_{j \neq y} \mathbf{w}_j \cdot \mathbf{x}$ is convex in W

Sum of convex functions is convex $\Rightarrow$ $\gamma + \max_j \mathbf{w}_j \cdot \mathbf{x} - \mathbf{w}_y \cdot \mathbf{x}$ is convex in W

Using convexity of maximum again: $\ell^{\mathrm{MM}}(\mathbf{z}) = \max\left\{0, \gamma + \max_{j \neq y} z_j - z_y\right\}$

Implies that $\ell^{\mathrm{MM}}(\mathbf{z})$ is convex in W

Multivariate Logistic Regression

As before $\mathbf{z} = W\mathbf{x}$

Define probability of class c to be $\mathbf{P}\left[j \mid \mathbf{z}\right] = \dfrac{e^{z_j}}{Z}$ where $Z = \sum_{j=1}^{k} e^{z_j}$

Loss: -log-probability of correct class $\ell^{LR}(\mathbf{z}) = -\log\left(P\left[y \mid \mathbf{x}\right]\right)$

$$\hat{y} = \arg\max_{j=1}^{k} z_j \neq y \qquad = \underbrace{\log\left(\sum_j \exp(z_j)\right)}_{\text{SoftMax}} - z_y$$

$$\text{SoftMax}(\mathbf{z}) \equiv \log\left(\sum_i e^{z_i}\right) \geq \max_i z_i$$

SoftMax

$$\ell^{\mathrm{LR}}(\mathrm{z}) = -\log\left(P\left[\mathrm{y}\,|\,\mathbf{x}\right]\right) \;=\; \underbrace{\log\left(\sum_j \exp(z_j)\right)}_{\text{SoftMax}} - \mathrm{z}_{\mathrm{y}}$$

Recall $\hat{\mathrm{y}} = \arg\max_{\mathrm{j}=1}^{\mathrm{k}} \mathrm{z}_{\mathrm{j}} \;\Rightarrow\; \mathrm{SoftMax}(\mathbf{z}) = \log\left(\sum_{\mathrm{j}} \mathrm{e}^{\mathrm{z}_{\mathrm{j}}}\right) \geq \log\left(\mathrm{e}^{\mathrm{z}_{\hat{\mathrm{y}}}}\right) = \mathrm{z}_{\hat{\mathrm{y}}}$

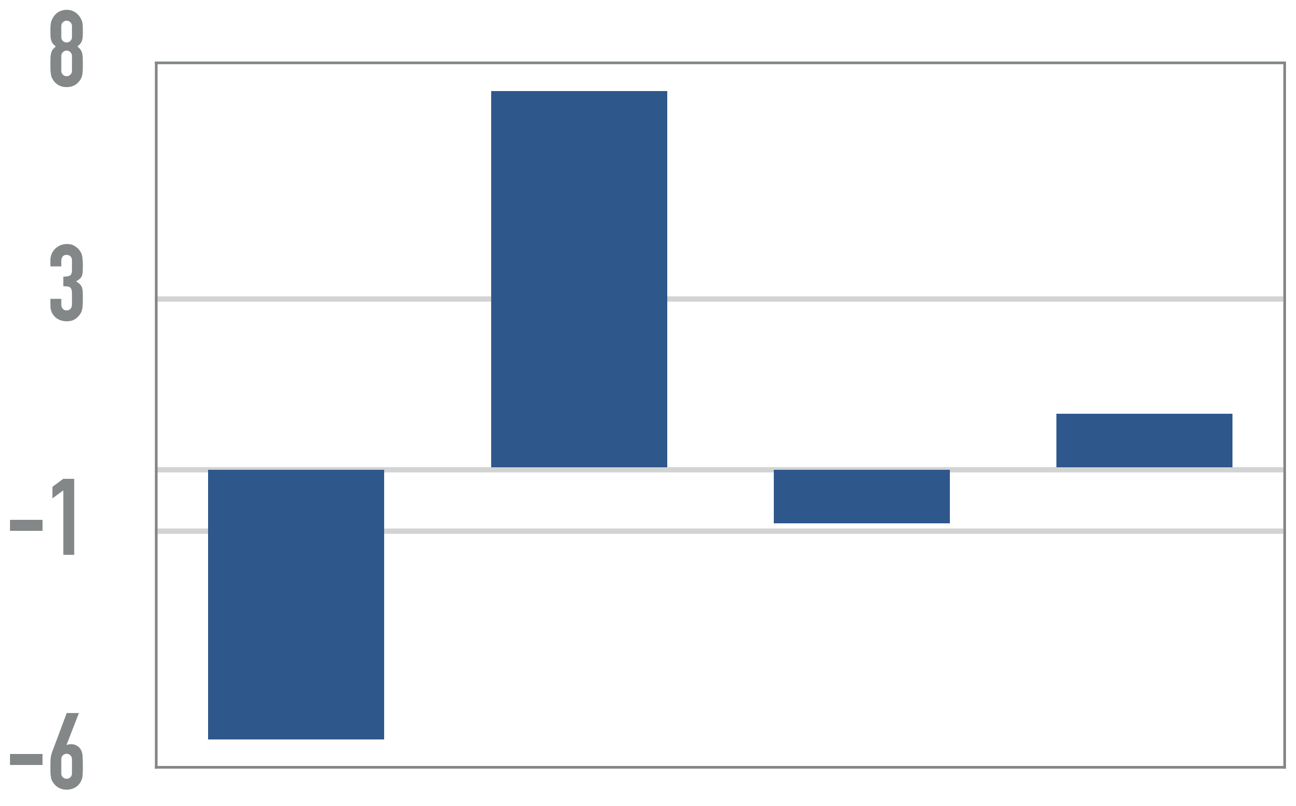
Case I: $\forall \mathrm{j} \neq \hat{\mathrm{y}} : \mathrm{z}_{\hat{\mathrm{y}}} \gg \mathrm{z}_{\mathrm{j}} \;\Rightarrow\; \mathrm{SoftMax}(\mathbf{z}) \approx \mathrm{z}_{\hat{\mathrm{y}}}$

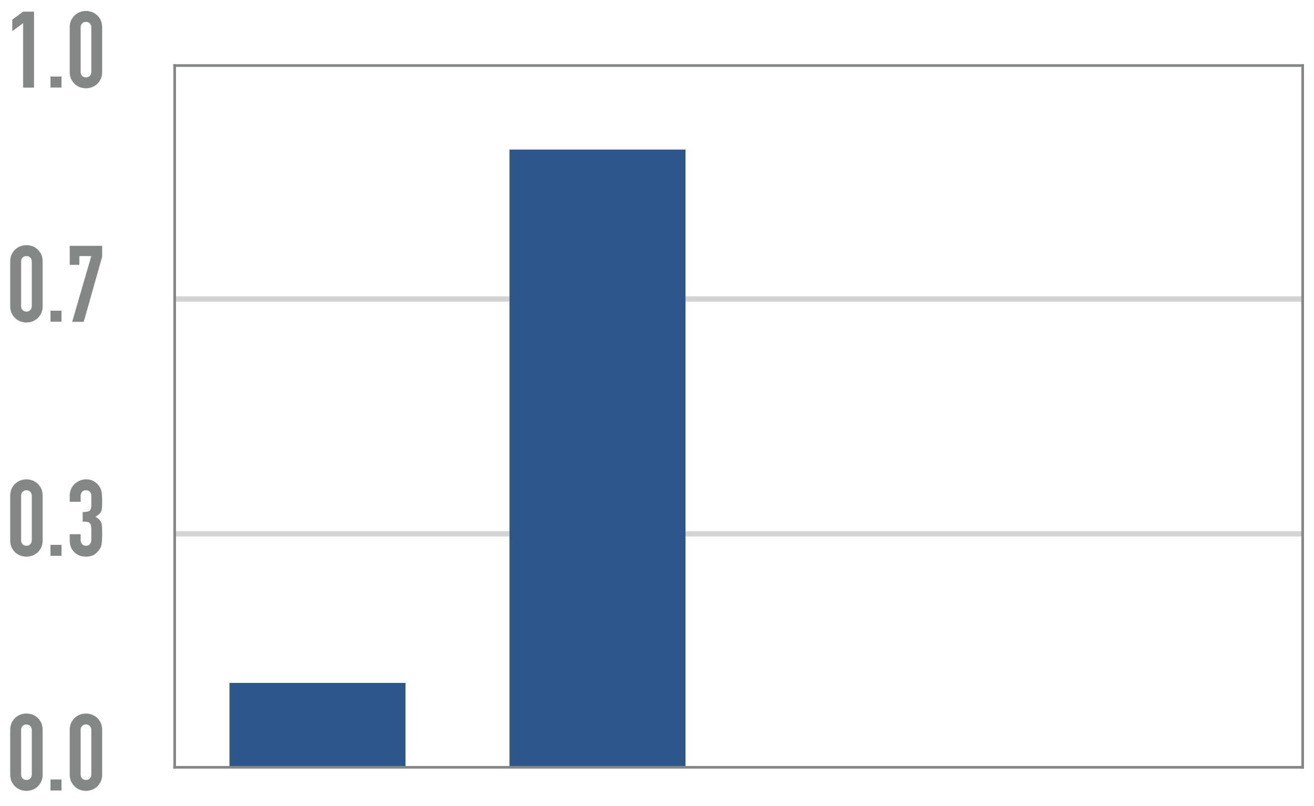
Case II: $\forall \mathrm{j} : \mathrm{z}_{\hat{\mathrm{y}}} \approx \mathrm{z}_{\mathrm{j}} \;\Rightarrow\; \mathrm{SoftMax}(\mathbf{z}) \approx \log\left(\sum_{\mathrm{j}} \mathrm{e}^{\mathrm{z}_{\hat{\mathrm{y}}}}\right) = \log(\mathrm{k})\,\mathrm{z}_{\hat{\mathrm{y}}}$

z	5	7	-1	1

p	0.1	0.9	0	0

$$\text{SoftMax}(\mathbf{z}) \approx 7.13$$



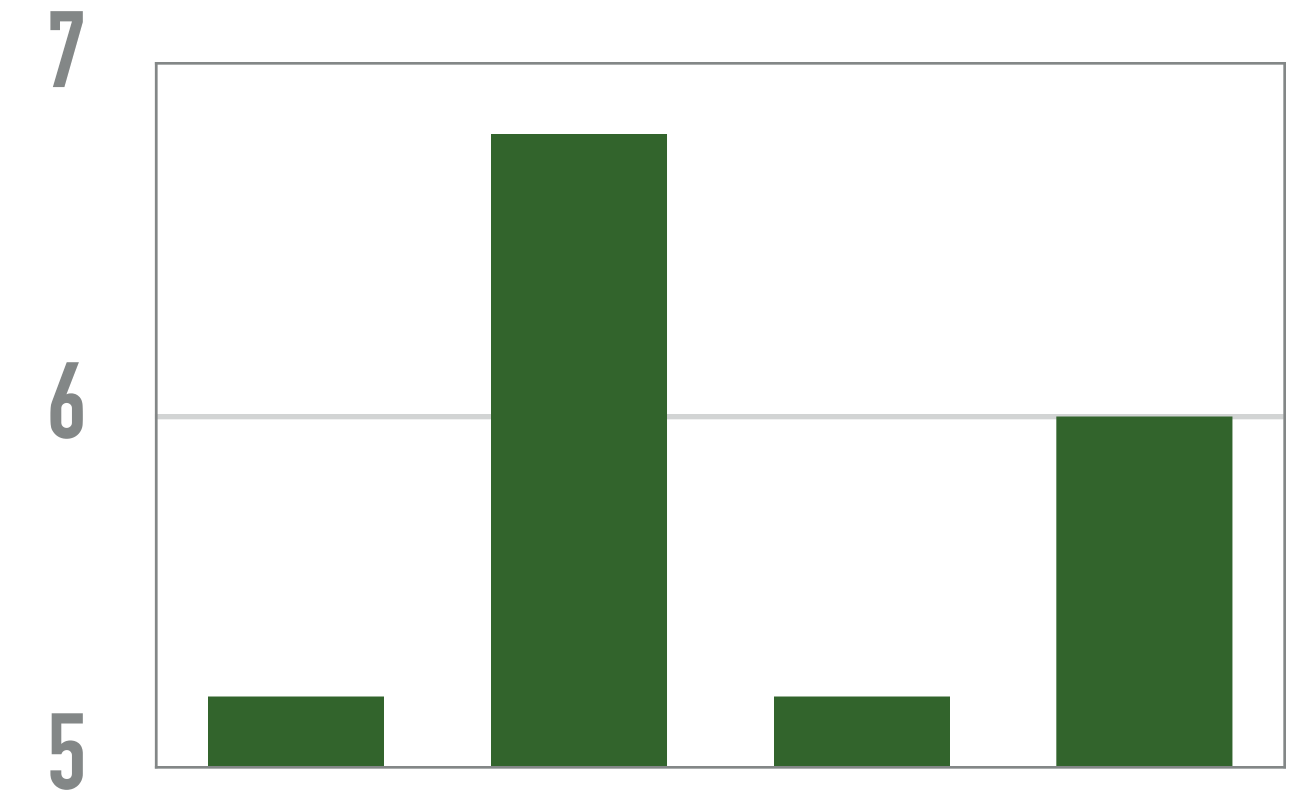


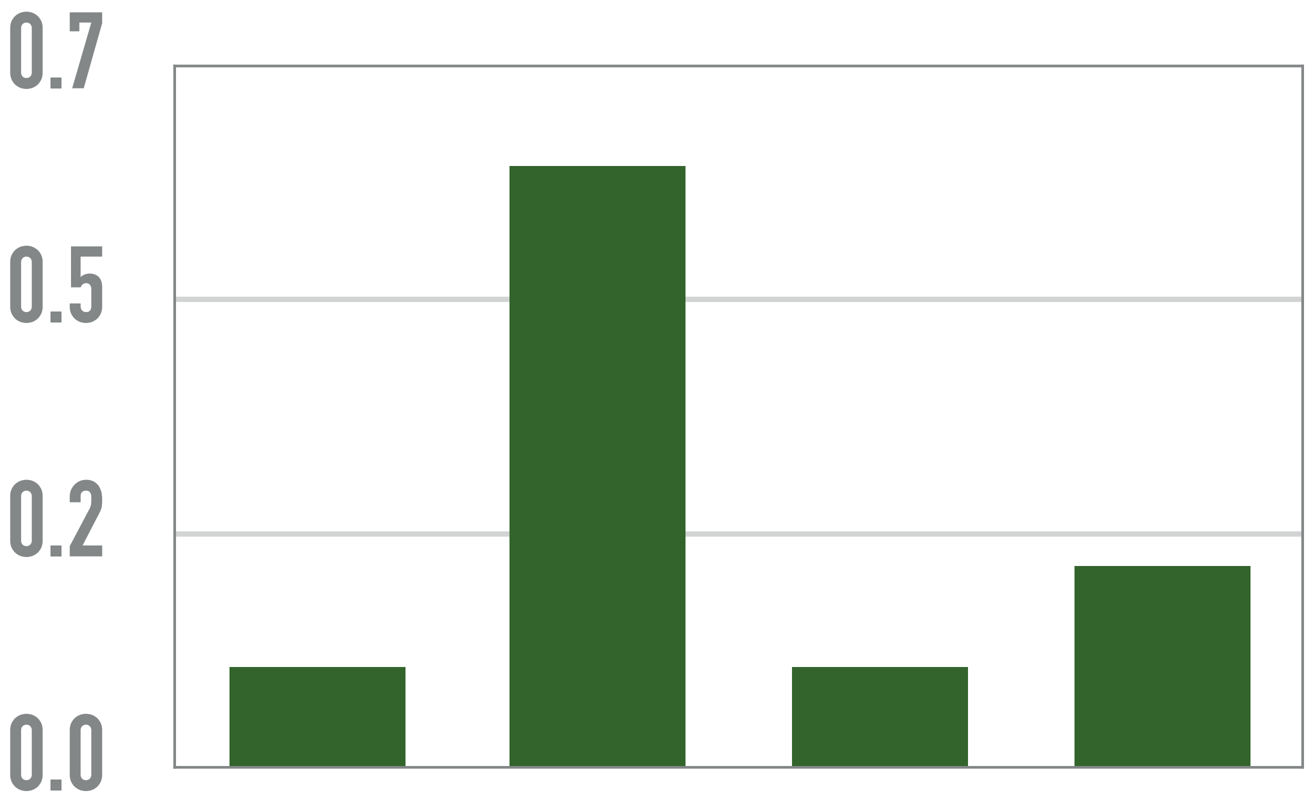
$$\ell^{\text{LR}}(\text{z}) = \text{SoftMax}(\mathbf{z}) - \text{z}_\text{y}$$
$$\approx 7.13 - 7 = 0.13$$

z	5	7	5	6

p	0.1	0.6	0.1	0.2

$$\text{SoftMax}(\mathbf{z}) \approx 7.5$$





$$\ell^{\text{LR}}(\text{z}) = \text{SoftMax}(\mathbf{z}) - \text{z}_\text{y}$$
$$\approx 7.5 - 7 = 0.5$$

Numerically Stable Evaluation

Exponentials of large values could cause overflow

Use the following properties:

➡ $$\log\Big(\sum_{\mathrm{j}} \exp(\mathrm{z_j})\Big) = \log\Big(\sum_{\mathrm{j}} \exp(\mathrm{z_j} - \mathrm{z_{\hat{y}}})\Big) + \mathrm{z_{\hat{y}}}$$

➡ $$\mathrm{P}[\mathrm{i}\,|\,\mathbf{x}] = \frac{\exp(\mathrm{z_i})}{\sum_{\mathrm{j}} \exp(\mathrm{z_j})} = \frac{\exp(\mathrm{z_i} - \mathrm{z_{\hat{y}}})}{\sum_{\mathrm{j}} \exp(\mathrm{z_j} - \mathrm{z_{\hat{y}}})} = \frac{\exp(\tilde{z}_i)}{\sum_j \exp(\tilde{z}_j)}$$

Transform $\mathbf{z} \mapsto \mathbf{z} - \mathrm{z_{\hat{y}}} = \tilde{\mathbf{z}}$

Define: $\ell^{\mathrm{LR}}(\tilde{\mathbf{z}}) = \mathrm{SoftMax}(\tilde{\mathbf{z}}) - \mathrm{z_y} + \mathrm{z_{\hat{y}}}$

Numerically Stable Evaluation

Exponentials of large values could cause overflow

Use the following properties:

- $$\log\Big(\sum_j \exp(z_j)\Big) = \log\Big(\sum_j \exp(z_j - z_{\hat{y}})\Big) + z_{\hat{y}}$$
- $$P[i\,|\,\mathbf{x}] = \frac{\exp(z_i)}{\sum_j \exp(z_j)} = \frac{\exp(z_i - z_{\hat{y}})}{\sum_j \exp(z_j - z_{\hat{y}})} = \frac{\exp(\tilde{z}_i)}{\sum_j \exp(\tilde{z}_j)}$$

Transform $\mathbf{z} \mapsto \mathbf{z} - z_{\hat{y}} = \tilde{\mathbf{z}}$

Define: $\ell^{\mathrm{LR}}(\tilde{\mathbf{z}}) = \mathrm{SoftMax}(\tilde{\mathbf{z}}) - z_y + z_{\hat{y}}$

MC Logistic Regression & Error

Multiclass prediction error:

$$\ell^{\mathsf{MC}}(\mathsf{y}, \mathbf{z}) = 1 \quad \Leftrightarrow \quad \hat{\mathsf{y}} = \arg\max_{\mathsf{j}=1}^{\mathsf{k}} \mathsf{z}_{\mathsf{j}} \neq \mathsf{y}$$

If $\hat{\mathsf{y}} \neq \mathsf{y}$ then $\ell^{\mathsf{LR}}(\mathsf{z}) = \log\left(\sum_j \exp(\mathsf{z}_j)\right) - \mathsf{z}_\mathsf{y}$

$$\geq \log\Big(\underbrace{\exp(\mathsf{z}_\mathsf{y}) + \exp(\mathsf{z}_{\hat{\mathsf{y}}})}_{\geq 2\exp(\mathsf{z}_\mathsf{y})}\Big) - \mathsf{z}_\mathsf{y} \geq \log(2)$$

Therefor $\ell^{\mathsf{MC}}(\mathsf{y}, \mathbf{z}) = 1 \quad \Rightarrow \quad \ell^{\mathsf{LR}}(\mathsf{y}, \mathbf{z}) \geq 2$

Multiclass LR & Max-Margin

If $\ell^{\mathrm{MM}}(\mathrm{z}) \geq \beta > 0 \quad \Rightarrow \quad \exists \mathrm{j} : \gamma + \mathrm{z_j} - \mathrm{z_y} \geq \beta$

Then $\ell^{\mathrm{LR}}(\mathrm{z}) = \log\left(\sum_j \exp(\mathrm{z}_j)\right) - \mathrm{z_y} \;\geq \log\left(\exp(\mathrm{z_y}) + \exp(\mathrm{z_j})\right) - \mathrm{z_y}$

Hence $\ell^{\mathrm{LR}}(\mathrm{z}) \geq \log\left((\exp(\mathrm{z_y}) + \mathrm{e}^{\beta-\gamma} \exp(\mathrm{z_y})\right) - \mathrm{z_y}$

$$\geq \log(1 + \mathrm{e}^{\beta-\gamma}) \geq \beta - \gamma$$

In summary: $\ell^{\mathrm{MM}}(\mathrm{y}, \mathbf{z}) = \beta \quad \Rightarrow \quad \ell^{\mathrm{LR}}(\mathrm{y}, \mathbf{z}) \geq \beta - \gamma$

SGD for Multiclass

Instead of $\mathbf{w}_t$ maintain a matrix W^t

SGD for Multiclass

Instead of $\mathbf{w}_t$ maintain a matrix W^t

Loss $\mathscr{L}(W)$ is a function of W $\Rightarrow$ Gradient is a matrix $G^t = \nabla_W \mathscr{L}(W)$

SGD for Multiclass

Instead of $\mathbf{w}_t$ maintain a matrix W^t

Loss $\mathscr{L}(W)$ is a function of W $\Rightarrow$ Gradient is a matrix $G^t = \nabla_W \mathscr{L}(W)$

Gradient step without (or before) projection $W^{t+1} \leftarrow W^t - \eta_t G^t$

SGD for Multiclass

Instead of $\mathbf{w}_t$ maintain a matrix W^t

Loss $\mathscr{L}(W)$ is a function of W $\Rightarrow$ Gradient is a matrix $G^t = \nabla_W \mathscr{L}(W)$

Gradient step without (or before) projection $W^{t+1} \leftarrow W^t - \eta_t G^t$

Projection adheres with matrix form:

$$\begin{bmatrix} \| - \mathbf{w}_1 - \| \leq r \\ \| - \mathbf{w}_2 - \| \leq r \\ \vdots \\ \| - \mathbf{w}_k - \| \leq r \end{bmatrix}$$

SGD for Multiclass

Instead of $\mathbf{w}_t$ maintain a matrix W^t

Loss $\mathscr{L}(W)$ is a function of W $\Rightarrow$ Gradient is a matrix $G^t = \nabla_W \mathscr{L}(W)$

Gradient step without (or before) projection $W^{t+1} \leftarrow W^t - \eta_t G^t$

Projection adheres with matrix form:

$$\begin{bmatrix} \| - \mathbf{w}_1 - \| \leq r \\ \| - \mathbf{w}_2 - \| \leq r \\ \vdots \\ \| - \mathbf{w}_k - \| \leq r \end{bmatrix} \qquad \mathbf{w}_j^{t+1/2} \leftarrow \mathbf{w}_j^t - \eta_t \mathbf{g}_j^t$$

SGD for Multiclass

Instead of $\mathbf{w}_t$ maintain a matrix W^t

Loss $\mathscr{L}(W)$ is a function of W $\Rightarrow$ Gradient is a matrix $G^t = \nabla_W \mathscr{L}(W)$

Gradient step without (or before) projection $W^{t+1} \leftarrow W^t - \eta_t G^t$

Projection adheres with matrix form:

$$\begin{bmatrix} \| - \mathbf{w}_1 - \| \leq r \\ \| - \mathbf{w}_2 - \| \leq r \\ \vdots \\ \| - \mathbf{w}_k - \| \leq r \end{bmatrix} \qquad \begin{aligned} \mathbf{w}_j^{t+1/2} &\leftarrow \mathbf{w}_j^t - \eta_t \mathbf{g}_j^t \\ \mathbf{w}_j^{t+1} &\leftarrow \min\left\{1, r/\|\mathbf{w}_j^{t+1/2}\|\right\} \mathbf{w}_j^{t+1/2} \end{aligned}$$

Multiclass Logistic Regression

For each example ($\mathbf{x}$,y) in mini-batch calculate $\mathbf{z} = W\mathbf{x}$

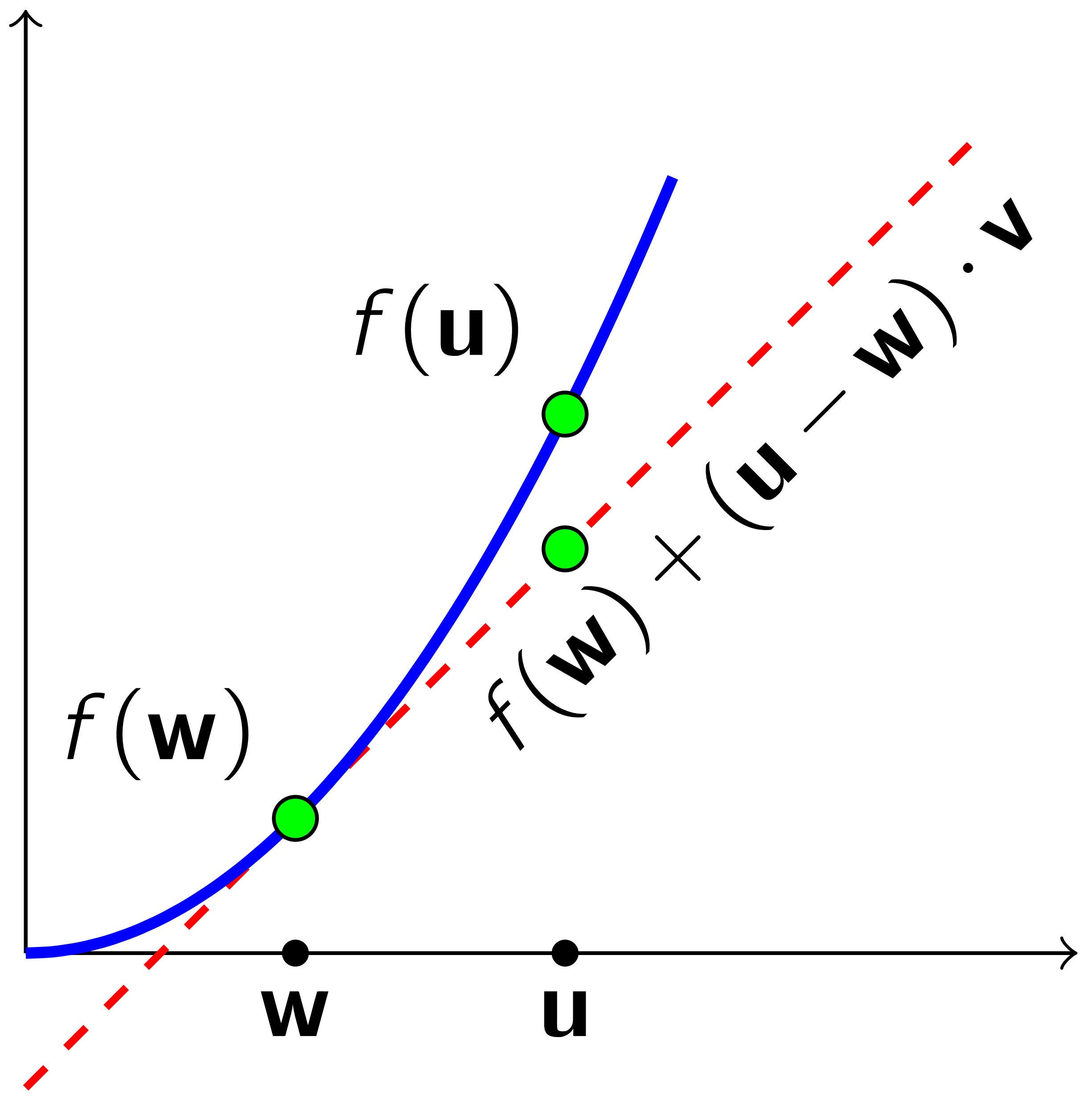
Define: $$v[j] = \frac{\exp(z_j)}{\sum_{i=1}^{k} \exp(z_i)} - \mathbf{1}[j=y]$$

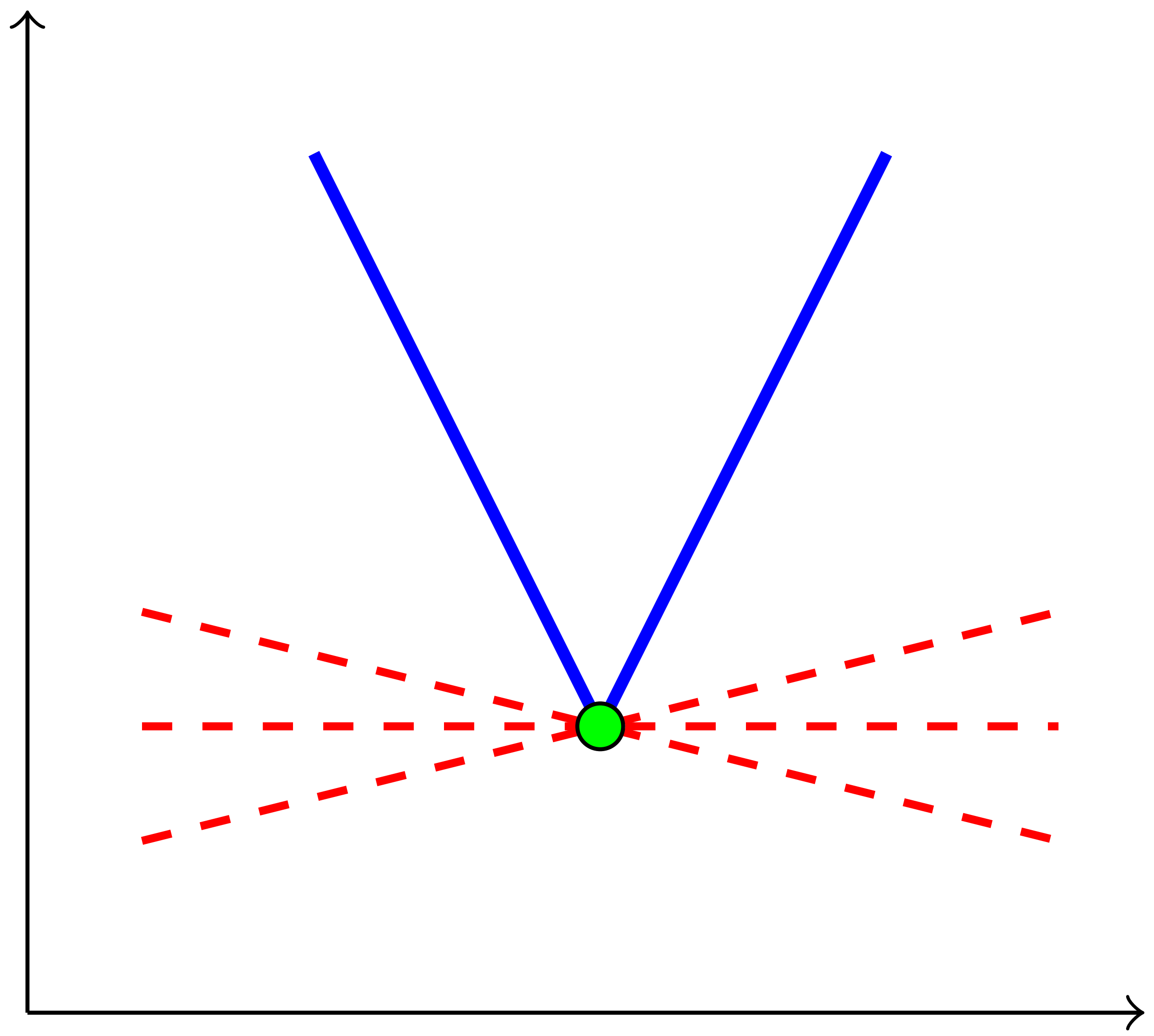
Gradient: $$\mathbf{v}\,\mathbf{x}^T = \begin{bmatrix} v[1]\,\mathbf{x} \\ v[2]\,\mathbf{x} \\ \vdots \\ v[k]\,\mathbf{x} \end{bmatrix}$$ and for mini-batch: $$G = \frac{1}{|S|} \sum_{i \in S} \mathbf{v_i}\,\mathbf{x_i}^T$$

Sub-gradients*

$\mathbf{w}$ is a sub-gradient of f at $\mathbf{w}$ if $\forall \mathbf{u},\quad \mathbf{f(u) \geq f(w) + v \cdot (u - w)}$

Differential set ∂f($\mathbf{w}$) is the set of sub-gradients of f at $\mathbf{w}$





Sub-gradient for Max Margin*

Set of labels with margin error $\Gamma = \{j \neq y \mid \gamma + z_j - z_y \geq 0\}$

Sub-gradients for MM loss are vectors **p** of the form:

$$p[y] = -1 \; ; \text{ for } j \notin \Gamma : p[j] = 0 \; ; \sum_{j \in \Gamma} p[j] = 1 \;\; (p[j] \geq 0)$$

Example: y = 2 **z** = [-2 3 2.5 1 7 4 1.9] $\gamma = 1$

$\Gamma = \{3, 5, 6\}$ **p** = [0 -1 1 0 0 0 0] or **p** = [0 -1 0.1 0 0.4 0.5 0] or …

Families of Updates

$y = 3 \quad \gamma = 2$

z	-1	2.5	3	4	1	2.5

For all forms of updates: $p[y] = 1$

Max only: $p[\hat{y}] = -1$

p	0	0	-1	1	0	0

Uniform: $\forall j \in \Gamma : p[j] = \dfrac{1}{|\Gamma|}$

p	0	1/3	-1	1/3	0	1/3

Margin-based: $\forall j \in \Gamma : p[j] = \dfrac{z_j - z_y}{Z}$ where $Z = \sum_{j \in \Gamma} z_j - z_y$

p	0	1.5	-1	3	0	1.5

/Z=6

p	0	1/4	-1	1/2	0	1/4

Mini-Batch Max-Margin Subgradient*

For each $i \in S$:

1. Calculate predicted values: $\mathbf{z}_i = W\,\mathbf{x}_i$

2. Calculate margin-error sets: $\Gamma = \{j \neq y_i \mid \gamma + z_i[j] - z_i[y] \geq 0\}$

3. Form update vectors: $\mathbf{p}_i$

4.Gradient: $G = \dfrac{1}{|S|} \sum_{i \in S} \mathbf{p}_i\,\mathbf{x}_i^T$

Max-Margin vs. Soft-Max*

Both updates of the form: $W^{t+1} \leftarrow W^t - \eta_t \, \mathbf{p} \, \mathbf{x}^\top$

Both satisfy $\sum_j p[j] = 0 \;\; ; \;\; \sum_{j \neq y} p[j] \leq 1$

If $\Gamma \neq \emptyset$ then for MM $p[y] = -1$ and for LR $p[y] > -1$

LR is a dense update $|\{j : p[j] > 0\}| = k - 1$

MM is a sparse update $|\{j : p[j] > 0\}| \leq |\Gamma|$

Cost-Sensitive Multiclass*

Classes often have semantic meaning and similarities

Image classification: Ape $\approx$ Baboon but Ape $\not\approx$ Subaru

Cost of confusing class y with class y': C(y,y')>0   [and C(y,y)=0]

Replace a fixed margin of γ with label-dependent margin C(y,y')

Cost Sensitive Multiclass*

Proxy for bounding $C(y, \hat{y})$

$$
\begin{aligned}
C(y, \hat{y}) &\le C(y, \hat{y}) + z_{\hat{y}} - z_y \\
&\le \underbrace{\max_r C(y, r) + z_r - z_y}_{\equiv \ell(y, \mathbf{z})}
\end{aligned}
$$

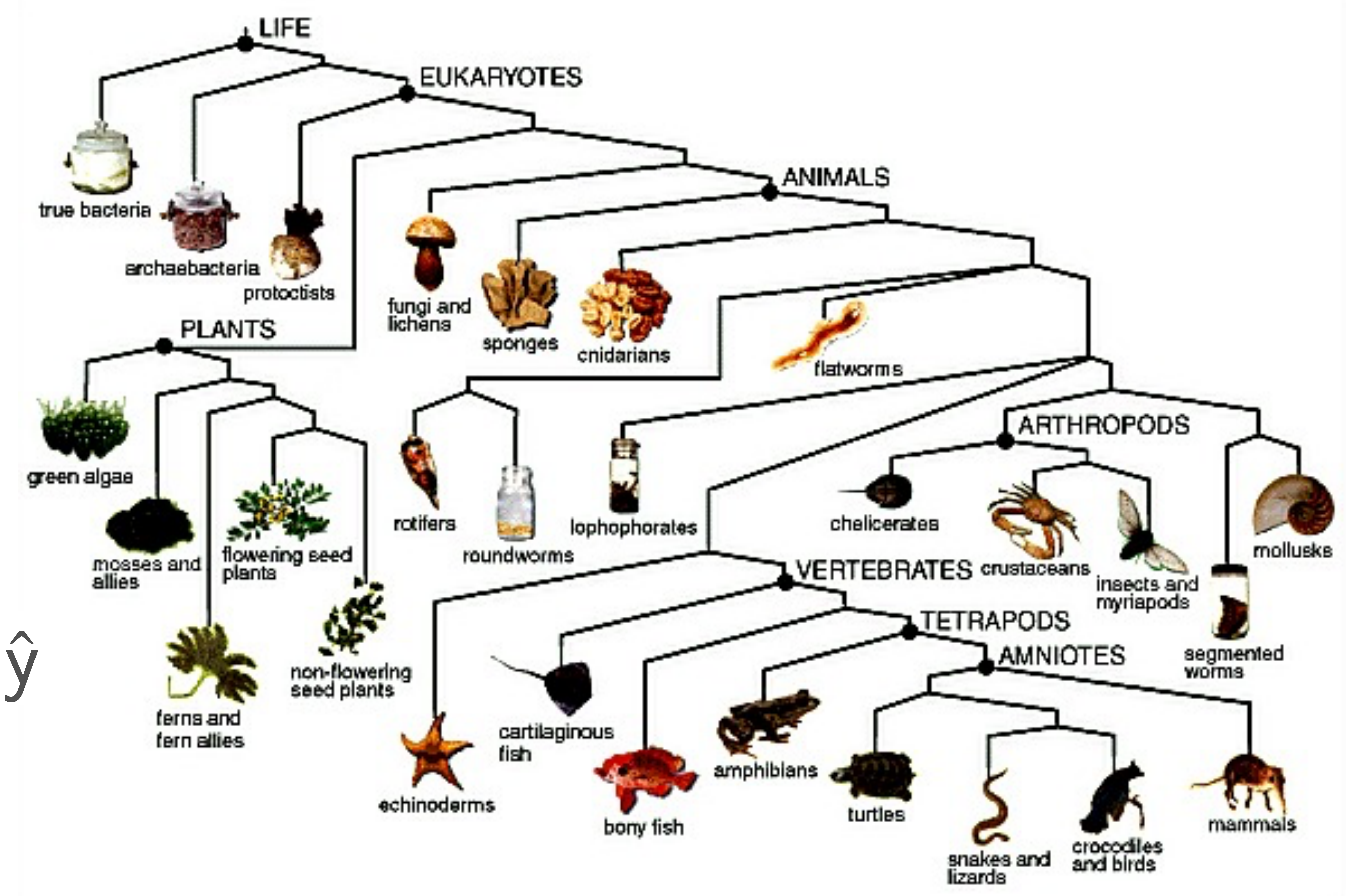
Usage: Hierarchical Classification*

Classes organized in a hierarchy

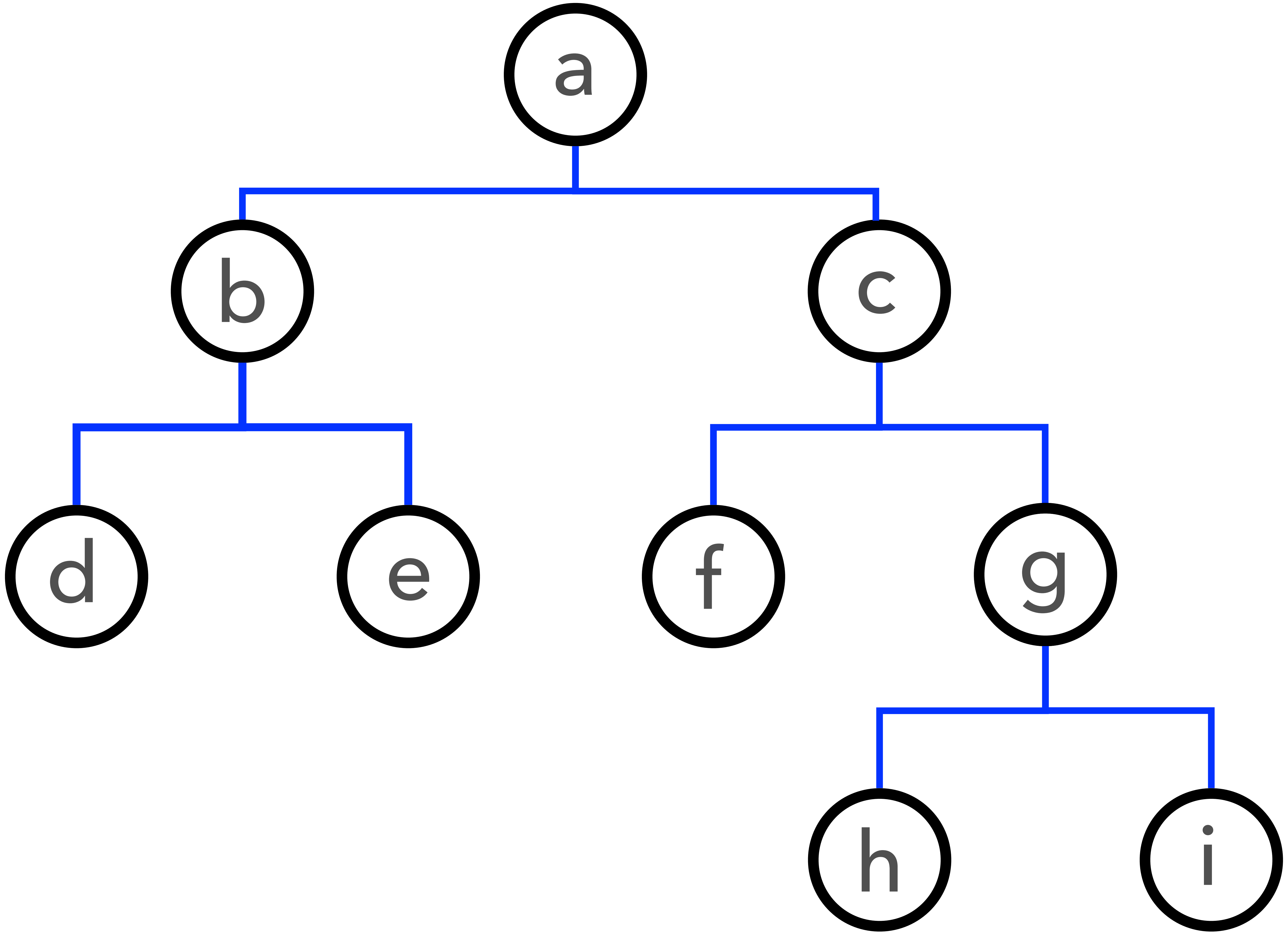
Cost of $C(y, \hat{y})$:

Length of (unique) path from y to $\hat{y}$

C(turtles, snakes) = 1
C(bacteria, mammals) = 14 ...

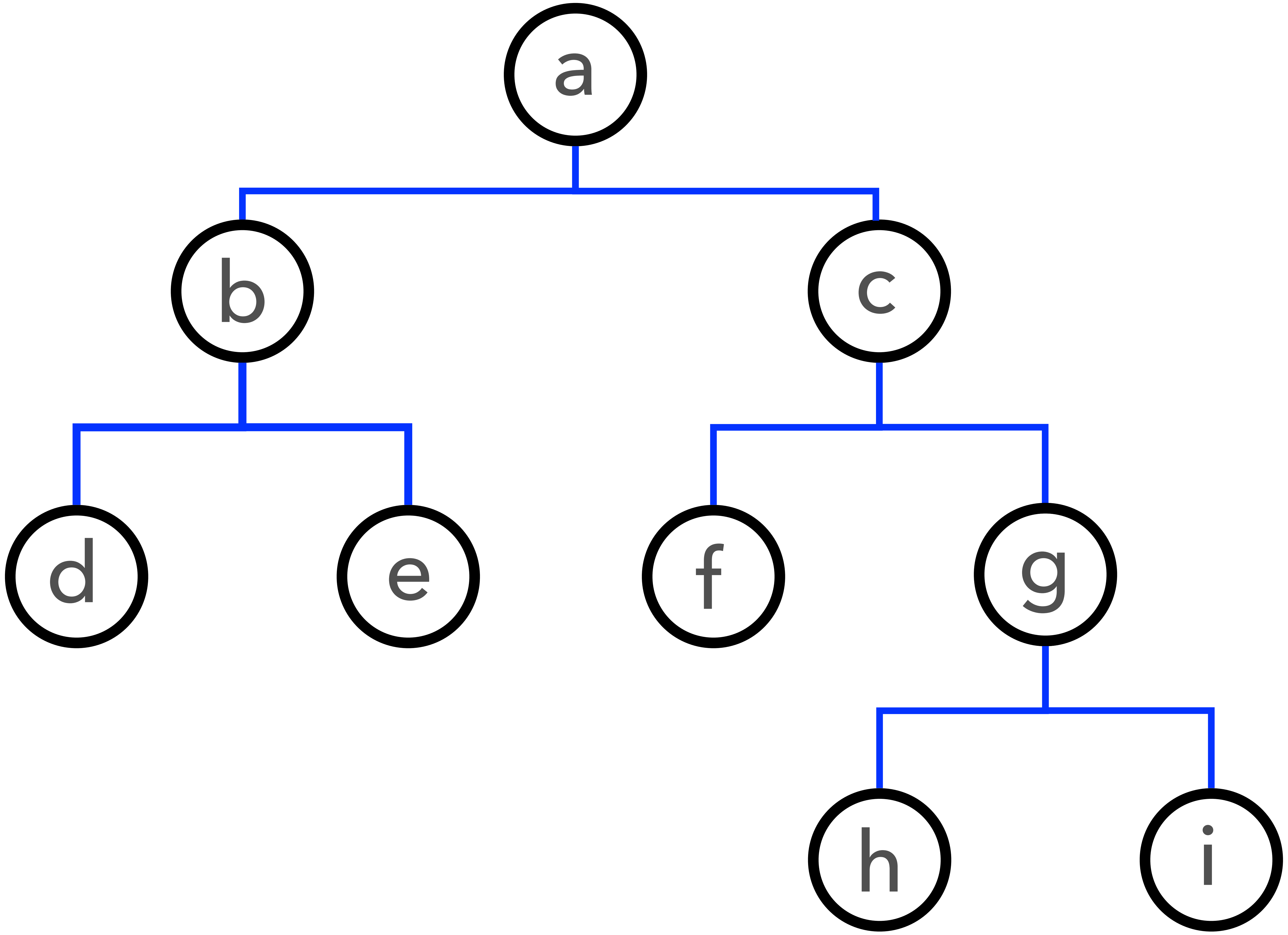


Hierarchical Cost*



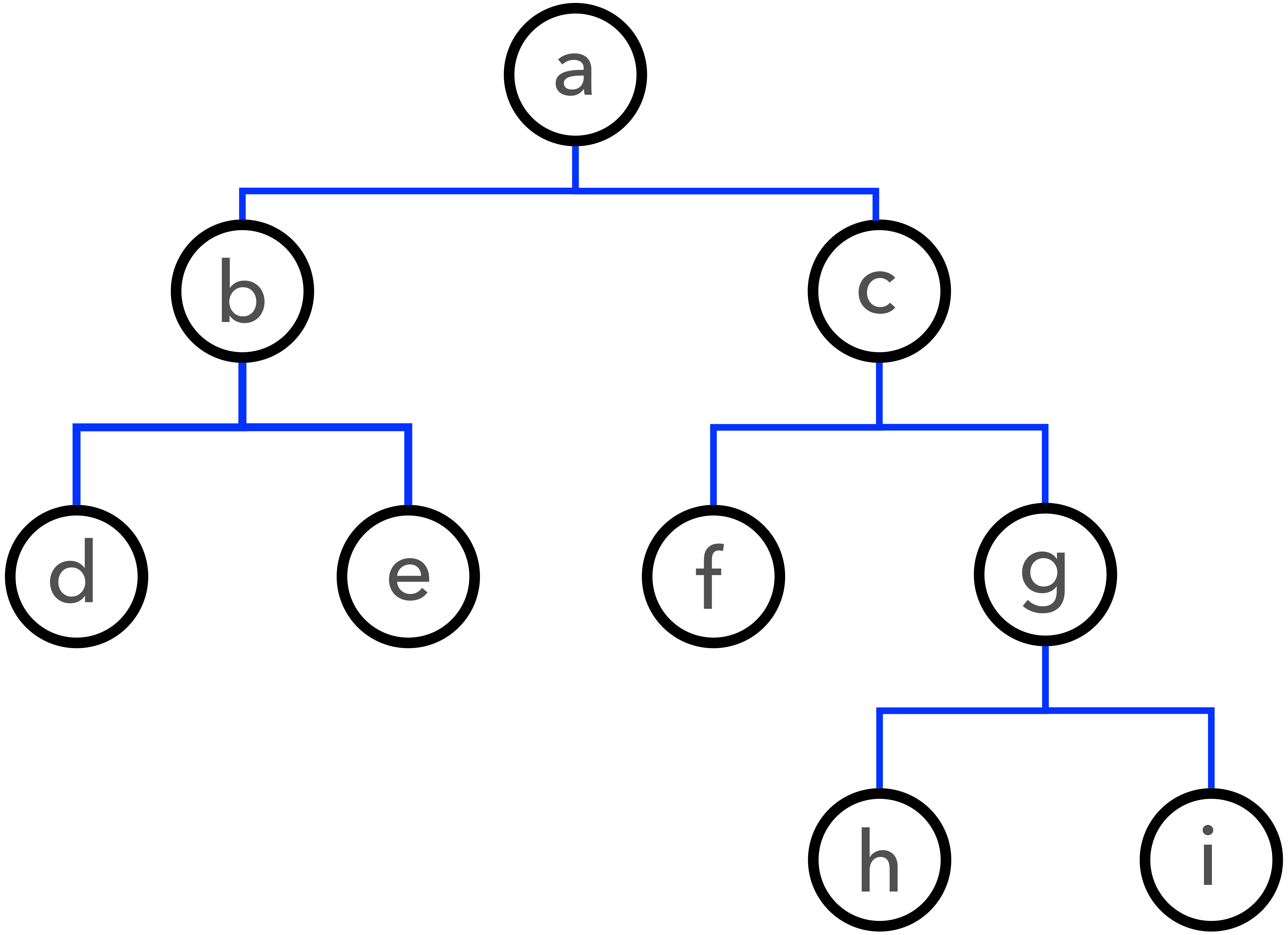
Hierarchical Cost*

$C(a,e) = ?$



Hierarchical Cost*

C(a,e) = 2



Hierarchical Cost*

C(a,e) = 2

C(b,h) = 4