Lecture 26:
Deriving Randomized Algorithms from Relaxations

Recap: Recipe

1. Write down sequential Rademacher relaxation for the problem
2. Move to upper bound by cutting down the tree
3. Ensure that admissibility condition holds
4. Solve for the prediction given by relaxation based algorithm

Online Vs Statistical Learning Rates

- Often optimal Online and statistical learning rates match

Online Vs Statistical Learning Rates

- Often optimal Online and statistical learning rates match
- Get rid of tree by draw of future from fixed distribution D

$$\mathbf{Rad}_n\left(x_{1:t}, y_{1:t}\right) = \sup_{\mathbf{x}} \mathbb{E}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left\{ 2 \sum_{s=t+1}^{n} \epsilon_s f\left(\mathbf{x}_{s-t}(\epsilon)\right) - \sum_{s=1}^{t} \ell\left(f\left(x_s\right), y_s\right) \right\}$$

Online Vs Statistical Learning Rates

- Often optimal Online and statistical learning rates match
- Get rid of tree by draw of future from fixed distribution D

$$\mathbf{Rad}_n\left(x_{1:t}, y_{1:t}\right) = \overset{\underset{x_{t+1:n}\sim D}{\mathbb{E}}}{\cancel{\sup_{\mathbf{x}}}}\, \mathbb{E}_{\epsilon_{t+1:n}} \sup_{f\in\mathcal{F}} \left\{ 2 \sum_{s=t+1}^{n} \epsilon_s f\left(\mathbf{x}_{s-t}(\epsilon\right) - \sum_{s=1}^{t} \ell(f\left(x_s\right), y_s\right) \right\}$$

Online Vs Statistical Learning Rates

- Often optimal Online and statistical learning rates match
- Get rid of tree by draw of future from fixed distribution D

$$\mathbf{Rad}_n\left(x_{1:t}, y_{1:t}\right) = \sup_{\mathbf{x}} \mathbb{E}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left\{ 2C \sum_{s=t+1}^{n} \epsilon_s f\left(\mathbf{x}_{s-t}(\epsilon)\right) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$

- Assume loss ℓ is convex and 1-Lipchitz in first argument

Random Playout

Define $R_t = x_{t+1:n}, \epsilon_{t+1:n}$ and let $D_t = D^{n-t} \times \text{Unif}\{\pm 1\}^{n-t}$

$$\phi_t(x_{1:t}, y_{1:t}; R_t) = \sup_{f \in \mathcal{F}} \left\{ 2C \sum_{s=t+1}^{n} \epsilon_s f(x_s) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$

Algorithm : **Draw $R_t \sim D_t$, and return,**

$$\tilde{q}_t(R_t) = \operatorname*{argmin}_{q \in \Delta(\mathcal{Y})} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \Phi_t(x_{1:t}, y_{1:t}, R_t) \right\}$$

Random Playout

Define $R_t = x_{t+1:n}, \epsilon_{t+1:n}$ and let $D_t = D^{n-t} \times \text{Unif}\{\pm 1\}^{n-t}$

$$\phi_t(x_{1:t}, y_{1:t}; R_t) = \sup_{f \in \mathcal{F}} \left\{ 2C \sum_{s=t+1}^{n} \epsilon_s f(x_s) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$

Algorithm : **Draw $R_t \sim D_t$, and return,**

$$\tilde{q}_t(R_t) = \underset{q \in \Delta(\mathcal{Y})}{\text{argmin}} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \Phi_t(x_{1:t}, y_{1:t}, R_t) \right\}$$

Why/When does this work?

Random Playout: Condition

Sufficient condition for randomized algorithm to work:

$$\sup_{x_t} \mathbb{E}_{\epsilon_t} \left[\sup_{f \in \mathcal{F}} \left\{ 2C \sum_{s=t+1}^{n} \epsilon_s f(x_s) + 2\epsilon_t f(x_t) - \sum_{s=1}^{t-1} \ell(f(x_s), y_s) \right\} \right]$$
$$\leq \mathbb{E}_{x_t \sim D, \epsilon_t} \left[\sup_{f \in \mathcal{F}} \left\{ 2C \sum_{s=t}^{n} \epsilon_s f(x_s) - \sum_{s=1}^{t-1} \ell(f(x_s), y_s) \right\} \right]$$

Random Playout

Initial condition is obvious, as for admissibility,

$$\inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbf{Rel}_n \left(x_{1:t}, y_{1:t} \right) \right\}$$

Random Playout

Initial condition is obvious, as for admissibility,

$$\inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbf{Rel}_n \left(x_{1:t}, y_{1:t} \right) \right\}$$

$$= \inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

Random Playout

Initial condition is obvious, as for admissibility,

$$\inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbf{Rel}_n \left(x_{1:t}, y_{1:t} \right) \right\}$$

$$= \inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

$$\leq \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim \tilde{q}_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

Random Playout

Initial condition is obvious, as for admissibility,

$$\inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbf{Rel}_n \left(x_{1:t}, y_{1:t} \right) \right\}$$

$$= \inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

$$\leq \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim \tilde{q}_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

$$= \sup_{y_t} \left\{ \mathbb{E}_{R_t \sim D_t} \left[\mathbb{E}_{\hat{y}_t \sim \tilde{q}_t(R_t)} \left[\ell(\hat{y}_t, y_t) \right] \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

Random Playout

Initial condition is obvious, as for admissibility,

$$\inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbf{Rel}_n \left(x_{1:t}, y_{1:t} \right) \right\}$$

$$= \inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

$$\leq \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim \tilde{q}_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

$$= \sup_{y_t} \left\{ \mathbb{E}_{R_t \sim D_t} \left[\mathbb{E}_{\hat{y}_t \sim \tilde{q}_t(R_t)} \left[\ell(\hat{y}_t, y_t) \right] \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

$$\leq \mathbb{E}_{R_t \sim D_t} \left[\sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim \tilde{q}_t(R_t)} \left[\ell(\hat{y}_t, y_t) \right] + \Phi_t(x_{1:t}, y_{1:t}, R_t) \right\} \right]$$

Random Playout

Initial condition is obvious, as for admissibility,

$$
\begin{aligned}
&\inf_{q_t}\sup_{y_t}\left\{\mathbb{E}_{\hat{y}_t\sim q_t}\left[\ell(\hat{y}_t,y_t)\right]+\mathbf{Rel}_n\left(x_{1:t},y_{1:t}\right)\right\}\\
&\quad=\inf_{q_t}\sup_{y_t}\left\{\mathbb{E}_{\hat{y}_t\sim q_t}\left[\ell(\hat{y}_t,y_t)\right]+\mathbb{E}_{R_t\sim D_t}\left[\Phi_t(x_{1:t},y_{1:t},R_t)\right]\right\}\\
&\quad\le\sup_{y_t}\left\{\mathbb{E}_{\hat{y}_t\sim\tilde{q}_t}\left[\ell(\hat{y}_t,y_t)\right]+\mathbb{E}_{R_t\sim D_t}\left[\Phi_t(x_{1:t},y_{1:t},R_t)\right]\right\}\\
&\quad=\sup_{y_t}\left\{\mathbb{E}_{R_t\sim D_t}\left[\mathbb{E}_{\hat{y}_t\sim\tilde{q}_t(R_t)}\left[\ell(\hat{y}_t,y_t)\right]\right]+\mathbb{E}_{R_t\sim D_t}\left[\Phi_t(x_{1:t},y_{1:t},R_t)\right]\right\}\\
&\quad\le\mathbb{E}_{R_t\sim D_t}\left[\sup_{y_t}\left\{\mathbb{E}_{\hat{y}_t\sim\tilde{q}_t(R_t)}\left[\ell(\hat{y}_t,y_t)\right]+\Phi_t(x_{1:t},y_{1:t},R_t)\right\}\right]\\
&\quad=\mathbb{E}_{R_t\sim D_t}\left[\inf_{q_t}\sup_{y_t}\left\{\mathbb{E}_{\hat{y}_t\sim q_t}\left[\ell(\hat{y}_t,y_t)\right]+\Phi_t(x_{1:t},y_{1:t},R_t)\right\}\right]
\end{aligned}
$$

Random Playout

Initial condition is obvious, as for admissibility,

$$
\begin{aligned}
&\inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbf{Rel}_n \left(x_{1:t}, y_{1:t} \right) \right\} \\
&\quad = \inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\} \\
&\quad \le \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim \tilde{q}_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\} \\
&\quad = \sup_{y_t} \left\{ \mathbb{E}_{R_t \sim D_t} \left[\mathbb{E}_{\hat{y}_t \sim \tilde{q}_t(R_t)} \left[\ell(\hat{y}_t, y_t) \right] \right] + \mathbb{E}_{R_t \sim D_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\} \\
&\quad \le \mathbb{E}_{R_t \sim D_t} \left[\sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim \tilde{q}_t(R_t)} \left[\ell(\hat{y}_t, y_t) \right] + \Phi_t(x_{1:t}, y_{1:t}, R_t) \right\} \right] \\
&\quad = \mathbb{E}_{R_t \sim D_t} \left[\inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \Phi_t(x_{1:t}, y_{1:t}, R_t) \right\} \right] \\
&\quad = \mathbb{E}_{R_t \sim D_t} \left[\sup_{p_t} \left\{ \inf_{\hat{y}_t \in \mathcal{Y}} \mathbb{E}_{y_t \sim p_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{y_t \sim p_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\} \right]
\end{aligned}
$$

Random Playout

To finish admissibility, note that

$$\sup_{p_t} \left\{ \inf_{\hat{y}_t \in \mathcal{Y}} \mathbb{E}_{y_t \sim p_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{y_t \sim p_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

Random Playout

To finish admissibility, note that

$$\sup_{p_t} \left\{ \inf_{\hat{y}_t \in \mathcal{Y}} \mathbb{E}_{y_t \sim p_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{y_t \sim p_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

$$= \sup_{p_t} \left\{ \inf_{\hat{y}_t \in \mathcal{Y}} \mathbb{E}_{y_t \sim p_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{y_t \sim p_t} \left[\sup_{f \in \mathcal{F}} \left\{ 2C \sum_{s=t+1}^{n} \epsilon_s f(x_s) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\} \right] \right\}$$

Random Playout

To finish admissibility, note that

$$\sup_{p_t} \left\{ \inf_{\hat{y}_t \in \mathcal{Y}} \mathbb{E}_{y_t \sim p_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{y_t \sim p_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\}$$

$$= \sup_{p_t} \left\{ \inf_{\hat{y}_t \in \mathcal{Y}} \mathbb{E}_{y_t \sim p_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{y_t \sim p_t} \left[\sup_{f \in \mathcal{F}} \left\{ 2C \sum_{s=t+1}^{n} \epsilon_s f(x_s) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\} \right] \right\}$$

$$\leq \sup_{x_t} \mathbb{E}_{\epsilon_t} \left[\sup_{f \in \mathcal{F}} \left\{ 2C \sum_{s=t+1}^{n} \epsilon_s f(x_s) + 2\epsilon_t f(x_t) - \sum_{s=1}^{t-1} \ell(f(x_s), y_s) \right\} \right]$$

Random Playout

To finish admissibility, note that

$$\sup_{p_t}\left\{\inf_{\hat{y}_t\in\mathcal{Y}}\mathbb{E}_{y_t\sim p_t}\left[\ell(\hat{y}_t,y_t)\right]+\mathbb{E}_{y_t\sim p_t}\left[\Phi_t(x_{1:t},y_{1:t},R_t)\right]\right\}$$

$$=\sup_{p_t}\left\{\inf_{\hat{y}_t\in\mathcal{Y}}\mathbb{E}_{y_t\sim p_t}\left[\ell(\hat{y}_t,y_t)\right]+\mathbb{E}_{y_t\sim p_t}\left[\sup_{f\in\mathcal{F}}\left\{2C\sum_{s=t+1}^{n}\epsilon_s f(x_s)-\sum_{s=1}^{t}\ell(f(x_s),y_s)\right\}\right]\right\}$$

$$\leq\sup_{x_t}\mathbb{E}_{\epsilon_t}\left[\sup_{f\in\mathcal{F}}\left\{2C\sum_{s=t+1}^{n}\epsilon_s f(x_s)+2\epsilon_t f(x_t)-\sum_{s=1}^{t-1}\ell(f(x_s),y_s)\right\}\right]$$

Condition:

$$\sup_{x_t}\mathbb{E}_{\epsilon_t}\left[\sup_{f\in\mathcal{F}}\left\{2C\sum_{s=t+1}^{n}\epsilon_s f(x_s)+2\epsilon_t f(x_t)-\sum_{s=1}^{t-1}\ell(f(x_s),y_s)\right\}\right]$$

$$\leq\mathbb{E}_{x_t\sim D,\epsilon_t}\left[\sup_{f\in\mathcal{F}}\left\{2C\sum_{s=t}^{n}\epsilon_s f(x_s)-\sum_{s=1}^{t-1}\ell(f(x_s),y_s)\right\}\right]$$

Random Playout

Hence,

$$\sup_{x_t} \inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbf{Rel}_n \left(x_{1:t}, y_{1:t} \right) \right\}$$

$$\leq \sup_{x_t} \mathbb{E}_{R_t \sim D_t} \left[\sup_{p_t} \left\{ \inf_{\hat{y}_t \in \mathcal{Y}} \mathbb{E}_{y_t \sim p_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{y_t \sim p_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\} \right]$$

Random Playout

Hence,

$$\sup_{x_t} \inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbf{Rel}_n \left(x_{1:t}, y_{1:t} \right) \right\}$$

$$\leq \sup_{x_t} \mathbb{E}_{R_t \sim D_t} \left[\sup_{p_t} \left\{ \inf_{\hat{y}_t \in \mathcal{Y}} \mathbb{E}_{y_t \sim p_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{y_t \sim p_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\} \right]$$

$$\leq \mathbb{E}_{R_t \sim D_t} \left[\mathbb{E}_{x_t \sim D, \epsilon_t} \left[\sup_{f \in \mathcal{F}} \left\{ 2C \sum_{s=t}^{n} \epsilon_s f(x_s) - \sum_{s=1}^{t-1} \ell(f(x_s), y_s) \right\} \right] \right]$$

Random Playout

Hence,

$$\sup_{x_t} \inf_{q_t} \sup_{y_t} \left\{ \mathbb{E}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbf{Rel}_n \left(x_{1:t}, y_{1:t} \right) \right\}$$

$$\leq \sup_{x_t} \mathbb{E}_{R_t \sim D_t} \left[\sup_{p_t} \left\{ \inf_{\hat{y}_t \in \mathcal{Y}} \mathbb{E}_{y_t \sim p_t} \left[\ell(\hat{y}_t, y_t) \right] + \mathbb{E}_{y_t \sim p_t} \left[\Phi_t(x_{1:t}, y_{1:t}, R_t) \right] \right\} \right]$$

$$\leq \mathbb{E}_{R_t \sim D_t} \left[\mathbb{E}_{x_t \sim D, \epsilon_t} \left[\sup_{f \in \mathcal{F}} \left\{ 2C \sum_{s=t}^{n} \epsilon_s f(x_s) - \sum_{s=1}^{t-1} \ell(f(x_s), y_s) \right\} \right] \right]$$

$$= \mathbb{E}_{R_{t-1} \sim D_{t-1}} \left[\Phi_t(x_{1:t-1}, y_{1:t-1}, R_{t-1}) \right]$$

$$= \mathbf{Rel}_n \left(x_{1:t-1}, y_{1:t-1} \right)$$

Example: Bit Prediction

- $\mathcal{F} \subset \{\pm 1\}^n$ $\mathcal{X} = \{\}$, $\ell(y', y) = \mathbf{1}\{y \neq y'\} = \frac{1 - y \cdot y'}{2}$
- Since there are no x's the condition is obvious.
- Algorithm : at round t, draw $\epsilon_{t+1:n}$ then play

$$
\begin{aligned}
&2q_t(\epsilon) - 1 \\
&= \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{n} \epsilon_s f_s - \sum_{s=1}^{t-1} \mathbf{1}\{f_s \neq y_s\} - \mathbf{1}\{f_t \neq 1\} \right\} - \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{n} \epsilon_s f_s - \sum_{s=1}^{t-1} \mathbf{1}\{f_s \neq y_s\} - \mathbf{1}\{f_t \neq -1\} \right\} \\
&= \inf_{f \in \mathcal{F}} \left\{ 2 \sum_{s=t+1}^{n} \mathbf{1}\{\epsilon_s \neq f_s\} + \sum_{s=1}^{t-1} \mathbf{1}\{f_s \neq y_s\} + \mathbf{1}\{f_t \neq 1\} \right\} \\
&\qquad - \inf_{f \in \mathcal{F}} \left\{ 2 \sum_{s=t+1}^{n} \mathbf{1}\{\epsilon_s \neq f_s\} + \sum_{s=1}^{t-1} \mathbf{1}\{f_s \neq y_s\} + \mathbf{1}\{f_t \neq -1\} \right\}
\end{aligned}
$$

Solve two ERM's per round.

EXAMPLE: LINEAR PREDICTORS

- Online linear optimization, $\mathcal{F} = \{f : \|f\| \leq 1\}$, $\mathbf{D} = \{\nabla : \|\nabla\|_* \leq 1\}$
- Condition: $\exists D$ and constant C, such that, for any vector w,

$$\sup_{x_t} \mathbb{E}_{\epsilon_t}\left[\|w + 2\epsilon_t x_t\|_*\right] \leq \mathbb{E}_{x_t \sim D}\left[\|w + C x_t\|_*\right]$$

Example: Linear Predictors

- Online linear optimization, $\mathcal{F} = \{f : \|f\| \leq 1\}$, $\mathbf{D} = \{\nabla : \|\nabla\|_* \leq 1\}$
- Condition: $\exists D$ and constant C, such that, for any vector w,

$$\sup_{x_t} \mathbb{E}_{\epsilon_t}\left[\|w + 2\epsilon_t x_t\|_*\right] \leq \mathbb{E}_{x_t \sim D}\left[\|w + C x_t\|_*\right]$$

- ℓ_1^d/ℓ_∞^d : $D = \mathrm{Unif}\{\pm 1\}^d$ or any other symmetric distribution on each coordinate (Eg. normal distribution)

EXAMPLE: LINEAR PREDICTORS

- Online linear optimization, $\mathcal{F} = \{f : \|f\| \le 1\}$, $\mathbf{D} = \{\nabla : \|\nabla\|_* \le 1\}$
- Condition: $\exists D$ and constant C, such that, for any vector w,

$$\sup_{x_t} \mathbb{E}_{\epsilon_t}\left[\|w + 2\epsilon_t x_t\|_*\right] \le \mathbb{E}_{x_t \sim D}\left[\|w + C x_t\|_*\right]$$

- ℓ_1^d / ℓ_∞^d : $D = \text{Unif}\{\pm 1\}^d$ or any other symmetric distribution on each coordinate (Eg. normal distribution)
- Algorithm : Round t draw $R_t \sim N(0, (n-t) I_d)$

$$\hat{y}_t = \underset{i \in [d]}{\operatorname{argmin}} \left|\sum_{j=1}^{t} \nabla_t[i] + R_t[i]\right|$$

Example: Linear Predictors

- Online linear optimization, $\mathcal{F} = \{f : \|f\| \leq 1\}$, $\mathbf{D} = \{\nabla : \|\nabla\|_* \leq 1\}$
- Condition: $\exists D$ and constant C, such that, for any vector w,

$$\sup_{x_t} \mathbb{E}_{\epsilon_t}\left[\|w + 2\epsilon_t x_t\|_*\right] \leq \mathbb{E}_{x_t \sim D}\left[\|w + C x_t\|_*\right]$$

- ℓ_1^d/ℓ_∞^d : $D = \mathrm{Unif}\{\pm 1\}^d$ or any other symmetric distribution on each coordinate (Eg. normal distribution)
- Algorithm : Round t draw $R_t \sim N(0, (n-t)I_d)$

$$\hat{y}_t = \operatorname*{argmin}_{i \in [d]} \left|\sum_{j=1}^{t} \nabla_t[i] + R_t[i]\right|$$

- Bound : $\mathbb{E}\left[\mathrm{Reg}_n\right] \leq \frac{1}{n}\mathbf{Rel}_n(\cdot) = O\left(\sqrt{\frac{\log d}{n}}\right)$

Rough Sketch of Proof

- $w = 2C \sum_{s=t+1}^{n} \nabla_s - \sum_{s=1}^{t-1} \nabla_s$ where $\nabla_{1:t-1}$ are past losses and $\nabla_{t+1:n}$ are drawn from $\mathrm{Unif}\{-1, 1\}^d$

Rough Sketch of Proof

- $w = 2C \sum_{s=t+1}^{n} \nabla_s - \sum_{s=1}^{t-1} \nabla_s$ where $\nabla_{1:t-1}$ are past losses and $\nabla_{t+1:n}$ are drawn from $\mathrm{Unif}\{-1, 1\}^d$
- Assume $t < n - \sqrt{n}$, for last $\sqrt{n}$ rounds even if we are completely off, regret bound does not change

Rough Sketch of Proof

- $w = 2C\sum_{s=t+1}^{n} \nabla_s - \sum_{s=1}^{t-1} \nabla_s$ where $\nabla_{1:t-1}$ are past losses and $\nabla_{t+1:n}$ are drawn from $\mathrm{Unif}\{-1,1\}^d$
- Assume $t < n - \sqrt{n}$, for last $\sqrt{n}$ rounds even if we are completely off, regret bound does not change
- Hence w can be seen as vector $-\sum_{s=1}^{t-1} \nabla_s$ where each coordinate is perturbed by $2C\sum_{s=t+1}^{n} \nabla_s$

Rough Sketch of Proof

- $w = 2C\sum_{s=t+1}^{n} \nabla_s - \sum_{s=1}^{t-1} \nabla_s$ where $\nabla_{1:t-1}$ are past losses and $\nabla_{t+1:n}$ are drawn from $\mathrm{Unif}\{-1,1\}^d$
- Assume $t < n - \sqrt{n}$, for last $\sqrt{n}$ rounds even if we are completely off, regret bound does not change
- Hence w can be seen as vector $-\sum_{s=1}^{t-1} \nabla_s$ where each coordinate is perturbed by $2C\sum_{s=t+1}^{n} \nabla_s$
- With very high probability, if i^* and j^* are top two coordinates of w, $|w[i^*]| - |w[j^*]| > 4$, hence, with high probability,

$$\sup_{x_t \in [-1,1]^d} \mathbb{E}_{\epsilon_t}\left[\|w + 2\epsilon_t x_t\|_\infty\right] = \sup_{x_t \in [-1,1]^d} \mathbb{E}_{\epsilon_t}\left[|w[i^*] + 2\epsilon_t x_t[i^*]|\right]$$

$$= \mathbb{E}_{\epsilon_t}\left[|w[i^*] + 2\epsilon_t|\right] = \mathbb{E}_{x_t \sim D}\left[\|w + 2\epsilon_t x_t\|_\infty\right]$$

Rough Sketch of Proof

- $w = 2C\sum_{s=t+1}^{n} \nabla_s - \sum_{s=1}^{t-1} \nabla_s$ where $\nabla_{1:t-1}$ are past losses and $\nabla_{t+1:n}$ are drawn from $\text{Unif}\{-1,1\}^d$
- Assume $t < n - \sqrt{n}$, for last $\sqrt{n}$ rounds even if we are completely off, regret bound does not change
- Hence w can be seen as vector $-\sum_{s=1}^{t-1} \nabla_s$ where each coordinate is perturbed by $2C\sum_{s=t+1}^{n} \nabla_s$
- With very high probability, if i^* and j^* are top two coordinates of w, $|w[i^*]| - |w[j^*]| > 4$, hence, with high probability,

$$\sup_{x_t \in [-1,1]^d} \mathbb{E}_{\epsilon_t}\left[\|w + 2\epsilon_t x_t\|_\infty\right] = \sup_{x_t \in [-1,1]^d} \mathbb{E}_{\epsilon_t}\left[|w[i^*] + 2\epsilon_t x_t[i^*]|\right]$$
$$= \mathbb{E}_{\epsilon_t}\left[|w[i^*] + 2\epsilon_t|\right] = \mathbb{E}_{x_t \sim D}\left[\|w + 2\epsilon_t x_t\|_\infty\right]$$

- In general we don't need this high probability stuff, we can directly prove the condition, just need to check cases.

Rough Sketch of Proof

- Why update of form $\hat{y}_t = \mathrm{argmin}_{i \in [d]} \left| \sum_{j=1}^{t} \nabla_j[i] + R_t[i] \right|$
- To see this, note that the algorithm we need is originally of form,

$$\hat{y}_t = \underset{\hat{y} \in \mathcal{F}}{\mathrm{argmin}} \sup_{\nabla_t} \left\{ \langle \hat{y}, \nabla_t \rangle + \sup_{f \in \mathcal{F}} \left\{ \langle f, -R_t \rangle - \left\langle f, \sum_{s=1}^{t} \nabla_s \right\rangle \right\} \right\}$$

Rough Sketch of Proof

- Why update of form $\hat{y}_t = \text{argmin}_{i\in[d]} \left|\sum_{j=1}^{t} \nabla_j[i] + R_t[i]\right|$
- To see this, note that the algorithm we need is originally of form,

$$\hat{y}_t = \underset{\hat{y}\in\mathcal{F}}{\text{argmin}} \sup_{\nabla_t} \left\{ \langle \hat{y}, \nabla_t \rangle + \sup_{f\in\mathcal{F}} \left\{ \langle f, -R_t \rangle - \left\langle f, \sum_{s=1}^{t} \nabla_s \right\rangle \right\} \right\}$$

$$= \underset{\hat{y}\in\mathcal{F}}{\text{argmin}} \sup_{f\in\mathcal{F}} \left\{ \sup_{\nabla_t} \langle \hat{y} - f, \nabla_t \rangle + \left\langle f, -R_t - \sum_{s=1}^{t-1} \nabla_s \right\rangle \right\}$$

Rough Sketch of Proof

- Why update of form $\hat{y}_t = \mathrm{argmin}_{i \in [d]} \left| \sum_{j=1}^{t} \nabla_j[i] + R_t[i] \right|$
- To see this, note that the algorithm we need is originally of form,

$$\hat{y}_t = \operatorname*{argmin}_{\hat{y} \in \mathcal{F}} \sup_{\nabla_t} \left\{ \langle \hat{y}, \nabla_t \rangle + \sup_{f \in \mathcal{F}} \left\{ \langle f, -R_t \rangle - \left\langle f, \sum_{s=1}^{t} \nabla_s \right\rangle \right\} \right\}$$

$$= \operatorname*{argmin}_{\hat{y} \in \mathcal{F}} \sup_{f \in \mathcal{F}} \left\{ \sup_{\nabla_t} \langle \hat{y} - f, \nabla_t \rangle + \left\langle f, -R_t - \sum_{s=1}^{t-1} \nabla_s \right\rangle \right\}$$

$$= \operatorname*{argmin}_{\hat{y} \in \mathcal{F}} \sup_{f \in \mathcal{F}} \left\{ \| \hat{y} - f \|_\infty - \left\langle f, R_t + \sum_{s=1}^{t-1} \nabla_s \right\rangle \right\}$$

Example: Linear Predictors

- Online linear optimization, $\mathcal{F} = \{f : \|f\| \le 1\}$, $\mathbf{D} = \{\nabla : \|\nabla\|_* \le 1\}$
- Condition: $\exists D$ and constant C, such that, for any vector w,

$$\sup_{x_t} \mathbb{E}_{\epsilon_t}\left[\|w + 2\epsilon_t x_t\|_*\right] \le \mathbb{E}_{x_t \sim D}\left[\|w + C x_t\|_*\right]$$

- ℓ_2/ℓ_2 : $D = \text{Unif}\{\text{unit sphere}\}$ or normalized Gaussian distribution
- Algorithm : Round t draw $R_t \sim N(0, (n-t)I_d)/\sqrt{d}$

$$\hat{y}_t = \operatorname*{argmin}_{f : \|f\|_2 \le 1} \left\langle f, \sum_{j=1}^{t} \nabla_t + R_t \right\rangle$$

- Bound : $\mathbb{E}\left[\text{Reg}_n\right] \le \frac{1}{n}\mathbf{Rel}_n(\cdot) = O\left(\sqrt{\frac{1}{n}}\right)$

EXAMPLE: FINITE EXPERTS

- Very similar to ℓ_1/ℓ_∞, think about subtracting -1 from every loss, makes no difference for regret
- But then ℓ_1/ℓ_∞ is same as finite experts
- Algorithm : Round t draw $R_t \sim N(0, (n-t)I_{|\mathcal{F}|})$

$$\hat{y}_t = \operatorname*{argmin}_{i \in [d]} \sum_{j=1}^{t} \ell(i, z_t) + R_t[i]$$

- Bound : $\mathbb{E}\left[\mathrm{Reg}_n\right] \leq \frac{1}{n}\mathbf{Rel}_n\left(\cdot\right) = O\left(\sqrt{\frac{\log|\mathcal{F}|}{n}}\right)$

Example: Online Shortest Path

- Graph $G = (V, E)$, source node S and destination node D.

Example: Online Shortest Path

- Graph $G = (V, E)$, source node S and destination node D.
- Every round, we need to pick a path from S to D

Example: Online Shortest Path

- Graph $G = (V, E)$, source node S and destination node D.
- Every round, we need to pick a path from S to D
- Adversary picks a delay on every edge $W : E \mapsto [0, 1]$

EXAMPLE: ONLINE SHORTEST PATH

- Graph $G = (V, E)$, source node S and destination node D.
- Every round, we need to pick a path from S to D
- Adversary picks a delay on every edge $W : E \mapsto [0, 1]$
- Learner suffers delay on path chosen which is sum of delays on edges of the path

EXAMPLE: ONLINE SHORTEST PATH

- Graph $G = (V, E)$, source node S and destination node D.
- Every round, we need to pick a path from S to D
- Adversary picks a delay on every edge $W : E \mapsto [0, 1]$
- Learner suffers delay on path chosen which is sum of delays on edges of the path
- Experts bound $|E|\sqrt{\frac{|V| \log |V|}{n}}$

Example: Online Shortest Path

- Graph $G = (V, E)$, source node S and destination node D.
- Every round, we need to pick a path from S to D
- Adversary picks a delay on every edge $W : E \mapsto [0, 1]$
- Learner suffers delay on path chosen which is sum of delays on edges of the path
- Experts bound $|E|\sqrt{\frac{|V|\log|V|}{n}}$
- However naive time complexity $O(\#\text{paths})$

EXAMPLE: ONLINE SHORTEST PATH

- Can view it as a different online linear optimization problem
- $\mathcal{F} = \{f \in \{0,1\}^{|E|} : f \text{ is a path}\}$
- $\mathbf{D} = [0,1]^{|E|}$ the delays on each edge.

Example: Online Shortest Path

- Can view it as a different online linear optimization problem
- $\mathcal{F} = \{f \in \{0,1\}^{|E|} : f \text{ is a path}\}$
- $\mathbf{D} = [0,1]^{|E|}$ the delays on each edge.
- Random playout condition satisfied by distribution $D = N(0,1)$

Example: Online Shortest Path

- Can view it as a different online linear optimization problem
- $\mathcal{F} = \{f \in \{0,1\}^{|E|} : f \text{ is a path}\}$
- $\mathbf{D} = [0,1]^{|E|}$ the delays on each edge.
- Random playout condition satisfied by distribution $D = N(0,1)$
- Algorithm: Draw $R_t \sim N(0,(n-t)I_{|E|})$,

$$\text{path}_t = \operatorname*{argmin}_{f \in \mathcal{F}} \left\langle f, \sum_{j=1}^{t-1} \nabla_j + R_t \right\rangle$$

Example: Online Shortest Path

- Can view it as a different online linear optimization problem
- $\mathcal{F} = \{f \in \{0,1\}^{|E|} : f \text{ is a path}\}$
- $\mathbf{D} = [0,1]^{|E|}$ the delays on each edge.
- Random playout condition satisfied by distribution $D = N(0,1)$
- Algorithm: Draw $R_t \sim N(0,(n-t)I_{|E|})$,

$$\text{path}_t = \operatorname*{argmin}_{f \in \mathcal{F}} \left\langle f, \sum_{j=1}^{t-1} \nabla_j + R_t \right\rangle$$

- That is solve shortest path algorithm with delay on edge $e \in E$ given by $\sum_{j=1}^{t-1} \nabla_j[e] + R_t[e]$
- Can be solves in poly-time using Bellman-ford algorithm.

Learning with Non-repeated entries

For $t = 1$ to $|\mathcal{X}|$

Adversary picks $x_t \in \mathcal{X} \setminus \{x_1, \ldots, x_{t-1}\}$
Learner predicts $q_t \in \Delta(\mathcal{Y})$
Adversary picks $y_t \in \mathcal{Y}$
Learner draws $\hat{y}_t \sim q_t$ and suffers loss $\ell(\hat{y}_t, y_t)$

End

Regret :

$$\mathrm{Reg}_{|\mathcal{X}|} = \sum_{t=1}^{|\mathcal{X}|} \ell(\hat{y}_t, y_t) - \inf_{f \in \mathcal{F}} \sum_{t=1}^{|\mathcal{X}|} \ell(f(x_t), y_t)$$

Learning with Non-repeated entries

- For convex Lipschitz loss and binary loss, the symmetrization idea just goes through, only on each path, no node is repeated.
- Sequential Rademacher relaxation:

$$\mathbf{Rad}_{|\mathcal{X}|}(x_{1:t}, y_{1:t}) = \sup_{\mathbf{x}} \mathbb{E}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(\mathbf{x}_{s-t}(\epsilon)) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$

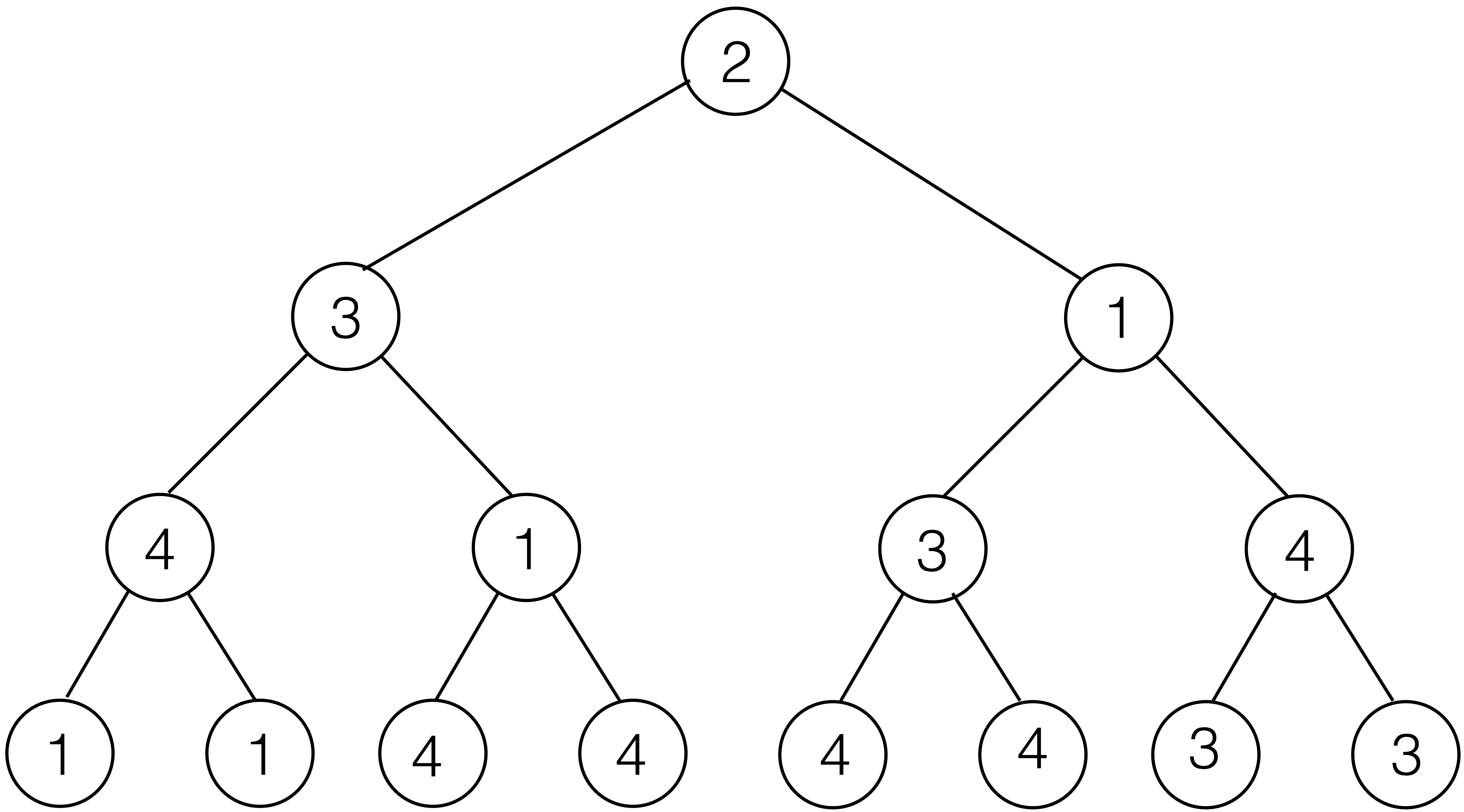
where $\mathbf{x}$ is a tree with values in $\mathcal{X} \setminus \{x_1, \ldots, x_t\}$ with no node repeated on any path.

Learning with Non-repeated entries

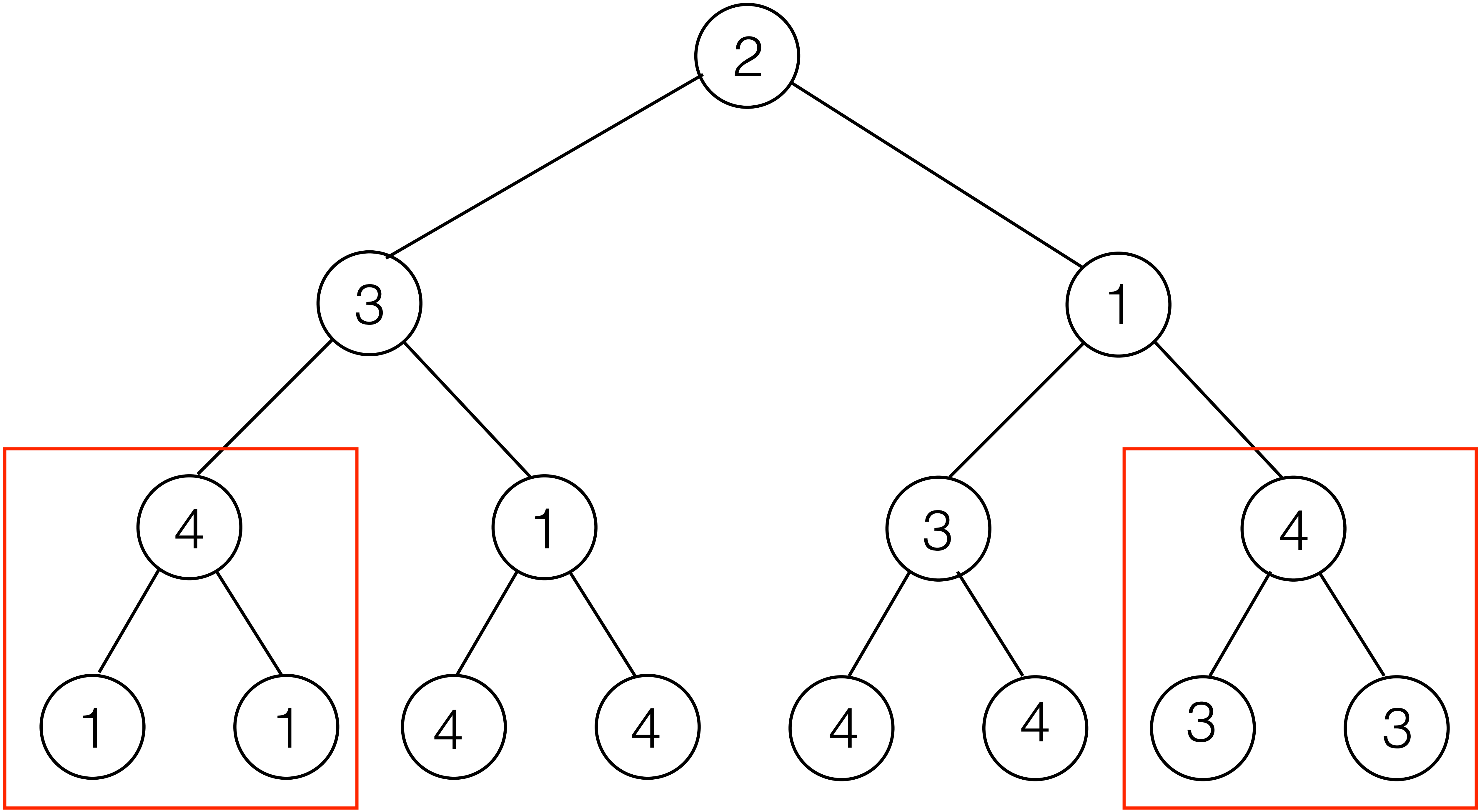
- Inductively we can show that:

$$\mathbf{Rad}_{|\mathcal{X}|}(x_{1:t}, y_{1:t}) = \mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(x_s) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$

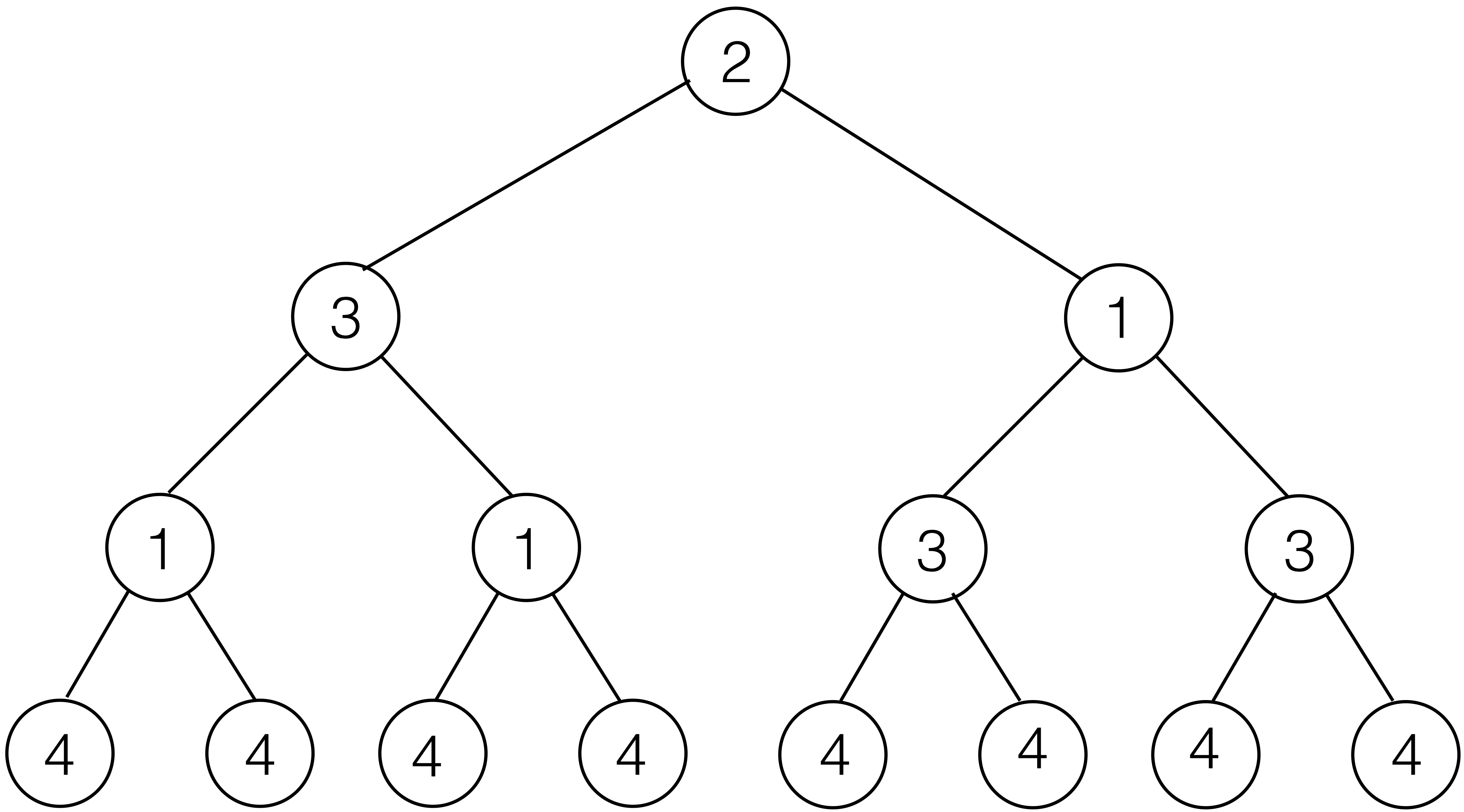
where $x_{t+1}, \ldots, x_{|\mathcal{X}|}$ are elements from $\mathcal{X} \setminus \{x_1, \ldots, x_t\}$ in any order non-repeated.



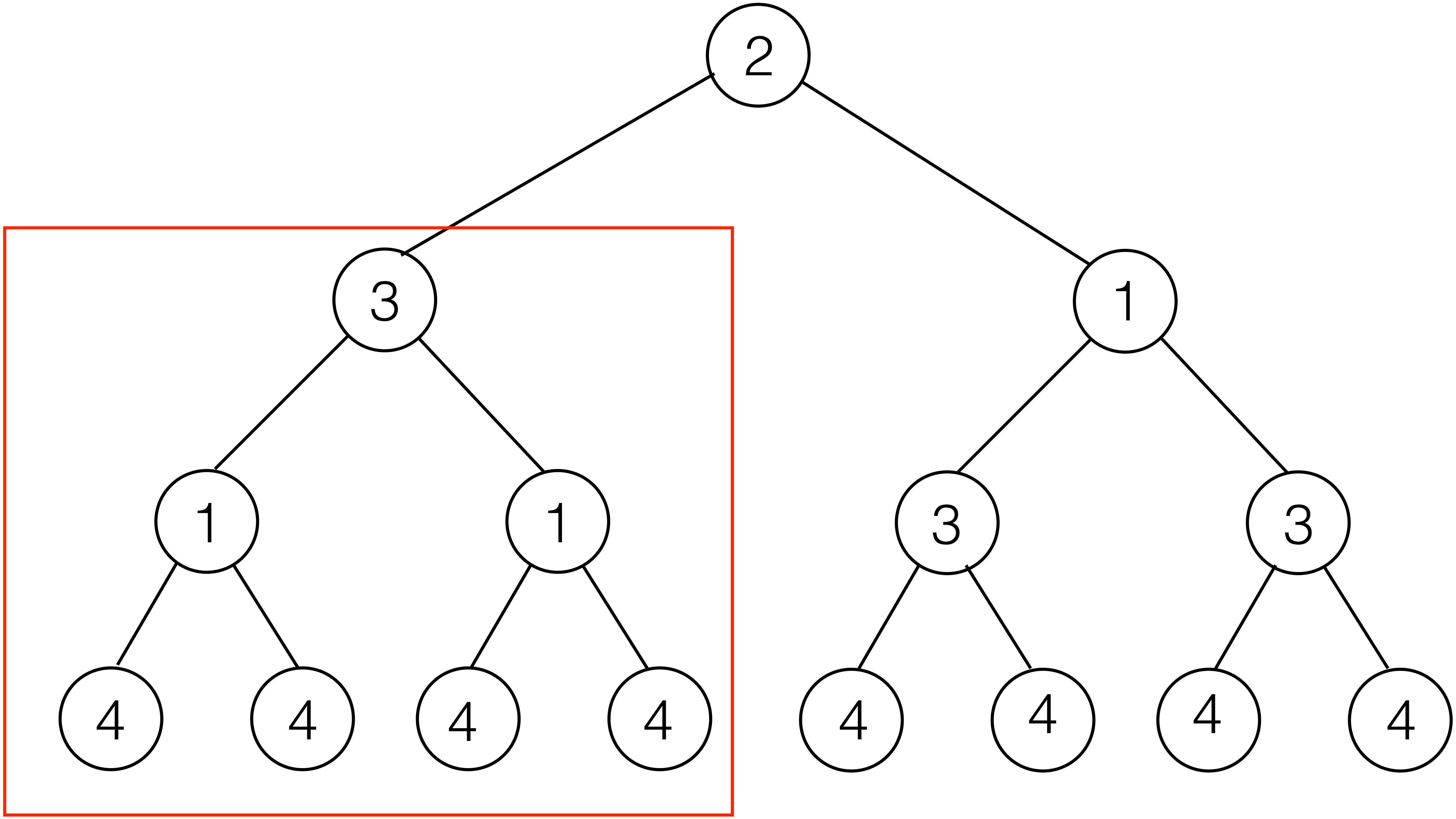
$$\mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f\in\mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(\mathbf{x}_{s-t}(\epsilon)) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$



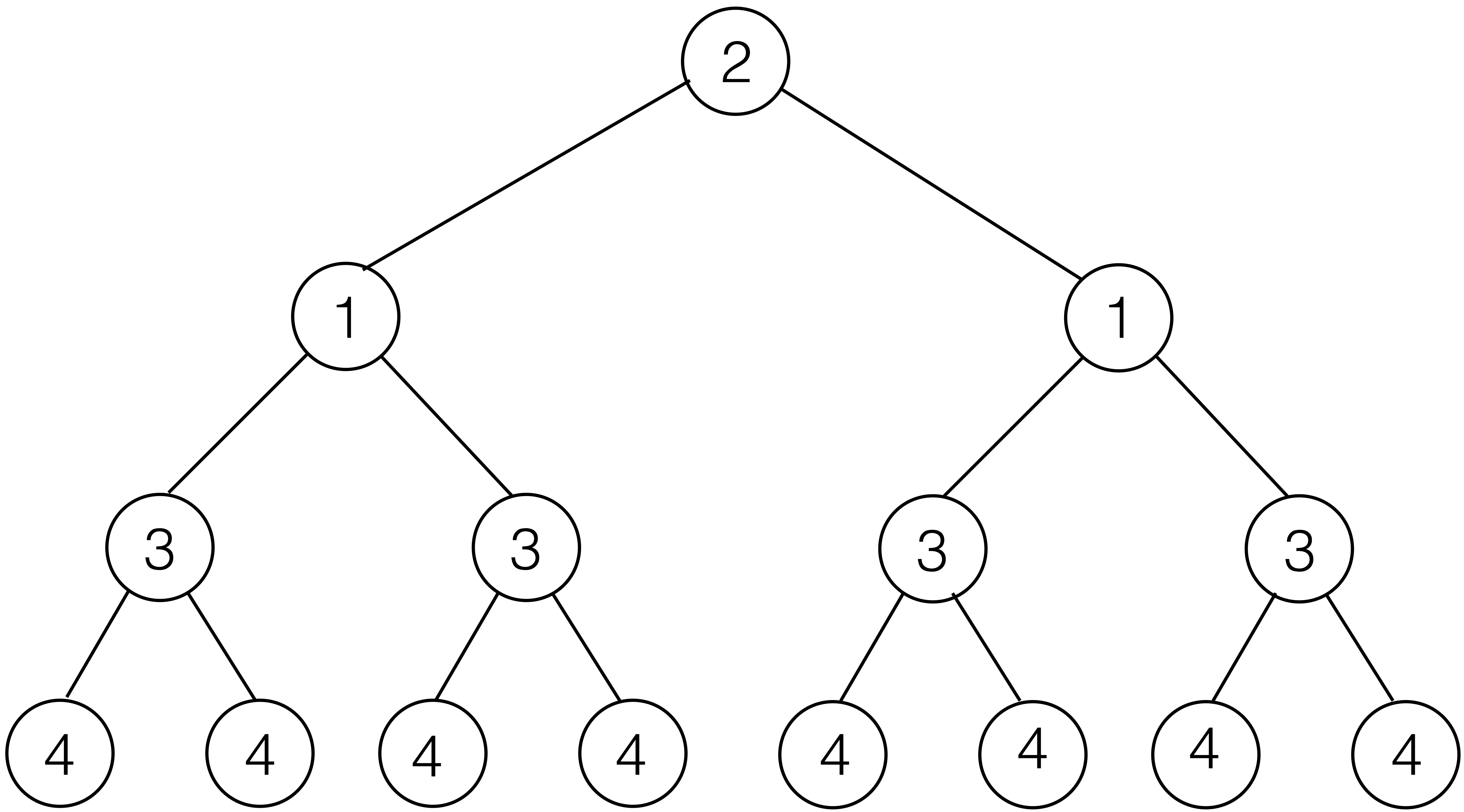
$$\mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f\in\mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(\mathbf{x}_{s-t}(\epsilon)) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$



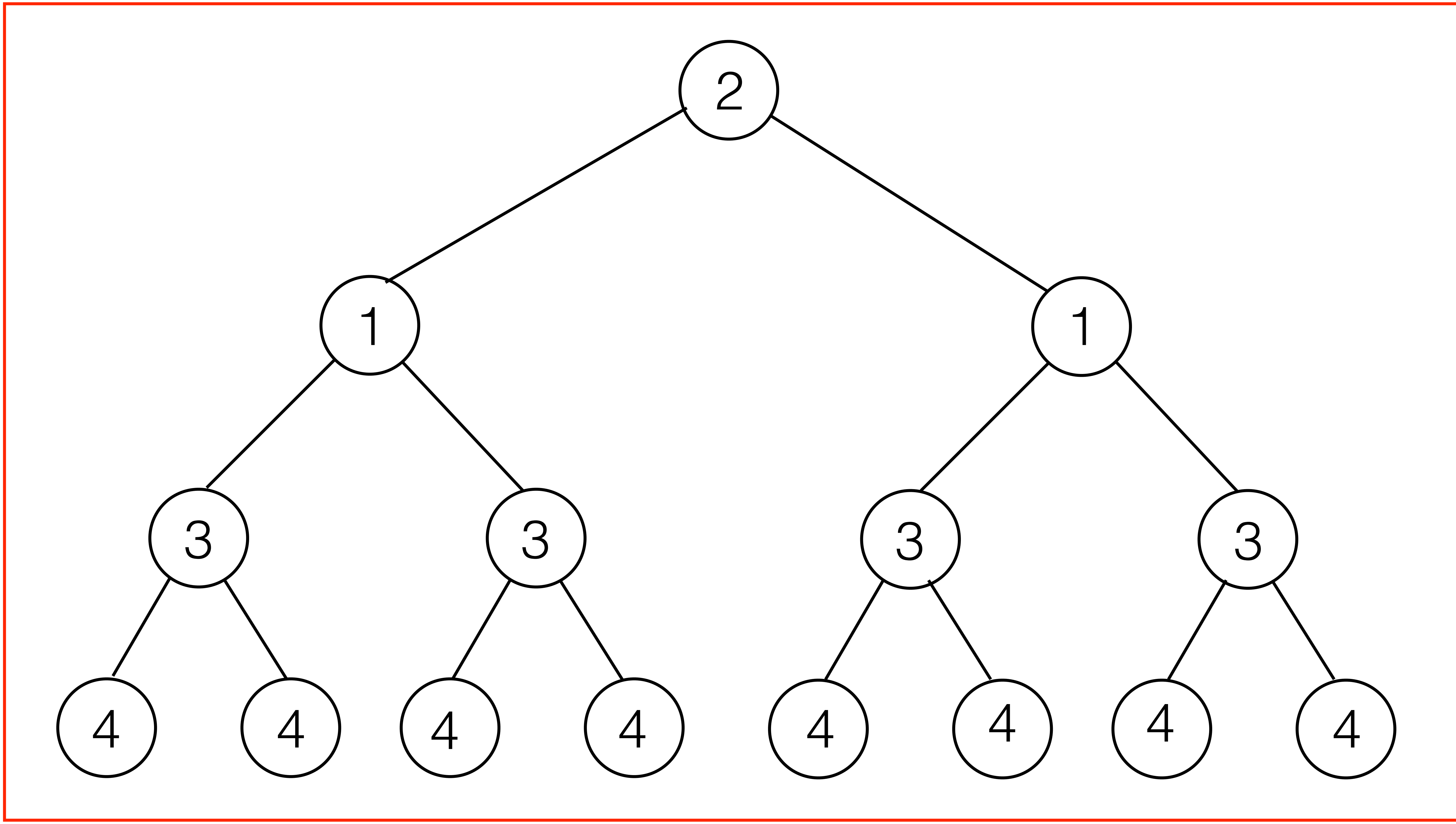
$$\mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f\in\mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(\mathbf{x}_{s-t}(\epsilon)) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$



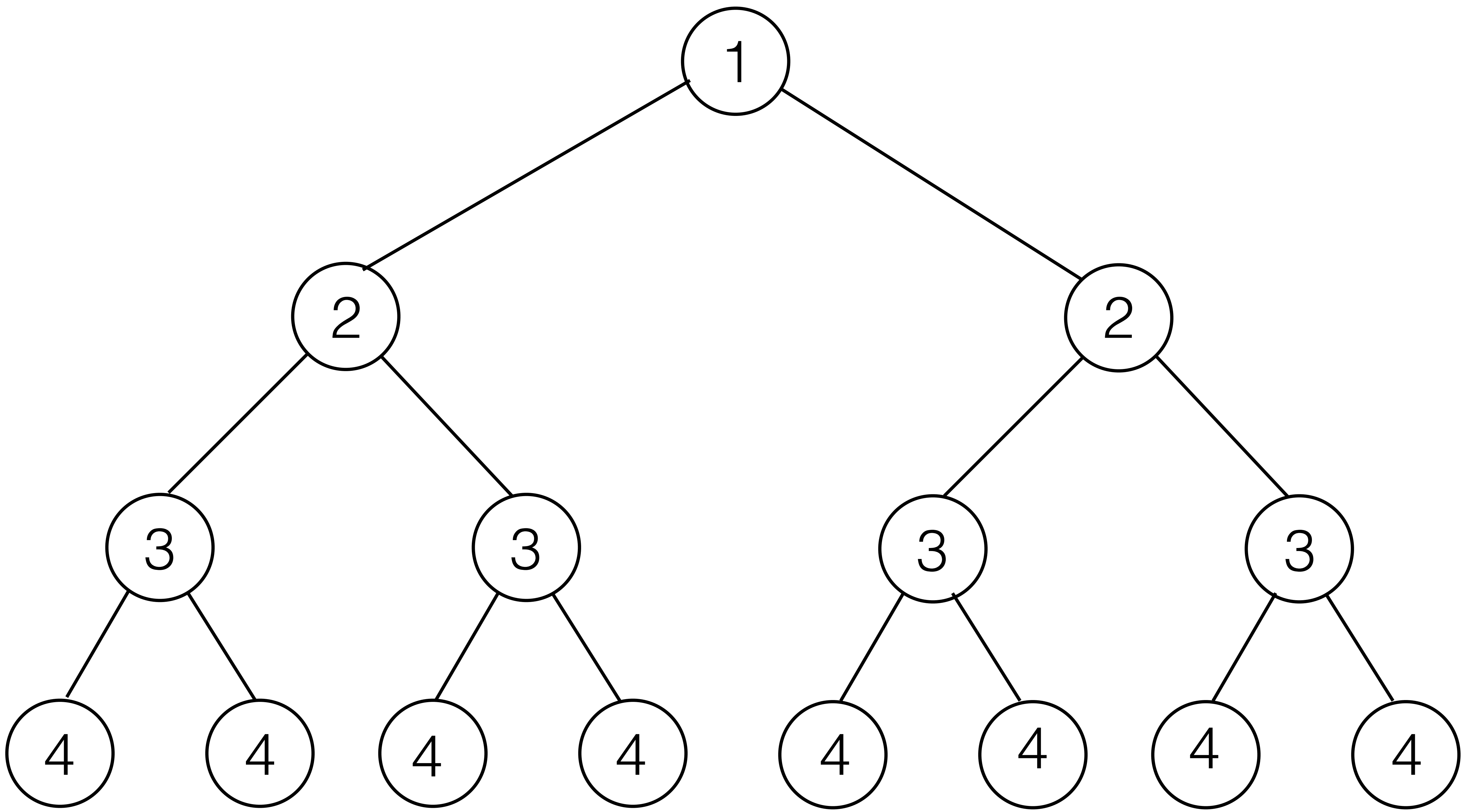
$$\mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f\in\mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(\mathbf{x}_{s-t}(\epsilon)) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$



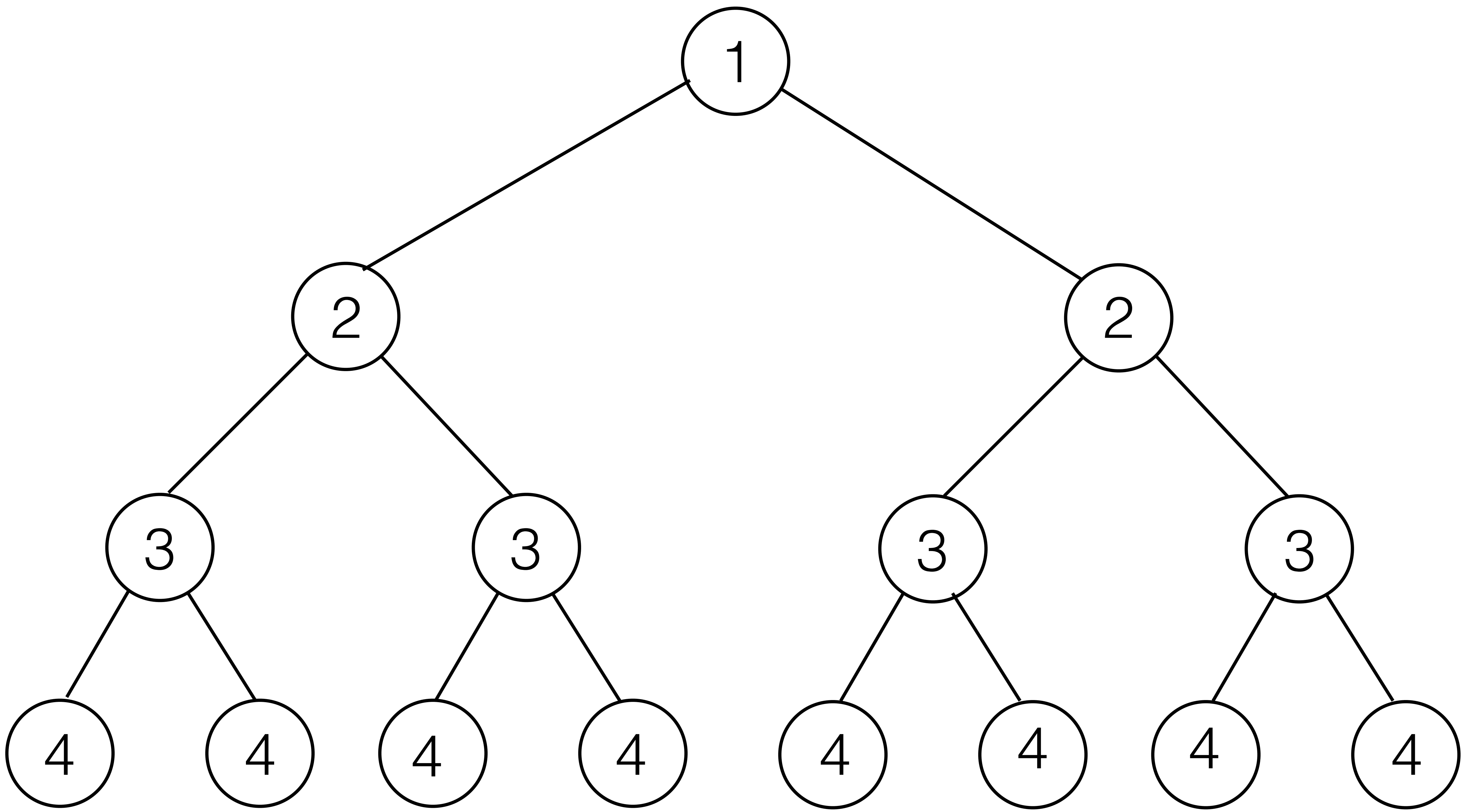
$$\mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(\mathbf{x}_{s-t}(\epsilon)) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$



$$\mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(\mathbf{x}_{s-t}(\epsilon)) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$



$$\mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(\mathbf{x}_{s-t}(\epsilon)) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$



$$= \mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(x_s) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$

Learning with Non-repeated entries

- Inductively we can show that:

$$\mathbf{Rad}_{|\mathcal{X}|}(x_{1:t}, y_{1:t}) = \mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f\in\mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(x_s) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$

where $x_{t+1}, \ldots, x_{|\mathcal{X}|}$ are elements from $\mathcal{X} \setminus \{x_1, \ldots, x_t\}$ in any order non-repeated.

Learning with Non-repeated entries

- Inductively we can show that:

$$\mathbf{Rad}_{|\mathcal{X}|}\left(x_{1:t}, y_{1:t}\right) = \mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f\left(x_s\right) - \sum_{s=1}^{t} \ell(f\left(x_s\right), y_s) \right\}$$

where $x_{t+1}, \ldots, x_{|\mathcal{X}|}$ are elements from $\mathcal{X} \setminus \{x_1, \ldots, x_t\}$ in any order non-repeated.

- We can use $\mathbf{Rel}_{|\mathcal{X}|}\left(x_{1:t}, y_{1:t}\right) = \mathbf{Rad}_{|\mathcal{X}|}\left(x_{1:t}, y_{1:t}\right)$ as a relaxation

Learning with Non-repeated entries

- Inductively we can show that:

$$\mathbf{Rad}_{|\mathcal{X}|}(x_{1:t}, y_{1:t}) = \mathop{\mathbb{E}}_{\epsilon_{t+1:n}} \sup_{f\in\mathcal{F}} \left\{ \sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(x_s) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\}$$

  where $x_{t+1}, \ldots, x_{|\mathcal{X}|}$ are elements from $\mathcal{X} \setminus \{x_1, \ldots, x_t\}$ in any order non-repeated.

- We can use $\mathbf{Rel}_{|\mathcal{X}|}(x_{1:t}, y_{1:t}) = \mathbf{Rad}_{|\mathcal{X}|}(x_{1:t}, y_{1:t})$ as a relaxation
- Condition satisfied trivially, with constant 1,

$$\sup_{x_t \in \mathcal{X}\setminus\{x_1,\ldots,x_{t-1}\}} \mathbb{E}_{\epsilon_t}\left[\sup_{f\in\mathcal{F}} \left\{ 2\sum_{s=t+1}^{|\mathcal{X}|} \epsilon_s f(x_s) + 2\epsilon_t f(x_t) - \sum_{s=1}^{t-1} \ell(f(x_s), y_s) \right\}\right]$$

$$= \mathbb{E}_{\epsilon_t}\left[\sup_{f\in\mathcal{F}} \left\{ 2\sum_{s=t}^{|\mathcal{X}|} \epsilon_s f(x_s) - \sum_{s=1}^{t-1} \ell(f(x_s), y_s) \right\}\right]$$

  because the sum $2\sum_{s=t}^{n} \epsilon_s f(x_s)$ is independent of order.

Learning with Non-repeated entries

- Algorithm: Fix some order over elements of $\mathcal{X}$. On each round t, draw $\epsilon_{t+1},\ldots,\epsilon_{|\mathcal{X}|}$.
- Solve

$$q_t = \operatorname*{argmin}_{q\in\Delta(\mathcal{Y})} \sup_{y_t} \left\{ \mathop{\mathbb{E}}_{\hat{y}_t\sim q_t}\left[\ell(\hat{y}_t, y_t)\right] + \sup_{f\in\mathcal{F}} \left\{ 2\sum_{s=t+1}^{n} \epsilon_s f(x_s) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\} \right\}$$

- Bound : $\mathbb{E}\left[\mathrm{Reg}_n\right] \le \mathcal{R}_n^{stat}(\mathcal{F})$

Learning with Non-repeated entries

- Algorithm: Fix some order over elements of $\mathcal{X}$. On each round t, draw $\epsilon_{t+1}, \ldots, \epsilon_{|\mathcal{X}|}$.
- Solve

$$q_t = \underset{q \in \Delta(\mathcal{Y})}{\operatorname{argmin}} \sup_{y_t} \left\{ \mathop{\mathbb{E}}_{\hat{y}_t \sim q_t} \left[\ell(\hat{y}_t, y_t) \right] + \sup_{f \in \mathcal{F}} \left\{ 2 \sum_{s=t+1}^{n} \epsilon_s f(x_s) - \sum_{s=1}^{t} \ell(f(x_s), y_s) \right\} \right\}$$

- Bound : $\mathbb{E}\left[\mathrm{Reg}_n\right] \leq \mathcal{R}_n^{stat}(\mathcal{F})$
- Example: binary classification

$$q_t = \frac{1}{2} + \frac{1}{2} \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{n} \epsilon_s f(x_s) + \frac{1}{2} \sum_{s=1}^{t-1} y_s f(x_s) + \frac{1}{2} f(x_t) \right\}$$
$$- \frac{1}{2} \sup_{f \in \mathcal{F}} \left\{ \sum_{s=t+1}^{n} \epsilon_s f(x_s) + \frac{1}{2} \sum_{s=1}^{t-1} y_s f(x_s) - \frac{1}{2} f(x_t) \right\}$$

Online Collaborative Filtering

for $t = 1$ to n

Online Collaborative Filtering

for $t = 1$ to n

Entry to predict $x_t =$ 

Online Collaborative Filtering

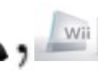
for $t = 1$ to n

Entry to predict $x_t = ($, $)$

Learner picks $\hat{y}_t \in [-1, 1]$

Online Collaborative Filtering

for $t = 1$ to n

Entry to predict $x_t = ($, $)$

Learner picks $\hat{y}_t \in [-1, 1]$

True rating $y_t \in \{\pm 1\}$ revealed

Learner suffers loss $|\hat{y}_t - y_t|$

Online Collaborative Filtering

for $t = 1$ to n

Entry to predict $x_t = (\quad,\quad)$

Learner picks $\hat{y}_t \in [-1, 1]$

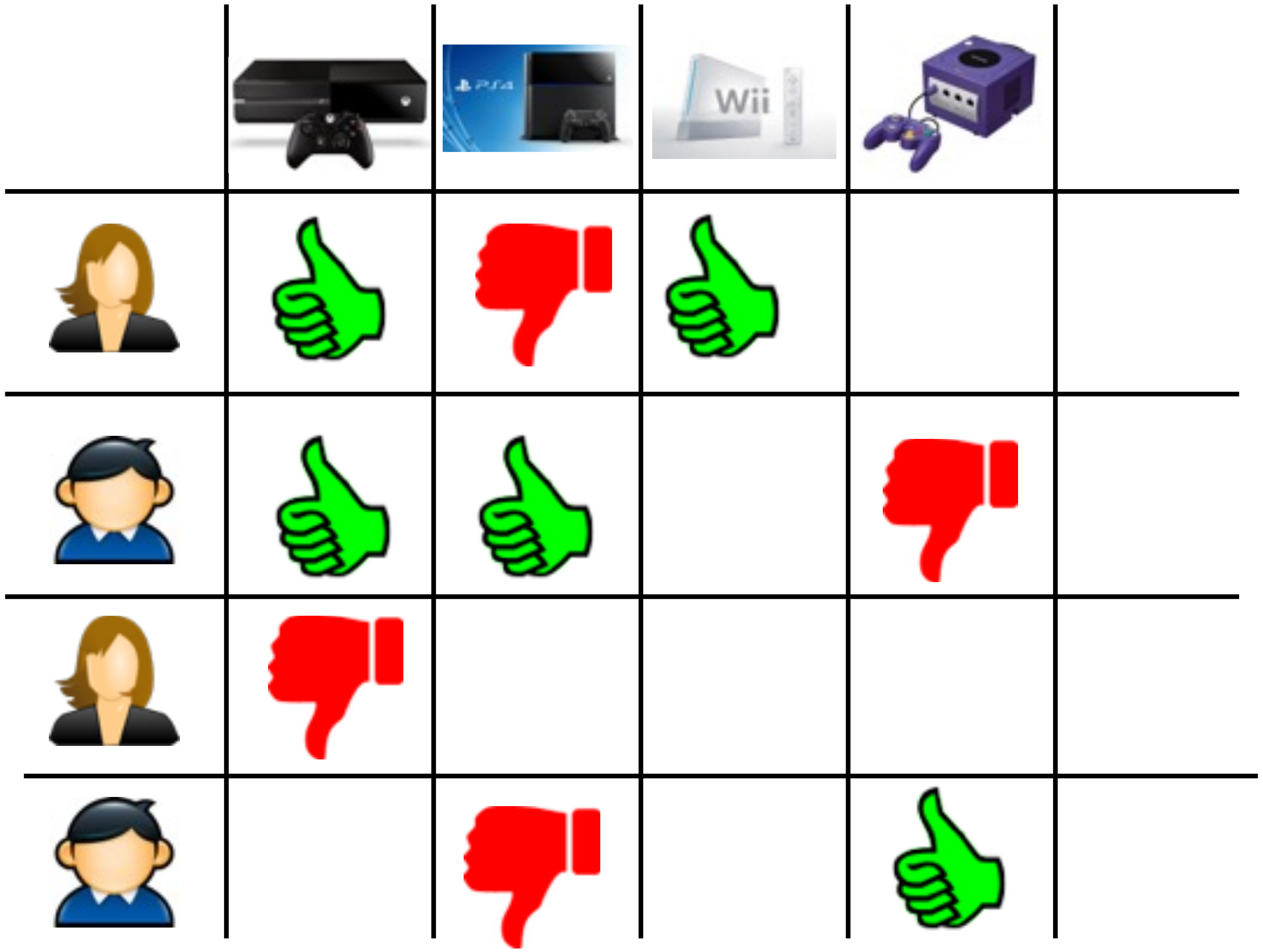
True rating $y_t \in \{\pm 1\}$ revealed

Learner suffers loss $|\hat{y}_t - y_t|$

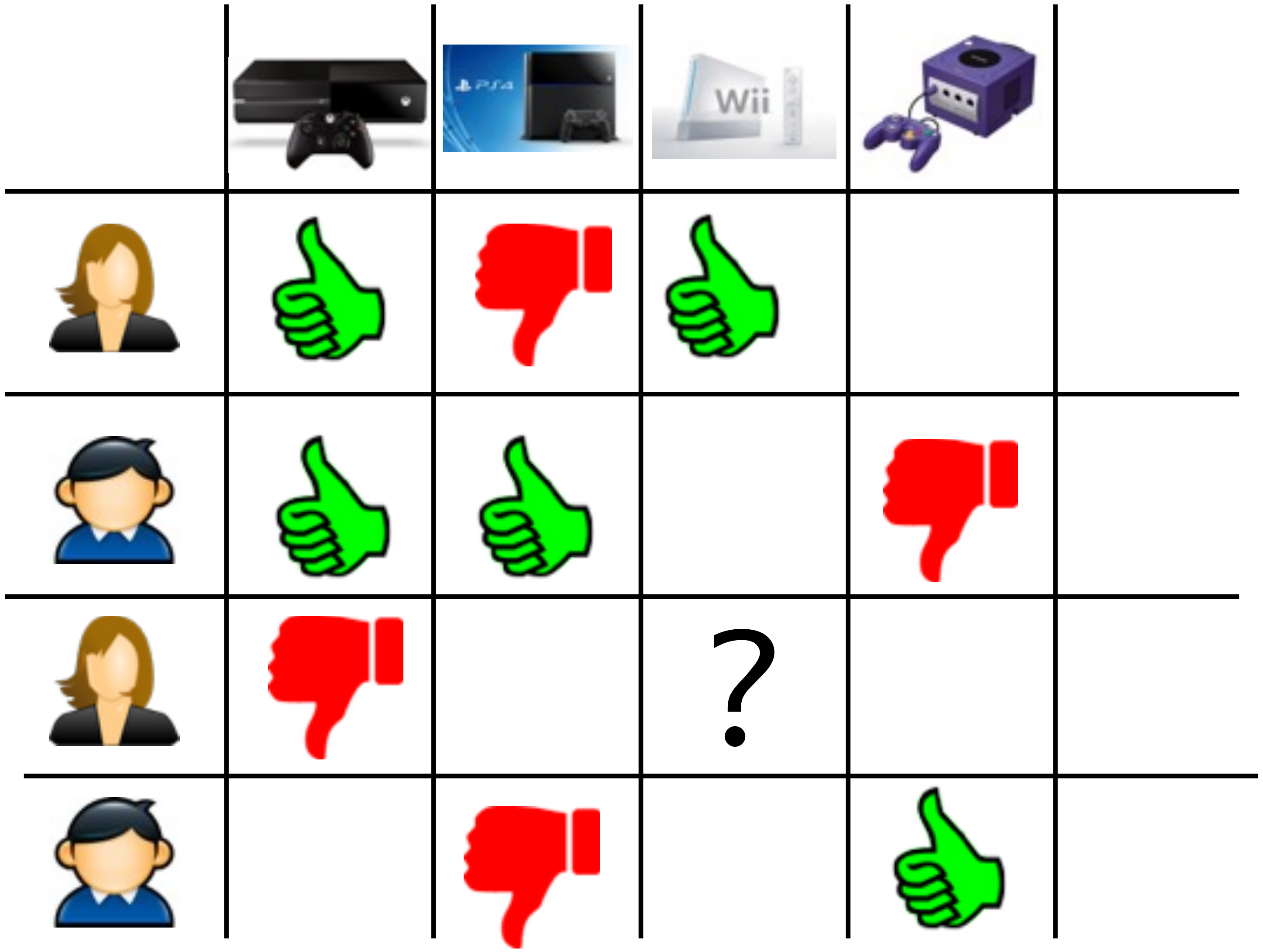
$$\mathrm{Reg}_n := \frac{1}{n}\sum_{t=1}^{n} |\hat{y}_t - y_t| - \inf_{M:\|M\|\le B} \frac{1}{n}\sum_{t=1}^{n} |M[x_t] - y_t|$$

$(\|\cdot\| : \text{ trace norm})$

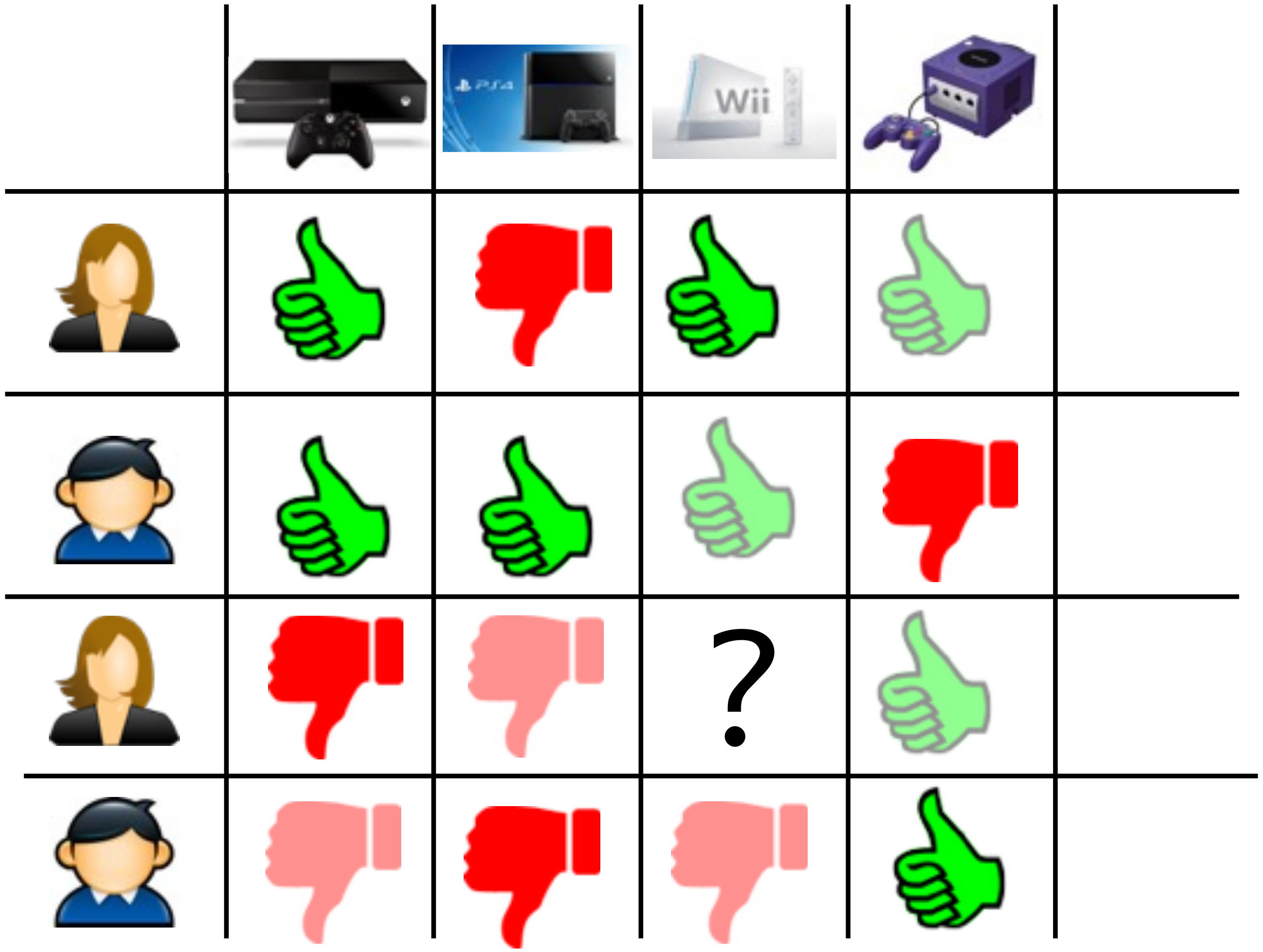
Online Collaborative Filtering



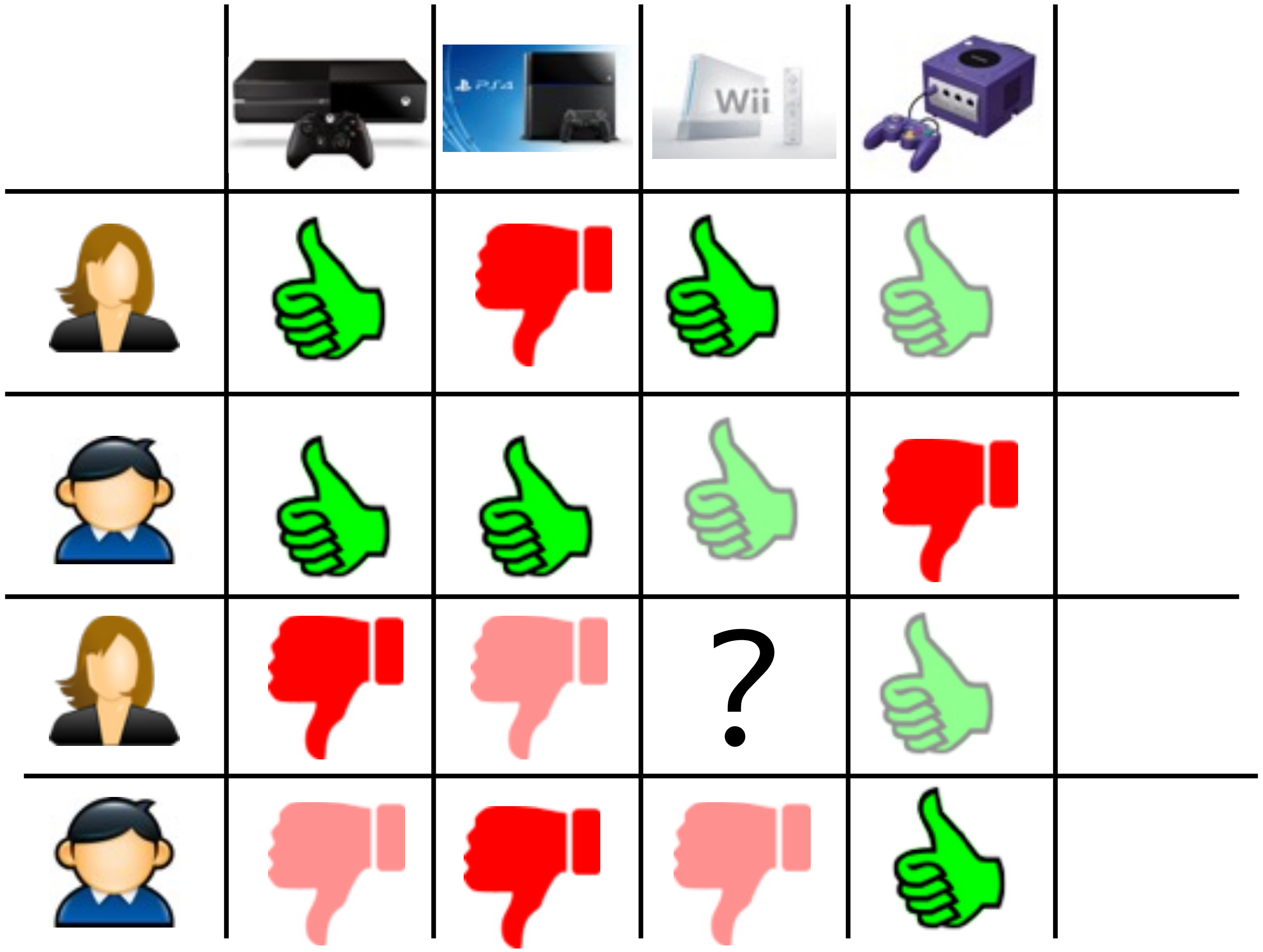
Online Collaborative Filtering



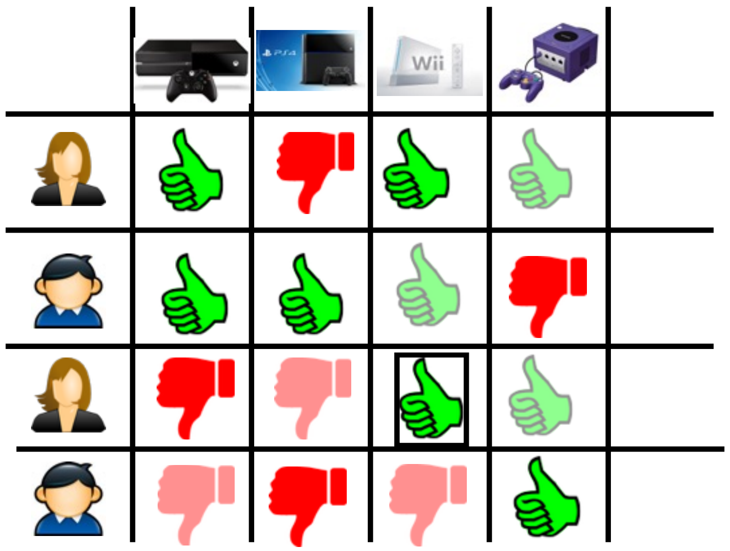
Online Collaborative Filtering

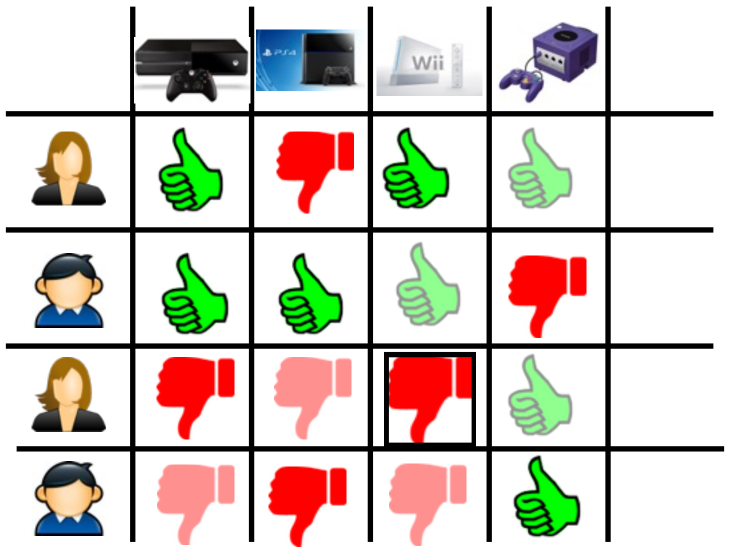


Online Collaborative Filtering

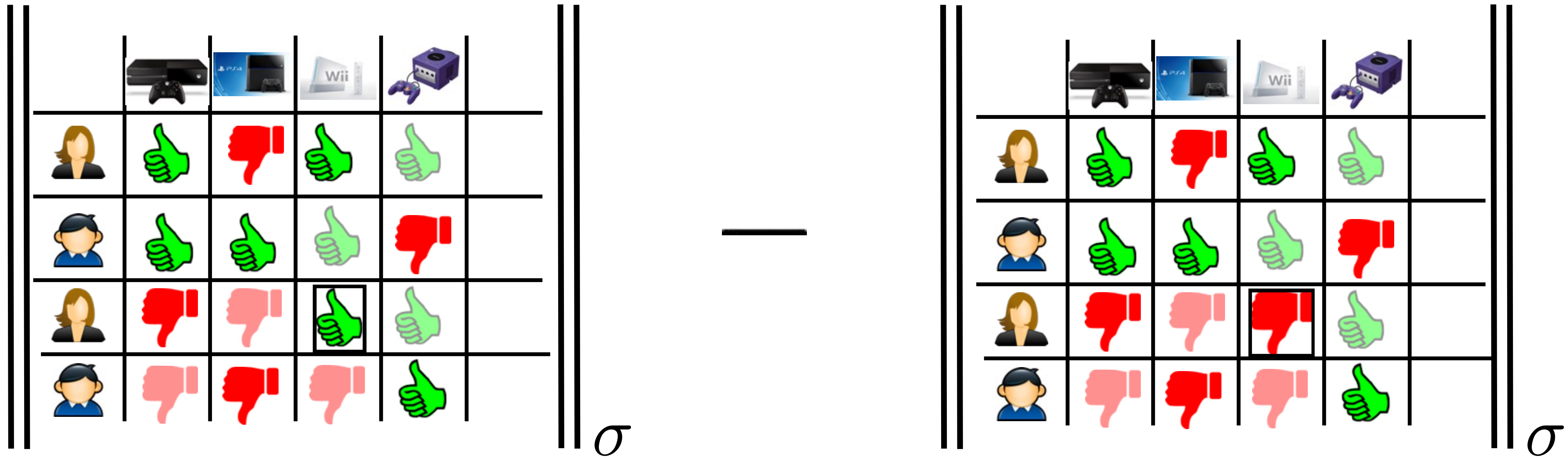


Online Collaborative Filtering





Online Collaborative Filtering



Online Collaborative Filtering

- M users and N products, regret bound :

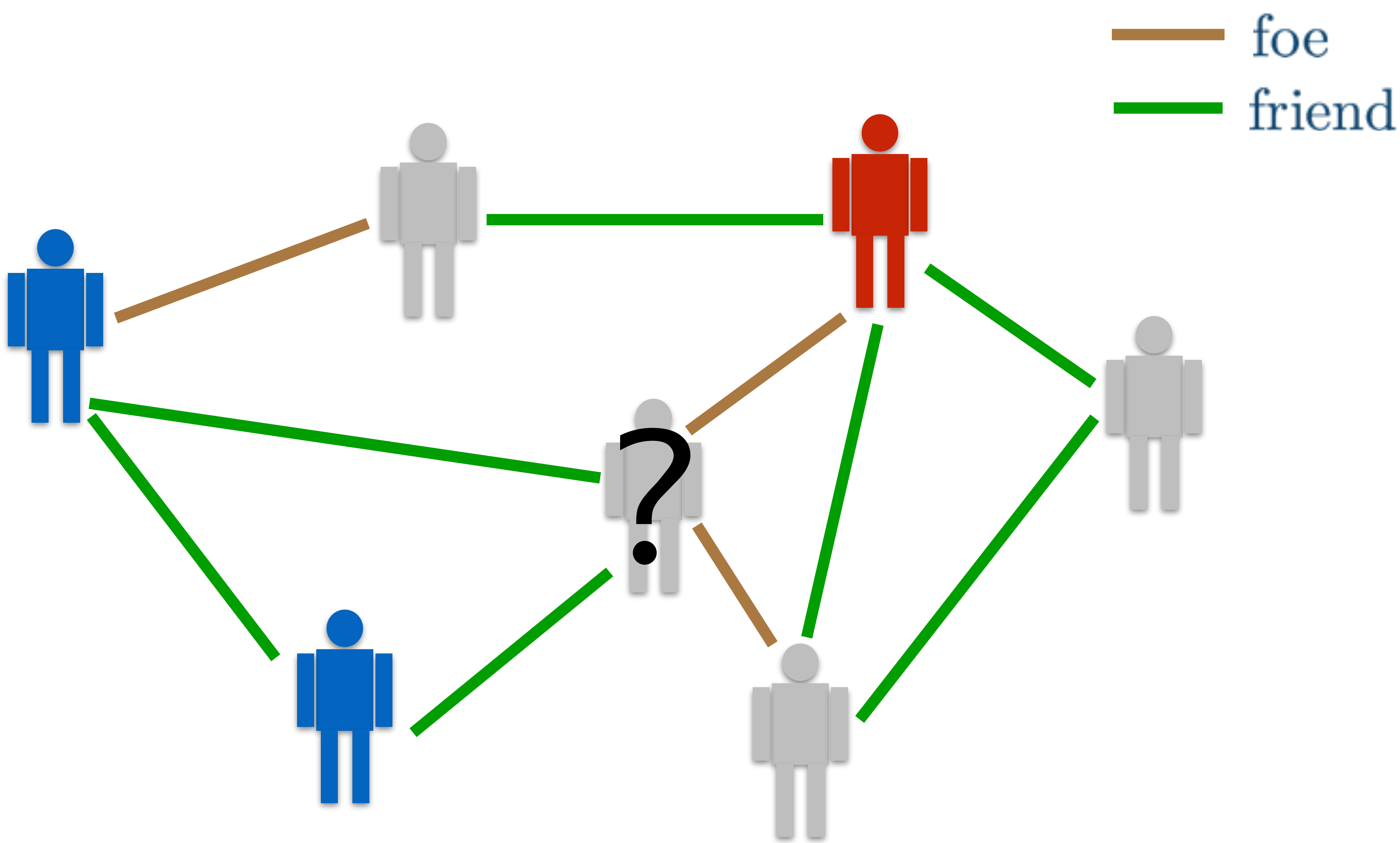
$$\mathbb{E}\left[\mathrm{Reg}_n\right] \leq \frac{B\sqrt{M+N}}{n}$$

- Statistical learning : for same rate, require assumption that user product pair is uniformly distributed [Srebro & Shraibman'05]

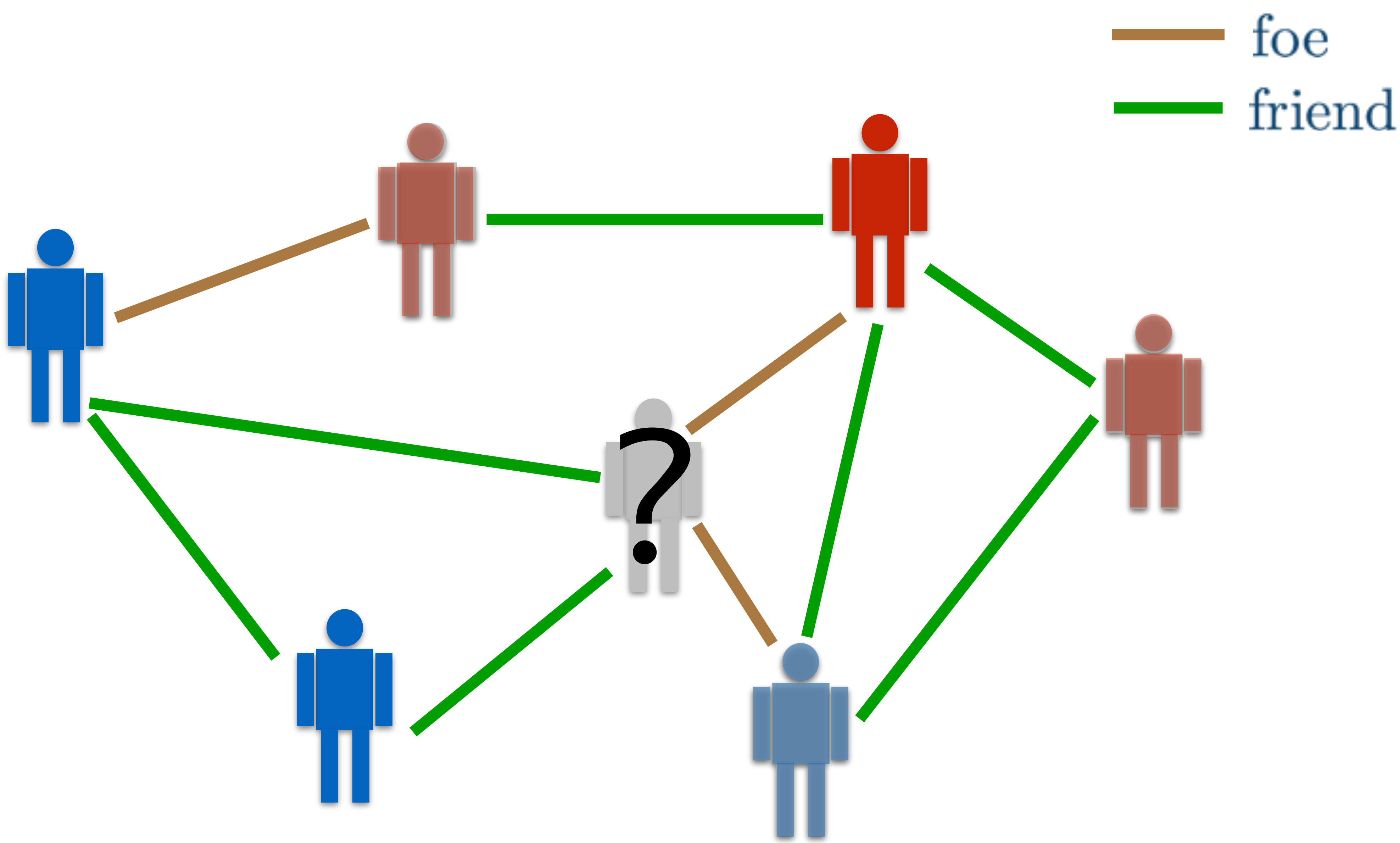
- Improves over [Cesa-Bianchi & Shamir'11], [Hazan et al'12] both in terms of regret bound and time complexity.

- Algorithm for online edge prediction and link classification in social networks (adjacency matrix)

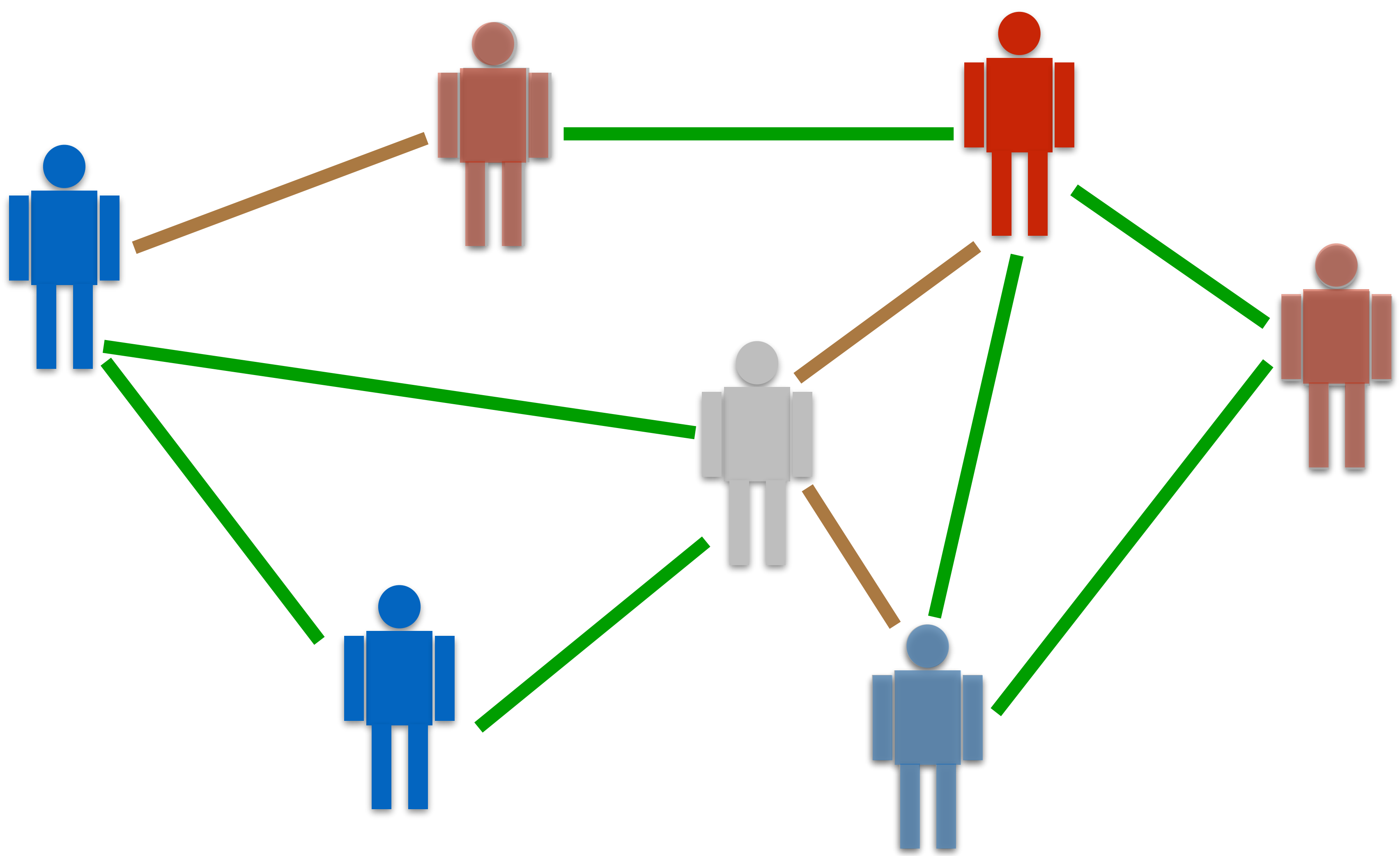
Online Node Classification



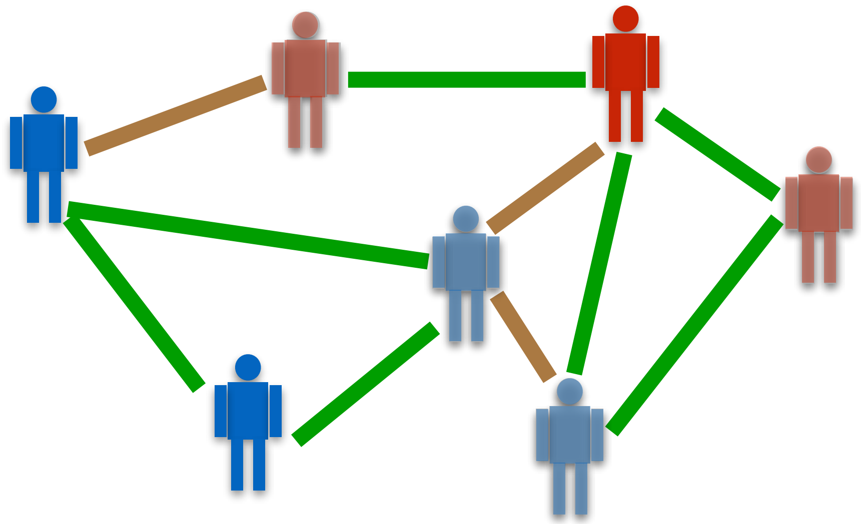
ONLINE NODE CLASSIFICATION

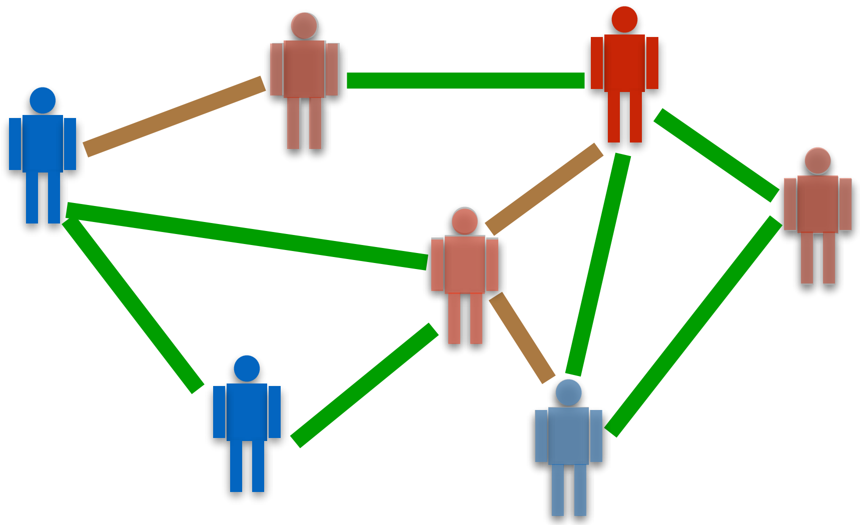


Online Node Classification

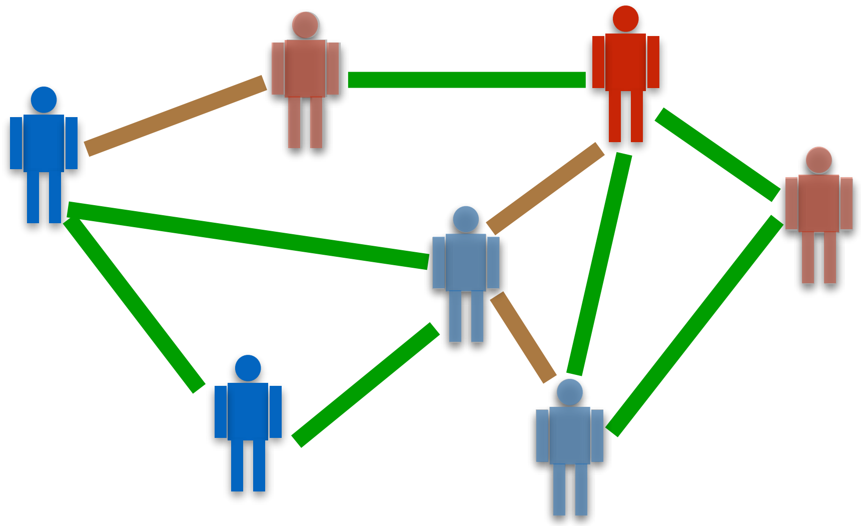


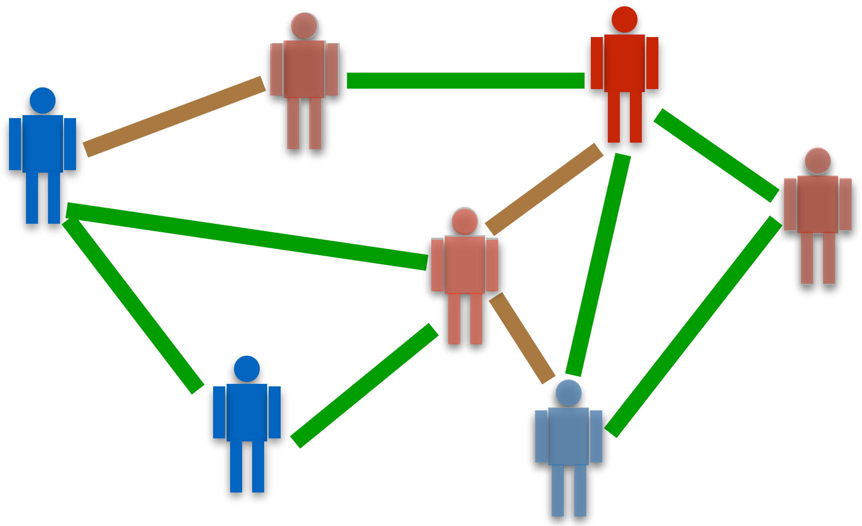
Online Node Classification





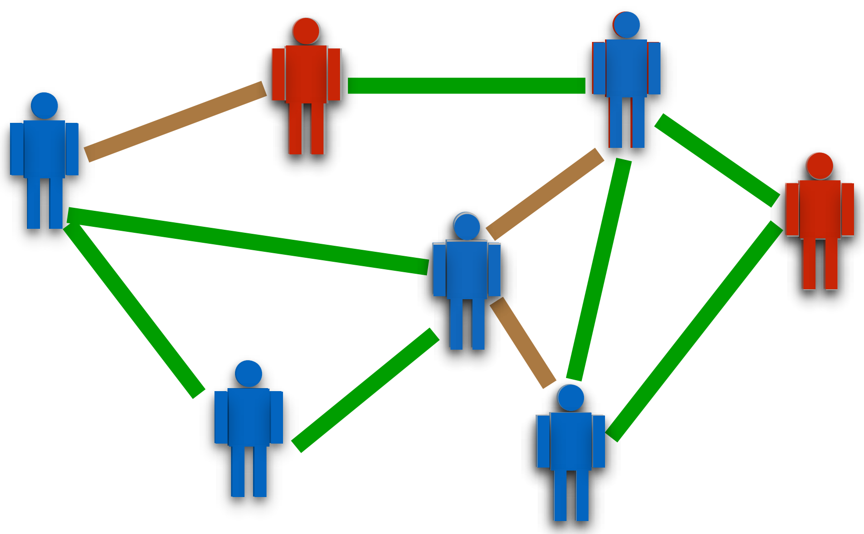
Online Node Classification



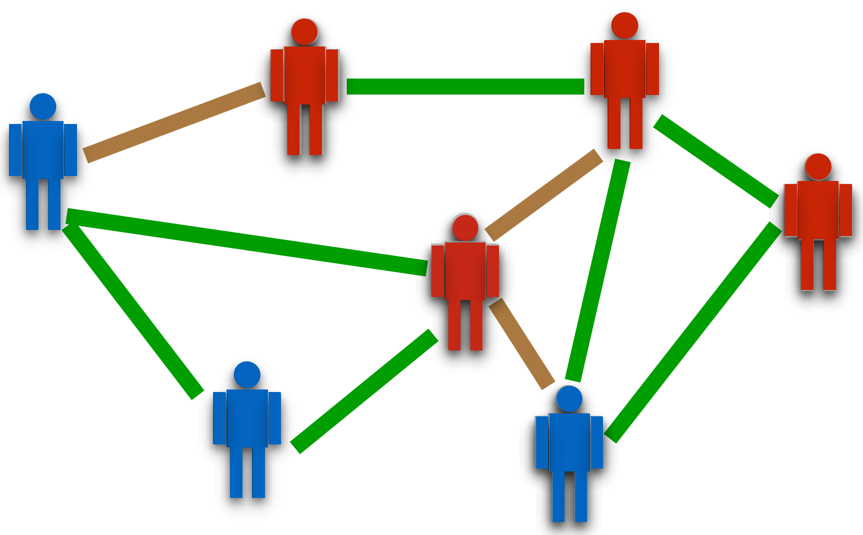


$$\mathcal{F}$$

Online Node Classification

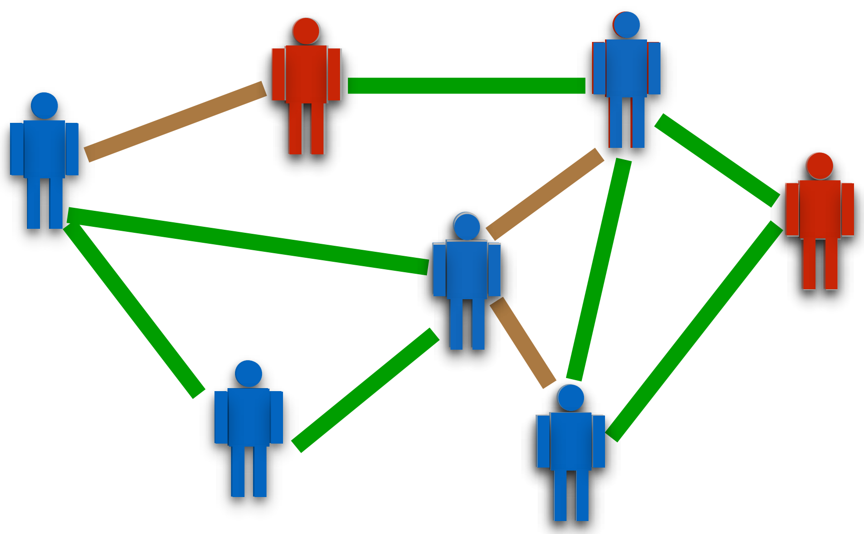


$$-$$

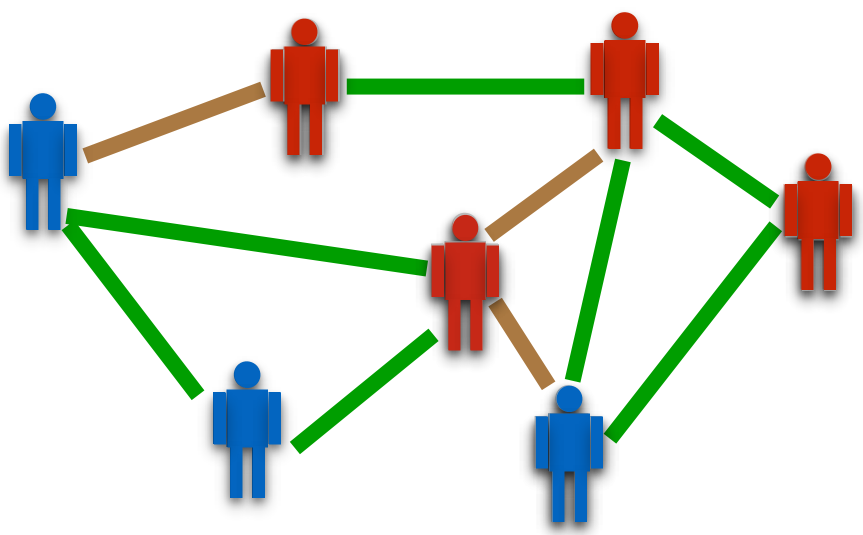


$$\mathcal{F}$$

Online Node Classification



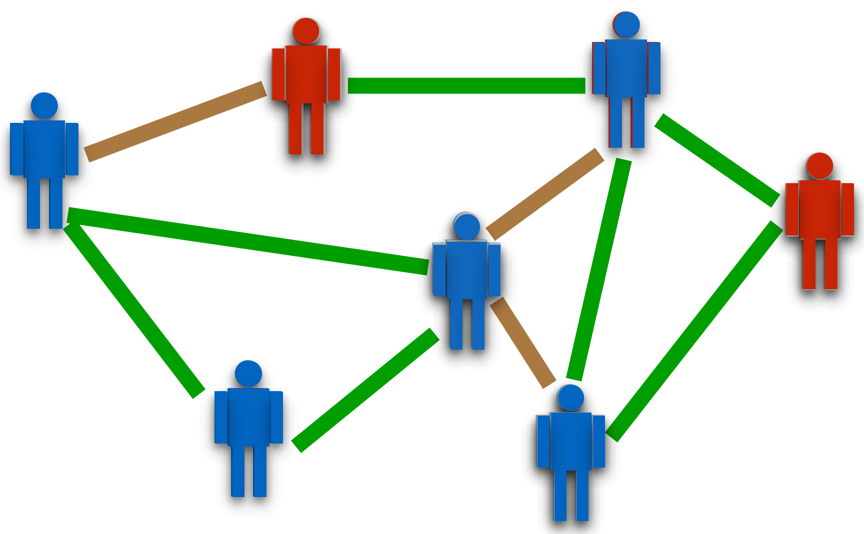
$$-$$



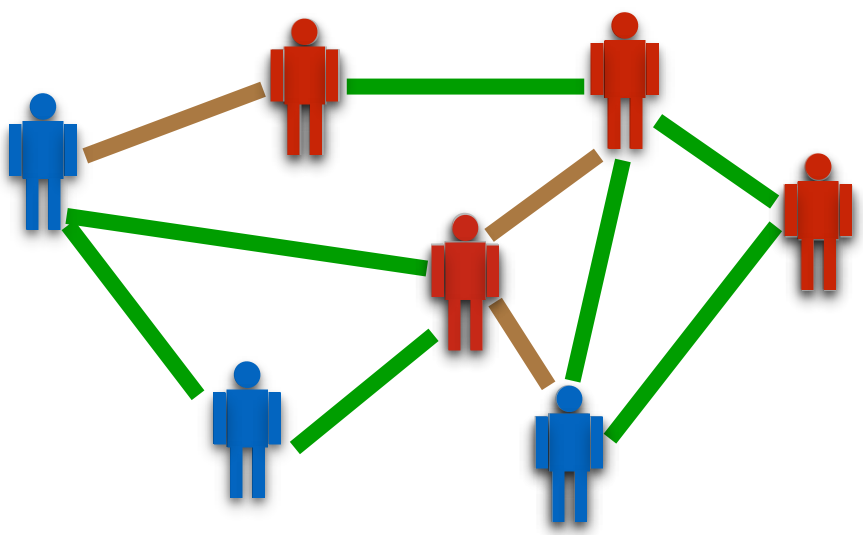
$$\mathcal{F}$$

Computationally hard!

Online Node Classification



$-$

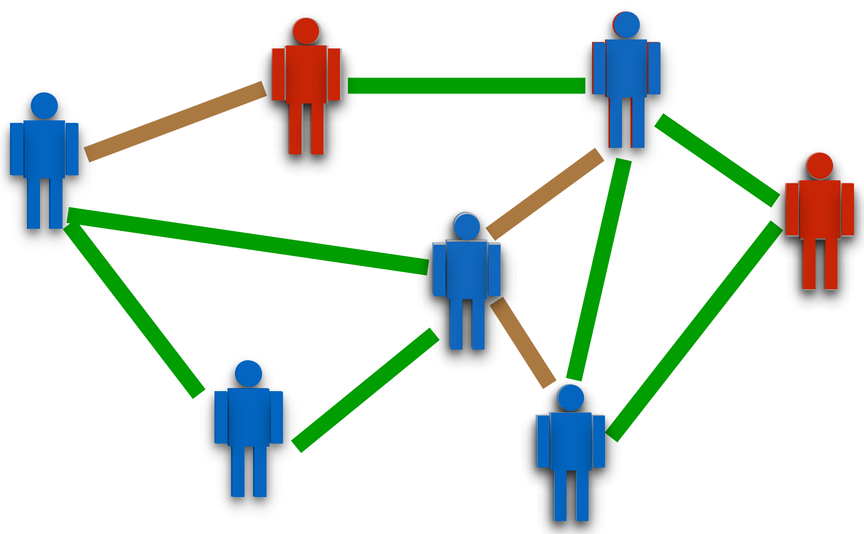


$\mathcal{F}$

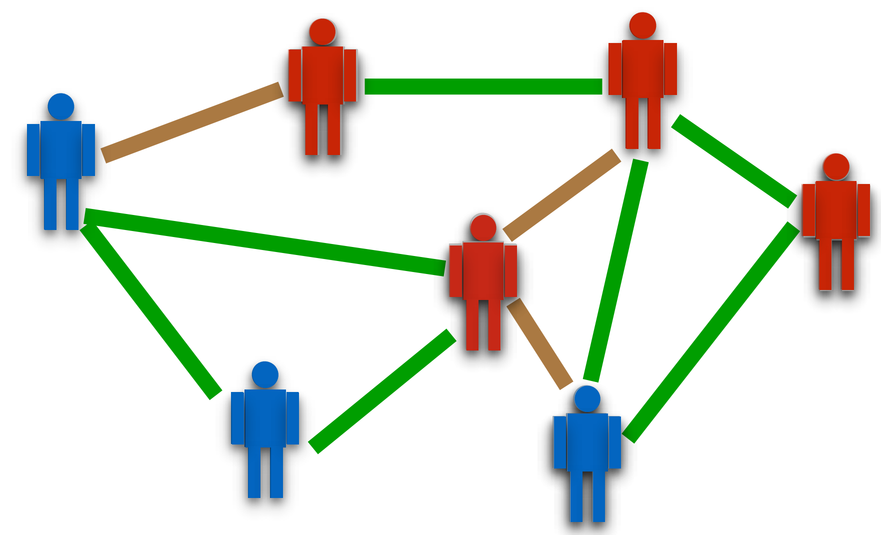
Relax

Computationally hard!

Online Node Classification



$-$



Regret bound : $\mathbb{E}\left[\mathrm{Reg}_n\right] \leq n^{-1}\sqrt{|V| \;\log|\mathcal{F}|}$