Rademacher observations, private data and boosting

Richard Nock, **Giorgio Patrini,** Arik Friedman

NICTA, ANU & UNSW

ICML 2015

Overview

NICTA

- Definition of **Rademacher observations,** rados
- Surrogate minimization with examples = surrogate minimization with rados
- An efficient **boosting algorithm** to learn from rados + Experiments
- Rados allow to **protect information** in examples from many standpoints: computational, algebraic, geometric and differential privacy

Learning setting

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\quad \boldsymbol{x}_i \in \mathbb{R}^d \quad y_i \in \{-1, 1\}$
- Sampled according to unknown but fixed distribution $\mathcal{D}$
- Objective: find algorithm $\mathcal{A}$ returning classifier $h \in \mathcal{H}$ with small true risk $\mathbb{E}_{\mathcal{D}}\left[1_{yh(\boldsymbol{x}) \leq 0}\right]$
- In practice, focus on a surrogate $\varphi(x) \geq 1_{x \leq 0}$ and minimize

$$\mathbb{E}_{\mathcal{S}}[\varphi(yh(x))]$$

- Example:

$$\varphi(x) = \log(1 + \exp(-x))$$

logistic loss

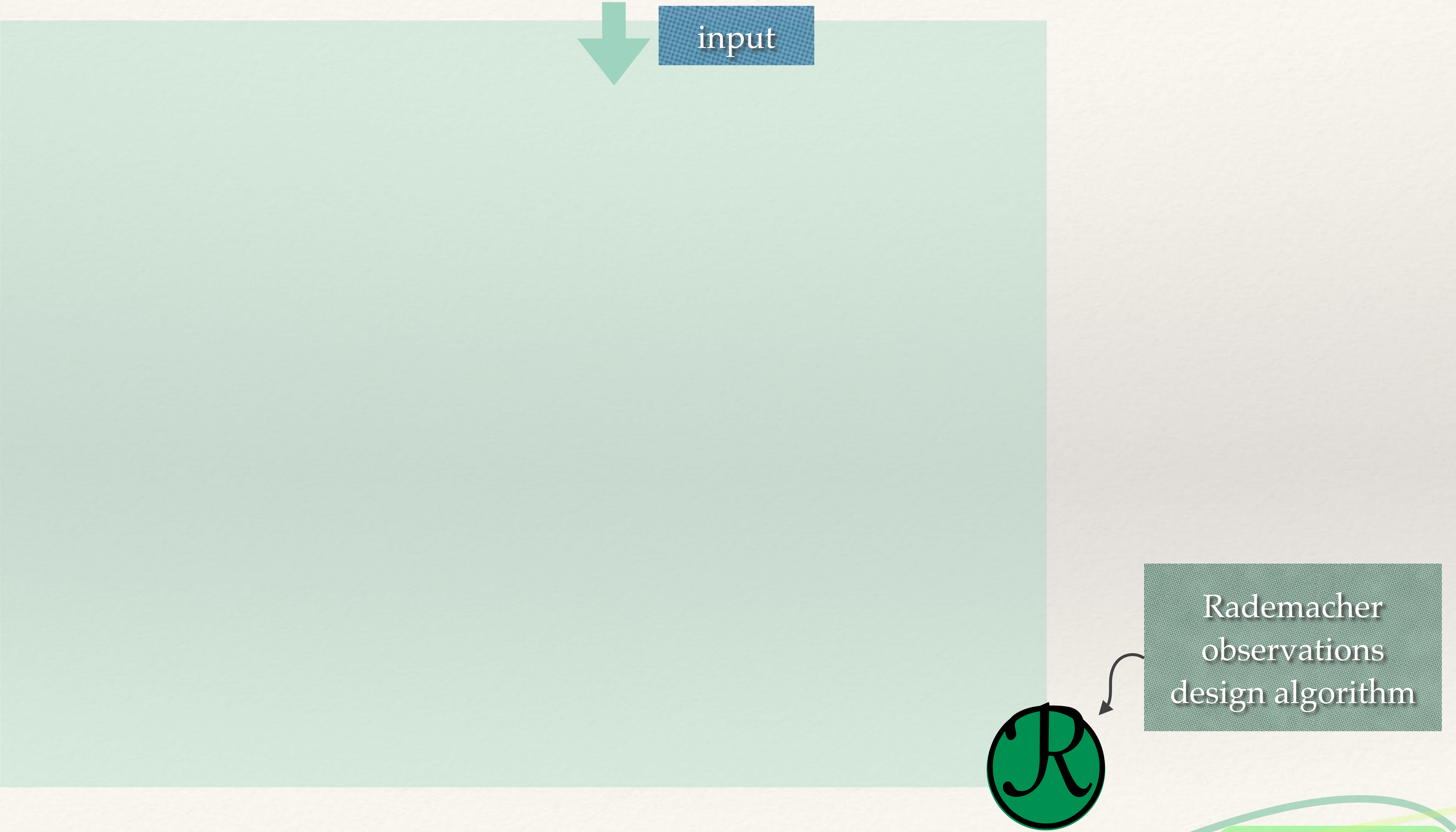
$\mathcal{H}$ = linear classifiers

$h(\boldsymbol{x}) \doteq \boldsymbol{\theta}^\top \boldsymbol{x}$

From examples to **rados**

Rademacher observations

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\quad \boldsymbol{x}_i \in \mathbb{R}^d \quad y_i \in \{-1, 1\}$



Rademacher observations

NICTA

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\boldsymbol{x}_i \in \mathbb{R}^d$ $y_i \in \{-1, 1\}$
- Compute products $y_i \cdot \boldsymbol{x}_i$

$$\begin{array}{c} y_1 \cdot \boldsymbol{x}_1 \\ y_2 \cdot \boldsymbol{x}_2 \\ \vdots \\ y_m \cdot \boldsymbol{x}_m \end{array}$$

Do all products

Rademacher observations

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\boldsymbol{x}_i \in \mathbb{R}^d$ $y_i \in \{-1, 1\}$
- Compute products $y_i \cdot \boldsymbol{x}_i$

$$\begin{array}{c} y_1 \cdot \boldsymbol{x}_1 \\ y_2 \cdot \boldsymbol{x}_2 \\ \vdots \\ y_m \cdot \boldsymbol{x}_m \end{array}$$

Do all products

Repeat…

Rademacher observations

NICTA

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\quad \boldsymbol{x}_i \in \mathbb{R}^d \quad y_i \in \{-1, 1\}$
- Compute products $y_i \cdot \boldsymbol{x}_i$

$$\begin{array}{c} y_1 \cdot \boldsymbol{x}_1 \\ y_2 \cdot \boldsymbol{x}_2 \\ \vdots \\ y_m \cdot \boldsymbol{x}_m \end{array}$$

pick $\boldsymbol{\sigma} \in \Sigma_m$

// can be (non) random,
// (non) i.i.d.,
// learned from data,
etc.

$\Sigma_m \doteq \{-1, 1\}^m$

$\mathcal{R}$

Rademacher observations

NICTA

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\quad \boldsymbol{x}_i \in \mathbb{R}^d \quad y_i \in \{-1, 1\}$
- Compute products $y_i \cdot \boldsymbol{x}_i$

$$y_1 \cdot \boldsymbol{x}_1$$
$$y_2 \cdot \boldsymbol{x}_2$$
$$\vdots$$
$$y_m \cdot \boldsymbol{x}_m$$

combine

$$1/2 \cdot \sum_i (\sigma_i + y_i) \cdot \boldsymbol{x}_i \doteq \boldsymbol{\pi}_{\boldsymbol{\sigma}}$$

1 rado

output

$$\boldsymbol{\sigma} \in \Sigma_m$$

$$\Sigma_m \doteq \{-1, 1\}^m$$

Rademacher observations

NICTA

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\quad \boldsymbol{x}_i \in \mathbb{R}^d \quad y_i \in \{-1, 1\}$
- Compute products $y_i \cdot \boldsymbol{x}_i$

$$y_1 \cdot \boldsymbol{x}_1$$
$$y_2 \cdot \boldsymbol{x}_2$$
$$\vdots$$
$$y_m \cdot \boldsymbol{x}_m$$

combine

$$1/2 \cdot \sum_i (\sigma_i + y_i) \cdot \boldsymbol{x}_i \doteq \boldsymbol{\pi}_{\boldsymbol{\sigma}}$$

1 rado

output

for each i, either y_i or 0

$$\equiv \sum_{i:\ \sigma_i = y_i} y_i \mathbf{x}_i$$

$$\boldsymbol{\sigma} \in \Sigma_m$$

$$\Sigma_m \doteq \{-1, 1\}^m$$

Rademacher observations

NICTA

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\boldsymbol{x}_i \in \mathbb{R}^d$ $y_i \in \{-1, 1\}$
- Compute products $y_i \cdot \boldsymbol{x}_i$

$y_1 \cdot \boldsymbol{x}_1$
$y_2 \cdot \boldsymbol{x}_2$
$\vdots$
$y_m \cdot \boldsymbol{x}_m$

another combination

$$1/2 \cdot \sum_i (\sigma'_i + y_i) \cdot \boldsymbol{x}_i \doteq \boldsymbol{\pi}_{\boldsymbol{\sigma}'}$$

$\boldsymbol{\pi}_{\boldsymbol{\sigma}}$

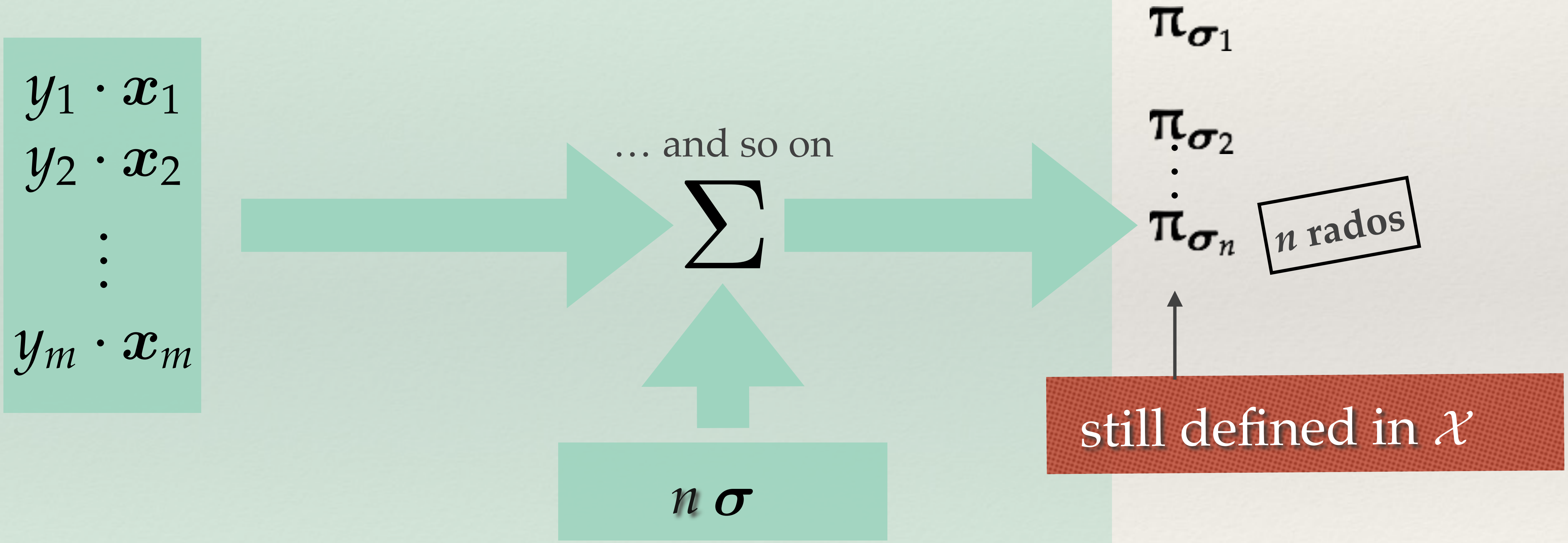
2 rados

pick $\boldsymbol{\sigma}' \in \Sigma_m$

$\Sigma_m \doteq \{-1, 1\}^m$

Rademacher observations

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\quad \boldsymbol{x}_i \in \mathbb{R}^d \quad y_i \in \{-1, 1\}$
- Compute products $y_i \cdot \boldsymbol{x}_i$



Why rados ?

Rado-loss factorization Thm

NICTA

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\boldsymbol{x}_i \in \mathbb{R}^d$ $y_i \in \{-1, 1\}$

Loss described on examples	Loss described on rados
$F_{\log}(\mathcal{S}, \boldsymbol{\theta}) \doteq \frac{1}{m} \sum_{i=1}^{m} \log\left(1 + \exp\left(-y_i \boldsymbol{\theta}^\top \boldsymbol{x}_i\right)\right)$	$F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \mathcal{U}) \doteq \frac{1}{n} \sum_{\boldsymbol{\sigma} \in \mathcal{U}} \exp\left(-\boldsymbol{\theta}^\top \boldsymbol{\pi}_{\boldsymbol{\sigma}}\right)$
	$\mathcal{U} \subseteq \Sigma_m$

Rado-loss factorization Thm

NICTA

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\quad \boldsymbol{x}_i \in \mathbb{R}^d \quad y_i \in \{-1, 1\}$

Loss described on examples

$$F_{\log}(\mathcal{S}, \boldsymbol{\theta}) \doteq \frac{1}{m} \sum_{i=1}^{m} \log\left(1 + \exp\left(-y_i \boldsymbol{\theta}^\top \boldsymbol{x}_i\right)\right)$$

Loss described on rados

$$F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \mathcal{U}) \doteq \frac{1}{n} \sum_{\boldsymbol{\sigma} \in \mathcal{U}} \exp\left(-\boldsymbol{\theta}^\top \boldsymbol{\pi}_{\boldsymbol{\sigma}}\right)$$

$\mathcal{U} \subseteq \Sigma_m$

Thm

$$F_{\log}(\mathcal{S}, \boldsymbol{\theta}) = \log(2) + \frac{1}{m} \log F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \Sigma_m)$$

- Hence,

$$\arg\min_{\boldsymbol{\theta}} F_{\log}(\mathcal{S}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \Sigma_m)$$

Same classifier…

Bottleneck

NICTA

- Learning sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i = 1, 2, ..., m\}$ $\boldsymbol{x}_i \in \mathbb{R}^d$ $y_i \in \{-1, 1\}$

Loss described on examples

$$F_{\log}(\mathcal{S}, \boldsymbol{\theta}) \doteq \frac{1}{m} \sum_{i=1}^{m} \log\left(1 + \exp\left(-y_i \boldsymbol{\theta}^\top \boldsymbol{x}_i\right)\right)$$

Loss described on rados

$$F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \mathcal{U}) \doteq \frac{1}{n} \sum_{\boldsymbol{\sigma} \in \mathcal{U}} \exp\left(-\boldsymbol{\theta}^\top \boldsymbol{\pi}_{\boldsymbol{\sigma}}\right)$$

$\mathcal{U} \subseteq \Sigma_m$

Thm $$F_{\log}(\mathcal{S}, \boldsymbol{\theta}) = \log(2) + \frac{1}{m} \log F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \Sigma_m)$$

$\mathcal{U} = \Sigma_m$

- Hence,

$$\arg\min_{\boldsymbol{\theta}} F_{\log}(\mathcal{S}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \Sigma_m)$$

Same classifier… but $|\mathcal{U}| = 2^m$…

Workaround

Thm

- Let $\mathcal{U} \sim_{i.u.d.} \Sigma_m$ with $|\mathcal{U}| = n$. Then with probability $\geq 1 - \eta$ over the sampling of $\mathcal{U}$,

$$F_{\log}(\mathcal{S}, \boldsymbol{\theta}) \leq \log(2) + \frac{1}{m} \log F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \mathcal{U}) + O\left(\frac{\varrho}{m^\beta} \cdot \sqrt{\frac{r_\theta \pi^*_r}{n} + \frac{d}{nm} \log \frac{2en}{d\eta}}\right)$$

$(\forall \beta < 1/2)$

- Holds for **any** learning sample $\mathcal{S}$,
- Provided a sufficient number of rados, the minimization of $F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \mathcal{U})$ is a **good proxy** for the minimization of $F_{\log}(\mathcal{S}, \boldsymbol{\theta})$

.

Improved workaround

$\forall \Sigma_r \subseteq \Sigma_m$

Thm

- Let $\mathcal{U} \sim_{i.u.d.} \Sigma_r$ with $|\mathcal{U}| = n$. Then with probability $\geq 1 - \eta$ over the sampling of $\mathcal{U}$,

$$F_{\log}(\mathcal{S}, \boldsymbol{\theta}) \leq \log(2) + \frac{1}{m} \log F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \mathcal{U}) + O\left(\frac{\varrho}{m^\beta} \cdot \sqrt{\frac{r_\theta \pi^*_r}{n} + \frac{d}{nm} \log \frac{2en}{d\eta}}\right) + Q$$

$(\forall \beta < 1/2)$

NICTA

Improved workaround

$\forall \Sigma_r \subseteq \Sigma_m$

Thm
- Let $\mathcal{U} \sim_{i.u.d.} \Sigma_r$ with $|\mathcal{U}| = n$. Then with probability $\geq 1 - \eta$ over the sampling of $\mathcal{U}$,

$$F_{\log}(\mathcal{S}, \boldsymbol{\theta}) \leq \log(2) + \frac{1}{m} \log F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \mathcal{U}) + O\left(\frac{\varrho}{m^\beta} \cdot \sqrt{\frac{r_\theta \pi^*_r}{n} + \frac{d}{nm} \log \frac{2en}{d\eta}}\right) + Q$$

$(\forall \beta < 1/2)$

Authorizes sophisticated design mechanisms for Σ_r, to solve particular problems.

Any efficient learning algorithm with rados ?

$\min_{\boldsymbol{\theta}} F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \mathcal{U})$

Radoboost

NICTA

Algorithm 1 Rado boosting (RadoBoost)

Input set of rados $\mathcal{S}^r \doteq \{\pi_1, \pi_2, ..., \pi_n\}$; $T \in \mathbb{N}_*$;
Step 1 : let $\boldsymbol{\theta}_0 \leftarrow \mathbf{0}$, $\boldsymbol{w}_0 \leftarrow (1/n)\mathbf{1}$;
Step 2 : **for** $t = 1, 2, ..., T$
Step 2.1 : $[d] \ni \iota(t) \leftarrow \text{WFI}(\mathcal{S}^r, \boldsymbol{w}_t)$;
Step 2.2 : let

$$r_t \leftarrow \frac{1}{\pi_{*\iota(t)}} \sum_{j=1}^{n} w_{tj} \pi_{j\iota(t)} \; ; \quad (1)$$

$$\alpha_t \leftarrow \frac{1}{2\pi_{*\iota(t)}} \log \frac{1 + r_t}{1 - r_t} \; ; \quad (2)$$

Step 2.3 : **for** $j = 1, 2, ..., n$

$$w_{(t+1)j} \leftarrow w_{tj} \cdot \left(\frac{1 - \frac{r_t \pi_{j\iota(t)}}{\pi_{*\iota(t)}}}{1 - r_t^2} \right) \; ; \quad (3)$$

Return $\boldsymbol{\theta}_T$ defined by $\theta_{Tk} \doteq \sum_{t:\iota(t)=k} \alpha_t$, $\forall k \in [d]$;

Radoboost

Algorithm 1 Rado boosting (RadoBoost)

Input set of rados $\mathcal{S}^r \doteq \{\pi_1, \pi_2, ..., \pi_n\}$; $T \in \mathbb{N}_*$;

Step 1 : let $\theta_0 \leftarrow \mathbf{0}$, $\boldsymbol{w}_0 \leftarrow (1/n)\mathbf{1}$;

Step 2 : **for** $t = 1, 2, ..., T$

Step 2.1 : $[d] \ni \iota(t) \leftarrow \textsc{wfi}(\mathcal{S}^r, \boldsymbol{w}_t)$;

Step 2.2 : let

$$r_t \leftarrow \frac{1}{\pi_{*\iota(t)}} \sum_{j=1}^{n} w_{tj} \pi_{j\iota(t)} \ ; \quad (1)$$

$$\alpha_t \leftarrow \frac{1}{2\pi_{*\iota(t)}} \log \frac{1 + r_t}{1 - r_t} \ ; \quad (2)$$

Step 2.3 : **for** $j = 1, 2, ..., n$

$$w_{(t+1)j} \leftarrow w_{tj} \cdot \left(\frac{1 - \frac{r_t \pi_{j\iota(t)}}{\pi_{*\iota(t)}}}{1 - r_t^2} \right) \ ; \quad (3)$$

Return θ_T defined by $\theta_{Tk} \doteq \sum_{t:\iota(t)=k} \alpha_t$, $\forall k \in [d]$;

Input: set of rados

NICTA

Radoboost

Algorithm 1 Rado boosting (RadoBoost)

Input set of rados $\mathcal{S}^r \doteq \{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, ..., \boldsymbol{\pi}_n\}$; $T \in \mathbb{N}_*$;
Step 1 : let $\boldsymbol{\theta}_0 \leftarrow \mathbf{0}$, $\boldsymbol{w}_0 \leftarrow (1/n)\mathbf{1}$;
Step 2 : **for** $t = 1, 2, ..., T$
Step 2.1 : $[d] \ni \iota(t) \leftarrow \text{WFI}(\mathcal{S}^r, \boldsymbol{w}_t)$;
Step 2.2 : let

$$r_t \leftarrow \frac{1}{\pi_{*\iota(t)}} \sum_{j=1}^{n} w_{tj} \pi_{j\iota(t)} \; ; \quad (1)$$

$$\alpha_t \leftarrow \frac{1}{2\pi_{*\iota(t)}} \log \frac{1 + r_t}{1 - r_t} \; ; \quad (2)$$

Step 2.3 : **for** $j = 1, 2, ..., n$

$$w_{(t+1)j} \leftarrow w_{tj} \cdot \left(\frac{1 - \frac{r_t \pi_{j\iota(t)}}{\pi_{*\iota(t)}}}{1 - r_t^2} \right) \; ; \quad (3)$$

Return $\boldsymbol{\theta}_T$ defined by $\theta_{Tk} \doteq \sum_{t:\iota(t)=k} \alpha_t$, $\forall k \in [d]$;

Input: set of rados

Weak choice of a feature

Radoboost

Algorithm 1 Rado boosting (RadoBoost)

Input set of rados $\mathcal{S}^r \doteq \{\pi_1, \pi_2, ..., \pi_n\}$; $T \in \mathbb{N}_*$;

Step 1 : let $\boldsymbol{\theta}_0 \leftarrow \mathbf{0}$, $\boldsymbol{w}_0 \leftarrow (1/n)\mathbf{1}$;

Step 2 : **for** $t = 1, 2, ..., T$

Step 2.1 : $[d] \ni \iota(t) \leftarrow \textsc{wfi}(\mathcal{S}^r, \boldsymbol{w}_t)$;

Step 2.2 : let

$$r_t \leftarrow \frac{1}{\pi_{*\iota(t)}} \sum_{j=1}^{n} w_{tj} \pi_{j\iota(t)} \ ; \quad (1)$$

$$\alpha_t \leftarrow \frac{1}{2\pi_{*\iota(t)}} \log \frac{1 + r_t}{1 - r_t} \ ; \quad (2)$$

Step 2.3 : **for** $j = 1, 2, ..., n$

$$w_{(t+1)j} \leftarrow w_{tj} \cdot \left(\frac{1 - \frac{r_t \pi_{j\iota(t)}}{\pi_{*\iota(t)}}}{1 - r_t^2} \right) \ ; \quad (3)$$

Return $\boldsymbol{\theta}_T$ defined by $\theta_{Tk} \doteq \sum_{t:\iota(t)=k} \alpha_t$, $\forall k \in [d]$;

Input: set of rados

Weak choice of a feature

normalized rado edge

Radoboost

Algorithm 1 Rado boosting (RadoBoost)

Input set of rados $\mathcal{S}^r \doteq \{\pi_1, \pi_2, ..., \pi_n\}$; $T \in \mathbb{N}_*$;

Step 1 : let $\boldsymbol{\theta}_0 \leftarrow \mathbf{0}$, $\boldsymbol{w}_0 \leftarrow (1/n)\mathbf{1}$;

Step 2 : **for** $t = 1, 2, ..., T$

Step 2.1 : $[d] \ni \iota(t) \leftarrow \text{WFI}(\mathcal{S}^r, \boldsymbol{w}_t)$;

Step 2.2 : let

$$r_t \leftarrow \frac{1}{\pi_{*\iota(t)}} \sum_{j=1}^{n} w_{tj} \pi_{j\iota(t)} \ ; \quad (1)$$

$$\alpha_t \leftarrow \frac{1}{2\pi_{*\iota(t)}} \log \frac{1 + r_t}{1 - r_t} \ ; \quad (2)$$

Step 2.3 : **for** $j = 1, 2, ..., n$

$$w_{(t+1)j} \leftarrow w_{tj} \cdot \left(\frac{1 - \frac{r_t \pi_{j\iota(t)}}{\pi_{*\iota(t)}}}{1 - r_t^2} \right) \ ; \quad (3)$$

Return $\boldsymbol{\theta}_T$ defined by $\theta_{Tk} \doteq \sum_{t:\iota(t)=k} \alpha_t$, $\forall k \in [d]$;

Input: set of rados

Weak choice of a feature

normalized rado edge

leveraging coefficient

Radoboost

NICTA

Algorithm 1 Rado boosting (RadoBoost)

Input set of rados $\mathcal{S}^r \doteq \{\pi_1, \pi_2, ..., \pi_n\}$; $T \in \mathbb{N}_*$;

Step 1 : let $\boldsymbol{\theta}_0 \leftarrow \mathbf{0}$, $\boldsymbol{w}_0 \leftarrow (1/n)\mathbf{1}$;

Step 2 : **for** $t = 1, 2, ..., T$

Step 2.1 : $[d] \ni \iota(t) \leftarrow \text{WFI}(\mathcal{S}^r, \boldsymbol{w}_t)$;

Step 2.2 : let

$$r_t \leftarrow \frac{1}{\pi_{*\iota(t)}} \sum_{j=1}^{n} w_{tj} \pi_{j\iota(t)} \ ;$$

$$\alpha_t \leftarrow \frac{1}{2\pi_{*\iota(t)}} \log \frac{1 + r_t}{1 - r_t} \ ;$$

Step 2.3 : **for** $j = 1, 2, ..., n$

$$w_{(t+1)j} \leftarrow w_{tj} \cdot \left(\frac{1 - \frac{r_t \pi_{j\iota(t)}}{\pi_{*\iota(t)}}}{1 - r_t^2} \right) \ ;$$

Return $\boldsymbol{\theta}_T$ defined by $\theta_{Tk} \doteq \sum_{t:\iota(t)=k} \alpha_t$, $\forall k \in [d]$;

Input: set of rados

Weak choice of a feature

normalized rado edge

leveraging coefficient

No renormalization step

Radoboost

Algorithm 1 Rado boosting (RadoBoost)

Input set of rados $\mathcal{S}^r \doteq \{\pi_1, \pi_2, ..., \pi_n\}$; $T \in \mathbb{N}_*$;
Step 1 : let $\theta_0 \leftarrow \mathbf{0}$, $\boldsymbol{w}_0 \leftarrow (1/n)\mathbf{1}$;
Step 2 : **for** $t = 1, 2, ..., T$
Step 2.1 : $[d] \ni \iota(t) \leftarrow \textsc{wfi}(\mathcal{S}^r, \boldsymbol{w}_t)$;
Step 2.2 : let

$$r_t \leftarrow \frac{1}{\pi_{*\iota(t)}} \sum_{j=1}^{n} w_{tj} \pi_{j\iota(t)} \ ;$$

$$\alpha_t \leftarrow \frac{1}{2\pi_{*\iota(t)}} \log \frac{1 + r_t}{1 - r_t} \ ;$$

Step 2.3 : **for** $j = 1, 2, ..., n$

$$w_{(t+1)j} \leftarrow w_{tj} \cdot \left(\frac{1 - \frac{r_t \pi_{j\iota(t)}}{\pi_{*\iota(t)}}}{1 - r_t^2} \right) \ ;$$

Return θ_T defined by $\theta_{Tk} \doteq \sum_{t:\iota(t)=k} \alpha_t$, $\forall k \in [d]$;

Input: set of rados

Weak choice of a feature

normalized rado edge

leveraging coefficient

No renormalization step

Final classifier can be used directly on new observations

Radoboost… boosts !

NICTA

- Weak learning assumption (WLA): $\exists \gamma > 0$ such that $|r_t| \geq \gamma, \forall t$
- Then after T rounds of boosting, the output $\boldsymbol{\theta}_T$ of RadoBoost meets:

Thm

$$F^{r}_{\exp}(\mathcal{S}, \boldsymbol{\theta}_T, \mathcal{U}) \leq \exp\left(-T\gamma^2/2\right)$$

Radoboost… boosts !

NICTA

- Weak learning assumption (WLA): $\exists \gamma > 0$ such that $|r_t| \geq \gamma, \forall t$
- Then after T rounds of boosting, the output $\boldsymbol{\theta}_T$ of RadoBoost meets:

Thm

$$F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}_T, \mathcal{U}) \quad \leq \quad \exp\left(-T\gamma^2/2\right)$$

- So, since $F_{\log}(\mathcal{S}, \boldsymbol{\theta}_T) \quad = \quad \log(2) + \frac{1}{m} \log F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}_T, \Sigma_m),$
- we have

$$F_{\log}\left(\mathcal{S}, \boldsymbol{\theta}_T\right) \leq \log(2) - \frac{T\gamma^2}{2m}$$

If $\mathcal{U} = \Sigma_m \ldots$

Radoboost… boosts !

NICTA

- Weak learning assumption (WLA): $\exists\gamma > 0$ such that $|r_t| \geq \gamma, \forall t$
- Then after T rounds of boosting, the output $\boldsymbol{\theta}_T$ of RadoBoost meets:

Thm

$$F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}_T, \mathcal{U}) \quad \leq \quad \exp\left(-T\gamma^2/2\right)$$

- So, since $F_{\log}(\mathcal{S}, \boldsymbol{\theta}_T) \quad = \quad \log(2) + \dfrac{1}{m} \log F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}_T, \Sigma_m)$,
- we have

$$F_{\log}(\mathcal{S}, \boldsymbol{\theta}_T) \leq \log(2) - \frac{T\gamma^2}{2m}$$

If $\mathcal{U} = \Sigma_m$ …

- … in the general case ($\forall\mathcal{U}$),

Thm

$$F_{\log}(\mathcal{S}, \boldsymbol{\theta}_T) \quad \leq \quad \log(2) - \frac{T\gamma^2}{2m} + Q'$$

not a function of T

Experiments

Experiments (some)

NICTA

- RadoBoost vs AdaBoost $(T = 1000)$

number of rados / examples

$n = \min\{1000, \text{train fold size}/2\}$

Domain	m	d	AdaBoost err±σ	AdaBoost(n) err±σ	AdaBoost(n) $\frac{n}{m}$	RadoBoost err±σ	RadoBoost $\frac{n}{2^m}$
Abalone	4 177	8	22.96±1.44	23.20±1.44	0.24	25.14±1.83	[3:−[1:3]]
Wine-white	4 898	11	30.93±3.42	30.44±3.25	0.20	32.48±3.55	[3:−[1:3]]
Magic	19 020	10	21.07±0.98	20.91±0.99	0.05	22.75±1.51	[3:−[5:3]]
EEG	14 980	14	46.04±1.38	44.36±1.99	0.07	44.23±1.73	[4:−[4:3]]
Hardware	28 179	95	16.82±0.72	16.76±0.73	0.04	7.61±3.24	[2:−[8:3]]
Twitter	583 250	77	53.75±1.48	53.09±11.23	[1:−3]	6.00±0.77	[1:−[1:5]]
SuSy	5 000 000	17	27.76±0.14	27.43±0.19	[2:−4]	27.26±0.55	[1:−[1:6]]
Higgs	11 000 000	28	42.55±0.19	45.39±0.28	[9:−5]	47.86±0.06	[1:−[1:7]]

$9 \cdot 10^{-5}$ vs $10^{-10000000}$

Improved workaround

$\forall \Sigma_r \subseteq \Sigma_m$

NICTA

Thm

- Let $\mathcal{U} \sim_{i.u.d.} \Sigma_r$ with $|\mathcal{U}| = n$. Then with probability $\geq 1 - \eta$ over the sampling of $\mathcal{U}$,

$$F_{\log}(\mathcal{S}, \boldsymbol{\theta}) \leq \log(2) + \frac{1}{m} \log F^r_{\exp}(\mathcal{S}, \boldsymbol{\theta}, \mathcal{U}) + O\left(\frac{\varrho}{m^\beta} \cdot \sqrt{\frac{r_\theta \pi^*_r}{n} + \frac{d}{nm} \log \frac{2en}{d\eta}}\right) + Q$$

$(\forall \beta < 1/2)$

$\boldsymbol{\theta} \in \mathcal{B}(0, r_\theta)$

Authorizes sophisticated design mechanisms for Σ_r, to solve particular problems.

Example: privacy

Rados and privacy

NICTA

- Protection guarantees:
  - Crafting of **differentially private (DP)** rados from examples
  - **Computational hardness** of approximate sparse recovery of examples from rados
  - **Computational hardness** of pinpointing examples used to craft rados
  - **Geometric** and **algebraic hardness** of recovering examples from rados
  - Learning with rados from **differentially private** (noisified) examples.

Rados and privacy

NICTA

- Protection guarantees:
  - Crafting of **differentially private (DP)** rados from examples
  - **Computational hardness** of approximate sparse recovery of examples from rados
    See paper
  - **Computational hardness** of pinpointing examples used to craft rados
  - **Geometric** and algebraic **hardness** of recovering examples from rados
    See paper
  - Learning with rados from **differentially private** (noisified) examples.
    See paper

DP-rados from non-DP examples

- **Definition**: statistical protection of one sensitive feature f so that changing one *example* (in $\mathcal{S}$) does not change **significantly** the (statistical) distribution of that feature in *rados* (wrt Σ_r):

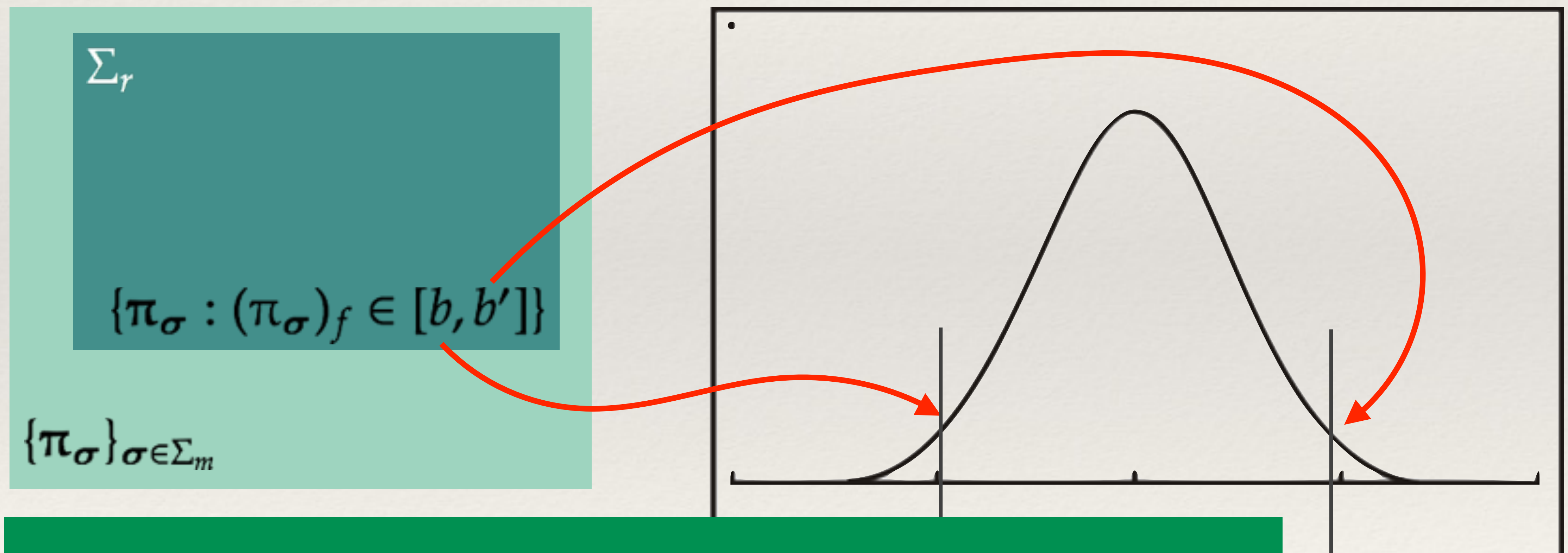
$$\mu(f \text{ in rados}|\mathcal{S}) \leq \mu(f \text{ in rados}|\mathcal{S}') \cdot \exp(\epsilon) + \delta$$

DP-rados from non-DP examples

NICTA

- **Definition**: statistical protection of one sensitive feature *f* so that changing one *example* (in $\mathcal{S}$) does not change **significantly** the (statistical) distribution of that feature in *rados* (wrt Σ_r):

$$\mu(f \text{ in rados}|\mathcal{S}) \leq \mu(f \text{ in rados}|\mathcal{S}') \cdot \exp(\epsilon) + \delta$$



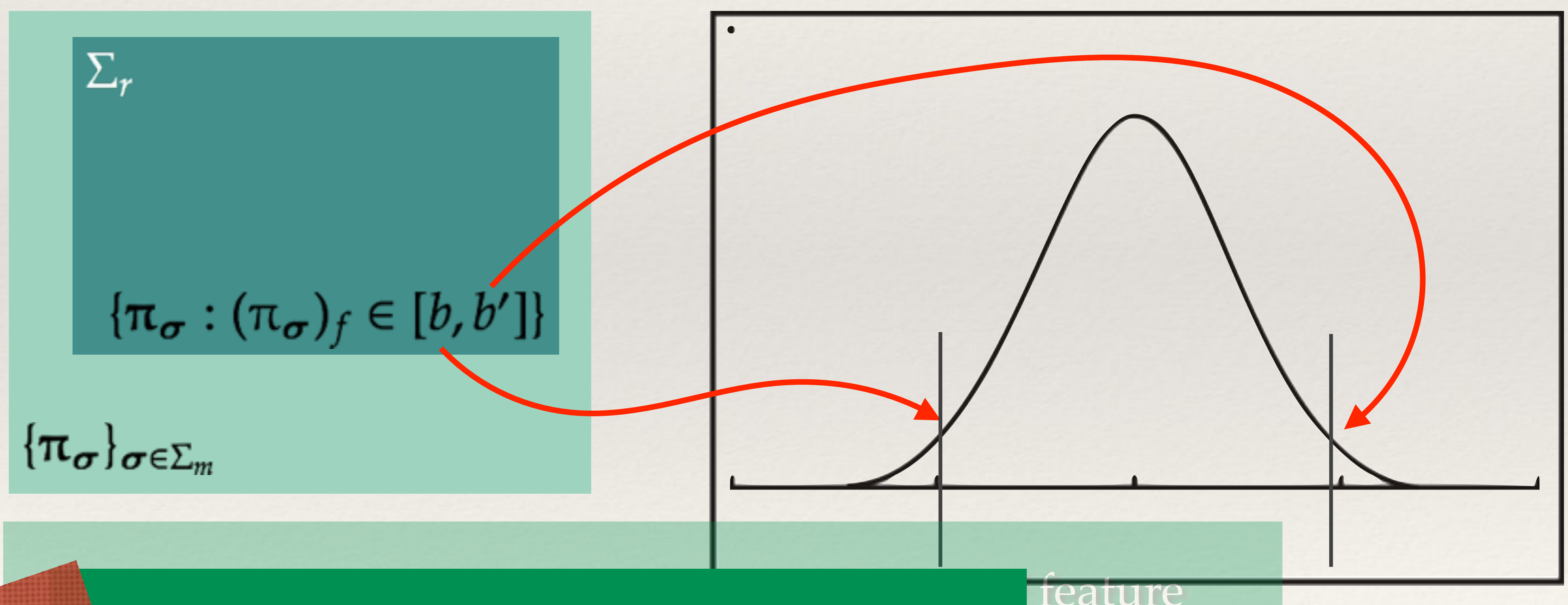
principle: prune the tails of the distribution of feature *f*

DP-rados from non-DP examples

NICTA

- **Definition**: statistical protection of one sensitive feature f so that changing one *example* (in $\mathcal{S}$) does not change **significantly** the (statistical) distribution of that feature in *rados* (wrt Σ_r):

$$\mu(f \text{ in rados}|\mathcal{S}) \leq \mu(f \text{ in rados}|\mathcal{S}') \cdot \exp(\epsilon) + \delta$$

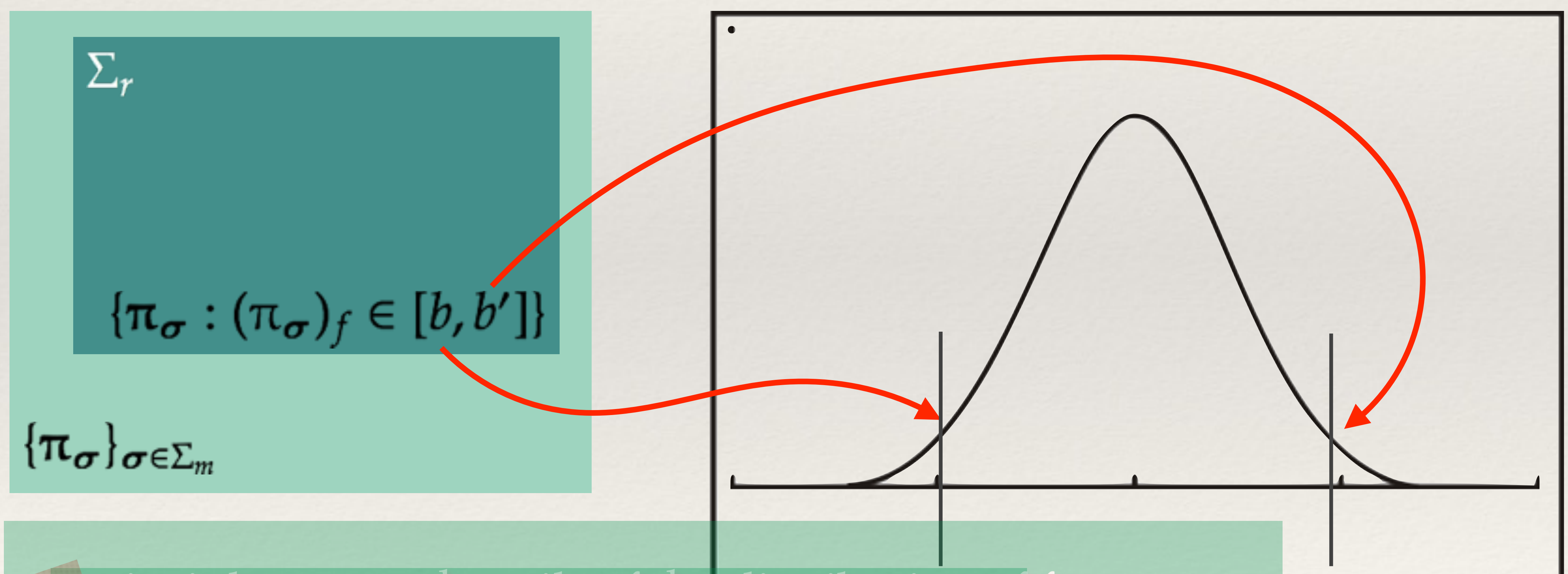


Thm $\Rightarrow$ (ϵ, δ)-differential privacy on feature f

DP-rados from non-DP examples

- **Definition**: statistical protection of one sensitive feature f so that changing one *example* (in $\mathcal{S}$) does not change **significantly** the (statistical) distribution of that feature in *rados* (wrt Σ_r):

$$\mu(f \text{ in rados}|\mathcal{S}) \leq \mu(f \text{ in rados}|\mathcal{S}') \cdot \exp(\epsilon) + \delta$$



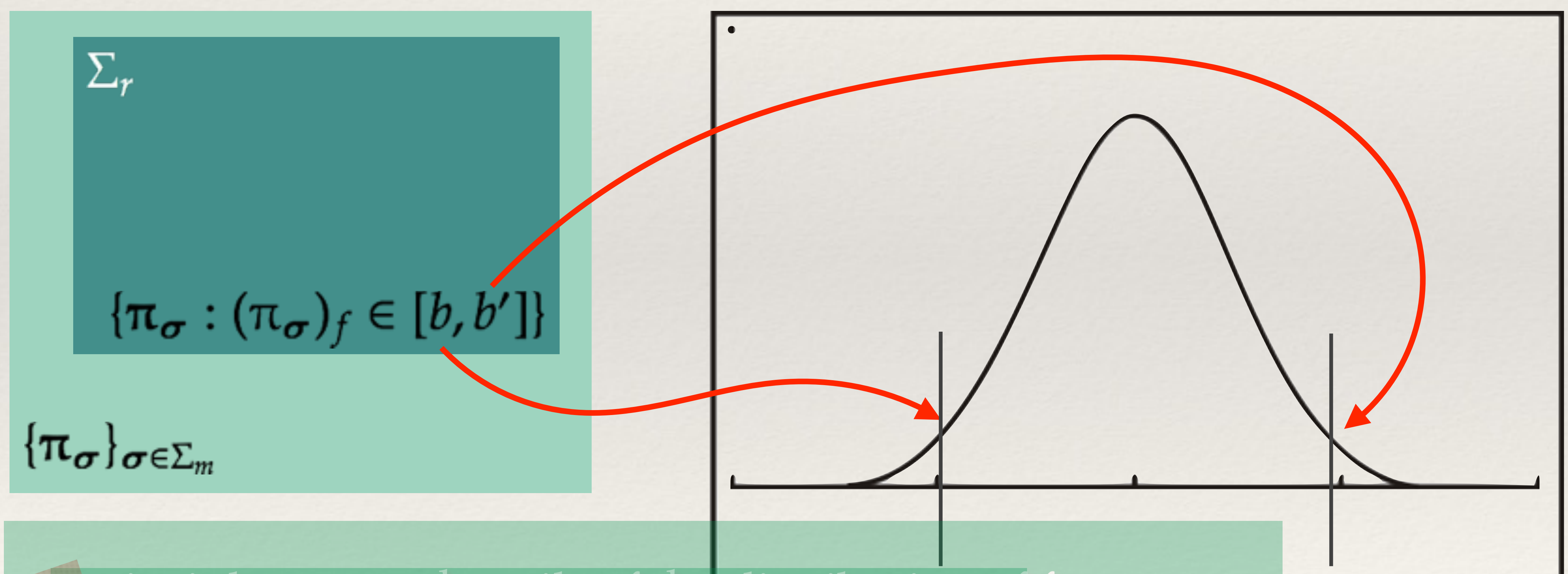
Thm There exists an efficient rejection sampling to implement the mechanism

DP-rados from non-DP examples

NICTA

- **Definition**: statistical protection of one sensitive feature *f* so that changing one *example* (in $\mathcal{S}$) does not change **significantly** the (statistical) distribution of that feature in *rados* (wrt Σ_r):

$$\mu(f \text{ in rados}|\mathcal{S}) \leq \mu(f \text{ in rados}|\mathcal{S}') \cdot \exp(\epsilon) + \delta$$



principle: prune the tails of the distribution of feature

This mechanism does not require noise injection!

Ɍpinpointing examples from rados

NICTA

- Problem (informal): a malicious agency $\mathcal{A}$ has a big database of people identities $\mathcal{S}$. $\mathcal{A}$ intercepts some set of rados $\mathcal{S}^r$ sent over the network.
- Question: does there exist a subset of $\mathcal{S}$ of size m that may have been used to *approximately* craft the rados in $\mathcal{S}^r$?

pinpointing examples from rados

- Problem (informal): a malicious agency $\mathcal{A}$ has a big database of people identities $\mathcal{S}$. $\mathcal{A}$ intercepts a some set of rados $\mathcal{S}^r$ sent over the network.
- Question: does there exist a subset of $\mathcal{S}$ of size m that may have been used to *approximately* craft the rados in $\mathcal{S}^r$?

NP-HARD

Geometric hardness of recovering examples

- Suppose $\mathcal{A}$ is given *only* a set of rados. $\mathcal{A}$ knows **nothing else** about the examples $\mathcal{S}$, except that all lie in a ball of radius R.
- Then there exists a set of examples $\mathcal{S}'$ with just *one* more example, that produces the *same* rados but lies very far away (in Hausdorff distance)

$$D_H(\mathcal{S}, \mathcal{S}') = \Omega\left(\frac{R \log d}{\sqrt{d} \log m}\right) \qquad (m \geq 2^d)$$

$$D_H(\mathcal{S}, \mathcal{S}') = \Omega\left(\frac{R}{\sqrt{d}}\right) \qquad \text{(Otherwise)}$$

Geometric hardness of recovering examples

- Suppose $\mathcal{A}$ is given *only* a set of rados. $\mathcal{A}$ knows **nothing else** about the examples $\mathcal{S}$, except that all lie in a ball of radius R.

- Then there exists a set of [illegible] with just *one* [illegible] that produces the same rados but lies very far (Hausdorff distance)

$$D_H(S, S') [illegible] \quad (m \geq 2^d)$$

$$D_H(S, S') = \Omega\left(\frac{R}{\sqrt{d}}\right) \quad \text{(Otherwise)}$$

Hardness does not rely on the computational power at hand

Summary

- Learning over (small) sets of rados
  - may be **as efficient as learning over examples**
  - can **ensure additional properties** that are hard to meet with examples alone.
- The final classifier can be used **as is** to classify new observations.
- So far, we made no optimisation of the rados set for learning, just **plain random selection** — this was sufficient to beat supervised learning algorithms on fairly big domains :-)
- **Other domains** may benefit from the rado representation (incl. on-line and distributed learning).

Thank you ! Questions ?

DP-rados from non-DP examples

- RadoBoost + rados *vs.* RadoBoost + DP rados
- f = random binary feature