DATA 61

Fast learning from distributed datasets without entity matching

Giorgio Patrini, ***Richard Nock***, Stephen Hardy, Tiberio Caetano

ANU | **DATA61** | U. SYDNEY | UNSW | AMBIATA

Australian National University

THE UNIVERSITY OF SYDNEY

UNSW THE UNIVERSITY OF NEW SOUTH WALES

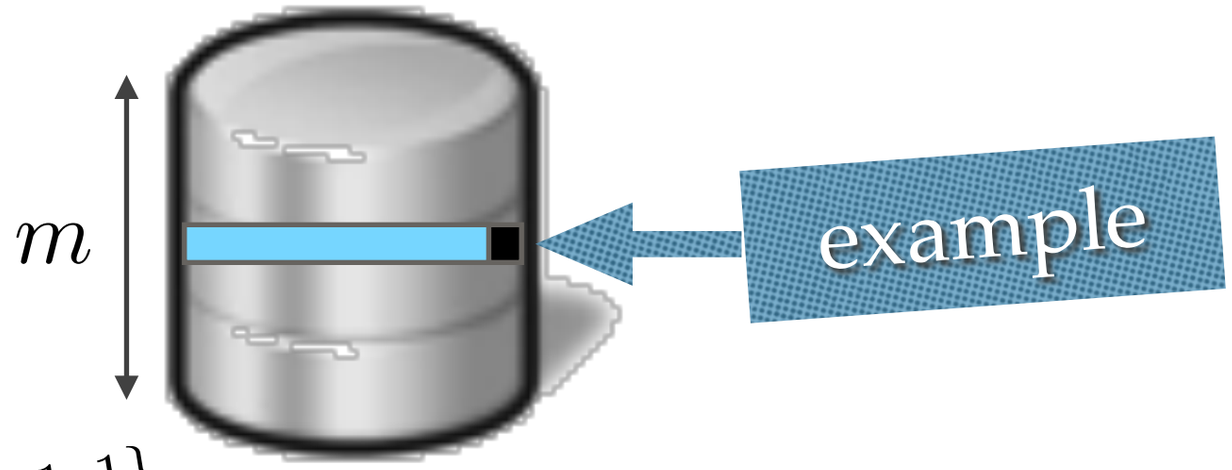
ambiata

CSIRO

www.data61.csiro.au

Setting: supervised learning

basic



- Input: $\mathcal{S} = \{(\boldsymbol{x}_i, \overset{\in\{-1,1\}}{y_i})\}_{i=1}^m$ with m examples, Γ sym. pos. def.
- Objective: minimize Ridge regularized square loss for $\boldsymbol{\theta}$:

$$\ell_{\mathrm{sql}}(\mathcal{S}, \boldsymbol{\theta}; \Gamma) \quad \doteq \quad \frac{1}{m} \cdot \sum_i (1 - y_i \boldsymbol{\theta}^\top \boldsymbol{x}_i)^2 + \boldsymbol{\theta}^\top \Gamma \boldsymbol{\theta} \ .$$

linear classifier

Setting: supervised learning

basic

EASY !

- Input: $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ with m examples, Γ symm. pos. def.
- Objective: minimize Ridge regularized square loss for $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^\star_{\text{ex}} = \left(\mathsf{X}\mathsf{X}^\top + m \cdot \Gamma\right)^{-1} \boldsymbol{\pi}_{\boldsymbol{y}}$$

$$\ell_{\text{sq}}(S, \boldsymbol{\theta}; \Gamma) \doteq \frac{1}{m} \sum_i (1 - y_i \boldsymbol{\theta}^\top \boldsymbol{x}_i)^2 + \boldsymbol{\theta}^\top \Gamma \boldsymbol{\theta} \, .$$

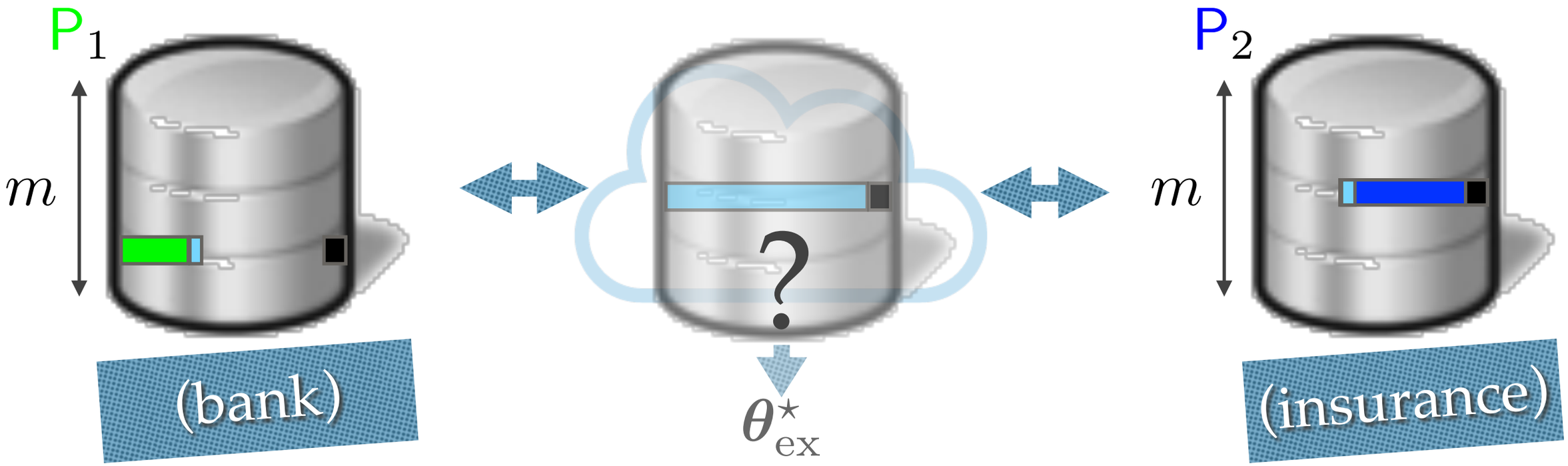
$$\mathsf{X} \doteq [\boldsymbol{x}_1 | \boldsymbol{x}_2 | \cdots | \boldsymbol{x}_m] \qquad \boldsymbol{\pi}_{\boldsymbol{y}} \doteq \sum_i y_i \cdot \boldsymbol{x}_i$$

rademacher observation (rado)

Setting: distributed supervised learning

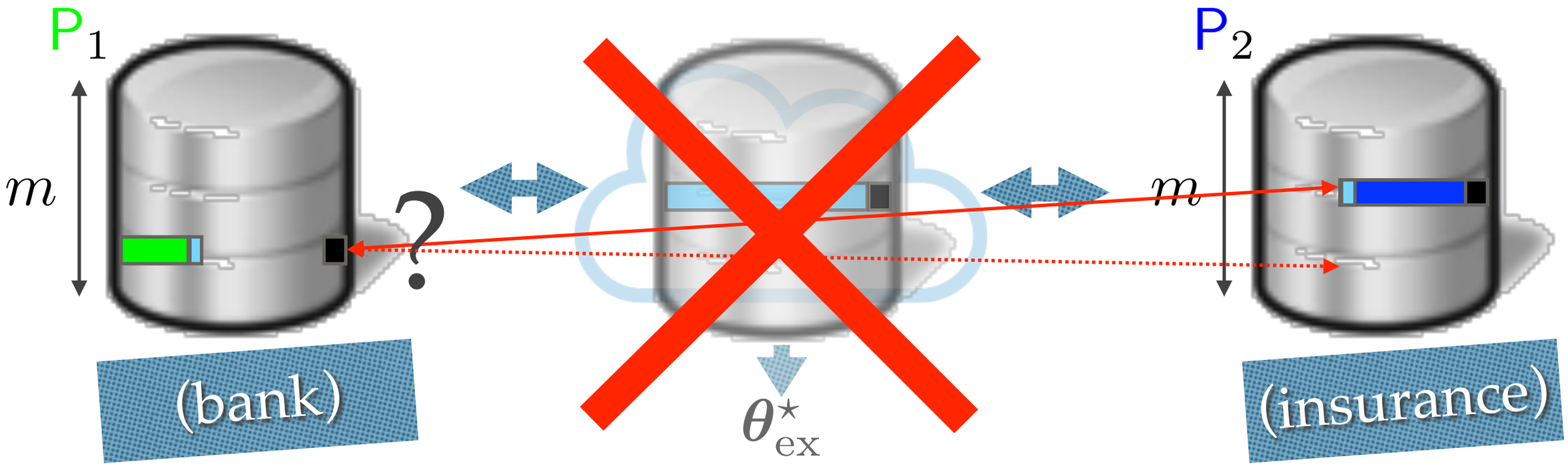
- Dataset "vertically" partitioned between 2 peers, P_1 and P_2.
- Have *few* shared features (postcode, gender, etc.)
- And lots of *specific* features (credit history, blood tests, etc.)

Setting: distributed supervised learning



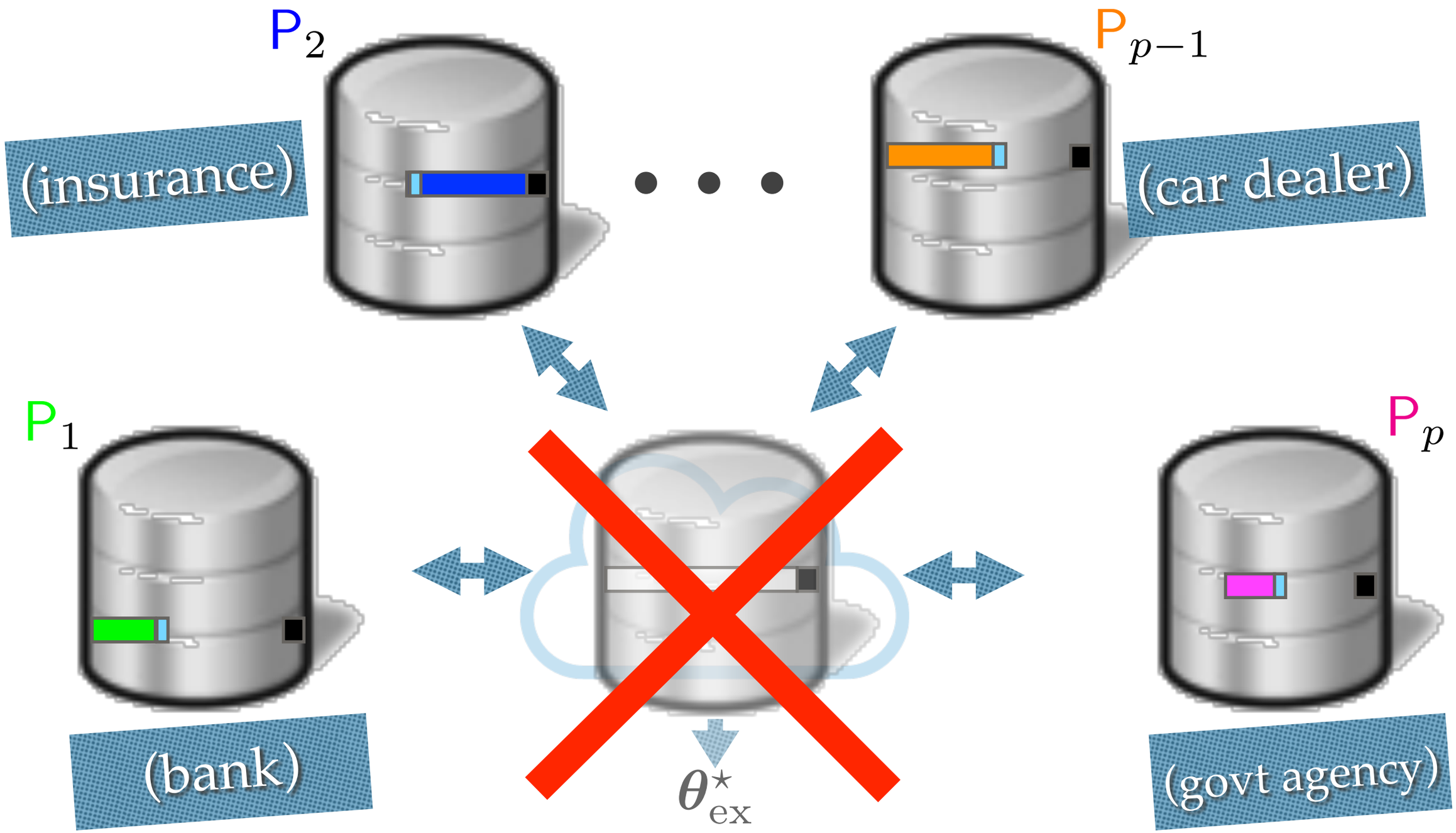
- Dataset "vertically" partitioned between 2 peers, P_1 and P_2.
- Have *few* shared features (postcode, gender, etc.)
- And lots of ***specific*** features (credit history, blood tests, etc.)
- Would like to learn $\boldsymbol{\theta}^\star_{\text{ex}}$ over the union of all features…

Setting: distributed supervised learning



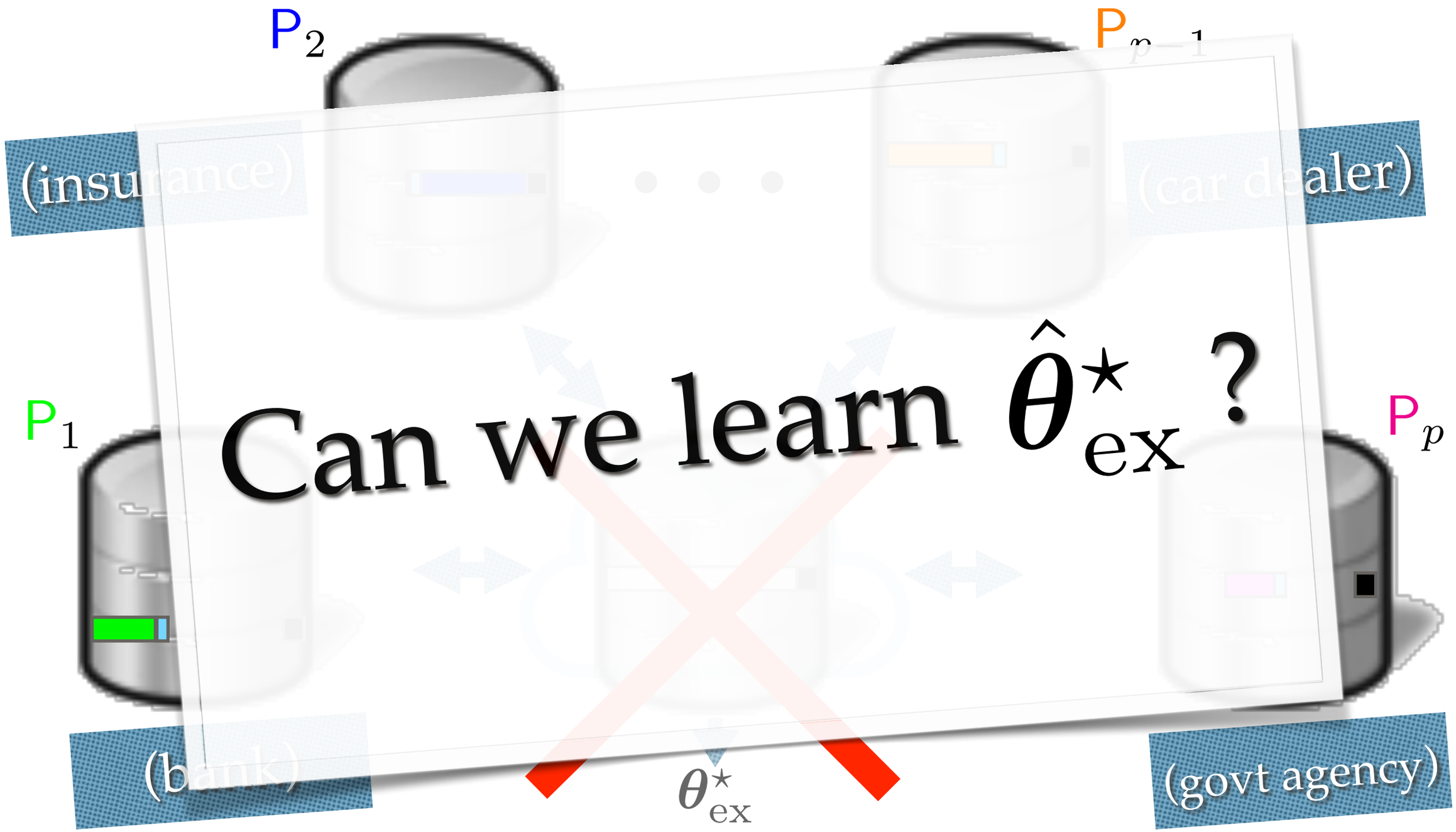
- Dataset "vertically" partitioned between 2 peers, P_1 and P_2.
- Have *few* shared features (postcode, gender, etc.)
- And lots of ***specific*** features (credit history, blood tests, etc.)
- Would like to learn $\boldsymbol{\theta}^\star_{\text{ex}}$ over the union of all features…
- But no entity matching possible ! (privacy / security)

Let's get more challenging !



- Same setting & constraint, but arbitrary number of peers

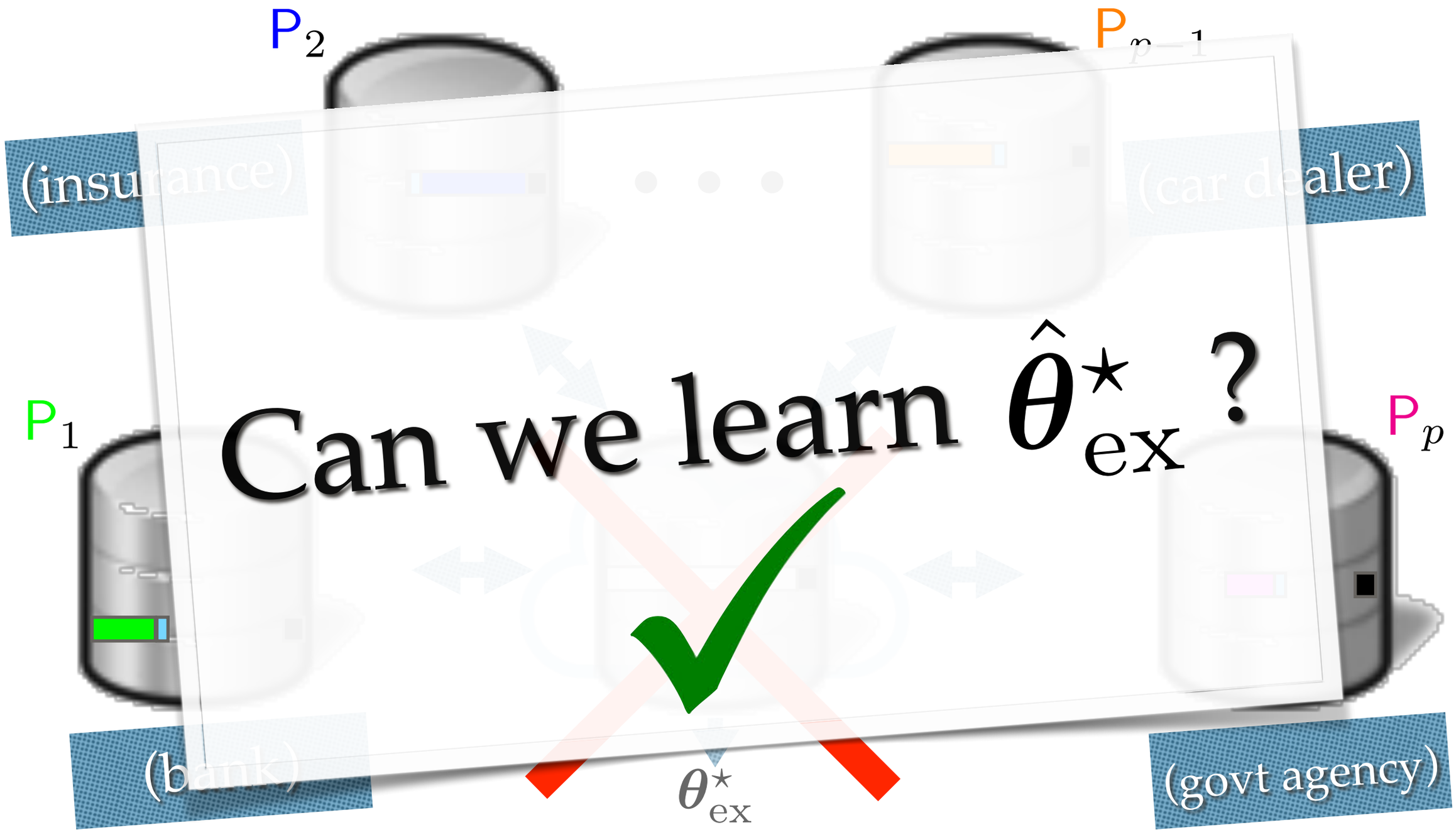
Let's get more challenging !



Can we learn $\hat{\theta}^\star_{\text{ex}}$?

- Same setting & constraint, but arbitrary number of peers

Let's get more challenging !



Can we learn $\hat{\theta}^\star_{\text{ex}}$?

- Same setting & constraint, but arbitrary number of peers

The trick

- Entity matching needed to build complete examples…

The trick

- Entity matching needed to build complete examples… ***but*** complete examples not needed to learn !

The trick

- Entity matching needed to build complete examples… ***but*** complete examples not needed to learn!

Bypass the construction of examples, and thereby the need to solve entity matching !

Main Theorem

- Entity matching needed to build complete examples… ***but*** complete examples not needed to learn !
- For ***any*** $\mathcal{S}$ and ***any*** $\boldsymbol{\theta}$,

$$\ell_{\mathrm{sql}}(\mathcal{S}, \boldsymbol{\theta}; \Gamma) = 1 + (4/m) \cdot \ell_{\mathrm{M}}(\mathcal{R}_{\mathcal{S},\Sigma_m}, \boldsymbol{\theta}; \Gamma)$$

Main Theorem

- Entity matching needed to build complete examples… ***but*** complete examples not needed to learn !
- For ***any*** $\mathcal{S}$ and ***any*** $\boldsymbol{\theta}$,

$$\ell_{\mathrm{sql}}(\mathcal{S}, \boldsymbol{\theta}; \Gamma) = 1 + (4/m) \cdot \ell_{\mathrm{M}}(\mathcal{R}_{\mathcal{S},\Sigma_m}, \boldsymbol{\theta}; \Gamma)$$

Ridge regularized square loss

Loss described using different data: *Rademacher observations.*

(Nock & al., ICML'15)

Main Theorem

- Entity matching needed to build complete examples… ***but*** complete examples not needed to learn !
- For ***any*** $\mathcal{S}$ and ***any*** $\boldsymbol{\theta}$,

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

$$\ell_{\mathrm{sql}}(\mathcal{S}, \boldsymbol{\theta}; \Gamma) = 1 + (4/m) \cdot \ell_{\mathrm{M}}(\mathcal{R}_{\mathcal{S}, \Sigma_m}, \boldsymbol{\theta}; \Gamma)$$

Ridge regularized square loss

Loss described using different data: *Rademacher observations.*

(Nock & al., ICML'15)

All Theorems (almost on 1 slide !)

- Entity matching needed to build complete examples… ***but*** complete examples not needed to learn !
- For ***any*** $\mathcal{S}$ and ***any*** $\boldsymbol{\theta}$,

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

$$\ell_{\text{sql}}(\mathcal{S}, \boldsymbol{\theta}; \Gamma) = 1 + (4/m) \cdot \ell_{\text{M}}(\mathcal{R}_{\mathcal{S}, \Sigma_m}, \boldsymbol{\theta}; \Gamma)$$

- ***Rado*** set $\mathcal{R}_{\mathcal{S}, \Sigma'} = \{\boldsymbol{\pi}_{\boldsymbol{\sigma}} \doteq \sum_{y_i = \sigma_i} y_i \cdot \boldsymbol{x_i} : \boldsymbol{\sigma} \in \Sigma'\}$, with $\Sigma' \subseteq \Sigma_m$

All Theorems (almost on 1 slide !)

- Entity matching needed to build complete examples… ***but*** complete examples not needed to learn !
- For ***any*** $\mathcal{S}$ and ***any*** $\boldsymbol{\theta}$,

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

$$\ell_{\mathrm{sql}}(\mathcal{S}, \boldsymbol{\theta}; \Gamma) = 1 + (4/m) \cdot \ell_{\mathrm{M}}(\mathcal{R}_{\mathcal{S}, \Sigma_m}, \boldsymbol{\theta}; \Gamma)$$

- ***Rado*** set $\mathcal{R}_{\mathcal{S}, \Sigma'} = \{\boldsymbol{\pi}_{\boldsymbol{\sigma}} \doteq \sum_{y_i = \sigma_i} y_i \cdot \boldsymbol{x}_i : \boldsymbol{\sigma} \in \Sigma'\}$, with $\Sigma' \subseteq \Sigma_m$
- A significant subset $\mathcal{R}_{\mathcal{S}, \Sigma^\star} \subset \mathcal{R}_{\mathcal{S}, \Sigma_m}$ with exponential size (in m) can be built without knowing entity matching
- classifier $\boldsymbol{\theta}^\star_{\mathrm{rad}} \doteq \arg\min_{\boldsymbol{\theta}} \ell_{\mathrm{M}}(\mathcal{R}_{\mathcal{S}, \Sigma'}, \boldsymbol{\theta}; \Gamma)$ is ***faster*** to build than $\boldsymbol{\theta}^\star_{\mathrm{ex}}$
- …and we ***also*** have $\boldsymbol{\theta}^\star_{\mathrm{rad}} \to \boldsymbol{\theta}^\star_{\mathrm{ex}}$

All algorithms (on 1 slide !)

- Step 1: build a particular subset of $\mathcal{R} \subset \mathcal{R}_{\mathcal{S},\Sigma^\star}$ with $|\mathcal{R}| \leq m$

- Step 2: build $\boldsymbol{\theta}^\star_{\text{rad}}$: it can be shown that

$$\boldsymbol{\theta}^\star_{\text{rad}} = \left(\mathsf{R}\mathsf{R}^\top + \gamma \cdot \Gamma\right)^{-1} \mathsf{R}\mathbf{1}$$

where R stacks $\mathcal{R}$ in columns and $\gamma \in \mathbb{R}_{+*}$

All algorithms (on 1 slide !)

- Step 1: build a particular subset of $\mathcal{R} \subset \mathcal{R}_{\mathcal{S},\Sigma^\star}$ with $|\mathcal{R}| \leq m$

$O(\text{nb_features} \cdot m)$

- Step 2: build $\boldsymbol{\theta}^\star_{\text{rad}}$: it can be shown that

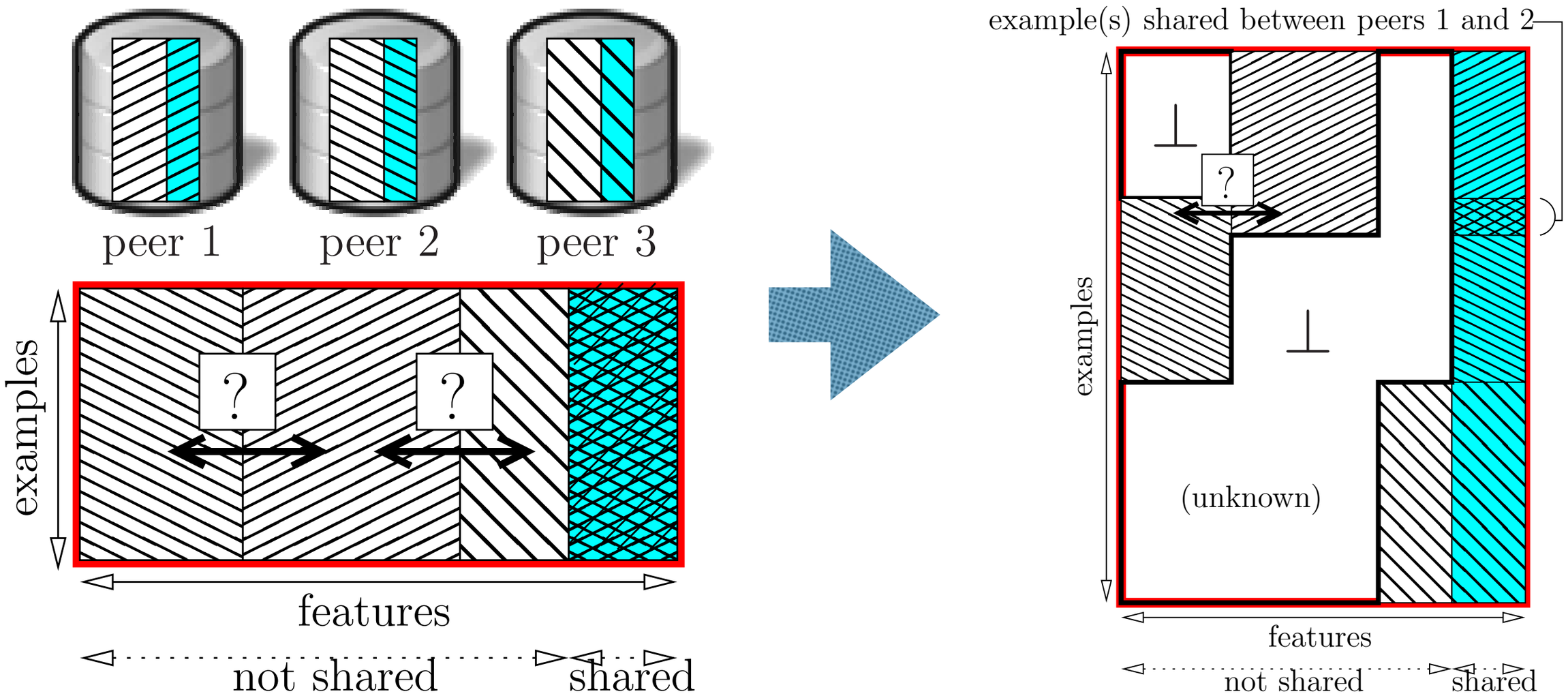
$$\boldsymbol{\theta}^\star_{\text{rad}} = \left(\mathsf{R}\mathsf{R}^\top + \gamma \cdot \Gamma\right)^{-1} \mathsf{R}\mathbf{1}$$

where R stacks $\mathcal{R}$ in columns and $\gamma \in \mathbb{R}_{+*}$

$O(\text{nb_features}^2 \cdot m)$

Generalisation

- Works for *any* number of peers
- Works if we relax the vertical partition assumption



Experiments

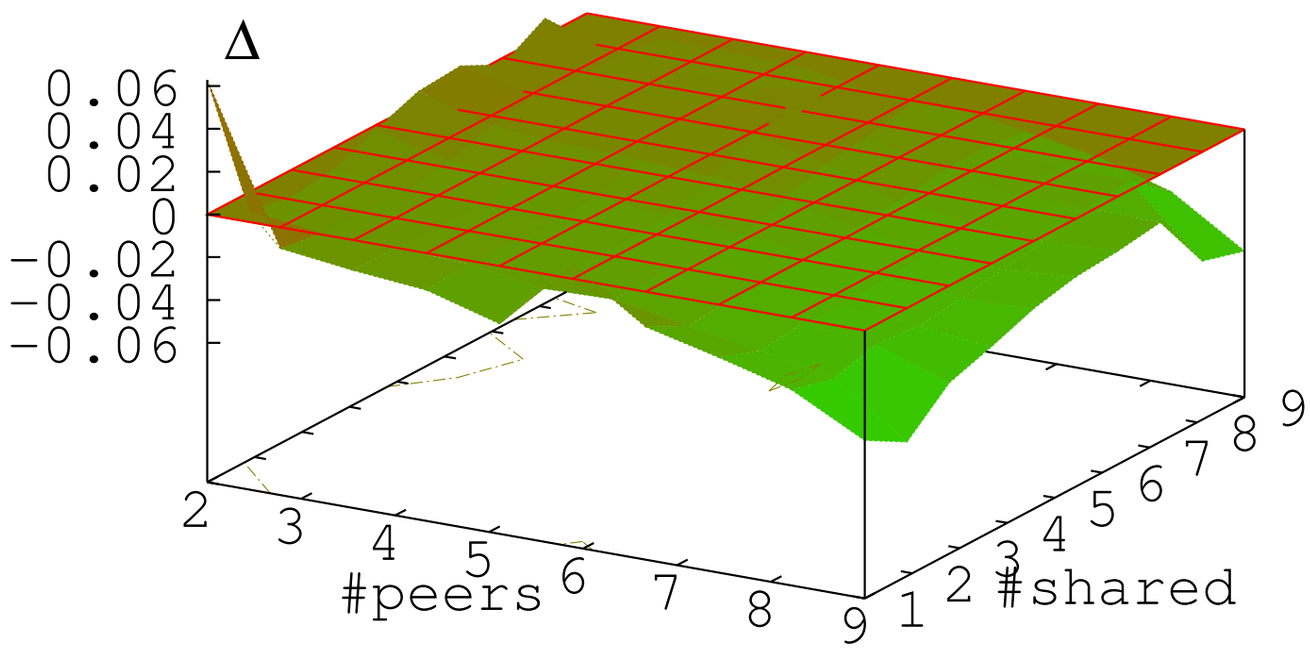
- simulation: split datasets — vary #peers, #shared(features), #bins, #joint_examples
- Little experimental influence of #bins (in range 2-5)
- Tested no #joint_examples (peers see all different examples, harder) + small % of #joint_examples

objective: beat the ***best*** peer in hindsight

Experiments

- vary #peers, #shared(features), #joint_examples = **0**

$$\Delta \doteq \hat{p}_{\text{err}}(\text{our algo}) - \min_j \hat{p}_{\text{err}}(\mathsf{P}_j) \;\; (\in [-1, 1])$$

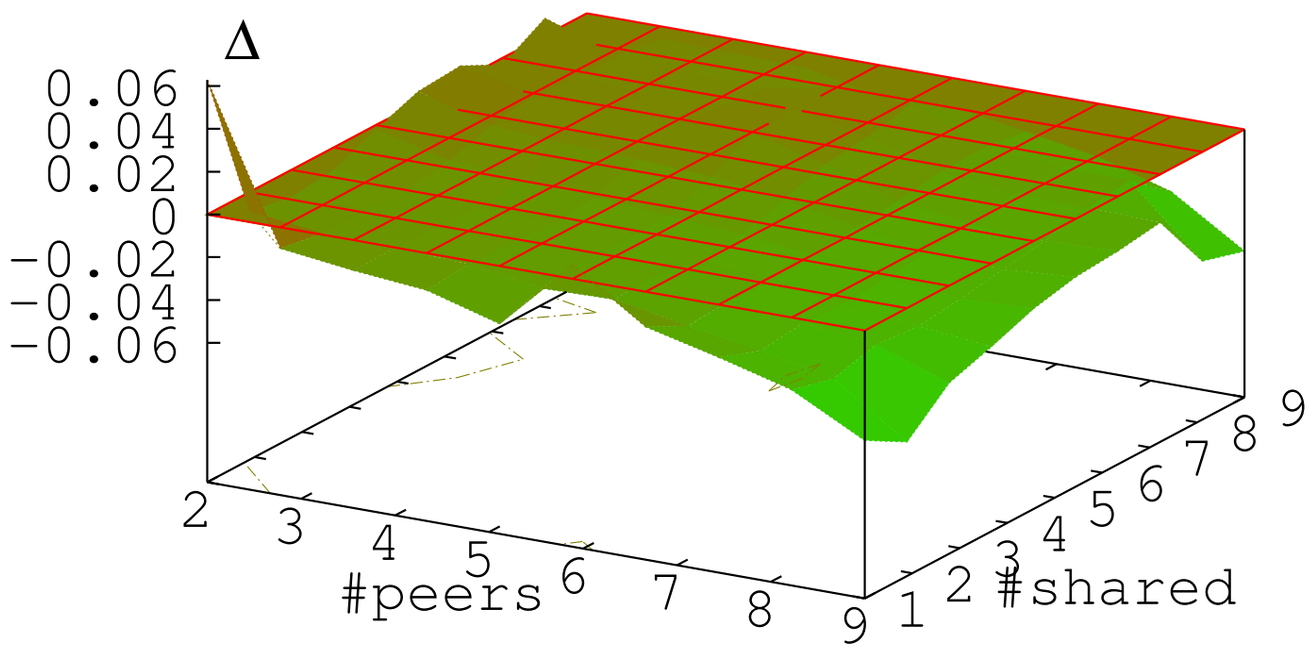


UCI Ionosphere

Experiments

- vary #peers, #shared(features); #joint_examples = **0**

$$\Delta \doteq \hat{p}_{\text{err}}(\text{our algo}) - \min_j \hat{p}_{\text{err}}(\mathsf{P}_j) \;\; (\in [-1, 1])$$



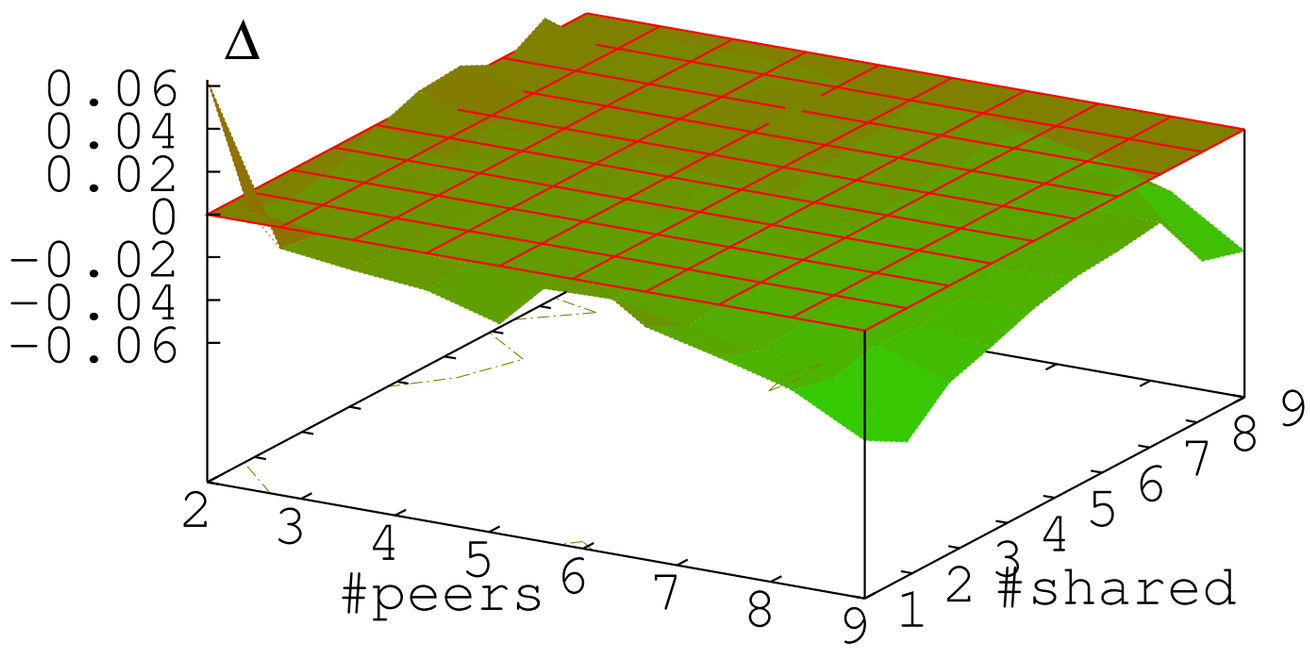
Almost systematically beats all peers

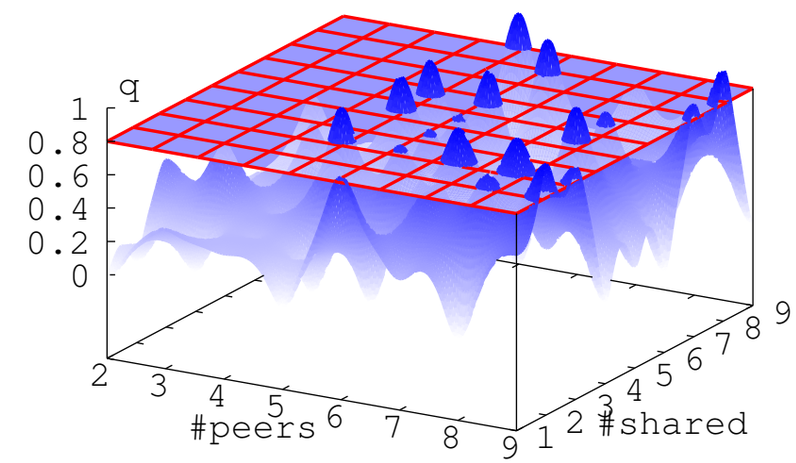
UCI Ionosphere

Experiments

- vary #peers, #shared(features); #joint_examples = **0**

$$q \doteq \text{ proportion of peers } \textit{statistically} \text{ beaten by our algo}$$



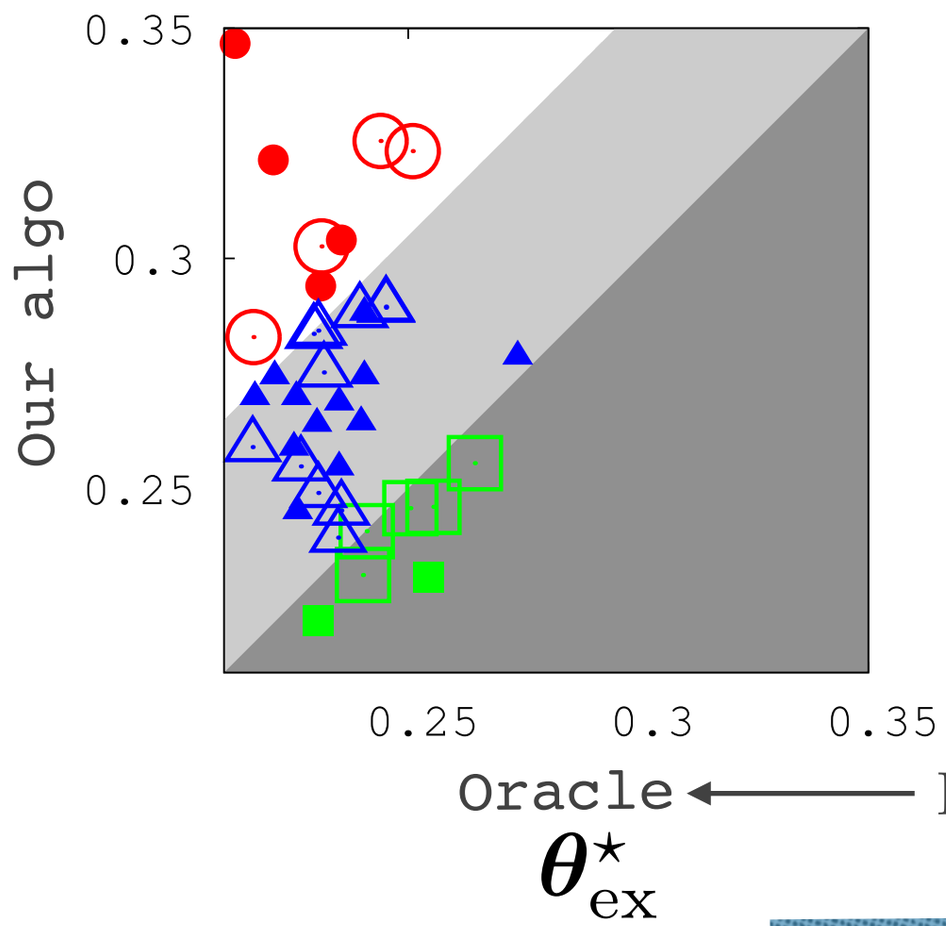


Almost systematically beats all peers… but not always significantly

UCI Ionosphere

Experiments

- See poster, paper & long ArXiv version for more experiments



Sometimes we are worse (statistically or not)
Sometimes we are better (not statistically) !

Knows the solution to ER !

UCI Sonar

Conclusion

- Entity matching / entity resolution may not be a bottleneck to supervised learning from distributed data; bypassing it is achieved by hardworking loss functions
- Works for *any* number of peers
- Works if we relax the vertical partition assumption
- In many cases, there is a minimal number of peers p after which we significantly beat all peers. Sometimes, p=2.
- Sometimes, we can also compete with $\boldsymbol{\theta}^{\star}_{\mathrm{ex}}$ (the Oracle)!

Thank you, questions ?