

Fast Learning from Distributed Datasets without Entity Matching

Giorgio Patrini

The Australian National University & Nicta
 giorgio.patrini@anu.edu.au

Richard Nock

Nicta & The Australian National University
 richard.nock@nicta.com.au

Stephen Hardy

Nicta
 stephen.hardy@nicta.com.au

Tiberio Caetano

Ambiata, The Australian National University & The University of New South Wales
 tiberio.caetano@gmail.com

Abstract

Consider the following data fusion scenario: two datasets/peers contain the same real-world entities described using partially shared features, *e.g.* banking and insurance company records of the same customer base. Our goal is to learn a classifier in the cross product space of the two domains, in the hard case in which no shared ID is available —*e.g.* due to anonymization. Traditionally, the problem is approached by first addressing entity matching and subsequently learning the classifier in a standard manner. We present an end-to-end solution which bypasses matching entities, based on the recently introduced concept of *Rademacher observations* (rados). Informally, we replace the minimisation of a loss over examples, which requires to solve entity resolution, by the *equivalent* minimisation of a (different) loss over rados. Among others, key properties we show are (i) a potentially huge subset of these rados *does not require* to perform entity matching, and (ii) the algorithm that provably minimizes the rado loss over these rados has time and space complexities *smaller* than the algorithm minimizing the equivalent example loss. Last, we relax a key assumption of the model, that the data is vertically partitioned among peers — in this case, we would not even know the *existence* of a solution to entity resolution. In this more general setting, experiments validate the possibility of significantly beating even the *optimal* peer in hindsight.

Keywords: entity resolution, distributed learning, Rademacher observations, square loss, Ridge regularization.

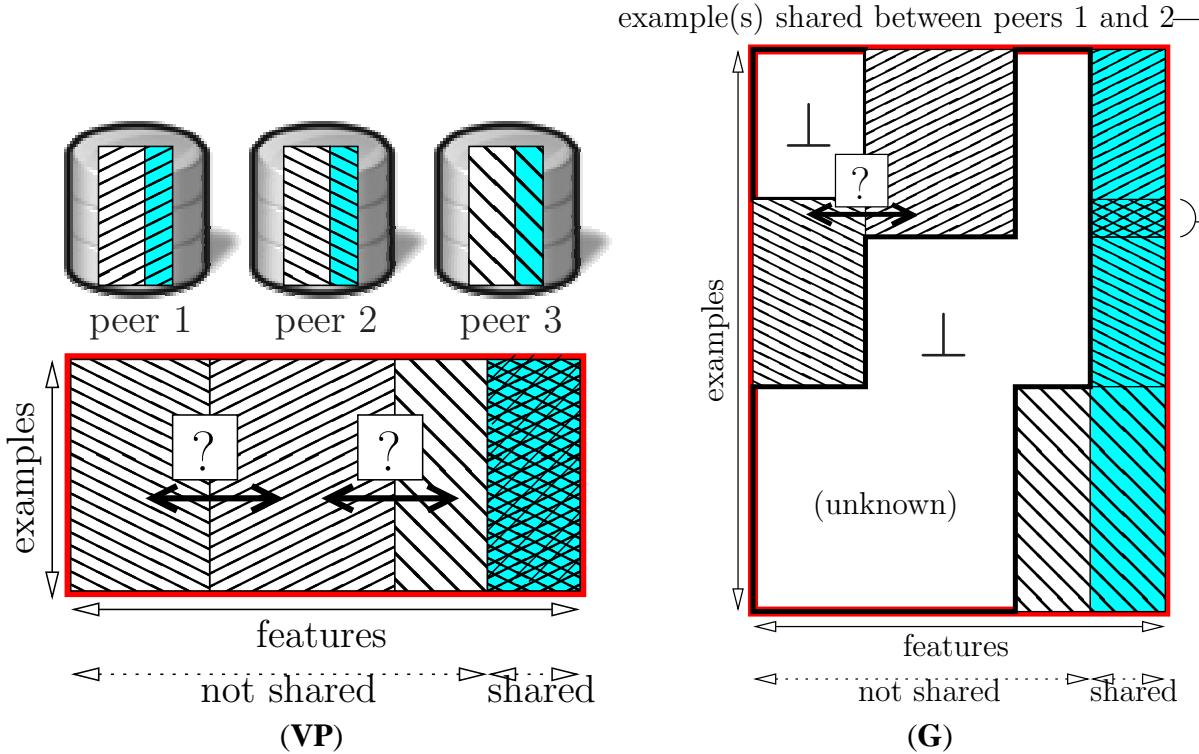


Figure 1: Schematic views of our settings, with $p = 3$ peers. In both cases, some features (cyan) are described in each peer (best viewed in color) and one these shared features is a class. Non-shared features are split among peers. A so-called *total* sample \mathcal{S} is figured by the red rectangle. *Left*: in the vertical partition (**VP**) case, all peers see different views of the same examples, but do not know who is who among their datasets ("?"). Hence, each bit of the total sample is seen by one peer. *Right*: in the more general setting (**G**), it is not even known whether one example, viewed by a peer, also exist in other peers' datasets. In this case, there may be a lot of missing data (\perp), but it is not known which example has missing data.

1 Introduction

Learning from massively distributed data collections and multiple information sources has become a pivotal problem, yet it faces critical challenges, among which is the fact that it relies on reconstructing consistent examples from diverse features distributed between different data handling *peers*. Exhaustive search to solve this problem is simply not scalable, nor communication efficient, and sometimes not even accurate [11, 34].

— A key technical message of our paper is:

Entity resolution can be bypassed to carry out supervised learning almost as accurate as if its solution were known.

A main motivation of this work comes from the reported experience that combining features from different sources leads to better predictive power. For instance, insurance and banking data together can improve fraud detection; shopping records well complement medical history for estimating risk of disease [29]; joining heterogeneous data helps prediction in genomics [16, 33];

Peer 1		Peer 2	
		shared	shared
	x_1	x_3	c
e_1	1	1	1
e_2	-1	1	1

Peer 1		Peer 2	
		shared	shared
	x_2	x_3	c
e'_1	-1	1	1
e'_2	1	1	1

Table 1: A simple case of the **(VP)** setting, with $p = 2$ peers, with two shared variables x_3 and c (the class to predict). This toy example has binary description features and a binary shared feature, but this restriction does not need to hold in the general case. For example, each shared feature can be any categorical/ordinal feature, like “postcode”, “age-bracket”, etc.

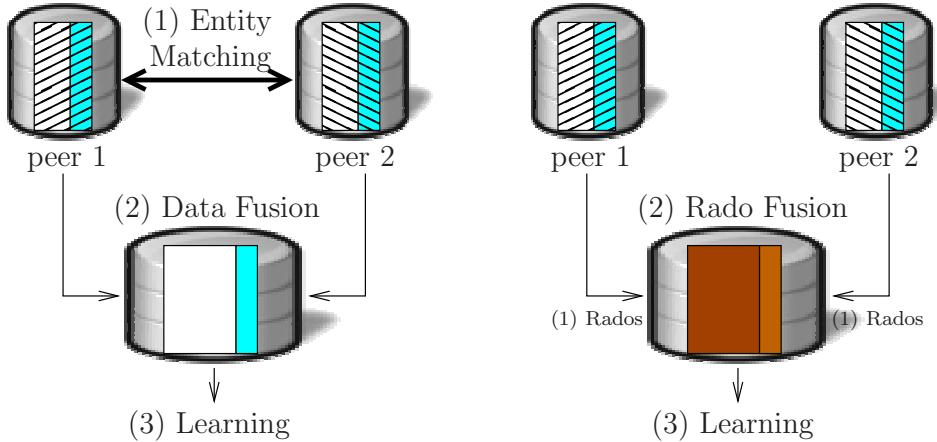


Figure 2: Learning on top of ER (left) or with rados (right).

security agencies integrate various sources for terrorism intelligence [28, 9, 27].

Typical data fusion methods however rely on a known map between entities [6], *i.e.*, peers have partially different views of the *same* examples. Instead, we assume the datasets do not share a common ID, as shown in Figure 1 ((VP), left); that is, for example, the case when data collection of was performed independently by each peer, or when sources were deliberately anonymized. *Entity resolution* (ER), or entity matching [10], would be the traditional approach for reconciling entities with no shared ID¹. It approximates a JOIN operation, assuming that some of the attributes are shared, *e.g.*, *age-band*, *gender*, *postcode* (etc.), and hence can be used as “weak IDs”. Most techniques for ER are based on similarity functions and thresholding: candidate entities are selected as matches when their similarity is above a threshold. Both components can be tuned on some ground truth matches and effectively enhanced with learning techniques [5, 10]. The various metrics of ER encompass lots of different parameters, including generality, accuracy, soundness, scalability, parallelizability [25]. The standard pipeline for learning with ER is depicted in Figure 2 (left): (1) entities are matched based on similarity and heuristics, (2) they are merged in one unique database and (3) a model is learnt on the joint data. Common issues in fusion, such as *conflicts* and *heterogeneity* [6], are not considered in this work.

From a high level view, ER integrates data as a pre-process for other tasks. When it comes

¹This is clearly non trivial: if just two rows in each dataset have the same exact values for the shared features across the p peers, this yields 2^p possible matchings for the reconstruction of the two examples involved.

to learning from ER’ed data, small changes in ER can have large impact on evaluating classifiers, even for simple classifiers as linear models. To see this, suppose we are in the toy example of Table 1. Here, all shared variables have the same values, so entity matching has two potential solutions (notice that one of the shared variable is class c). One, say ER1, is matching e_1 with e'_1 and e_2 with e'_2 . We denote the examples obtained by $e_{11} \doteq ((1, -1, 1), 1)$ and $e_{22} \doteq ((-1, 1, 1), 1)$ (an example is a pair (observation, class)). The other solution, say ER2, is matching e_1 with e'_2 and e_2 with e'_1 . We denote the examples obtained by $e_{12} \doteq ((1, 1, 1), 1)$ and $e_{21} \doteq ((-1, -1, 1), 1)$. Now, consider linear classifier $\theta = (1, 1, 1) \in \mathbb{R}^3$; the class it gives is the sign of its inner product with an observation, $\theta(z) = \text{sign}(\theta^\top z)$. While θ classifies perfectly on $\{e_{11}, e_{22}\}$ (zero error), it classifies no better than random on $\{e_{12}, e_{21}\}$ (error 50%).

This is a potential consequence of non-accurate ER in a setting in which we *know* that there is a solution to ER, *i.e.* a one-one matching between data peers that recovers the examples as they are in the total sample. What happens if remove this assumption, *i.e.* if we remove the assumption that each example is seen by *all* peers? This is a much more realistic model. Since there is no shared ID — and the data may have been anonymized — we are not even in a situation where we can guarantee that a specific client of the bank *is*, or *is not*, a client of the insurance company. Thus, there may be significant unknown data to reconstruct the total sample \mathcal{S} (Figure 1), but we do not know which specific examples have missing features. This is our most general setting, **(G)**, shown in Figure 1 (right).

To cope with **(VP)** or **(G)**, we use a recently introduced trick to learn from private data [21]: examples are not necessary to learn an accurate linear classifier. We insist on the fact that “accurate” refers to the quality of the class prediction for observations and examples. The input of the algorithm consists of *Rademacher observations*, rados. One rado is just a sum, over a subset of examples, of the observations times their class. Surprisingly, we can not only learn with data on this form but the output classifier does not require any post-processing since it is the same as if we were learning with example.

Contributions — Our contribution starts from noticing that many rados are invariant to the selection of different solutions for entity resolution. For example, consider again Table 1. Since all classes are positive, computing a rado is just summing observations. Let $\pi_{ij,kl}$ be the rado that sums those of examples e_{ij} and e_{kl} . Then, surprisingly, regardless of the solution to ER, this rado is the *same*:

$$\begin{aligned}
 (\text{E1}) \quad \pi_{11,22} &= (1, -1, 1) + (-1, 1, 1) \\
 &= (0, 0, 2) \\
 &= (1, 1, 1) + (-1, -1, 1) = \pi_{12,21} \quad (\text{E2}) .
 \end{aligned}$$

This, as we show, always holds in the **(VP)** setting: there exists a huge, *i.e.*, of potential exponential size, set of rados that match the set of rados that could be built *knowing* the true entity resolution. In the most general setting, **(G)**, we show that a very simple transformation of the rados, involving only the shared features, has in expectation the same properties. We give the algorithm that builds these rados. It is easily parallelizable and requires *sublinear* communication, *i.e.* the amount of information that transits is no larger — and may be much smaller — than the size of all peers’ data.

These “ideal” rados are not just interesting *per se*: learning from them (Figure 2, right) is both efficient and accurate. We show that using them leads approximating the classifier that would be

optimal *on the set of all (ideally ER’ed) examples*. This involves three technical contributions. The first is an elementary proof that the minimisation of the Ridge regularized square loss [14] (on examples) is equivalent to the minimisation of a regularized rado loss, which we call the M-loss. We then give the closed-form solution for the classifier minimizing the M-loss. Surprisingly, it shows that the minimisation of the regularized M-loss, over the complete (eventually exponential-size) set of ”ideal” rados can be done not just in polynomial time: it is in general *faster* than the minimization of the Ridge regularized square loss over examples. Finally, the optimal M-loss classifier, learnt using only the set of ”ideal” rados, converges (as the number of shared features increases) to the minimizer of the Ridge regularized square loss over *all* ideally ER’ed examples. In other words, as the number of shared features increases or as the number of modalities of shared features increases, we are *guaranteed* that the classifier learned over rados will converge to the best classifier learned over examples.

Last, but not least, while we focus on the two-classes setting, description features need not be boolean. There is in fact no restriction apart from the fact that shared features are treated as ordinal instead of plain real: if one feature had as many modalities as there are examples, then there would be no need to address ER. The rest of this paper is as follows. Section §2 provides preliminaries. § 3 follows that shows how to learn from distributed data to minimise, indirectly, the Ridge regularized square loss over the ER’ed complete data. § 4 presents experimental results. Finally, § 5 discusses our approach and § 6 concludes with open problems.

2 Preliminaries

Learning setting We let $[n] \doteq \{1, 2, \dots, n\}$ for $n \in \mathbb{N}_*$; boldfaces like \boldsymbol{x} indicate vectors, whose coordinates are denoted as x_i . We briefly recall the task of standard (binary) classification with linear models $\boldsymbol{\theta}$ as learning a predictor for label (or class) $y \in \{-1, +1\}$, from a *total* (learning, training) sample $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i \in [m]\}$. Each example is an observation-label pair $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \{-1, +1\}$, with $\mathcal{X} \subseteq \mathbb{R}^d$ the *feature space*, and it is drawn i.i.d. from an unknown distribution. It is convenient to let $\mathcal{X} \doteq \times_{k=1}^d \mathcal{X}_k$. We reserve the word *entity* for a generic record in a dataset, the object of matching, and *attributes* or *features* to its fields.

Our learning setting departs from the standard setting in what follows. Instead of one total training sample, we have p (sub)samples, \mathcal{S}^j of size m_j , $j \in [p]$ for some $p > 1$. Each one is defined in its own feature space $\mathcal{X}^j \doteq \times_{k=1}^{d_j} \mathcal{X}_{j_k}$, where $j_k \in [d], \forall k$. To get a simple case of this framework, shown in Figure 1, one may see each $\mathcal{S}^j \doteq \{(\boldsymbol{x}_i^j, y_i^j), i \in [m_j]\}$ handled by a *peer* P_j . We rely on the following assumption:

- (G) The class, and a subset of features \mathcal{J} from \mathcal{X} , are shared by all peers. Each other feature is exclusive to one peer.

Hence, each of the dimensions of \mathcal{J} is in all \mathcal{X}^j ’s. There exists $\dim(\mathcal{J}) + 1$ columns that represent the same set of variables among peers, and one of them is the class. This is a very weak and realistic assumption for the features in \mathcal{J} , as well as for labels, in at least two situations. The first is our setting **(VP)**, which is a gold standard of database frameworks, when the domain is vertically partitioned for the non-shared features, implying $m_j = m_{j'} = m, \forall j, j' \in [p]$. In this case, there exists a one-to-one mapping between the peers’ rows, but it may be extremely hard to compute [25]. The other scenario is when at least one peer has classes, as that turns out to be what is

sufficient for all other peers to get labels as well, by the use of algorithms that learn with label proportions [23, 24], as argued in Section 5. The assumption that each non-shared feature is seen by exactly one peer simplifies the technicalities: we discuss relaxing this assumption in Section 3 (after Theorem 5).

Rademacher observations In the standard classification model, a rademacher observation (rado) is a simple transformation of the examples in sample \mathcal{S} . Let $\sigma \in \Sigma_m \doteq \{-1, 1\}^m$. Then rado π_σ is $\pi_\sigma \doteq \sum_{y_i=\sigma_i} y_i \cdot \mathbf{x}_i$ [20, 21], where $y_i \cdot \mathbf{x}_i$ is an *edge* vector. In our distributed setting, we extend the definition in the following way. We let $s \in \mathcal{J}$ denote a *signature*, and $\forall y \in \{-1, +1\}$ and peer P_j ,

$$\pi_{(s,y)}^j \doteq \text{proj}_{\mathcal{X}^j \setminus \mathcal{J}} \left(\sum_{i=1}^{m_j} 1_{\text{proj}_{\mathcal{J}}(\mathbf{x}_i^j) = s \wedge y_i^j = y} y_i^j \cdot \mathbf{x}_i^j \right) . \quad (1)$$

Notation 1. is the indicator function, and $\text{proj}_{\mathcal{J}}(\mathbf{z})$ denotes the restriction of \mathbf{z} to \mathcal{J} . In short, $\pi_{(s,y)}^j$ sums edge vectors local to P_j whose examples match signature s and class y . Let $\mathcal{F}(\mathbf{z})$ be the set of features of \mathbf{z} , assumed to be in \mathcal{X} . We also define, for any $\mathcal{F}' \supseteq \mathcal{F}(\mathbf{z})$, $\text{lift}_{\mathcal{F}'}(\mathbf{z})$ to be the vector \mathbf{z}' described using \mathcal{F}' such that $\text{proj}_{\mathcal{F}(\mathbf{z})}(\mathbf{z}') = \mathbf{z}$ and $\text{proj}_{\mathcal{F}' \setminus \mathcal{F}(\mathbf{z})}(\mathbf{z}') = \mathbf{0}$. While $\text{proj}_{\mathcal{F}}(\mathbf{z})$ removes coordinates of \mathbf{z} , $\text{lift}_{\mathcal{F}'}(\mathbf{z})$ "completes" the coordinates of \mathbf{z} with zeroes.

By analogy with entity resolution [32], we define *block rados* as rados, lifted to \mathcal{X} , that are the (weighted) sums of examples matching a particular signature and class in all peers.

Definition 1 For any $s \in \mathcal{J}$, $y \in \{-1, 1\}$, $u \in \mathbb{R}$ the *u-basic block (BB) rado* for pair (s, y) is

$$\pi_{(s,y)}^u \doteq u \cdot \text{lift}_{\mathcal{X}}(y \cdot s) + \sum_{j=1}^p \text{lift}_{\mathcal{X}}(\pi_{(s,y)}^j) . \quad (2)$$

Let $\mathcal{J}_+ \doteq \mathcal{J} \times \{-1, 1\}$, and $\mathcal{J}_* \doteq \{(s, y) \in \mathcal{J}_+ : \exists j \in [p], \pi_{(s,y)}^j \neq \mathbf{0}\}$. This latter set, which can easily be computed from all peers, has cardinal $m_* \doteq |\mathcal{J}_*| \leq m$, and even $m_* \ll m$ when few features are shared. For any $\mathbf{u} \in \mathbb{R}^{m_*}$, we let $\mathcal{R}_B^u \doteq \{\pi_{v_i}^{u_i}, \forall i \in [m_*]\}$ denote the set of each u_i -BB rado, each coordinate of \mathbf{u} being in one-one correspondence with an element of \mathcal{J}_* (represented by \mathbf{v}_i). A superset of \mathcal{R}_B^u is interesting, that considers all sums of vectors from \mathcal{R}_B^u :

$$\mathcal{R}_*^u \doteq \left\{ \sum_{i \in \mathcal{U}} \pi_{v_i}^{u_i}, \forall \mathcal{U} \subseteq [m_*] \right\} . \quad (3)$$

We call \mathcal{R}_*^u the set of *u-block rados*. Notice that we may have $|\mathcal{R}_*^u| = \Omega(2^{\sum_j |\mathcal{S}^j|})$. It is therefore intractable in general to *explicitly* compute \mathcal{R}_*^u . However, $|\mathcal{R}_B^u| = O(\sum_j |\mathcal{S}^j|)$ and to compute it, we just need the set of $\pi_{(s,y)}^j$, hence a communication complexity that can be much smaller than $\sum_j |\mathcal{S}^j|$.

3 Building and learning from BB rados

We address two questions: why/how we can use (basic block) rados to learn accurate classifiers, and how we should fix \mathbf{u} .

Example vs rado losses Learning $\boldsymbol{\theta}$ on \mathcal{S} is done by minimizing a loss function. Here, we consider the Ridge regularized square loss [14] (Γ is sym. positive definite, SPD),

$$\ell_{\text{sql}}(\mathcal{S}, \boldsymbol{\theta}; \Gamma) \doteq \frac{1}{m} \cdot \sum_i (1 - y_i \boldsymbol{\theta}^\top \mathbf{x}_i)^2 + \boldsymbol{\theta}^\top \Gamma \boldsymbol{\theta} . \quad (4)$$

It is crucial to remark that this loss is described over the total sample \mathcal{S} of examples (see the red rectangle in Figure 1). This is the loss we want to minimize, exactly or approximately. One reason we choose this loss is that in the standard classification framework, it admits a simple closed form solution:

$$\boldsymbol{\theta}_{\text{ex}}^* \doteq \arg \min_{\boldsymbol{\theta}} \ell_{\text{sql}}(\mathcal{S}, \boldsymbol{\theta}; \Gamma) = (\mathbf{X} \mathbf{X}^\top + m \cdot \Gamma)^{-1} \boldsymbol{\pi}_y , \quad (5)$$

where $\mathbf{X} \doteq [\mathbf{x}_1 | \mathbf{x}_2 | \cdots | \mathbf{x}_m]$, and so, $\mathbf{X} \mathbf{X}^\top = \sum_i \mathbf{x}_i \mathbf{x}_i^\top$. Remark that $\boldsymbol{\theta}_{\text{ex}}^*$ involves one rado, $\boldsymbol{\pi}_y$. For any $\mathcal{P} \subseteq \{-1, 1\}^m$, we let $\mathcal{R}_{\mathcal{S}, \mathcal{P}} \doteq \{\boldsymbol{\pi}_\sigma : \boldsymbol{\pi}_\sigma \in \mathcal{P}\}$ denote the set of rados that can be crafted from \mathcal{P} using \mathcal{S} .

Definition 2 *The M-loss over $\mathcal{R}_{\mathcal{S}, \mathcal{P}}$ of classifier $\boldsymbol{\theta}$ is:*

$$\ell_M(\mathcal{R}_{\mathcal{S}, \mathcal{P}}, \boldsymbol{\theta}) \doteq - \left(\mathbb{E}_{\mathcal{P}}[\boldsymbol{\theta}^\top \boldsymbol{\pi}_\sigma] - \frac{1}{2} \cdot \mathbb{V}_{\mathcal{P}}[\boldsymbol{\theta}^\top \boldsymbol{\pi}_\sigma] \right) , \quad (6)$$

where expectation and variance are computed with respect to the uniform sampling of σ in \mathcal{P} .

What is inside the parenthesis looks like a (vanilla) Markowitz mean-variance criterion [19] — “vanilla” because there is no variable coefficient for the risk aversion. What this means is that a good classifier trained on rados should have large “return” and small “risk”, where the risk is the variance of its predictions and the return is its inner product with the expected rado.

The Theorem to follow shows that what was known for the logistic loss in [21] also holds for the square loss: there exists a loss described over rados, ℓ_M , such that $\ell_{\text{sql}}(\boldsymbol{\theta})$ (dependences on other parameters omitted) is equal to a strictly increasing function of $\ell_M(\boldsymbol{\theta})$, for any $\boldsymbol{\theta}$. Hence, minimizing $\ell_{\text{sql}}(\boldsymbol{\theta})$ over examples is *equivalent* to minimizing $\ell_M(\boldsymbol{\theta})$ for the *same* classifier. The proof of the Theorem, elementary, is interesting in itself as it simplifies the long derivation for the equivalence between rado and example losses in [20].

Theorem 3 *Let $\Sigma_m \doteq \{-1, 1\}^m$. Then, for any \mathcal{S} , any Γ and any $\boldsymbol{\theta}$, $\ell_{\text{sql}}(\mathcal{S}, \boldsymbol{\theta}; \Gamma) = 1 + (4/m) \cdot \ell_M(\mathcal{R}_{\mathcal{S}, \Sigma_m}, \boldsymbol{\theta}; \Gamma)$ with*

$$\ell_M(\mathcal{R}_{\mathcal{S}, \Sigma_m}, \boldsymbol{\theta}; \Gamma) = \ell_M(\mathcal{R}_{\mathcal{S}, \Sigma_m}, \boldsymbol{\theta}) + \frac{m}{4} \boldsymbol{\theta}^\top \Gamma \boldsymbol{\theta} . \quad (7)$$

Proof First, we remark that $\mathbb{E}_{\Sigma_m}[\boldsymbol{\theta}^\top \boldsymbol{\pi}_\sigma] = \boldsymbol{\theta}^\top \mathbb{E}_{\Sigma_m}[\boldsymbol{\pi}_\sigma] = (1/2) \cdot \boldsymbol{\theta}^\top \boldsymbol{\pi}_y$, since each example participates to half of the 2^m rados. Letting $\tilde{v} \doteq 2^{m+2} \cdot \mathbb{V}_{\Sigma_m}[\boldsymbol{\theta}^\top \boldsymbol{\pi}_\sigma]$, we also have

$$\begin{aligned}
\tilde{v} &= 4 \cdot \sum_{\sigma \in \Sigma_m} \left(\boldsymbol{\theta}^\top \boldsymbol{\pi}_\sigma - \frac{1}{2} \cdot \boldsymbol{\theta}^\top \boldsymbol{\pi}_y \right)^2 \\
&= \sum_{\sigma \in \Sigma_m} \left(\sum_i \sigma_i \boldsymbol{\theta}^\top \mathbf{x}_i \right)^2 \\
&= \sum_{\sigma \in \Sigma_m} \left[\sum_{i=1}^m (\boldsymbol{\theta}^\top \mathbf{x}_i)^2 + \sum_{i=1}^m \sum_{i' \neq i} \sigma_i \sigma_{i'} \boldsymbol{\theta}^\top \mathbf{x}_i \boldsymbol{\theta}^\top \mathbf{x}_{i'} \right] \\
&= 2^m \cdot \sum_{i=1}^m (\boldsymbol{\theta}^\top \mathbf{x}_i)^2 + \sum_{i=1}^m \sum_{i' \neq i} v_{ii'} \cdot \boldsymbol{\theta}^\top \mathbf{x}_i \boldsymbol{\theta}^\top \mathbf{x}_{i'} , \tag{8}
\end{aligned}$$

with $v_{ii'} \doteq \sum_{\sigma \in \Sigma_m} \sigma_i \sigma_{i'}$. Now, for any $i \neq i'$, $\sigma_i \sigma_{i'}$ takes exactly the same number of times value $+1$ and value -1 , and so $v_{ii'} = 0, \forall i \neq i'$. We get from eq. (8) $\mathbb{V}_{\Sigma_m}[\boldsymbol{\theta}^\top \boldsymbol{\pi}_\sigma] = (1/4) \cdot \sum_{i=1}^m (\boldsymbol{\theta}^\top \mathbf{x}_i)^2 = (1/4) \cdot \sum_{i=1}^m (y_i \boldsymbol{\theta}^\top \mathbf{x}_i)^2$. Finally,

$$\begin{aligned}
&1 + \frac{4}{m} \cdot \ell_M(\mathcal{S}, \Sigma_m, \boldsymbol{\theta}) \\
&= 1 - \frac{2}{m} \cdot \sum_{i=1}^m y_i \boldsymbol{\theta}^\top \mathbf{x}_i + \frac{1}{m} \cdot \sum_{i=1}^m (y_i \boldsymbol{\theta}^\top \mathbf{x}_i)^2 \\
&= \frac{1}{m} \cdot \sum_i (1 - y_i \boldsymbol{\theta}^\top \mathbf{x}_i)_2^2 , \tag{9}
\end{aligned}$$

and we get Theorem 3 by integrating Ridge regularization. ■

Hence, minimizing the Ridge regularized square loss over examples is equivalent to minimizing a regularized version of the M-loss, over the complete set of all rados. This set has exponential size. The usual trick would be to randomly subsample this huge set, along with proving good uniform convergence bounds for the M-loss — this can be done in the same way as for the logistic loss [21]. However, in the case of the square loss, greed pays twice: learning from all rados in \mathcal{R}_*^u may be both cheap (computationally) and accurate.

Computation and optimality of \mathcal{R}_*^u In our distributed context, we do not have access to all rados because we do not assume that we have access to an entity matching function. Yet, we are going to show a first result which is, in a sense, *stronger*: in very general settings, there exists $\mathbf{u} \in \mathbb{R}^{m*}$ such that \mathcal{R}_*^u , *systematically* or in expectation, belongs to $\mathcal{R}_{\mathcal{S}, \Sigma_m}$. This set, \mathcal{R}_*^u of potentially exponential size, therefore gives us a set of rados that would have been built *from \mathcal{S} , had we known the perfect solution to entity matching*. So, even without carrying out entity matching, we have access to a potentially huge set of "ideal" rados which we can use to learn $\boldsymbol{\theta}$ via the minimization of $\ell_M(\cdot, \boldsymbol{\theta}; \Gamma)$. Furthermore, there exists a simple algorithm to build \mathcal{R}_*^u . The communication protocol, in Figure 3 for $p = 3$ peers, summarizes what happens when peers have received message "PART(s, y)"

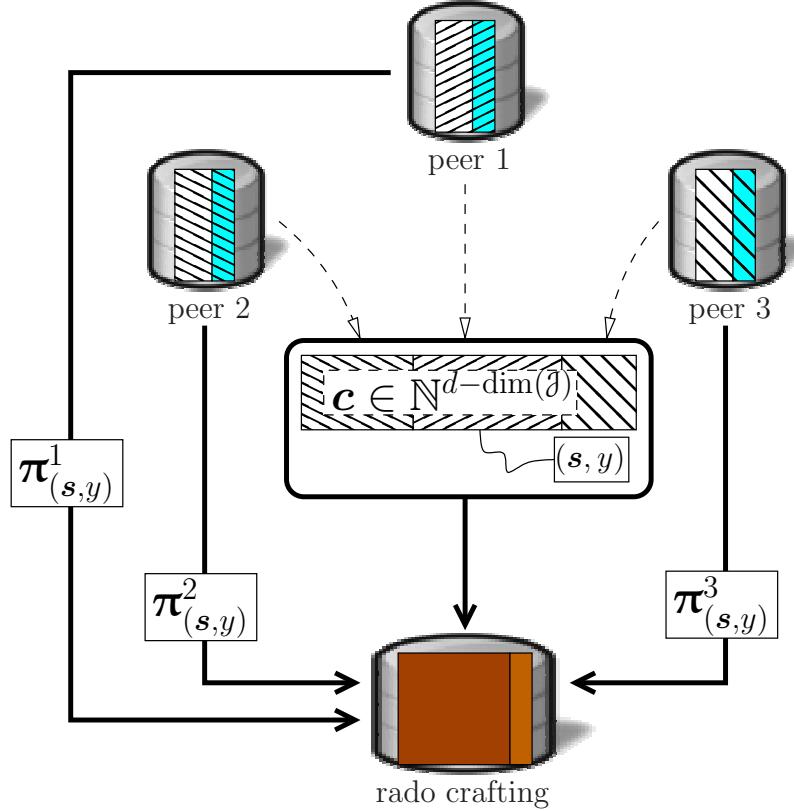


Figure 3: Communication for one BB rado, with $(s, y) \in \mathcal{J}_*$. Counter c is defined in Algorithm RADO CRAFT (see text).

from a "rado crafting" peer implementing Algorithm 1 below (" \rightsquigarrow " symbolizes message sending). Specifically, P_j does the following:

- it computes and return $\pi_{(s,y)}^j$; let C_j be the number of examples that are counted in the sum in eq. (1);
- it updates counter vector c : for each feature $k \notin \mathcal{J}$ it possesses in its database, it does $c_k \leftarrow c_k + C_j$;

Remark that the updates of c can easily be done in parallel, as well as the computation of each $\pi_{(s,y)}^j$ for each peer. Letting $v_i \doteq (s, y) \in \mathcal{J}_*$, the corresponding value of u_i is given by:

$$u_i = \tilde{u}_i \doteq (\mathbf{1}^\top c)(|d| - \dim(\mathcal{J}))^{-1}, \quad (10)$$

which is guaranteed to be non-zero since $v_i \in \mathcal{J}_*$. We now show one of the main results of this paper.

Theorem 4 *In setting (VP), for any $p \geq 2$, any \mathcal{S} , any \mathcal{J} , the following holds on the output of Algorithm 1: $\mathcal{R}_*^{\tilde{u}} \subseteq \mathcal{R}_{\mathcal{S}, \Sigma_m}$.*

Proof (sketch) Let $w \in \mathbb{R}^{m*}$ be such that w_i is the number of examples in \mathcal{S} that match v_i . The proof follows two steps, (i) $\mathcal{R}_B^{\tilde{u}} = \mathcal{R}_B^w$ and (ii) $\mathcal{R}_*^w \subseteq \mathcal{R}_{\mathcal{S}, \Sigma_m}$. Then, (i) is immediate; for (ii),

Algorithm 1 RADOCAFT(P_1, P_2, \dots, P_p)

Input Peers P_1, P_2, \dots, P_p ;
 Step 1: Let $\mathcal{R}_B^{\tilde{u}} \leftarrow \emptyset$;
 Step 2: **for** $s \in \mathcal{J}$, $y \in \{-1, +1\}$
 2.1: Let $\boldsymbol{\pi} \leftarrow \mathbf{0} \in \mathbb{R}^d$, $\mathbf{c} \leftarrow \mathbf{0} \in \mathbb{N}^{d-\dim(\mathcal{J})}$;
 2.2: **for** $j \in [p]$
 2.2.1: $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi} + \text{lift}_{\mathcal{X}}(\text{PART}(s, y) \rightsquigarrow P_j)$;
 2.3: Let $\tilde{u} \leftarrow (\mathbf{1}^\top \mathbf{c}) (d - \dim(\mathcal{J}))^{-1}$;
 2.4: $\mathcal{R}_B^{\tilde{u}} \leftarrow \mathcal{R}_B^{\tilde{u}} \cup (\tilde{u} \cdot \text{lift}_{\mathcal{X}}(y \cdot s) + \boldsymbol{\pi})$;
Return $\mathcal{R}_B^{\tilde{u}}$;

the proof follows once three simple facts are established in the **(VP)** setting: (a) the true entity matching exists, (b) any w_i -BB rado for pair $v_i \doteq (s, y)$ would be obtained as a rado summing the contribution of all examples in \mathcal{S} matching the corresponding signature s and class y , (c) we obtain $\mathcal{R}_B^{\tilde{u}} \subseteq \mathcal{R}_{\mathcal{S}, \Sigma_m}$, from which follows the Theorem's statement with eq. (3) and the fact that any sum of a subset of rados in $\mathcal{R}_B^{\tilde{u}}$ would also be in $\mathcal{R}_{\mathcal{S}, \Sigma_m}$ since an example cannot match two distinct couples (signature, class). \blacksquare

Hence, in the **(VP)** setting, Algorithm 1 always provides the basis for a (possibly exponential-sized) set $\mathcal{R}_*^{\tilde{u}}$ of these "ideal" rados. Let us now generalize Theorem 4 to **(G)**, which is obviously more difficult to tackle since (i) there may be a huge amount of missing data (\perp in Figure 1) and (ii) there would be no one-one correspondence between the peers' examples in general. Yet, there is a interesting property which can be shown in the following **(R)**andomized model: each peer's features remain fixed (and all peer's features comply with **(G)**), but there exists a fixed $\boldsymbol{\eta} \in [0, 1]^m$ such that example i has probability η_i to be seen by a peer. Let $\bar{\mathcal{S}}$ denote the "expected" sample, where each example is weighted by its probability. For any signature s and class y , $\mathbb{E}[\boldsymbol{\pi}_{(s, y)}]$ denotes the expected rado put in $\mathcal{R}_B^{\tilde{u}}$ in step 2.4 of Algorithm 1.

Theorem 5 *Under **(R)**, $\forall (s, y) \in \mathcal{J}_+$, $\mathbb{E}[\boldsymbol{\pi}_{(s, y)}] \in \mathcal{R}_{\bar{\mathcal{S}}, \Sigma_m}$.*

(Proof omitted) Hence, under setting **(G)**, if examples are "seen" independently at random by peers, the expected output of Algorithm 1 still meets the guarantees of Theorem 4 with respect to the expected sample. The fact that $\mathcal{R}_B^{\tilde{u}} \subseteq \mathcal{R}_{\mathcal{S}, \Sigma_m}$ from Theorem 4 is also a consequence of Theorem 5 for $\boldsymbol{\eta} = \mathbf{1}$. Finally, there is one way to relax further assumption **(G)**, which is to let each feature not shared by all peers to be shared by any subset of peers. In this case, it is possible to modify RADOCAFT so that it *still* builds rados that would meet Theorem 5. This mainly requires a careful adjustment of each counter \mathbf{c} .

Learning from all rados of $\mathcal{R}_*^{\tilde{u}}$ The questions that remain are how we minimize the regularized M-loss and, more importantly, what subset of rados from $\mathcal{R}_*^{\tilde{u}}$ we shall use. As already discussed, we choose "greediness" against randomization [21]: instead of picking a (small) random subset of $\mathcal{R}_*^{\tilde{u}}$, we want to use them *all* because we know that all of them are "ideal" or close to being so via Theorems 4, 5. Recall that $|\mathcal{R}_*^{\tilde{u}}|$ may be of exponential size (in $m, d, |\mathcal{J}_*|$, etc.). We now show that if we consider all of $\mathcal{R}_*^{\tilde{u}}$, the optimal $\boldsymbol{\theta}_{\text{rad}}^*$ of $\ell_M(\mathcal{R}_*^{\tilde{u}}, \boldsymbol{\theta}; \Gamma)$ has an analytic expression which

Algorithm 2 DRL($P_1, P_2, \dots, P_p; \Gamma$)

Input Peers P_1, P_2, \dots, P_p , SPD matrix $\Gamma, \gamma > 0$;
 Step 1: $B \leftarrow \text{Column}(\text{RADO-CRAFT}(P_1, P_2, \dots, P_p))$;
 Step 2: $\theta \leftarrow (BB^\top + \gamma \cdot \Gamma)^{-1} B1$;
Return θ ;

depends *only* on the rados of $\mathcal{R}_B^{\tilde{u}}$. In short, it is even *faster* to compute than θ_{ex}^* from \mathcal{S} in eq. (5), and can be directly computed from the output of Algorithm 1.

Theorem 6 Let $\theta_{\text{rad}}^* \doteq \arg \min_{\theta} \ell_M(\mathcal{R}_*^{\tilde{u}}, \theta; \Gamma)$ (eq. (7)). Then

$$\theta_{\text{rad}}^* = (BB^\top + \dim_c(B) \cdot \Gamma)^{-1} B1, \quad (11)$$

where B stacks in columns the rados of $\mathcal{R}_B^{\tilde{u}}$, and $\dim_c(B)$ is the number of columns of B .

Proof The proof uses the following trick: consider any sample \mathcal{S}' such that its edge vectors match the basic block rados. Remark that $XX^\top = \sum_i (y_i \mathbf{x}_i)(y_i \mathbf{x}_i)^\top$ in eq. (5) depends only on edge vectors, and so, since $\pi_y = B1$, the optimal square loss classifier on \mathcal{S}' is θ_{rad}^* in eq. (11), which, through Theorem 3, is also the optimal classifier on $\ell_M(\mathcal{R}_*^{\tilde{u}}, \theta; \Gamma)$. ■

When $m_* = m$, each element of $\mathcal{R}_B^{\tilde{u}}$ is in fact an example, and we retrieve eq. (5). One consequence of Theorem 6 is the following convergence property which we sketch: in the **(VP)** setting, for any $\varepsilon \geq 0$, there exists a minimal size for \mathcal{J}_* such that θ_{rad}^* will be ε -close to θ_{ex}^* , where the closeness can be measured by $\|\theta_{\text{rad}}^* - \theta_{\text{ex}}^*\|_2$ or $|\cos(\theta_{\text{rad}}^*, \theta_{\text{ex}}^*)|$. The statement of DRL (Distributed Rado-Learn) is given in Algorithm 2. In Step 1, "column(.)" takes a set of vectors and put them in column in a matrix.

4 Experiments

Algorithms We have evaluated the leverage that DRL provides compared to the peers, that would learn using only their local dataset. Each peer P_j estimates learns through a ten-folds stratified cross-validation (CV) minimization of $\ell_{\text{sql}}(\mathcal{S}^j, \theta; \gamma \cdot \text{Id}_{d_j})$ (see eq. (5)), where γ is also locally optimized through a ten-folds CV in set $\mathcal{G} \doteq \{.01, 1.0, 100.0\}$. DRL minimizes $\ell_M(\mathcal{R}_*^{\tilde{u}}, \theta; \Gamma)$ (solution in eq. (11)) where $\mathcal{R}_B^{\tilde{u}}$ is built using RADO-CRAFT, with the set of all peers as input.

We have carried out a very simple optimisation of the regularisation matrix of DRL as a diagonal matrix which weights differently the shared features, $\Gamma \doteq \text{Diag}(\text{lift}_x(\text{proj}_j(\mathbf{1}))) + \gamma \cdot \text{Diag}(\text{lift}_x(\text{proj}_{x \setminus j}(\mathbf{1})))$, for $\gamma \in \mathcal{G}$. γ is optimized by a 10-folds CV on \mathcal{J}_* . CV is performed on *rados* as follows: first, $\mathcal{R}_B^{\tilde{u}}$ is split in 10 folds, $\mathcal{R}_{B,\ell}^{\tilde{u}}$, for $\ell = 1, 2, \dots, 10$. Then, we repeat for $\ell = 1, 2, \dots, 10$ (and then average) the following CV routine:

1. DRL is trained using $\mathcal{R}_B^{\tilde{u}} \setminus \mathcal{R}_{B,\ell}^{\tilde{u}}$;
2. DRL's solution, θ_{rad}^* , is evaluated on "test rados" by computing $\ell_M(\mathcal{R}_{B,\ell}^{\tilde{u}}, \theta_{\text{rad}}^*; \Gamma)$.

The expression of Γ for rados exploits the idea that the estimations related to a shared feature can be much more accurate than for another, non shared feature.

Domain	m	d	$\min_j \hat{p}_{\text{err}}(\mathcal{P}_j)$	p	$\dim(\mathcal{J})$	Results
Wine	178	12	0.07	{2, 3, ..., 8}	{1, 2, 3, 4}	Table 12
Sonar	208	60	0.29	{2, 3, ..., 16}	{1, 2, ..., 20}	Table 6
Ionosphere	351	33	0.20	{2, 3, ..., 9}	{1, 2, ..., 9}	Table 8
Mice	1 080	77	0.30	{2, 3, ..., 20}	{1, 2, ..., 20}	Table 4
Winered	1 599	11	0.26	{2, 3, ..., 7}	{1, 2, 3, 4}	Table 9
Steelplates	1 941	33	0.16	{2, 3, ..., 14}	{1, 2, ..., 5}	Table 14
Statlog	4 435	36	0.05	{2, 3, ..., 30}	{1, 2, ..., 5}	Table 13
Winewhite	4 898	11	0.32	{2, 3, ..., 7}	{1, 2, 3, 4}	Table 10
Page	5 473	10	0.21	{2, 3, ..., 6}	{1, 2, 3, 4}	Table 3
Firmteacher	10 800	16	0.26	{2, 3, ..., 7}	{1, 2, ..., 7}	Table 7
Phishing	11 055	30	0.11	{2, 3, 4, 5}	{1, 2, 3, 4}	Table 11

Table 2: UCI domains used in our experiments [1], with for each the indication of the total number of features (d), examples (m) and the error of the optimal peer in hindsight obtained in our experiments, $\min_j \hat{p}_{\text{err}}(\mathcal{P}_j)$. Two of the right columns present, for each domain, the range of values for the number of peers (p) and the number of shared features ($\dim(\mathcal{J})$) considered. Experiments were performed considering *all* possible combinations of values of p and $\dim(\mathcal{J})$ within the allocated sets. The rightmost column points to the Table collecting specific results for each domain.

Domain generation we have used a dozen UCI domains [1], presented in Table 2. For each domain, we have varied (i) the number of peers p , (ii) the number of shared features $\dim(\mathcal{J})$, and (iii) the number b of numeric modalities ("bins") each shared feature was reduced to (controls the size of \mathcal{J}_*). The training sample is split among peers, each keeping record of \mathcal{J} and its own features (non shared features are evenly partitioned among peers). Finally, for some $p_s \in [0, 1]$, each peer \mathcal{P}_j selects a proportion p_s of its examples index and for each of them, another peer $\mathcal{P}_{j'}$, chosen at random, gets the example as well (on its own set of features $\mathcal{X}^{j'}$). When $p_s = 0$, this is setting **(VP)**. We then run *all* algorithms for *each* value p , $\dim(\mathcal{J})$, b , p_s . As we shall see, b appears to have a relatively small influence compared to the other factors, so we mainly report results combining various values for p , $\dim(\mathcal{J})$ and p_s , for the range of values of p , $\dim(\mathcal{J})$ specified in Table 2, and for $p_s \in \{0.0, 0.2\}$. We have chosen $b = 4$ for all domains, except when it is not possible (if for example all features are boolean), in which case we pick $b = 2$. Table 2 also provides the smallest test error obtained for a peer among all runs for each domain: this is an indication of the room of improvement for DRL, and it also shows that in general, at least some (and in fact most) peers were always very significantly better than random guessing, a safe-check that DRL is not just beating unbiased coins.

Metric We used two metrics. The first,

$$\Delta \doteq \hat{p}_{\text{err}}(\text{DRL}) - \min_j \hat{p}_{\text{err}}(\mathcal{P}_j) \ (\in [-1, 1]) , \quad (12)$$

is the test error for DRL minus that of the *optimal* peer *in hindsight* (since we consider the peer's test error). when $\Delta < 0$, DRL beats *all* peers. For example, Table 3 (left) provides the results

obtained on UCI domain page. We see that for almost all combinations of p and $\dim(\mathcal{J})$, DRL beats all peers.

To evaluate the statistical significance, we compute

$$q \doteq \text{proportion of peers } \textit{statistically beaten by DRL} \ (\in [0, 1]) . \quad (13)$$

To compute the test, we use the powerful Benjamini-Hochberg procedure on top of paired t -tests with $q^* = p\text{-val} = 0.05$, [3]; $q = 0.8$ surface helps see when DRL *statistically beats all peers*. For example, Table 3 (right) displays that DRL does not always *statistically* beat all peers when $\Delta < 0$, yet it manages to statistically beat all of them in approximately one third to one half of the total tests, which implies that, on this domain, there is a significant chance that DRL improves on the peers, regardless of their number and the number of shared features.

Results Due to the large number of domains considered, results are split among different tables, one for each domain in general, in Tables 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14 (also referenced in Table 2). All domains display that there exists regimes $(p, \dim(\mathcal{J}))$ for which DRL improves on all peers, in some cases significantly. Sometimes, the improvement is sparse (phishing), but sometimes it is quite spectacular and in fact (almost) systematic (page, ionosphere, steelplates). domain steelplate’s case is interesting, since the so-called *Oracle*, *i.e.* the learner that leans from the complete training fold *before* it is split among peers — and therefore knows the solution to entity matching —, has for this domain almost optimal error, but local peers are in fact very far from this optimum. This indicates that many features, properly combined, are necessary to attain the best performances. DRL’s performances are close to the Oracle, which accounts for the huge gap in classification compared to peers — sometimes, DRL’s test error is smaller than that of the *best* peer by more than 20% —, and so it seems that DRL indeed successfully bypasses entity matching to learn a classifier that almost matches the Oracle’s performances, and therefore represents a very significant leverage of each peer’s data.

To drill down into more specific results, Table 15 (left) displays that binning indeed does not affect significantly DRL on average, which is also good news, since it means that there is no restriction on the shared features for DRL to perform well: shared features can be binary, or categorical with any number of modalities. Table 16 displays that while the CV tuning of Γ offers leverage to DRL (*vs* $\Gamma = \text{Id}_d$) in general (firmteacher), there are some (rare) domains (mice) on which relying on the simplest $\Gamma = \text{Id}_d$ improves upon the results of CV. This, we believe, comes from the fact that CV as we have carried out is certainly not optimal because one rado can aggregate any number of examples. Last, Table 15 (right) drills down a bit more into the performances of DRL with respect to those of the Oracle on a domain for which DRL obtains somehow “median” performances among all domains, sonar. The Oracle (10-folds CV from the *total* ER’ed \mathcal{S}) is *idealistic* since in general we do not know the solution to ER, yet it gives clues on how close DRL may be from the “graal”. Interestingly, DRL comes frequently under the statistical significance radar ($\alpha = 0.05$). In notable cases (more frequent as p_s increases), DRL beats Oracle — but not significantly. Aside from theory, these are good news as DRL does not assume ER’ed data, and uses an amount of data which can be $\sim p^2$ times *smaller* than Oracle.

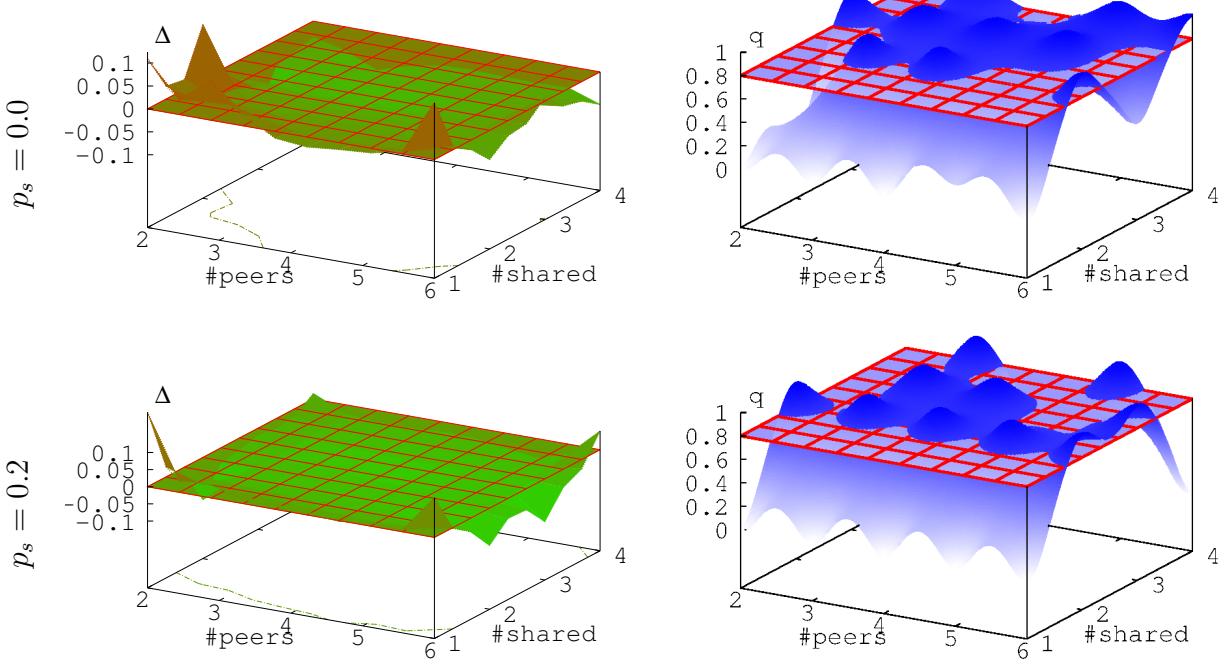


Table 3: Results on domain page: plots of $\Delta \doteq \hat{p}_{\text{err}}(\text{DRL}) - \min_j \hat{p}_{\text{err}}(P_j)$ (left) and $q = \text{prop. peers simultaneously beaten by DRL}$ (right) as a function of the number of peers p and the number of shared features $\dim(\mathcal{J})$. Top: proportion of shared examples $p_s = 0.0$ (setting **(VP)**); bottom: proportion of shared examples $p_s = 0.2$. The isoline on the left plots is $\Delta = 0$.

5 Discussion and related work

We remark that our framework is not formally comparable with ER, since the two address different problems. On one hand, ER has a much broader applicability than the problem object of this paper; learning on distributed datasets is less general than ER: in fact, we show a solution that bypasses ER. On the other hand, *learning-based* ER [5] as well as manifold alignment techniques [15] are viable only knowing some ground truth matches — which are not required for working with rados. From another perspective, in concert with the *open issues* in [13], we study ER as component of a pipeline for classification, and highlight how matching is not necessary for the purpose of learning.

In spite of those considerations, we can still draw comparisons with methods that learn on top of data merged through ER (Table 5). In both settings, no ID is shared between datasets but some attributes must be so, in order to allow entities comparison for matching or for building rados. Obviously, entity matching does not require the labels to be one of those shared attributes, while this is a fundamental hypothesis of our approach. Although, it is not as restrictive as may be expected at first: if just one peer has labels, then *all* can obtain labels on their own data, via *learning from label proportions* [23, 24]: the label handling peer computes the label proportions per each block; the “bags” are defined by examples matching a particular signature. Proportions are then shared among all other peers, which can train a classifier with them so as to obtain approximate labels for each observation.

To discuss time complexity, let us consider a simplified problem with only 2 peers with m

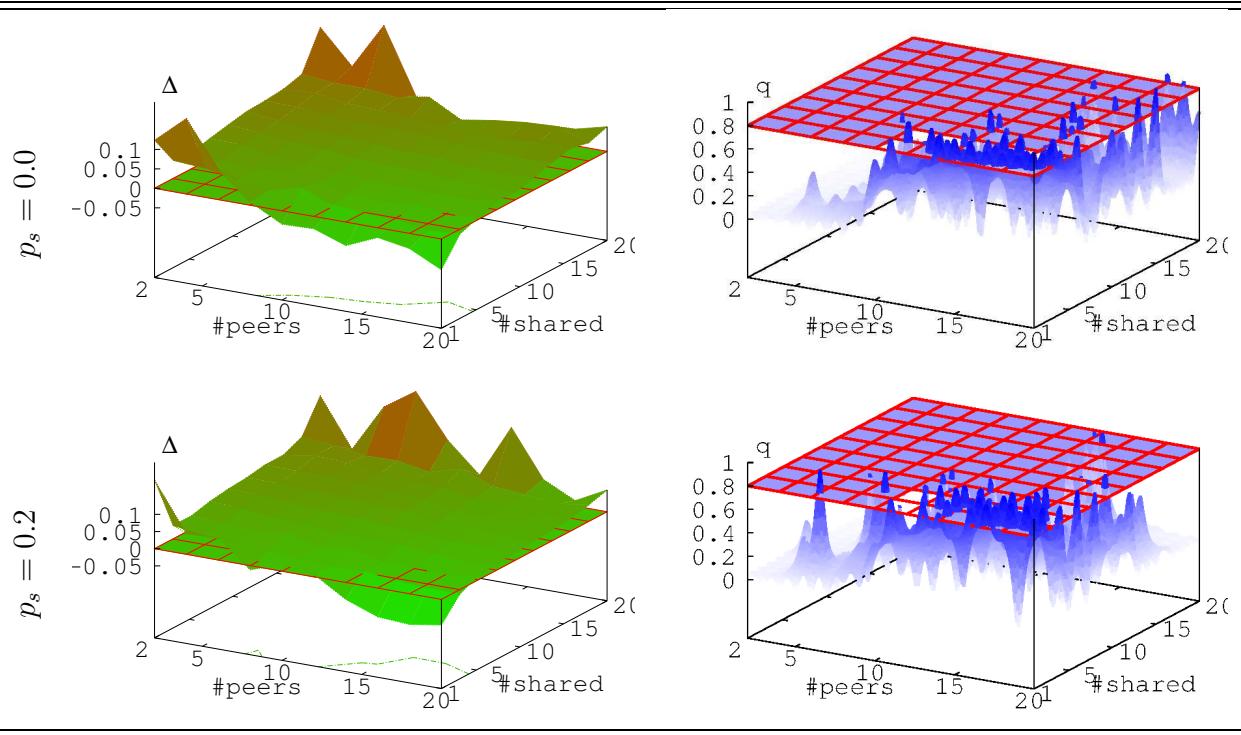


Table 4: Results on domain mice, using the same convention as Table 3.

examples each in the **(VP)** scenario. In terms of complexity of fusion, if we assume that examples are uniformly distributed in the blocks, each block has size m/m^* . DRL builds each block rado in time $O(m/H)$, with total cost linear in m . ER takes $O(m^2/H^2 \cdot T_{sim})$ to match entities in each of the H blocks, where T_{sim} is the cost of evaluation a similarity function; learning-based methods spend additional time for training; advanced blocking strategies can reduce the average complexity [4, 32, 31].

Most literature on distributed learning is concerned with limiting communication and designing optimal strategies for merging models [2, 17]; beside that, previous works focus on horizontal split by observations, with few exceptions [18]. In contrast, we exploit what is sufficient to merge *about the data*. The communication protocol is extremely simple. Once rados are crafted locally, they are sent to a central learner in one shot. By Theorem 6, only d -dimensional m^* blocks rados are needed. *Data is not accessed anymore* and learning takes place centrally. Moreover, rados help with data compression, being $m^* \times d$, $m^* \ll m$ the problem size. ER needs to transfer and learn from all entities, for a total size of $m \times d$.

Learning on data described by different feature sets is the topic of multiple view learning and co-training [7, 26]. To the best of our knowledge, co-training with unknown matches has not been addressed before. [8] presents a multi-view distributed algorithm with co-regularization; although it requires matches for all unlabelled examples.

In settings with multiple data providers, privacy can be crucial [2]. The agents have to trade off model enhancements and information leaks. A learner receives rados to train the model; this can be done by one of the agents, or by a third party — paralleling multi-party ER scenarios [9]. The only information sent through the channel consists of rados, while examples, with their individual

Metric	ER + Learning	RADO CRAFT + DRL
Assumption: shared IDs	no	no
Assumption: some shared variables	necessary	necessary
Assumption: shared labels	no	may be relaxed
Fusion / Rados crafting	$O(m^2/m^* \cdot T_{sim})$	$O(m)$
Communication	$m \times d$	$m^* \times d, m^* \ll m$
Learning problem	$m \times d$	$m^* \times d, m^* \ll m$
Privacy	complex	many guarantees

Table 5: Multiple metrics of comparison between learning on top of ER and our approach. Time complexity are estimated for 2 peers in the **(VP)** scenario, assuming all blocks of equal size. See Section 5 for details.

sensible features, are never shared. Hardness results on reconstruct-ability of examples have been proven, along with NP-HARD characterizations, and protection in the sense of differential privacy [21]. Furthermore, due to their compressive power, rados represent an alternative to bulk data collection [27]: storing examples becomes superfluous. Regarding ER, since matching has the potential of de-anonymizing the entities, privacy is usually a very relevant issue to address [9]. However, solutions are not straightforward, as proven by the vast amount of research on the topic [30]; techniques based on partial share of attributes, anonymization or hashing can severely impair the process.

Even assuming labelled examples, no (observation, label) pair is actually available for training, and thus the task can be seen as weakly supervised [12, 22]. Although, a set of aggregate quantities, *i.e.* sums of examples over subsets of the total sample (the rados), turns out to be enough for learning. Theorem 3 expresses a form of *sufficiency* of the whole set of rados with regard to the square loss; a similar property is proposed for logistic loss in [21]. One of the 2^m rados the *mean operator*, $\mu_S \doteq (1/m) \cdot \pi_y$, is formally proven a *sufficient statistics* for the class for a wide set of losses [23, 22]. This work, along with the cited predecessors, shows how the interplay between aggregate statistics and losses can lead to effective solutions to difficult learning problems.

6 Conclusion

The key message of our paper is that Entity Matching addresses a very general *but* difficult problem, and in the comparatively restricted context of supervised learning from distributed datasets, accurate learning evading the pitfalls of Entity Matching *is* possible with Rademacher observations. Rados have another advantage: they offer a cheap, easily parallelizable material which somehow “compresses” examples while allowing accurate learning. They also offer readily available solution for guarantees private exchange of data in a distributed setting. Finally, some domains display that there is significant room space for improvement of how cross-validation of optimized parameters are handled. This interesting problem comes in part from the fact that statistical properties of cross-validation on rados are *not* the same as when carried out on examples; this particular aspect will deserve further analysis in the future.

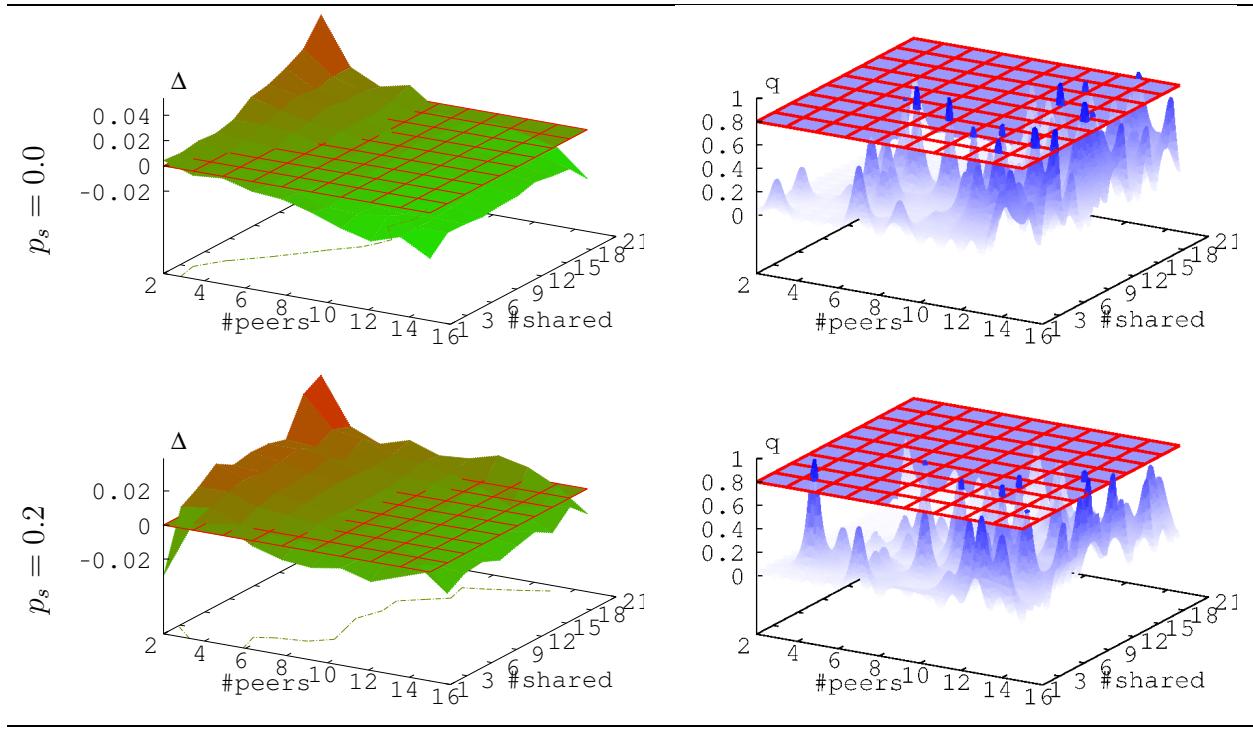


Table 6: Results on domain sonar, using the same convention as Table 3.

7 Acknowledgments

ICT is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Center of Excellence Program.

References

- [1] K. Bache and M. Lichman. UCI machine learning repository, 2013.
- [2] M. F. Balcan, A. Blum, S. Fine, and Y. Mansour. Distributed learning, communication complexity and privacy. *arXiv preprint arXiv:1204.3514*, 2012.
- [3] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. of the Royal Stat. Society. Series B*, 57(1):289–300, 1995.
- [4] M. Bilenko, B. Kamath, and R. J. Mooney. Adaptive blocking: Learning to scale up record linkage. In *6th ICDM*, pages 87–96. IEEE, 2006.
- [5] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proc. of the 9th ACM KDD*, pages 39–48. ACM, 2003.
- [6] J. Bleiholder and F. Naumann. Data fusion. *ACM Computing Surveys (CSUR)*, 41(1):1, 2008.

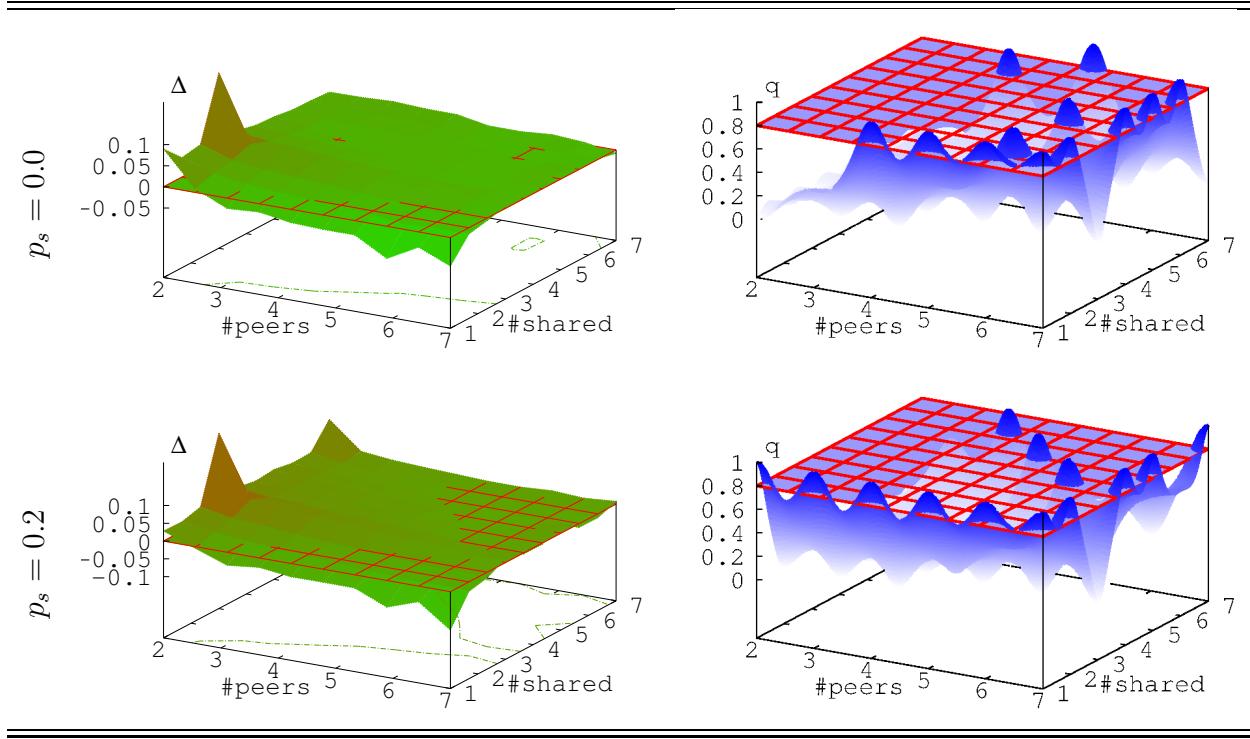


Table 7: Results on domain firmteacher, using the same convention as Table 3.

- [7] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *9th COLT*, pages 92–100, 1998.
- [8] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient co-regularised least squares regression. In *23th ICML*, pages 137–144, 2006.
- [9] P Christen. Privacy-preserving data linkage and geocoding: Current approaches and research directions. In *ICDMW06*, pages 497–501. IEEE, 2006.
- [10] P. Christen. *Data Matching Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer Data-Centric Systems and Applications, 2012.
- [11] T. Estrada, R. Armen, and M. Taufer. Automatic selection of near-native protein-ligand conformations using a hierarchical clustering and volunteer computing. In *ACM BCB*, pages 204–213, 2010.
- [12] D. García-García and R. C. Williamson. Degrees of supervision. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems Workshops (NIPS)*, 2011.
- [13] L. Getoor and A. Machanavajjhala. Entity resolution: theory, practice & open challenges. *Proceedings of the VLDB Endowment*, 5(12):2018–2019, 2012.
- [14] A.-E. Hoerl and R.-W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

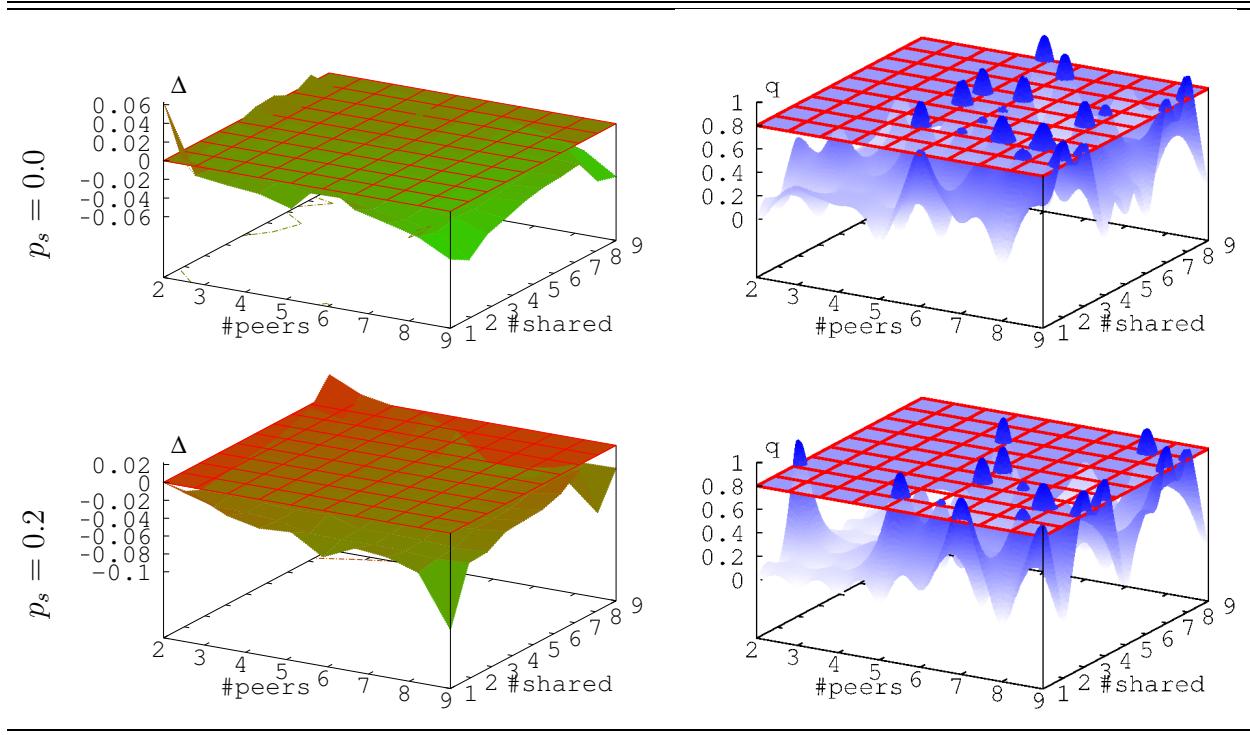


Table 8: Results on domain ionosphere, using the same convention as Table 3.

- [15] S. Lafon, Y. Keller, and R. R. Coifman. Data fusion and multicue data matching by diffusion maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1784–1797, 2006.
- [16] G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.
- [17] Q. Liu and A. T. Ihler. Distributed estimation, information loss and exponential families. In *NIPS*27*, pages 1098–1106, 2014.
- [18] Q. Liu and A.T. Ihler. Distributed parameter estimation via pseudo-likelihood. In *29th ICML*, pages 1487–1494, 2012.
- [19] H. Markowitz. Portfolio selection. *J. of Finance*, 6:77–91, 1952.
- [20] R. Nock. Learning games and Rademacher observations losses. *CoRR*, abs/1512.05244, 2015.
- [21] R. Nock, G. Patrini, and A. Friedman. Rademacher observations, private data, and boosting. *32th ICML*, 2015.
- [22] G. Patrini, F. Nielsen, R. Nock, and M. Carioni. Loss factorization, weakly supervised learning and label noise robustness. *CoRR*, abs/1602.02450, 2016.
- [23] G. Patrini, R. Nock, P. Rivera, and T. Caetano. (Almost) no label no cry. In *NIPS*27*, 2014.

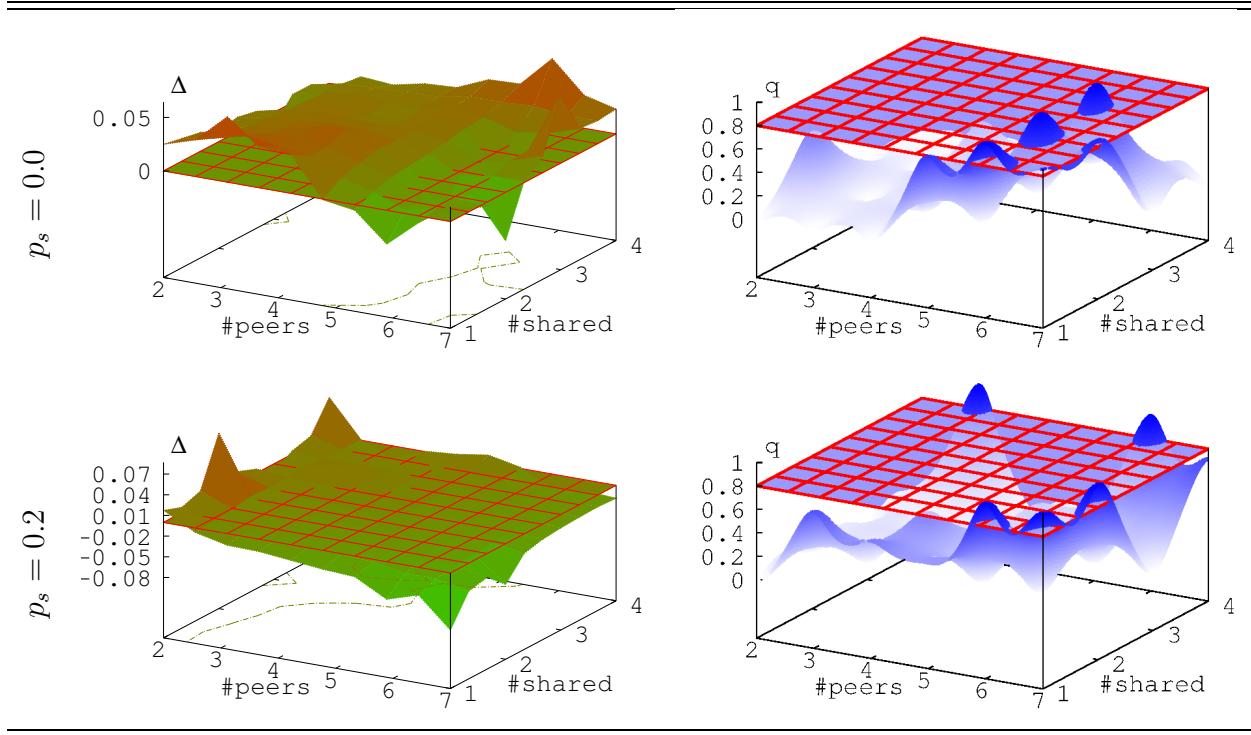


Table 9: Results on domain winered, using the same convention as Table 3.

- [24] N. Quadrianto, A. Smola, T. Caetano, and Q. Le. Estimating labels from label proportions. *JMLR*, 10:2349–2374, 2009.
- [25] V. Rastogi, N.-N. Dalvi, and M.-N. Garofalakis. Large-scale collective entity matching. *Proc. VLDB Endowment*, 4(4):208–218, 2011.
- [26] V. Sindhwani, P. Niyogi, and M. Belkin. A co-regularized approach to semi-supervised learning with multiple views. In *Proceedings of the ICML Workshop on Learning with Multiple Views*, 2005.
- [27] R. F. Sproull, W. H. DuMouchel, M. Kearns, B. W. Lampson, S. Landau, M. E. Leiter, E. R. Parker, and P. J. Weinberger. Bulk collection of signal intelligence: technical options. In *Committee on Responding to Section 5(d) of Presidential Policy Directive 28: The Feasibility of Software to Provide Alternatives to Bulk Signals Intelligence Collection*. National Academy Press, 2015.
- [28] L. Sweeney. Privacy-enhanced linking. *ACM SIGKDD Explorations Newsletter*, 7(2):72–75, 2005.
- [29] F.C. Tsui, J. U. Espino, V. M. Dato, P. H. Gesteland, J. Hutman, and M. M. Wagner. Technical description of rods: a real-time public health surveillance system. *Journal of the American Medical Informatics Association*, 10(5):399–408, 2003.
- [30] D. Vatsalan, P. Christen, and V. S. Verykios. A taxonomy of privacy-preserving record linkage techniques. *Information Systems*, 38(6):946–969, 2013.

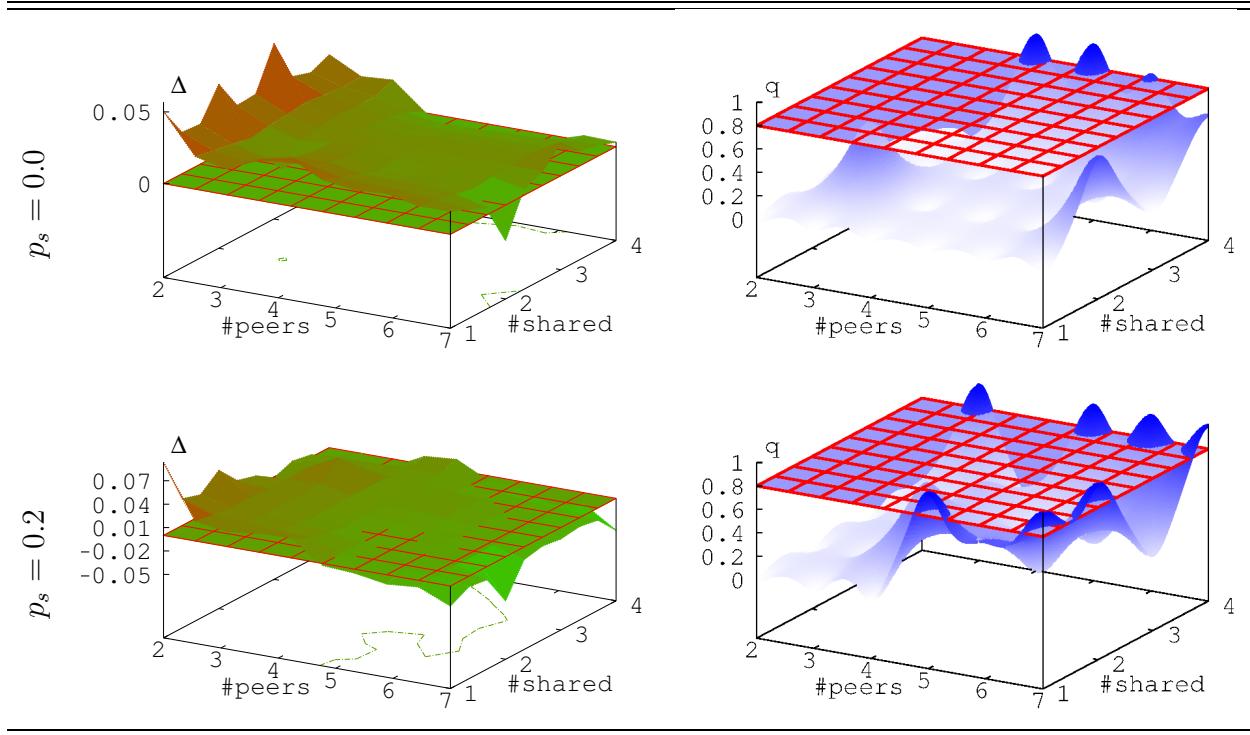


Table 10: Results on domain *winewhite*, using the same convention as Table 3.

- [31] S. E. Whang and H. Garcia-Molina. Joint entity resolution. In *ICDE, 2012 IEEE 28th International Conference on Data Engineering*, pages 294–305. IEEE, 2012.
- [32] S.-E. Whang, D. Menestrina, G. Koutrika, M. Theobald, and H. Garcia-Molina. Entity resolution with iterative blocking. In *Proc. ACM SIGMOD*, pages 219–232, 2009.
- [33] Y. Yamanishi, J.-P. Vert, and K. Kanehisa. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20(suppl 1):i363–i370, 2004.
- [34] B. Zhang, T. Estrada, P. Cicotti, P. Balaji, and M. Taufer. Accurate scoring of drug conformations at the extreme scale. In *15th IEEE/ACM CCGrid*, pages 817–822, 2015.

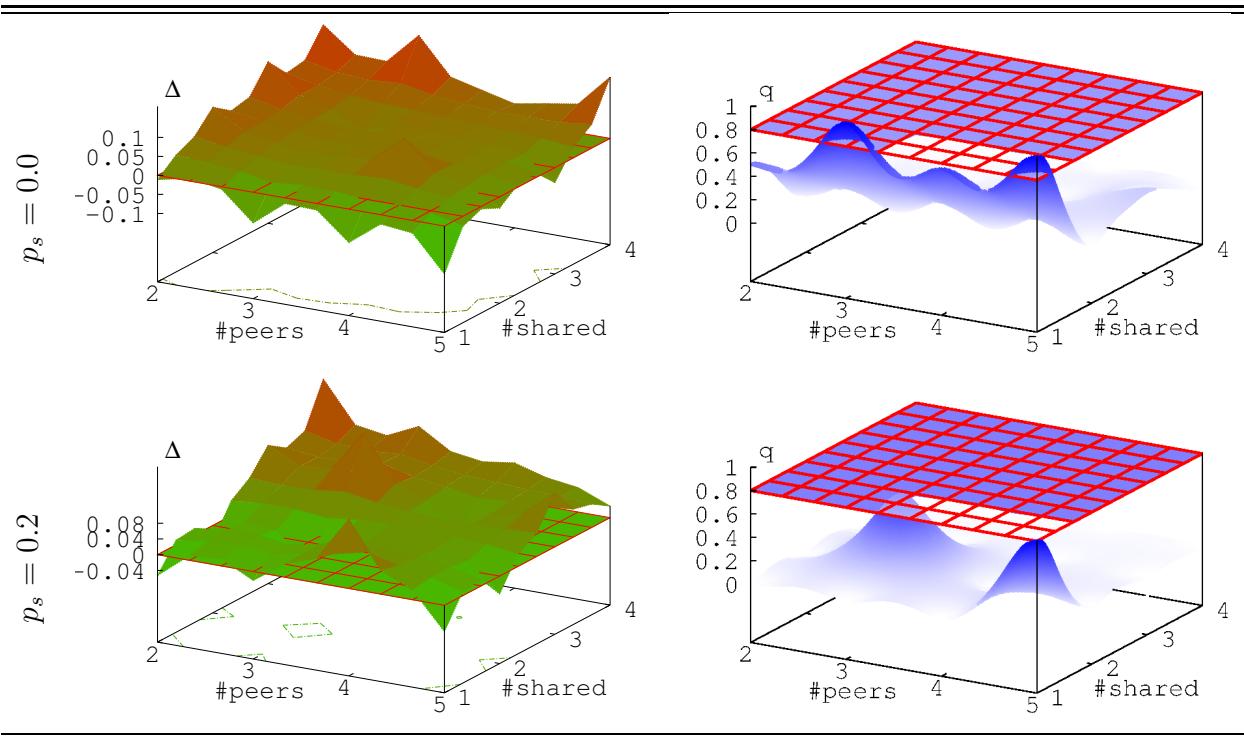


Table 11: Results on domain phishing, using the same convention as Table 3.

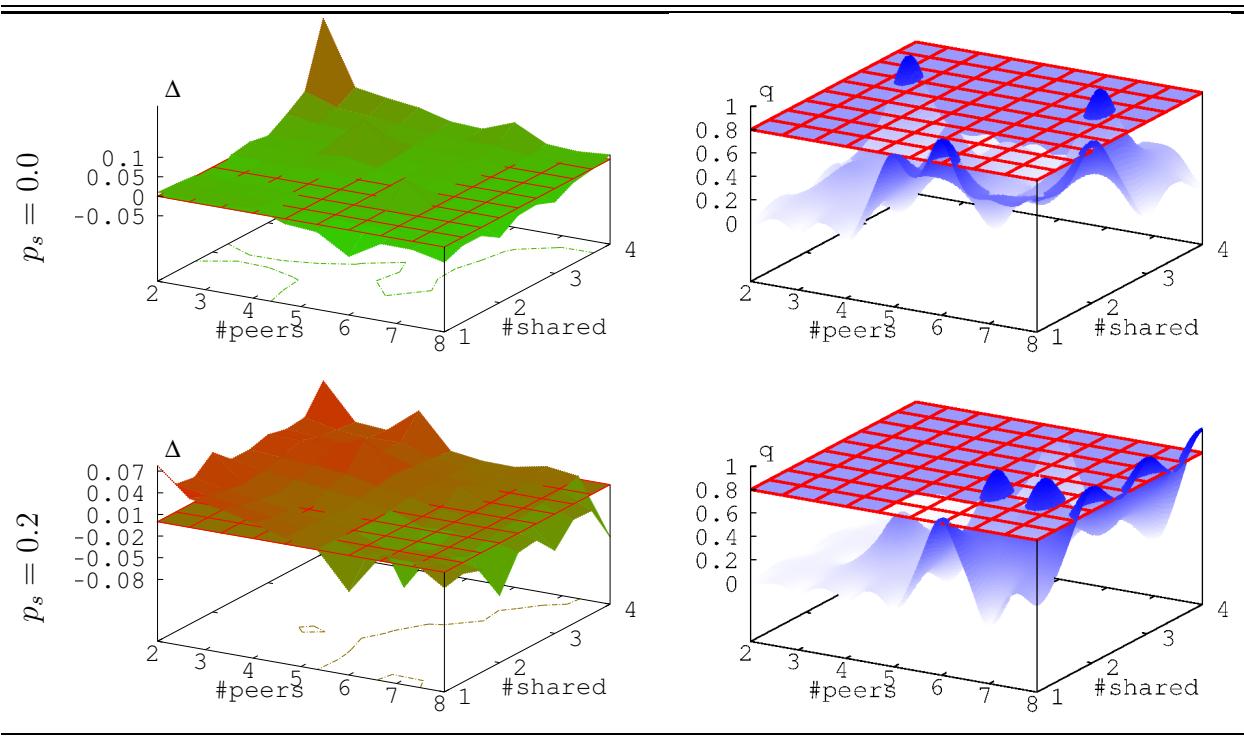


Table 12: Results on domain wine, using the same convention as Table 3.

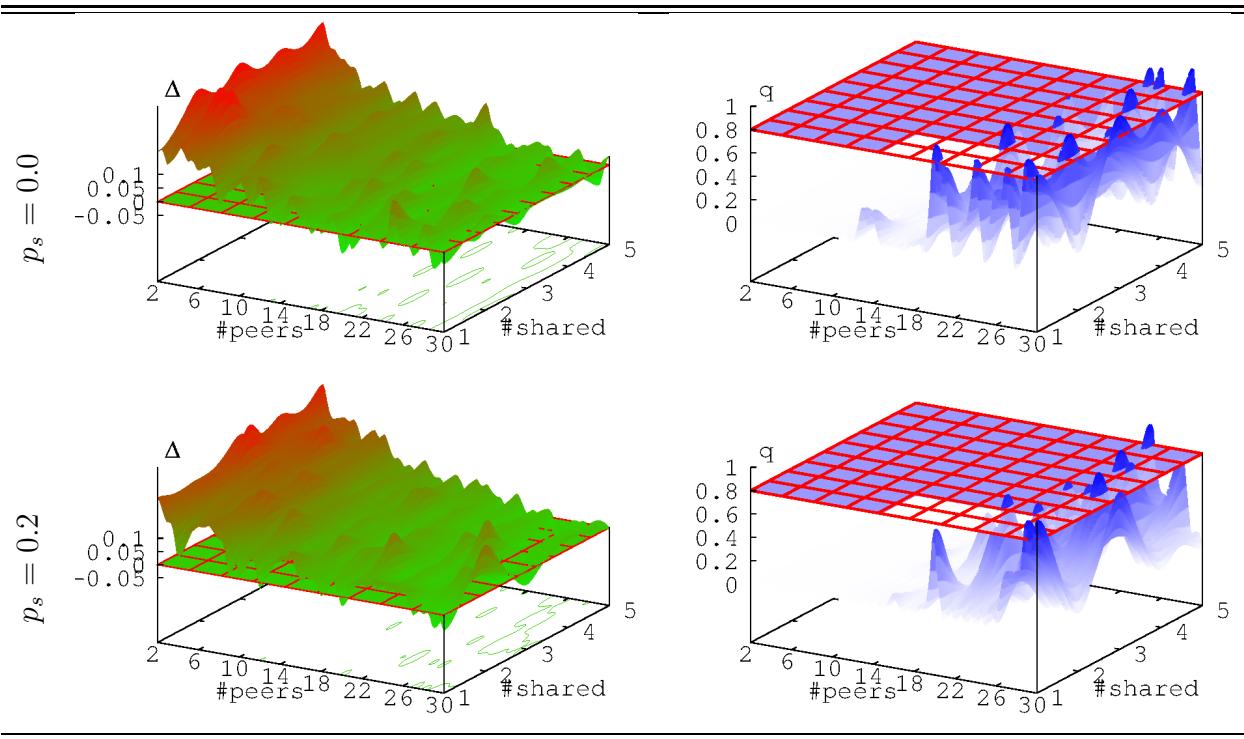


Table 13: Results on domain statlog, using the same convention as Table 3.

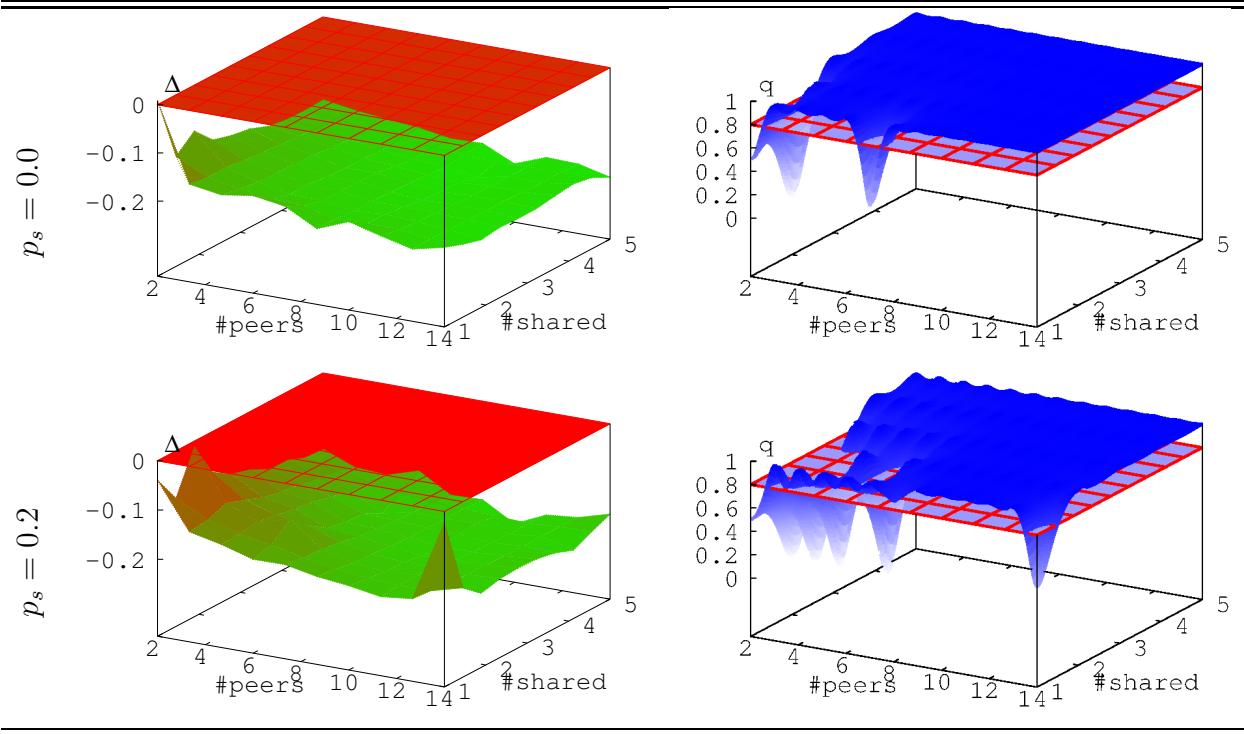


Table 14: Results on domain steelplates, using the same convention as Table 3.

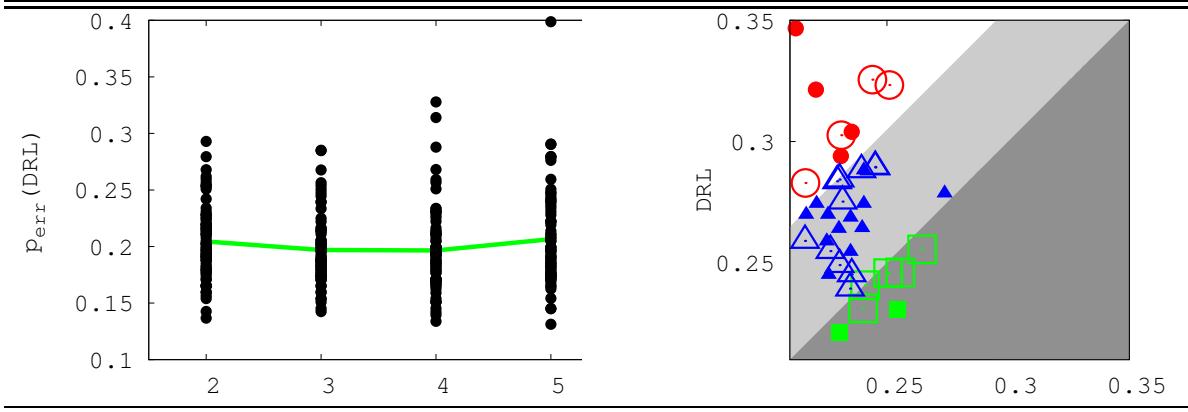


Table 15: *Left*: test error of DRL on domain ionosphere, as a function of the number of bins, aggregating all values of the number of peers p and number of shared features $\dim(\mathcal{J})$ used in Table 8; the green line denotes the average values. *Right*: scatterplot of the test error of DRL (y) vs that of the Oracle (learning using the complete entity-resolved domain). Points in the dark grey area (green) denote better performances of DRL; points in the light grey area (blue) denote better performances of the Oracle (but not statistically better). Points in the white area (red) denote *statistically* better performances of the Oracle (filled points: $p_s = 0.2$; empty points: $p_s = 0.8$).

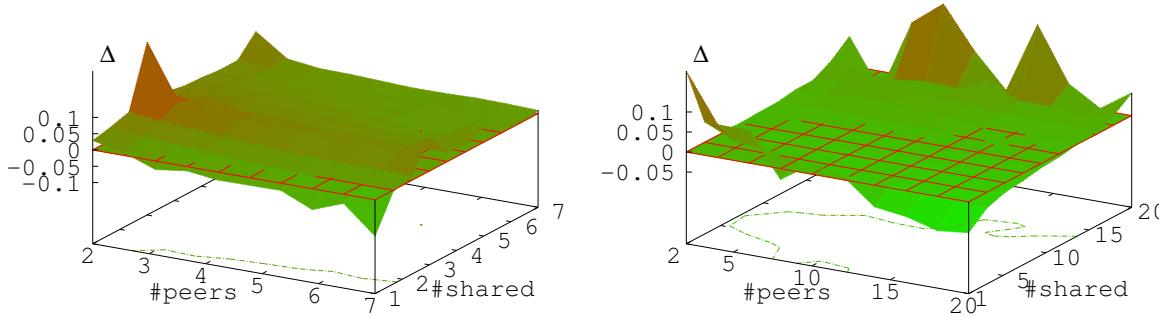


Table 16: Results of the dummy regularized DRL ($\Gamma = \text{Id}_d$) on domains firmteacher (left) and mice (right), following the convention of Table 8 ($p_s = 0.2$).