

---

# Learning with Missing Features

---

**Afshin Rostamizadeh**  
Dept. of Electrical Engineering  
and Computer Science,  
UC Berkeley

**Alekh Agarwal**  
Dept. of Electrical Engineering  
and Computer Science,  
UC Berkeley

**Peter Bartlett**  
Mathematical Sciences, QUT and  
EECS and Statistics,  
UC Berkeley

## Abstract

We introduce new online and batch algorithms that are robust to data with missing features, a situation that arises in many practical applications. In the online setup, we allow for the comparison hypothesis to change as a function of the subset of features that is observed on any given round, extending the standard setting where the comparison hypothesis is fixed throughout. In the batch setup, we present a convex relaxation of a non-convex problem to jointly estimate an imputation function, used to fill in the values of missing features, along with the classification hypothesis. We prove regret bounds in the online setting and Rademacher complexity bounds for the batch i.i.d. setting. The algorithms are tested on several UCI datasets, showing superior performance over baseline imputation methods.

## 1 Introduction

Standard learning algorithms assume that each training example is *fully observed* and doesn't suffer any corruption. However, in many real-life scenarios, training and test data often undergo some form of corruption. We consider settings where all the features might not be observed in every example, allowing for both adversarial and stochastic feature deletion models. Such situations arise, for example, in medical diagnosis—predictions are often desired using only a partial array of medical measurements due to time or cost constraints. Survey data are often incomplete due to partial non-response of participants. Vision tasks routinely need to deal with partially corrupted or occluded images. Data collected through multiple sensors, such as multiple cameras, is often subject to the sudden failure of a subset of the sensors.

In this work, we design and analyze learning algorithms that address these examples of learning with missing features. The first setting we consider is online learning where both examples and missing features are chosen in an arbitrary, possibly adversarial, fashion. We define a novel notion of regret suitable to the setting and provide an algorithm which has a provably bounded regret on the order of  $O(\sqrt{T})$ , where  $T$  is the number of examples. The second scenario is batch learning, where examples and missing features are drawn according to a fixed and unknown distribution. We design a learning algorithm which is guaranteed to globally optimize an intuitive objective function and which also exhibits a generalization error on the order of  $O(\sqrt{d/T})$ , where  $d$  is the data dimension.

Both algorithms are also explored empirically across several publicly available datasets subject to various artificial and natural types of feature corruption. We find very encouraging results, indicating the efficacy of the suggested algorithms and their superior performance over baseline methods.

Learning with missing or corrupted features has a long history in statistics [14, 10], and has received recent attention in machine learning [9, 15, 5, 7]. Imputation methods (see [14, 15, 10]) fill in missing values, generally independent of any learning algorithm, after which standard algorithms can be applied to the data. Better performance might be expected, though, by learning the imputation and prediction functions simultaneously. Previous works [15] address this issue using EM, but can get stuck in local optima and do not have strong theoretical guarantees. Our work also is different from settings where features are missing only at test time [9, 11], settings that give access to noisy versions of all the features [6] or settings where observed features are picked by the algorithm [5].

Section 2 introduces both the general online and batch settings. Sections 3 and 4 detail the algorithms and theoretical results within the online and batch settings resp. Empirical results are presented in Section 5.

## 2 The Setting

In our setting it will be useful to denote a training instance  $\mathbf{x}_t \in \mathbb{R}^d$  and prediction  $y_t$ , as well as a corruption vector  $\mathbf{z}_t \in \{0, 1\}^d$ , where

$$[\mathbf{z}_t]_i = \begin{cases} 0 & \text{if feature } i \text{ is not observed,} \\ 1 & \text{if feature } i \text{ is observed.} \end{cases}$$

We will discuss as specific examples both classification problems where  $y_t \in \{-1, 1\}$  and regression problems where  $y_t \in \mathbb{R}$ . The learning algorithm is given the corruption vector  $\mathbf{z}_t$  as well as the corrupted instance,

$$\mathbf{x}'_t = \mathbf{x}_t \circ \mathbf{z}_t,$$

where  $\circ$  denotes the component-wise product between two vectors. Note that the training algorithm is never given access to  $\mathbf{x}_t$ , however it is given  $\mathbf{z}_t$ , and so has knowledge of exactly which coordinates have been corrupted. The following subsections explain the online and batch settings respectively, as well as the type of hypotheses that are considered in each.

### 2.1 Online learning with missing features

In this setting, at each time-step  $t$  the learning algorithm is presented with an arbitrarily (possibly adversarially) chosen instance  $(\mathbf{x}'_t, \mathbf{z}_t)$  and is expected to predict  $y_t$ . After prediction, the label is then revealed to the learner which then can update its hypothesis.

A natural question to ask is what happens if we simply ignore the distinction between  $\mathbf{x}'_t$  and  $\mathbf{x}_t$  and just run an online learning algorithm on this corrupted data. Indeed, doing so would give a small bound on regret:

$$R(T, \ell) = \sum_{t=1}^T \ell(\langle \mathbf{w}_t, \mathbf{x}'_t \rangle, y_t) - \inf_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^T \ell(\langle \mathbf{w}, \mathbf{x}'_t \rangle, y_t), \quad (1)$$

with respect to a convex loss function  $\ell$  and for any convex compact subset  $\mathcal{W} \subseteq \mathbb{R}^d$ . However, any fixed weight vector  $\mathbf{w}$  in the second term might have a very large loss, making the regret guarantee useless—both the learner and the comparator have a large loss making the difference small. For instance, assume one feature perfectly predicts the label, while another one only predicts the label with 80% accuracy, and  $\ell$  is the quadratic loss. It is easy to see that there is no fixed  $\mathbf{w}$  that will perform well on both examples where the first feature is observed and examples where the first feature is missing but the second one is observed.

To address the above concerns, we consider using a linear *corruption-dependent hypothesis* which is permitted to change as a function of the observed corruption  $\mathbf{z}_t$ . Specifically, given the corrupted instance

and corruption vector, the predictor uses a function  $\mathbf{w}_t(\cdot) : \{0, 1\}^d \rightarrow \mathbb{R}^d$  to choose a weight vector, and makes the prediction  $\hat{y}_t = \langle \mathbf{w}_t(\mathbf{z}_t), \mathbf{x}'_t \rangle$ . In order to provide theoretical guarantees, we will bound the following notion of regret,

$$R^z(T, \ell) = \sum_{t=1}^T \ell(\langle \mathbf{w}_t, \mathbf{x}'_t \rangle, y_t) - \inf_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^T \ell(\langle \mathbf{w}(\mathbf{z}_t), \mathbf{x}'_t \rangle, y_t), \quad (2)$$

where it is implicit that  $\mathbf{w}_t$  also depends on  $\mathbf{z}_t$  and  $\mathcal{W}$  now consists of corruption-dependent hypotheses. Similar definitions of regret have been looked at in the setting learning with side information [8, 12], but our special case admits stronger results in terms of both upper and lower bounds. In the most general case, we may consider  $\mathcal{W}$  as the class of all functions which map  $\{0, 1\}^d \rightarrow \mathbb{R}^d$ , however we show this can lead to an intractable learning problem. This motivates the study of interesting subsets of this most general function class. This is the main focus of Section 3.

### 2.2 Batch learning with missing features

In the setup of batch learning with i.i.d. data, examples  $(\mathbf{x}_t, \mathbf{z}_t, y_t)$  are drawn according to a fixed but unknown distribution and the goal is to choose a hypothesis that minimizes the expected error, with respect to an appropriate loss function  $\ell$ :  $\mathbf{E}_{\mathbf{x}_t, \mathbf{z}_t, y_t} [\ell(h(\mathbf{x}_t, \mathbf{z}_t), y_t)]$ .

The hypotheses  $h$  we consider in this scenario will be inspired by imputation-based methods prevalent in statistics literature used to address the problem of missing features [14]. An imputation mapping is a function used to fill in unobserved features using the observed features, after which the *completed* examples can be used for prediction. In particular, if we consider an imputation function  $\phi : \mathbb{R}^d \times \{0, 1\}^d \rightarrow \mathbb{R}^d$ , which is meant to fill missing feature values, and a linear predictor  $\mathbf{w} \in \mathbb{R}^d$ , we can parameterize a hypothesis with these two function  $h_{\phi, \mathbf{w}}(\mathbf{x}'_t, \mathbf{z}_t) = \langle \mathbf{w}, \phi(\mathbf{x}'_t, \mathbf{z}_t) \rangle$ .

It is clear that the multiplicative interaction between  $\mathbf{w}$  and  $\phi$  will make most natural formulations non-convex, and we elaborate more on this in Section 4. In the i.i.d. setting, the natural quantity of interest is the generalization error of our learned hypothesis. We provide a Rademacher complexity bound on the class of  $\mathbf{w}, \phi$  pairs we use, thereby showing that any hypothesis with a small empirical error will also have a small expected loss. The specific class of hypotheses and details of the bound are presented in Section 4. Furthermore, the reason as to why an imputation-based hypothesis class is not analyzed in the more general adversarial setting will also be explained in that section.

### 3 Online Corruption-Based Algorithm

In this section, we consider the class of *corruption-dependent* hypotheses defined in Section 2.1. Recall the definition of regret (2), which we wish to control in this framework, and of the comparator class of functions  $\mathcal{W} \subseteq \{0, 1\}^d \rightarrow \mathbb{R}^d$ . It is clear that the function class  $\mathcal{W}$  is much richer than the comparator class in the corruption-free scenario, where the best linear predictor is fixed for all rounds. It is natural to ask if it is even possible to prove a non-trivial regret bound over this richer comparator class  $\mathcal{W}$ . In fact, the first result of our paper provides a lower bound on the minimax regret when the comparator is allowed to pick arbitrary mappings, i.e. the set  $\mathcal{W}$  contains all mappings. The result is stated in terms of the minimax regret under the loss function  $\ell$  under the usual (corruption-free) definition (1):

$$R^*(T, \ell) = \inf_{\mathbf{w}_1 \in \mathcal{W}} \sup_{(\mathbf{x}_1, \mathbf{z}_1, y_1)} \cdots \inf_{\mathbf{w}_T \in \mathcal{W}} \sup_{(\mathbf{x}_T, \mathbf{z}_T, y_T)} R(T, \ell)$$

**Proposition 1** *If  $\mathcal{W} = \{0, 1\}^d \rightarrow \mathbb{R}^d$  the minimax value of the corruption dependent regret for any loss function  $\ell$  is lower bounded as*

$$\begin{aligned} \inf_{\mathbf{w}_1 \in \mathcal{W}} \sup_{(\mathbf{x}_1, \mathbf{z}_1, y_1)} \cdots \inf_{\mathbf{w}_T \in \mathcal{W}} \sup_{(\mathbf{x}_T, \mathbf{z}_T, y_T)} R^z(T, \ell) \\ = \Omega \left( 2^{d/2} R^* \left( \frac{T}{2^{d/2}}, \ell \right) \right). \end{aligned}$$

This proposition (the proof of which appears in the appendix [17]) shows that the minimax regret is lower bounded by a term that is exponential in the dimensionality of the learning problem. For most non-degenerate convex and Lipschitz losses,  $R^*(T, \ell) = \Omega(\sqrt{T})$  without further assumptions (see e.g. [1]) which yields a  $\Omega(2^{d/4}\sqrt{T})$  lower bound. The bound can be further strengthened to  $\Omega(2^{d/2}\sqrt{T})$  for linear losses which is unimprovable since it is achieved by solving the classification problem corresponding to each pattern independently.

Thus, it will be difficult to achieve a low regret against arbitrary maps from  $\{0, 1\}^d$  to  $\mathbb{R}^d$ . In the following section we consider a restricted function class and show that a mirror-descent algorithm can achieve regret polynomial in  $d$  and sub-linear in  $T$ , implying that the average regret is vanishing.

#### 3.1 Linear Corruption-Dependent Hypotheses

Here we analyze a corruption-dependent hypothesis class that is parametrized by a matrix  $\mathbf{A} \in \mathbb{R}^{d \times k}$ , where  $k$  may be a function of  $d$ . In the simplest case of  $k = d$ , the parametrization looks for weights  $\mathbf{w}(\mathbf{z}_t)$  that depend linearly on the corruption vector

$\mathbf{z}_t$ . Defining  $\mathbf{w}_{\mathbf{A}}(\mathbf{z}_t) = \mathbf{A}\mathbf{z}_t$  achieves this, and intuitively this allows us to capture how the presence or absence of one feature affects the weight of another feature. This will be clarified further in the examples.

In general, the matrix  $\mathbf{A}$  will be  $d \times k$ , where  $k$  will be determined by a function  $\psi(\mathbf{z}_t) \in \{0, 1\}^k$  that maps  $\mathbf{z}_t$  to a possibly higher dimension space. Given, a fixed  $\psi$ , the explicit parameterization in terms of  $\mathbf{A}$  is,

$$\mathbf{w}_{\mathbf{A}, \psi}(\mathbf{z}_t) = \mathbf{A}\psi(\mathbf{z}_t). \tag{3}$$

In what follows, we drop the subscript from  $\mathbf{w}_{\mathbf{A}, \psi}$  in order to simplify notation. Essentially this allows us to introduce non-linearities as a function of the corruption vector, but the non-linear transform is known and fixed throughout the learning process. Before analyzing this setting, we give a few examples and intuition as to why such a parametrization is useful. In each example, we will show how there exists a choice of a matrix  $\mathbf{A}$  that captures the specific problem’s assumptions. This implies that the fixed comparator can use this choice in hindsight, and by having a low regret, our algorithm would implicitly learn a hypothesis close to this reasonable choice of  $\mathbf{A}$ .

##### 3.1.1 Corruption-free special case

We start by noting that in the case of no corruption (i.e.  $\forall t, \mathbf{z}_t = \mathbf{1}$ ) a standard linear hypothesis model can be cast within the matrix based framework by defining  $\psi(\mathbf{z}_t) = \mathbf{1}$  and learning  $\mathbf{A} \in \mathbb{R}^{d \times 1}$ .

##### 3.1.2 Ranking-based parameterization

One natural method for classification is to order the features by their predictive power, and to weight features proportionally to their ranking (in terms of absolute value; that is, the sign of weight depends on whether the correlation with the label is positive or negative). In the corrupted features setting, this naturally corresponds to taking the available features at any round and putting more weight on the most predictive observed features. This is particularly important while using margin-based losses such as the hinge loss, where we want the prediction to have the right sign and be large enough in magnitude.

Our parametrization allows such a strategy when using a simple function  $\psi(\mathbf{z}_t) = \mathbf{z}_t$ . Without loss of generality, assume that the features are arranged in decreasing order of discriminative power (we can always rearrange rows and columns of  $\mathbf{A}$  if they’re not). We also assume positive correlations of all features with the label; a more elaborate construction works for  $\mathbf{A}$  when they’re not. In this case, consider the parameter

matrix and the induced classification weights

$$[\mathbf{A}]_{i,j} = \begin{cases} 1, & j = i \\ -\frac{1}{d}, & j < i \\ 0, & j > i \end{cases}, \quad [\mathbf{w}(\mathbf{z}_t)]_i = [\mathbf{z}_t]_i \left( 1 - \sum_{\substack{j < i: \\ [\mathbf{z}_t]_j = 1}} \frac{1}{d} \right).$$

Thus, for all  $i < j$  such that  $[\mathbf{z}_t]_i = [\mathbf{z}_t]_j = 1$  we have  $[\mathbf{w}(\mathbf{z}_t)]_i \geq [\mathbf{w}(\mathbf{z}_t)]_j$ . The choice of 1 for diagonals and  $1/d$  for off-diagonals is arbitrary and other values might also be picked based on the data sequence  $(\mathbf{x}_t, \mathbf{z}_t, y_t)$ . In general, features are weighted monotonically with respect to their discriminative power with signs based on correlations with the label.

### 3.1.3 Feature group based parameterization

Another class of hypotheses that we can define within this framework are those restricted to consider up to  $p$ -wise interactions between features for some constant  $0 < p \leq d$ . In this case, we index the  $k = \sum_{i=1}^p \binom{d}{i} = O\left(\left(\frac{d}{p}\right)^p\right)$  unique subsets of features of size up to  $p$ . Then define  $[\psi(\mathbf{z}_t)]_j = 1$  if the corresponding subset  $j$  is uncorrupted by  $\mathbf{z}_t$  and equal to 0 otherwise. An entry  $[\mathbf{A}]_{i,j}$  now specifies the importance of feature  $j$ , assuming that at least the subset  $i$  is present. Such a model would, for example, have the ability to capture the scenario of a feature that is only discriminative in the presence of some  $p-1$  other features. For example, we can generalize the ranking example from above to impose a soft ranking on groups of features.

### 3.1.4 Corruption due to failed sensors

A common scenario for missing features arises in applications involving an array of measurements, for example, from a sensor network, wireless motes, array of cameras or CCDs, where each sensor is bound to fail occasionally. The typical strategy for dealing with such situations involves the use of redundancy. For instance, if a sensor fails, then some kind of an averaged measurement from the neighboring sensors might provide a reasonable surrogate for the missing value.

It is possible to design a choice of  $\mathbf{A}$  matrix for the comparator that only uses the local measurement when it is present, but uses an averaged approximation based on some fixed averaging distribution on neighboring features when the local measurement is missing. For each feature, we consider a probability distribution  $p_i$  which specifies the averaging weights to be used when approximating feature  $i$  using neighboring observations. Let  $\mathbf{w}^*$  be the weight vector that the comparator would like to use if all the features were present. Then, with  $\psi(\mathbf{z}) = \mathbf{z}$  and for  $j \neq i$  we define,

$$[\mathbf{A}]_{i,i} = \mathbf{w}_i^* + \sum_{j \neq i} \mathbf{w}_j^* p_{ji}, \quad [\mathbf{A}]_{i,j} = -\mathbf{w}_j^* p_{ji}. \quad (4)$$

Thus, say only feature  $k$  is missing, we still have  $\mathbf{x}'_t \top \mathbf{A} \mathbf{z}_t = \sum_{i,j} [\mathbf{x}'_t]_i [\mathbf{z}_t]_j [\mathbf{A}]_{i,j} = \sum_{i \neq k, j \neq k} [\mathbf{x}'_t]_i [\mathbf{A}]_{i,j} = \sum_{i \neq k} [\mathbf{x}'_t]_i [\mathbf{w}^*]_i + [\mathbf{w}^*]_k \sum_{i \neq k} [\mathbf{x}'_t]_i p_{ki}$ , where by assumption  $\sum_{i \neq k} [\mathbf{x}'_t]_i p_{ki} \approx [\mathbf{x}'_t]_k$ .

Of course, the averaging in such applications is typically local, and we expect each sensor to put large weights only on neighboring sensors. This can be specified via a neighborhood graph, where nodes  $i$  and  $j$  have an edge if  $j$  is used to predict  $i$  when feature  $i$  is not observed and vice versa. From the construction (4) it is clear that the only off-diagonal entries that are non-zero would correspond to the edges in the neighborhood graph. Thus we can even add this information to our algorithm and constrain several off-diagonal elements to be zero, thereby restricting the complexity of the problem.

## 3.2 Matrix-Based Algorithm and Regret

We use a standard mirror-descent style algorithm [16, 3] in the matrix based parametrization described above. It is characterized by a strongly convex regularizer  $\mathcal{R} : \mathbb{R}^{d \times k} \rightarrow \mathbb{R}$ , that is

$$\mathcal{R}(\mathbf{A}) \geq \mathcal{R}(\mathbf{B}) + \langle \nabla \mathcal{R}(\mathbf{B}), \mathbf{A} - \mathbf{B} \rangle_F + \frac{1}{2} \|\mathbf{A} - \mathbf{B}\|^2 \quad \forall \mathbf{A}, \mathbf{B} \in \mathcal{A},$$

for some norm  $\|\cdot\|$  and where  $\langle \mathbf{A}, \mathbf{B} \rangle_F = \text{Tr}(\mathbf{A} \top \mathbf{B})$  is the trace inner product. An example is the squared Frobenius norm  $\mathcal{R}(\mathbf{A}) = \frac{1}{2} \|\mathbf{A}\|_F^2$ . For any such function, we can define the associated Bregman divergence

$$D_{\mathcal{R}}(\mathbf{A}, \mathbf{B}) = \mathcal{R}(\mathbf{A}) - \mathcal{R}(\mathbf{B}) - \langle \nabla \mathcal{R}(\mathbf{B}), \mathbf{A} - \mathbf{B} \rangle_F.$$

We assume  $\mathcal{A}$  is a convex subset of  $\mathbb{R}^{d \times k}$ , which could encode constraints such as some off-diagonal entries being zero in the setup of Section 3.1.4. To simplify presentation in what follows, we will use the shorthand  $\ell_t(\mathbf{A}) = \ell(\langle \mathbf{A} \psi(\mathbf{z}_t), \mathbf{x}'_t \rangle, y_t)$ . The algorithm initializes with any  $\mathbf{A}_0 \in \mathcal{A}$  and updates

$$\mathbf{A}_{t+1} = \arg \min_{\mathbf{A} \in \mathcal{A}} \{ \eta_t \langle \nabla \ell_t(\mathbf{A}_t), \mathbf{A} \rangle_F + D_{\mathcal{R}}(\mathbf{A}, \mathbf{A}_t) \} \quad (5)$$

If  $\mathcal{A} = \mathbb{R}^{d \times k}$  and  $\mathcal{R}(\mathbf{A}) = \frac{1}{2} \|\mathbf{A}\|_F^2$ , the update simplifies to gradient descent  $\mathbf{A}_{t+1} = \mathbf{A}_t - \eta_t \nabla \ell_t(\mathbf{A}_t)$ .

Our main result of this section is a guarantee on the regret incurred by Algorithm (5). The proof follows from standard arguments (see e.g. [16, 4]). Below, the dual norm is defined as  $\|\mathbf{V}\|_* = \sup_{\mathbf{U}: \|\mathbf{U}\| \leq 1} \langle \mathbf{U}, \mathbf{V} \rangle_F$ .

**Theorem 1** *Let  $\mathcal{R}$  be strongly convex with respect to a norm  $\|\cdot\|$  and  $\|\nabla \ell_t(\mathbf{A})\|_* \leq G$ , then Algorithm 5 with learning rate  $\eta_t = \frac{R}{G\sqrt{T}}$  exhibits the following regret upper bound compared to any  $\mathbf{A}$  with  $\|\mathbf{A}\| \leq R$ ,*

$$\sum_{t=1}^T \ell(\langle \mathbf{A}_t \mathbf{z}_t, \mathbf{x}'_t \rangle, y_t) - \inf_{\mathbf{A} \in \mathcal{A}} \sum_{t=1}^T \ell(\langle \mathbf{A} \mathbf{z}_t, \mathbf{x}'_t \rangle, y_t) \leq 3RG\sqrt{T}.$$

## 4 Batch Imputation Based Algorithm

Recalling the setup of Section 2.2, in this section we look at imputation mappings of the form

$$\phi_{\mathbf{M}}(\mathbf{x}', \mathbf{z}) = \mathbf{x}' + \text{diag}(1 - \mathbf{z})\mathbf{M}^\top \mathbf{x}'. \quad (6)$$

Thus we retain all the observed entries in the vector  $\mathbf{x}'$ , but for the missing features that are predicted using a linear combination of the observed features and where the  $i_{th}$  column of  $\mathbf{M}$  encodes the averaging weights for the  $i_{th}$  feature. Such a linear prediction framework for features is natural. For instance, when the data vectors  $\mathbf{x}$  are Gaussian, the conditional expectation of any feature given the other features is a linear function. The predictions are now made using the dot product

$$\langle \mathbf{w}, \phi(\mathbf{x}', \mathbf{z}) \rangle = \langle \mathbf{w}, \mathbf{x}' \rangle + \langle \mathbf{w}, \text{diag}(1 - \mathbf{z})\mathbf{M}^\top \mathbf{x}' \rangle,$$

where we would like to estimate  $\mathbf{w}, \mathbf{M}$  based on the data samples. From a quick inspection of the resulting learning problem, it becomes clear that optimizing over such a hypothesis class leads to a non-convex problem. The convexity of the loss plays a critical role in the regret framework of online learning, which is why we restrict ourselves to a batch i.i.d. setting here.

In the sequel we will provide a convex relaxation to the learning problem resulting from the parametrization (6). While we can make this relaxation for natural loss functions in both classification and regression scenarios, we restrict ourselves to a linear regression setting here as the presentation for that example is simpler due to the existence of a closed form solution for the ridge regression problem.

In what follows, we consider only the corrupted data and thus simply denote corrupted examples as  $\mathbf{x}_i$ . Let  $\mathbf{X}$  denote the matrix with  $i_{th}$  row equal to  $\mathbf{x}_i$  and similarly define  $\mathbf{Z}$  as the matrix with  $i_{th}$  row equal to  $\mathbf{z}_i$ . It will also be useful to define  $\bar{\mathbf{Z}} = \mathbf{1}\mathbf{1}^\top - \mathbf{Z}$  and  $\bar{\mathbf{z}}_i = \mathbf{1} - \mathbf{z}_i$  and finally let  $\bar{\mathbf{Z}}_i = \text{diag}(\bar{\mathbf{z}}_i)$ .

### 4.1 Imputed Ridge Regression (IRR)

In this section we will consider a modified version of the ridge regression (RR) algorithm, robust to missing features. The overall optimization problem we are interested in is as follows,

$$\min_{\{\mathbf{w}, \mathbf{M}: \|\mathbf{M}\|_F \leq \gamma\}} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{T} \sum_{i=1}^T (y_i - \mathbf{w}^\top (\mathbf{x}_i + \bar{\mathbf{Z}}_i \mathbf{M}^\top \mathbf{x}_i))^2 \quad (7)$$

where the hypothesis  $\mathbf{w}$  and imputation matrix  $\mathbf{M}$  are simultaneously optimized. In order to bound the size of the hypothesis set, we have introduced the constraint  $\|\mathbf{M}\|_F^2 \leq \gamma^2$  that bounds the Frobenius norm of the imputation matrix. The global optimum of the problem as presented in (7) cannot be easily found as

it is not jointly convex in both  $\mathbf{w}$  and  $\mathbf{M}$ . We next present a convex relaxation of the formulation (7). The key idea is to take a dual over  $\mathbf{w}$  but not  $\mathbf{M}$ , so that we have a saddle-point problem in the dual vector  $\alpha$  and  $\mathbf{M}$ . The resulting saddle point problem, while being concave in  $\alpha$  is still not convex in  $\mathbf{M}$ . At this step we introduce a new tensor  $\mathbf{N} \in \mathbb{R}^{d \times d \times d}$ , where  $[\mathbf{N}]_{i,j,k} = [\mathbf{M}]_{i,k} [\mathbf{M}]_{j,k}$ . Finally we drop the non-convex constraint relating  $\mathbf{M}$  and  $\mathbf{N}$  replacing it with a matrix positive semidefiniteness constraint.

Before we can describe the convex relaxation, we need one more piece of notation. Given a matrix  $\mathbf{M}$  and a tensor  $\mathbf{N}$ , we define the matrix  $\mathbf{K}_{\mathbf{MN}} \in \mathbb{R}^{T \times T}$

$$[\mathbf{K}_{\mathbf{MN}}]_{i,j} = \mathbf{x}_i^\top \mathbf{x}_j + \mathbf{x}_i^\top \mathbf{M} \bar{\mathbf{Z}}_i \mathbf{x}_j + \mathbf{x}_i^\top \bar{\mathbf{Z}}_j \mathbf{M}^\top \mathbf{x}_j + \sum_{k=1}^d [\bar{\mathbf{z}}_i]_k [\bar{\mathbf{z}}_j]_k \mathbf{x}_i^\top \mathbf{N}_k \mathbf{x}_j. \quad (8)$$

The following proposition gives the convex relaxation of the problem (7) that we refer to as Imputed Ridge Regression (IRR) and which includes a strictly larger hypothesis than the  $(\mathbf{w}, \mathbf{M})$  pairs with which we began.

**Proposition 2** *The following semi-definite programming optimization problem provides a convex relaxation to the non-convex problem (7):*

$$\begin{aligned} & \min_{t, \mathbf{M}: \|\mathbf{M}\|_F^2 \leq \gamma^2, \mathbf{N}: \sum_k \|\mathbf{N}_k\|_F^2 \leq \gamma^4} t \\ & \text{s.t.} \quad \begin{bmatrix} \mathbf{K}_{\mathbf{MN}} + \lambda T \mathbf{I} & \mathbf{y} \\ \mathbf{y}^\top & t \end{bmatrix} \succeq 0, \quad \mathbf{K}_{\mathbf{MN}} \succeq 0. \end{aligned} \quad (9)$$

The proof is deferred to the appendix for lack of space. The main idea is to take the quadratic form that arises in the dual formulation of (7) with the matrix  $\mathbf{K}_{\mathbf{M}}$ ,

$$[\mathbf{K}_{\mathbf{M}}]_{i,j} = \mathbf{x}_i^\top \mathbf{x}_j + \mathbf{x}_i^\top \mathbf{M} \bar{\mathbf{Z}}_i \mathbf{x}_j + \mathbf{x}_i^\top \bar{\mathbf{Z}}_j \mathbf{M}^\top \mathbf{x}_j + \mathbf{x}_i^\top \mathbf{M} \bar{\mathbf{Z}}_i \bar{\mathbf{Z}}_j \mathbf{M}^\top \mathbf{x}_j,$$

and relax it to the matrix  $\mathbf{K}_{\mathbf{MN}}$  (8). The constraint involving positive semidefiniteness of  $\mathbf{K}_{\mathbf{MN}}$  is needed to ensure the convexity of the relaxed problem. The norm constraint on  $\mathbf{N}$  is a consequence of the norm constraint on  $\mathbf{M}$ .

One tricky issue with relaxations is using the relaxed solution in order to find a good solution to the original problem. In our case, this would correspond to finding a good  $\mathbf{w}, \mathbf{M}$  pair for the primal problem (7). We bypass this step, and instead directly define the prediction on any point  $(\mathbf{x}_0, \mathbf{z}_0)$  as:

$$\begin{aligned} & \sum_{i=1}^T \alpha_i (\mathbf{x}_i^\top \mathbf{x}_0 + \mathbf{x}_i^\top \mathbf{M} \bar{\mathbf{Z}}_i \mathbf{x}_0 + \mathbf{x}_i^\top \bar{\mathbf{Z}}_0 \mathbf{M}^\top \mathbf{x}_0 \\ & + \sum_{k=1}^d [\bar{\mathbf{z}}_i]_k [\bar{\mathbf{z}}_0]_k \mathbf{x}_i^\top \mathbf{N}_k \mathbf{x}_0). \end{aligned} \quad (10)$$

Here,  $\alpha, \mathbf{M}, \mathbf{N}$  are solutions to the saddle-point problem

$$\min_{\substack{\mathbf{M}: \|\mathbf{M}\|_F \leq \gamma \\ \mathbf{N}: \sum_k \|\mathbf{N}_k\|_F^2 \leq \gamma^4}} \max_{\alpha} 2\alpha^\top \mathbf{y} - \alpha^\top (\mathbf{K}_{\mathbf{M}\mathbf{N}} + \lambda T \mathbf{I}) \alpha. \quad (11)$$

We start by noting that the above optimization problem is equivalent to the one in Proposition 2. The intuition behind this definition (10) is that the solution to the problem (7) has this form, with  $[\mathbf{N}]_{i,j,k}$  replaced with  $[\mathbf{M}]_{i,k} [\mathbf{M}]_{j,k}$ . In the next section, we show a Rademacher complexity bound over functions of the form above to justify our convex relaxation.

## 4.2 Theoretical analysis of IRR

As mentioned in the previous section, we predict with a hypothesis of the form (10) rather than going back to the primal class indexed by  $(\mathbf{w}, \mathbf{M})$  pairs. In this section, we would like to show that the new hypothesis class parametrized by  $\alpha, \mathbf{M}, \mathbf{N}$  is not too rich for the purposes of learning. To do this, we give the class of all possible hypotheses that can be the solutions to the dual problem (9) and then prove a Rademacher complexity bound over that class. The set of all possible  $\alpha, \mathbf{M}, \mathbf{N}$  triples that can be potential solutions to (9) lie in the following set

$$\mathcal{H} = \left\{ h(\mathbf{x}_0, \mathbf{z}_0) \mapsto \sum_{i=1}^T \alpha_i (\mathbf{x}_i^\top \mathbf{x}_0 + \mathbf{x}_i^\top \mathbf{M} \bar{\mathbf{z}}_i \mathbf{x}_0 + \mathbf{x}_i^\top \bar{\mathbf{z}}_0 \mathbf{M}^\top \mathbf{x}_0 + \sum_{k=1}^d [\bar{\mathbf{z}}_i]_k [\bar{\mathbf{z}}_0]_k \mathbf{x}_i^\top \mathbf{N}_k \mathbf{x}_0) : \|\mathbf{M}\|_F \leq \gamma, \|\mathbf{N}\|_F \leq \gamma^2, \|\alpha\| \leq \frac{B}{\lambda \sqrt{T}} \right\}$$

The bound on  $\|\alpha\|$  is made implicitly in the optimization problem (assuming the training labels are bounded  $\forall i, |y_i| \leq B$ ). To see this, we note that the problem (9) is obtained from (11) by using the closed-form solution of the optimal  $\alpha = (\mathbf{K}_{\mathbf{M}\mathbf{N}} + \lambda T \mathbf{I})^{-1} \mathbf{y}$ . Then we can bound  $\|\alpha\| \leq \|\mathbf{y}\| / \lambda_{\min}(\mathbf{K}_{\mathbf{M}\mathbf{N}} + \lambda T \mathbf{I}) = \frac{B\sqrt{T}}{\lambda T}$ , where  $\lambda_{\min}(\mathbf{A})$  denotes the smallest eigenvalue of the matrix  $\mathbf{A}$ . Note that in general there is no linear hypothesis  $\mathbf{w}$  that corresponds to the hypotheses in the relaxed class  $\mathcal{H}$  and that we are dealing with a strictly more general function class. However, the following theorem demonstrates that the Rademacher complexity of this function class is reasonably bounded in terms of the number of training points  $T$  and dimension  $d$  and thereby still provides provable generalization performance [2].

Recall the Rademacher complexity of a class  $\mathcal{H}$

$$\mathfrak{R}_T(\mathcal{H}) = \mathbf{E}_S \mathbf{E}_\sigma \left[ \frac{1}{T} \sup_{h \in \mathcal{H}} \left| \sum_{i=1}^T \sigma_i h(\mathbf{x}_i, \mathbf{z}_i) \right| \right], \quad (12)$$

where the inner expectation is over independent Rademacher random variables  $(\sigma_1, \dots, \sigma_T)$  and the outer one over a sample  $S = ((\mathbf{x}_1, \mathbf{z}_1), \dots, (\mathbf{x}_T, \mathbf{z}_T))$ .

**Theorem 2** *If we assume a bounded regression problem  $\forall y, |y| \leq B$  and  $\forall \mathbf{x}, \|\mathbf{x}\| \leq R$ , then the Rademacher complexity of the hypothesis set  $\mathcal{H}$  is bounded as follows,*

$$\mathfrak{R}_T(\mathcal{H}) \leq (1 + \gamma + (\gamma + \gamma^2)\sqrt{d}) \frac{BR^2}{\lambda\sqrt{T}} = O\left(\sqrt{\frac{d}{T}}\right).$$

Due to space constraints, the proof is presented in the appendix. Theorem 2 allows us to control the gap between empirical and expected risks using standard Rademacher complexity results. Theorem 8 of [2], immediately provides the following corollary.

**Corollary 3** *Under the conditions of Theorem 2, for any  $0 < \delta \leq 1$ , with probability at least  $1 - \delta$  over samples of size  $T$ , every  $h \in \mathcal{H}$  satisfies*

$$\mathbf{E}[(y - h(\mathbf{x}', \mathbf{z}))^2] \leq \frac{1}{T} \sum_{t=1}^T (y_t - h(\mathbf{x}'_t, \mathbf{z}_t))^2 + \frac{BR^2(1 + \gamma)^2}{\lambda} \left( \frac{BR^2(1 + \gamma)^2}{\lambda} \sqrt{\frac{d}{T}} + \sqrt{\frac{8 \ln(2/\delta)}{T}} \right).$$

## 5 Empirical Results

This section presents empirical evaluation of the online matrix-based algorithm 5, as well as the Imputed Ridge Regression algorithm of Section 4.1. We use baseline methods *zero-imputation* and *mean-imputation* where the missing entries are replaced with zeros and mean estimated from observed values of those features resp. Once the data is imputed, a standard online gradient descent algorithm or ridge-regression algorithm is used. As reference, we also show the performance of a standard algorithm on uncorrupted data. The algorithms are evaluated on several UCI repository datasets, summarized in Table 1.

The `thyroid` dataset includes naturally corrupted/missing data. The `optdigits` dataset is subjected to artificial corruption by deleting a column of pixels, chosen uniformly at random from the 3 central columns of the image (each image contains 8 columns of pixels total). The remainder of the datasets are subjected to two types of artificial corruption: *data-independent* or *data-dependent* corruption. In the first case, each feature is randomly deleted independently, while the features are deleted based on thresholding values in the latter case.

We report average error and standard deviations over 5 trials, using 1000 random training examples and corruption patterns. We tune hyper-parameters using a grid search from  $2^{-12}$  to  $2^{10}$ . Further details and explicit corruption processes appear in the appendix.

dataset	$m$	$d$	$F_I$	$F_D$
abalone	4177	7	.62 ± .08	.61 ± .12
housing	20640	8	.64 ± .08	.68 ± .20
optdigits	5620	64	.88 ± .00	.88 ± .00
park	3000	20	.58 ± .06	.61 ± .08
thyroid	3163	5	.77 ± .00	.77 ± .00
splice	1000	60	.63 ± .01	.66 ± .03
wine	6497	11	.63 ± .10	.69 ± .13

Table 1: Size of dataset ( $m$ ), features ( $d$ ) and, the overall fraction of remaining features in the training set after data-independent ( $F_I$ ) or data-dependent ( $F_D$ ) corruption.

## 5.1 Online Corruption Dependent Hypothesis

Here we analyze the online algorithm presented in section 3.2 using two different types of regularization. The first method simply penalizes the Frobenius norm of the parameter matrix  $\mathbf{A}$  (frob-reg),  $\mathcal{R}(\mathbf{A}) = \|\mathbf{A}\|_F^2$ . The second method (sparse-reg) forces a sparse solution by constraining many entries of the parameter matrix equal to zero as mentioned in Section 3.1.4. We use the regularizer  $\mathcal{R}(\mathbf{A}) = \gamma\|\mathbf{A}\mathbf{1}\|^2 + \|\mathbf{A}\|_F^2$ , where  $\gamma$  is an additional tunable parameter. This choice of regularization is based on the example given in equation (4), where we would have  $\|\mathbf{A}\mathbf{1}\| = \|\mathbf{w}^*\|$ .

We apply these methods to the `splice` classification task and the `optdigits` dataset in several one vs. all classification tasks. For `splice`, the sparsity pattern used by the sparse-reg method is chosen by constraining those entries  $[\mathbf{A}]_{i,j}$  where feature  $i$  and  $j$  have a correlation coefficient less than 0.2, as measured with the corrupted training sample. In the case of `optdigits`, only entries corresponding to neighboring pixels are allowed to be non-zero.

Figure 1 shows that, when subject to data-independent corruption, the zero imputation, mean imputation and frob-reg methods all perform relatively poorly while the sparse-reg method provides significant improvement for the `splice` dataset. Furthermore, we find data-dependent corruption is quite harmful to mean imputation as might be expected, while both frob-reg and sparse-reg still provide significant improvement over zero-imputation. More surprisingly, these methods also perform better than training on uncorrupted data. We attribute this to the fact that we are using a richer hypothesis function that is parametrized by the corruption vector while the standard algorithm uses only a fixed hypothesis. In Table 2 we see that the sparse-reg performs at least as well as both zero and mean imputation in all tasks and offers significant improvement in the 3-vs-all and 6-vs-all task. In this case, the frob-reg method performs comparably to sparse-reg and is omitted from the table due to space.

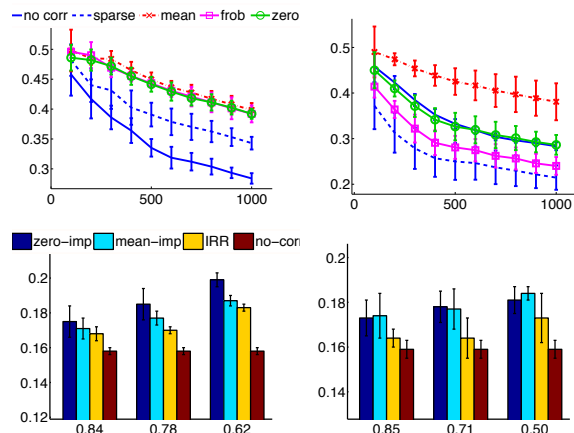


Figure 1: 0/1 loss as a function of  $T$  for `splice` dataset with independent (top left) and dependent corruption (top right). RMSE on `abalone` across varying amounts of independent (bottom left) and dependent corruption (bottom right); fraction of features remaining indicated on x-axis.

	zero-imp	mean-imp	sparse-reg	no corr
2	.035 ± .002	.039 ± .004	.033 ± .003	.024 ± .002
3	.041 ± .002	.043 ± .001	<b>.039 ± .002</b>	.027 ± .003
4	.020 ± .002	.023 ± .002	.021 ± .001	.015 ± .001
6	.026 ± .002	.024 ± .002	<b>.023 ± .002</b>	.015 ± .002

Table 2: One-vs-all classification results on `optdigits` dataset (target digit in first column) with column-based corruption for 0/1 loss.

## 5.2 Imputed Ridge Regression

In this section we consider the performance of IRR across many datasets. We found standard SDP solvers to be quite slow for problem (9). We instead use a semi-infinite linear program (SILP) to find an approximately optimal solution (see e.g. [13] for details).

In Tables 3 and 4 we compare the performance of the IRR algorithm to zero and mean imputation as well as to standard ridge regression performance on the uncorrupted data. Here we see IRR provides improvement over zero-imputation in all cases and does at least as well as mean-imputation when dealing with data-independent corruption. For data-dependent corruption, IRR continues to perform well, while mean-imputation suffers. For this setting, we have also compared to an *independent-imputation* method, which imputes data using an  $\mathbf{M}$  matrix that is trained independently of the learning algorithm. In particular the  $i_{th}$  column of  $\mathbf{M}$  is selected as the best linear predictor of the  $i_{th}$  feature given the rest, i.e. the solution to:  $\operatorname{argmin}_{\mathbf{v}} \sum_{\mathbf{x}_k \in \mathcal{X}_i} ([\mathbf{x}_k]_i - \sum_{j \neq i} [\mathbf{x}_k]_j [\mathbf{v}]_j)^2$ , where  $\mathcal{X}_i$  is the set of training examples that have the  $i_{th}$  feature present. Although, this method can perform better than mean-imputation, the joint optimization solution provided by IRR provides an even more significant improvement. At the bottom of Table 4 we also measure performance with `thyroid` which has naturally missing values. Here again IRR performs significantly

	zero-imp	mean-imp	IRR	no corr
A	.199 ± .004	.187 ± .003	<b>.183 ± .002</b>	.158 ± .002
H	.414 ± .025	<b>.370 ± .019</b>	<b>.373 ± .019</b>	.288 ± .001
P	.457 ± .006	<b>.445 ± .004</b>	.451 ± .004	.422 ± .004
W	.280 ± .006	<b>.268 ± .009</b>	<b>.269 ± .008</b>	.246 ± .001

Table 3: RMSE for various imputation methods across the datasets **abalone** (A), **housing** (H), **park** (P) and **wine** (W) when subject to data-independent corruption

	mean-imp	ind-imp	IRR	no corr
A	.180 ± .006	.183 ± .012	<b>.167 ± .011</b>	.159 ± .004
H	.400 ± .064	.363 ± .041	<b>.326 ± .035</b>	.289 ± .001
P	.444 ± .008	.423 ± .015	<b>.377 ± .035</b>	.422 ± .001
W	.264 ± .009	.260 ± .011	.256 ± .011	.247 ± .001
T	.531 ± .005	.528 ± .003	<b>.521 ± .004</b>	-

Table 4: RMSE for various imputation methods across the datasets **abalone** (A), **housing** (H), **park** (P) and **wine** (W) when subject to data-dependent corruption. The **thyroid** (T) dataset has naturally occurring missing features.

better than the competitor methods. Zero-imputation is not shown due to space, but it performs uniformly worse. Figure 1 shows more detailed results for the **abalone** dataset across different levels of corruption and displays the consistent improvement which the IRR algorithm provides.

In Table 5 we see that, with respect to the column-corrupted **optdigit** dataset, the IRR algorithm performs significantly better than zero-imputation and mean-imputation in majority of tasks.

## 6 Conclusion

We have introduced two new algorithms, addressing the problem of learning with missing features in both the adversarial online and i.i.d. batch settings. The algorithms are motivated by intuitive constructions and we also provide theoretical performance guarantees. Empirically we show encouraging initial results for online matrix-based corruption-dependent hypotheses as well as many significant results for the suggested IRR algorithm, which indicate superior performance when compared to several baseline imputation methods.

### Acknowledgements

We gratefully acknowledge the support of the NSF under award DMS-0830410. AA was partially sup-

	zero-imp	mean-imp	IRR	no corr
2	.352 ± .003	.351 ± .004	<b>.346 ± .002</b>	.321 ± .003
3	.450 ± .005	.435 ± .004	<b>.426 ± .005</b>	.398 ± .004
4	.372 ± .003	<b>.363 ± .002</b>	<b>.364 ± .003</b>	.345 ± .002
6	.369 ± .003	.360 ± .002	<b>.353 ± .003</b>	.333 ± .003

Table 5: RMSE (using binary labels) for one-vs-all classification on **optdigits** subject to column-based corruption.

ported by an MSR PhD Fellowship. We also thank anonymous reviewers for suggesting additional references and improvements to proofs.

## References

- [1] J. Abernethy, A. Agarwal, P. L. Bartlett, and A. Rakhlin. A stochastic view of optimal regret through minimax duality. *CoRR*, abs/0903.5328, 2009.
- [2] P.L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *JMLR*, 3, 2003.
- [3] A. Beck and M. Teboulle. Mirror descent and non-linear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3), 2003.
- [4] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambr. Univ. Press, 2006.
- [5] N. Cesa-Bianchi, S. Shalev-Shwartz, and O. Shamir. Efficient learning with partially observed attributes. *ICML*, 2010.
- [6] N. Cesa-Bianchi, S.S. Shwartz, and O. Shamir. Online Learning of Noisy Data with Kernels. *COLT*, 2010.
- [7] G. Chechik, G. Heitz, G. Elidan, P. Abbeel, and D. Koller. Max-margin classification of data with absent features. *JMLR*, 9, 2008.
- [8] T.M. Cover and E. Ordentlich. Universal portfolios with side information. *Information Theory, IEEE Transactions on*, 42(2):348–363, mar 1996.
- [9] O. Dekel, O. Shamir, and L. Xiao. Learning to classify with missing and corrupted features. *Machine learning*, 2010.
- [10] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journ. of the Royal Stat. Society*, 39(1), 1977.
- [11] A. Globerson and S. Roweis. Nightmare at test time: robust learning by feature deletion. In *ICML*, 2006.
- [12] E. Hazan and N. Megiddo. Online learning with prior information. In *COLT*, 2007.
- [13] K. Krishnan and J.E. Mitchell. Semi-infinite linear programming approaches to semidefinite programming problems. *Novel approaches to hard discrete optimization problems*, 37, 2003.
- [14] R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data*. Wiley New York, 1987.
- [15] B. M. Marlin. *Missing Data Problems in Machine Learning*. PhD thesis, University of Toronto, 2008.
- [16] A. S. Nemirovski and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. 1983.
- [17] A. Rostamizadeh, A. Agarwal, and P. Bartlett. Online and Batch Learning Algorithms for Data with Missing Features. *ArXiv e-prints*, 2011.