Sparse Linear Regression With Missing Data

Ravi Ganti [*1] and Rebecca M. Willett[†2]

[1]Wisconsin Institutes for Discovery, 330 N Orchard St, Madison, WI, 53715
[2]Department of Electrical and Computer Engineering, University of Wisconsin-Madison,
Madison, WI, 53706

March 31, 2015

Abstract

This paper proposes a fast and accurate method for sparse regression in the presence of missing data. The underlying statistical model encapsulates the low-dimensional structure of the incomplete data matrix and the sparsity of the regression coefficients, and the proposed algorithm jointly learns the low-dimensional structure of the data and a linear regressor with sparse coefficients. The proposed stochastic optimization method, Sparse Linear Regression with Missing Data (SLRM), performs an alternating minimization procedure and scales well with the problem size. Large deviation inequalities shed light on the impact of the various problem-dependent parameters on the expected squared loss of the learned regressor. Extensive simulations on both synthetic and real datasets show that SLRM performs better than competing algorithms in a variety of contexts.

1 Introduction

Modern statistical data analysis requires tools that can handle complex, large scale datasets. Due to constraints in the data collection process, one often has incomplete datasets, i.e., datasets with missing entries, with which we need to perform statistical inference. For instance, in sensor networks, readings from all the sensors might not be available at all the times because of malfunctions in sensors, or simply because it is too expensive to gather readings from all the sensors at all the times. Similarly, when conducting surveys, responders may avoid answering certain questions for the sake of privacy or otherwise, leading to missing entries in survey data. Recommender systems, implement algorithms that are required to train on data with missing entries. For example, popular recommendation engines such as Netflix, online radio services such as Pandora, social networks such as Facebook, LinkedIn regularly deal with prediction problems involving data with missing entries. An ever increasing demand to gather as much data as possible, clean or not, in this big-data era, has led to the need for statistical methods that can deal with not just clean data but also noisy data with missing components.

The focus of this paper is on sparse linear regression when the feature vectors or design matrix have missing elements. Matrix completion methods allow missing elements to be imputed accurately, but generally do not account for any auxiliary label information. Similarly, sparse linear regression and LASSO methods rely upon a fully-known design matrix. One might imagine using matrix completion to impute missing entries and then applying sparse linear regression methods to the completed design matrix; *we demonstrate that this two-stage approach is sub-optimal, and propose a unified regression framework that yields significantly better performance in a variety of tasks.*

[*] gantimahapat@wisc.edu
[†] rmwillett@wisc.edu

1.1 Contributions.

Our contributions are as follows

1. In this paper, we propose a statistical model (Section 2) for the problem of sparse linear regression with missing data. Our model captures low-rank structure in the data and sparsity of the regression coefficients in the lower dimensional representation of the data.
2. We provide an optimization-based approach that simultaneously learns the underlying subspace structure and the sparse regression coefficients (Section 4). Our optimization algorithm, called SLRM, takes a combination of stochastic first order and second order steps, alternating between the different parameters of the proposed statistical models.
3. We establish large deviation bounds (Section 5) for the risk of the regressor learned by our algorithm in terms of the empirical loss, the ambient dimension D, and a parameter γ used by our learning algorithm. Using our performance bounds we can understand the impact of the amount of missingness on the training error and the test error.
4. We provide extensive experimental results (Section 6) on synthetic and real datasets, comparing the performance of SLRM and a competing algorithm. From our experimental results, we conclude that SLRM has good noise tolerance properties, and uses the label information well to learn a good regressor, as measured by its mean squared error on a test dataset with missing features.

2 Problem Formulation: Sparse Regression With Missing Data

Given D-dimensional labeled data with missing features, we are interested in prediction, particularly regression problems. Let $X = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{D \times n}$ be a data matrix, where the columns have been sampled i.i.d. from a distribution. Since we are interested in regression problems with missing data, we do not get to see all the entries of the data matrix X. To formalize this notion, let $\Omega_1, \ldots, \Omega_n$ be subsets of $\{1, 2, \ldots, D\}$. Given an index set Ω, let $P_\Omega(x)$ denote a sub-vector of x consisting of elements whose indices are elements of the set. We observe a dataset $(P_{\Omega_1}(x_1), y_1, \Omega_1), \ldots, (P_{\Omega_n}(x_n), y_n, \Omega_n)$ of size n, i.e., we observe only a few entries of the data points $x_1, \ldots, x_n$, where the entries are indexed by the sets $\Omega_1, \ldots, \Omega_n$ respectively. We call the vector $Y = (y_1, y_2, \ldots, y_n)^\top$ the label vector. Given this training data, we are required to learn a regressor, which when given an unseen test point $(P_\Omega(x), \Omega)$, predicts a label $\hat{y}$ that is close to the true label of x. In order to solve this problem, we consider the following statistical model:

$$X = U_* A_* + \epsilon_X \tag{1}$$

$$Y = A_*^\top w_* + \epsilon_Y, \tag{2}$$

where w_* is a sparse vector in $\mathbb{R}^d$, U_* in $\mathbb{R}^{D \times d}$ ($d < D$) is a matrix with full column rank, and $A_* = [\alpha_{1*}, \ldots, \alpha_{n*}]$ is a matrix in $\mathbb{R}^{d \times n}$. We call α_{i*} the code of x_i w.r.t. the matrix U. The vector $\epsilon_Y = (\epsilon_{y_1}, \ldots, \epsilon_{y_n})^\top$ is random noise that is independent of other problem parameters such as $U, A, w, \Omega_1, \ldots, \Omega_n$. Similarly $\epsilon_X = [\epsilon_{x_1}, \ldots, \epsilon_{x_n}]$ is a noise matrix with i.i.d. entries, sampled independently of other problem parameters.

Our statistical model given in Equations 1,2 is motivated by the fact, for many data matrices of interest, even though the ambient data dimensionality is large, the data lies close to a lower dimensional subspace of dimensionality d. Given, this d-dimensional representation of the data, we are interested in learning a linear regressor with sparse coefficients that predicts the labels well.

To the best of our knowledge, for the problem of regression with missing data, our work is the first work that simultaneously exploits both a low-rank structure of the incomplete data matrix and the sparsity of regressor. The assumption of a parametric model for our regression problem allows us to go beyond the transductive setting which was inherent in the approach of Goldberg *et al.* (2010) (as detailed in Section 3). While we consider $d < D$, we are also interested in cases where d is of the same order as D and the regressor is sparse in the lower-dimensional representation of the data. This model is relevant to many applications, as described in Section 6

For instance, in a sensor network D sensors listen to d sources. As one would expect, this sensor data is far from being "clean": it is usually noisy, and has missing entries. A common approach in analyzing such sensor data is to perform a subspace analysis of the sensor data (Tuncer and Friedlander, 2009; Krim *et al.*, 1995; Roy and Kailath, 1989) and find the best fit d-dimensional subspace of the data. For modern sensor networks, both D and d are large; that is, a large number of heterogeneous sensors listen to a large number of sources. Exploiting the underlying d-dimensional structure during regression yields increased robustness to noise and missing data.

Notation. Like in the definition of A_*, $A = [\alpha_1, \alpha_2, \ldots, \alpha_n]$, $\hat{A} = [\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_n]$. Given a matrix M, denote $P_\Omega(M)$ as the matrix whose rows are those rows of M whose indices are elements of the set Ω. For example, if $\Omega = \{1, 3, 4\}$, then $P_\Omega(M)$ has rows 1,3,4 of matrix M. At times, for ease of notation we may write x_Ω, M_Ω to denote $P_\Omega(x), P_\Omega(M)$ respectively. By I_d we represent an identity matrix with d rows.

3 Related Work

Our statistical model bears resemblance to the statistical model used in partial least squares (PLS) (Hastie *et al.*, 2003). However, unlike PLS we enforce additional sparsity assumptions and can handle missing data. Dictionary learning was introduced for unsupervised data analysis for better data representation (Maurer and Pontil, 2010; Vainsencher *et al.*, 2011). The idea is to learn a dictionary so that each data point could be represented well as a sparse linear combination of the columns of the dictionary. Dictionary learning has also been extended to prediction problems (Mairal *et al.*, 2012; Szlam and Sapiro, 2009), where the problem is to learn a dictionary for the prediction problem at hand. The problem that we tackle in this paper can be seen as learning a dictionary for prediction problems in the presence of missing data. Sufficient dimensionality reduction (SDR) (Suzuki and Sugiyama, 2013; Fukumizu *et al.*, 2009), is a form of supervised dimensionality reduction, where the problem is to find a central subspace Z such that the prediction task is independent of the unlabeled data given the projection $\Pi_Z X$ of unlabeled data onto the central subspace. SDR focuses on achieving conditional independence between Y and X given $\Pi_Z X$ without making any assumptions on the functional dependency of the prediction task on the central subspace. SDR does not fully exploit linear relationships between labels and features that arise in many practical settings, and the problem of SDR with missing data has not been investigated. Loh and Wainwright (2011) investigate non-convex algorithms based on maximum likelihood estimation for the problem of high-dimensional regression with missing data. However, they work with a different statistical model which does not capture the low-rank structure of the data and assumes that the regressor is sparse in the ambient space. In contrast, our statistical model explicitly assumes that the missing data matrix has a low-rank structure and exploits this low-rank structure in data to learn a regressor with sparse coefficients in the low-dimensional representation of the data. Another closely related work is that of (Goldberg *et al.*, 2010), where the authors consider the problem of multi-task regression with missing data features and missing labels. The authors pose this problem as a matrix completion problem of the matrix formed by the concatenation of the data and label matrices. However, the authors deal with the transductive setting only – their approach does not allow one to predict a label for a new test datapoint. In contrast, this paper exploits an alternative statistical model for how the labels are generated that allows prediction on new test datapoints. Finally, Principal Component Regression (PCR) (Hastie *et al.*, 2003) is a dimensionality-reduction based procedure for regression without missing data. PCR first performs PCA on the unlabeled dataset, followed by least squares regression in the PCA space. This two-step approach does not exploit label information when estimating the underlying low-dimensional model; the limitations of this choice are detailed in Section 4.

As mentioned above, we use stochastic optimization methods that can operate on streaming data to ensure scalable algorithms. Thus the low-rank structure in our problem is estimated using techniques drawn from the subspace tracking literature. Oja's method (Oja, 1982), PAST (Yang, 1995) and variations such as OPAST (Abed-Meraim *et al.*, 2000) perform subspace tracking when there is no missing data. More recent developments, such as GROUSE (Balzano *et al.*, 2010b) and PETRELS (Chi *et al.*, 2012) can handle missing data quickly and accurately. However, these algorithms are inherently unsupervised and hence do not directly address the supervised regression problem considered in this paper.

4 An Optimization Approach And A Learning Algorithm

Before we describe our optimization based approach to the problem considered in this paper, we discuss a multi-step approach (essentially an extension of PCR to missing data problems) that exploits both the low rank of the incomplete data matrix and the sparsity of the regression coefficients:

1. Solve the following optimization problem

$$\min_{U,\alpha_1,\dots,\alpha_n} \sum_{i=1}^{n} ||P_{\Omega_i}(x_i) - P_{\Omega_i}(U\alpha_i)||_2^2. \tag{3}$$

   The above problem aims to consider a decomposition of the incomplete data matrix X as the product of two matrices U, A, such that the Frobenius norm of the difference between X and UA over the observed entries is minimized. This problem has been studied in the matrix completion literature (Koren *et al.*, 2009; Jain *et al.*, 2013), and in the subspace identification and tracking literature (Chi *et al.*, 2012; Hua *et al.*, 1999). A standard approach to solving this problem is via alternating minimization, where we alternate between optimization w.r.t. U and the vectors $\alpha_1, \dots, \alpha_n$. In the special case that for all $i = 1, \dots, n$, $\Omega_i = \{1, 2, \dots, D\}$ (as in classical PCR), the solution to the above problem is obtained by performing PCA of the data matrix.

2. Let $\hat{U}$, $\hat{A} \overset{\text{def}}{=} [\hat{\alpha}_1, \dots, \hat{\alpha}_n]$ be the solution of (3). Learn a linear regressor with sparse coefficient, using $\hat{A}$ as the design matrix and by solving the following ℓ_1 penalized problem

$$\hat{w} = \arg\min_{w} \frac{1}{n}||Y - \hat{A}^\top w||_2^2 + \lambda||w||_1. \tag{4}$$

We call the above two step procedure MPCR[1]. Note that in Step 1 of MPCR, the label data is not used. A merit of MPCR over other approaches previously proposed for our problem is that MPCR explicitly utilizes the low-rank structure of the data, and a linear model for the regression task at hand.

However, such multi-step algorithms that do not utilize the label information in all the steps are inherently label-inefficient. First, such multi-step algorithms fail to exploit information about U_* reflected by the labels. Second, the estimate $\hat{U}$ is one basis (of many potential bases) of the underlying subspace. Since we perform sparse regression on the subspace coefficients, the choice of basis matters. However, without label information, we have no way of knowing which basis rotation is best. Third, MPCR solves a harder problem than necessary. To see why, note that when w_* is sparse, then for the purpose of prediction, only those rows of A_* and columns of U_* that correspond to the non-zero coordinates of w_* matter.

In general, any multi-step procedure that does not utilize label information when estimating the underlying subspace will be label-inefficient for learning a good predictor. This is particularly true when both D and d are of the same order, as in the sensor network problems described in Section 2. This is because when d is comparable to D, there is a good deal of information in the labels that can be used to efficiently estimate the underlying subspace. We observe this in our experiments too, where on the CT slice dataset, where $D = 384, d = 181$, MPCR gives substantially worse performance than our proposed algorithm.

Armed with these insights, we are interested in procedures that utilize label information fully. We do this by proposing a joint optimization procedure that simultaneously learns all the relevant variables in our model.

4.1 Learning Via Joint Optimization

Given constants $\lambda_1, \lambda_2, \lambda_3 > 0$, we propose to solve the following optimization problem.

$$\begin{aligned} &\underset{U,A,w}{\text{minimize}} \ \frac{\lambda_1}{n}\sum_{i=1}^{n} ||P_{\Omega_i}(x_i) - P_{\Omega_i}(U\alpha_i)||_2^2 + \quad \frac{1}{n}||Y - A^\top w||_2^2 + \lambda_2||w||_1 + \lambda_3||w||_2^2 \\ &\text{subject to} \ \ U^\top U = I_d, \end{aligned} \tag{5}$$

[1] M in MPCR stands for missing

where $U \in \mathbb{R}^{D\times d}, A = [\alpha_1, \dots, \alpha_n] \in \mathbb{R}^{d\times n}, w \in \mathbb{R}^d$. Like in MPCR, the first term corresponds to a matrix completion term. The second term in the above optimization formulation measures the squared loss of a regressor w on a low-dimensional representation of the training data. The third term in our optimization formulation is the ℓ_1 norm penalty which encourages sparse w. Finally the last term is motivated by elastic net type formulation for sparse prediction. We optimize over U, w, A, under the constraints that the columns of U be orthonormal to each other to ensure uniqueness of the solution. The above optimization procedure outputs $\hat{w}, \hat{U}, \hat{A}$. Given an unlabeled data point with missing entries, $(P_\Omega(x), \Omega)$, and a constant $\gamma \in [0, 1)$, we first project the point onto the subspace spanned by the columns of the matrix $\hat{U}_\Omega$, to obtain $\tilde{x} = (\hat{U}_\Omega^\top \hat{U}_\Omega)^{-1}\hat{U}_\Omega^\top x_\Omega$. Our regressor, $\hat{f}_{\hat{w},\hat{U},\gamma}$ then predicts the label of $(P_\Omega(x), \Omega)$ as

$$\hat{f}_{\hat{w},\hat{U},\gamma}(P_\Omega(x), \Omega) = \hat{w}^\top \tilde{x}\mathbb{1}_{\{||(\hat{U}_\Omega^\top \hat{U}_\Omega)^{-1}||_2 \leq \frac{D}{m(1-\gamma)}\}}, \tag{6}$$

where $\mathbb{1}_{\{.\}}$ is the indicator function. Whenever $||(\hat{U}_\Omega^\top \hat{U}_\Omega)^{-1}||$ is large, it implies that the missing entries will not allow accurate subspace projection; in this case, our method outputs 0. A good choice of γ depends on d and $|\Omega|$, and we shall discuss this in detail in Section 5.

4.2 Solving The Optimization Problem

The optimization problem shown in (5) is individually convex in the optimization variables U, A, w, but jointly non-convex. We solve this problem via an alternating minimization approach, where we minimize over U, A, w alternatively. In addition, we adopt a stochastic optimization approach. Our algorithm is called Sparse Linear Regression with Missing data (SLRM). SLRM makes a pass over the dataset, and each time uses a single data point $(P_{\Omega_t}(x_t), y_t, \Omega_t)$ to make updates to all the parameters. SLRM uses stochastic second order steps to update matrices U, A, and stochastic first order steps to update w vector. Algorithm 4.3 provides a pseudocode of our proposed stochastic optimization algorithm. There are six main steps, which we shall discuss below in detail.

Initialization. In Step 1 we initialize U, w to $\hat{U}_0, \hat{w}_0$. $\hat{U}_0$ is obtained by performing SVD of the incomplete data matrix with 0's filled in the missing entries. The left singular vectors corresponding to the top d singular values form the $\hat{U}_0$ matrix. Similar initialization techniques have been proposed in matrix completion literature (Jain *et al.*, 2013; Hardt, 2013; Koren *et al.*, 2009). We initialize $\hat{A}_0$ by projecting each $P_{\Omega_i}(x_i)$ onto the subspace spanned by $\hat{U}_0$. We initialize $\hat{w}_0$ by solving the LASSO regression on Y and $\hat{A}_0$, similar to (4). We initialize matrices, $R_1^0, R_2^0, \dots, R_D^0$ to a multiple of the identity matrix. These matrices are required in Step 6 of our algorithm.

Updating A. In round t, SLRM uses $(P_{\Omega_t}(x_t), y_t, \Omega_t)$ to update our estimate of the A matrix. Since $(P_{\Omega_t}(x_t), y_t, \Omega_t)$ is only responsible for the t^{th} column of matrix A, in Step 5 of SLRM we replace the t^{th} column of $\hat{A}_{t-1}$, with $\hat{\alpha}_t$, to obtain $\hat{A}_t$. This update reduces to a simple unconstrained quadratic optimization problem over α_t, which can be solved in closed form by solving a system of linear equations.

Updating U. In Step 6, we update $\hat{U}_{t-1}$ by using the MODIFIED-PETRELS (MP) routine. The MP routine is inspired by the PETRELS algorithm (Chi *et al.*, 2012), which was designed for estimating subspaces from streaming data with missing entries. PETRELS can be seen as solving the optimization problem $\underset{U}{\arg\min} \sum_{i=1}^n f_i(U)$, where $f_i(U) = ||P_{\Omega_i}(x_i - U\alpha_i)||^2$, and α's correspond to a projection of the observations onto the current subspace estimate. MP solves the same optimization problem, but with the α_i from Step 5, *which uses label information.* Both methods update $\hat{U}_{t-1}$ to $\tilde{U}_t$ by performing a single stochastic Newton step on $f_t(U)$, starting at $\hat{U}_{t-1}$, and using $P_{\Omega_t}(x_t), \Omega_t, \alpha_t$. This Newton step can be implemented efficiently using recursive least squares, and a pseudocode for the MP routine is available in Algorithm 2.

Orthonormalization of updated U**.** Since we are optimizing over the manifold of rectangular matrices with orthonormal columns, we perform an orthonormalization step in Step 9, by solving the following nearest orthogonal matrix problem: $U_t = \underset{U}{\arg\min} ||U - \tilde{U}_t||_F$ subject to $U^\top U = I_d$. This problem has the closed form solution as shown in Step 7 of SLRM. Note that by construction, our orthonormalization step always guarantees, that the columns of $\hat{U}_t$ always span a d-dimensional subspace of $\mathbb{R}^D$.

Updating w. In Step 8, we perform one step of the stochastic projected gradient algorithm w.r.t. w. Our objective function is $\underbrace{\frac{1}{n}||Y - \hat{A}_t^\top w||_2^2 + \lambda_3||w||_2^2}_{F(w)} + \lambda_2||w||_1$. A step of the stochastic projected gradient method requires us to calculate a noisy estimate of, $\nabla F(w_{t-1})$, using α_t, y_t, followed by an application of the prox operator corresponding to $||w||_1$.

Validation Steps. We let $\hat{w}$ and $\hat{U}$ denote the estimate stored at the end of the previous round. In Steps 9-12, we determine, using a hold-out validation set, whether $\hat{w}$ and $\hat{U}$ form a better regressor than $\hat{w}_t$ and $\hat{U}_t$. The pair that achieves smaller hold-out error is then stored as $\hat{w}$ and $\hat{U}$ for the next round.

These steps are required since we are solving a non-convex optimization problem, and hence it is not necessarily true that $(\hat{w}_T, \hat{U}_T)$ leads to the best regressor.

Note that SLRM can easily be modified to handle the case where we have semi-supervised data. If we get unlabeled data in a round t, then we perform the optimization problem in Step 5 of the SLRM algorithm without the term $(y_t - \hat{w}_{t-1}^\top \alpha)^2$, and simply skip the weight update in Step 8.

4.3 Computational Complexity and Convergence

Step 5 of SLRM solves a system of linear equations and takes $O(|\Omega_t|d^2)$ time. Step 6 of SLRM allows a parallel implementation, where the rows of the matrix $\tilde{U}_t$ are updated in parallel. This takes $O(|\Omega_t|d^2)$ time. Finally, Step 7 of SLRM is the classical orthogonal Procrustes problem and takes $O(Dd^2 + d^3)$ time. Hence, all together the time complexity of our algorithm is $O(|\Omega_t|d^2 + Dd^2)$. In particular, since steps 5,7 are well studied numerical problems, they have efficient numerical implementations available. Since, our algorithm is built on exploiting the low rank structure of the missing data matrix, we attempt to get a rough estimate of the subspace spanned by the data. Algorithms that attempt to estimate the subspace spanned by missing data such as GROUSE (Balzano *et al.*, 2010b), PAST (Yang, 1995) need to expend $O(|\Omega_t|d^2)$ computation. Hence, it appears at least $O(|\Omega_t|d^2)$ amount of computation is inevitable. The overall higher computational complexity of our SLRM over algorithms such as PAST, GROUSE etc. is because of the additional prediction task that we aim to solve with SLRM.

SLRM, like other task driven dictionary learning approaches (Mairal *et al.*, 2012), uses a combination of stochastic updates and alternating minimization for a non-convex objective function. Empirically we observe similar convergence behavior to that reported in (Mairal *et al.*, 2012); to the best of our knowledge, no formal convergence guarantees are available for SGD based approaches to the task driven dictionary learning problem (Mairal *et al.*, 2012). Note that Mairal *et al.* (2009) also uses stochastic updates and alternating minimization for a biconvex objective and describes associated convergence analysis; however, the problem considered in that paper is far simpler than ours, in that it was not task-driven and didn't handle missing data.

5 Generalization Error Bounds

Definition 1. *Given a $\gamma \in [0, 1)$, let*

$$f_{w,U}(x_\Omega, \Omega) = w^\top (U_\Omega^\top U_\Omega)^{-1} U_\Omega^\top x_\Omega \mathbb{1}_{\{||(U_\Omega^\top U_\Omega)^{-1}||_2 \le \frac{D}{m(1-\gamma)}\}}, \tag{7}$$

and

$$\mathcal{F}_\gamma \overset{\text{def}}{=} \{f_{w,U} | w \in \mathbb{R}^d, U \in \mathbb{R}^{D\times d}, ||w||_1 \le R_1, U^\top U = I_d, f_{w,U} \textit{ is as given in Equation } 7\}.$$

Our main theorem is as follows

Theorem 1. *Consider a regression problem where a training set of n data samples $(P_{\Omega_i}(x_i), y_i, \Omega_i)$ are sampled i.i.d. from a probability distribution, with $|y_i| \le B_Y$, $||x_i||_\infty \le B_X$, almost surely. Let each Ω_i be a set of cardinality m, chosen uniformly at random with replacement from the set $\{1, 2, \ldots, D\}$. Let $b \overset{\text{def}}{=} 2(B_Y + \frac{DR_1}{m(1-\gamma)})^2$. Choose a*

Algorithm 1 SLRM. Input: Parameters $\lambda_1, \lambda_2, \lambda_3, \delta > 0, 0 \leq \gamma < 1$, Output: $\hat{w}, \hat{U}$

```
Initialize $\hat{w} = \hat{w}_0, \hat{U} = \hat{U}_0, \hat{A} = \hat{A}_0$, $(R_1^0)^\dagger = \delta I_d, (R_2^0)^\dagger = \delta I_d, \ldots, (R_D^0)^\dagger = \delta I_d$.
Initialize $curr\_best\_val\_err = \infty$.
for $t = 1, 2, \ldots n$ do
    Receive $(P_{\Omega_t}(x_t), y_t, \Omega_t)$
    Replace $t^{\text{th}}$ column of $\hat{A}_{t-1}$ with $\hat{\alpha}_t$ to get $\hat{A}_t$. $\hat{\alpha}_t$ is given by,
        $\hat{\alpha}_t = \arg\min_{\alpha} \lambda_1 ||P_{\Omega_t}(x_t) - P_{\Omega_t}(\hat{U}_{t-1}\alpha)||_2^2 + (y_t - \hat{w}_{t-1}^\top \alpha)^2$
    Update $U$, using the current sample $(P_{\Omega_t}(x_t), \Omega_t)$, as follows
        $\tilde{U}_t, R_1^t, R_2^t, \ldots, R_D^t \leftarrow \text{Modified-PETRELS}(\hat{U}_{t-1}, P_{\Omega_t}(x_t), \Omega_t, \hat{\alpha}_t, (R_1^{t-1})^\dagger, \ldots, (R_D^{t-1})^\dagger)$
    Orthonormalize by, $\hat{U}_t \leftarrow \tilde{U}_t(\tilde{U}_t^\top \tilde{U}_t)^{-1/2}$
    Perform stochastic proximal gradient type update using the following equations
        $\hat{w}_t = \text{prox}_{\eta_t\lambda_2, ||\cdot||_1}\Big[\hat{w}_{t-1} - \eta_t\Big(2(\hat{\alpha}_t\hat{\alpha}_t^\top \hat{w}_{t-1} - y_t\hat{\alpha}_t) + \lambda_3\hat{w}_{t-1}\Big)\Big]$
    val_err =Validation-Error($\hat{w}_t, \hat{U}_t, \gamma$)
    if $val\_err < curr\_best\_val\_err$ then
        curr_best_val_err = val_err.
        $\hat{w} \leftarrow \hat{w}_t, \hat{U} \leftarrow \hat{U}_t$
    end if
end for
```

$\gamma \in [0,1)$. *Let* $\hat{L}(f) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(P_{\Omega_i}(x_i), \Omega_i))^2$, $L(f) \overset{\text{def}}{=} \mathbb{E}_{x,y,\Omega}(y - f(P_\Omega(x), \Omega))^2$ *Then for any* $\delta > 0$, *and a universal constant* $K > 0$, *we have with probability at least* $1 - \delta$, *over a random sample of size* n, *for all* $f \in \mathcal{F}_\gamma$,

$$L(f) \leq \hat{L}(f) + K\left[\sqrt{\hat{L}(f)}\left(\left(\frac{m}{n} + \frac{1}{\sqrt{n}}\right)\frac{DR_1B_X}{1-\gamma} + \sqrt{\frac{b\log(1/\delta)}{n}}\right) + \frac{b\log(1/\delta)}{n} + \log^3(n)\left(\frac{m}{n} + \frac{1}{\sqrt{n}}\right)^2\left(\frac{DR_1B_X}{1-\gamma}\right)^2\right] \tag{8}$$

For appropriate values of $\lambda_1, \lambda_2, \lambda_3, R_1$, the output of SLRM $\hat{f}_{\hat{w},\hat{U},\gamma} \in \mathcal{F}_\gamma$. The complete proof is in the appendix. Here, we shall provide a brief synopsis of the proof.

Algorithm 2 MODIFIED-PETRELS. Input: $\hat{U}_{t-1}, P_{\Omega_t}(x_t), \Omega_t, \hat{\alpha}_t, R_1^{t-1}, \ldots, R_D^{t-1}$. Output: $\tilde{U}_t, R_1^t, \ldots, R_D^t$

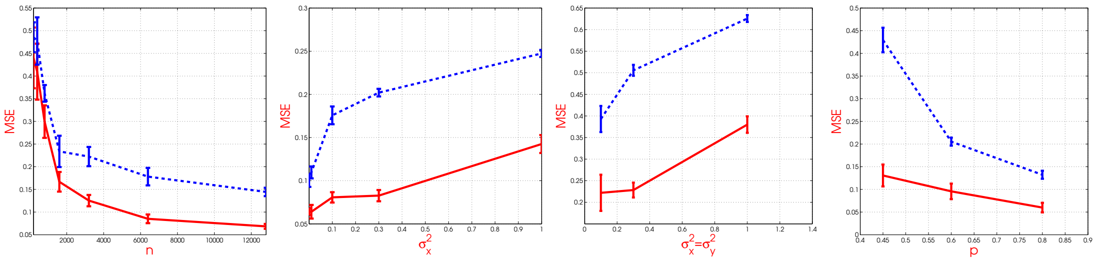
```
for $j = 1, \ldots, D$ do
    $\beta_j^t = 1 + \hat{\alpha}_t^\top (R_j^{t-1})^\dagger \hat{\alpha}_t$
    $v_j^t = (R_j^{t-1})^\dagger \hat{\alpha}_t$
    $p_j^t = \mathbb{1}[j \in \Omega_t]$
    $(R_j^t)^\dagger = (R_j^{t-1})^\dagger - p_j^t (\beta_j^t)^{-1} v_j^t (v_j^t)^\top$
    $\tilde{U}_{t,j} = \hat{U}_{t-1,j} + p_j^t (x_{t,j} - \hat{\alpha}_t^\top \hat{U}_{t-1,j})(R_j^t)^\dagger \alpha_t$
end for
```

Proof Sketch 1. *Our proof uses standard large deviation results connecting* $L(f)$ *and* $\hat{L}(f)$, *similar to (Srebro* et al.*, 2010, Thm. 1)). We upper bound the Rademacher complexity of the function class* $\mathcal{F}_\gamma$*; Lemmas 2 and 3 in the appendix show how to perform these calculations.*

We would like to remark that in Theorem 1 we assumed that $|\Omega_i| = m$ for all i. This assumption is only a technical convenience and allows us to state our result in the cleanest possible way. In general, we wish to choose γ so that both (a) the empirical error $\hat{L}(\hat{f}_{\hat{w},\hat{U},\gamma})$ is small and (b) the R.H.S. of the inequality in Theorem 1 (which scales like $(1-\gamma)^{-2}$) is small. A similar trade-off can also be found in structural risk minimization, commonly studied in classical supervised learning, where we know that functions belonging to a richer class have smaller training error, but potentially larger upper bounds on their generalization error.

Specifically, the R.H.S. of the inequality in Theorem 1 depends on a term of the form $(\frac{DR_1M}{1-\gamma})^2$. This term can be roughly thought of as a measure of the complexity of the function class, $\mathcal{F}_\gamma$. A large γ would imply that we are learning from a richer class of functions and, as can be seen from Theorem 1, the upper bound on the risk of functions in the class $\mathcal{F}_\gamma$ will be potentially larger. Thus we wish to keep γ as small as possible.

On the surface, it may appear that γ must be close to one to yield a small empirical error (by not predicting zero values). However, in many settings, it is possible to choose a small value of γ and still have a low empirical error. To see this, note that larger $m \overset{\text{def}}{=} |\Omega_i|$ leads to easier learning problems, and hence smaller error rates. Let $\hat{S}$ be the subspace spanned by the columns of $\hat{U}$. Let $\mu(\hat{S}) \overset{\text{def}}{=} \frac{D}{d}\max_j ||P_{\hat{S}} e_j||^2$, where $P_{\hat{S}}$ is the projection operator onto $\hat{S}$, and e_j is the standard basis element in D dimensions. $\mu(\hat{S})$ is known as coherence of subspace $\hat{S}$ (Candès and Recht, 2009). It is well known (Balzano *et al.*, 2010a) that $||(\hat{U}_\Omega^\top \hat{U}_\Omega)^{-1}|| \leq \frac{D}{m(1-\gamma_1)}$, with probability at least $1-\delta$ over the random choice of Ω, where $\gamma_1 = \sqrt{\frac{8d\mu(\hat{S})\log(\frac{2d}{\delta})}{3m}}$. Hence, it is enough to set $\gamma \geq \sqrt{\frac{8d\mu(\hat{S})\log(\frac{2d}{\delta})}{3m}}$ From our previous discussions, we know that a small γ would mean that the complexity of $\mathcal{F}_\gamma$ is also small. To see how this affects $\hat{L}(\hat{f}_{\hat{w},\hat{U},\gamma})$, notice that because with probability at least $1-\delta$, $||(\hat{U}_\Omega^\top \hat{U}_\Omega)^{-1}|| \leq \frac{D}{m(1-\gamma)}$, we can claim that on an expectation we are guaranteed to make a non-zero prediction on less than a fraction δ of our training examples. Hence by choosing a large m, we are guaranteed that we can work on a sufficiently small function class $\mathcal{F}_\gamma$, and yet not incur a large training error. This implies from Theorem 1, that $L(\hat{f}_{\hat{w},\hat{U},\gamma})$ is small. Hence, the correct choice of γ depends on m, and a suitable choice of m depends on d. We now have a nice interplay between the number of random measurements, m, the prediction error of the final regressor, the training error of the regressor, the ambient dimension D, and the intrinsic dimension d.



(a) Noiseless

(b) Non-zero $\sigma_x^2; \sigma_y^2 = 0$

(c) Non-zero σ_x^2, σ_y^2

(d) Impact of p, probability of feature being observed (not missing)

Figure 1: Comparison between SLRM (solid, red line) and SMPCR (broken, blue line) on synthetic datasets for $D = 100$ and $d = 30$.

6 Experimental Results

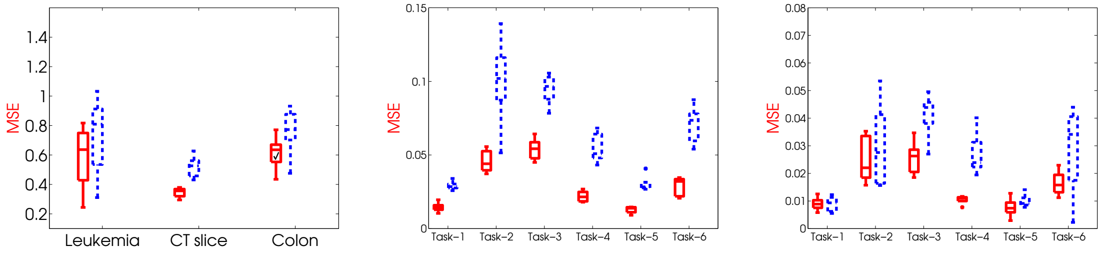
Experimental Setup. We generated datasets of increasing size, with $D = 100$ and $d = 30$. These datasets were generated by first generating a common U matrix of size $D \times d$ with random, orthonormal columns. For a given dataset size, five different A matrices of size $d \times n$ were generated by sampling each entry from a standard normal distribution. Separate validation and test datasets were also generated by generating additional random A matrices, in the same way as described before. We generate random ϵ_X with each entry of matrix ϵ_X (resp. ϵ_Y) having mean zero and variance σ_x^2 (resp. σ_y^2). In order to simulate missing data, we retain each element of each observed feature vector with probability p, and in the test and validation datasets with probability q. While in the theoretical results, for ease of analysis we assumed that the set Ω, is chosen uniformly at random, with replacement, from the set $\{1, 2, \ldots, D\}$, for our practical implementations, we choose Ω of size m by choosing each feature with probability p. Hence, $\mathbb{E}m \approx pD$. Previous analyses have shown that these two sampling strategies behave similarly (Recht, 2011). All the results reported here are averaged over the five different random datasets that we generated. Similarly, w vector used in our model was generated at random from a Gaussian distribution, and random coordinates of w were set to 0. The sparsity level of w was set to 10. γ is set to 0.001. $\lambda_1, \lambda_2, \lambda_3$ are chosen by using a held-out validation set, and searching for parameter values which give the smallest MSE. We found that the performance of SLRM is not very sensitive to the values of λ's, and hence a coarse range is enough during validation. d is set by performing PCA on a subset of the data, and calculating how many dimensions are required to capture about 99% of the variance

We compared our algorithm with a stochastic version of MPCR (PCR modified to handle missing data, detailed in Section 4), which uses the PETRELS algorithm to perform Step 1 of MPCR, and then follows it by a stochastic projected gradient method to solve the LASSO problem in Step 2 of MPCR. We shall call this stochastic implementation SMPCR.

For both SLRM and SMPCR, we allow multiple passes over the dataset in our experiments. The maximum number of passes is fixed to 500 for both SLRM and SMPCR. η_t used in Step 8 of SLRM is chosen to be a constant, ρ, for a fixed number of rounds, and then allowed to decay as ρ/t. This strategy has also been used advocated in (Murata, 1998; Mairal *et al.*, 2012), and we use this method in our algorithm. The value of ρ was found by trying a range of ρ, and choosing the one that gave the best error rate over the hold-out dataset.

Experiments in the noiseless setting. In our first set of experiments, we set $\sigma_x = \sigma_y = 0$. Figure 1(a) shows the error bars for the mean squared error (MSE) on the test dataset for both SLRM (in solid, red line), and SMPCR (in broken, blue line). As we can see from the figure, the performance of both SMPCR and SLRM improves with increasing n. Figure 1(a) also indicates that the average MSE of SLRM is lower than that of SMPCR for all n.

Impact of non-zero σ_x^2. While the above experiments, demonstrate the superior performance of SLRM over SMPCR in the noiseless setting, it does not tell us how these algorithms perform in the presence of noise. We shall now study the impact of non-zero σ_x^2 on the performance of both SLRM and SMPCR. For a clearer understanding, we set $\sigma_y^2 = 0$, and fix the size of the dataset to $n = 12800$. Figure 1(b) shows the impact of increasing σ_x^2 on the MSE



(a) Real datasets

(b) ATP1d

(c) ATP7d

Figure 2: Comparison between SLRM (solid, red boxes) and SMPCR (broken, blue boxes).

of SLRM, and SMPCR. As we can see from this figure, the MSE of both SLRM and SMPCR gradually increases with increasing σ_x^2. Like in the noiseless setting, the MSE of SLRM is always substantially smaller than the SMPCR method.

Impact of non-zero σ_x^2, σ_y^2. We shall now examine the effect of non-zero values for σ_x^2, σ_y^2 on the MSE of SLRM and SMPCR. In these experiments, we fixed the size of the dataset to $n = 12800$, and increased σ_x^2, σ_y^2. For the sake of simplicity, we keep $\sigma_x^2 = \sigma_y^2$. As we can see from Figure 1(c), the MSE of both SLRM and SMPCR increases with increasing σ_x, σ_y. In this case too, the MSE of SLRM is substantially smaller than that of SMPCR. From our plots, SLRM seems to be more noise tolerant than SMPCR.

Impact of increasing p. In this experiment, we examine the impact of increasing p on the error rate of the proposed learning algorithms. We fix $\sigma_x^2 = \sigma_y^2 = 1$, and the size of dataset to $n = 12800$. Like in previous experiments q is set to 0.75. As expected, the error rate of both SLRM, and SMPCR goes down as p increases.

6.1 Experimental Results on Real datasets

We performed experimental comparisons on 15 real world tasks. While an extended discussion of our datasets, has been relegated to the appendix, in this paper, we shall provide a brief description of the datasets. Our datasets are Leukemia, CT Slice, ATP1d, and ATP7d. Both ATP1d, and ATP7d (Groves and Gini, 2011) have six tasks each related to airline ticket price prediction. The CT-slice dataset consists of 384 features obtained from CT scan images and the task is to estimate the relative location of the CT slice on the axial axis of human body. The Leukemia dataset (Golub *et al.*, 1999), and the colon-cancer dataset are high dimensional datasets with $D = 7129$ and $D = 2000$ respectively. Both these datasets have binary labels, $\{-1, +1\}$ as the target but for this paper we treat it as a regression problem.

From Figures 2(a)- 2(c), it is clear that **the median MSE of SLRM is always lesser than the median MSE of SMPCR on all the tasks**. [2] When labels are quantized, such as in classification problems, or noisy, the advantage gained from utilizing label information is limited. This explains why on the leukemia and colon cancer datasets, SLRM might, at times perform worse than SMPCR. As mentioned in Section 4, the superior performance of SLRM over SMPCR, on the CT slice dataset can be explained by the fact that for CT slice dataset, both $D = 384$ and $d = 181$ are large. On both ATP1d and ATP7d datasets, on almost all of the tasks, SLRM far outperforms SMPCR.

7 Conclusions and Future Work

This paper studies the problem of regression with missing data. We proposed a new statistical model and an optimization based approach for learning the parameters of the model. We established risk bounds for our regressor, and demonstrate superior empirical performance over competing algorithms. This work can be extended in several ways. Instead of a single subspace assumption, it should be possible to extend our framework to handle the case when data

[2] The horizontal bar in the barplots show the median MSE of the method

is generated from a union of K subspaces, using ideas in (Xie *et al.*, 2012). Our framework can be extended to handle other tasks using different loss functions and to multi-task learning problems via the use of appropriate matrix norms.

A Towards Proof of Theorem 1

In this appendix we shall prove the result of Theorem 1. In order to establish Theorem 1, we need the following few important definitions and results which have been taken from Srebro *et al.* (2010)

Definition 2. *Let $\sigma_{1:n}$ be a collection of Rademacher random variables. The worst case empirical Rademacher complexity of a function class $\mathcal{F}$ is defined as*

$$\mathcal{R}_n(\mathcal{F}) = \sup_{z_{1:n}} \sup_{f \in \mathcal{F}} \frac{1}{n} \Big| \sum_{i=1}^{n} h(z_i)\sigma_i \Big|. \tag{9}$$

Empiricial Rademacher complexity is defined as

$$\hat{\mathcal{R}}_n(\mathcal{F}) = \sup_{f \in \mathcal{F}} \frac{1}{n} \Big| \sum_{i=1}^{n} h(z_i)\sigma_i \Big| \tag{10}$$

Lemma 1. *Let l be an H smooth non-negative loss, such that $\forall y_1, y_2, y_3, |l(y_1, y_2) - l(y_3, y_2)| \leq b$. Let $L(f) = \mathbb{E}_{z,y} l(f(z), y)$, and $\hat{L}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(z_i), y_i)$. Then for any $\delta > 0$, we have, with probability at least $1 - \delta$, over a random sample of size n, for all $f \in \mathcal{F}$,*

$$L(f) \leq \hat{L}(f) + K \left[\sqrt{\hat{L}(f)} \left(\sqrt{H}\ \log^{1.5} n \mathcal{R}_n(\mathcal{F}) + \sqrt{\frac{b \log(1/\delta)}{n}} \right) + H \log^3 n\, \mathcal{R}_n^2(\mathcal{F}) + \frac{b \log(1/\delta)}{n} \right] \tag{11}$$

The above lemma was proved in Srebro *et al.* (2010), and was used for prediction problems without missing data. We shall use the above result for our problem by using $z = (x, \Omega_x)$. This will enable us to provide generalization bounds for our regression problem with missing data.

Lemma 2. *Srebro* et al. *(2010) For any function class $\mathcal{F}$, containing functions $f : \mathcal{Z} \to \mathbb{R}$, we have that*

$$\hat{\mathcal{R}}_n(\mathcal{F}) = \inf_{\alpha \geq 0} \left[4\alpha + 10 \int_{\alpha}^{\sqrt{\hat{\mathbb{E}} f^2}} \sqrt{\frac{\log \mathcal{N}_2(\epsilon, \mathcal{F}, z_{1:n})}{n}}\, \mathrm{d}\epsilon \right].$$

Lemma 3.

$$\mathcal{R}_n(\mathcal{F}_\gamma) \leq \left(\frac{50m}{n} + \frac{14}{\sqrt{n}} \right) \frac{3DR_1B_X}{(1-\gamma)}.$$

Proof. For the sake of convenience, let $B \overset{\text{def}}{=} \frac{DR_1}{m(1-\gamma)}$. Also throught this document, for the sake of conciseness, we shall use the pair (x, Ω) to mean $(P_\Omega(x), \Omega)$. Similarily (x_i, Ω_i) would mean $(P_{\Omega_i}(x_i), \Omega_i)$ Instead of bounding the Rademacher complexity of $\mathcal{F}$, we shall work with a slightly different function class $\mathcal{F}_g \overset{\text{def}}{=} \{f(x, \Omega) = \beta^T x_\Omega : ||\beta||_2 \leq B\}$. It is now, enough to control the Rademacher complexity of the function class $\mathcal{F}_g$, since all functions $f \in \mathcal{F}_\gamma$, can be written as $f(x, \Omega) = \mu^T x_\Omega$, for an appropriate μ where, by definition of $\mathcal{F}_\gamma$, and using Cauchy-Schwartz inequality, we can guarantee that $||\mu||_2$ is upper bounded by B, and hence $\mathcal{R}_n(\mathcal{F}_\gamma) \leq \mathcal{R}_n(\mathcal{F}_g)$. The rest of the proof upper bounds $\mathcal{R}_n(\mathcal{F}_g)$. We control this Rademacher complexity via lemma 2. Let $f_1, f_2 \in \mathcal{F}_g$.

$$d_2(f_1, f_2, x_{1:n}, \Omega_{1:n}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_1(x_i, \Omega_i) - f_2(x_i, \Omega_i))^2} \tag{12}$$

$$= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\beta_1^T x_{i,\Omega_i} - \beta_2^T x_{i,\Omega_i})^2} \tag{13}$$

$$\le \sqrt{\frac{1}{n}\sum_{i=1}^{n}||\beta_1 - \beta_2||_2^2 ||x_{i,\Omega_i}||_2^2} \tag{14}$$

$$\le \sqrt{m||X||_\infty^2 ||\beta_1 - \beta_2||^2} \tag{15}$$

where, in order to obtain Equation 15 from 14 we used the fact that $||x_{i,\Omega_i}||_2^2 \le \sqrt{m||X||_\infty^2}$ To upper bound the above R.H.S. by ϵ, we need $||\beta_1 - \beta_2|| \le \frac{\epsilon}{\sqrt{m}||X||_\infty}$. Hence, to cover $\mathcal{F}_g$, it is enough to cover an ℓ_2 ball of radius B, with ℓ_2 ball of radius $\frac{\epsilon}{\sqrt{m}||X||_\infty}$. Now, we know that the cover a ball of radius R, with balls of radius ϵ, in d dimensions we need $(\frac{3R}{\epsilon})^d$. Using this result, we can conclude that $\mathcal{N}_2(\mathcal{F}_g, z_{1:n}) \le \left(\frac{3B\sqrt{m}||X||_\infty}{\epsilon}\right)^m$. Plugging, this into the Dudley entropy integral, and using lemma 2, we get

$$\hat{\mathcal{R}}_n(\mathcal{F}_g) \le \min_{\alpha \ge 0} 4\alpha + 10\int_{\alpha}^{\sup_{f\in\mathcal{F}_g}\sqrt{\hat{E}(f^2)}} \sqrt{\frac{m}{n}\log\left(\frac{3DR_1||X||_\infty}{\epsilon}\right)}\,\mathrm{d}\epsilon \tag{16}$$

For the sake of simplicity, let us denote by $F \overset{\text{def}}{=} \sup_{f\in\mathcal{F}_g}\sqrt{\hat{E}(f^2)}$, and by $C \overset{\text{def}}{=} \frac{3DR_1||X||_\infty}{\sqrt{m}(1-\gamma)}$. It is easy to see that $F \le \frac{DR_1||X||_2}{m(1-\gamma)} \le C$. With this notation, the above inequality can be manipulated as follows

$$\hat{\mathcal{R}}_n(\mathcal{F}_g) \le 4\alpha + \frac{10}{\sqrt{n}}\int_{\alpha}^{F}\sqrt{m(\log(C) - \log(\epsilon))}\,\mathrm{d}\epsilon \tag{17}$$

$$\le 4\alpha - \frac{20C\sqrt{m}}{\sqrt{n}}\int_{\sqrt{\log(C)-\log(\alpha)}}^{\sqrt{\log(C)-\log(F)}}\theta^2\exp(-\theta^2)\,\mathrm{d}\epsilon \tag{18}$$

where the above expression is obtained by the change of variable, $\theta^2 = \log(C) - \log(\epsilon)$. Substituting $K_2 = \sqrt{\log(C) - \log(F)}$, for the upper limits of the integral appearing above, we get

$$\hat{\mathcal{R}}_n(\mathcal{F}_g) \le 4\alpha - \frac{20C\sqrt{m}}{\sqrt{n}}\int_{\sqrt{\log(\frac{C}{\alpha})}}^{K_2}\theta^2\exp(-\theta^2)\,\mathrm{d}\theta \tag{19}$$

$$\le 4\alpha - \frac{10\alpha\sqrt{m}}{\sqrt{n}}\sqrt{\log\left(\frac{C}{\alpha}\right)} + 5C\sqrt{\frac{m\pi}{n}}\,\mathrm{erf}\left(\sqrt{\log\left(\frac{C}{\alpha}\right)}\right) + 10C\sqrt{\frac{m}{n}}K_2 e^{-K_2^2} - 5C\sqrt{\frac{m\pi}{n}}\,\mathrm{erf}(K_2)$$

$$\le 4\alpha + 10C\sqrt{\frac{m}{2en}} + 5C\sqrt{\frac{m\pi}{n}}\,\mathrm{erf}\left(\sqrt{\log\left(\frac{C}{\alpha}\right)}\right) \tag{20}$$

where last equation was obtained by using the inequality $xe^{-x^2} \le \frac{1}{\sqrt{2e}}$, and by dropping all the negative terms. We can now optimize over α, by setting the gradient to 0, to get

$$\alpha^* = C\exp\left(-\frac{4n + 25m + \sqrt{16n^2 + 200mn}}{50m}\right) \tag{21}$$

Substituting α^* for α in Equation 20, and over-estimating $\mathrm{erf}(\cdot)$ by 1, we get

$$\hat{\mathcal{R}}_n(\mathcal{F}_g) \leq 10C\sqrt{\frac{m}{2en}} + 5C\sqrt{\frac{m\pi}{n}} + \frac{4C}{\sqrt{e}}e^{-\frac{0.16n}{m}} \tag{22}$$

Replacing C, by its definition, we get

$$\hat{\mathcal{R}}_n(\mathcal{F}_g) \leq \left(\frac{10}{\sqrt{2en}} + \frac{5\sqrt{\pi}}{\sqrt{n}} + 4e^{-\frac{0.16n}{m}}\right)\frac{3DR_1||X||_\infty}{1-\gamma} \leq \frac{45DR_1||X||_\infty}{(1-\gamma)\sqrt{n}} \tag{23}$$

Since the above quantity is independent of the sample, hence the above bound on $\mathcal{R}_n(\mathcal{F}_g)$ also holds for $\mathcal{R}(\mathcal{F}_g)$, i.e.

$$\mathcal{R}_n(\mathcal{F}_g) \leq \left(\frac{10}{\sqrt{2en}} + \frac{5\sqrt{\pi}}{\sqrt{n}} + 4e^{-\frac{0.16n}{m}}\right)\frac{3DR_1||X||_\infty}{1-\gamma} \leq \left(\frac{50m}{n} + \frac{14}{\sqrt{n}}\right)\frac{3DR_1B_X}{(1-\gamma)}. \tag{24}$$

□

Proof of Theorem 1. The theorem now follows immediately from Lemma 1, the squared loss, which is by definition 2-smooth and Lemma 3. The role of z is played by the random pair (x, Ω). The following trivial bound for b, required in Theorem 1 holds

$$b \leq |l(y_1, y_2) - l(y_1, y_3)| \leq |l(y_1, y_2)| + |l(y_1, y_3)| \leq 2\left(B_Y + \frac{DR_1}{m(1-\gamma)}\right)^2 \tag{25}$$

B Description of datasets

In this appendix we provide information regarding the datasets that were used in our experiments in Section 6. We shall provide a description of the datasets that were used in Section 6.

1. **Leukamia dataset.** The Leukamia dataset was obtained from `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. It is a cancer classification dataset Golub *et al.* (1999) where the features are gene expression levels, and our task is to classify whether the gene expression levels indicate acute myeloid leukemia or acute lymphoblastic leukemia. This dataset comes with separate train and test datasets of a total of 72 points with $D = 7129$. For our experiments we merge both the train and test datasets, and randomly sample 30 data points for training, 22 data points for testing, and the remaining for validation. This was repeated five times to obtain five different training, test and validation datasets. On each of the training sets, we retained a feature with probability 0.1, to simulate the missing data scenario.

2. **CT slice.** The CT slice dataset was obtained from `https://archive.ics.uci.edu/ml/datasets/Relative+location+of+CT+slices+on+axial+axis`. The dataset consists of 384 features obtained from CT scan images. The task is to estimate the relative location of CT slice on the axial axis of human body. The original dataset consisted of 53500 data points. For our experiments we sampled 300 data points uniformly at random, of which 100 each were used for training, testing, and validation. This process was repeated five times, to obtain five different datasets.

3. **ATP1d, ATP7d.** The ATP1d, ATP7d datasets were obtained from `http://mulan.sourceforge.net/datasets-mtr.html`. The ATP1d and ATP7d datasets are airline ticket price prediction datasets. where the problem is to predict the prices for six target flight preferences, namely the price of any non-stop flight, Delta airlines, Continental airlines, Airtran, and United airlines. While for the ATP1d dataset these target prices are the next day price, for ATP7d the targets are the minimum price observed over the next 7 days. The input features for each sample are values that are useful for prediction of the airline ticket prices for a specific observation date-departure date pair. The features include quantities like day-of-the-week of the observation date, number of days between observation date and departure, and several other price related features such as minimum quoted price, mean quoted price etc...In order to normalize our features, we divided all our price related features by 1000,

and the feature which measures the number of days between observation date and the day of departure by 180. See Spyromitros-Xioufis *et al.* (2012); Groves and Gini (2011) for more details on these datasets. We converted the cardinal day-of-the-week feature into a 7 dimensional boolean vector. The sizes of our training, testing, and validation datasets are 200,46,50 respectively. These were obtained via random sampling from the original dataset.

References

Abed-Meraim, K., Chkeif, A., and Hua, Y. (2000). Fast orthonormal past algorithm. *Signal Processing Letters, IEEE*, **7**(3), 60–62.

Balzano, L., Recht, B., and Nowak, R. (2010a). High-dimensional matched subspace detection when data are missing. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pages 1638–1642. IEEE.

Balzano, L., Nowak, R., and Recht, B. (2010b). Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 704–711. IEEE.

Candès, E. J. and Recht, B. (2009). Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, **9**(6), 717–772.

Chi, Y., Eldar, Y. C., and Calderbank, R. (2012). Petrels: Subspace estimation and tracking from partial observations. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3301–3304. IEEE.

Fukumizu, K., Bach, F. R., Jordan, M. I., *et al.* (2009). Kernel dimension reduction in regression. *The Annals of Statistics*, **37**(4), 1871–1905.

Goldberg, A., Recht, B., Xu, J., Nowak, R., and Zhu, X. (2010). Transduction with matrix completion: Three birds with one stone. In *Advances in neural information processing systems*, pages 757–765.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., *et al.* (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, **286**(5439), 531–537.

Groves, W. and Gini, M. (2011). A regression model for predicting optimal purchase timing for airline tickets. Technical report, Technical Report 11-025, University of Minnesota, Minneapolis, MN.

Hardt, M. (2013). Understanding alternating minimization for matrix completion. *arXiv preprint arXiv:1312.0925*.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2003). *The Elements of Statistical Learning*. Springer.

Hua, Y., Xiang, Y., Chen, T., Abed-Meraim, K., and Miao, Y. (1999). A new look at the power method for fast subspace tracking. *Digital Signal Processing*, **9**(4), 297–314.

Jain, P., Netrapalli, P., and Sanghavi, S. (2013). Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 665–674. ACM.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, **42**(8), 30–37.

Krim, H., Viberg, M., *et al.* (1995). Sensor array signal processing: two decades later.

Loh, P.-L. and Wainwright, M. J. (2011). High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. In *Advances in Neural Information Processing Systems*, pages 2726–2734.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009). Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM.

Mairal, J., Bach, F., and Ponce, J. (2012). Task-driven dictionary learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **34**(4), 791–804.

Maurer, A. and Pontil, M. (2010). K-dimensional coding schemes in hilbert spaces. *Information Theory, IEEE Transactions on*, **56**(11), 5839–5846.

Murata, N. (1998). A statistical study of on-line learning. *Online Learning and Neural Networks. Cambridge University Press, Cambridge, UK*.

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, **15**(3), 267–273.

Recht, B. (2011). A simpler approach to matrix completion. *The Journal of Machine Learning Research*, **12**, 3413–3430.

Roy, R. and Kailath, T. (1989). Esprit-estimation of signal parameters via rotational invariance techniques. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, **37**(7), 984–995.

Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., and Vlahavas, I. (2012). Multi-label classification methods for multi-target regression.

Srebro, N., Sridharan, K., and Tewari, A. (2010). Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems*, pages 2199–2207.

Suzuki, T. and Sugiyama, M. (2013). Sufficient dimension reduction via squared-loss mutual information estimation. *Neural computation*, **25**(3), 725–758.

Szlam, A. and Sapiro, G. (2009). Discriminative k-metrics. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1009–1016. ACM.

Tuncer, T. E. and Friedlander, B. (2009). *Classical and modern direction-of-arrival estimation*. Academic Press.

Vainsencher, D., Mannor, S., and Bruckstein, A. M. (2011). The sample complexity of dictionary learning. *The Journal of Machine Learning Research*, **12**, 3259–3281.

Xie, Y., Huang, J., and Willett, R. (2012). Multiscale online tracking of manifolds. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 620–623. IEEE.

Yang, B. (1995). Projection approximation subspace tracking. *Signal Processing, IEEE Transactions on*, **43**(1), 95–107.