# PULasso: High-dimensional variable selection with presence-only data

Hyebin Song
Department of Statistics, University of Wisconsin-Madison
and
Garvesh Raskutti [*]
Department of Statistics, University of Wisconsin-Madison

November 22, 2017

**Abstract**

In various real-world problems, we are presented with *positive and unlabelled data*, referred to as presence-only responses and where the number of covariates $p$ is large. The combination of *presence-only responses* and *high dimensionality* presents both statistical and computational challenges. In this paper, we develop the *PUlasso* algorithm for variable selection and classification with positive and unlabelled responses. Our algorithm involves using the majorization-minimization (MM) framework which is a generalization of the well-known expectation-maximization (EM) algorithm. In particular to make our algorithm scalable, we provide two computational speed-ups to the standard EM algorithm. We provide a theoretical guarantee where we first show that our algorithm is guaranteed to converge to a stationary point, and then prove that any stationary point achieves the minimax optimal mean-squared error of $\frac{s \log p}{n}$, where $s$ is the sparsity of the true parameter. We also demonstrate through simulations that our algorithm out-performs state-of-the-art algorithms in the moderate $p$ settings in terms of classification performance. Finally, we demonstrate that our PUlasso algorithm performs well on a biochemistry example.

*Keywords:* PU-learning, majorization-minimization, non-convexity, regularization.

[*] Both HS and GR were partially supported by NSF-DMS 1407028. GR was also partially supported by ARO W911NF-17-1-0357 and NGA HM0476-17-1-2003.

# 1 Introduction

In many classification problems, we are presented with the problem where it is either prohibitively expensive or impossible to obtain negative responses and we only have positive and unlabelled *presence-only* responses (see e.g. Ward et al. [2009]). For example, presence-only data is prevalent in geographic species distribution modeling in ecology where presences of species in specific locations are easily observed but absences are difficult to track (see e.g. Ward et al. [2009]), text mining (see e.g. Liu et al. [2003]), bioinformatics (see e.g. Elkan and Noto [2008]) and many other settings. Classification with presence-only data is sometimes referred to as PU-learning(learning with positive and unlabelled responses)( Liu et al. [2003], Elkan and Noto [2008] ).

## 1.1 Motivating application: Biotechnology

Although the theory and methodology we develop applies generally, a concrete application that motivated this work arises from biological systems engineering. In particular recent high-throughput technologies generate millions of biological sequences from a library for a protein or enzyme of interest (see e.g. Fowler and Fields [2014], Hietpas et al. [2011]). In Section 5 the enzyme of interest is beta-glucosidase (BGL) which is used to decompose of di-saccharydes into glucose which is an important step in the process of converting plants to bio-fuels (Romero et al. [2015]). The performance of the BGL enzyme is measured by the concentration of glucose that is produced and a positive response arises when the di-saccharyde is decomposed to glucose and a negative response arises otherwise. Hence there are two scientific goals, firstly to determine how the sequence structure influences the biochemical functionality and secondly, using this relationship to engineer and design BGL sequences with improved functionality. Presence-only responses arise naturally in this setting because the high-throguhput technologies used produce functional (positive) sequences and unlabelled sequences (see e.g. Romero et al. [2015] for more details).

Given these two scientific goals, we are interested in both the *variable selection* and *classification* problem since we want to determine which positions in the sequence most influence positive responses as well as classify which protein sequences would be functional. Furthermore, the number of variables here is large since we need to model long and complex biological sequences. Hence our variable selection problem is *high-dimensional.* In Section 5 we present this application in greater detail and demonstrate the success of our algorithm.

## 1.2 Problem Setup

To state the problem formally, we have two sets of samples. The first set consists of $n_\ell$ *positive and labelled* observations, and a second set that contains $n_u$ observations randomly drawn from the population with only covariates not the responses being observed. Furthermore, we assume the true positive prevalence $\pi := P(y = 1)$ is known. In our biotechnology example, $\pi$ can be estimated extremely precisely using an alternative experiment. If the information of true prevalence is unavailable, it may be estimated and in fact several algorithms recently emerged in the literature (see e.g. Blanchard et al. [2010], Du Marthinus et al. [2015], Jain et al. [2016]). For the $i^{th}$ sample point, $z_i = 1$ if the subject is labelled and it clearly follows that $y_i = 1$ since all labelled samples are positive. Otherwise, we let $z_i = 0$ which indicates that the sample is unlabelled meaning $y_i = 0$ or $y_i = 1$. We use $x_i \in \mathbb{R}^p$ for the associated covariates within each sample. To incorporate the covariates, we model the relationship between the probability of a response being positive and $(x, \theta)$ using the standard logistic regression model:

$$P(y = 1|x; \theta) = \frac{e^{x^T\theta}}{1 + e^{x^T\theta}} \tag{1}$$

where $\theta \in \mathbb{R}^p$ where $x$ refer to the covariates and $\theta$ refer to the unknown parameters. Since one of our goals is variable selection, we want to estimate $\theta$ given samples $(x_i, z_i)_{i=1}^n$, and $(y_i)_{i=1}^n$ are unobserved variables. In the biological sequence engineering examples $(x_i)_{i=1}^n$ correspond to binary covariates of biological sequences. In the BGL example, for each of the $d$ positions, there are $M$ possible categories or aminos acids, so covariates correspond to the indicator of an amino acid appearing in a given position $p = O(dM)$ , as well as pairs of amino acids $p = O(d^2M^2)$, and so on. Here $d = O(1000)$ and $M \approx 20$ making the problem high-dimensional.

From a computational perspective, high-dimensional PU-learning presents computational challenges since the standard logistic regression objective leads to a non-convex likelihood when we have positive and unlabelled data. To address this challenge, we build on the expectation-maximization (EM) procedure developed in Ward et al. [2009] and provide two computational speed-ups. In particular we introduce the *PUlasso* for high-dimensional variable selection with positive and unlabelled data. Prior work that involves the EM algorithm in the low-dimensional setting in Ward et al. [2009] involves solving a logistic regression model at the M-step. To adapt to the high-dimensional setting and make the problem scalable, we include an $\ell_1$-sparsity or $\ell_1/\ell_2$-group sparsity penalty and provide two speed-ups. Firstly we use a quadratic majorizer of the logistic regression objective, and secondly we use techniques in linear algebra to exploit sparsity of the design matrix $X$ which commonly arises in the applications we are dealing with. Our PUlasso algorithm fits into the majorization-minimization (MM) framework (see e.g. Lange et al. [2000], Ortega and Rheinboldt [2000]) for which the EM algorithm is a special case.

## 1.3 Our contributions

In this paper, we make the following major contributions:

- Develop the PUlasso algorithm for doing variable selection and classification with presence-only data. In particular we build on the existing EM algorithm developed in Ward et al. [2009] and add two computational speed-ups, quadratic majorization and exploiting sparse matrices. These two speed-ups improve speed by several orders of magnitude and allows our algorithm to scale to datasets with millions of samples and covariates.

- Provide theoretical guarantees for our algorithm. First we show that our algorithm is guaranteed to converge to a stationary point of the non-convex objective, and then show that any stationary point achieves the minimax optimal mean-squared error of $\frac{s \log p}{n}$. To provide statistical guarantees we extend the exisiting results of generalized linear model with canonical link (Negahban et al. [2012], Loh and Wainwright [2013]) to non-canonical link and show optimality of stationary points of non-convex objectives in high-dimensional statistics. To the best of our knowledge the PUlasso is the first algorithm where PU-learning is provably optimal in the high-dimensional setting.

- Demonstrate through a simulation study that our algorithm performs well in terms of classification compared to state-of-the-art PU-learning methods in Du Marthinus et al. [2015], Elkan and Noto [2008], Liu et al. [2003], both for low-dimensional and high-dimensional problems.

- Demonstrate that our PUlasso algorithm allows us to develop improved protein-engineering approaches. In particular we apply our PUlasso algorithm to sequences of BGL (beta-glucosidase) enzymes to determine which sequences are functional. We demonstrate that sequences selected by our algorithm have statistically significant mutations and we also provide a scientific experiment which shows that the variables selected lead to BGL proteins that are engineered with improved functionality.

The remainder of the paper is organized as follows: Section 2 we provide the background and introduce the PUlasso algorithm, including our two computational speed-ups and provide algorithmic guarantees that our algorithm converges to a stationary point; in Section 3 we provide statistical mean-squared error guarantees for our PUlasso algorithm that achieve the minimax rate; Section 4 provides a comparison in terms of classification performance of our PUlasso algorithm to state-of-the-art PU-learning algorithms; finally in Section 5, we apply our PUlasso algorithm to the BGL data application and provide both a statistical validation and simple scientific validation for our selected variables.

# 2 PUlasso Algorithm

In this section, we introduce our PUlasso algorithm. First, we discuss the prior EM algorithm approach developed in Ward et al. [2009] and apply a simple regularization scheme. We then discuss our two computational speed-ups, the quadratic majorization for the M-step and exploiting sparse matrices. We prove that our algorithm has the descending property and is guaranteed to converge to a stationary point, and show that our two speed-ups increase speed by several orders of magnitude.

## 2.1 Prior approach: EM algorithm with regularization

First we use the prior result in Ward et al. [2009] to determine the observed log-likelihood (in terms of the $z_i$'s) and the full log-likelihood (in terms of the unobserved $y_i$'s and $z_i$'s). An important underlying assumption in our setup and the following lemma is that the unlabeled samples are assumed to be randomly drawn from the original population. This means we are in the case-control sampling scheme (Lancaster and Imbens [1993], Ward et al. [2009]) as opposed to the single-training sampling scheme (Elkan and Noto [2008]) for which the positive and unlabeled samples together are assumed to be a random sample from the joint population of $(x, y, z)$. For biotechnology application the case-control setting is appropriate since the high-throughput technology leads to the unlabelled samples being drawn from the original population (see Romero et al. [2015] for details). The following lemma, derived in Ward et al. [2009], gives the form of the observed and the full log-likelihood which is based on the conditional probabilities given the covariates and the case-control sampling scheme.

**Lemma 2.1** ( Ward et al. [2009])**.** *The observed log-likelihood* $\log L(\theta; x, z)$ *for our presence-only model in terms of* $(x_i, z_i)_{i=1}^n$ *is:*

$$\log L(\theta; x, z) = \log \left( \prod_i P_\theta(z_i|x_i) \right) \tag{2}$$

$$= \sum_{i=1}^n \left( z_i \left( \log \frac{n_\ell}{\pi n_u} + x_i^T\theta - \log(1 + e^{x_i^T\theta}) \right) - \log \left( 1 + e^{\log \frac{n_\ell}{\pi n_u} + x_i^T\theta - \log(1+e^{x_i^T\theta})} \right) \right). \tag{3}$$

*The full log-likelihood* $\log L_f(\theta; x, y, z)$ *in terms of* $(x_i, y_i, z_i)_{i=1}^n$ *is*

$$\log L_f(\theta; x, y, z) = \sum_{i=1}^n \log P_\theta(y_i, z_i|x_i) \tag{4}$$

$$\propto \sum_{i=1}^n [y_i(x_i^T\theta + \log \frac{n_\ell + \pi n_u}{\pi n_u}) - \log(1 + \exp(x_i^T\theta + \log \frac{n_\ell + \pi n_u}{\pi n_u}))] \tag{5}$$

*where* $n_\ell, n_u$ *as the number of positive and unlabelled observations and* $\pi = P_{\theta^*}(y = 1)$ *where* $\theta^*$ *is the true parameter.*

In the setting where $p$ is large, we add a regularization term, which means our overall optimization problem we are solving is:

$$\hat{\theta} \in \operatorname*{argmin}_{\theta \in \Theta_0} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log L(\theta; x_i, z_i) + P_\lambda(\theta) \right\} \tag{6}$$

where $\log L(\theta; x_i, z_i)$ is the observed log-likelihood and $P_\lambda : \mathbb{R}^p \to \mathbb{R}$ is a regularizer. The penalty function is indexed by a regularization parameter $\lambda > 0$. $\Theta_0 \subset \Theta$ is a subspace of $\Theta$. For the penalty $P_\lambda(\theta)$ we consider a group lasso penalty with $J$ groups and and an unpenalized intercept where $(K_0, K_1, ..., K_J)$ form a partition of $\{0, 1, \ldots, p-1\}$.

$$P_\lambda(\theta) = \lambda \sum_{j=1}^{J} \sqrt{|K_j|} \|\theta_{K_j}\|_2, \theta_{K_j} \in \mathbb{R}^{|K_j|} \tag{7}$$

We note that lasso is a special case when $J = p-1$ and $K_j = \{j\}$. In the original proposal by Yuan and Lin [2006], groups $X_j$ are assumed to be orthonormal. If group matrices are not orthonormal, however, it is unclear whether we should orthonormalize group matrices prior to application of the group lasso. This question was addressed in Simon and Tibshirani [2012], and the authors provide a compelling argument that prior orthonormalization has both theoretical and computational advantages. In particular, Simon and Tibshirani [2012] demonstrated that the following orthonormalization procedure is intimately connected with the uniformly most powerful invariant testing for inclusion of a group. To describe this orthonormalization explicitly, we obtain standardized group matrices $Q_j$ and scale matrices $R_j$ using the QR-decomposition such that

$$P_0 X_j = Q_j R_j \text{ and } Q_j^T Q_j = n I_{|K_j| \times |K_j|} \tag{8}$$

where $P_0 = (I_{n \times n} - \frac{\mathbb{1}_n \mathbb{1}_n^T}{n})$ is the projection matrix into the orthogonal space of $\mathbb{1}_n$. Letting $Q := [\mathbb{1}_n, Q_1, \ldots, Q_J] = [q_1^T, \ldots, q_n^T]^T$, the original optimization problem (6) can be expressed in terms of $q_i$'s becomes:

$$\operatorname*{argmin}_{\nu} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log L(\nu; q_i, z_i) + \lambda \sum_{j=1}^{J} \sqrt{|K_j|} \|\nu_j\|_2 \right\} \tag{9}$$

where we use the transformation $\theta$ to $\nu$:

$$\theta_j = \begin{cases} \nu_0 - \sum_{j=1}^{J} P_0 X_j R_j^{-1} \nu_j & j = 0 \\ R_j^{-1} \nu_j & j \geq 1 \end{cases} \tag{10}$$

For notational convenience we denote the overall objective as $\mathcal{L}(\theta)$ as

$$\mathcal{L}(\theta) := -\frac{1}{n} \sum_{i=1}^{n} \log L(\theta; x_i, z_i) + P_\lambda(\theta) \tag{11}$$

and our goal is to minimize $\mathcal{L}(\theta)$. A standard approach to performing this minimization is to use the EM-algorithm approach developed in Ward et al. [2009]. In particular we treat $(y_i)_{i=1}^{n_u}$ as hidden variables, estimate these in E-step and then use the full log-likelihood $\log L_f(\theta; x, y, z)$ in the M-step. Let $\theta_{null} = [\log \frac{\pi}{1-\pi}, 0, 0, ..., 0]^T$ denote the $\theta$ corresponding to the intercept-only model.

**Algorithm 1:** Regularized EM algorithm for the optimization problem (6)

```
Input: an initialization $\theta^0$ such that $\mathcal{L}(\theta^0) \leq \mathcal{L}(\theta_{null})$
for $m$=0,1,2,..., do
    while converged do
        • E-step : estimate $y_i$ at $\theta = \theta^m$ by
              $\hat{y}_i(\theta^m) = \left( \frac{e^{x_i^T \theta^m}}{1 + e^{x_i^T \theta^m}} \right)^{1-z_i}$        (12)
        • M-step : obtain $\theta^{m+1}$ by
              $\theta^{m+1} = \underset{\theta}{\operatorname{argmin}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \hat{y}_i(\theta^m) \left( x_i^T \theta + b \right) + \log \left( 1 + e^{x_i^T \theta + b} \right) + P_\lambda(\theta) \right\}$        (13)
          where $b := \log \frac{n_\ell + \pi n_u}{\pi n_u}$
    end
end
```

The E-step follows since $E_{\theta^m}[y_i|z, x, s = 1] = \left(\frac{e^{x_i^T\theta^m}}{1+e^{x_i^T\theta^m}}\right)^{1-z_i}$ since $z_i = 1$ implies $y_i = 1$ and when $z_i = 0$, observations in the unlabelled data are random draws from the population. An initialization $\theta^0$ can be any $R^p$ vector such that $\mathcal{L}(\theta^0) \leq \mathcal{L}(\theta_{null})$ where $\theta_{null}$ is the parameter corresponding to the intercept-only model. The condition $\mathcal{L}(\theta^0) \leq \mathcal{L}(\theta_{null})$ is considered to prevent the algorithm from starting where the objective function value is worse that that of the null model. The condition is not restrictive, as we expect any reasonable estimate of $\theta$ would perform as well as the null model. If we are provided with no additional information, we may use $\theta_{null}$ for the initialization. We use $\theta_0 = \theta_{null}$ as the initialization for the remainder of the paper. For the M-step we may use a regularized logistic regression solver such as the **glmnet** R package. We discuss a computationally more efficient way of solving (13) in the subsequent section.

## 2.2 PUlasso : A Quadratic Majorization for the M-step

Now we develop our PUlasso algorithm which is a faster algorithm for solving (6) by using quadratic majorization for the M-step. The main computational bottleneck in algorithm 1 is the M-step which requires minimizing a regularized logistic regression loss at each step. However, the most impotant property of the M-step is that it is a majorizer to the likelihood which ensures the descending property (see e.g. Lange et al. [2000]). Hence we use a computationally faster quadratic majorizer of the logistic loss function. Our approach is an exmple of the more general majorization-minimization (MM) algorithm (see e.g. Lange et al. [2000], Ortega and Rheinboldt [2000]) which also has the descending property.

More concretely, using the MM framework we construct the set of functions $\overline{Q}(\theta; \theta^m)$ having following two properties

$$\overline{Q}(\theta^m; \theta^m) = Q(\theta^m; \theta^m), \quad \overline{Q}(\theta; \theta^m) \leq Q(\theta; \theta^m), \forall\theta \tag{14}$$

where we define $Q$ as $Q(\theta;\theta^m) := n^{-1}E_{\theta^m}[logL_c(\theta)|z,x,s=1]$.

Using a Taylor expansion of $Q(\theta;\theta^m)$ at $\theta=\theta^m$, we obtain $Q(\theta;\theta^m)$

$$
\begin{aligned}
&= Q(\theta^m;\theta^m) + \frac{1}{n}[X^T(\hat{y}_i(\theta^m) - \mu^*(\theta^m))]^T\Delta_m - \frac{1}{2n}\int_0^1 \Delta_m^T X^T W(\theta + s(\theta-\theta^m))X\Delta_m ds \\
&\geq Q(\theta^m;\theta^m) + \frac{1}{n}(\hat{y}_i(\theta^m) - \mu^*(\theta^m))^T X\Delta_m - \frac{1}{8n}\Delta_m^T X^T X\Delta_m
\end{aligned}
$$

where $\Delta_m := \theta - \theta^m, \mu^*(\theta^m) := [\mu^*(\theta^m)_1, \ldots, \mu^*(\theta^m)_n], \mu^*(\theta^m)_i := \dfrac{e^{x_i^T\theta^m + b}}{1+e^{x_i^T\theta^m+b}}$, and $W$ is a diagnoal matrix with $[W^*(\theta)]_{ii} := \mu^*(\theta)_i(1-\mu^*(\theta)_i)$ . The inequality follows from $W(\theta) \prec \frac{1}{4}I_{n\times n}$, $\forall\, \theta$. Thus setting $\overline{Q}$ as follows:

$$\overline{Q}(\theta;\theta^m) := Q(\theta^m;\theta^m) + \frac{1}{n}(\hat{y}_i(\theta^m) - \mu^*(\theta^m))^T(X\theta - X\theta^m) - \frac{1}{8n}(\theta-\theta^m)^T X^T X(\theta - \theta^m),$$

it naturally follows that $\overline{Q}$ satisfies both conditions in (14). Also with some algebra, it follows that

$$\overline{Q}(\theta;\theta^m) = -\frac{1}{8n}(4(\hat{y}(\theta^m) - \mu^*(\theta^m)) + X\theta^m - X\theta)^T(4(\hat{y}(\theta^m) - \mu^*(\theta^m)) + X\theta^m - X\theta) + c(\theta^m)$$

for some $c(\theta^m)$ which does not depend on $\theta$. Hence $-\overline{Q}$ acts as a quadratic majorizers of $-Q$ which replaces our M-step for the original EM algorithm. Hence our PUlasso algorithm can be represented as follows in algorithm 2.

Now we state the following proposition to show that both the regularized EM and PUlasso algorithms have the desirable descending property and are guaranteed to conver to a stationary point. We let $\mathcal{S}$ be the set of stationary points which satisfy the first order optimality condition, i.e.,

$$\mathcal{S} := \{\theta'; \nabla\mathcal{L}(\theta')^T(\theta - \theta') \geq 0\},\ \forall\, \theta \in \Theta_0. \tag{15}$$

**Algorithm 2:** PUlasso : QM-EM algorithm for the optimization problem (6)

```
Input: an initialization $\widetilde{\theta}^0$ such that $\mathcal{L}(\widetilde{\theta}^0) \leq \mathcal{L}(\theta_{null})$
for $m$=0,1,2,..., do
    while converged do
        • E-step : estimate $y_i$ at $\theta = \widetilde{\theta}^m$ by
              $\hat{y}_i(\widetilde{\theta}^m) = \left( \frac{e^{x_i^T \widetilde{\theta}^m}}{1 + e^{x_i^T \widetilde{\theta}^m}} \right)^{1-z_i}$    (16)
        • QM-EM step : obtain $\widetilde{\theta}^{m+1}$ by
            1. create a working response vector $u(\widetilde{\theta}^m)$ at $\theta = \widetilde{\theta}^m$
                  $u(\widetilde{\theta}^m) := 4(\hat{y}(\widetilde{\theta}^m) - \mu^*(\widetilde{\theta}^m)) + X\widetilde{\theta}^m$    (17)
            2. solve a quadratic loss problem with a penalty
                  $\widetilde{\theta}^{m+1} = \underset{\theta}{\operatorname{argmin}} \left\{ \frac{1}{2n} (u(\widetilde{\theta}^m) - X\theta)^T (u(\widetilde{\theta}^m) - X\theta) + 4P_\lambda(\theta) \right\}$    (18)
    end
end
```

**Proposition 2.1.** *The sequence of estimators* $\{\theta^m\}$ *obtained by Algorithms* (1) *or* (2) *satisfy*

*(i)* $\mathcal{L}(\theta^m) \geq \mathcal{L}(\theta^{m+1})$ *where* $\mathcal{L}$ *defined in* (11)*.*

*(ii)* $\mathcal{L}(\theta^m) > \mathcal{L}(\theta^{m+1})$ *if* $\theta^m \notin \mathcal{S}$*.*

*(iii)* $\mathcal{L}(\theta^m)$ *converges monotonically to* $\mathcal{L}(\theta_*)$ *for some* $\theta_* \in \mathcal{S}$*.*

Proposition 2.1 shows that both the regularized EM algortihm and our PUlasso algorithm are guaranteed to converge to a stationary point of the objective (11). The proof uses the standard arguments based on Jensen's inequality, convergence of EM algorithm and MM algorithms and is deferred to the supplementary material. In Fig. 1 we provide an example of iterations of the PUlasso algorithm based on our logistic regression model. $\lambda$ was chosen by 10-fold cross validation, and at such lambda level ($\lambda = 0.00750$), model was refitted starting from the intercept only model. We see that the algorithm finds a true support, and as iteration goes we obtain latent responses closer to their true probabilities.

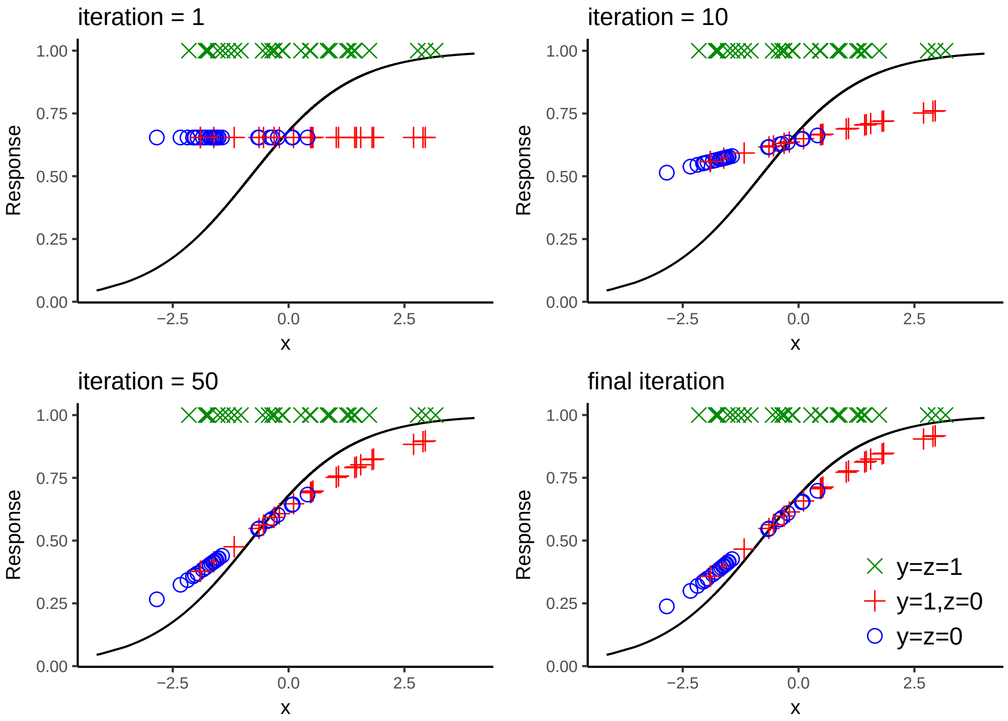

Figure 1: A simple illustration of iterations of the PUlasso algorithm when $n_\ell = n_u = 10000, p = 10, \pi = 0.654$, and one active variable. The $x$-axis represents the active variable, and $y$- axis represents estimated latent responses at each iteration. The solid curve represents true probabilities and the samples are represented at the specidied iterations in the plot.

### 2.2.1 Block Coordinate Descent Algorithm for M-step and Sparse Calculation

In this section, we discuss the specifics of finding a minimizer for the M-step (18) for each iteration of our PUlasso algorithm. After pre-processing the design matrix as described in (9), (10), we solve the following optimization problem using a standard block-wise coordinate descent algorithm (see e.g. Simon and Tibshirani [2012], Breheny and Huang [2013]).

$$\underset{\nu}{\text{argmin}} \left\{ \frac{1}{2n} \|u - Q\nu\|^2 + 4\lambda \sum_{j=1}^{J} \sqrt{|K_j|} \|\nu_{K_j}\|_2 \right\} \tag{19}$$

**Algorithm 3:** Fitting (19) using Block Coordinate Descent

```
Given initial parameter $\nu = [\nu_0, \ldots, \nu_J]^T$, $r = u - \sum_{j=1}^{p} Q_j \nu_j$
$\nu_0' \leftarrow \mathbb{1}_n^T u + \nu_0$
$r' \leftarrow r + \mathbb{1}_n(\nu_0 - \nu_0')$
$r \leftarrow r', \nu_0 \leftarrow \nu_0'$
while *converged* do
    for *j=1,...,J* do
        $z_j = n^{-1} Q_j^T r + \nu_j$    (20)
        $\nu_j' \leftarrow S(z_j, 4\lambda\sqrt{|K_j|})$    (21)
        $r' \leftarrow r + Q_j(\nu_j - \nu_j')$    (22)
        $r \leftarrow r', \nu_j \leftarrow \nu_j'$    (23)
    end
end
```

$S(., \lambda)$ is the soft thresholding operator defined as follows:

$$S(z, \lambda) := \begin{cases} (\|z\|_2 - \lambda)\dfrac{z}{\|z\|_2} & \text{if } \|z\|_2 > \lambda \\ 0 & \text{otherwise.} \end{cases}$$

Note that we do not need to update the intercept since $Q_j$ are orthogonal to $\mathbb{1}_n$. For more details, see for e.g. Breheny and Huang [2013].

For our biochemistry example and many other examples, $X$ is a sparse matrix. In Algorithm 3, we do not exploit this sparsity since $Q$ will not be sparse even when $X$ is sparse. If we want to exploit sparse $X$ we use the following algorithm.

**Algorithm 4:** Fitting (19) and exploiting sparse X

```
Given initial parameter $\nu = [\nu_0, \ldots, \nu_J]^T$, $r = u - P_0(\sum_{j=1}^{p} X_j R_j^{-1} \nu_j)$
$\nu_0' \leftarrow \mathbb{1}_n^T u + \nu_0$
$r' \leftarrow r + \mathbb{1}_n(\nu_0 - \nu_0')$
$\widetilde{r} \leftarrow r', \nu_0 \leftarrow \nu_0'$
while converged do
    for j=1,...,J do
        $z_j = n^{-1} R_j^{-1} X_j^T \widetilde{r} - R_j^{-1} \left( \frac{X_j^T 1_n}{n} \right) \left( \frac{\mathbb{1}_n^T \widetilde{r}}{n} \right) + \nu_j$   (24)
        $\nu_j' \leftarrow S(z_j, 4\lambda\sqrt{|K_j|})$   (25)
        $\widetilde{r}' \leftarrow \widetilde{r} + X_j R_j^{-1}(\nu_j - \nu_j')$   (26)
        $a_j \leftarrow \frac{\mathbb{1}_n^T}{n} X_j R_j^{-1}(\nu_j - \nu_j')$   (27)
        $\widetilde{r} \leftarrow \widetilde{r}', \nu_j \leftarrow \nu_j'$   (28)
    end
    $\widetilde{r} \leftarrow \widetilde{r} - (\sum_{j=1}^{J} a_j)\mathbb{1}_n$
end
```

To explain the changes to this algorithm, we modify (20), (22) to exploit the sparsity of $X$. Using (8), we first substitute $Q_j$ with $P_0 X_j R_j^{-1}$ to obtain

$$z_j = n^{-1} R_j^{-1} X_j^T P_0 r + \nu_j \tag{29}$$

$$r' \leftarrow r + P_0 X_j R_j^{-1}(\nu_j - \nu_j') \tag{30}$$

If $X_j$ are not centered, $P_0 X_j \neq X_j$. Therefore, at least an additional $2|K_j|n$ operations are required in (29), (30) compared to (20), (22). For a more efficient calculation, we carry out calculations as if $X_j$ are centered and do the correction all at once. Suppose we have calculated (20)-(22) replacing $Q_j$ with $X_j R_j^{-1}$. Then, the calculated residual $\widetilde{r}$ is off by a constant vector, as we see below:

$$r + P_0 X_j R_j^{-1}(\nu_j - \nu_j') = r + X_j R_j^{-1}(\nu_j - \nu_j') - \mathbb{1}_n \frac{\mathbb{1}_n^T}{n} X_j R_j^{-1}(\nu_j - \nu_j') \tag{31}$$

$$= r + X_j R_j^{-1}(\nu_j - \nu_j') - a_j \mathbb{1}_n \tag{32}$$

where $a_j := \frac{\mathbb{1}_n^T}{n} X_j R_j^{-1}(\nu_j - \nu_j')$, since $P_0 = (I_{n\times n} - \frac{\mathbb{1}_n \mathbb{1}_n^T}{n})$.

Now we show that (29) correctly calculates $z_j$. Since $P_0 \mathbb{1}_n = 0$, we have $P_0 \widetilde{r} = P_0 r$. Therefore,

$$z_j = n^{-1} R_j^{-1} X_j^T P_0 \widetilde{r} + \nu_j \tag{33}$$

$$= n^{-1} R_j^{-1} X_j^T \widetilde{r} - R_j^{-1} \left( \frac{X_j^T \mathbb{1}_n}{n} \right) \left( \frac{\mathbb{1}_n^T \widetilde{r}}{n} \right) + \nu_j \tag{34}$$

In (34) , $\left( \frac{X_j^T \mathbb{1}_n}{n} \right) = [\overline{X_{j,1}}, \ldots, \overline{X_{j,|K_j|}}]^T$ can be pre-calculated during the QR decomposition. Thus for the additional term in (34) calculating mean of $\widetilde{r}$ is needed, whcih takes $n+1$ operations. Also, since both terms are $|K_j| \times 1$ vectors, subtracting the second term from the first term is cheap as well. At the end of the cycle, we correct the residual vector all at once by $r \leftarrow \widetilde{r} - (\sum_{j=1}^J a_j) \mathbb{1}_n$. Note that $a_j$ are scalars, so this involves only $n$ operations.

## 2.3 R Package details

We provide a publicly available R implementation of our algorithm in the **PUlasso** package. For a fast and efficient implementation, all underlying computation is implemented in C++. The package uses warm start and strong rule (Friedman et al. [2007], Tibshirani et al.

[2012]), and a cross-validation function is provided as well for the selection of the regularization parameter $\lambda$. Our package supports a parallel computation through OpenMP, which is an application programming interface(API) supporting a multi-threading programming in C++. Also, for a scalable computing, the package allows for users to use memory-mapped files for the data larger than memory size. This functionality is implemented using the R package **bigmemory**, which provides memory-mapped data structures in R.

## 2.4 Run-time improvement

We illustrate the improvements in run-time for our two speed-ups. Note that we only include $p$ up to 100 so that we can compare to the original regularized EM algorithm. For our example that involves $p = O(10^4)$ and $n = O(10^6)$ the regularized EM algorithm is too slow to run efficiently. It is clear from our results that the quadratic majorization step is several orders of magnitude faster than the original EM algorithm, and exploiting the sparsity of $X$ provides a further 30% speed-up.

| | (n,p) | PUlasso | EM | time reduction(%) |
|---|---|---|---|---|
| Dense matrix | n=1000, p=10 | 0.94 | 443.72 | 99.79 |
| | n=5000, p=50 | 2.52 | 1844.98 | 99.86 |
| | n=10000, p=100 | 9.45 | 5066.86 | 99.81 |
| Sparse matrix | n=1000, p=10 | 0.40 | 196.86 | 99.80 |
| | n=5000, p=50 | 2.01 | 614.65 | 99.67 |
| | n=10000, p=100 | 4.29 | 1201.09 | 99.64 |

Table 1: Timings (in seconds). Sparsity level in $X = 0.95$, $n_\ell/n_u = 0.5$. Total time for $100\,\lambda$ values, averaged over 3 runs.

| (n,p) | sparse calculation | dense calculation | time reduction(%) |
|---|---|---|---|
| n=10000, p=100 | 12.91 | 19.24 | 32.89 |
| n=30000, p=100 | 25.64 | 38.73 | 33.79 |
| n=50000, p=100 | 39.47 | 57.18 | 30.97 |

Table 2: Timings (in seconds) using sparse and dense calculation for fitting the same simulated data. Sparsity level in X = 0.95, $n_\ell/n_u = 0.5$. Total time for $100\,\lambda$ values, averaged over 3 runs.

# 3 Statistical Guarantee

We now turn our attention to statistical guarantees for our PUlasso algorithm under the statistical model (1). In particular we provide error bounds for any stationary point of the non-convex optimization problem (6) and note that Proposition 2.1 guarantees that our PUlasso algorithm converges to a stationary point.

First note that the observed likelihood (2) is in an exponential family with non-canonical link and subsequently define the collection of functions $\mathcal{F} = \{f_\theta(\cdot); f_\theta(x) = \log \frac{n_\ell}{\pi n_u} + x^T\theta - \log(1+e^{x^T\theta}), \theta \in \Theta\}$ where $\Theta$ is a convex subset of $\mathbb{R}^p$. The negative observed log-likelihood function is used as the loss function, which is denoted as $\gamma_{f_\theta}(x,z) = -\log L(\theta; x, z) = -zf_\theta(x)+\log(1+e^{f_\theta(x)})$. Note that the link function $f_\theta(x)$ is not the canonical linear function anymore, which makes (6) a non-convex problem. The theoretical risk is defined as $\mathcal{R}(\theta) := \mathbb{E}[\gamma_{f_\theta}(X,Z)]$, and corresponding empirical risk is defined as $\mathcal{R}_n(\theta) := \frac{1}{n}\sum_{i=1}^n \gamma_{f_\theta}(X_i, Z_i)$.

We consider a regularized M-estimator of the form

$$\hat{\theta} = \underset{\|\theta\|_1 \le R_n, \|\theta\|_2 \le r_0}{\text{argmin}} \{\mathcal{R}_n(\theta) + P_\lambda(\theta)\} \tag{35}$$

following a similar framework to that developed in Loh and Wainwright [2013]. Note that the $\ell_1$ and $\ell_2$-constraints is in addition to the regularizer for theoretical convenience. The

$\ell_1$-constraint is also used in prior work in Loh and Wainwright [2013]. Our analysis here focuses on the case $P_\lambda(\theta) = \lambda\|\theta\|_1$. We note that Negahban et al. [2012] has shown that if the loss and regularization functions satisfy restricted strong convexity and decomposability properties, the arguments used to analyze $\ell_1$-regularizers can be generalized to other decomposable regularizers, including $\ell_1/\ell_2$ norms. Thus the results that we establish in the following section will have analogous generalizations to the group sparsity setting.

## 3.1 Assumptions

We impose the following assumptions.

**Assumption 1.** *The rows $x_i$ , $i = 1, 2, \ldots, n$ of the design matrix are i.i.d. samples from a sub-Gaussian distribution with parameter $\sigma_x$. Moreover, $\Sigma_x := E[XX^T]$ is a positive definite and with minimum eigenvalue $\lambda_{min}(\Sigma_x) > 0$.*

This independent design assumption is standard and is needed to prove the restricted strong convexity assumption to follow (see e.g. Raskutti et al. [2010] for details) .

**Assumption 2.** *For any $r > 0$ such that $\|\theta - \theta^*\|_2 \leq r$, there is a constant $K(r)$ such that $\max_i |x_i^T\theta| \leq K$ for all $\theta$ and all $1 \leq i \leq n$.*

Assumption 2 is a mild assumption that ensures that within a compact ball around $\theta - \theta^*$, $|x_i^T\theta|$ is also bounded which ensures that the probability $(1 + e^{-x_i^T\theta})^{-1}$ is between 0 and 1. This assumption is required for standard generalized linear models (see e.g. Negahban et al. [2012], Loh and Wainwright [2013]) to ensure that the curvature of the likelihood is bounded above and below.

**Assumption 3.** *The ratio of the number of labelled to unlabelled data , i.e. $n_\ell/n_u$ is lower bounded away from 0 and upper bounded for all $n = n_\ell + n_u$. Equivalently, there is a constant $K_2$ such that $|\log(n_\ell/\pi n_u)| \leq K_2$*

Assumption 3 ensures that the number of labelled samples $n_\ell$ is not too small or large relative to $n$. The reason why $n_\ell$ can not be too large is that the labelled samples are only positive and we need a reasonable number of negative samples which are a part of the unlablled samples. Finally we define the restricted strong convexity assumption for a loss function following the definition in Loh and Wainwright [2013].

**Definition 3.1** (Restricted strong convexity)**.** *We say $\mathcal{R}_n$ satisfies* restricted strong convexity *(RSC) condition with respect to $\theta^*$ with curvature $\alpha > 0$ and tolerance function $\tau$ over $\Theta_0$ if the following inequality is satisfied for all $\theta \in \Theta_0$:*

$$(\nabla \mathcal{R}_n(\theta) - \nabla \mathcal{R}_n(\theta^*))^T \Delta \geq \alpha \|\Delta\|_2^2 - \tau(\|\Delta\|_1) \tag{36}$$

*where $\Delta := \theta - \theta^*$ and $\tau(\|\Delta\|_1) = \tau_1 \|\Delta\|_1^2 \frac{\log p}{n} + \tau_2 \|\Delta\|_1 \sqrt{\frac{\log p}{n}}$.*

Similar RSC conditions were discussed in Negahban et al. [2012] and Loh and Wainwright [2013] with different $\tau, \Theta_0$. One of the important steps in our proof is to prove that RSC holds for the objective function $\mathcal{R}_n(\theta)$.

## 3.2 Guarantee

With Assumptions 1-3, we will show in Theorem 3.3 that RSC holds with high probability over $\Theta_0 := \{\theta; \|\theta\|_2 \leq r_0\}$ for a fixed $r_0$. Under the RSC assumption, the following lemma, which is a modification of Theorem 1 in Loh and Wainwright [2013], provides $\ell_1$ and $\ell_2$ bounds of an error vector $\hat{\Delta} := \hat{\theta} - \theta^*$.

**Lemma 3.1.** *Suppose the empirical loss $\mathcal{R}_n$ satisfies the RSC condition (*36*) with $\tau(\|\Delta\|_1) = \tau_1 \|\Delta\|_1^2 \frac{\log p}{n} + \tau_2 \|\Delta\|_1 \sqrt{\frac{\log p}{n}}$ over $\Theta_0$ where $\Theta_0$ is feasible region for the objective* (35) *and*

*$\theta^* \in \Theta_0$. Consider $\lambda$ such that*

$$4 \max \left\{ \|\nabla \mathcal{R}_n(\theta^*)\|_\infty, \left( \tau_1 \frac{2R_n \log p}{n} + \tau_2 \sqrt{\frac{\log p}{n}} \right) \right\} \leq \lambda$$

*Let $\hat{\theta}$ be a stationary point of* (35). *Then the following error bounds*

$$\|\hat{\Delta}\|_2 \leq \frac{3\sqrt{s}\lambda}{2\alpha} \qquad \textit{and} \qquad \|\hat{\Delta}\|_1 \leq \frac{6s\lambda}{\alpha}, \tag{37}$$

*hold where $s := \|\theta^*\|_0$, the number of non-zero elements in $\theta^*$.*

The proof for Lemma 3.1 is deferred to the supplementary material. From (37), note that the squared $\ell_2$-error grows proportionally with $s$ and $\lambda^2$. The random quantity $\|\nabla \mathcal{R}_n(\theta^*)\|_\infty$ can be bounded with high probability by $O\left(\sqrt{\frac{\log p}{n}}\right)$ by Lemma 3.2. Thus we obtain $\frac{\log p}{n}$ rate by choosing $\lambda$ proportional to $\sqrt{\frac{\log p}{n}}$ and $R_n$ proportional to $\sqrt{\frac{n}{\log p}}$.

**Lemma 3.2.** *$P\left( \|\nabla \mathcal{R}_n(f^*_\theta)\|_\infty \geq c\sqrt{\frac{\log p}{n}} \right) \leq c_1 exp(-c_2 \log p)$.*

Now we state our main theorem in this section which shows that RSC condition holds uniformly over a neighborhood of the true parameter.

**Theorem 3.3.** *There exist strictly positive constants $\alpha, \tau_1$ and $\tau_2$ depending on $\sigma_x, \lambda_{min}(\Sigma_x)$, $r, \|\theta^*\|_2$ such that*

$$\left(\nabla \mathcal{R}_n(\theta) - \nabla \mathcal{R}_n(\theta^*)\right)^T \Delta \geq \alpha \|\Delta\|_2^2 - \tau_1 \|\Delta\|_1^2 \frac{\log p}{n} - \tau_2 \|\Delta\|_1 \sqrt{\frac{\log p}{n}} \;\; \textit{for} \;\; \|\Delta\|_2 \leq r \tag{38}$$

*holds with probability at least $1 - c' \exp(-c'' n)$, where $\alpha, \tau_1$ and $\tau_2$ are defined as $\alpha := \frac{L_0}{4}\lambda_{\min}(\Sigma_x), \tau_1 := \frac{16 L_0 c_1^2 \tau^2}{\lambda_{min}(\Sigma_x)}, \tau_2 = \frac{c_1(9+K)}{4}$, where $L_0 := \inf_{|u| \leq K_2 + \min\{T\|\theta^*\|_2 + \tau r, 3K\}} A''(u)/(1 + e^{\min\{T\|\theta^*\|_2 + \tau r, 3K\}})^2$ and $c_1$ is a universal constant and $\tau, T$ are constants depending only on $\sigma_x$ and $\lambda_{\min}(\Sigma_x)$.*

There are a couple of notable remarks about Theorem 3.3 and Lemma 3.1.

- The mean-squared error $\frac{s\log p}{n}$ is verified below in Fig. 2 and both the mean-squared error and $\ell_1$ errorare minimax optimal for high-dimensional linear regression Raskutti et al. [2011].

- Suppose we want to achieve $\ell_2$ error less than $\epsilon > 0$. Let $R_n := \sqrt{\frac{K_0 n}{4\log p}}$ for some $K_0$. We assume that $K_0, r_0$ are carefully chosen so that $\|\theta^*\|_1 \leq R_n, \|\theta^*\|_2 \leq r_0/2$. Applying Theorem 3.3 with $r := r_0/2$, (38) holds for $\theta$ such that $\|\theta - \theta^*\|_2 \leq r_0/2$ with probability at least $1 - c' \exp(-c''n)$, which implies that RSC condition over $\{\theta; \|\theta\|_2 \leq r_0\}$ by triangle inequality. Now in light of Lemma 3.1, we set $\lambda$ such that $\lambda := \dfrac{2\alpha\epsilon}{3\sqrt{s}}$. The conditions of Lemma 3.1 requires $\tau_1 \dfrac{2R\log p}{n} + \tau_2\sqrt{\dfrac{\log p}{n}} \leq \dfrac{\lambda}{4} = \dfrac{\alpha\epsilon}{6\sqrt{s}}$, which is ensured if we set each term less than $\dfrac{\alpha\epsilon}{12\sqrt{s}}$. Therefore

$$\frac{16L_0c_1^2\tau^2}{\lambda_{\min}(\Sigma_x)}\sqrt{\frac{K_0\log p}{n}} \leq \frac{\epsilon L_0 \lambda_{\min}(\Sigma_x)}{48\sqrt{s}} \quad \text{and} \quad \frac{c_1(9+K)}{4}\sqrt{\frac{\log p}{n}} \leq \frac{\epsilon L_0 \lambda_{\min}(\Sigma_x)}{48\sqrt{s}}.$$

  Therefore a sufficient condition to achieve $\ell_2$-error less than $\epsilon$ in terms of $n$ is

$$\min\left\{c'\frac{\tau^4 K_0 s\log p}{\epsilon^2\lambda_{min}^4(\Sigma_x)}, c''\frac{(9+K)^2 s\log p}{\epsilon^2 L_0^2\lambda_{min}^2(\Sigma_x)}\right\} \leq n \tag{39}$$

  for universal constants $c', c''$. We note from (39) that the required sample size is inversely related with $\epsilon$ and $\lambda_{\min}(\Sigma_x)$ and proportional to $K_0, r_0$ and $K$ since $L_0$ is inversely related to $r_0$.

To validate the mean-squared error upper bound of $\frac{s\log p}{n}$ in Section 3, a synthetic dataset was generated according to our logistic regression model with $p = 500$ covariates and $X \sim N(0, I_{500\times500})$. Varying $s$ and $n$ were considered to study the rate of convergence of $\|\hat{\theta} - \theta^*\|_2$. The ratio $n_\ell/n_u$ was fixed to be 1. For each dataset, $\hat{\theta}$ was obtained by applying PUlasso algorithm with a lambda sequence $\lambda_n := K_s\sqrt{\frac{\log p}{n}}$ for a suitably chosen $K_s$ for each $s$. We repeated the experiment 100 times and average $\ell_2-$error was calculated.

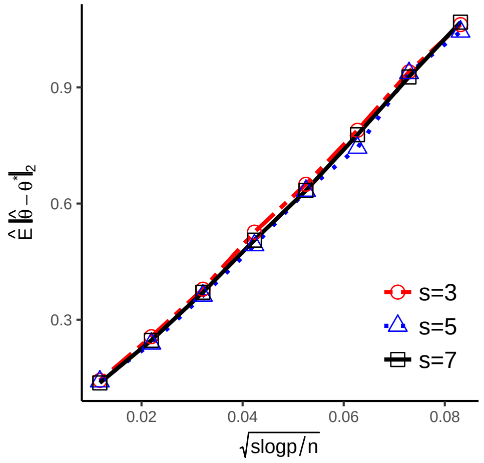

Figure 2: $\hat{E}[\|\hat{\theta}-\theta\|_2]$ plotted against $\sqrt{s\log p/n}$ with fixed p=500 and varying $s$ and $n$

In Figure 2, we illustrate the rate of convergence of $\|\hat{\theta}-\theta^*\|_2$. In particular, $\|\hat{\theta}-\theta^*\|_2$ against $\sqrt{\frac{s\log p}{n}}$ is plotted with varying $s$ and $n$. The error appears to be linear in $\sqrt{\frac{s\log p}{n}}$, and thus we also empirically conclude that our algorithm achieves the optimal $\sqrt{\frac{s\log p}{n}}$ rate.

# 4 Simulation study: Classification performance

In this section, we provide a simulation study whih validates the classification performance for PUlasso. In particular we provide a comparison in terms of classification performance to state-of-the-art methods developed in Du Marthinus et al. [2015], Elkan and Noto [2008], Liu et al. [2003]. The focus of this section is classification rather than variable selection since the state-of-the-art methods we compare to are developed mainly for classification and are not developed for variable selection.

### 4.1 Setup

We consider a number of different simulation settings: (i) small and large $p$ to distinguish the low and high-dimensional setting; (ii) probabilistic and deterministic responses based on the original samples; and (iii) weakly and strongly separated populations. More precisely, in the deterministic scheme we draw covariates $x_i$ from positive or negative population, $P_{pos} := N(\mu, I_{p\times p})$ or $P_{neg} := N(-\mu, I_{p\times p})$, and assign $y_i = 1$ or 0 respectively. It is assumed that 50% of the population comes from $P_{pos}$. Since the response $y$ no longer follows a logistic model given $x$ in this scheme, it is a misspecified model for the PUlasso algorithm. To make only the first $s$ variables active, we let the centers of the first $s$ variables of $\mu$ to be non-zero and take values of $\mu_0$, i.e. $\mu = (\mu_0, \ldots, \mu_0, 0, \ldots, 0)^T$. To address the effect of separation between the positive and negative samples, we consider $\mu_0 \in \{0.5, 1, 2\}$. To compare performances both in low and high dimensional setting, we consider $(p = 10, s = 2)$ and $(p = 10000, s = 5)$. We set the sample size $n_\ell = n_u = 1000$ is used in both cases.

In the probabilistic scheme, we generate the $X$ matrix in the same way (i.e. $x_i \sim P_{pos}$ or $x_i \sim P_{neg}$), but further simulate the response $y$ based on model (1). In order to make only the first $s$ variables active, we let $\theta^* = [\theta_0, \ldots, \theta_s, 0, \ldots, 0]^T$. Non-zero $\theta_i$ are sampled from uniform distribution $Unif[0.5, 1]$.

### 4.2 Comparison methods

Our experiments compare six algorithms: (i) logistic regression model assuming we know the true responses (*oracle* estimator); (ii) our PUlasso algorithm; (iii) a bias-corrected logistic regression algorithm in Elkan and Noto [2008]; (iv) a second algorithm from Elkan and Noto [2008] that is effectively a one-step EM algorithm (v) the biased SVM algorithm from Liu et al. [2003] and (vi) the PU-classification algorithm based on assymetric loss functions from Du Marthinus et al. [2015]. We briefly describe algorithms (iii) - (vi) here. To avoid potential confusion with notation from the original papers, we use the same notations in original papers and denote negative samples as 0 in algorithms (i)-(iv) and $-1$ in algorithms (v)-(vi).

The biased SVM algorithm from Liu et al. [2003] involves solving the following optimization problem:

$$\text{minimize: } \frac{1}{2}w^T w + C_+ \sum_{i=1}^{k-1} \xi_i + C_- \sum_{i=k}^{n} \xi_i \qquad \text{subject to: } y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1, \dots, n$$
$$\xi_i \geq 0, i = 1, \dots, n,$$

where $C_+, C_-$ are tuning parameters. By setting $C_+$ and $C_-$ differently, positive errors and negative errors are penalized differently. Since having larger values of $C$ would increase the "cost" of misclassification, we impose $C_+ \geq C_-$. $C_+$ and $C_-$ are chosen by a two-dimensional grid search based on its performance on the validation data.

Elkan and Noto [2008] provided two algorithms for the PU-learning classification problem; Their first algorithm estimates $P(z = 1|x)$ using observed data, then corrects the bias because of using $z's$ instead of $y's$ via estimation of selection probability $P(z = 1|y = 1)$. The noiseless assumption in positive samples, i.e. $P(y = 1|x) = 1, \forall x \sim P_{pos}$ is crucial to estimate $P(z = 1|y = 1)$. Their second method is a modification of the first method; a unit weight is assigned to each labeled sample, and each unlabeled example is treated as a combination of a positive and negative example with weight $P(y = 1|x, z = 0)$ and $1 - P(y = 1|x, z = 0)$, respectively. Then classifier is built using augmented positive and negative examples with associated weights. The second algorithm can be viewed as a one-step iteration of the EM algorithm in Section 2.

To be more concrete, we consider augmented data $(\tilde{x}_i)_{i=1}^{n_\ell+2n_u}, (\tilde{y}_i)_{i=1}^{n_\ell+2n_u}$ :

$$\tilde{x}_i = \begin{cases} x_i, & \text{for } 1 \leq i \leq n_\ell + n_u \\ x_{i-n_u}, & \text{for } n_\ell + n_u + 1 \leq i \leq n_\ell + 2n_u \end{cases} \qquad \tilde{y}_i = \begin{cases} 1, & \text{for } 1 \leq i \leq n_\ell + n_u \\ 0, & \text{for } n_\ell + n_u + 1 \leq i \leq n_\ell + 2n_u \end{cases}$$

and weights $(w_i)_{i=1}^{n_\ell+2n_u}$

$$w_i(\theta^m) = \begin{cases} 1, & \text{for } 1 \leq i \leq n_\ell \\ P_{\theta^m}(y=1|x), & \text{for } n_\ell + 1 \leq i \leq n_\ell + n_u \\ 1 - P_{\theta^m}(y=1|x), & \text{for } n_\ell + n_u + 1 \leq i \leq n_\ell + 2n_u \end{cases}$$

Then logistic regression model with non integer responses $\hat{y}_i(\theta^m)$ is

$$\begin{aligned} Q(\theta;\theta^m) &= \sum_{i=1}^{n_\ell+n_u} \hat{y}_i(\theta^m)(x_i^T\theta + b) - \log(1 + e^{x_i^T\theta+b}) \\ &= \sum_{i=1}^{n_\ell+2n_u} w_i(\theta^m)\left[\tilde{y}_i(x_i^T\theta + b) - \log(1 + e^{x_i^T\theta+b})\right]. \end{aligned}$$

Recent work in Du Plessis and Sugiyama [2014], Du Marthinus et al. [2015] suggests asymmetric loss functions to cancel the bias induced by separating positive and unlabelled samples rather than positive and negative samples. More concretely, authors considered following optimization problem

$$g^* = \operatorname*{argmin}_{\alpha,b} \pi\hat{E}_1[L(g(x)) - L(-g(x))] + \hat{E}_x[L(-g(x))] + \frac{\lambda}{2}\alpha^T\alpha$$

where $g(x) = \alpha^T x + b$, $L$ is a surrogate convex loss function of 0-1 loss. There is a publicly available matlab implementation of the algorithm when $L$ is a squared loss on the author's webpage and since we use their code and implementation, squared loss is considered.

Algorithm (v) involves a two-dimensional grid search to determine $C_+$ and $C_-$. In the original paper Liu et al. [2003], $C_- \in \{0.01, 0.03, 0.05, \ldots, 0.61\}$ and $C_+/C_- \in \{10, 20, 30, \ldots, 200\}$

were considered. $C_+/C_-$ encodes the relative cost of misclassification the resulting classification is more sensitive to the ratio than the absolute value of $C_-$. Also when samples are weakly separated, the resulting classification tends to have a very high proportion of positives in because of the assymetric costs of misclassification. Therefore in our simulation setting we use a wider range of $C_+/C_-$ ratio while making $C_-$ more spread out. Specifically, we considered $C_+/C_- \in \{1, 2, \dots, 200\}$ for each $C_- \in \{0.01, 0.16, 0.31, 0.46, 0.61\}$. In the high dimensional setting, we excluded algorithm (v) since (v) requires a grid search over two dimension, which makes the computational cost prohibitive.

## 4.3 Classification comparison

We use two criteria, mis-classification rate and $F_1$ score, to evaluate performances. $F_1$ is the harmonic mean of the precision and recall, which is calculated as $F_1 := 2 * \frac{\text{precision+recall}}{\text{precision*recall}}$. Experiments are repeated 10 times and the average score is reported. The best and equivalent results (5% z-test) are in bold and the best is underlined. The results were stable over the 10 repetitions. For concise presentation we do not include error bars.

Results are displayed in Tables 3-4. Not surprisingly the oracle estimator has the best accuracy in all cases. PUlasso and algorithm (vi) performs almost as well as the oracle in the low-dimensional setting, and better than remaining methods in most cases. It must be pointed out that both PUlasso and algorithm (vi) use additional knowledge $\pi$ of the true prevalence of positives in unlabelled samples. In contrast, the performance of PUlasso is significantly better than algorithm (vi) in the high-dimensional setting as estimation errors can be greatly reduced by imposing many 0's on the estimates in PUlasso due to the $\ell_1$-penalty (compared to $\ell_2$-penalty in algorithm (vi)).

The performance of (iii)-(v) are greatly improved in the deterministic case, especially when samples are strongly separated as algorithms (iii)-(v) depend on no noise assumption

|  |  | weak separation<br>($\mu_0 = .5$) |  | medium separation<br>($\mu_0 = 1$) |  | strong separation<br>($\mu_0 = 2$) |  |
|---|---|---|---|---|---|---|---|
|  | low-d | error rate | $F_1$ score | error rate | $F_1$ score | error rate | $F_1$ score |
| (i) | Reference | 31.2% | 0.7507 | 26.0% | 0.7888 | 12.1% | 0.8842 |
| (ii) | PUlasso | **32.1%** | **0.7471** | **26.4%** | **0.7826** | **12.2%** | **0.8830** |
| (iii) | EN-1 | 39.9% | **0.7485** | 39.1% | 0.7551 | 19.1% | 0.8453 |
| (iv) | EN-2 | 38.5% | **0.7537** | 36.3% | 0.7668 | 19.2% | 0.8426 |
| (v) | Biased SVM | 40.0% | **0.7480** | 37.5% | 0.7562 | **12.0%** | **0.8848** |
| (vi) | PNS | **32.9%** | **0.7453** | **27.4%** | **0.7872** | **12.1%** | **0.8846** |
|  | high-d | error rate | $F_1$ score | error rate | $F_1$ score | error rate | $F_1$ score |
| (i) | Reference | 28.4% | 0.7743 | 18.3% | 0.8356 | 5.1% | 0.9492 |
| (ii) | PUlasso | **32.3%** | **0.7540** | **19.1%** | **0.8225** | **5.2%** | **0.9485** |
| (iii) | EN-1 | 39.9% | **0.7491** | 38.2% | 0.7459 | **5.4%** | **0.9470** |
| (iv) | EN-2 | 39.8% | **0.7498** | 33.6% | 0.7680 | **5.8%** | **0.9427** |
| (vi) | PNS | 39.9% | **0.7479** | 41.9% | 0.7275 | 30.3% | 0.7676 |

Table 3: Average misclassification rate and $F_1$ score in probabilistic scheme

|  |  | weak separation<br>($\mu_0 = .5$) |  | medium separation<br>($\mu_0 = 1$) |  | strong separation<br>($\mu_0 = 2$) |  |
|---|---|---|---|---|---|---|---|
|  | low-d | error rate | $F_1$ score | error rate | $F_1$ score | error rate | $F_1$ score |
| (i) | Reference | 23.9% | 0.7593 | 7.7% | 0.9239 | 0.3% | 0.9975 |
| (ii) | PUlasso | **24.1%** | **0.7518** | **7.8%** | **0.9217** | 0.4% | 0.9955 |
| (iii) | EN-1 | 40.2% | 0.7093 | 11.4% | 0.8966 | 0.3% | 0.9968 |
| (iv) | EN-2 | 36.1% | 0.7284 | 11.6% | 0.8949 | **0.3%** | **0.9974** |
| (v) | Biased SVM | 27.1% | 0.7304 | 9.1% | 0.9134 | **0.2%** | **0.9979** |
| (vi) | PNS | **24.2%** | **0.7576** | **7.9%** | **0.9219** | **0.3%** | **0.9973** |
|  | high-d | error rate | $F_1$ score | error rate | $F_1$ score | error rate | $F_1$ score |
| (i) | Reference | 13.9% | 0.8599 | 1.3% | 0.9873 | 0.0% | 1.0000 |
| (ii) | PUlasso | **16.4%** | **0.8172** | 3.5% | 0.9639 | 0.2% | 0.9980 |
| (iii) | EN-1 | 32.9% | 0.7515 | **2.2%** | **0.9786** | **0.0%** | **0.9998** |
| (iv) | EN-2 | 27.2% | 0.7839 | **2.3%** | **0.9774** | **0.0%** | **0.9999** |
| (vi) | PNS | 48.7% | 0.6702 | 43.9% | 0.6966 | 24.2% | 0.8062 |

Table 4: Average misclassification rate and $F_1$ score in deterministic scheme

in positive labels, i.e. $P(y = 1|x) = 1, \forall x \sim P_{pos}$. Although methods (iii)-(v) slightly out-perform PUlasso in the deterministic case with strong separation, it must be noted that PUlasso is developed for the probabilistic scheme whilst the deterministic setting is developed to suit methods (iii)-(v).

# 5 Analysis of beta-glucosidase sequence data

Our original motivation for developing the PUlasso algorithm was to analyze a large-scale dataset with postive and unlablled responses developed by the lab of Dr.Philip Romero (Romero et al. [2015]). Prior approaches did not scale to the size of this dataset and in this section, we discuss the performance of our PUlasso algorithm on a dataset involving mutations of a natural beta-glucosidase (BGL) enzyme. To provide context, BGL is a hydrolytic enzyme involved in the deconstruction of biomass into fermentable sugars for biofuel production. Functionality of the BGL enzyme is measured in terms of whether the enzyme deconstructs d-saccharides into glucose or not. Dr. Romero used a microfluidic screen to generate a BGL dataset containing millions of sequences (Romero et al. [2015]).

Three different validation schemes are used for the PUlasso algorithm on the BGL dataset. Firstly in Section 5.2 we discuss a method for validating the selected features by developing a test statistic and corresponding p-value. We also test the classification performance using a modified ROC and AUC approach. Finally a scientific validation is performed based on a follow-up experiment conducted by the Romero lab which used the variables selected by PUlasso to design a new BGL enzyme and the performance is compared to the original natural BGL enzyme.

## 5.1 Data Description

The dataset consists of $n_\ell = 2647877$ labelled and functional sequences and $n_u = 1567203$ unlabelled sequences where for each of the observation of length $p = 500$. Each of the position takes one of $M = 21$ discrete values, which correspond to the 20 amino acids in the DNA code and an extra to include the possibility of a gap. Another important aspect of the millions of sequences generated is that a "base wild-type BGL sequence" was considered and known to be functional ($y = 1$), and the millions of sequences were generated by *mutating* the base sequence. Single mutations (changing one position from the base sequence) and double mutations (changing two positions) from the base sequence were common but higher-order mutations were not prevalent using the high-throughput sequencing approach (Romero et al. [2015]). Hence the sequences generated were not random samples across the entire enzyme sequence space, but rather very local sequences around the wild-type sequence. Hence the number of possible sequences is also reduced dramatically and we labeled the base sequence as the 0-vector. With this dataset, we want to determine which mutations should we apply to the wild-type BGL sequence.

Although there are in principle $p \approx d(M-1)$ variables for a main-effects logistic regression model and $p \approx d^2(M-1)^2$ if we include main-effects and two-way interactions, there are many amino acids that never appear in any position or appear only once, and these are removed from the feature space. Using this basic pre-processing we obtained only 3075 corresponding to single mutations and 930 binary variables corresponding to double mutations and 500 unique positions and 820 two-way interactions between positions respectively. Higher-order interactions were not modeled as they did not frequently arise. We use both main-effect and two-way interaction logistic regression models.

In summary, we consider following two models and corresponding design matrices

$$X_{main} := [\text{Intercept(1)+ main effects(3075)}] \in \{0,1\}^{4215080\times 3076}$$

$$X_{int} := [\text{Intercept(1)+ main effects(3075)+ two way interactions(930)}] \in \{0,1\}^{4215080\times 4006}$$

and the response vector $z = [1,\dots,1,0,\dots,0]^T \in \{0,1\}^{4215080}$.

## 5.2 Variable selection validation

In this section we develop a variable selection validation method which we apply to the selected models for our BGL example. One of the challenges is that variables are adaptively chosen, so a typical chi-squared test to test the significance of an additional variable between two nested models would be too liberal. Lockhart et al. [2014] suggested the covariance test statistic and Cai and Yuan [2014] proposed an alternative test statistic which both make a correction of the difference of the deviances $R_j$ between a reduced model $X_A$ and a full model $X_{A\cup\{j\}}$. Although those two test statistics are mainly discussed in the context of linear model, the latter can be extended to generalized linear model, thus here we follow the approach by Cai and Yuan [2014]. The basic idea is for a given subset $A$, if we have an orthogonal design, the next selected index $j$ can be identified with $T_j = \max_{m\in A^c} R_m$ where $R_m := 2(\log L(\theta_{A\cup\{m\}}) - \log L(\theta_A))$. Under the null hypothesis (a variable is not in the model), $R_m \sim \chi^2(1)$, if we assume $X^TX = I_{p\times p}$, we have

$$T_j - 2\log|A^c| + \log\log(|A^c|) \xrightarrow{d} \text{Gumbel}(-\log\pi, 2) \tag{40}$$

. Even though $X^TX \neq I_{p\times p}$, it can be shown that the limiting distribution is still Gumbel. For a more detailed discussion, see Cai and Yuan [2014].

Now we extend this result to group sparsity case. Let the set of groups $\mathcal{G}$ be $\{K_0, K_1, \dots, K_J\}$ where $K_m$ is a collection of indices for group $m$. Given a collection of grouped indices $K_m$

and a reduced model $X_A$ where $A$ is an union of some $K_m$, $T_j := \max_{m \in \mathcal{G}\setminus A} R_m$, but now $R_m$ has $|K_m|$ degrees of freedom rather than 1. The derivation of asymptotic distribution of $\mathcal{T}$ is rather cumbersome as $R_m$ are not identical anymore. Instead, we can consider $\widetilde{T}_j$ which bounds $T_j$ almost surely and has a tractable asymptotic distribution.

**Lemma 5.1.** *Under $H_0 : supp(\theta^*) \subset A$, there is $\widetilde{T}_j$ such that $T_j \le \widetilde{T}_j$ a.s. and*

$$\widetilde{T}_j - 2\log|\mathcal{G}\setminus A| - (K_{\max} - 2)\log\log(|\mathcal{G}\setminus A|) \xrightarrow{d} \textit{Gumbel}(-2\log\Gamma(K_{\max}/2), 2) \qquad (41)$$

*Proof.* Define $K_{\max} := \max_j |K_j|$. Introducing a random variable $U$ such that $U$ is independent of $R_m$ and $U \sim \chi^2(K_{\max} - |K_m|)$, we can construct $\widetilde{R}_m := R_m + U$ such that $\widetilde{R}_m \sim \chi^2(K_{\max})$ and $R_m \le \widetilde{R}_m$ almost surely. Now we denote $\widetilde{T}_j$ as $\widetilde{T}_j := \max_{m\in\mathcal{G}\setminus A} \widetilde{R}_m$. Then obviously $T_j \le \widetilde{T}_j$. The claimed asymptotic distribution is based on the asymptotic result about the distribution of Gamma random variables. In particular, if $M_n := \max_n X_n, X_n \sim \text{Gamma}(\alpha, \frac{1}{\beta})$, then

$$\beta(M_n - \frac{1}{\beta}(\log n + (\alpha - 1)\log\log n - \log\Gamma(\alpha))) \xrightarrow{d} \text{Gumbel}(0, 1)$$

For more details about this asymptotic result, see for example, p. 156 in Embrechts et al. [1997]. Our desired result is a special case when $\alpha = \frac{K_{\max}}{2}$ and $\beta = \frac{1}{2}$. □

We note that (40) is a special case of (41) when $K_{\max} = 1$. By (41), we can test the significance of adding another variable or a group of variables by comparing $T_j$ or $\widetilde{T}_j$ with $1-\alpha$ quantile of $\text{Gumbel}(-\log\pi, 2)$ or $\text{Gumbel}(-2\log\Gamma(\frac{K_{\max}}{2}), 2)$.

We considered two models for the BGL enzyme: a main-effect model where we use the standard $\ell_1$-penalty (i.e. $P_\lambda(\theta) = \sum_{j=1}^{3075} |\theta_j|$) and a main-effect model plus two-way interactions with group $\ell_1/\ell_2$-penalty (i.e. $P_\lambda(\theta) = \sum_{j=1}^{1320} \sqrt{|K_j|}\|\theta_{K_j}\|_2$ where $\theta_j \in \mathbb{R}^{|K_j|}$) where 1320 comes from the sum of 500 positions plus 820 pairs of positions along the sequence. In the main-effects models, we aimed to select the important positions and corresponding amino acids in terms of the functionality of the BGL enzyme and design a particular functional sequence using a fitting result with the first model. On the other hand, for the interaction models we tried to identify a group of variables corresponding to a particular position or interaction with the second model.

First 5 selected states

| State | Group Size | Statistic | p-value |
|---|---|---|---|
| F288L | 1 | 4322.1 | 0.000 |
| W286R | 1 | 3854.0 | 0.000 |
| F263L | 1 | 3239.5 | 0.000 |
| W24R | 1 | 2921.3 | 0.000 |
| W47R | 1 | 2830.0 | 0.000 |

First 5 selected states with positive Co-efs

| State | Group Size | Statistic | p-value |
|---|---|---|---|
| G327A | 1 | 224.8 | 0.000 |
| E495G | 1 | 128.3 | 0.000 |
| D164E | 1 | 131.5 | 0.000 |
| S486P | 1 | 103.7 | 0.000 |
| T478S | 1 | 87.6 | 0.000 |

Table 5: Main effect model

First 5 selected positions

| Position | Group Size | Statistic | p-value* |
|---|---|---|---|
| 288 | 6 | 7882.7 | 0.000 |
| 263 | 6 | 6857.4 | 0.000 |
| 307 | 7 | 6968.3 | 0.000 |
| 293 | 6 | 5214.5 | 0.000 |
| 286 | 6 | 4055.3 | 0.000 |

First 5 selected interactions

| Interaction | Group Size | Statistic | p-value* |
|---|---|---|---|
| 384/400 | 1 | 143.2 | 0.000 |
| 427/435 | 1 | 123.8 | 0.000 |
| 288/289 | 1 | 96.9 | 0.000 |
| 231/234 | 1 | 234.4 | 0.000 |
| 288/293 | 1 | 135.3 | 0.000 |

Table 6: Main effect+Interaction model

p-value*s are upper bounds of conditional p-values which indicate significance of adding additional grouped variables and were obtained by comparing test statistics with null distribution of Gumbel$(-2\log\Gamma(\frac{K_{\text{max}}}{2}), 2)$, when $K_{\text{max}} = 8$.

In Tables 5 and 6 we report the 5 most influential variables and variables with positive coefficients using the main-effects model , and the 5 most influential positions and interactions using the main effect plus interaction model. The reason we are interested in the mutations with positive co-efficients is because the ultimate scientific goal is to improve functionality, so we seek mutations that are more likley to predict a 1. We note that the test statistics for variables with positive coefficients in the main effect model or interactions in the main effect plus interaction model are much smaller. Mutations of the BGL sequence in general decrease the functionality of the enzyme from the original base sequence (which is functional). Also only a small number of two-way interactions are observed in the experiments, thus the parameter values for the interaction terms are small.

We also examined the stability of the selected features for both models as the training data changes. Following the methodology of Kalousis et al. [2007], we measure similarity between two subsets of features $s, s'$ using $S(s, s')$ defined as $S_S(s, s') := 1 - \frac{|s| + |s'| - 2|s \cup s'|}{|s| + |s'| - |s \cup s'|}$. $S_s$ takes values in $[0, 1]$, where 0 means that there is no overlap between the two sets, and 1 that the two sets are identical. For each of cross-validation training folds, we obtain a set of selected features. $S_s$ is computed for each pair of two training folds (i.e. we have $\frac{9 \cdot 10}{2}$ pairs ) and finally we use the average $S_s$ over all pairs.

Feature selection turned out to be very stable across all tuning parameter $\lambda$ values: on average we had about 95% overlap of selection in main effect model (M) and about 99% overlap in main effect+interaction model (M+I). Stability score is higher in the latter model since we do a feature selection on groups, whose number is much less than individual variables (1320 groups versus 3076 individual variables).

| Model | 1Q | Median | Mean | 3Q |
|---|---|---|---|---|
| M | 93.3% | 94.9% | 94.9% | 96.8% |
| M+I | 98.5% | 99.7% | 99.2% | 100.0% |

Table 7: Summary of stability scores across all tuning parameter $\lambda$ values

## 5.3 Classification validation

Next we validate the classification performance for both the main-effect and two-way interaction models. We use Area Under the ROC Curve (AUC) to evaluate the classification performance. Since positive and negative samples are mixed in the unlabelled test dataset this is a non-trivial task with presence-only responses. A standard approach is to treat unlabelled samples as negative and estimate AUC, but if we do so, the AUC is inevitably downward-biased since the maximum AUC is $1 - \pi/2$. To adjust such bias, we follow the methodology suggested in Jain et al. [2017] and adjust AUC using the following equation:

$$AUC^{adj} = \frac{AUC^{PU} - \pi/2}{1 - \pi}$$

where $\pi$ is the prevalence of positive samples.

As Fig. 3 shows, we have a significant improvement in AUC from random assignment ($AUC = 0.5$). A small improvement in AUC occurs when we consider the two-way interaction model (M+I) compared to the main-effect model (M). This can be explained by the fact that only a small number of two-way interactions are observed in the experiments.

## 5.4 Scientific validation: Designed BGL sequence

Finally we provide scientific validation of the mutations estimated by our PUlasso algorithm. In particular, using the mutations with positive estimated coefficients because we are interested in mutations that enhance the performance of the sequence, Dr. Romero's lab

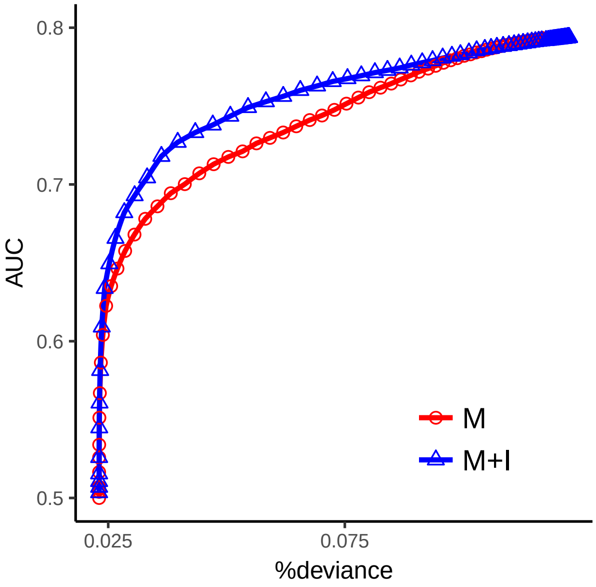

Figure 3: Plots of AUC scores of main effect model (M) and main effect+interaction model (M+I) over deviance ratio on the validation set. Deviance ratio is calculated as %deviance := 1 − deviance(model)/deviance(null), where deviance(model) := $-2 \log L(\theta_{model})$, and $\theta_{null} = [\log \frac{\pi}{1-\pi}, \ldots, 0]^T$. Deviance ratio serves as a measurement of the degree of model saturation; we use it as the x axis to approximately put both main effect and main effect+interaction model on the same scale.

designed the BGL sequence with the 10 positive mutations from Table 8. This sequence was synthesized, expressed, and assayed for its hydrolytic activity. Hence the designed sequence has 10 mutations compared to the wild-type (base) BGL sequence.

Figure 4 shows that the designed sequence has increased activity relative to the wild-type enzyme. These promising results suggest that our variable selection method is able to identify positions of the wild-type sequences with improved functionality.

| Base/Position/Mutated | |
|---|---|
| T197P | E495G |
| K300P | A38G |
| G327A | S486P |
| A150D | T478S |
| D164E | D481N |

Table 8: Ten positive mutations

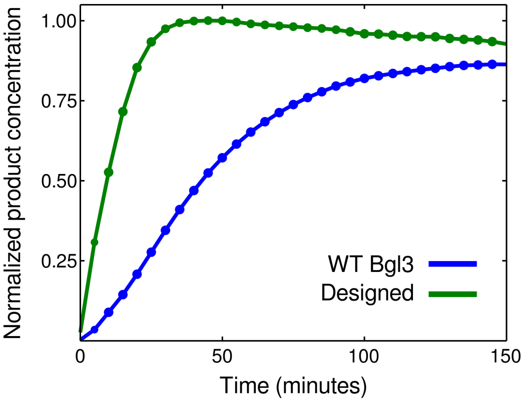

Figure 4: kinetics

10 positive mutations used in the lab(Base state/Position/Mutated state) and kinetics of designed BGL enzyme versus wild-type (WT) BGL sequence. The designed BGL enzyme based on mutations from Table 8 displays faster kinetics than the wild-type (WT) BGL sequence.

# 6 Conclusion

In this paper we developed the PUlasso algorithm for both variable selection and classification for high-dimensional classification with presence-only responses. Theoretically, we showed that our algorithm converges to a stationary point and every stationary points achieves optimal mean squared error (up to constant). We also demonstrated that our algorithm performs well on both simulated and real data. In particular, our algorithm produces more accurate results than the existing techniques in simulations and performs well on a real biochemistry application.

# References

G. Blanchard, G. Lee, and C. Scott. Semi-Supervised Novelty Detection. *Journal of Machine Learning Research*, 11:2973–3009, 2010.

P. Breheny and J. Huang. Group descent algorithms for nonconvex penalized linear and logistic regression models with grouped predictors. *Statistics and Computing*, 25(2): 173–187, 2013.

T. T. Cai and M. Yuan. Discussion: A significance test for the lasso. *The Annals of Statistics*, 42(2):478–482, apr 2014.

P. Du Marthinus, G. Niu, and M. Sugiyama. Convex Formulation for Learning from Positive and Unlabeled Data. *Proceedings of The 32nd International Conference on Machine Learning*, pages 1386–1394, 2015.

M. C. Du Plessis and M. Sugiyama. Class prior estimation from positive and unlabeled data. *IEICE Transactions on Information and Systems*, E96-D(5):1358–1362, 2014.

C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the* $14^{th}$ *ACM SIGKDD International Conference on Knowledge Discovery* *and Data Mining*, KDD '08, pages 213–220, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4.

P. Embrechts, C. Klüppelberg, and T. Mikosch. *Modelling Extremal Events*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997. ISBN 978-3-642-08242-9.

D. M. Fowler and S. Fields. Deep mutational scanning: a new style of protein science. *Nature Methods*, 11:801–807, 2014.

J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

R. T. Hietpas, J. D. Jensen, and D. N. A. Bolon. Experimental illumination of a fitness landscape. *Proceedings of the National Academy of Sciences of the United States of America*, 108(19):7896–7901, 2011.

S. Jain, M. White, M. W. Trosset, and P. Radivojac. Nonparametric semi-supervised learning of class proportions. *CoRR*, abs/1601.01944, 2016.

S. Jain, M. White, and P. Radivojac. Recovering true classifier performance in positive-unlabeled learning. In *AAAI*, 2017.

A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms: A study on high-dimensional spaces. *Knowl. Inf. Syst.*, 12(1):95–116, May 2007.

T. Lancaster and G. Imbens. Case-control studies with contaminated controls. 71:145–160, 02 1993.

K. Lange, D. R. Hunter, and I. Yang. Optimization Transfer Using Surrogate Objective Functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes.* Classics in Mathematics. Springer Berlin Heidelberg, 2013. ISBN 9783642202124.

B. Liu, Y. Dai, X. Li, W. S. Lee, and P. Yu. Building Text Classifiers Using Positive and Unlabeled Examples. *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)*, 2003.

R. Lockhart, J. Taylor, R. J. Tibshirani, and R. Tibshirani. A significance test for the lasso. *Ann. Statist.*, 42(2):413–468, 04 2014.

P.-L. Loh and M. J. Wainwright. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 1:1–9, 2013.

S. N. Negahban, R. Pradeep, B. Yu, and M. J. Wainwright. A Unified Framework for High-Dimensional Analysis of M-Estimators with Decomposable Regularizers. *Statistica Sinica*, 27(4):538–557, 2012.

J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables.* Classics in applied mathematics. SIAM, New York, 2000.

G. Raskutti, M. J. Wainwright, and B. Yu. Restricted eigenvalue conditions for correlated Gaussian designs. *Journal of Machine Learning Research*, 11:2241–2259, 2010.

G. Raskutti, M. J. Wainwright, and B. Yu. Minimax rates of estimation for high-dimensional linear regression over $\ell_q$-balls. *IEEE Trans. on Information Theory*, 57 (10):6976–6994, 2011.

P. A. Romero, T. M. Tran, and A. R. Abate. Dissecting enzyme function with microfluidic-based deep mutational scanning. *Proceedings of the National Academy of Sciences of the United States of America*, 112(23):7159–7164, 2015.

N. Simon and R. Tibshirani. Standardization and the Group Lasso Penalty. *Statistica Sinica*, 22(3):1–21, 2012.

R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 74(2):245–266, 2012.

A. W. van der Vaart and J. Wellner. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer Series in Statistics. Springer, 1996. ISBN 9780387946405.

G. Ward, T. Hastie, S. Barry, J. Elith, and J. R. Leathwick. Presence-only data and the em algorithm. *Biometrics*, 65(2):554–563, 2009.

C. F. J. Wu. On the convergence properties of the em algorithm. *Ann. Statist.*, 11(1): 95–103, 03 1983.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Statist. Soc. B*, 68(1):49–67, 2006.

W. I. Zangwill. *Nonlinear programming: a unified approach*. Prentice-Hall international series in management. Prentice-Hall, 1969.

# SUPPLEMENTARY MATERIAL

## 1.1 Proof of Proposition 1

We first discuss the proof for Algorithm (1). Define $Q, H, \widetilde{Q}$ as follows:

$$\begin{aligned} Q(\theta;\theta^m) &:= \frac{1}{n}E_{\theta^m}[\log L_f(\theta)|z,x] \\ H(\theta;\theta^m) &:= \frac{1}{n}E_{\theta^m}[\log P_\theta(y|z,x)|z,x] \\ \widetilde{Q}(\theta;\theta^m) &:= -Q(\theta;\theta^m) + P_\lambda(\theta). \end{aligned}$$

Note that for any $\theta^m$, $\mathcal{L}(\theta) = \widetilde{Q}(\theta;\theta^m) + H(\theta;\theta^m)$ holds and $H(\theta^m;\theta^m) \geq H(\theta;\theta^m)$ by Jensen's inequality. Also since $\theta^{m+1}$ is a (unique) minimizer of $\widetilde{Q}(\theta;\theta^m)$, we have

$$\mathcal{L}(\theta^{m+1}) = \widetilde{Q}(\theta^{m+1};\theta^m) + H(\theta^{m+1};\theta^m) \leq \widetilde{Q}(\theta^m;\theta^m) + H(\theta^m;\theta^m) = \mathcal{L}(\theta^m). \tag{42}$$

Now we show that the inequality is strict if $\theta^m \notin \mathcal{S}$.

$$\nabla\mathcal{L}(\theta) = -\nabla Q(\theta;\theta^m) + \nabla P_\lambda(\theta) + \nabla H(\theta;\theta^m).$$

Since $\theta^m$ is a maximizer of $H(\cdot;\theta^m)$, $\nabla H(\theta^m;\theta^m) = 0$. Thus $\nabla\mathcal{L}(\theta^m) = -\nabla Q(\theta^m;\theta^m) + \nabla P_\lambda(\theta^m) = \nabla\widetilde{Q}(\theta^m;\theta^m)$ and by (15), $\theta^m$ is not a stationary point of $\widetilde{Q}(\cdot;\theta^m)$. Since $\theta^{m+1}$ minimizes $\widetilde{Q}(\cdot;\theta^m)$, $\widetilde{Q}(\theta^{m+1};\theta^m) < \widetilde{Q}(\theta^m;\theta^m)$. Then the result follows from (42). Finally we show that $\{\mathcal{L}(\theta^m)\}_{m=0}^\infty$ converges to a stationary point of $\mathcal{L}$. Using the same argument as in Wu [1983], we appeal to the global convergence theorem stated below as Theorem 1.3 in Zangwill [1969] with $A = \widetilde{Q}, \Gamma = \mathcal{S}$ and $\alpha = \mathcal{L}$. Condition (iii) in Theorem 1.3 follows from the continuity of $\widetilde{Q}(\theta,\theta')$ in both $\theta,\theta'$. Thus, if we show $\{\theta^m\}_{m=0}^\infty$ lie in a compact set in $\mathbb{R}^p$, the result follows. Let $\theta := [\theta_0, \theta_1 \ldots, \theta_{p-1}]^T$ and $\theta_{-0} := [\theta_1 \ldots, \theta_{p-1}]^T$. Firstly, we state the following lemma.

**Lemma 1.1.** *$\{\theta_0; \mathcal{L}(\theta_0, \ldots, \theta_{p-1}) \leq \mathcal{L}(\theta_{null})\}$ is compact for any given $\theta_{-0}$ where $\theta_{null} = (\log \frac{\pi}{1-\pi}, 0, 0, ..., 0)^T$.*

*Proof.* We claim $\lim_{\theta_0 \to -\infty} \mathcal{L}(\theta) = +\infty$ and $\lim_{\theta_0 \to \infty} \mathcal{L}(\theta) = c(\theta_{-0}) > \mathcal{L}(\theta_{null})$. Then the result follows from the continuity of $\mathcal{L}(\theta)$. Defining $u_i(\theta_{-0})$ as $u_i(\theta_{-0}) := x_{i,1}\theta_1 + \cdots + x_{i,p-1}\theta_{p-1}$,

$$\mathcal{L}(\theta) = -\frac{1}{n}\left\{\sum_{i;z_i=0} \log(1-p_i(\theta)) + \sum_{i;z_i=1} \log p_i(\theta)\right\} + \lambda \sum_{j=1}^{J} \sqrt{|K_j|}\|\theta_{K_j}\|_2$$

where $p_i(\theta) = \dfrac{\frac{n_\ell}{\pi n_\ell} e^{\theta_0 + u_i(\theta_{-0})}}{1 + (1 + \frac{n_\ell}{\pi n_u})e^{\theta_0 + u_i(\theta_{-0})}}$. Since

$$\lim_{\theta_0 \to -\infty} p_i(\theta) = 0, \lim_{\theta_0 \to \infty} p_i(\theta) = \frac{n_\ell}{n_\ell + \pi n_u}$$

, $\lim_{\theta_0 \to -\infty} \mathcal{L}(\theta) = \infty$ if there exists at least one $z_i$ such that $z_i = 1$.
Defining $g(p) := -\left(\bar{z} \log \dfrac{p}{1-p} + n \log(1-p)\right)$,

$$\lim_{\theta_0 \to \infty} \mathcal{L}(\theta) = g\left(\frac{n_\ell}{n_\ell + \pi n_u}\right) + \lambda \sum_{j=1}^{J} \sqrt{|K_j|}\|\theta_{K_j}\|_2 \geq g\left(\frac{n_\ell}{n_\ell + \pi n_u}\right) > g\left(\frac{n_\ell}{n_\ell + n_u}\right)$$

. The last strict inequality comes from the fact that $g(p)$ achieves a unique minimum at $p = \bar{z} = \dfrac{n_l}{n_l + n_u}$. We note that $\mathcal{L}(\theta_{null}) = g\left(\dfrac{n_l}{n_l + n_u}\right)$, and a closed form of $\theta_{null}$ can be obtained by solving the equation $\bar{z} = \dfrac{\frac{n_l}{\pi n_u} e^{\theta_0}}{1 + (1 + \frac{n_l}{\pi n_u})e^{\theta_0}}$ for $\theta_0$. □

Next we proceed to prove that iterates $\{\theta^m\}_{m=0}^{\infty}$ stay in a compact set.

**Lemma 1.2.** *$\{\theta^m\}_{m=0}^{\infty}$ lies in a compact set in $\mathbb{R}^p$.*

*Proof.* By duality of the lasso problem, $\{\theta_{-0}^m\}_{m=0}^{\infty}$ are constrained in the space

$$K(\lambda) := \{(\theta_1^m \ldots, \theta_{p-1}^m); \sum_{j=1}^{J} \sqrt{|K_j|}\|\theta_{K_j}^m\|_2 \leq t(\lambda)\}$$

for some $t(\lambda)$ . Then,

$$\begin{aligned}\{\theta_0^m\} &= \{\theta_0^m; \mathcal{L}(\theta^m) \leq \mathcal{L}(\theta_{null})\} \\ &\subset \{\theta_0^m; \inf_{\theta_{-0}^m \in K(\lambda)} \mathcal{L}(\theta_0^m, \theta_{-0}^m) \leq \mathcal{L}(\theta_{null})\} \\ &= \{\theta_0^m; \mathcal{L}(\theta_0^m, \widetilde{\theta}_{-0}) \leq \mathcal{L}(\theta_{null})\} \qquad (43)\end{aligned}$$

for some $\widetilde{\theta}_{-0} \in K(\lambda)$ such that $\inf_{\theta_{-0}^m \in K(\lambda)} \mathcal{L}(\theta_0^m, \theta_{-0}^m) = \mathcal{L}(\theta_0^m, \widetilde{\theta}_{-0})$. The existence of such $\widetilde{\theta}$ is guaranteed by the extreme value theorem and continuity in $\theta$. Applying Lemma 1.1 to (43), $\{\theta_0^m\}$ is compact. The desired result follows as the product of compact set is compact as well. □

For Algorithm 2 (PUlasso algorithm), since $\theta^{m+1}$ is a (unique) minimizer of $-\overline{Q}(\theta;\theta^m) + P_\lambda(\theta)$, combining with (14), we have

$$\begin{aligned}\mathcal{L}(\theta^{m+1}) &= -Q(\theta^{m+1};\theta^m) + P_\lambda(\theta^{m+1}) + H(\theta^{m+1};\theta^m) \\ &\leq -\overline{Q}(\theta^{m+1};\theta^m) + P_\lambda(\theta^{m+1}) + H(\theta^m;\theta^m) \\ &\leq -\overline{Q}(\theta^m;\theta^m) + P_\lambda(\theta^m) + H(\theta^m;\theta^m) \\ &= -Q(\theta^m;\theta^m) + P_\lambda(\theta^m) + H(\theta^m;\theta^m) = \mathcal{L}(\theta^m).\end{aligned}$$

The strict inequality follows from the fact that $\nabla Q(\theta^m) = \nabla \overline{Q}(\theta^m)$. The convergence of $\{\mathcal{L}(\theta^m)\}_{m=0}^\infty$ by applying 1.3 and verifying conditions (i)-(iii) similarly as in the proof for the Algorithm 1. Specifically, condition (i) is satisfied, as lemma 1.1, 1.2 are directly applicable as proofs of Lemma 1.1, 1.2 do not depend on the specific form of majorizing function. We can take $A = -\overline{Q} + P_\lambda, \Gamma = \mathcal{S}$ and $\alpha = \mathcal{L}$ for condition (ii). Condition (iii) follows from the fact that the function $-\overline{Q}(\theta;\theta') + P_\lambda(\theta)$ is jointly continuous in $(\theta, \theta')$.

**Theorem 1.3** (Global Convergence Theorem, Zangwill [1969]). *Let the sequence $\{x_k\}_{k=0}^{\infty}$ be generated by $x_{k+1} \in A(x_k)$, where $A$ is a point-to-set map on $X$. Let a solution set $\Gamma \in X$ be given, and suppose that:*

*(i) The sequence $\{x_k\}_{k=0}^{\infty} \subset S$ for $S \subset X$ a compact set.*

*(ii) There is a continuous function $\alpha$ on $X$ such that (a) if $x \notin \Gamma$, then $\alpha(y) < \alpha(x)$ for all $y \in A(x)$. (b) if $x \in \Gamma$, then $\alpha(y) \leq \alpha(x)$ for all $y \in A(x)$.*

*(iii) The mapping $A$ is closed at all points of $X \setminus \Gamma$.*

*Then all the limit points of any convergent subsequence of $\{x_k\}_{k=0}^{\infty}$ are in the solution set $\Gamma$ and $\alpha(x_k)$ converges monotonically to $\alpha(x)$ for some $x \in \Gamma$.*

## 1.2 Proof of Theorem 3.3

Taking a derivate with respect to $\theta$ of $\mathcal{R}_n(\theta)$, we obtain

$$\nabla \mathcal{R}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \left(-z_i + \mu(f_\theta(x_i))\right) \dot{f}_\theta(x_i)$$

and

$$(\nabla \mathcal{R}_n(f_\theta) - \nabla \mathcal{R}_n(f_\theta^*))^T \Delta = \left(\frac{1}{n}\sum_{i=1}^{n} \left(\mu(f_\theta(x_i)) - z_i\right) \dot{f}_\theta(x_i) - \left(\mu(f_{\theta^*}(x_i)) - z_i\right) \dot{f}_{\theta^*}(x_i)\right)^T \Delta \tag{44}$$

where we define

$$\Delta := \theta - \theta^* \tag{45}$$

$$A(s) := \log(1 + e^s), \mu(s) = A'(s) \tag{46}$$

$$f_\theta(x) = \log \frac{n_\ell}{\pi n_u} + x^T\theta - \log(1 + e^{x^T\theta}) \tag{47}$$

and $\dot{f}_\theta$ as the derivative of $f_\theta$ with respect to $\theta$.

To prove that LHS of (44) is positive with high probability, we decompose RHS into the addition of two terms, whose first term has positive expectation and the second has expectation zero. To do so, we add and subtract $\frac{1}{n}\sum_{i=1}^n(\mu(f_{\theta^*}(x_i)) - z_i)\dot{f}_\theta(x_i)$ to the RHS of (44) to obtain

$$RHS = \frac{1}{n}\sum_{i=1}^n (\mu(f_\theta(x_i)) - \mu(f_{\theta^*}(x_i)))\dot{f}_\theta(x_i)^T\Delta + (\mu(f_{\theta^*}(x_i)) - z_i)(\dot{f}_\theta(x_i) - \dot{f}_{\theta^*}(x_i))^T\Delta.$$

Applying a Taylor expansion around $\theta^*$, we obtain

$$\begin{aligned}(\nabla\mathcal{R}_n(f_\theta) - \nabla\mathcal{R}_n(f_\theta^*))^T\Delta &= \underbrace{\frac{1}{n}\sum_{i=1}^n A''(f_{\theta^*}(x_i) + v_i(f_\theta(x_i) - f_{\theta^*}(x_i)))(f_\theta(x_i) - f_{\theta^*}(x_i))\dot{f}_\theta(x_i)^T\Delta}_{\text{I}} \\ &\quad + \underbrace{\frac{1}{n}\sum_{i=1}^n(\mu(f_{\theta^*}(x_i)) - z_i)(\dot{f}_\theta(x_i) - \dot{f}_{\theta^*}(x_i))^T\Delta}_{\text{II}} \text{ for } v_i \in [0,1].\end{aligned}$$

We use a similar argument to Proposition 2 in Negahban et al. [2012] to obtain a lower bound for the first term. The main difference is that we get the dependence on $\theta$ for a curvature term, which is not the case for a canonical link $f_\theta(x) = \theta^T x$. Since

$$f_\theta(x) - f_{\theta^*}(x) = x^T\theta - \log(1 + e^{x^T\theta}) - (x^T\theta^* - \log(1 + e^{x^T\theta^*})) = \frac{x^T(\theta - \theta^*)}{1 + e^{x^T\theta^* + vx^T(\theta-\theta^*)}} \quad (48)$$

for some $v \in [0,1]$ and $\dot{f}_\theta(x) = \dfrac{x}{1 + e^{x^T\theta}}$, $I$ becomes

$$I = \frac{1}{n}\sum_{i=1}^n A''(f_{\theta^*}(x_i) + v_i(f_\theta(x_i) - f_{\theta^*}(x_i)))\frac{(x^T\Delta)^2}{(1 + e^{x_i^T\theta^* + v_i x_i^T\Delta})(1 + e^{x_i^T\theta})}.$$

Now we lower bound $I$ by

$$\frac{1}{n}\sum_{i=1}^n \frac{A''(f_{\theta^*}(x_i) + v_i(f_\theta(x_i) - f_{\theta^*}(x_i)))(x^T\Delta)^2}{(1 + e^{x_i^T\theta^* + v_i x_i^T\Delta})(1 + e^{x_i^T\theta})}\mathbb{1}\{|\theta^{*T}x_i| \le T\|\theta^*\|_2, |\Delta^T x_i| \le \tau\|\Delta\|_2\}$$

for any $\tau, T \geq 0$, as $A''(u) \geq 0, \forall u$. Defining $L_0$ by

$$L_0 := \inf_{|u| \leq K_2 + \min\{T\|\theta^*\|_2 + \tau r, 3K\}} \frac{A''(u)}{(1 + e^{\min\{T\|\theta^*\|_2 + \tau r, 3K\}})^2},$$

$I$ can be further lower-bounded by

$$\frac{L_0}{n} \sum_{i=1}^{n} (x^T \Delta)^2 \mathbb{1}\{|\theta^{*T} x_i| \leq T\|\theta^*\|_2, |\Delta^T x_i| \leq \tau\|\Delta\|_2\},$$

since on $\{|\theta^{*T} x_i| \leq T\|\theta^*\|_2, |\Delta^T x_i| \leq \tau\|\Delta\|_2\}$

$$\theta^T x_i \leq |\theta^{*T} x_i| + |\Delta^T x_i| \leq \min\{T\|\theta^*\|_2 + \tau r, 3K\}$$

and

$$\begin{aligned} |f_{\theta^*}(x_i) + v_i(f_{\theta^*}(x_i) - f_\theta(x_i))| &\leq |f_{\theta^*}(x_i)| + |(f_{\theta^*}(x_i) - f_\theta(x_i))| \\ &\leq \left|\log \frac{n_l}{\pi n_u}\right| + |\theta^{*T} x_i| + |\Delta^T x_i| \\ &\leq K_2 + |\theta^{*T} x_i| + |\Delta^T x_i| \end{aligned}$$

by Assumption 3, equation (48) and the fact that $x^T\theta - \log(1 + e^{x^T\theta})$ is 1-Lipschitz in $\theta$.

For a truncation level $\tau > 0$, we define the following function:

$$\varphi_\tau(u) = \begin{cases} u^2 & \text{if } |u| \leq \frac{\tau}{2} \\ (\tau - u)^2 & \text{if } \frac{\tau}{2} \leq |u| \leq \tau \\ 0 & \text{otherwise} \end{cases}$$

We now lower-bound $I$ by $\frac{1}{n}\sum_{i=1}^{n} L_0 \varphi_{\tau\|\Delta\|_2}(\Delta^T x_i \mathbb{1}\{|\theta^{*T} x_i| \leq T\|\theta^*\|_2\})$ using the fact that

$$(\Delta^T x_i)^2 \mathbb{1}\{|\theta^* x_i| \leq T\|\theta^*\|_2, |\Delta^T x_i| \leq \tau\|\Delta\|_2\} \geq \varphi_{\tau\|\Delta\|_2}(\Delta^T x_i \mathbb{1}\{|\theta^{*T} x_i| \leq T\|\theta^*\|_2\})$$

since $(\Delta^T x_i)^2 \leq \tau \|\Delta\|_2 \geq \varphi_{\tau\|\Delta\|_2}(\Delta^T x_i \mathbb{1}\{|\theta^{*T} x_i| \leq T\|\theta^*\|_2\})$ is true on the event

$$|\theta^* x_i| \leq T\|\theta^*\|_2, |\Delta^T x_i| \leq \tau\|\Delta\|_2 \tag{49}$$

and both left and right-hand sides are 0 if the event (49) does not hold. Now we show that the expectation of $\frac{1}{n}\sum_{i=1}^n \varphi_{\tau\|\Delta\|_2}(\Delta^T x_i \mathbb{1}\{|\theta^{*T} x_i| \leq T\|\theta^*\|_2\})$ is strictly positive with a suitably chosen $\tau$ and $T$.

$$E\left[\varphi_{\tau\|\Delta\|_2}(\Delta^T x \mathbb{1}\{|\theta^{*T} x| \leq T\|\theta^*\|_2\})\right] = E[(\Delta^T x)^2] - E[(\Delta^T x)^2 - \varphi_{\tau\|\Delta\|_2}(\Delta^T x \mathbb{1}\{|\theta^{*T} x| \leq T\|\theta^*\|_2\})]$$

$$E[(\Delta^T x)^2] \geq \lambda_{min}(\Sigma_x)\|\Delta\|_2^2$$

and

$$\begin{aligned}&E[(\Delta^T x)^2 - \varphi_{\tau\|\Delta\|_2}(\Delta^T x \mathbb{1}\{|\theta^{*T} x| \leq T\|\theta^*\|_2\})] \\ &\leq E\left[(\Delta^T x)^2 \mathbb{1}\left\{|\Delta^T x| \geq \frac{\tau\|\Delta\|_2}{2}\right\}\right] + E\left[(\Delta^T x)^2 \mathbb{1}\{|\theta^{*T} x| \geq T\|\theta^*\|_2\}\right].\end{aligned}$$

Applying the Cauchy-Schwarz inequality, we obtain

$$\begin{aligned}E\left[(\Delta^T x)^2 \mathbb{1}\{|\Delta^T x| \geq \frac{\tau\|\Delta\|_2}{2}\}\right] &\leq \sqrt{E(\Delta^T x)^4}\sqrt{P(|\Delta^T x| \geq \frac{\tau\|\Delta\|_2}{2})} \\ E\left[(\Delta^T x)^2 \mathbb{1}\{|\theta^{*T} x| \geq T\|\theta^*\|_2\}\right] &\leq \sqrt{E(\Delta^T x)^4}\sqrt{P(|\theta^{*T} x| \geq T\|\theta^*\|_2)}.\end{aligned}$$

Since $\Delta^T x \sim \text{subG}(\|\Delta\|_2 \sigma_x), \theta^{*T} x \sim \text{subG}(\|\theta^*\|_2 \sigma_x)$,

$$\sqrt{E(\Delta^T x)^4} \leq 4\|\Delta\|_2^2 \sigma_x^2$$

$$\sqrt{E(\Delta^T x)^4}\sqrt{P(|\Delta^T x| \geq \frac{\tau\|\Delta\|_2}{2})} \leq 4\sqrt{2}\|\Delta\|_2^2 \sigma_x^2 \exp\left(-\frac{\tau^2}{16\sigma_x^2}\right)$$

$$\sqrt{E(\Delta^T x)^4}\sqrt{P(|\theta^{*T} x| \geq T)} \leq 4\sqrt{2}\|\Delta\|_2^2 \sigma_x^2 \exp\left(-\frac{T^2}{4\sigma_x^2}\right)$$

Set $\tau^2 = 16\sigma_x^2 log \frac{32\sqrt{2}\sigma_x^2}{\lambda_{min}(\Sigma_x)}$ and $T^2 = 4\sigma_x^2 \log \frac{32\sqrt{2}\sigma_x^2}{\lambda_{min}(\Sigma_x)}$ ensures

$$4\sqrt{2}\left(\exp\left(-\frac{\tau^2}{16\sigma_x^2}\right) + \exp\left(-\frac{T^2}{4\sigma_x^2}\right)\right) \leq \frac{\lambda_{min}(\Sigma_x)}{4}$$

. Then,

$$\begin{aligned} &E\left[\varphi_{\tau\|\Delta\|_2}(\Delta^T x \mathbb{1}\{|\theta^{*T}x| \leq T\|\theta^*\|_2\})\right] \\ &\geq \|\Delta\|_2^2\left(\lambda_{min}(\Sigma_x) - 4\sqrt{2}\sigma_x^2\left(\exp\left(-\frac{\tau^2}{16\sigma_x^2}\right) + \exp\left(-\frac{T^2}{4\sigma_x^2}\right)\right)\right) \\ &\geq \|\Delta\|_2^2 \frac{3\lambda_{min}(\Sigma_x)}{4}. \end{aligned}$$

We now prove that the first term, which is a random quantity is also positive with high probability using the concentration property of an empirical process. Considering the set $S(t) := \{\Delta : \frac{\|\Delta\|_1}{\|\Delta\|_2} = t, \|\Delta\|_2 = \delta\}$, we define

$$U_1(t) := \sup_{\Delta \in S(t)} \left| \frac{1}{n}\sum_{i=1}^n \varphi_{\tau\|\Delta\|_2}(\Delta^T x_i \mathbb{1}\{|\theta^{*T}x_i| \leq T\|\theta^*\|_2\}) - E\left[\varphi_{\tau\|\Delta\|_2}(\Delta^T x \mathbb{1}\{|\theta^{*T}x| \leq T\|\theta^*\|_2\})\right] \right|.$$

Since $\|\varphi_{\tau\|\Delta\|_2}\|_\infty \leq \frac{(\tau\|\Delta\|_2)^2}{4} \leq \frac{\tau^2 r^2}{4}$, using the Azuma-Hoeffding inequality, we have

$$P(U_1(t) \geq EU_1(t) + u_1^*(t)) \leq \exp(-\frac{2nu_1^*(t)^2}{r^4\tau^4}).$$

Setting $u_1^*(t) = \frac{\lambda_{min}(\Sigma_x)\|\Delta\|_2^2}{8}$,

$$P(U_1(t) \geq EU_1(t) + \frac{\lambda_{min}(\Sigma_x)}{8}\|\Delta\|_2^2) \leq \exp(-cn)$$

where $c$ constant depending on $\lambda_{min}(\Sigma_x), r, \tau, \|\Delta\|_2$.

Now we calculate $EU_1(t)$. First we introduce the following two results from empirical processe theory.

**Theorem 1.4.** *(Symmetrization theorem[Lemma 2.3.6 in van der Vaart and Wellner [1996]]) Let $U_1, \dots, U_n$ be independent random variables vith values in $\mathcal{U}$ and $\{\epsilon_i\}$ be an i.i.d. sequence of Rademacher variables, which take values $\pm 1$ each with probability 1/2. Let $\Gamma$ be a class of real-valued functions on $\mathcal{U}$. then*

$$E\left(\sup_{\gamma\in\Gamma}\left|\sum_{i=1}^{n}\{\gamma(U_i)-E(\gamma(U_i))\}\right|\right)\leq 2E\left(\sup_{\gamma\in\Gamma}\left|\sum_{i=1}^{n}\epsilon_i\gamma(U_i)\right|\right).$$

**Theorem 1.5.** *(Contraction theorem[Theorem 4.12 in Ledoux and Talagrand [2013]]) Let $\varphi_i : \mathbb{R} \to \mathbb{R}$ be contractions which satisfy $|\varphi_i(s) - \varphi_i(t)| \leq |s - t|$ and $\varphi_i(0) = 0, \forall i \leq N$ Then for any bounded subeset $T$ in $\mathbb{R}^N$,*

$$E\left(\sup_{t\in T}\left|\sum_{i=1}^{n}\epsilon_i\varphi_i(t_i)\right|\right)\leq 2E\left(\sup_{t\in T}\left|\sum_{i=1}^{n}\epsilon_i t_i\right|\right).$$

By Theorems 1.4,1.5 and Lemma 1.6 we have,

$$\begin{aligned}
E[U_1(t)] &\leq 2E\left[\sup_{\Delta\in S(t)}\left|\ \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\varphi_{\tau\|\Delta\|_2}(\Delta^T x_i \mathbb{1}\{|\theta^{*T}x_i|\leq T\|\theta^*\|_2\})\right|\right]\\
&\leq 4\tau\|\Delta\|_2 E\left[\sup_{\Delta\in S(t)}\left|\ \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\Delta^T x_i\mathbb{1}\{|\theta^{*T}x_i|\leq T\|\theta^*\|_2\}\right|\right]\\
&\leq 4\tau\|\Delta\|_2\left(\sup_{\Delta\in S(t)}\|\Delta\|_1\right)E\left[\|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i x_i\mathbb{1}\{|\theta^{*T}x_i|\leq T\|\theta^*\|_2\}\|_\infty\right]\\
&\leq 4c_1\tau t\|\Delta\|_2^2\sqrt{\frac{\log p}{n}}\ \text{w.p. at least } 1-2\exp(-cn).
\end{aligned}$$

Note that $\varphi_{\tau\|\Delta\|_2}$ is a Lipschitz function with the Lipschitz constant $= \tau\|\Delta\|_2$ which allows us to apply the Ledoux-Talagrand contraction theorem. Also let $\sup\limits_{\Delta\in S} \leftrightarrow \sup\limits_{\Delta^T x_i\in T}$ for a given $(x_1, \dots, x_n)$, where T is defined as $T := T_1 \times \dots T_n$ and $T_i := \{\Delta^T x_i : \Delta \in S\}$. Ledoux-Talagrand contraction theorem requires T to be a bounded subset of $R^n$ which is satisfied because $|\Delta^T x_i| \leq \tau r, \forall i$.

**Lemma 1.6.** *$E\left[\|\frac{1}{n}\sum_{i=1}^n \epsilon_i x_i \mathbb{1}\{|\theta^{*T}x_i| \leq T\|\theta^*\|_2\}\|_\infty\right] \leq c_1\sqrt{\frac{\log p}{n}}$ w.p. at least $1-2\exp(-c'n)$.*

*Proof.* Define $u_{ij} := x_{ij}\mathbb{1}\{|\theta^{*T}x_i| \leq T\|\theta^*\|_2\}$. Also consider the event $\mathcal{T} := \{\frac{1}{n}\sum_{i=1}^n u_{ij}^2 \leq C, \forall 1 \leq j \leq p\}$. First, we show that the probability of $\mathcal{T}$ is large:

$$P(\mathcal{T}) \geq 1-pP(\frac{1}{n}\sum_{i=1}^n u_{ij}^2 > C) \geq 1-pP(\frac{1}{n}\sum_{i=1}^n x_{ij}^2 > C) \geq 1-2\exp(-cn+\log p) \geq 1-2\exp(-c'n).$$

Since we know $x_{ij}^2$'s are i.i.d. sub-exponential random variables, letting $K := \|x_{ij}^2\|_{\psi_1}$, we have $P(\sum_i x_{ij}^2 > Cn) \leq 2\exp[-c''\min(\frac{C^2n^2}{K^2n}, \frac{Cn}{K})]$. The last inequality follows since $n \gg \log p$

Now we show the suggested inequality is true on $\mathcal{T}$

$$E\left[\|\frac{1}{n}\sum_{i=1}^n \epsilon_i x_i \mathbb{1}\{|\theta^{*T}x_i| \leq T\|\theta^*\|_2\}\|_\infty|\mathcal{T}\right] = E\left[E\left[\|\frac{1}{n}\sum_{i=1}^n \epsilon_i x_i \mathbb{1}\{|\theta^{*T}x_i| \leq T\|\theta^*\|_2\}\|_\infty|X,\mathcal{T}\right]|\mathcal{T}\right].$$

Conditioned on $\{X_i\}_{i=1}^n$, $|\frac{1}{n}\sum_{i=1}^n \epsilon_i u_{ij}|$ is sub-gaussian with a parameter bounded by $\frac{1}{n}\sqrt{\sum_i u_{ij}^2}$, since $\epsilon_i \sim \text{subG}(1)$.

$$E\left[\|\frac{1}{n}\sum_{i=1}^n \epsilon_i x_i \mathbb{1}\{|\theta^{*T}x_i| \leq T\|\theta^*\|_2\}\|_\infty|X,\mathcal{T}\right] \leq c_0\frac{1}{n}\max_{1\leq j\leq p}\sqrt{\sum_i u_{ij}^2}\sqrt{\log p} \leq c_1\sqrt{\frac{\log p}{n}}.$$

□

For the second term, define

$$\begin{aligned} g(\Delta^T x_i, \theta^{*T}x_i, z_i) &:= (\mu(f_{\theta^*}(x_i)) - z_i)(\dot{f}_{\theta^*+\Delta}(x_i) - \dot{f}_{\theta^*}(x_i))^T\Delta \\ &= \left(\frac{\frac{n_l}{\pi n_u}e^{\theta^{*T}x_i}}{1+(1+\frac{n_l}{\pi n_u})e^{\theta^{*T}x_i}}\right)^{1-z_i}\left(\frac{-1-e^{\theta^{*T}x_i}}{1+(1+\frac{n_l}{\pi n_u})e^{\theta^{*T}x_i}}\right)^{z_i} \\ &\qquad \times\left(\frac{1}{1+e^{(\theta^*+\Delta)^T x_i}} - \frac{1}{1+e^{\theta^{*T}x_i}}\right)x_i^T\Delta. \end{aligned}$$

Then

$$II = \frac{1}{n}\sum_{i=1}^{n} g(\Delta^T x_i, \theta^{*T} x_i, z_i) = \frac{1}{n}\sum_{i=1}^{n} \left\{ g(\Delta^T x_i, \theta^{*T} x_i, z_i) - E[g(\Delta^T x_i, \theta^{*T} x_i, z_i)] \right\}$$

, since $E[g(\Delta^T x_i, \theta^{*T} x_i, z_i)] = 0, \forall i$, by $E[Z|X = x] = \mu(f^*_\theta(x))$.

$$U_2(t) := \sup_{\Delta \in S(t)} \left| \frac{1}{n}\sum_{i=1}^{n} \left\{ g(\Delta^T x_i, \theta^{*T} x_i, z_i) - E[g(\Delta^T x_i, \theta^{*T} x_i, z_i)] \right\} \right|$$

. Similarly, we bound $E(U_2(t))$ using symmetrization and contraction theorem. We first prove that $g_i/L$ is a contraction map with a fixed constant $L$, where $g_i$ is defined as $g_i(\cdot) := g(\cdot, \theta^{*T} x_i, z_i)$.

**Lemma 1.7.** *$g_i(s)/L$ is a contraction map with $g_i(0) = 0$.*

*Proof.* Sufficient to show $|g_i(s) - g_i(t)| \le L|s - t|$ with $L > 0, g_i(0) = 0$

$$\begin{aligned} |g_i(s) - g_i(t)| &= \left| h_{i,\theta^*} \left\{ \left( \frac{s}{1 + e^{\theta^{*T} x_i + s}} - \frac{t}{1 + e^{\theta^{*T} x_i + t}} \right) - \frac{(s-t)}{1 + e^{\theta^{*T} x_i}} \right\} \right| && (50) \\ &\le |h_{i,\theta^*}| \left| \left\{ \frac{1 - (t + v_i(s-t) - 1) e^{\theta^{*T} x_i + t + v_i(s-t)}}{(1 + e^{\theta^{*T} x_i + t + v_i(s-t)})^2} - \frac{1}{1 + e^{\theta^{*T} x_i}} \right\} \right| |(s-t)| && (51) \end{aligned}$$

for some $v_i \in [0, 1]$. (51) follows from the Mean Value Theorem since

$$\frac{s}{(1 + e^{\theta^{*T} x_i + s})} = \frac{t}{(1 + e^{\theta^{*T} x_i + t})} + \frac{1 - e^{\theta^{*T} x_i + \tilde{s}}(\tilde{s} - 1)}{(1 + e^{\theta^{*T} x_i + t})^2}.$$

We bound the coefficient term independent of $i$.

$$
\begin{aligned}
&|\left|\left\{\frac{1-(t+v_i(s-t)-1)e^{\theta^{*T}x_i+t+v_i(s-t)}}{(1+e^{\theta^{*T}x_i+t+v_i(s-t)})^2}-\frac{1}{1+e^{\theta^{*T}x_i}}\right\}\right| \\
&\leq |\left|\frac{1}{(1+e^{\theta^{*T}x_i+t+v_i(s-t)})^2}\right|+\left|\frac{(t+v_i(s-t)-1)e^{\theta^{*T}x_i+t+v_i(s-t)}}{(1+e^{\theta^{*T}x_i+t+v_i(s-t)})^2}\right|+\left|\frac{1}{1+e^{\theta^{*T}x_i}}\right| \\
&\leq 2+\left|\frac{(\theta^{*T}x_i+t+v_i(s-t)-1)e^{\theta^{*T}x_i+t+v_i(s-t)}}{(1+e^{\theta^{*T}x_i+t+v_i(s-t)})^2}-\frac{\theta^{*T}x_ie^{\theta^{*T}x_i+t+v_i(s-t)}}{(1+e^{\theta^{*T}x_i+t+v_i(s-t)})^2}\right| \\
&\leq 2+\max_u|\frac{ue^u}{(1+e^u)^2}|+\max_{u,i}|\frac{\theta^{*T}x_ie^u}{(1+e^u)^2}|\leq\frac{9+K}{4},
\end{aligned}
$$

where $u:=\theta^{*T}x_i+t+v_i(s-t)$. Since

$$\max_u|\frac{ue^u}{(1+e^u)^2}|\leq\frac{1}{4},|h_{i,\theta^*}|\leq\frac{1}{4}$$

and

$$\max_{u,i}|\frac{\theta^{*T}x_ie^u}{(1+e^u)^2}|\leq\max_{u,i}|\theta^{*T}x_i||\frac{e^u}{(1+e^u)^2}|\leq\frac{K}{4}$$

we set lipschitz constant $L:=\dfrac{9+K}{16}$. $g_i(0)=0$ is obvious from the form of $g_i$. □

Taking $U_i=(X_i,Z_i)$ in Theorem 1.4, a symmetrization argument gives

$$E(U_2(t))\leq 2E\left[\sup_{\Delta\in S(t)}\left|\frac{1}{n}\sum_{i=1}^n\epsilon_ig(\Delta^Tx_i,\theta^{*T}x_i,z_i)\right|\right]. \tag{52}$$

Now we apply the contraction theorem with conditional expectation given $\{X_i,Z_i\}$ and $\varphi_i=g_i/L$ to find

$$E\left[\sup_{\Delta\in S(t)}\left|\frac{1}{n}\sum_{i=1}^n\epsilon_ig(\Delta^Tx_i,\theta^{*T}x_i,z_i)\right||(X,Z)\right]\leq 2LE\left[\sup_{\Delta\in S(t)}\left|\frac{1}{n}\sum_{i=1}^n\epsilon_i\Delta^Tx_i\right||(X,Z)\right]. \tag{53}$$

Combining (52) and (53),

$$\begin{aligned}E(U_2(t)) &\le 4LE\left[\sup_{\Delta\in S(t)}\left|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\Delta^T x_i\right|\right]\\ &\le 4LE\left[\sup_{\Delta\in S(t)}\|\Delta\|_1\|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i x_i\|_\infty\right]\\ &\le 4c_1Lt\|\Delta\|_2\sqrt{\frac{\log p}{n}}\text{ w.p at least }1-2\exp(-cn).\end{aligned}$$

Now we apply Azuma-Hoeffding inequality to show that $U_2(t)$ is close to $E(U_2(t))$ with probability at least $1-\exp(-c'n)$. From the proof of lemma 1.7 we have $\|g\|_\infty \le K(K+9)/8$ since

$$\|g\|_\infty = \max_{i,\theta}|g_i(x_i^T\theta)-g_i(0)| \le L|x_i^T\theta - 0| \le 2LK.$$

Hence we obtain

$$P(U_2(t) \ge EU_2(t) + u_2^*(t)) \le \exp(-cnu_2^*(t)^2)$$

for some $c$. We set $u_2^*(t) = L_0\frac{\lambda_{min}(\Sigma_x)\|\Delta\|_2^2}{8}$ to have

$$P(U_2(t) \ge EU_2(t) + L_0\frac{\lambda_{min}(\Sigma_x)\|\Delta\|_2^2}{8}) \le \exp(-c'n)$$

where $c'$ is a constant depending on $\lambda_{min}(\Sigma_x)$, $\|\Delta\|_2$ and $K$. Putting the pieces together, with probability at least $1-c_1\exp(-c_2n)$ we have

$$\begin{aligned}&(\nabla\mathcal{R}_n(f_\theta)-\nabla\mathcal{R}_n(f_\theta^*))^T\Delta\\ &\ge L_0\frac{\lambda_{min}(\Sigma_x)}{2}\|\Delta\|_2^2 - 4L_0c_1\tau t\|\Delta\|_2^2\sqrt{\frac{\log p}{n}} - 4c_1Lt\|\Delta\|_2\sqrt{\frac{\log p}{n}}\\ &= \kappa_1\|\Delta\|_2^2 - \kappa_2 t\|\Delta\|_2^2\sqrt{\frac{\log p}{n}} - \kappa_3 t\|\Delta\|_2\sqrt{\frac{\log p}{n}}\end{aligned}$$

where $\kappa_1 := L_0 \frac{\lambda_{min}(\Sigma_x)}{2}, \kappa_2 := 4L_0 c_1 \tau, \kappa_3 := 4c_1 L$.

Now we want to show that the probability of following event

$$\mathcal{E} := \left\{ \left(\nabla \mathcal{R}_n(f_\theta) - \nabla \mathcal{R}_n(f_\theta^*)\right)^T \Delta \geq \kappa_1 \|\Delta\|_2^2 - \kappa_2 \|\Delta\|_1 \|\Delta\|_2 \sqrt{\frac{\log p}{n}} - \kappa_3 \|\Delta\|_1 \sqrt{\frac{\log p}{n}} \right\} \tag{54}$$

is large uniformly in the ratio of $\frac{\|\Delta\|_1}{\|\Delta\|_2}$. Defining functions $f_\Delta(x), g(t)$ as

$$f_\Delta(x) := \kappa_1 \|\Delta\|_2^2 - \left(\nabla \mathcal{R}_n(f_\theta) - \nabla \mathcal{R}_n(f_\theta^*)\right)^T \Delta$$

$$g(t) := t \|\Delta\|_2 \sqrt{\frac{\log p}{n}} (\kappa_2 \|\Delta\|_2 + \kappa_3),$$

we have

$$P\left(f_\Delta(x) \geq g(t)\right) \leq c_1 \exp(-c_2 n).$$

By the union bound,

$$\begin{aligned} P(\mathcal{E}^c) &= \left( \bigcup_{m=1}^{M} \left\{ f_\Delta(x) \geq g(\frac{\|\Delta\|_1}{\|\Delta\|_2}), 2^{m-1} \leq \frac{\|\Delta\|_1}{\|\Delta\|_2} \leq 2^m \right\} \right) \\ &\leq \sum_{m=1}^{M} P(f_\Delta(x) \geq g(2^{m-1})) \leq M c_1 \exp(-c_2 n) \end{aligned}$$

where $M := \lceil c \log p \rceil$ since $\|\Delta\|_1 \leq \sqrt{p} \|\Delta\|_2$. By the scaling of $n \gg \log p$, we have $c_1 \exp(-c_2 n + \log M) = c_1 \exp(-c'' n)$. Note that

$$\kappa_2 \sqrt{\frac{\log p}{n}} \|\Delta\|_1 \|\Delta\|_2 \leq \frac{1}{2} \left( \kappa_1 \|\Delta\|_2^2 + \frac{\kappa_2^2}{\kappa_1} \frac{\log p}{n} \|\Delta\|_1^2 \right). \tag{55}$$

Applying (55) to (54), we have

$$\left(\nabla R_n(f_\theta) - \nabla R_n(f_\theta^*)\right)^T \Delta \geq \frac{\kappa_1}{2} \|\Delta\|_2^2 - \frac{\kappa_2^2}{2\kappa_1} \|\Delta\|_1^2 \frac{\log p}{n} - \kappa_3 \|\Delta\|_1 \sqrt{\frac{\log p}{n}} \text{ for } \|\Delta\|_2 \leq r$$

as desired.

## 1.3 Proof of Lemma 3.1

The proof of this result follows similar lines to the proof of Theorem 1 in Loh and Wainwright [2013], which established the result with a diffrent tolerance function. Since $\theta^*$ is feasible, by first order optimality condition, we have the following inequality

$$(\nabla \mathcal{R}_n(\hat{\theta}) + \nabla P_\lambda(\hat{\theta}))^T(\theta^* - \hat{\theta}) \geq 0.$$

Letting $\hat{\Delta} := \hat{\theta} - \theta^*$, since $\hat{\theta} \in \Theta_0$ by the setup of the problem, we can apply RSC condition to obtain

$$\alpha \|\hat{\Delta}\|_2^2 - \tau(\|\hat{\Delta}\|_1) \leq (-\nabla P_\lambda(\hat{\theta}) - \nabla \mathcal{R}_n(\theta^*))^T \hat{\Delta}. \tag{56}$$

On the other hand, convexity of $P_\lambda(\theta)$ implies

$$P_\lambda(\theta^*) - P_\lambda(\hat{\theta}) \geq -P_\lambda(\hat{\theta})^T \hat{\Delta}. \tag{57}$$

Combining (56)) with (57), we obtain

$$\begin{aligned}\alpha \|\hat{\Delta}\|_2^2 - \tau(\|\hat{\Delta}\|_1) &\leq (-\nabla P_\lambda(\hat{\theta}) - \nabla \mathcal{R}_n(\theta^*))^T \hat{\Delta} \\ &\leq P_\lambda(\theta^*) - P_\lambda(\hat{\theta}) + \|\nabla \mathcal{R}_n(\theta^*)\|_\infty \|\hat{\Delta}\|_1.\end{aligned}$$

Since $\tau(\|\hat{\Delta}\|_1) = \tau_1 \frac{\log p}{n} \|\hat{\Delta}\|_1^2 + \tau_2 \sqrt{\frac{\log p}{n}} \|\hat{\Delta}\|_1$,

$$\alpha \|\hat{\Delta}\|_2^2 \leq P_\lambda(\theta^*) - P_\lambda(\hat{\theta}) + \|\hat{\Delta}\|_1 \left( \tau_1 \frac{\log p}{n} \|\hat{\Delta}\|_1 + \tau_2 \sqrt{\frac{\log p}{n}} + \|\nabla \mathcal{R}_n(\theta^*)\|_\infty \right),$$

By the choice of $\lambda$,

$$\tau_1 \frac{\log p}{n} \|\hat{\Delta}\|_1 + \tau_2 \sqrt{\frac{\log p}{n}} + \|\nabla \mathcal{R}_n(\theta^*)\|_\infty \leq \frac{\lambda}{2}.$$

Then by using the triangle inequality

$$\begin{aligned}\alpha\|\hat{\Delta}\|_2^2 &\leq P_\lambda(\theta^*) - P_\lambda(\hat{\theta}) + \|\hat{\Delta}\|_1\frac{\lambda}{2} \\ &\leq \lambda\|\theta^*\|_1 - \lambda\|\hat{\theta}\|_1 + \frac{\lambda}{2}(\|\theta^*\|_1 + \|\hat{\theta}\|_1) \\ &= \frac{\lambda}{2}(3\|\theta^*\|_1 - \|\hat{\theta}\|_1).\end{aligned}$$

In particular, we have $3\|\theta^*\|_1 - \|\hat{\theta}\|_1 \geq 0$ and Lemma 5 in Loh and Wainwright [2013] gives $3\|\theta^*\|_1 - \|\hat{\theta}\|_1 \leq 3\|\hat{\Delta}_A\|_1 - \|\hat{\Delta}_{A^c}\|_1$ where A denotes the index set of the s largest elements of $\hat{\Delta}$ in magnitude. Thus we conclude

$$\alpha\|\hat{\Delta}\|_2^2 \leq \frac{\lambda}{2}(3\|\hat{\Delta}_A\|_1 - \|\hat{\Delta}_{A^c}\|_1) \leq \frac{3\lambda}{2}\|\hat{\Delta}_A\|_1 \leq \frac{3\lambda}{2}\sqrt{s}\|\hat{\Delta}\|_2$$

as desired. The $\ell_1$ upper bound follows from the $\ell_2$-bound and

$$\|\hat{\Delta}\|_1 \leq \|\hat{\Delta}_A\|_1 + \|\hat{\Delta}_{A^c}\|_1 \leq 4\|\hat{\Delta}_A\|_1 \leq 4\sqrt{s}\|\hat{\Delta}\|_2$$

.

### 1.4 Proof of Lemma 3.2

$\nabla\mathcal{R}_n(f_{\theta^*}) = \frac{1}{n}\sum_{i=1}^n \left(-z_i + \mu(f_{\theta^*}(x_i))\right)\frac{1}{1+e^{\theta^{*T}x_i}}x_i$ where $\mu, f_\theta$ as in (46), (47). For $1 \leq i \leq n$, $1 \leq j \leq p$, we define $V_{ij} := \left(-z_i + \mu(f_{\theta^*}(x_i))\right)\frac{1}{1+e^{\theta^{*T}x_i}}x_{ij}$, and consider the event

$$\mathcal{E} = \{\max_{1\leq j\leq p}\frac{1}{n}\sum_{i=1}^n x_{ij}^2 \leq C\}$$

. Then

$$P\left[\max_{1\leq j\leq p}|\frac{1}{n}\sum_{i=1}^n V_{ij}| \geq c\sqrt{\frac{\log p}{n}}\right] \leq P(\mathcal{E}^c) + P\left[\max_{1\leq j\leq p}|\frac{1}{n}\sum_{i=1}^n V_{ij}| \geq c\sqrt{\frac{\log p}{n}}|\mathcal{E}\right]P(\mathcal{E}).$$

To use the Chernoff bound, we calculate the moment generating function of $\frac{1}{n}\sum_{i=1}^n V_{ij}$. Defining $t_i := \dfrac{t}{n(1+e^{\theta^{*T}x_i})}$,

$$\begin{aligned}
E\left[\exp(\frac{t}{n}V_{ij})|x_i\right] &= E\left\{\exp\left(-t_i z_i x_{ij}\right)\cdot\exp\left(t_i\mu(f_{\theta^*}(x_i))x_{ij}\right)|x_i\right\} \\
&= E\left[\exp\left(-t_i z_i x_{ij}\right)|x_i\right]\cdot\exp\left(t_i\mu(f_{\theta^*}(x_i))x_{ij}\right) \\
&= \int\exp\left(-t_i z x_{ij}\right)\cdot\exp(zf_{\theta^*}(x_i) - A(f_{\theta^*}(x_i))dz\cdot\exp\left(t_i\mu(f_{\theta^*}(x_i))x_{ij}\right) \\
&= \exp\left\{A(f_{\theta^*}(x_i) - t_i x_{ij}) - A(f_{\theta^*}(x_i)) + t_i\mu(f_{\theta^*}(x_i))x_{ij}\right\} \\
&= \exp\left\{\frac{1}{2}A''(f_{\theta^*}(x_i) - v_i t_i x_{ij})(t_i x_{ij})^2\right\} \\
&\leq \exp\left\{\frac{1}{8n^2}(tx_{ij})^2\right\}(\because \sup_u A''(u) = \frac{1}{4}, t_i \leq \frac{t}{n}).
\end{aligned}$$

Therefore

$$\prod_{i=1}^n E\left[\exp(\frac{t}{n}V_{ij})|x_i\right] \leq \exp\left(\frac{t^2}{8n^2}\sum_{i=1}^n x_{ij}^2\right).$$

Conditioned on $\mathcal{E}$, $\exp\left(\frac{t^2}{8n^2}\sum_{i=1}^n x_{ij}^2\right) \leq \exp\left(\frac{t^2C}{8n}\right)$,

$$P\left[\max_{1\leq j\leq p}|\frac{1}{n}\sum_{i=1}^n V_{ij}| \geq c\sqrt{\frac{\log p}{n}} \quad |\mathcal{E}\right] \leq 2p\exp\left(-c''\log p\right).$$

$$\begin{aligned}
P\left[\max_{1\leq j\leq p}|\frac{1}{n}\sum_{i=1}^n V_{ij}| \geq c\sqrt{\frac{\log p}{n}}\right] &\leq 2\exp(-c'n) + 2\exp(-c''\log p)(1 - 2\exp(-c'n)) \\
&= c_1\exp(-c_2\log p).
\end{aligned}$$

## 2 Robustness to initialization

Since the PUlasso is an iterative algorithm and the objective can be non-convex, a natural question is whether the algorithm is robust to this choice. To study the effect of different initializations, we generated two sets of data in $\mathbb{R}^{100}$ following the probabilistic scheme setup in section 4 with $\mu_0 = 1.5$ and $n_l = n_u = 1000$. All 100 variables are set to be active. The model was fitted using Algorithm 2 with 10000 different initial parameters chosen randomly within radius distance 5 of $\theta_0$ and $\theta^*$. More specifically letting $r_0 := \|\theta_0 - \theta^*\|_2$, we created an equally spaced sequence of length 100 from 0 to $5r_0$. Each radius defines a contour, and we considered 100 different random $\theta$ on each contour. Figure 5 plots the number of iterations needed until convergence and deviance at convergence. The number of iterations increased as radius increased, which suggests that if we start from worse initial points, we need more iterations until convergence. In all 10000 experiments, the algorithm converged to the same stationary point.

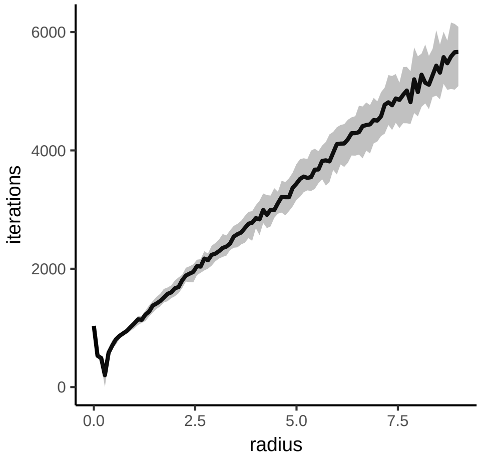

Figure 5: Iterations needed until convergence. Solid line represents the median number of iterations, and the band represents the inter-quartile range.