New Bounds for Hyperparameter Tuning of Regression Problems Across Instances

Maria-Florina Balcan
Carnegie Mellon University
ninamf@cs.cmu.edu

Anh Tuan Nguyen
Carnegie Mellon University
atnguyen@cs.cmu.edu

Dravyansh Sharma
Carnegie Mellon University
dravyans@cs.cmu.edu

Abstract

The task of tuning regularization coefficients in regularized regression models with provable guarantees across problem instances still poses a significant challenge in the literature. This paper investigates the sample complexity of tuning regularization parameters in linear and logistic regressions under ℓ_1 and ℓ_2-constraints in the data-driven setting. For the linear regression problem, by more carefully exploiting the structure of the dual function class, we provide a new upper bound for the pseudo-dimension of the validation loss function class, which significantly improves the best-known results on the problem. Remarkably, we also instantiate the first matching lower bound, proving our results are tight. For tuning the regularization parameters of logistic regression, we introduce a new approach to studying the learning guarantee via an approximation of the validation loss function class. We examine the pseudo-dimension of the approximation class and construct a uniform error bound between the validation loss function class and its approximation, which allows us to instantiate the first learning guarantee for the problem of tuning logistic regression regularization coefficients.

1 Introduction

Regularized linear models, including the Elastic Net [1], and Regularized Logistic Regression [2, 3, 4], as well as their variants [5, 6, 7], have found widespread use in diverse fields and numerous application domains. Thanks to their simplicity and interpretability, those methods are popular choices for controlling model complexity, improving robustness, and preventing overfitting by selecting relevant features [4, 8, 9]. Moreover, regularized linear models can be adapted to the non-linear regime using kernel methods [10, 11], significantly expanding their applicability to a wide range of problems. In typical applications, one needs to solve not only a single regression problem instance, but several related problems from the same domain. Can we learn how to regularize with good generalization across the related problem instances?

Suppose we have a regression dataset $(X, y) \in \mathbb{R}^{m\times p} \times \mathcal{Y}^m$, where X is a design matrix with m samples and p features, and y is a target vector. Regularized linear models aim to compute an estimator $\hat{\beta}^{(X,y)}(\lambda)$ by solving the optimization problem

$$\hat{\beta}^{(X,y)}(\lambda) = \operatorname*{argmin}_{\beta\in\mathbb{R}^p} \left[l(\beta, (X, y)) + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2\right], \tag{1}$$

where $(\lambda_1, \lambda_2) \in \mathbb{R}^2_{\geq 0}$ are the regularization coefficients. For instance, if $\lambda \in \mathbb{R}^2_{>0}$, $y \in \mathbb{R}^m$, and $l(\beta, (X, y)) = \frac{1}{2}\|y - X\beta\|_2^2$ (squared-loss function), we get the well-known Elastic Net [1]. On the other hand, if $\lambda \in \{(\lambda_1, 0), (0, \lambda_2)\}$ for $\lambda_1, \lambda_2 > 0$, $y \in \{\pm 1\}^m$, and $l(\beta, (X, y)) = \frac{1}{m}\sum_{i=1}^m \log(1 + \exp(-y_i x_i^\top \beta))$, we obtain regularized logistic regression.

Correspondence: atnguyen@cs.cmu.edu.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

In regularized linear models, the parameters λ play a crucial role in controlling the sparsity (ℓ_1) and shrinkage (ℓ_2) constraints, and are essential in ensuring better generalization and robustness [9, 4, 12]. A popular approach in practice is cross-validation, which involves choosing a finite grid of values of λ and iteratively solving the regression problem for multiple values of λ and evaluating on held-out validation sets to determine the optimal parameter. Principled techniques with theoretical guarantees suffer from various limitations, for example require strong assumptions about the original problem [13], or aim to search the optimal parameter over a discrete subset instead of the whole continuous domain. Moreover, repeatedly solving the regression problem is particularly inefficient if we have multiple problem instances from the same problem domain.

In this work, we investigate an alternative setting for tuning regularization parameters, namely *data-driven algorithm design*, following the previous line of work by Balcan et al. [14]. Unlike the traditional approach, which involves considering a single dataset (X, y), in the data-driven approach, we analyze a collection of datasets or problem instances $(X^{(i)}, y^{(i)}, X^{(i)}_{\text{val}}, y^{(i)}_{\text{val}})$ drawn from an underlying problem distribution $\mathcal{D}$. Our objective is to determine the optimal regularization parameters λ so that when using the training set $(X^{(i)}, y^{(i)})$ and λ to select a model in Optimization problem 1, the selected model minimizes loss on the validation set $(X^{(i)}_{\text{val}}, y^{(i)}_{\text{val}})$. As remarked by Balcan et al. [14], data-driven algorithm design can handle more diverse data generation scenarios in practice, including cross validation and multitask-learning [15, 16]. We emphasize that the data-driven setting differs significantly from the standard single dataset setting.

In this paper, we consider the problem of tuning regularization parameters in regularized logistic regression and the Elastic Net across multiple problem instances. Our contributions are:

- We present an improved upper bound (Theorem 3.3) on the pseudo-dimension for tuning the Elastic Net regularization parameters across problem instances by establishing a novel structural result for the validation loss function class (Theorem 3.2). We provide a crucial refinement to the piecewise structure of this function class established by Balcan et al. [14], by providing a bound on the number of distinct functional behaviors across the pieces. This enables us to describe the computation of the validation loss function as a GJ algorithm [17], which yields an upper-bound of $O(p)$ on the pseudo-dimension, a significant improvement of the prior best bound of $O(p^2)$ by Balcan et al. [14], and a corresponding improvement in the sample complexity (Theorem 3.4).
- Furthermore, we establish the tightness of our result by providing the first asymptotically matching lower bound of $\Omega(p)$ on the pseudo-dimension (Theorem 3.5). It is worth noting that our results have direct implications for other specialized cases, such as LASSO and Ridge Regression.
- We further extend our results on the Elastic Net to regularized kernel linear regression problem (Corollary 3.6).
- We propose a novel approach to analyze the problem of tuning regularization parameters in regularized logistic regression, which involves indirectly investigating an approximation of the validation loss function class. Using this approach, we instantiate the first learning guarantee for this problem in the data-driven setting (Theorem 4.4).

1.1 Related work

Model selection for regularized linear models. Extensive research has focused on the selection of optimal parameters for regularized linear models, including the Elastic Net and regularized logistic regression. This process usually entails choosing the appropriate regularization coefficients for a given dataset [18, 19]. Nevertheless, a substantial proportion of this research relies on heuristic approaches that lack theoretical guarantees [20, 21]. Others have concentrated on creating tuning objectives that go beyond validation error [22, 23], but with no clearly defined procedures for provably optimizing them. The conventional method for selecting a tuning regularization parameter is through grid-based selection, which aims to choose the parameter from a subset, known as a grid, within the parameter space. While this approach provides certain guarantees [24], it falls short in delivering an optimal solution across the entire continuous parameter space, particularly when using tuning objectives that exhibit numerous discontinuities. Additionally, the grid-based technique is highly sensitive to density, as selecting a grid that is either too dense or too coarse might result in inefficient search or highly inaccurate solutions. Other guarantees require strong assumptions on the data distribution, such as sub-Gaussian noise [25, 13]. Some studies focus on evaluating regularized linear models by

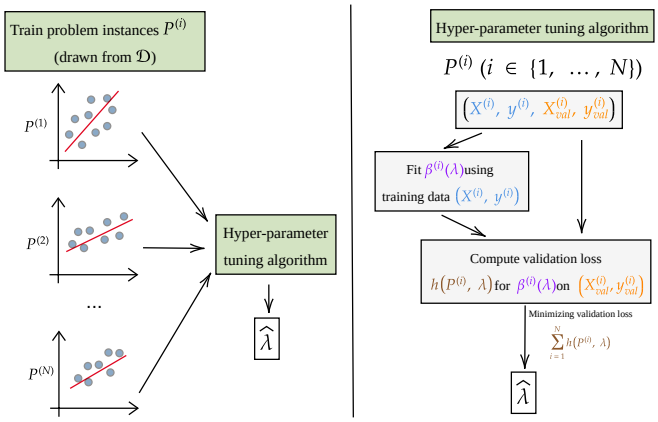

Figure 1: The process of tuning regularization parameter λ across problem instances. Given a set of N problem instances $\{P^{(1)}, \ldots, P^{(N)}\}$ drawn from some problem distribution $\mathcal{D}$, one seeks to choose the best parameter $\hat{\lambda}$ by minimizing the total validation loss $\sum_{i=1}^{N} h(P^{(i)}; \lambda)$.

constructing solution paths [26, 2, 27]. However, it is important to note that these approaches are primarily computational in nature and do not provide theoretical guarantees.

Data-driven algorithm design. Data-driven algorithms can adapt their internal structure or parameters to problem instances from unknown application-specific distributions. It is proved to be effective for a variety of combinatorial problems, such as clustering, integer programming, auction design, and graph-based semi-supervised learning [28, 29, 30, 31]. Balcan et al. [14] recently introduced a novel approach to tuning regularization parameters in regularized linear regression models, such as Elastic Net and its variants. They applied data-driven analysis to reveal the underlying discrete structure of the problem and leveraged a general result from [32] to obtain an upper bound on the pseudo-dimension of the problem. To provably tune the regularization parameters across problem instances, they proposed a simple ERM learner and provided sample complexity guarantee for such learner. However, the general techniques from [32] do not always lead to optimal bounds on the pseudodimension. Our paper is an example of a problem where these bounds (as derived in [14]) are sub-optimal, and more specialized techniques due to [31] result in the tighter bounds that we obtain. Also prior work does not establish any lower bound on the pseudodimension. Furthermore, it should be noted that their analysis heavily relies on the assumption of having a closed-form representation of the Elastic Net estimator [27]. This approach may not be applicable in analyzing other regularized linear models, such as regularized logistic regression, for which we propose an alternative approach.

2 Problem setting

In this section, we provide a formal definition of the problem of tuning regularization parameters in the Elastic Net and regularized logistic regression (RLR) across multiple problem instances, which follows the settings by Balcan et al. [14]. Given a problem instance $P = (X, y, X_{\text{val}}, y_{\text{val}})$, where $(X, y) \in \mathbb{R}^{m \times p} \times \mathcal{Y}^m$ represents the training dataset with m samples and p features, and $(X_{\text{val}}, y_{\text{val}}) \in \mathbb{R}^{m' \times p} \times \mathcal{Y}^{m'}$ denotes the validation split with m' samples, we consider the estimator $\hat{\beta}_{(X,y)}(\lambda)$ defined as:

$$\hat{\beta}_{(X,y)}(\lambda) \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} l(\beta, (X, y)) + \langle \lambda, R(\beta) \rangle, \tag{2}$$

where $l(\beta, (X, y))$ represents the objective loss function, λ denotes the regularization coefficients, and $R(\beta) = (\|\beta\|_1, \|\beta\|_2^2)$ represents the regularization vector function.

For instance, if $\lambda \in \mathbb{R}^2_{>0}$, $\mathcal{Y} \equiv \mathbb{R}$, and $l(\beta, (X, y)) = l_{\text{EN}}(\beta, (X, y)) = \frac{1}{2m}\|y - X\beta\|_2^2$, we get the well-known Elastic Net. On the other hand, if $\lambda \in \{(\lambda_1, 0), (0, \lambda_2)\}$ for $\lambda_1, \lambda_2 > 0$, $y \in \{\pm 1\}^m$, and $l(\beta, (X, y)) = l_{\text{RLR}}(\beta, (X, y)) = \frac{1}{m}\sum_{i=1}^{m} \log(1 + \exp(-y_i x_i^\top \beta))$, we obtain RLR with ℓ_1 or ℓ_2 regularization. Note that for the Elastic Net hyperparameter tuning problem, we allows the regularization coefficients of both ℓ_1, ℓ_2 are positive, while in the Regularized Logistic Regression

problem, we consider either ℓ_1 or ℓ_2 as the regularization term. We then use the validation set $(X_{\text{val}}, y_{\text{val}})$ to calculate the validation loss $h(\lambda, P) = l(\hat{\beta}_{(X,y)}(\lambda), (X_{\text{val}}, y_{\text{val}}))$ corresponding to the problem instance P and learned regularization parameters λ.

In the *data-driven setting*, we receive a collection of n problem instances $P^{(i)} = (X^{(i)}, y^{(i)}, X^{(i)}_{\text{val}}, y^{(i)}_{\text{val}}) \in \mathcal{R}_{m_i,p_i,m'_i}$ for $i \in [n]$, where $\mathcal{R}_{m_i,p_i,m'_i} := \mathbb{R}^{m_i \times p_i} \times \mathcal{Y}^{m_i} \times \mathbb{R}^{m'_i \times p_i} \times \mathcal{Y}^{m'_i}$. The problem space $\Pi_{m,p}$ is given by $\Pi_{m,p} = \cup_{m_1 \geq 0, m_2 \leq m, p_1 \leq p} \mathcal{R}_{m_1,p_1,m_2}$, and we assume that problem instance P is drawn i.i.d from the problem distribution $\mathcal{D}$ over $\Pi_{m,p}$. Remarkably, in this setting, problem instances can have varying training and validation sample sizes, as well as different sets of features. This general framework applies to practical scenarios where the feature sets differ among instances and allows one to learn regularization parameters that effectively work on average across multiple different but related problem instances. See Figure 1 for an illustration of the setting.

The goal here is to learn the value $\hat{\lambda}$ s.t. with high probability over the draw of n problem instances, the expected validation loss $\mathbb{E}_{P \sim \mathcal{D}} h(\hat{\lambda}, P)$ is close to $\min_\lambda \mathbb{E}_{P \sim \mathcal{D}}[h(\lambda, P)]$. This paper primarily focuses on providing learning guarantees in terms of sample complexity for the problem of tuning regularization parameters in the Elastic Net and regularized logistic regression (RLR). Specifically, we aim to address the question of how many problem instances are required to learn a value of λ that performs well across all problems P drawn from the problem distribution $\mathcal{D}$. To achieve this, we analyze the pseudo-dimension (in the case of the Elastic Net) or the Rademacher Complexity (for RLR) of the validation loss function class $\mathcal{H} = \{h(\lambda, \cdot) \mid \lambda \in \Lambda\}$, where Λ represents the search space for λ.

3 Tight pseudo-dimension bounds for Elastic Net hyperparameter tuning

In this section, we will present our results on the pseudo-dimension upper and lower bounds for the regularized linear regression problem in the data-driven setting. Classic learning-theoretic results [33, 34] connect the pseudo-dimension of the validation loss function class (parameterized by the regularization coefficient) with the *sample complexity* of the number of problem instances $\{P^{(1)}, \ldots, P^{(n)}\}$ drawn i.i.d. from some unknown problem distribution $\mathcal{D}$ needed for learning good regularization parameters with high confidence. Let $h_{\text{EN}}(\lambda, P) = l_{\text{EN}}(\hat{\beta}_{(X,y)}(\lambda), (X_{\text{val}}, y_{\text{val}}))$ be the validation loss function of the Elastic Net, and $\mathcal{H}_{\text{EN}} = \{h_{\text{EN}}(\lambda, P) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}^2_{>0}\}$ be the corresponding validation loss function class, we now present tight bounds for $\text{Pdim}(\mathcal{H}_{\text{EN}})$.

3.1 The Goldberg-Jerrum framework

Recently, Bartlett et al. [31] instantiate a simplified version of the well-known Goldberg-Jerrum (GJ) Framework [17]. The GJ framework offers a general pseudo-dimension upperbound for a wide class of functions in which each function can be computed by a *GJ algorithm*. We provide a brief overview of the GJ Framework which is useful in establishing our improved pseudo-dimension upper bound.

Definition 1 (GJ Algorithm, [31])**.** *A **GJ algorithm** Γ operates on real-valued inputs, and can perform two types of operations:*

- *Arithmetic operators of the form $v'' = v \odot v'$, where $\odot \in \{+, -, \times, \div\}$, and*
- *Conditional statements of the form "if $v \geq 0 \ldots$ else $\ldots$".*

In both cases, v and v' are either inputs or values previously computed by the algorithm.

General speaking, each intermediate value of the GJ algorithm Γ can be described by a *rational function*, which is a fractional between two polynomials, of the algorithm's inputs. The degree of a rational function is equal to the maximum degree of the polynomials in its numerator and its denominator. We can define two quantities that represent the complexity of GJ algorithms.

Definition 2 (Complexity of GJ algorithm, [31])**.** *The **degree** of a GJ algorithm is the maximum degree of any rational function it computes of the inputs. The **predicate complexity** of a GJ algorithm is the number of distinct rational functions that appear in its conditional statements.*

The following theorem essentially shows that for any function class $\mathcal{F}$, if we can describe any function $f \in \mathcal{F}$ by a GJ algorithm of which the degree and predicate complexity are at most Δ and Λ, respectively, then we can automatically obtain the upper bound for the pseudo-dimension of $\mathcal{F}$.

Theorem 3.1 ([31]). *Suppose that each function $f \in \mathcal{F}$ is specified by n real parameters. Suppose that for every $x \in \mathcal{X}$ and $r \in \mathbb{R}$, there is a GJ algorithm $\Gamma_{x,r}$ that given $f \in \mathcal{F}$, returns "true" if $f(x) \geq r$ and "false" otherwise. Assume that $\Gamma_{x,r}$ has degree Δ and predicate complexity Λ. Then,* $\text{Pdim}(\mathcal{F}) = O(n \log(\Delta\Lambda))$.

3.2 Upper bound

Our work improves on prior research [14] by presenting an upper bound on the pseudo-dimension of Elastic Net validation loss function class $\mathcal{H}_{\text{EN}}$ parameterized by λ. We extend the previous piecewise-decomposable structure of the loss function by providing a bound on the number of distinct rational piece functions for any fixed problem instance (Definition 3). This allows us to use a GJ algorithm and Theorem 3.1 to obtain better bounds on the number of distinct predicates that need to be computed. While prior research only used a bound on the number of distinct loss function pieces generated by algebraic boundaries, our new observation that the loss function has a limited number of possible distinct functional behaviors yields a tighter upper bound on the pseudo-dimension (Theorem 3.2). In Theorem 3.5, we will demonstrate the tightness of our upper bound by providing a novel lower bound for the problem.

We first provide a refinement of the piece-wise decomposable function class terminology introduced by [32] which is useful for establishing our improved upper bound. Intuitively, this corresponds to real-valued functions for which the domain is partitioned by finitely many *boundary functions* such that the function is well-behaved in each piece in the partition, i.e. can be computed using a *piece function* from another function class.

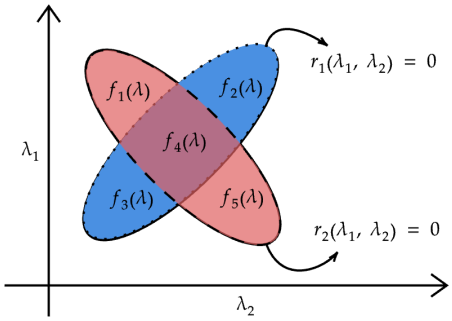

Figure 2: An illustration of piece-wise structure of $\mathcal{H}^*_{EN} = \{h^*_P : \mathcal{H}_{EN} \to \mathbb{R}_{\geq 0} \mid P \in \Pi_{m,p}\}$. Given a problem instance P, the function $h^*_P(\lambda) = h(P; \lambda)$ is a fixed rational function $f_i(\lambda)$ in each piece (piece function), that is regulated by boundary functions g_{r_i} of the form $\mathbb{1}\{r_i(\lambda) < 0\}$. As mentioned in our main result, there are at most 3^p functions f_i of degree at most $2p$, and at most $p3^p$ functions g_{r_i} where r_i is a polynomial of degree at most p.

Definition 3. *A function class $\mathcal{H} \subseteq \mathbb{R}^{\mathcal{Y}}$ that maps a domain $\mathcal{Y}$ to $\mathbb{R}$ is $(\mathcal{F}, k_{\mathcal{F}}, \mathcal{G}, k_{\mathcal{G}})-$piece-wise decomposable for a class $\mathcal{G}$ of boundary functions and a class $\mathcal{F} \in \mathbb{R}^{\mathcal{Y}}$ of piece functions if the following holds: for every $h \in \mathcal{H}$, (1) there are $k_{\mathcal{G}}$ functions $g^{(1)}, \ldots, g^{(k_{\mathcal{G}})} \in \mathcal{G}$ and a function $f_{\boldsymbol{b}} \in \mathcal{F}$ for each bit vector $\boldsymbol{b} \in \{0,1\}^{k_{\mathcal{G}}}$ s.t. for all $y \in \mathcal{Y}$, $h(y) = h_{\boldsymbol{b}_y}(y)$ where $\boldsymbol{b}_y = \{(g^{(1)}(y), \ldots, g^{(k_{\mathcal{G}})}(y))\} \in \{0,1\}^{k_{\mathcal{G}}}$, and (2) there is at most $k_{\mathcal{F}}$ different functions in $\mathcal{F}$.*

A key distinction from [32] is the finite bound $k_{\mathcal{F}}$ on the number of different piece functions needed to define any function in the class $\mathcal{H}$. Under this definition we give the following more refined structure for the Elastic Net loss function class by extending arguments from [14].

Theorem 3.2. *Let $\mathcal{H}_{EN} = \{h_{EN}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}^2_{\geq 0}\}$ be the class of Elastic Net validation loss function class. Consider the dual class $\mathcal{H}^*_{EN} = \{h^*_P : \mathcal{H}_{EN} \to \mathbb{R}_{\geq 0} \mid P \in \Pi_{m,p}\}$, where $h^*_P(h_{EN}(\lambda, \cdot)) = h_{EN}(\lambda, P)$. Then $\mathcal{H}^*_{EN}$ is $(\mathcal{F}, 3^p, \mathcal{G}, p3^p)$-piecewise decomposable, where the piece function class $\mathcal{F} = \{f_q : \mathcal{H}_{EN} \to \mathbb{R}\}$ consists at most 3^p rational functions $f_{q_1,q_2} : h_{EN}(\lambda, \cdot) \mapsto \frac{q_1(\lambda_1,\lambda_2)}{q_2(\lambda_1,\lambda_2)}$ of degree at most $2p$, and the boundary function class $\mathcal{G} = \{g_r : \mathcal{H}_{EN} \to \{0,1\}\}$ consists*

of semi-algebraic sets bounded by at most $p3^p$ algebraic curves $g_r : h_{EN}(\lambda, \cdot) \mapsto \mathbb{1}\{r(\lambda_1, \lambda_2) < 0\}$, where r is a polynomial of degree at most p.

Figure 2 demonstrates the piece-wise structure of $\mathcal{H}^*_{\text{EN}}$, which allows us to establish an improved upper bound on the pseudo-dimension.

Theorem 3.3. *Let $\mathcal{H}_{EN} = \{h_{EN}(\lambda, \cdot) : \Pi \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}^2_{>0}\}$ be the Elastic Net validation loss function class that maps problem instance P to validation loss $h_{val}(\lambda, P)$. Then $\text{Pdim}(\mathcal{H}_{EN})$ is $O(p)$.*

Proof Sketch. For every problem instance $P \in \Pi_{m,p}$, and a threshold $r \in \mathbb{R}$, consider the computation $\mathbb{1}\{h_{\text{EN}}(\lambda, P) - r \geq 0\}$ for any $h_{\text{EN}}(\lambda, \cdot) \in \mathcal{H}_{\text{EN}}$. From Theorem 3.2, we can describe $\mathbb{1}\{h_{\text{EN}}(\lambda, P) - r \geq 0\}$ as a GJ algorithm $\Gamma_{P,r}$ which is specified by 2 parameters λ_1, λ_2, has degree of at most $2p$, and has predicate complexity of at most $(p+1)3^p$ (See Figure 3). Then Theorem 3.1 implies that $\text{Pdim}(\mathcal{H}_{\text{EN}}) = O(p)$. □

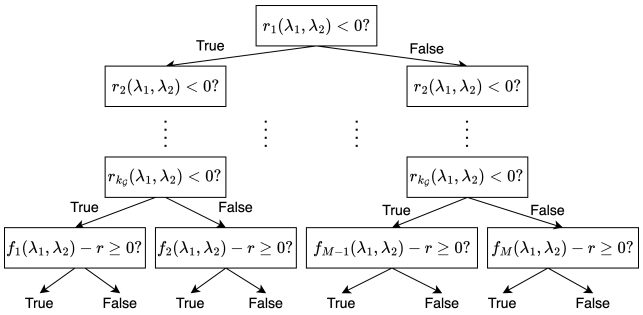

Figure 3: An illustration of how $\mathbb{1}\{h_{\text{EN}}(\lambda, P) - r \geq 0\}$ is computed as a GJ algorithm. The number of boundary (polynomial) functions $k_\mathcal{G}$ is at most $p3^p$, and there are at most $M = 3^p$ distinct (rational) piece functions. All the polynomial and rational functions are of degree at most $2p$.

The detailed proof of Theorem 3.3 can be found on Appendix B.1.2. Recent work by Balcan et al. [14] also studied the Elastic Net, and showed the piece-wise structure of the dual function of the validation loss function which implies an upper bound of $O(p^2)$ by employing the general tool from [32]. We establish a tighter bound of $O(p)$ in Theorem 3.3 by establishing additional properties of the loss function class and giving a GJ algorithm for computing the loss functions.

To guarantee the boundedness of the considered validation loss function classes, we will have the following assumptions for the data and regularization parameters. The first assumption is that all features and target values in the training and validation examples are bounded. The second assumption is that we only consider regularization coefficient values λ within an interval $[\lambda_{\min}, \lambda_{\max}]$. In practice, those assumptions are naturally satisfied by data normalization.

Assumption 1 (Bounded covariate and label)**.** *We assume that all the feature vectors and target values in training and validation set is upper-bounded by absolute constants R_1 and R_2, i.e. $\max\{\|X\|_\infty, \|X_{val}\|_\infty\} \leq R_1$, and $\max\{\|y\|_\infty, \|y_{val}\|_\infty\} \leq R_2$.*

Assumption 2 (Bounded Coefficient)**.** *We assume that $\lambda \in [\lambda_{\min}, \lambda_{\max}]^2$ with $\lambda_{\min} > 0$.*

Under Assumptions 2, 1, Theorem 3.3 immediately implies the following generalization guarantee for Elastic Net hyperparameter tuning.

Theorem 3.4. *Let $\mathcal{D}$ be an arbitrary distribution over the problem instance space $\Pi_{m,p}$. Under Assumptions 1, 2, the loss functions in $\mathcal{H}_{EN}$ have range bounded by some constant H (Lemma C.1). Then there exists an algorithm s.t. for any $\epsilon, \delta > 0$, given $N = O(\frac{H^2}{\epsilon^2}(p + \log(\frac{1}{\delta})))$ sample problem instances drawn from $\mathcal{D}$, the algorithm outputs a regularization parameter $\hat{\lambda}$ such that with probability at least $1 - \delta$, $\mathbb{E}_{P\sim\mathcal{D}} h_{EN}(\hat{\lambda}, P) < \min_\lambda \mathbb{E}_{P\sim\mathcal{D}} h_{EN}(\lambda, P) + \epsilon$.*

Proof. Denote $\lambda^* = \text{argmin}_\lambda \mathbb{E}_{P\sim\mathcal{D}} h_{\text{EN}}(\lambda, P)$. From Theorems 3.3 and A.2, given $n = O(\frac{H^2}{\epsilon^2}(p + \log(\frac{1}{\delta})))$ problem instances $P^{(i)}$ for $i \in [N]$ drawn from $\mathcal{D}$, w.p. $1 - \delta$, we have $\mathbb{E}_{P\sim\mathcal{D}} h_{\text{EN}}(\hat{\lambda}, P) < \frac{1}{N}\sum_{i=1}^N h_{\text{EN}}(\hat{\lambda}, P^{(i)}) + \frac{\epsilon}{2} < \frac{1}{N}\sum_{i=1}^N h_{\text{EN}}(\lambda^*, P^{(i)}) + \frac{\epsilon}{2} < \mathbb{E}_{P\sim\mathcal{D}} h_{\text{EN}}(\lambda^*, P) + \epsilon$. □

3.3 Lower bound

Remarkably, we are able to establish a matching lower bound on the pseudo-dimension of the Elastic Net loss function class, parameterized by the regularization parameters. Note that every Elastic Net problem can be converted to an equivalent LASSO problem [1]. In fact, we show something stronger, that the pseudo-dimension of even the LASSO regression loss function class (parameterized by regression coefficient λ_1) is $\Omega(p)$, from which the above observation follows (by taking $\lambda_2 = 0$ in our construction). Our proof of the lower bound adapts the "adversarial strategy" of [35] which is used to design a worst-case LASSO regularization path. While [35] construct a single dataset to bound the number of segments in the piecewise-linear LASSO solution path, we create a collection of problem instances for which all above-below sign patterns may be achieved by selecting regularization parameters from different segments of the solution path.

Theorem 3.5. *Let $\mathcal{H}_{LASSO}$ be a set of functions $\{h_{LASSO}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}^+\}$ that map a regression problem instance $P \in \Pi_{m,p}$ to the validation loss $h_{LASSO}(\lambda, P)$ of LASSO trained with regularization parameter λ. Then* $\text{Pdim}(\mathcal{H}_{LASSO})$ *is $\Omega(p)$.*

Proof Sketch. Consider $N = p$ problem instances for LASSO regression given by $P^{(i)} = (X^{(i)}, y^{(i)}, X^{(i)}_{\text{val}}, y^{(i)}_{\text{val}})$, where the training set $(X^{(i)}, y^{(i)}) = (X^*, y^*)$ is fixed and set using the "adversarial strategy" of [35], Proposition 2. The validation sets are given by single examples $(X^{(i)}_{\text{val}}, y^{(i)}_{\text{val}}) = (\mathbf{e}_i, 0)$, where $\mathbf{e}_i$ are standard basis vectors in $\mathbb{R}^p$. We will now proceed to provide the witnesses $r_1, \ldots, r_N$ and λ values to exhibit a pseudo-shattering of these problem instances.

Corresponding to subset $T \subseteq [p]$ of problem instances, we will provide a value of λ_T such that, we have $\ell_{\text{LASSO}}(\lambda_T, P^{(i)}) > r_i$ iff $i \in T$, for each $i \in [p]$ and each $T \subseteq [p]$. We set all witnesses $r_i = 0$ for all $i \in [p]$. As a consequence of Theorem 1 in [35], the regularization path of (X^*, y^*) consists of a linear segment corresponding all 2^p unsigned sparsity patterns in $\{0, 1\}^p$ (we will not need all the segments in the construction, but note that it is guaranteed to contain all distinct unsigned sparsity patterns) and we select λ_T as any interior point corresponding to a linear segment with sparsity pattern $\{(c_1, \ldots, c_p) \mid c_i = 0 \text{ iff } i \in T\}$, i.e. elements in T are exactly the ones with sparsity pattern 0. Therefore, $|\beta^*_T \cdot \mathbf{e}_i| = 0$ iff $i \in T$, where β^*_T is the LASSO regression fit for regularization parameter λ_T. This implies the desired shattering condition w.r.t. witnesses $r_1 = 0, \ldots, r_N = 0$. Therefore, $\text{Pdim}(\mathcal{H}_{\text{LASSO}}) \geq p$. See Appendix B.2 for a full proof. □

3.4 Hyperparameter tuning in Regularized Kernel Regression

The Kernel Least Squares Regression ([4]) is a natural generalization of the linear regression problem, which uses a kernel to handle non-linearity. In this problem, each sample has p_1 feature, corresponding to a real-valued target. Formally, each problem instance P drawn from Π can be described as

$$P = (X, y, X_{\text{val}}, y_{\text{val}}) \in \mathbb{R}^{m \times p_1} \times \mathbb{R}^m \times \mathbb{R}^{m' \times p_1} \times \mathbb{R}^{m'}.$$

A common issue in practice is that the relation between y and X is non-linear in the original space. To overcome this issue, we consider the mapping $\phi : \mathbb{R}^{p_1} \to \mathbb{R}^{p_2}$ which maps the original input space to a new feature space in which we hopefully can perform linear regression. Define $\phi(X) = (\phi(x_1), \ldots, \phi(x_m))_{m \times p_2}$, our goal is to find a vector $\theta \in \mathbb{R}^{p_2}$ so that the squared loss $\frac{1}{2} \|y - \phi(X)\theta\|_2^2 + R(\|\theta\|)$ is minimized, where the regularization term $R(\|\theta\|)$ is any strictly monotonically increasing function of the Hilbert space norm. It is well-known from the literature (e.g. [36]) that under the Representer Theorem's conditions, the optimal value θ^* can be linearly represented by row vectors of $\phi(X)$, i.e., $\theta^* = \phi(X)\beta = \sum_{i=1}^m \phi(x_i)\beta_i$, where $\beta = (\beta_1, \ldots, \beta_m) \in \mathbb{R}^m$. This directly includes the ℓ_2 regularizer but does not include ℓ_1 regularization. To overcome this issue, Roth ([3]) proposed an alternative approach to regularized kernel regression, which directly restricts the representation of coefficient θ via a linear combination of $\phi(x_i)$, for $i \in [m]$. The regularized kernel regression hence can be formulated as

$$\hat{\beta}^{(X,y)}_{l,\lambda} = \underset{\beta \in \mathbb{R}^m}{\text{argmin}} \frac{1}{2} \|y - K\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2,$$

where $k(x, x') = \langle \phi(x), \phi(x') \rangle$ is the kernel mapping, and the Gram matrix K satisfies $[K]_{i,j} = k(x_i, x_j)$ for all $i, j \in [m]$.

Clearly, the problem above is a linear regression problem. Formally, denote $h_{\text{KER}}(\lambda, P) = \frac{1}{2}\|y - K\hat{\beta}_{(X,y)}(\lambda)\|_2$ and let $\mathcal{H}_{\text{KER}} = \{h_{\text{KER}}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}_+^2\}$. The following result is a direct corollary of Theorem 3.3, which gives an upper bound for the pseudo-dimension of $\mathcal{H}_{\text{KER}}$.

Corollary 3.6. $\text{Pdim}(\mathcal{H}_{KER}) = O(m)$.

Note that m here denotes the training set size for a single problem instance, and Corollary 3.6 implies a guarantee on the number of problem instances needed for learning a good regularization parameter for kernel regression via classic results [33, 34]. Our results do not make any assumptions on the m samples within a problem instance/dataset; if these samples within problem instances are further assumed to be i.i.d. draws from some data distribution (distinct from problem distribution $\mathcal{D}$), then well-known results imply that $m = O(k \log p)$ samples are sufficient to learn the optimal LASSO coefficient [37, 38], where k denotes the number of non-zero coefficients in the optimal regression fit.

4 Hyperparameter tuning for Regularized Logistic Regression

Logistic regression is more naturally suited to applications modeling probability of an event, like medical risk for a patient [39], predicting behavior in markets [40], failure probability of an engineering system [41] and many more applications [42]. It is a fundamental statistical technique for classification, and regularization is again crucial for avoiding overfitting and estimating variable importance. In this section, we will present learning guarantees for tuning the Regularized Logistic Regression (RLR) regularization coefficients across instances. Given a problem instance P drawn from a problem distribution $\mathcal{D}$ over $\Pi_{m,p}$, let $h_{\text{RLR}}(\lambda, P) = l_{\text{RLR}}(\hat{\beta}_{(X,y)}(\lambda), (X_{\text{val}}, y_{\text{val}}))$ be the RLR validation loss function class (defined in Section 2), and let $\mathcal{H}_{\text{RLR}} = \{h_{\text{RLR}}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}_{>0}\}$ be the RLR validation loss function class, our goal is to provide a learning guarantee for $\mathcal{H}_{\text{RLR}}$. Besides, we also study the commonly used 0-1 validation loss function class $\mathcal{H}_{\text{RLR}}^{\text{0-1}} = \{h_{\text{RLR}}^{\text{0-1}}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}_{>0}\}$, where $h_{\text{RLR}}^{\text{0-1}}(\lambda, P) = \frac{1}{m'}\sum_{i=1}^{m'} \mathbb{1}\{y_i x_i^\top \hat{\beta}_{X,y}(\lambda) \leq 0\}$, which we will cover in Section 4.3. Similarly, to guarantee the boundedness of $\mathcal{H}_{\text{RLR}}$, we also assume that Assumptions 1 and 2 also hold in this setting.

4.1 Approximate solutions of Regularized Logistic Regression

The main challenge in analyzing the regularized logistic regression, unlike the regularized logistic regression problem, is that the solution $\hat{\beta}_{(X,y)}(\lambda)$ corresponding to a problem instance P and particular value $\lambda > 0$ does not have a closed form depending on λ. We then propose an alternative approach to this end, which is examining via the approximation $\beta_{(X,y)}^{(\epsilon)}(\lambda)$ of the solution $\hat{\beta}_{(X,y)}(\lambda)$.

Algorithm 1 Approximate incremental quadratic algorithm for RLR with ℓ_1 penalty, [2]

Set $\beta_0^{(\epsilon)} = \hat{\beta}_{(X,y)}(\lambda_{\min})$, $t = 0$, small constant $\delta \in \mathbb{R}_{>0}$, and $\mathcal{A} = \{j \mid [\hat{\beta}_{(X,y)}(\lambda_{\min})]_j \neq 0\}$.
while $\lambda_t < \lambda_{\max}$ **do**
- $\lambda_{t+1} = \lambda_t + \epsilon$
- $\left(\beta_{t+1}^{(\epsilon)}\right)_{\mathcal{A}} = \left(\beta_t^{(\epsilon)}\right)_{\mathcal{A}} - \left[\nabla^2 l\left(\beta_t^{(\epsilon)}, (X,y)\right)_{\mathcal{A}}\right]^{-1} \cdot \left[\nabla l\left(\beta_t^{(\epsilon)}, (X,y)\right)_{\mathcal{A}} + \lambda_{t+1}\,\text{sgn}\left(\beta_t^{(\epsilon)}\right)_{\mathcal{A}}\right]$
- $\left(\beta_{t+1}^{(\epsilon)}\right)_{-\mathcal{A}} = \vec{0}$
- $\mathcal{A} = \mathcal{A} \cup \{j \neq \mathcal{A} \mid \nabla l(\beta_{t+1}^{(\epsilon)}, (X,y)) > \lambda_{t+1}\}$
- $\mathcal{A} = \mathcal{A} \setminus \{j \in \mathcal{A} \mid \left|\beta_{t+1,j}^{(\epsilon)}\right| < \delta\}$
- $t = t + 1$

The approximation Algorithm 1 (Algorithm 2) for the solution $\hat{\beta}(\lambda)$ of RLR under ℓ_1 (or ℓ_2) constraint were first proposed by Rosset [26, 2]. Given a problem instance P, and a sufficiently small step-size $\epsilon > 0$, using Algorithms 1, 2 yields an approximation $\beta_{(X,y)}^{(\epsilon)}$ of $\hat{\beta}_{(X,y)}$ that are piece-wise linear functions of λ in total $(\lambda_{\max} - \lambda_{\min})/\epsilon$ [26]. Moreover, it is also guaranteed that the error between $\beta_{(X,y)}^{(\epsilon)}$ and $\hat{\beta}_{(X,y)}$ is uniformly upper bounded for all $\lambda \in [\lambda_{\min}, \lambda_{\max}]$.

Algorithm 2 Approximate incremental quadratic algorithm for RLR with ℓ_2 penalty, [2]

Set $\beta_0^{(\epsilon)} = \hat{\beta}_{(X,y)}(\lambda_{\min})$, $t = 0$.
while $\lambda_t < \lambda_{\max}$ **do**
  $\lambda_{t+1} = \lambda_t + \epsilon$
  $\beta^{(\epsilon)}(\lambda) = \beta_t^{(\epsilon)} - \left[\nabla^2 l\left(\beta_t^{(\epsilon)}, (X,y)\right) + 2\lambda_{t+1} I\right]^{-1} \cdot \left[\nabla l\left(\beta_t^{(\epsilon)}, (X,y)\right) + 2\lambda_{t+1}\beta_t^{(\epsilon)}\right]$
  $t = t + 1$

Theorem 4.1 (Theorem 1, [2])**.** *Given a problem instance P, for small enough ϵ, there is a uniform bound $O(\epsilon^2)$ on the error $\|\hat{\beta}_{(X,y)}(\lambda) - \beta^{(\epsilon)}_{(X,y)}(\lambda)\|_2$ for any $\lambda \in [\lambda_{\min}, \lambda_{\max}]$.*

Denote $h^{(\epsilon)}_{\text{RLR}}(\lambda, P) = l_{\text{RLR}}(\beta^{\epsilon}(\lambda), (X_{\text{val}}, y_{\text{val}}))$ the approximation function of the validation loss $h_{\text{RLR}}(\lambda, P)$. Using Theorem 4.1 and note that the loss $f(z) = \log(1 + e^{-z})$ is 1-Lipschitz, we can show that the difference between $h^{(\epsilon)}_{\text{RLR}}(\lambda, P)$ and $h_{\text{RLR}}(\lambda, P)$ is uniformly upper-bounded.

Lemma 4.2. *The approximation error of the validation loss function is uniformly upper-bounded $|h^{(\epsilon)}_{RLR}(\lambda, P) - h_{RLR}(\lambda, P)| = O(\epsilon^2)$, for all $\lambda \in [\lambda_{\min}, \lambda_{\max}]$.*

We now present one of our main results, which is the pseudo-dimension bound of the approximate validation loss function class $\mathcal{H}^{(\epsilon)}_{\text{RLR}}$.

Theorem 4.3. *Consider the RLR under ℓ_1 (or ℓ_2) constraint with parameter $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ that take a problem instance P drawn from an unknown problem distribution $\mathcal{D}$ over $\Pi_{m,p}$. Under Assumptions 1 and 2, $\mathcal{H}_{RLR}$ is bounded by some constant H (Lemma C.2). Suppose that we use Algorithm 1 (or Algorithm 2) to approximate the solution $\hat{\beta}_{(X,y)}(\lambda)$ by $\beta^{(\epsilon)}_{(X,y)}(\lambda)$ with a uniform error $O(\epsilon^2)$ for any $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, where ϵ is the approximation step-size. Consider the approximation validation loss function class $\mathcal{H}^{(\epsilon)}_{RLR} = \{h^{(\epsilon)}_{RLR}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$, where*

$$h^{(\epsilon)}_{RLR}(\lambda, P) = \frac{1}{m'} \sum_{i=1}^{m'} \log(1 + \exp(-y_i x_i^\top \beta^{(\epsilon)}_{(X,y)}(\lambda)))$$

is the approximate validation loss. Then we have $\text{Pdim}(\mathcal{H}^{(\epsilon)}_{RLR}) = O(m^2 + \log(1/\epsilon))$. Further, we assume that $\epsilon = O(\sqrt{H})$ where H is the upperbound of $\mathcal{H}_{RLR}$ under Assumptions 1 and 2. Given any set $\mathcal{S}$ of T problem instances drawn from a problem distribution $\mathcal{D}$ over $\Pi_{m,p}$, the empirical Rademacher complexity $\hat{\mathscr{R}}(\mathcal{H}^{(\epsilon)}_{RLR}, \mathcal{S}) = O(H\sqrt{(m^2 + \log(1/\epsilon))/T})$.

The key observation here is that the approximation solution $\hat{\beta}^{(\epsilon)}_{(X,y)}$ is piece-wise linear over $(\lambda_{\max} - \lambda_{\min})/\epsilon$ pieces, leading to the fact that the approximate validation loss function $h^{(\epsilon)}_{\text{RLR}}(\lambda, \cdot)$ is a "special function" (Pfaffian function [43]) in each piece, which is a combination of exponentiation of linear functions of λ. The detailed proof of Theorem 4.3 can be found on the Appendix D.3.

4.2 Learning guarantees for Regularized Logistic Regression hyperparameter tuning

Our goal now is to use the upper bound for empirical Rademacher complexity of the validation loss function class $\mathcal{H}_{\text{RLR}}$. We use techniques for approximate data-driven algorithm design due to [29], combining the uniform error upper bound between validation loss function $h_{\text{RLR}}(\lambda, P)$ and its approximation $h^{(\epsilon)}_{\text{RLR}}(\lambda, P)$ (Lemma 4.2) and empirical Rademacher complexity of approximation validation loss function class $\mathcal{H}^{(\epsilon)}_{\text{RLR}}$ (Theorem 4.3), to obtain a bound on the empirical Rademacher complexity of $\mathcal{H}_{\text{RLR}}$. This allows us to give a learning guarantee for the regularization parameters λ, which is formalized by the following theorem.

Theorem 4.4. *Consider the RLR under ℓ_1 (or ℓ_2) constraint. Under Assumptions 1, 2, $\mathcal{H}_{RLR}$ is bounded by some constant H (Lemma C.2). Consider the class function $\mathcal{H}_{RLR} = \{h_{RLR}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$ where $h_{RLR}(\lambda, P)$ is the validation loss corresponding to problem*

instance P and the ℓ_1 (ℓ_2) parameter λ. Given any set $\mathcal{S}$ of T problem instances drawn from a problem distribution $\mathcal{D}$ over $\Pi_{m,p}$, for any $h_{RLR}(\lambda, \cdot) \in \mathcal{H}_{RLR}$, w.p. $1-\delta$ for any $\delta \in (0,1)$, we have

$$\left| \frac{1}{T} \sum_{i=1}^{T} h_{RLR}(\lambda, P^{(i)}) - \mathbb{E}_{P \sim \mathcal{D}}[h_{RLR}(\lambda, P)] \right| \leq O\left(H \sqrt{\frac{m^2 + \log(1/\epsilon)}{T}} + \epsilon^2 + \sqrt{\frac{1}{T} \log \frac{1}{\delta}} \right),$$

for some sufficiently small ϵ.

The proof detail of Theorem 4.4 is included in the Appendix D.3. The above generalization guarantee gives a bound on the average error on RLR validation loss over the problem distribution, for the parameter λ learned from T problem instances. In commonly used approaches, the validation set size is small or a constant, and our result can be interpreted as the upper bound on the generalization error in terms of the number of problem instances T and the step length ϵ. We only consider RLR under ℓ_1 (or ℓ_2) constraints, which are commonly studied in the literature, our analysis could be easily extended to RLR under ℓ_q constraint for any $q \geq 1$.

4.3 An extension to 0-1 loss

Since logistic regression is often used for binary classification tasks, it is interesting to consider the 0-1 loss as the validation loss function. It has been shown that $\mathbb{1}\{z \leq 0\} \leq 4\log(1+e^{-z})$ for any z [44]. This inequality, combined with Theorem 4.4, directly provides a learning guarantee for the 0-1 validation loss function.

Theorem 4.5. *Let $\tau > 2\epsilon^2$ and $\delta \in (0,1)$, where ϵ is the approximation step-size. Then for any $n \geq s(\tau/2, \delta) = \Omega\left(\frac{H^2(m^2 + \log\frac{1}{\epsilon}) + \log\frac{1}{\delta}}{(\tau/2 - \epsilon^2)^2}\right)$, if we have n problem instances $\{P^{(i)}, \ldots, P^{(n)}\}$ drawn i.i.d. from some problem distribution $\mathcal{D}$ over $\Pi_{m,p}$ to learn the regularization parameter λ^{ERM} for RLR via ERM, then*

$$\mathbb{E}_{P \sim \mathcal{D}}(h^{0\text{-}1}_{RLR}(\lambda^{ERM}, P))) \leq 4 \min_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} \mathbb{E}_{P \sim \mathcal{D}}(h_{RLR}(\lambda, P)) + 4\tau.$$

The detailed proof of Theorem 4.5 can be found on Appendix D.4. It is worth noting that we are providing learning guarantee for 0-1 validation loss function class $\mathcal{H}^{0\text{-}1}_{\text{RLR}}$ indirectly via the validation loss function class $\mathcal{H}_{\text{RLR}}$ with cross-entropy objective function, which is arguably not optimal. The question of how to provide a true PAC-learnable guarantee for $\mathcal{H}^{0\text{-}1}_{\text{RLR}}$ remains an interesting challenge.

5 Conclusion and future work

In this work, we present novel learning guarantees for tuning regularization parameters for both the Elastic Net and Regularized Logistic Regression models, across problem instances. For the Elastic Net, we propose fine-grained structural results that pertain to the tuning of regularization parameters. We use them to give an improved upper bound on the pseudo-dimension of the relevant validation loss function class of and we prove that our new bound is tight.

For the problem of tuning regularization parameters in regularized logistic regression, we propose an alternative approach that involves analyzing the approximation of the original validation loss function class. This approximation, characterized by a piece-wise linear representation, provides a useful analytical tool in the absence of an exact dependence of the logistic loss on the regularization parameters. Additionally, we employ an upper bound on the approximation error between the original and approximated functions, to obtain a learning guarantee for the original validation loss function class. Remarkably, our proposed approach is not restricted solely to regularized logistic regression but can be extended to a wide range of other problems, demonstrating its generality and applicability.

It is worth noting that this work only focuses on the sample complexity aspect of the hyperparameter tuning in the Elastic Net and Regularized Logistic Regression. The question of computational complexity in this setting is an interesting future direction. Other interesting questions include designing hyperparameter tuning techniques for this setting that are robust to adversarial attacks, and hyperparameter tuning for Regularized Logistic Regression with both ℓ_1 and ℓ_2 constraints.

6 Acknowledgement

We thank Yingyu Liang for useful discussions during early stages of this work and Mikhail Khodak for helpful feedback. This work was supported in part by NSF grants CCF-1910321 and SES-1919453, the Defense Advanced Research Projects Agency under cooperative agreement HR00112020003, and a Simons Investigator Award.

References

[1] Hui Zou and Trevor Hastie. Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[2] Saharon Rosset. Following curved regularized optimization solution paths. *Advances in Neural Information Processing Systems*, 17, 2004.

[3] Volker Roth. The generalized LASSO. *IEEE transactions on neural networks*, 15(1):16–28, 2004.

[4] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[5] Arthur Hoerl and Robert Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[6] Robert Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[7] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[8] Edgar Dobriban and Stefan Wager. High-dimensional asymptotics of prediction: Ridge regression and classification. *The Annals of Statistics*, 46(1):247–279, 2018.

[9] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[10] Gang Wang, Dit-Yan Yeung, and Frederick H Lochovsky. The kernel path in kernelized LASSO. In *Artificial Intelligence and Statistics*, pages 580–587. PMLR, 2007.

[11] Yunlong Feng, Shao-Gao Lv, Hanyuan Hang, and Johan Suykens. Kernelized Elastic Net regularization: generalization bounds, and sparse recovery. *Neural computation*, 28(3):525–562, 2016.

[12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[13] Tong Zhang. Some sharp performance bounds for least squares regression with L1 regularization. *The Annals of Statistics*, pages 2109–2143, 2009.

[14] Maria-Florina Balcan, Mikhail Khodak, Dravyansh Sharma, and Ameet Talwalkar. Provably tuning the Elastic Net across instances. In *Advances in Neural Information Processing Systems*, 2022.

[15] Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.

[16] Mervyn Stone. Cross-validation: A review. *Statistics: A Journal of Theoretical and Applied Statistics*, 9(1):127–139, 1978.

[17] Paul Goldberg and Mark Jerrum. Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 361–369, 1993.

[18] Li Wang, Michael D Gordon, and Ji Zhu. Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 690–700. IEEE, 2006.

[19] Denny Wu and Ji Xu. On the optimal weighted L2 regularization in overparameterized linear regression. *Advances in Neural Information Processing Systems*, 33:10112–10123, 2020.

[20] Diane Galarneau Gibbons. A simulation study of some ridge estimators. *Journal of the American Statistical Association*, 76(373):131–139, 1981.

[21] Lisa-Ann Kirkland, Frans Kanfer, and Sollie Millard. LASSO tuning parameter selection. In *Annual Proceedings of the South African Statistical Association Conference*, volume 2015, pages 49–56. South African Statistical Association (SASA), 2015.

[22] Yosiyuki Sakamoto, Makio Ishiguro, and Genshiro Kitagawa. Akaike information criterion statistics. *Dordrecht, The Netherlands: D. Reidel*, 81(10.5555):26853, 1986.

[23] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[24] Michael Chichignoud, Johannes Lederer, and Martin Wainwright. A practical scheme and fast algorithm to tune the LASSO with optimality guarantees. *The Journal of Machine Learning Research*, 17(1):8162–8181, 2016.

[25] Denis Chetverikov, Zhipeng Liao, and Victor Chernozhukov. On cross-validated LASSO in high dimensions. *The Annals of Statistics*, 49(3):1300–1317, 2021.

[26] Saharon Rosset. *Topics in regularization and boosting*. PhD thesis, Stanford University, 2003.

[27] Ryan Tibshirani. The LASSO problem and uniqueness. *Electronic Journal of Statistics*, 7:1456–1490, 2013.

[28] Rishi Gupta and Tim Roughgarden. A PAC approach to application-specific algorithm selection. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 123–134, 2016.

[29] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. Refined bounds for algorithm configuration: The knife-edge of dual class approximability. In *International Conference on Machine Learning*, pages 580–590. PMLR, 2020.

[30] Maria-Florina Balcan, Siddharth Prasad, Tuomas Sandholm, and Ellen Vitercik. Structural analysis of branch-and-cut and the learnability of gomory mixed integer cuts. *Advances in Neural Information Processing Systems*, 35:33890–33903, 2022.

[31] Peter Bartlett, Piotr Indyk, and Tal Wagner. Generalization bounds for data-driven numerical linear algebra. In *Conference on Learning Theory*, pages 2013–2040. PMLR, 2022.

[32] Maria-Florina Balcan, Dan DeBlasio, Travis Dick, Carl Kingsford, Tuomas Sandholm, and Ellen Vitercik. How much data is sufficient to learn high-performing algorithms? Generalization guarantees for data-driven algorithm design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 919–932, 2021.

[33] Martin Anthony and Peter Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, USA, 1st edition, 2009.

[34] Maria-Florina Balcan. Book chapter Data-Driven Algorithm Design. In *Beyond Worst Case Analysis of Algorithms, T. Roughgarden (Ed)*. Cambridge University Press, 2020.

[35] Julien Mairal and Bin Yu. Complexity analysis of the LASSO regularization path. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1835–1842, 2012.

[36] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *Computational Learning Theory: 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001 Amsterdam, The Netherlands, July 16–19, 2001 Proceedings 14*, pages 416–426. Springer, 2001.

[37] Martin Wainwright. Information-theoretic bounds on sparsity recovery in the high-dimensional and noisy setting. In *2007 IEEE International Symposium on Information Theory*, pages 961–965. IEEE, 2007.

[38] Martin Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using L1-constrained quadratic programming (LASSO). *IEEE transactions on information theory*, 55(5):2183–2202, 2009.

[39] Peter Armitage, Geoffrey Berry, and John Nigel Scott Matthews. *Statistical methods in medical research*. John Wiley & Sons, 2008.

[40] Michael JA Berry and Gordon S Linoff. *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 2004.

[41] Sanjay Kumar Palei and Samir Kumar Das. Logistic regression model for prediction of roof fall risks in bord and pillar workings in coal mines: an approach. *Safety science*, 47(1):88–96, 2009.

[42] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[43] Askold G Khovanski. *Fewnomials*, volume 88. American Mathematical Soc., 1991.

[44] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in neural information processing systems*, 32, 2019.

[45] Richard M Dudley. Universal, Donsker classes and metric entropy. *The Annals of Probability*, 15(4):1306–1326, 1987.

A Classical Generalization Bounds

In this section, we will provide basic terminologies from classical learning theory which will be useful in our analysis.

A.1 Pseudo-dimension

The pseudo-dimension is frequently used to analyze the learning theoretic complexity of a real-valued function class. The formal definition is stated here for convenience.

Definition 4 (Shattering and Pseudo-dimension, [33]). *Let $\mathcal{F}$ be a set of functions mapping from $\mathcal{X}$ to $\mathbb{R}$, and suppose that $S = \{x_1, \ldots, x_m\} \subseteq \mathcal{X}$. Then S is pseudo-shattered by $\mathcal{F}$ if there are real numbers $r_1, \ldots, r_m$ such that for each $b \in \{0,1\}^m$ there is a function f_b in $\mathcal{F}$ with sign$(f_b(x_i) - r_i) = b_i$ for $i \in [m]$. We say that $r = (r_1, \ldots, r_m)$ witnesses the shattering. We say that $\mathcal{F}$ has pseudo-dimension d if d is the maximum cardinality of a subset S of $\mathcal{X}$ that is pseudo-shattered by $\mathcal{F}$, denoted* $\text{Pdim}(\mathcal{F}) = d$. *If no such maximum exists, we say that $\mathcal{F}$ has infinite pseudo-dimension.*

The following lemma is particularly useful when we analyze the pseudo-dimension of a function class that is a composition of a monotonic function and another simpler function class. The result is useful in our analysis of regularized logistic regression (Section D).

Lemma A.1 ([45]). *Suppose $\mathcal{F}$ is a class of real-valued functions and $\sigma : \mathbb{R} \to \mathbb{R}$ is a non-decreasing function. Let $\sigma(\mathcal{F})$ denote the class $\{\sigma \circ f : f \in \mathcal{F}\}$. Then* $\text{Pdim}(\sigma(\mathcal{F})) \leq \text{Pdim}(\mathcal{F})$. *The equality holds if σ is a continuous and strictly increasing function.*

On other hand, if σ is a non-increasing function then $\text{Pdim}(\sigma(\mathcal{F})) \geq \text{Pdim}(\mathcal{F})$. *The equality holds if σ is a continuous and strictly decreasing function.*

A.2 (Empirical) Rademacher Complexity

Another tool for analyzing the complexity of a real-valued function is the empirical Rademacher Complexity, which will be defined below.

Definition 5 (Empirical Rademacher Complexity, [9]). *Let $\mathcal{F}$ be a set of functions mapping from $\mathcal{X}$ to $\mathbb{R}$, and let $S = \{x_1, \ldots, x_T\} \subseteq \mathcal{X}$ be a set of T samples from $\mathcal{X}$. The empirical Rademacher Complexity of $\mathcal{F}$ with respect to S is defined as*

$$\hat{\mathscr{R}}(\mathcal{F}, S) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{T} \sigma_i f(x_i) \right],$$

where σ_i is Rademacher random variable for $i \in [T]$.

A.3 Uniform Convergence

The following classical result establishes a connection between the uniform convergence and the pseudo-dimension of real-valued function classes.

Theorem A.2 ([33]). *Suppose $\mathcal{F}$ is a class of real-valued functions with range in $[0, H]$ and finite* $\text{Pdim}(\mathcal{F})$. *Then for any $\epsilon > 0$ and $\delta \in (0,1)$, for any distribution $\mathcal{D}$ and for any set S of $m = O\left(\frac{H^2}{\epsilon^2}(\text{Pdim}(\mathcal{F}) + \log\frac{1}{\delta})\right)$ samples drawn from $\mathcal{D}$, w.p. at least $1 - \delta$, we have*

$$|L_S^m(f) - L_{\mathcal{D}}(f)| < \epsilon, \quad \text{for all } f \in \mathcal{F}.$$

B Lemmas, Proof Details for Section 3

In this section, we provide the details for results discussed in Section 3.

B.1 Upper bound

We will state results from prior research which are useful in establishing our pseudo-dimension upper bound, followed by full proof details.

B.1.1 Basic Structural Results About Elastic Net

We first present basic structural results about the Elastic Net. The result allows us to rewrite any Elastic Net problem into an equivalent LASSO problem.

Lemma B.1 (LASSO reduction of the Elastic Net, [1]). *Given $(X, y) \in \mathbb{R}^{m\times p} \times \mathbb{R}^m$ and $(\lambda_1, \lambda_2) \in (0, \infty) \times [0, \infty)$, define the ElasticNet problem*

$$\min_{\beta\in\mathbb{R}^p} \|y - \beta X\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2 .$$

Define the new dataset (X', y')

$$X'_{(n+p)\times p} = (1 + \lambda_2)^{-1/2} \begin{pmatrix} X \\ \sqrt{\lambda_2} I_p \end{pmatrix}, \quad y'_{n+p} = \begin{pmatrix} y \\ 0 \end{pmatrix}.$$

Let $\gamma = \frac{\lambda_1}{\sqrt{1+\lambda_2}}$. Then the original ElasticNet problem can be written as

$$\min_{\beta'\in\mathbb{R}^p} \|y' - X'\beta'\|_2^2 + \gamma \|\beta'\|_1 .$$

Let $\hat\beta = \operatorname{argmin}_{\beta\in\mathbb{R}^p} \|y - \beta X\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2$ and $\hat\beta' = \operatorname{argmin}_{\beta'\in\mathbb{R}^p} \|y' - X'\beta'\|_2^2 + \gamma \|\beta'\|_1$, then

$$\hat\beta = \frac{1}{\sqrt{1+\lambda_2}} \hat\beta'.$$

The following result (Lemma B.2) characterizes the solutions of the LASSO problem. To state it, we will need a couple definitions. The *general position* is a standard mild assumption on the design matrix X.

Definition 6 (General position, [27]). *A matrix $X \in \mathbb{R}^{m\times p}$ is said to have its columns in the general position if the affine span of any $k \le m$ points $(\sigma_i x_{j_i})_{i\in[k], j_i=J\subseteq[p]}$ for arbitrary signs $\sigma_{[k]} \in \{-1, 1\}^k$ and subset J of the columns of size k, does not contain any element of $\{x_i | i \notin J\}$.*

Equicorrelation set (sometimes called active set) is the set of covariates with maximum absolute value of correlation for the LASSO fit corresponding to a given value of λ_1.

Definition 7 (Equicorrelation sets, [27]). *Let $\hat\beta \in \operatorname{argmin}_{\beta\in\mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1$. The equicorrelation set corresponding to $\hat\beta$, $\mathcal{E} = \left\{ j \in [p] \mid \left| \boldsymbol{x}_j^\top \left(y - X\hat\beta \right) \right| = \lambda_1 \right\}$ is simply the set of covariates with maximum absolute correlation. We also define the equicorrelation sign vector for $\hat\beta$ as $s = sign(X_\mathcal{E}^\top (y - X\hat\beta)) \in \{\pm 1\}^{|\mathcal{E}|}$.*

We are now ready to state the unique closed-form solution of the LASSO under general position assumption, in terms of equicorrelation sets.

Lemma B.2 (Closed-form solution of the LASSO, [27]). *If the columns of X are in general position, then for any y and $\lambda_1 > 0$, the LASSO solution is unique and is given by*

$$\hat\beta_\mathcal{E} = (X_\mathcal{E}^\top X_\mathcal{E})^{-1}(X_\mathcal{E}^\top y - \lambda_1 s), \quad \hat\beta_{[p]\setminus\mathcal{E}} = 0$$

where $\mathcal{E}$ and s are the equicorrelation set and equicorrelation sign vector corresponding to $\hat\beta$.

Therefore, the solution of Elastic Net can be written as below.

Lemma B.3 (Closed-form solution of the ElasticNet, [1]). *Let X be the matrix with columns in the general position, and $\lambda = (\lambda_1, \lambda_2) \in \mathbb{R}^2_{>0}$. Then the ElasticNet solution $\hat\beta(\lambda) \in \operatorname{argmin}_{\beta\in\mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2$ is unique for any dataset (X, y) and satisfies*

$$(\hat\beta(\lambda))_\mathcal{E} = \left(X_\mathcal{E}^\top X_\mathcal{E} + \lambda_2 I_{|\mathcal{E}|}\right)^{-1} X_\mathcal{E}^\top y - \lambda_1 \left(X_\mathcal{E}^\top X_\mathcal{E} + \lambda_2 I_{|\mathcal{E}|}\right)^{-1} s, \quad (\hat\beta(\lambda))_{[p]\setminus\mathcal{E}} = 0$$

for some $\mathcal{E} \in [p]$ and $s \in \{-1, 1\}^p$.

The following result describes the relation between the solution of Elastic Net and the coefficient parameters λ. The proof of the result can be easily derived based on simple algebra.

Lemma B.4 ([14]). *Let A be an $r \times s$ matrix. Consider $B(\lambda) = (A^\top A + \lambda I_s)^{-1}$.*

1. *Each entry of $B(\lambda)$ is a rational polynomial $P_{ij}(\lambda)/Q(\lambda)$ for $i, j \in [s]$ with each P_{ij} of degree at most $s-1$, and Q of degree s.*
2. *Further, for $i = j$, P_{ij} has degree $s-1$ and leading coefficient 1, and for $i \neq j$, P_{ij} has degree at most $s-2$. Also, $Q(\lambda)$ has leading coefficient 1.*

B.1.2 Proofs of Main Theorems

We now give a detailed proof for the main structural results (Theorem 3.2). We start with a useful definition.

Definition 8 (Semi-algebraic sets, Algebraic curves). *A semi-algebraic sets of $\mathbb{R}^n$ is a finite union of sets of the form $\{x \in \mathbb{R}^n \mid p_i(x) \geq 0, \text{ for } i \in [m]\}$, where p_i are polynomials. An algebraic curve is the zero set of a polynomial in two dimensions.*

We will now restate and prove Theorem 3.2.

Theorem 3.2 (restated). *Let $\mathcal{H}_{EN} = \{h_{EN}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}^2_{>0}\}$ the class of Elastic Net validation loss function class. Consider the dual class $\mathcal{H}^*_{EN} = \{h^*_P : \mathcal{H}_{EN} \to \mathbb{R}_{\geq 0} \mid P \in \Pi_{m,p}\}$, where $h^*_P(h_{EN}(\lambda, \cdot)) = h_{EN}(\lambda, P)$. Then $\mathcal{H}^*_{EN}$ is $(\mathcal{F}, 3^p, \mathcal{G}, p3^p)$-piecewise decomposable, where the piece function class $\mathcal{F} = \{f_q : \mathcal{H}_{EN} \to \mathbb{R}\}$ consists at most 3^p rational function $f_{q_1,q_2} : h_{EN}(\lambda, \cdot) \mapsto \frac{q_1(\lambda_1,\lambda_2)}{q_2(\lambda_1,\lambda_2)}$ of degree at most $2p$, and the boundary function class $\mathcal{G} = \{g_r : \mathcal{H}_{EN} \to \{0, 1\}\}$ consists of semi-algebraic sets bounded by at most $p3^p$ algebraic curves $g_r : h_{EN}(\lambda, \cdot) \mapsto \mathbb{1}\{r(\lambda_1, \lambda_2) < 0\}$, where r is a polynomial of degree at most p.*

Proof. Given a problem instance $P = (X, y, X_{\text{val}}, y_{\text{val}}) \in \Pi_{m,p}$, from Lemma B.3, for each λ, the solution $\hat{\beta}(\lambda)$ of the Elastic Net can be characterized as follow

$$\hat{\beta}(\lambda) = (X_\mathcal{E}^\top X_\mathcal{E} + \lambda_2 I_{|\mathcal{E}|})^{-1} X_\mathcal{E}^\top y - \lambda_1 (X_\mathcal{E}^\top X_\mathcal{E} + \lambda_2 I_{|\mathcal{E}|})^{-1} s,$$

for some $\mathcal{E} \in [p]$ and $s \in \{\pm 1\}^p$. Therefore, the prediction $\hat{y}$ on any validation example with features $\boldsymbol{x} \in \mathbb{R}^p$ is

$$\hat{y} = \boldsymbol{x}\hat{\beta}(\lambda) = \boldsymbol{x}[(X_\mathcal{E}^\top X_\mathcal{E} + \lambda_2 I_{|\mathcal{E}|})^{-1} X_\mathcal{E}^\top y - \lambda_1 (X_\mathcal{E}^\top X_\mathcal{E} + \lambda_2 I_{|\mathcal{E}|})^{-1} s].$$

This implies that: for any region $R \subset \mathbb{R}^2_{>0}$, if the equicorrelation set and sign vector $(\mathcal{E}, s)$ is fixed over R, then the solution $\hat{\beta}(\lambda)$ and the prediction y corresponding to $\boldsymbol{x}$ is also fixed. Consequently, within any region R where $(\mathcal{E}, s)$ remains unchanged, Lemma B.4 establishes that the validation loss function $h_{\text{EN}}(\lambda, P)$ (associated with a given problem instance P) is a constant rational function of the form $\frac{q_1(\lambda_1,\lambda_2)}{q_2(\lambda_1,\lambda_2)}$, where q_1 and q_2 are polynomials of degree at most $2p$ (since $2\,|\mathcal{E}| \leq 2p$ by definition). Notably, there are at most 3^p distinct values of $(\mathcal{E}, s)$, which implies that $h_{\text{EN}}(\lambda, P)$ can take on at most 3^p different polynomial forms.

The only remaining task is to examine the semi-algebraic sets and algebraic curves that separates region R. Consider such region R, in which the equicorrelation set and sign $(\mathcal{E}, s)$ is fixed.

- *Condition for a feature enters $\mathcal{E}$*: consider a feature $j \notin \mathcal{E}$, the condition for j to enters $\mathcal{E}$ is

$$(\boldsymbol{x}_j^*)^\top (y^* - X_\mathcal{E}^* (c_1 - c_2 \lambda_1^*)) = \pm \lambda_1^*$$

where $c_1 = (X_\mathcal{E}^{*\top} X_\mathcal{E}^*)^{-1}$, $c_2 = (X_\mathcal{E}^{*\top} X_\mathcal{E}^*)^{-1} s$, $X^* = \frac{1}{\sqrt{1+\lambda_2}} \begin{bmatrix} X \\ \sqrt{\lambda_2} I_p \end{bmatrix}$, $y^* = \begin{bmatrix} y \\ 0 \\ \vdots \\ 0 \end{bmatrix}$. Simplifying the equation above, we have

$$\lambda_1^* - \frac{(\boldsymbol{x}_j^*)^\top X_\mathcal{E}^* (X_\mathcal{E}^* X_\mathcal{E}^*)^{-1} (X_\mathcal{E}^*)^\top y^* - (x_j^*)^\top y^*}{(\boldsymbol{x}_j^*)^\top X_\mathcal{E}^* (X_\mathcal{E}^{*\top} X_\mathcal{E}^*)^{-1} s \pm 1} = 0, \text{ or}$$

$$\lambda_1 (\boldsymbol{x}_j^\top (X_\mathcal{E} X_\mathcal{E}^{*\top})^{-1} X_\mathcal{E} s \pm 1) - \boldsymbol{x}_j^\top X_\mathcal{E} (X_\mathcal{E}^\top X_\mathcal{E} + \lambda_2 I_{|\mathcal{E}|})^{-1} X_\mathcal{E}^\top y - \boldsymbol{x}_j^\top y = 0,$$

which is an algebraic curve with the RHS is a polynomial of degree at most p.

- *Condition for a feature leaves $\mathcal{E}$*: consider a feature $j' \in \mathcal{E}$. Similar to the previous case, the condition for j' to leave $\mathcal{E}$ can be described by an algebraic curve with the RHS as a polynomial of degree at most p.

Finally, notice that there are at most $\sum_{i=0}^{p} \binom{p}{i}((p-i)+i) = p3^p$ curves, across which the equicorrelation set and sign $(\mathcal{E}, s)$ might change, which concludes the proof. □

Using the GJ framework (Theorem 3.1), one can show that if a function class $\mathcal{H}$ has its dual-class $\mathcal{H}^*$ is piece-wise decomposable (in the sense of Definition 3), and all the piece and boundary functions are rational functions with upper bounded degree, then $\text{Pdim}(\mathcal{H})$ is upper bounded.

Lemma B.5. *Consider the function class $\mathcal{H} = \{h(a,\cdot) : \mathcal{X} \to \mathbb{R} \mid a \in \mathbb{R}^n\}$ be a function class parameterized by $a \in \mathbb{R}^W$. Consider the dual class $\mathcal{H}^* = \{h_x(\cdot) : \mathbb{R}^n \to \mathbb{R} \mid x \in \mathcal{X}\}$, where $h_x(a) = h(a,x)$. Assume that $\mathcal{H}^*$ is $(\mathcal{F}, k_\mathcal{F}, \mathcal{G}, k_\mathcal{G})$ piece-wise decomposable, and $\mathcal{F}$, $\mathcal{G}$ contains only rational functions in a of degree at most Δ. Then* $\text{Pdim}(\mathcal{F}) = O(n\log(\Delta(k_\mathcal{F} + k_\mathcal{G})))$.

Proof. Given an input $x \in \mathcal{X}$ and a threshold $t \in \mathbb{R}$, for any function $h(a,\cdot) \in \mathcal{H}$ corresponding to parameter a, consider the computation $\Gamma_{x,t} : \mathcal{H} \to \{0,1\}$, where

$$\Gamma_{x,t}(h(a,\cdot)) = \mathbb{1}\{h(a,x) - t \geq 0\}, \text{ for any } h(a,\cdot) \in \mathcal{H}.$$

Our goal now is to show that $\Gamma_{x,t}$ is a GJ algorithm in the sense of Definition 1.

From assumptions, we know that the dual class $\mathcal{H}^*$ is $(\mathcal{F}, k_\mathcal{F}, \mathcal{G}, k_\mathcal{G})$ piece-wise decomposable, where $\mathcal{F}, \mathcal{G}$ consists of rational function in a of degree at most Δ. This implies that for any $h(a,\cdot) \in \mathcal{H}$, the function $h_x(a) = h(a,x)$ is a rational function of a, of which the form is one of $k_\mathcal{F}$ rational functions in $\mathcal{F}$. Hence, to compute $\Gamma_{x,t}(h(a,\cdot))$, one needs to specify the closed-form of $h(a,\cdot)$, which is determined by binary-valued vector $b_a = \{g^{(1)}(a), \ldots, g^{(k_\mathcal{G})}(a)\}$, and can be calculated as conditional statements in the form $\mathbb{1}\{g^{(i)}(a) \geq 0\}$ for $i \in k_\mathcal{G}$. Therefore, we conclude that the computation of $\Gamma_{x,t}$ can be described by a GJ algorithm.

The predicate complexity of $\Gamma_{x,t}$ is the total number of functions in $\mathcal{F}$ and $\mathcal{G}$, which is equal to $k_\mathcal{F} + k_\mathcal{G}$. The degree of $\Gamma_{x,t}$ is the maximum degree of rational functions in $\mathcal{F}$ and $\mathcal{G}$, which is Δ from assumptions. From Theorem 3.1, we conclude that $\text{Pdim}(\mathcal{F}) = O(n\log(\Delta(k_\mathcal{F} + k_\mathcal{G}))$. □

Theorem 3.3. *Let $\mathcal{H}_{EN} = \{h_{EN}(\lambda,\cdot) : \Pi \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}^2_{>0}\}$ be the Elastic Net validation loss function class that maps problem instance P to validation loss $\ell_{val}(\lambda, P)$. Then* $\text{Pdim}(\mathcal{H}_{EN})$ *is $O(p)$.*

Proof. Given a problem instance $P \in \Pi_{m,p}$ and a threshold $t \in \mathbb{R}$, for any validation loss function $h_{\text{EN}}(\lambda,\cdot) \in \mathcal{H}_{\text{EN}}$, consider the computation $\Gamma_{P,t} : \mathcal{H}_{\text{EN}} \to \{0,1\}$, where

$$\Gamma_{P,t}(h(\lambda,\cdot)) = \mathbb{1}\{h(\lambda,P) - t \geq 0\}, \text{ for any } h(\lambda,\cdot) \in \mathcal{H}_{\text{EN}}.$$

From Theorem 3.2, for a given problem instance P, we know that the dual-class $\mathcal{H}^*_{\text{EN}}$ is $(\mathcal{F}, 3^p, \mathcal{G}, p3^p)-$piecewise decomposable, where $\mathcal{F}$ consists at most 3^p rational function of degree at most $2p$, and $\mathcal{G}$ consists of at most $p3^p$ algebraic curves of degree at most p. From Lemma B.5, $\text{Pdim}(\mathcal{H}_{\text{EN}}) = O(2\log(2p(p+1)3^p) = O(p)$.

□

B.2 Lower bound

We now instantiate a formal proof for Theorem 3.5.

Theorem 3.5 (restated). *Let $\mathcal{H}_{LASSO}$ be a set of functions $\{h_{LASSO}(\lambda,\cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}^+\}$ that map a regression problem instance $P \in \Pi_{m,p}$ to the validation loss $h_{LASSO}(\lambda, P)$ of LASSO trained with regularization parameter λ. Then* $\text{Pdim}(\mathcal{H}_{LASSO})$ *is $\Omega(p)$.*

Proof. Our proof of the lower bound in Theorem 3.5 builds on the "adversarial strategy" due to [35], where a data set (X, y) is constructed with the largest possible number of segments in the LASSO regularization path, for any p. Here we will include and discuss the main results from [35] that are useful in understanding our proof.

Our approach is to construct $N = p$ problem instances such that all 2^N above-below patterns (w.r.t. witness values) for the validation loss are achieved by choosing appropriate points (λ values) on the piecewise linear regularization path of the training instance, by utilizing the property that all unsigned sparsity patterns are achieved by the construction of [35]. In more detail, recall that the *signed* sparsity pattern $\{\eta_1, \ldots, \eta_k\}$ of a piecewise-linear regularization path P for dataset (X, y) is a sequence of vectors in $\{\pm 1, 0\}^p$ corresponding to the signs of the coefficients of the LASSO fit $\hat{\beta}^{(X,y)}(\lambda)$ in consecutive pieces of P, i.e. $\eta_j = (\text{sign}(\hat{\beta}_i^{(X,y)}(\lambda_j)))_{i=1}^p$ where λ_j corresponds to an interior point of the j-th piece of P. Let's further denote by $U_P = \{\overline{\eta}_j \mid 1 \leq j \leq k\}$ where $\overline{\eta}_j = (|\eta_{j1}|, \ldots, |\eta_{jp}|) \in \{0, 1\}^p$ as the *unsigned* sparsity pattern of path P.

We use the same training set (X, y) (but different validation sets) across our problem instances, namely the one with $(3^p + 1)/2$ segments constructed by Mairal and Yu (Theorem 1 of [35]). A useful property of this problem instance is that it achieves all the unsigned sparsity patterns, which follows from the following proposition.

Proposition B.6 ([35]). *Consider y in $\mathbb{R}^n$ and X in $\mathbb{R}^{n \times p}$ such that $X_{\mathcal{E}}$ is full rank for each $\mathcal{E} \subseteq [p]$ and y is in the span of X. Denote by P the regularization path of the Lasso problem corresponding to (X, y), and by k the number of linear segments of P. Then, there exist y' in $\mathbb{R}^{n+1}$ and X' in $\mathbb{R}^{(n+1)\times(p+1)}$ such that the regularization path P' of the Lasso problem associated to (X', y') has $3k - 1$ linear segments. Moreover, let $\{\eta_1 = 0, \eta_2, \ldots, \eta_k\}$ denote the sequence of sparsity patterns in $\{-1, 0, 1\}^p$ of P (the coordinate-wise signs of the solutions $\hat{\beta}^{(X,y)}(\lambda)$), ordered from large to small values of λ. The sequence of sparsity patterns in $\{-1, 0, 1\}^{p+1}$ of the new path P' is the following:*

$$\left\{ \begin{bmatrix} \eta_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \eta_2 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} \eta_k \\ 0 \end{bmatrix}, \begin{bmatrix} \eta_k \\ 1 \end{bmatrix}, \begin{bmatrix} \eta_{k-1} \\ 1 \end{bmatrix}, \ldots, \begin{bmatrix} \eta_1 = 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -\eta_2 \\ 1 \end{bmatrix}, \ldots, \begin{bmatrix} -\eta_k \\ 1 \end{bmatrix} \right\}.$$

Formally, one could use a simple inductive argument to establish the above claim. In the base case ($p = 1$), $X = y = [1]$ and it is easy to verify that the regularization path P_1 consists of two segments with $U_{P_1} = \{0, 1\}$. In the inductive case ($p + 1$ features), consider the first $2k$ sign patterns for the path P' in Proposition B.6. Using the inductive hypothesis, it is readily verified that the number of unsigned sparsity patterns in the regularization path P' is $|U_{P'}| = 2|U_P| = 2^{p+1}$.

In other words, all subsets of the p features appear as "active sets" of coefficients along the regularization path of the training set (X, y). By carefully setting the validation sets across the p problem instances in our proof of Theorem 3.5, we are able to ensure that the validation loss is non-zero exactly in the subset of problems corresponding to the unsigned sparsity patterns of $\hat{\beta}^{(X,y)}(\lambda)$. Thus, the property that all 2^p unsigned sparsity patterns are achieved for certain values of λ implies that all 2^N validation loss patterns are achieved w.r.t. witnesses 0^p. □

C Boundedness results for validation loss function classes of Elastic Net and Regularized Logistic Regression

In this section, we will give a formal guarantee for the boundedness of the validation loss function class of Elastic Net $\mathcal{H}_{\text{EN}}$ and Regularized Logistic Regression $\mathcal{H}_{\text{RLR}}$, which is essential for establishing learning guarantees for both function classes.

C.1 Boundedness of the validation loss function class of Elastic Net

The following lemma essentially shows that under mild assumptions on the value of data and the search space of hyperparameters, the validation loss function class $\mathcal{H}_{\text{EN}}$ is uniformly bounded by some constant $H > 0$.

Lemma C.1. *Under Assumptions 1 and 2, there exists a uniform constant $H > 0$ so that for all $h_{EN}(\lambda, \cdot) \in \mathcal{H}_{EN} = \{h_{EN}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$, we have $\|h_{EN}(\lambda, \cdot)\|_\infty = \sup_{P \in \Pi_{m,p}} |h_{EN}(\lambda, P)| \leq H$.*

Proof. For any problem instance $P = (X, y, X_{\text{val}}, y_{\text{val}}) \in \Pi_{m,p}$, and for any $\lambda = (\lambda_1, \lambda_2) \in [\lambda_{\min}, \lambda_{\max}]^2$, consider the optimization problem for training set

$$\operatorname*{argmin}_{\beta} F(\beta), \tag{3}$$

where $F(\beta) = \frac{1}{2m}\|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2$. If we set $\beta = \vec{0}$, we have

$$F(\vec{0}) = \frac{1}{2m}\|y\|_2^2 \leq C,$$

for some constant C that only depends on R_2, due to Assumption 2. Let $\hat{\beta}_{(X,y)}(\lambda)$ be the optimal solution of 3, we have

$$C \geq F(\hat{\beta}_{(X,y)}(\lambda)) \geq \lambda_1 \left\|\hat{\beta}_{(X,y)}(\lambda)\right\|_1 + \lambda_2 \left\|\hat{\beta}_{(X,y)}(\lambda)\right\|_2^2.$$

Therefore, for any problem instance P, the solution of the training optimization problem $\hat{\beta}_{(X,y)}(\lambda)$ has bounded norm, i.e. $\left\|\hat{\beta}_{(X,y)}(\lambda)\right\|_1, \left\|\hat{\beta}_{(X,y)}(\lambda)\right\|_2^2 \leq \frac{C}{\lambda_{\min}}$, which implies

$$h_{\text{EN}}(\lambda, P) = \frac{1}{2m}\left\|y_{\text{val}} - \hat{\beta}_{(X,y)}(\lambda)X_{\text{val}}\right\|_2^2 \leq \frac{1}{2m}\|y_{\text{val}}\|_2^2 + \frac{1}{2m}\left\|\hat{\beta}_{(X,y)}(\lambda)X_{\text{val}}\right\|_2^2 \leq H,$$

for some constant H (that only depends on R_1, R_2 and $\lambda_{\min}$). □

C.2 Boundedness of the validation loss function class of Regularized Logistic Regression

Using similar argument, we also have the following claim for the boundedness of validation loss function class of Regularized Logistic Regression.

Lemma C.2. *There exists a uniform constant $H > 0$ so that for all $h_{RLR}(\lambda, \cdot) \in \mathcal{H}_{RLR} = \{h_{RLR}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$, we have $\|h_{RLR}(\lambda, \cdot)\|_\infty = \sup_{P \in \Pi_{m,p}} |h_{RLR}(\lambda, P)| \leq H$.*

D Lemmas and Proof Details for Section 4

In this section, we present the detailed proofs of main results in Section 4.

D.1 Connected Components and Classical Results

We first present some classical results which are useful for analyzing the approximation validation loss function class $\mathcal{H}_{\text{RLR}}^{(\epsilon)}$.

We recall a well-known notion to analyze the pseudo-dimension of a function class, called the *solution set components bound* [33]. The bound on the solution set components essentially refers to the largest number of connected components within the parameter space of a parameterized function class $\mathcal{F}$. This component is generated from the solution set of a system of equations, corresponding to zero sets of functions in $\mathcal{F}$.

Definition 9 ([33]). *Let $\mathcal{F}$ be a set of real-valued functions defined on $\mathbb{R}^W$. We say that $\mathcal{F}$ has solution set components bound B if for any $1 \leq K \leq W$ and any $\{f_1, \ldots, f_K\} \subseteq \mathcal{F}$ that has regular zero-set intersections, we have*

$$\max_{K \leq W} CC\left(\bigcap_{i=1}^{K}\{a \in \mathbb{R}^W : f_i(a) = 0\}\right) = B$$

where $CC(X)$ is the number of connected components of X.

Let us now introduce a definition for the growth function of a binary-valued class function $\mathcal{H}$. This concept essentially quantifies the maximum number of distinct sign patterns $\{h(x_1), \ldots, h(x_m)\}$ that can be observed when we vary the function h across $\mathcal{H}$, considering a set of data points $x_1, \ldots, x_m$.

Definition 10 (Growth function, [33]). *Given m samples $x_1, \ldots, x_m \in \mathcal{X}$ and let $S = \{x_1, \ldots, x_m\}$. Consider a class function $\mathcal{H}$, of which each $h \in \mathcal{H}$ is a function from $\mathcal{X}$ to $\{-1, 1\}$, and let*

$$\mathcal{H}_S = \{(h(x_1), \ldots, h(x_m)) : h \in \mathcal{H}\}$$

is the total number of possible ways that S can be classified by $\mathcal{H}$. Then the growth function $G_{\mathcal{H}}(m)$ is defined as

$$G_{\mathcal{H}}(m) = \sup_{x_1, \ldots, x_m} |\mathcal{H}_S|$$

The next classical result establishes a connection between the growth function and the solution set components bound.

Theorem D.1 (Growth function bound, [33]). *Suppose that $\mathcal{F}$ is a class of real-valued functions defined on $\mathbb{R}^W \times \mathcal{X}$, and that $\mathcal{H}$ is defined as $\{sgn(f) : f \in \mathcal{F}\}$. If $\mathcal{F}$ is closed under addition of constants, has solution set components bound B, and functions in $\mathcal{F}$ are C^W in their parameters, then*

$$G_{\mathcal{H}}(m) \leq B\left(\frac{em}{W}\right)^W$$

for $m \geq W$.

D.2 The Empirical Rademacher Complexity of Approximate Logistic Validation Loss

We first restate important properties of the approximation solution $\beta^{(\epsilon)}(\lambda)$ accquired using Algorithm 1 (2) in RLR with ℓ_1 (ℓ_2) constraint.

Theorem 4.1 (restated) ([26]). *Given a problem instance $P = (X, y, X_{val}, y_{val}) \in \Pi_{m,p}$, for small enough ϵ, if we use Algorithm 1 (2) to approximate the solution $\hat{\beta}_{(X,y)}(\lambda)$ of RLR under ℓ_1 (ℓ_2) constraint by $\beta^{(\epsilon)}_{(X,y)}(\lambda)$ then there is a uniform bound $O(\epsilon^2)$ on the error $\|\hat{\beta}_{(X,y)}(\lambda) - \beta^{(\epsilon)}_{(X,y)}(\lambda)\|_2$ for any $\lambda \in [\lambda_{\min}, \lambda_{\max}]$.*

For any $\lambda \in [\lambda_t, \lambda_{t+1}]$, where $\lambda_k = \lambda_{\min} + k\epsilon$, the approximate solution $\beta^{(\epsilon)}(\lambda)$ is calculated by

$$\beta^{(\epsilon)}_{(X,y)}(\lambda) = \beta^{(\epsilon)}_t - \left[\nabla^2 l\left(\beta^{(\epsilon)}_t, (X,y)\right)_{\mathcal{A}}\right]^{-1} \cdot \left[\nabla l\left(\beta^{(\epsilon)}_t, (X,y)\right)_{\mathcal{A}} + \lambda \operatorname{sgn}\left(\beta^{(\epsilon)}_t\right)_{\mathcal{A}}\right] = a_t\lambda + b_t,$$

if we use Algorithm 1 for RLR under ℓ_1 constraint, or

$$\beta^{(\epsilon)}_{(X,y)}(\lambda) = \beta^{(\epsilon)}_t - \left[\nabla^2 l\left(\beta^{(\epsilon)}_t, (X,y)\right) + 2\lambda_{t+1} I\right]^{-1} \cdot \left[\nabla l\left(\beta^{(\epsilon)}_t, (X,y)\right) + 2\lambda\beta^{(\epsilon)}_t\right] = a'_t\lambda + b'_t,$$

if we use Algorithm 2 for RLR under ℓ_2 constraint.

The uniform error bound in Theorem 4.1 directly implies the error bound between he validation loss function $h^{(\epsilon)}_{\text{RLR}}(\lambda, P)$ and its approximation $h^{(\epsilon)}_{RLR}(\lambda, P)$. As in prior work [2], we will omit dependence on $\lambda_{\min}, \lambda_{\max}, R$ in our asymptotic upper bounds below.

Lemma 4.2 (restated). *The approximation error of the validation loss function is uniformly upper-bounded*

$$|h^{(\epsilon)}_{RLR}(\lambda, P) - h_{RLR}(\lambda, P)| = O(\epsilon^2) \text{ , for all } \lambda \in [\lambda_{\min}, \lambda_{\max}].$$

Proof. Using triangle inequality and the 1-Lipschitzness of $\mathcal{L}_{log}(z) := \log(1 + e^{-z})$ we have that

$$\begin{aligned}
\left|h^{(\epsilon)}_{\text{RLR}}(\lambda, P) - h_{\text{RLR}}(\lambda, P)\right| &= \frac{1}{m'}\left|\sum_{i=1}^{m'}[\mathcal{L}_{log}(y_i x_i^\top \beta^{(\epsilon)}_{(X,y)}(\lambda)) - \mathcal{L}_{log}(y_i x_i^\top \hat{\beta}_{(X,y)}(\lambda))]\right| \\
&\leq \frac{1}{m'}\sum_{i=1}^{m'}\left|\mathcal{L}_{log}(y_i x_i^\top \beta^{(\epsilon)}_{(X,y)}(\lambda)) - \mathcal{L}_{log}(y_i x_i^\top \hat{\beta}_{(X,y)}(\lambda))\right| \\
&\leq \frac{1}{m'}\sum_{i=1}^{m'}\left|y_i x_i^\top \beta^{(\epsilon)}_{(X,y)}(\lambda) - y_i x_i^\top \hat{\beta}_{(X,y)}(\lambda)\right|.
\end{aligned}$$

Using Hölder's inequality, Assumption 1, and Theorem 4.1, for any $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, we further have

$$\begin{aligned}
\left|h^{(\epsilon)}_{\text{RLR}}(\lambda, P) - h_{\text{RLR}}(\lambda, P)\right| &\leq \frac{1}{m'}\sum_{i=1}^{m'}\left|y_i x_i^\top \beta^{(\epsilon)}_{(X,y)}(\lambda) - y_i x_i^\top \hat{\beta}_{(X,y)}(\lambda)\right| \\
&= \frac{1}{m'}\sum_{i=1}^{m'}\left|y_i x_i^\top \left(\beta^{(\epsilon)}_{(X,y)}(\lambda) - \hat{\beta}_{(X,y)}(\lambda)\right)\right| \\
&\leq \frac{1}{m'}\sum_{i=1}^{m'}\|x_i\|_2 \left\|\hat{\beta}_{(X,y)}(\lambda) - \beta^{(\epsilon)}_{(X,y)}(\lambda)\right\|_2 \\
&\leq \left\|\hat{\beta}_{(X,y)}(\lambda) - \beta^{(\epsilon)}_{(X,y)}(\lambda)\right\|_2 \max_i \|x_i\|_2 \\
&= O(\epsilon^2).
\end{aligned}$$

□

D.3 Learning the Regularization Hyperparameter in Logistic Regression

In this section, we give a formal proof for Theorem 4.3. We begin by revisiting a fundamental result that proves invaluable when examining function classes that incorporate exponential functions.

Lemma D.2 ([43]). *Let Q_i ($i \leq m$) be elements of the polynomial ring $\boldsymbol{R}[y_1, \ldots, y_l, e^{\Lambda_1}, \ldots, e^{\Lambda_q}]$, where Λ_i are linear function of $y_1, \ldots, y_l$. Suppose that the system*

$$Q_1 = \cdots = Q_m = 0$$

is regular for $m \leq l$. If Q_i has degree at most d (in $y_1, \ldots, y_l, e^{\Lambda_1}, \ldots, e^{\Lambda_q}$), then the system above has the connected components bound

$$B_M = 2^{q(q-1)/2} d^l [(l+1)(d+1)]^{l+q}.$$

We now present the formal proof of Theorem 4.3.

Theorem 4.3 (restated). *Consider the RLR under ℓ_1 (or ℓ_2) constraint with parameter $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ that take a problem instance P drawn from an unknown problem distribution $\mathcal{D}$ over $\Pi_{m,p}$ under Assumptions 1 and 2. Consider the approximation validation loss function class $\mathcal{H}^{(\epsilon)}_{RLR} = \{h^{(\epsilon)}_{RLR}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$, where*

$$h^{(\epsilon)}_{RLR}(\lambda, P) = \frac{1}{m'}\sum_{i=1}^{m'} \log(1 + \exp(-y_i x_i^\top \beta^{(\epsilon)}_{(X,y)}(\lambda)))$$

is the approximate validation loss. Then we have $\text{Pdim}(\mathcal{H}^{(\epsilon)}_{RLR}) = O(m^2 + \log(1/\epsilon))$. Given any set $\mathcal{S}$ of T problem instances drawn from a problem distribution $\mathcal{D}$ over $\Pi_{m,p}$, the empirical Rademacher complexity $\hat{\mathscr{R}}(\mathcal{H}^{(\epsilon)}_{RLR}, \mathcal{S}) = O(H\sqrt{(m^2 + \log(1/\epsilon))/T})$, where H is the upperbound of original validation loss function class $\mathcal{H}_{RLR}$ (under Assumptions 1 and 2)..

Proof. The proof consists of following steps.

Step 1: Simplifying the analysis of $\mathcal{H}^{(\epsilon)}_{\text{RLR}}$ by considering an alternative function class.

Consider any $h(\lambda, \cdot) \in \mathcal{H}^{(\epsilon)}_{\text{RLR}}$, by definition, we have

$$h(\lambda, P) = \frac{1}{m'}\sum_{i=1}^{m'} \log(1 + \exp(-y_i x_i^\top \beta^{(\epsilon)}_{(X,y)}(\lambda)) = \frac{1}{m'} \log\left(\prod_{i=1}^{m'}(1 + \exp(-y_i x_i^\top \beta^{(\epsilon)}_{(X,y)}(\lambda)))\right).$$

From Lemma A.1 and note that $\log(\cdot)$ is a strictly monotonic, continuous function, analyzing the pseudo-dimension of $\mathcal{H}^{(\epsilon)}_{\text{RLR}}$ is equivalent to analyzing the VC-dimension of the class function $\mathcal{G}_{\text{RLR}} = \{\text{sign}(g_\lambda) : \Pi_{m,p} \times \mathbb{R} \to \{-1, 1\} \mid \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$, where

$$g_\lambda(P, \tau) = \prod_{i=1}^{m'}(1 + \exp(-y_i x_i^\top \beta^{(\epsilon)}_{(X,y)}(\lambda))) - \tau, \tag{4}$$

where τ is a new variable.

Step 2: Using the piece-wise linear property of approximation solution $\beta^{(\epsilon)}_{(X,y)}(\lambda)$, bounding the number of distinct sign patterns of $\{\text{sign}(g_\lambda(P^{(1)}, \tau_1), \ldots, \text{sign}(g_\lambda(P^{(N)}, \tau_N)\}$, where $(P^{(i)}, \tau_i) \in \Pi_{m,p} \times \mathbb{R}$ for $i \in [N]$, when varying $\lambda \in [\lambda_t, \lambda_{t+1}]$.

Consider N problem instances $\{P^{(1)}, \ldots P^{(N)}\}$ where $P^{(i)} \in \Pi_{m,p}$ for $i \in [N]$, and N corresponding thresholds $\tau_1, \ldots, \tau_N$, to bound the number of distinct sign patterns $\{\text{sign}(g_\lambda(P^{(1)}, \tau_1), \ldots, \text{sign}(g_\lambda(P^{(N)}, \tau_N)\}$ when varying $\lambda \in [\lambda_t, \lambda_{t+1}]$, we can use Theorem D.1 and bound the solution set components bound B, where

$$B = \max_{K \leq 1} CC\left(\bigcap_{i=1}^{K} \{\lambda \in [\lambda_t, \lambda_{t+1}] : g_i(\lambda)\}\right),$$

where $g_i(\lambda) = g_\lambda(P^{(i)}, \tau_i)$. From Theorem 4.3 (restated), $\beta^{(\epsilon)}_{(X^{(i)}, y^{(i)})}(\lambda)$ is a linear function of λ, which implies $g_i(\lambda)$ ($i \in [1]$) is element of polynomials ring $\boldsymbol{R}[\lambda, e^{\Lambda_1}, \ldots, e^{\Lambda_q}]$, where Λ_j is linear function of λ for $j \in [q]$.

Since $P^{(i)} \in \Pi_{m,p}$, we have $m'_i \leq m$ where m'_i is the number of validation sample in validation set of $P^{(i)}$, which implies there is at most m function Λ_j (see Eq. 4). Also from Eq. 4, we can see that $g_i(\lambda)$ is a polynomial (in $[\lambda, e^{\Lambda_1}, \ldots, e^{\Lambda_q}]$) of degree at most m.

Following Lemma D.2, we conclude that $B \leq 2^{m(m-1)/2} m(2(m+1))^{m+1}$. Combining with Theorem D.1, we have the number of distinct sign patterns $\{\text{sign}(g_\lambda(P^{(1)}, \tau_1)), \ldots, \text{sign}(g_\lambda(P^{(N)}, \tau_N))\}$ is upper bounded by $2^{m(m-1)/2} m(2(m+1))^{m+1} \left(\frac{eN}{2}\right)^2$.

Step 3: Bounding the pseudo-dimension of $\mathcal{H}^{(\epsilon)}_{\text{RLR}}$.

Note that there are total $(\lambda_{\max} - \lambda_{\min})/\epsilon$ pieces in which $\beta^{(\epsilon)}(\lambda)$ is linear function of λ. Therefore, using result from **Step 2**, the number of distinct sign patterns $\{\text{sign}(g_\lambda(\tau_1, P^{(1)})), \ldots, \text{sign}(g_\lambda(\tau_N, P^{(N)}))\}$ when varying $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ is upper bounded by $\frac{\lambda_{\max} - \lambda_{\min}}{\epsilon} 2^{m(m-1)/2} m(2(m+1))^{m+1} \left(\frac{eN}{2}\right)^2$. Solving the inequality

$$2^N \leq \frac{\lambda_{\max} - \lambda_{\min}}{\epsilon} 2^{m(m-1)/2} m(2(m+1))^{m+1} \left(\frac{eN}{2}\right)^2$$

implies $N = O(m^2 + \log(1/\epsilon))$, which means $\text{Pdim}(\mathcal{H}_{\text{RLR}}) = \text{VCdim}(\mathcal{G}_{\text{RLR}}) = O(m^2 + \log(1/\epsilon))$.

Step 4: Bounding the empirical Rademacher complexity of $\mathcal{H}^{(\epsilon)}_{\text{RLR}}$ over a set $\mathcal{S}$ of T problem instances. First, note that under Assumptions 1 and 2, there exists a universal upperbound H for the original validation loss function class $\mathcal{H}_{\text{RLR}}$. Combining with Lemma 4.2, we have the upperbound of $H + O(\epsilon^2)$ for the approximation loss function class $\mathcal{H}^{(\epsilon)}_{\text{RLR}}$.

Using result from **Step 3** and note that $\hat{\mathscr{R}}(\mathcal{H}^{(\epsilon)}_{\text{RLR}}, \mathcal{S}) = O((H + O(\epsilon^2))\sqrt{\text{Pdim}(\mathcal{H}^{(\epsilon)}_{\text{RLR}})/T})$, and $\epsilon = o(\sqrt{H})$, we concludes that $\hat{\mathscr{R}}(\mathcal{H}^{(\epsilon)}_{\text{RLR}}, \mathcal{S}) = O(H\sqrt{(m^2 + \log(1/\epsilon))/T})$ for any set $\mathcal{S}$ of T problem instances drawn from problem distribution $\mathcal{D}$ over $\Pi_{m,p}$. □

We now give a detailed proof for Theorem 4.4, which establishes the learning guarantee for the validation loss function class $\mathcal{H}_{\text{RLR}}$. We first recall an useful result by Balcan et al. [29], which allows us to upper bound the empirical Rademacher complexity of validation loss function class $\mathcal{H}_{\text{RLR}}$ via that of its approximation $\mathcal{H}^{(\epsilon)}_{\text{RLR}}$.

Theorem D.3 ([29]). *Let $\mathcal{F} = \{f_r \mid r \in \mathcal{R}\}$ and $\mathcal{G} = \{g_r \mid r \in \mathcal{R}\}$ consist of function mapping $\mathcal{X}$ to $[0, 1]$. For any $\mathcal{S} \subseteq \mathcal{X}$, $\hat{\mathscr{R}}(\mathcal{F}, \mathcal{S}) \leq \hat{\mathscr{R}}(\mathcal{G}, \mathcal{S}) + \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} \|f^*_x - g^*_x\|$.*

Theorem 4.4 (restated). *Consider the class function $\mathcal{H}_{RLR} = \{h_{RLR}(\lambda, \cdot) : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in [\lambda_{\min}, \lambda_{\max}]\}$ where $h_{RLR}(\lambda, P)$ is the validation loss corresponding to problem instance P and the ℓ_1 (ℓ_2) parameter λ. Under Assumptions 1 and 2, there exists a universal upperbound for the*

validation loss function class $\mathcal{H}_{RLR}$. Given any set $\mathcal{S}$ of T problem instances drawn from a problem distribution $\mathcal{D}$ over $\Pi_{m,p}$, for any $h_{RLR}(\lambda, \cdot) \in \mathcal{H}_{RLR}$, w.p. $1-\delta$ for any $\delta \in (0,1)$, we have

$$\left| \frac{1}{T} \sum_{i=1}^{T} h_{RLR}(\lambda, P^{(i)}) - \mathbb{E}_{P\sim\mathcal{D}}[h_{RLR}(\lambda, P)] \right| \leq O\left(H\sqrt{\frac{m^2 + \log(1/\epsilon)}{T}} + \epsilon^2 + \sqrt{\frac{1}{T}\log\frac{1}{\delta}} \right).$$

Proof. Theorem D.3, 4.3, and Lemma 4.2 directly imply that $\hat{\mathcal{R}}(\mathcal{H}_{\text{RLR}}, \mathcal{S}) = O(H\sqrt{(m^2 + \log(1/\epsilon))/T} + \epsilon^2)$, where $\mathcal{S}$ is a set of T problem instances drawn from problem distribution $\mathcal{D}$ over $\Pi_{m,p}$. Finally, classical uniform convergence bound based on Rademacher complexity concludes the result. □

D.4 An extension to 0-1 loss

We now give a formal proof for the learning guarantee of Regularized Logistic Regression hyperparameter tuning problem under 0-1 loss.

Theorem 4.5 (restated). *Let $\tau > 2\epsilon^2$ and $\delta \in (0,1)$, where ϵ is the approximation step-size. Then for any $n \geq s(\tau/2, \delta) = \Omega\left(\frac{H^2(m^2+\log\frac{1}{\epsilon})+\log\frac{1}{\delta}}{(\tau/2-\epsilon^2)^2}\right)$, if we have n problem instances $\{P^{(i)}, \ldots, P^{(n)}\}$ drawn i.i.d. from some problem distribution $\mathcal{D}$ over $\Pi_{m,p}$ to learn the regularization parameter λ^{ERM} for RLR via ERM, then*

$$\mathbb{E}_{P\sim\mathcal{D}}(h^{\text{0-1}}_{RLR}(\lambda^{ERM}, P))) \leq 4 \min_{\lambda\in[\lambda_{\min},\lambda_{\max}]} \mathbb{E}_{P\sim\mathcal{D}}(h_{RLR})(\lambda, P)) + 4\tau.$$

Proof. If $\epsilon < \sqrt{\tau}$, we can rearrange the result in Theorem 4.4 to get

$$s(\tau, \delta) \geq \Omega\left(\frac{H^2(m^2 + \log\frac{1}{\epsilon}) + \log\frac{1}{\delta}}{(\tau - \epsilon^2)^2}\right)$$

samples are sufficient for $(\tau, \delta)-$uniform convergence. Therefore, if $\tau > 2\epsilon^2$, then $\mathcal{H}_{\text{RLR}}$ is PAC-learnable with the ERM algorithm and $s(\tau/2, \delta)$ samples:

$$\mathbb{E}_{P\sim\mathcal{D}}(h_{\text{RLR}}(\lambda^{ERM}, P))) \leq \min_{\lambda\in[\lambda_{\min},\lambda_{\max}]} \mathbb{E}_{P\sim\mathcal{D}}(h_{\text{RLR}})(\lambda, P)) + \tau.$$

□