Provably tuning the ElasticNet across instances

Anonymous Author(s)
Affiliation
Address
email

Abstract

An important unresolved challenge in the theory of regularization is to set the regularization coefficients of popular techniques like the ElasticNet with general provable guarantees. We consider the problem of tuning the regularization parameters of Ridge regression, LASSO, and the ElasticNet across multiple problem instances, a setting that encompasses both cross-validation and multi-task hyperparameter optimization. We obtain a novel structural result for the ElasticNet which characterizes the loss as a piecewise rational function of the tuning parameters, with algebraic boundaries. We use this to bound the structural complexity of the regularized loss functions, and show generalization guarantees for tuning the ElasticNet regression coefficients in the statistical setting. We also consider the more challenging online learning setting, and show vanishing average expected regret relative to the optimal parameter pair. We also extend our results to tuning classification algorithms obtained by thresholding regression fits regularized by Ridge, LASSO or ElasticNet. Our results are the first general learning-theoretic guarantees, without strong assumptions on the data distribution, for this important class of problems. Our guarantees hold for both validation and popular information criterion objectives.

1 Introduction

Ridge regression [26, 36], LASSO [34], and their generalization the ElasticNet [21] are among the most popular algorithms in machine learning and statistics, with applications to linear classification, regression, data analysis, and feature selection [12, 39, 25, 16, 20]. Given a supervised dataset $(X, y) \in \mathbb{R}^{m\times p} \times \mathbb{R}^m$ with m datapoints and p features, these algorithms compute a linear predictor

$$\hat{\beta}^{(X,y)}_{\lambda_1,\lambda_2} = \arg\min_{\beta\in\mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2 \tag{1}$$

Here $\lambda_1, \lambda_2 \geq 0$ are *regularization coefficients* constraining the ℓ_1 and ℓ_2 norms, respectively, of the model β. For general λ_1 and λ_2 the above algorithm is the ElasticNet, while setting $\lambda_1 = 0$ recovers Ridge and setting $\lambda_2 = 0$ recovers LASSO.

These coefficients play a crucial role: in machine learning by controling the norm of β they imply provable generalization guarantees and prevent over-fitting in-practice, in data analysis their combined use yields parsimonious and interpretable models, and in Bayesian statistics, they correspond to imposing specific priors on β [30, 29]. In-practice, λ_2 regularizes β by uniformly shrinking all coefficients, while λ_1 encourages the model vector to be sparse. This means that while they do yield learning-theoretic and statistical benefits, setting them to be too high will cause models to under-fit the data. The question of how to set the regularization coefficients becomes even more unclear in the case of the ElasticNet, as one must juggle trade-offs between sparsity, feature correlation, and bias when setting both λ_1 and λ_2 simultaneously. As a result, there has been intense empirical and theoretical effort devoted to automatically tuning these parameter [23, 37, 27, 13]. Yet the

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

state-of-the-art is quite unsatisfactory – the proposed approaches are either heuristics without formal guarantees, optimize over a finite grid or random set instead of the full continuous domain, or involve very strong theoretical assumptions.

In this work, we study a variant on the above well-established and intensely studied formulation. The key distinction is that instead of a single dataset (X, y), we consider a collection of datasets or instances of the same underlying regression problem $(X^{(i)}, y^{(i)})$ and would like to learn λ_1, λ_2 which select a good (low validation loss) model using (1) on a new, unseen dataset. This can be useful to model practical settings, for example where new supervised data is obtained several times, or the set of features may change frequently [15]. We do not require all examples across datasets to be i.i.d. draws from the same data distribution, and can capture more general data generation scenarios like cross-validation, and multi-task learning [38]. Despite these advantages, we remark that our problem formulation is quite different from the standard single dataset setting. Our formulation treats the selection of regularization coefficients as *data-driven algorithm design* which has been used to study combinatorial problems [24, 3].

We establish a general set of conditions under which optimal tuning parameters for Ridge, LASSO, and ElasticNet can be found in our setting. We show guarantees both when the problems are generated statistically (generalization) and in an online fashion (regret). Our results for ElasticNet may be summarized as follows.

Theorem 1.1 (Informal)**.** *For ElasticNet regression problems with at most p parameters, for any problem distribution $\mathcal{D}$, $O\left(\frac{1}{\epsilon^2}(p^3 \log \frac{1}{\epsilon} + \log \frac{1}{\delta})\right)$ problems (datasets) are sufficient to learn an ϵ-approximation to the best (λ_1, λ_2), with probability at least $1 - \delta$. Moreover, for an online sequence of T problems with smoothly distributed labels y, there is an algorithm which sets the regularization coefficients with regret $\tilde{O}(\sqrt{T})$*

Because we only make weak κ-boundedness assumptions on the data within the problem instances, our results hold for both the multi-task case, when the instances are associated with different but possibly related problems, and in the important cross-validation setting, where we generated instances i.i.d. from a fixed dataset by using different random splits. For the latter case our results imply an upper bound of $\tilde{O}(\frac{p^3}{\epsilon^2})$ on the number of random splits needed to ϵ-approximate the best parameter with high probability, where p is the number of features. Our results also extend to information criterion objectives, and capture settings where the feature set or the set of available (or relevant) training examples may change over time. In summary, we make the following key contributions:

- We formulate the problem of tuning ElasticNet as a question of learning λ_1 and λ_2 simultaneously across multiple problem instances, with the objective possibly being based on popular information criteria. Our new formulation captures interesting relevant settings like cross-validation and multi-task learning.
- We provide a novel structural result (Lemma 2.1) that characterizes the loss of the ElasticNet fit. We show that the parameter space can be partitioned by polynomial curves of bounded degrees into pieces where the loss is a bivariate rational function.
- An important consequence of our structural result is a bound on the pseudo-dimension for the loss function class, which yields strong generalization bounds for tuning λ_1 and λ_2 simultaneously in the statistical learning setting (Theorem 3.2). In particular, our results have interesting implications for cross-validation – we provide (w.h.p.) upper bounds on number of repetitions for convergence.
- In the online setting, we show under very mild (weaker than prior theoretical work) data assumptions that the problem satisfies a dispersion condition [5, 8]. As a result we can tune all parameters across a sequence of instances appearing online and obtain vanishing regret w.r.t. an optimal parameter in hindsight over the sequence (Theorem 3.3).
- We show how to extend our results to regularized classifiers that perform thresholding on Ridge, LASSO or ElasticNet regression estimates, again providing strong generalization and online learning guarantees (Theorems 4.1, 4.2).

Key challenges and insights. A major challenge in learning the ElasticNet parameters is that the variation of the solution path as a function of λ_2 is hard to characterize. Indeed the original ElasticNet paper [40] suggests using the heuristic of grid search to learn a good λ_2, even though λ_1 may be exactly optimized by computing full solution paths (for each λ_2). We approach this indirectly by utilizing a

characterization of the LASSO solution by [35], which is based on the KKT (Karush–Kuhn–Tucker) optimality conditions, to arrive at a precise piecewise structure for the problem. In more detail, we use these conditions to come up with a set of algebraic curves (polynomial equations in λ_1 and λ_2) of bounded degrees, such that the set of possible discontinuities is contained within the zero-set of these curves, and the loss function behaves well in the each piece of the partition of the parameter domain by these curves. This characterization is crucial in establishing a bound on the structural complexity needed to provide strong generalization guarantees. We further show additional structure on these algebraic curves that (roughly speaking) imply that the curves do not concentrate in any region of the domain, allowing us to use the powerful recipe of [7] for online learning.

1.1 Related work

Model selection for Ridge regression, LASSO or ElasticNet is typically done by selecting the regularization parameter λ that works well for given data, although some parameter-free techniques for variable selection have been recently proposed [28]. Choosing 'optimal' parameters for tuning the regularization has been a subject of extensive theoretical and applied research. Much of this effort is heuristic [23, 27] or focused on developing tuning objectives beyond validation accuracy like AIC or BIC [1, 33] without providing procedures for provably optimizing them. The standard approach given a tuning objective is to optimize it over a grid or random set of parameters, for which there are guarantees [13], but this does not ensure optimality over the entire continuous tuning domain, especially since objectives such as 0-1 validation error or information criteria can have many discontinuities. Selecting a grid that is too fine or too coarse can result in either very inefficient or highly inaccurate estimates (respectively) for good parameters. Other guarantees make strong assumptions on the data distribution such as sub-Gaussian noise [37] or depend on unknown parameters that are hard to quantify in practice [19]. A successful line of work has focused on efficiently obtaining models for different values of λ_1 using regularization paths [18], but the guarantees are computational rather than learning-theoretic or statistical. In contrast, we provide principled approaches that guarantee near-optimality of selected parameters with high confidence over the entire continuous domain of parameters.

Data-driven algorithm design has proved successful for tuning parameters for a variety of combinatorial problems like clustering, integer programming, auction design and graph-based learning [6, 11, 10, 4]. We provide an application of these techniques to parameter tuning in a problem that is not inherently combinatorial by revealing a novel discrete structure. We identify the underlying piecewise structure of the ElasticNet loss function which is extremely effective in establishing learning-theoretic guarantees [9]. To exploit this piecewise structure, we analyze the learning-theoretic complexity of rational algebraic function classes and infer generalization guarantees. We also employ and extend general tools and techniques for online data-driven learning from [7, 4] to rational functions in order to prove our online learning guarantees for regularization coefficient tuning.

2 Preliminaries

Given data (X, y) with $X \in \mathbb{R}^{m\times p}$ and $y \in \mathbb{R}^m$, consisting of m labeled examples with p features, we seek estimators $\beta \in \mathbb{R}^p$ which minimize the regularized loss. Popular regularization methods like LASSO and ElasticNet can be expressed as computing the solution of an optimization problem given by

$$\hat{\beta}^{(X,y)}_{\lambda,f} \in \arg\min_{\beta\in\mathbb{R}^p} \|y - X\beta\|_2^2 + \langle \lambda, f(\beta)\rangle$$

where $f : \mathbb{R}^p \to \mathbb{R}^d_{\geq 0}$ gives the regularization penalty for estimator β, and $\lambda \in \mathbb{R}^d_{\geq 0}$ is the regularization parameter. d is the number of regularization parameters, $d = 1$ for Ridge and LASSO, $d = 2$ for the ElasticNet. Setting $f = f_2$ with $f_2(\beta) = \|\beta\|_2^2$ yields the ridge regression, and setting $f(\beta) = f_1(\beta) = \|\beta\|_1$ corresponds to LASSO. Also setting $f_{\text{EN}}(\beta) = (f_1(\beta), f_2(\beta))$ gives the ElasticNet with regularization parameter $\lambda = (\lambda_1, \lambda_2)$. Note that we use the same λ (with some notational overloading) to denote the regularization parameters for ridge, LASSO, or ElasticNet. We write $\hat{\beta}^{(X,y)}_{\lambda,f}$ as simply $\hat{\beta}_{\lambda,f}$ when the dataset (X, y) is clear from context. On any instance $x \in \mathbb{R}^p$ from the feature space, the prediction of the regularized estimator is given by the dot product

$\langle x, \hat{\beta}_{\lambda,f} \rangle$. The average squared loss over a dataset (X', y') with $X' \in \mathbb{R}^{m' \times p}$ and $y' \in \mathbb{R}^{m'}$ is given by $l_r(\hat{\beta}_{\lambda,f}, (X', y')) = \frac{1}{m'} \left\| y' - X' \hat{\beta}_{\lambda,f} \right\|_2^2$. By setting (X', y') to be the training data (X, y), we get the training loss $l_r(\hat{\beta}_{\lambda,f}, (X, y))$. We use $(X_{\text{val}}, y_{\text{val}})$ to denote a validation split.

Given a class of regularization algorithms $\mathcal{A}$ parameterized by regularization parameter λ over a set of problem instances $\mathcal{X}$, and given loss function $\ell : \mathcal{A} \times \mathcal{X} \to \mathbb{R}$ which measures the loss of any algorithm in $\mathcal{A}$ on any fixed problem instance, consider the set of functions $\mathcal{H}_{\mathcal{A}} = \{\ell(A, \cdot) \mid A \in \mathcal{A}\}$. Bounding the pseudo-dimension of $\mathcal{H}_{\mathcal{A}}$ gives a bound on the sample complexity for uniform convergence guarantees, i.e. a bound on the sample size n for which the algorithm $\hat{A}_S \in \mathcal{A}$ which minimizes the average loss on any sample S of size n drawn i.i.d. from any problem distribution $\mathcal{D}$ is guaranteed to be near-optimal with high probability [17]. See Appendix A for the relevant classic definitions and results. Define the *dual class* $\mathcal{H}^*$ of a set of real-valued functions $\mathcal{H} \subseteq 2^{\mathcal{X}}$ as $\mathcal{H}^* = \{h_x^* : \mathcal{H} \to \mathbb{R} \mid x \in \mathcal{X}\}$ where $h_x^*(h) = h(x)$. For a function class $\mathcal{H}$, showing that dual class $\mathcal{H}^*$ is piecewise-structured in the sense of Definition 1 and bounding the complexity of the duals of boundary and piece functions of $\mathcal{H}^*$ are useful to understand the learnability of $\mathcal{H}$ [9].

Definition 1 (Piecewise structured functions, [9])**.** *A function class $H \subseteq \mathbb{R}^{\mathcal{X}}$ that maps a domain $\mathcal{X}$ to $\mathbb{R}$ is* (F, G, k)-piecewise decomposable *for a class $G \subseteq \{0,1\}^{\mathcal{X}}$ of boundary functions and a class $F \subseteq \mathbb{R}^{\mathcal{X}}$ of piece functions if the following holds: for every $h \in H$, there are k boundary functions $g_1, \ldots, g_k \in G$ and a piece function $f_{\mathbf{b}} \in F$ for each bit vector $\mathbf{b} \in \{0,1\}^k$ such that for all $x \in \mathcal{X}$, $h(x) = f_{\mathbf{b}_x}(x)$ where $\mathbf{b}_x = (g_1(x), \ldots, g_k(x)) \in \{0,1\}^k$.*

To understand the proof insights, we need the definition of equicorrelation sets, the subset of features with maximum absolute correlation for any fixed λ_1, useful for characterizing LASSO/ElasticNet solutions. For any subset $\mathcal{E} \subseteq [p]$ of the features, we define $X_{\mathcal{E}} = (\ldots X_{*i} \ldots)_{i \in \mathcal{E}}$ as the $m \times |\mathcal{E}|$ matrix of columns X_{*i} of X corresponding to indices $i \in \mathcal{E}$. Similarly $\beta_{\mathcal{E}} \in \mathbb{R}^{|\mathcal{E}|}$ is the subset of estimators in β corresponding to indices in $\mathcal{E}$. We will assume all the feature matrixes X (for training datasets) are in the general position (Definition 5).

Definition 2 (Equicorrelation sets, [35])**.** *Let $\beta^* \in \arg\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda_1 ||\beta||_1$. The equicorrelation set corresponding to β^*, $\mathcal{E} = \{j \in [p] \mid |\boldsymbol{x}_j^T (y - X\beta^*)| = \lambda_1\}$, is simply the set of covariates with maximum absolute correlation. We also define the equicorrelation sign vector for β^* as $s = \mathsf{sign}(X_{\mathcal{E}}^T (y - X\beta^*))$.*

2.1 A key structural result

Consider the class of algorithms consisting of ElasticNet regressors for different values of $\lambda = (\lambda_1, \lambda_2) \in (0, \infty) \times [0, \infty)$. We assume $\lambda_1 > 0$ for technical simplicity (cf. [35]). We seek to solve problems of the form $P = (X, y, X_{\text{val}}, y_{\text{val}}) \in \mathcal{R}_{m_1, p_1, m_2} = \mathbb{R}^{m_1 \times p_1} \times \mathbb{R}^{m_1} \times \mathbb{R}^{m_2 \times p_1} \times \mathbb{R}^{m_2}$, where (X, y) is the training set and $(X_{\text{val}}, y_{\text{val}})$ is the validation set. Our problems are in the problem space given by $\Pi_{m,p} = \bigcup_{m_1 \geq 0, m_2 \leq m, p_1 \leq p} \mathcal{R}_{m_1, p_1, m_2}$.

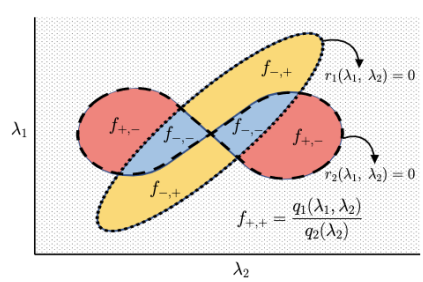

Figure 1: An illustration of the piecewise structure of the ElasticNet loss, as a function of the regularization parameters, for a fixed problem instance. Pieces are regions where some bounded degree polynomials (r_1, r_2) have a fixed sign pattern (one of $\pm 1, \pm 1$), and in each piece the loss is a fixed (rational) function.

Let $\mathcal{H}_{\text{EN}} = \{\ell_{\text{EN}}(\lambda, \cdot) \mid \lambda \in (0, \infty) \times [0, \infty)\}$ denote the set of loss functions for the class of algorithms consisting of ElasticNet regressors for different values of $\lambda \in \mathbb{R}^+ \times \mathbb{R}_{\geq 0}$. Additionally, we will consider information criterion based loss functions, $\ell_{\text{EN}}^{\mathsf{AIC}}(\lambda, P) = \ell_{\text{EN}}(\lambda, P) + 2||\hat{\beta}_{\lambda, f_{\text{EN}}}^{(X,y)}||_0$ and $\ell_{\text{EN}}^{\mathsf{BIC}}(\lambda, P) = \ell_{\text{EN}}(\lambda, P) + 2||\hat{\beta}_{\lambda, f_{\text{EN}}}^{(X,y)}||_0 \log m$ [1, 33]. Let $\mathcal{H}_{\text{EN}}^{\mathsf{AIC}}$ and $\mathcal{H}_{\text{EN}}^{\mathsf{BIC}}$ denote the corresponding sets of loss functions. These criteria are popularly used to compute the squared loss on the training set, to give alternatives to cross-validation. We do not make any assumption on the relation between training and validation sets in our formulation, so our analysis can capture these settings as well.

We will now establish a piecewise structure of the dual class loss functions (Definition 1). A key observation is that if the signed equicorrelation set $(\mathcal{E}, s)$ (i.e. a subset of features $\mathcal{E} \subseteq [p]$ with the same maximum absolute correlation, assigned a fixed sign pattern $\{-1, +1\}^{|\mathcal{E}|}$, see Definition 2) is fixed, then the ElasticNet coefficients may be characterized (Lemma C.1) which gives the rational polynomial piece functions for a fixed sign pattern of functions in $\mathcal{G}$. The boundary functions are a collection of possible curves at which a covariate may enter or leave the set $\mathcal{E}$ and correspond to algebraic curves. More formally, we have the following.

Lemma 2.1. *Let $\mathcal{L}$ be a set of functions $\{l_\lambda : \Pi_{m,p} \to \mathbb{R}_{\geq 0} \mid \lambda \in \mathbb{R}^+ \times \mathbb{R}_{\geq 0}\}$ that map a regression problem instance $P \in \Pi_{m,p}$ to the validation loss $\bar{\ell}_{EN}(\lambda, P)$ of ElasticNet trained with regularization parameter $\lambda = (\lambda_1, \lambda_2)$. The dual class $\mathcal{L}^*$ is $(\mathcal{F}, \mathcal{G}, p3^p)$-piecewise decomposable, with $\mathcal{F} = \{f_q : \mathcal{L} \to \mathbb{R}\}$ consisting of rational polynomial functions $f_{q_1,q_2} : l_\lambda \mapsto \frac{q_1(\lambda_1,\lambda_2)}{q_2(\lambda_2)}$, where q_1, q_2 have degrees at most $2p$, and $\mathcal{G} = \{g_r : \mathcal{L} \to \{0, 1\}\}$ consisting of semi-algebraic sets bounded by algebraic curves $g_r : u_\lambda \mapsto \mathbb{I}\{r(\lambda_1, \lambda_2) < 0\}$, where r is a polynomial of degree 1 in λ_1 and at most p in λ_2.*

Proof. Let $P = (X, y, X_{\text{val}}, y_{\text{val}}) \in \Pi_{m,p}$ be a regression problem instance. By using the standard reduction to LASSO [40] and well-known characterization of the LASSO solution in terms of equicorrelation sets, we can characterize the solution of the Elastic Net as follows (Lemma C.1). The ElasticNet fit $\hat{\beta}_{\lambda,f_{EN}}$ satisfies

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

for some $\mathcal{E} \in [p]$ and $s \in \{-1, 1\}^p$. Thus for any $\lambda = (\lambda_1, \lambda_2)$, the prediction $\hat{y}$ on any validation example with features $\boldsymbol{x} \in \mathbb{R}^p$ satisfies (for some $\mathcal{E}, s \in 2^{[p]} \times \{-1, 1\}^p$)

$$\hat{y}_j = \boldsymbol{x}\hat{\beta}_{\lambda,f_{EN}} = \boldsymbol{x}(X_{\mathcal{E}}^T X_{\mathcal{E}} + \lambda_2 I_{|\mathcal{E}|})^{-1} X_{\mathcal{E}}^T y - \lambda_1 \boldsymbol{x}(X_{\mathcal{E}}^T X_{\mathcal{E}} + \lambda_2 I_{|\mathcal{E}|})^{-1} s$$

For any subset $R \subseteq \mathbb{R}^2$, if the signed equicorrelation set $(\mathcal{E}, s)$ is fixed over R, then the above observation, together with Lemma C.2 implies that the loss function $\ell_{\text{EN}}(\lambda, P)$ is a rational function of the form $\frac{q_1(\lambda_1,\lambda_2)}{q_2(\lambda_2)}$, where q_1 is a bivariate polynomial with degree at most $2|\mathcal{E}|$ and q_2 is univariate with degree $2|\mathcal{E}|$.

To show the piecewise structure, we need to demonstrate a set boundary functions $\mathcal{G} = \{g_1, \ldots, g_k\}$ such that for any sign pattern $\mathbf{b} \in \{0, 1\}^k$, the signed equicorrelation set $(\mathcal{E}, s)$ for the region with sign pattern $\mathbf{b}$ is fixed. To this end, based on the observation above, we will consider the conditions (on λ) under which a covariate may enter or leave the equicorrelation set. We will show that this can happen only at one of a finite number of algebraic curves (with bounded degrees).

Condition for joining $\mathcal{E}$. Fix $\mathcal{E}, s$. Also fix $j \notin \mathcal{E}$. If covariate j enters the equicorrelation set, the KKT conditions (Lemma B.1) applied to the LASSO problem corresponding to the ElasticNet (Lemma C.1) imply

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

where $c_1 = ({X_{\mathcal{E}}^*}^T X_{\mathcal{E}}^*)^{-1} {X_{\mathcal{E}}^*}^T y^*$ and $c_2 = ({X_{\mathcal{E}}^*}^T X_{\mathcal{E}}^*)^{-1} s$. Rearranging, and simplifying, we get

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

Using simple algebra, the denominator $(\boldsymbol{x}_j^*)^T (X_{\mathcal{E}} {X_{\mathcal{E}}^*}^T)^{-1} X_{\mathcal{E}}^* s \pm 1$ can be shown to not depend on λ_1 or λ_2. Using Lemma C.2, we get an algebraic curve $r_{j,\mathcal{E},s}(\lambda_1, \lambda_2) = 0$ with degree 1 in λ_1 and $|\mathcal{E}|$ in λ_2 corresponding to addition of $j \notin \mathcal{E}$ given $\mathcal{E}, s$.

Condition for leaving $\mathcal{E}$. Now consider a fixed $j' \in \mathcal{E}$, given fixed $\mathcal{E}, s$. The coefficient of j' will be zero for $\lambda_1^* = \frac{(c_1)_{j'}}{(c_2)_{j'}}$, which simplifies to $\lambda_1 (({X_{\mathcal{E}}}^T X_{\mathcal{E}} + \lambda_2 I_{|\mathcal{E}|})^{-1} s)_{j'} = (({X_{\mathcal{E}}}^T X_{\mathcal{E}} + \lambda_2 I_{|\mathcal{E}|})^{-1} {X_{\mathcal{E}}}^T y)_{j'}$. Again by Lemma C.2, we get an algebraic curve $r_{j',\mathcal{E},s}(\lambda_1, \lambda_2) = 0$ with degree 1 in λ_1 and at most $|\mathcal{E}|$ in λ_2 corresponding to removal of $j' \in \mathcal{E}$ given $\mathcal{E}, s$.

Putting together, we get $\sum_{i=0}^{p} 2^i \binom{p}{i} ((p - i) + i) = p3^p$ algebraic curves of degree 1 in λ_1 and at most p in λ_2, across which the signed equicorrelation set may change. These curves characterize

the complete set of points (λ_1, λ_2) at which $(\mathcal{E}, s)$ may possibly change. Thus by setting these $p3^p$ curves as the set of boundary functions $\mathcal{G}$, $\mathcal{E}, s$ is guaranteed to be fixed for each sign pattern, and the corresponding loss takes the rational function form shown above. □

The exact same piecewise structure can be established for the dual function classes for loss functions $\ell_{\text{EN}}^{\mathsf{AIC}}(\lambda, \cdot)$ and $\ell_{\text{EN}}^{\mathsf{BIC}}(\lambda, \cdot)$. This is evident from the proof of Lemma 2.1, since any dual piece has a fixed equicorrelation set, and therefore $||\beta||_0$ is fixed. Given this piecewise structure, a challenge to learning values of λ that minimize the loss function is that the function may not be differentiable (or may even be discontinuous, for the information criteria based losses) at the piece boundaries, making well-known gradient-based (local) optimization techniques inapplicable here. In the following we will show that techniques from data-driven design may be used to overcome this optimization challenge.

3 Learning to Regularize the ElasticNet

We will consider the problem of learning provably good ElasticNet parameters for a given problem domain, from multiple datasets (problem instances) either available as a collection (Section 3.1), or arriving online (Section 3.2). Our parameter tuning techniques also apply to simpler regression techniques typically used for variable selection, like LARS and LASSO, which are reasonable choices if the features are not multicollinear. Additional proof details for the results in this section are located in Appendix C.

3.1 Distributional Setting

Suppose we receive n instances of the regression problem $P^{(i)} = (X^{(i)}, y^{(i)}, X_{\text{val}}^{(i)}, y_{\text{val}}^{(i)}) \in \mathcal{R}_{m_i, p_i, m_i'} = \mathbb{R}^{m_i \times p_i} \times \mathbb{R}^{m_i} \times \mathbb{R}^{m_i' \times p_i} \times \mathbb{R}^{m_i'}$ for $i \in [n]$ generated i.i.d. from some problem distribution $\mathcal{D}$. On any given instance $P^{(i)}$ the loss is given by the squared loss on the validation set, $\ell_{\text{EN}}(\lambda, P^{(i)}) = l_r(\hat{\beta}_{\lambda, f_{\text{EN}}}^{(X^{(i)}, y^{(i)})}, (X_{\text{val}}^{(i)}, y_{\text{val}}^{(i)}))$. Our main result in this section is the following upper bound on the pseudo-dimension of the classes of loss functions for the ElasticNet, which implies that in our distributional setting it is possible to learn near-optimal values of λ with polynomially many problem instances.

Theorem 3.1. $\textsc{Pdim}(\mathcal{H}_{EN}) = O(p^3)$. *Further,* $\textsc{Pdim}(\mathcal{H}_{EN}^{\mathsf{AIC}}) = O(p^3)$ *and* $\textsc{Pdim}(\mathcal{H}_{EN}^{\mathsf{BIC}}) = O(p^3)$.

Proof Sketch. The crucial ingredient is the $(\mathcal{F}, \mathcal{G}, p3^p)$-piecewise decomposable structure for the dual class function $\mathcal{H}_{\text{EN}}^*$ established in Lemma 2.1. We can bound the pseudo-dimension of the dual class of piece functions $\mathcal{F}^*$ (a class of bivariate rational functions) by $O(\log p)$ by giving an upper bound (of $O(k^3 d^3)$) on the number of sign patterns over $\mathbb{R}^2$ induced by k algebraic curves of degree at most d. We can also bound the VC dimension of the dual class of boundary functions $\mathcal{G}^*$ (semi-algebraic sets in two variates) by $O(p^2)$ using a standard linearization argument. Finally, a powerful result from [9] (Theorem C.3) allows us to bound the pseudodimension of $\mathcal{H}$ by combining the above results. □

A key challenge to establish the theorem is providing new bounds on the pseudo-dimension of rational functions of bounded degrees . The upper bound above implies a guarantee on the sample complexity of learning the ElasticNet tuning parameter, using standard learning-theoretic results [2]. In our setting of learning from multiple problem instances, each sample is a dataset instance, so the sample complexity is simply the number of regression problem instances needed to learn the tuning parameters to any given approximation and confidence level.

Theorem 3.2 (Sample complexity of tuning the ElasticNet)**.** *Let $\mathcal{D}$ be an arbitary distribution over the problem space $\Pi_{m,p}$. There is an algorithm which given $n = O\left(\frac{1}{\epsilon^2}(p^3 \log \frac{1}{\epsilon} + \log \frac{1}{\delta})\right)$ problem samples drawn from $\mathcal{D}$, for any $\epsilon > 0$ and $\delta \in (0, 1)$, outputs a regularization parameter $\hat{\lambda}$ for the ElasticNet such that with probability at least $1 - \delta$ over the draw of the problem samples, we have that*

$$\left| \mathbb{E}_{P \sim \mathcal{D}}[\ell_{EN}(\hat{\lambda}, P)] - \min_{\lambda} \mathbb{E}_{P \sim \mathcal{D}}[\ell_{EN}(\lambda, P)] \right| \leq \epsilon$$

Proof. This follows from substituting our result in Theorem 3.1 into well-known generalization guarantee for function classes with bounded pseudo-dimensions (Theorem A.1). □

Discussion and applications. Note that we do not assume the set of features is the same across problem instances, so our approach can handle feature reset. Moreover, as a special case application, we consider the commonly used techniques of leave-one-out cross validation (LOOCV) and Monte Carlo cross validation (repeated random test-validation splits, typically independent and in a fixed proportion). Given a dataset of size m_{tr}, LOOCV would require m_{tr} regression fits which can be inefficient for large dataset size. Alternately, we can consider draws from a distribution $\mathcal{D}_{LOO}$ which generates problem instances P from a fixed dataset $(X, y) \in \mathbb{R}^{m+1\times p} \times \mathbb{R}^{m+1}$ by uniformly selecting $j \in [m+1]$ and setting $P = (X_{-j*}, y_{-j}, X_{j*}, y_j)$. Theorem 3.2 now implies that $\tilde{O}(\frac{p^3}{\epsilon^2})$ iterations are enough to determine a ElasticNet parameter $\hat{\lambda}$ with loss within ϵ (w.h.p.) of parameter λ^* obtained from running the full LOOCV. Similarly, we can define a distribution $\mathcal{D}_{MC}$ to capture the Monte Carlo cross validation procedure and determine the number of iterations sufficient to get an ϵ-approximation of the loss corresponding parameter selection with arbitrarily large number of runs of the procedure. Thus, in a very precise sense, our results answer the question 'how much cross-validation is enough' to effectively implement the above techniques.

3.2 Online Learning

Some of our results will require the data distribution to be of bounded density. A continuous probability distribution is said to be *κ-bounded* if the probability density function $p(x)$ satisfies $p(x) \leq \kappa$ for any x in the sample space. For example, the *normal* distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and standard deviation σ is $\frac{1}{\sigma\sqrt{2\pi}}$-bounded. We will also need the definition of *dispersion* which, informally speaking, captures how amenable a set of non-Lipschitz functions is to online learning by measuring the worst rate of occurrence of non-Lipschitzness (or discontinuities) between any pair of points in the domain. [5, 8] show that dispersion is necessary and sufficient for learning piecewise Lipschitz functions.

Definition 3. *Dispersion [7]. The sequence of random loss functions $l_1, \ldots, l_T$ is β-dispersed for the Lipschitz constant L if, for all T and for all $\epsilon \geq T^{-\beta}$, we have that, in expectation, at most $\tilde{O}(\epsilon T)$ functions (the soft-O notation suppresses dependence on quantities beside ϵ, T and β, as well as logarithmic terms) are not L-Lipschitz for any pair of points at distance ϵ in the domain $\mathcal{C}$. That is, for all T and for all $\epsilon \geq T^{-\beta}$, $\mathbb{E}\Big[\max_{\substack{\rho,\rho'\in\mathcal{C} \\ \|\rho-\rho'\|_2\leq\epsilon}} \big|\{t \in [T] \mid l_t(\rho) - l_t(\rho') > L\,\|\rho-\rho'\|_2\}\big|\Big] \leq \tilde{O}(\epsilon T)$.*

Formal setup. Suppose we receive a sequence of T instances of the ElasticNet regression problem $P^{(i)} = (X^{(i)}, y^{(i)}, X^{(i)}_{\text{val}}, y^{(i)}_{\text{val}})$ for $i \in [T]$ online. On any given instance $P^{(i)}$, the online learner is required to select the regularization parameter $\lambda^{(i)}$ without observing $y^{(i)}_{\text{val}}$, and experiences loss given by $\ell(\lambda^{(i)}, P^{(i)}) = l_c(\hat{\beta}^{(X^{(i)},y^{(i)})}_{\lambda^{(i)},f_{EN}}, (X^{(i)}_{\text{val}}, y^{(i)}_{\text{val}}))$. The goal is to minimize the regret w.r.t. choosing the best fixed parameter in hindsight for the same problem sequence, i.e. $R_T = \sum_{i=1}^{T} \ell(\lambda^{(i)}, P^{(i)}) - \min_{\lambda\in[0,\infty)} \sum_{i=1}^{T} \ell(\lambda, P^{(i)})$. We also define average regret as $\frac{1}{T}R_T$ and expected regret as $\mathbb{E}[R_T]$ where the expectation is over both the randomness of the loss functions and any random coins used by the online algorithm.

Note that this is more challenging than the distributional setting, since now we may be faced with adversarially designed problem sequences. Our results for this setting are therefore stronger, but they hold only under some mild data smoothness assumptions (while the distributional results are worst-case). We assume that all feature values and predictions are bounded, for training as well as validation examples, and that the predicted variable y in the training set comes from a κ-bounded (i.e. smooth) distribution. This distribution could model noise in the data, our assumption of smoothness is much weaker than sub-Gaussian noise assumptions in the literature. Note that our assumption also captures a common data preprocessing step, for example the jitter parameter in the popular Python library scikit-learn [31] adds a uniform noise to the y values to help model stability. Stated formally, we need the following two assumptions.

Assumption 1 (Boundedness)**.** *The predicted variable and all feature values are bounded by an absolute constant R, i.e.* $\max\{||X^{(i)}||_{\infty,\infty}, ||y^{(i)}||_\infty, ||X^{(i)}_{val}||_{\infty,\infty}, ||y^{(i)}_{val}||_\infty\} \leq R$.

Assumption 2 (Smooth predictions)**.** *The predicted variables $y^{(i)}$ in the training set are drawn from a joint κ-bounded distribution, i.e. for each i, the varibles $y^{(i)}$ have a joint distribution with probability density bounded by κ.*

Algorithm 1 Data-driven Regularization (ζ)

Input: Problems $(X^{(i)}, y^{(i)})$ and regularization penalty function f.
Hyperparameter: step size parameter $\zeta \in (0, 1]$.
Output: Regularization parameter $(\lambda_i)_{i\in[T]} \in C$, $C \subset \mathbb{R}^+$ (LASSO/Ridge) or $C \subset \mathbb{R}^{+2}$ (ElasticNet).
Set $w_1(\lambda) = 1$ for all $\lambda \in C$.
for $i = 1, 2, \ldots, T$ **do**
  $W_i := \int_C w_i(\lambda) d\lambda$.
  Sample λ with probability $p_t(\lambda) = \frac{w_i(\lambda)}{W_i}$, output as λ_i.
  Compute average loss function $l_i(\lambda) = \frac{1}{|y^{(i)}|} l(\hat{\beta}_{\lambda,f}, (X^{(i)}, y^{(i)}))$.
  For each $\lambda \in C$, update weights $w_{i+1}(\lambda) = e^{\zeta(1-l_i(\lambda))} w_i(\lambda)$.
end for

Under these assumptions, we can show that it is possible to learn the ElasticNet parameters when the problem instances arrive online, with $\tilde{O}(\sqrt{T})$ expected regret. The online learning algorithm (Algorithm 1) that achieves this regret is a continuous variant of the classic Exponential Weights algorithm [5]. It samples points in the domain with probability inversely propotional to the exponentiated loss. Our key contribution is to show that the loss sequence is dispersed (Definition 3) under the above assumptions. This involves establishing additional structure for the problem, specifically about the location of boundary functions in the piecewise structure from Lemma 2.1. This stronger characterization coupled with results from [7] on dispersion of algebraic discontinuities completes the proof.

Theorem 3.3. *Suppose Assumptions 1 and 2 hold. Let $l_1, \ldots, l_T : (0, \lambda_{\max})^2 \to \mathbb{R}_{\geq 0}$ denote an independent sequence of losses (e.g. fresh randomness is used to generate the validation set features in each round) as a function of the ElasticNet regularization parameter $\lambda = (\lambda_1, \lambda_2)$, $l_i(\lambda) = l_r(\hat{\beta}^{(X^{(i)}, y^{(i)})}_{\lambda, f_{EN}}, (X^{(i)}_{val}, y^{(i)}_{val}))$. The sequence of functions is $\frac{1}{2}$-dispersed, and there is an online algorithm with $\tilde{O}(\sqrt{T})$ expected regret. The result also holds for loss functions adjusted by information criteria AIC and BIC.*

Proof Sketch. We start with the $(\mathcal{F}, \mathcal{G}, p3^p)$-piecewise decomposable structure for the dual class function $\mathcal{H}^*_{\text{EN}}$ from Lemma 2.1. Observe that the rational piece functions in $\mathcal{F}$ do not introduce any new discontinuities since the denominator polynomials do not have positive roots. For each of two types of boundary functions in $\mathcal{G}$ (corresponding to leaving/entering the equicorrelation set) we show that the discontinuities between any pair of points λ, λ' lie along the roots of polynomials with non-leading coefficients bounded and smoothly distributed (bounded joint density). This allows us to use results from [7] to establish dispersion, and therefore online learnability. □

We remark that the above result holds for arbitrary training features and validation sets in the problem sequence (previous section needed them to be drawn from the same distribution). Also the parameters need to be selected online, and cannot be changed for already seen instances. This setting captures interesting practical settings, where the set of features (including feature dimensions) and the relevant training set (including training set size) may change over the online sequence. It is not clear how usual model selection techniques like cross-validation may be adapted to these challenging settings.

4 Extension to Regularized Least Squares Classification

Regression techniques can also be used to train binary classifiers by using an appropriate threshold on top of the regression estimate. The use of thresholds to make discrete classifications adds discontinuities to the empirical loss function. Thus, in general, the classification setting is strictly more challenging as it already includes the piecewise structure in the regression loss. We provide statistical and online learning guarantees for Ridge and LASSO (these can be simplified to give guarantees in the regression setting). For the ElasticNet we present the extensions needed to the arguments from the previous sections to obtain results in the classification setting.

More formally, we will restrict y to $\{0, 1\}^m$. The estimator $\hat{\beta}_{\lambda,f}$ is obtained as before, and the prediction on test instance x may be obtained by taking the sign of a thresholded regression es-

timate, $\text{sign}(\langle x, \hat{\beta}_{\lambda,f}\rangle - \frac{1}{2})$. For both regression and classification, the estimator $\hat{\beta}_{\lambda,f}$ depends on the regularization parameter λ. The average squared loss over the dataset (X, y) is given by $l_c(\hat{\beta}_{\lambda,f}, (X, y)) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \text{sign}(\langle x_i, \hat{\beta}_{\lambda,f}\rangle - \frac{1}{2}))^2$. (Proofs from this section are in Appendix D.)

4.1 Distributional setting

The problem setting is the same as in Section 3.1, except that the labels y are binary and we use threshold for prediction. We bound the pseudo-dimension for classification loss on these problem instances, which imply that only polynomially many problem samples to generalize well over the problem distribution $\mathcal{D}$. For Ridge and LASSO we upper bound the number of discontinuities of the piecewise constant classification loss by determining the values of λ where any prediction changes.

Theorem 4.1. *Let $\mathcal{H}^c_{Ridge}$, $\mathcal{H}^c_{LASSO}$ and $\mathcal{H}^c_{EN}$ denote the set of loss functions for classification problems with at most m examples and p features.*

(i) $\text{PDIM}(\mathcal{H}^c_{Ridge}) = O(\log mp)$
(ii) $\text{PDIM}(\mathcal{H}^c_{LASSO}) = O(p \log m)$. *Further, in the overparameterized regime ($p \gg m$), we have* $\text{PDIM}(\mathcal{H}^c_{LASSO}) = O(m \log \frac{p}{m})$.
(iii) $\text{PDIM}(\mathcal{H}^c_{EN}) = O(p^3 + p^2 \log m)$.

The key difference with the bound for the regression loss in Theorem 3.1 is the additional $O(p^2 \log m)$ term which corresponds to discontinuities induced by the thresholding in the regression based classifiers. We can establish a structure similar to Lemma 2.1 in this case (Lemma D.1).

4.2 Online setting

As in Section 3.2, we can define an online learning setting for classification. Note that the smoothness of the predicted variable is not meaningful here, since y is a binary vector. Instead we will assume that the validation examples have smooth feature values. Intuitively this means that small perturbations to the feature values does not meaningfully change the problem.

Assumption 3 (Smooth validation features)**.** *The feature values $(X^{(i)}_{val})_{jk}$ in the validation examples are drawn from a joint κ-bounded distribution.*

Under the assumption, we show that we can learn the regularization parameters online, for each of Ridge, LASSO and ElasticNet estimators. The proofs are straightforward extensions of the structural results developed in the previous sections, with minor technical changes to use the above validation set feature smoothness instead of Assumption 2, and are deferred to the appendix.

Theorem 4.2. *Suppose Assumptions 1 and 3 hold. Let $l_1, \ldots, l_T : \mathbb{R} \to \mathbb{R}$ denote an independent sequence of losses as a function of the Tikhonov (ridge) regularization parameter λ, $l_i(\lambda) = l_c(\hat{\beta}_{\lambda,f}, (X^{(i)}, y^{(i)}))$. The sequence of functions is $\frac{1}{2}$-dispersed, and there is an online algorithm with $\tilde{O}(\sqrt{T})$ expected regret, if f is given by*

(i) $f = f_1$,
(ii) $f = f_2$, *or*
(iii) $f = f_{EN}$ *(The result also holds for loss functions adjusted by information criteria AIC or BIC).*

5 Conclusions and Future Work

We obtain a novel structural result for the ElasticNet loss as a function of the tuning parameters. Our characterization is useful in giving upper bounds for the sample complexity of learning the parameters from multiple problem instances from the same problem domain. Efficient algorithms are immediate from our results for Ridge and LASSO. For the ElasticNet we show generalization and online regret guarantees, but efficient implementation of the algorithms is an interesting question for further work. Also we show general learning-theoretic guarantees, i.e. without any significant restrictions on the data-generating distribution, in learning from multiple problems. The problems may be drawn i.i.d. from an arbitrary *problem* distribution, or even arrive in an online sequence but with some smoothness properties. It is unclear if such guarantees may be given for tuning parameters for the more standard setting of tuning a single training, set generated by i.i.d. draws from an *example* distribution.

References

[1] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[2] Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*, volume 9. cambridge university press Cambridge, 1999.

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

[4] Maria-Florina Balcan and Dravyansh Sharma. Data driven semi-supervised learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[5] Maria-Florina Balcan, Travis Dick, and Ellen Vitercik. Dispersion for data-driven algorithm design, online learning, and private optimization. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 603–614. IEEE, 2018.

[6] Maria-Florina Balcan, Travis Dick, and Manuel Lang. Learning to link. In *International Conference on Learning Representations*, 2019.

[7] Maria-Florina Balcan, Travis Dick, and Wesley Pegden. Semi-bandit optimization in the dispersed setting. In *Conference on Uncertainty in Artificial Intelligence*, pages 909–918. PMLR, 2020.

[8] Maria-Florina Balcan, Travis Dick, and Dravyansh Sharma. Learning piecewise Lipschitz functions in changing environments. In *International Conference on Artificial Intelligence and Statistics*, pages 3567–3577. PMLR, 2020.

[9] 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.

[10] Maria-Florina F Balcan, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of automated mechanism design. *Advances in Neural Information Processing Systems*, 29, 2016.

[11] Maria-Florina F Balcan, Siddharth Prasad, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of tree search configuration: Cutting planes and beyond. *Advances in Neural Information Processing Systems*, 34, 2021.

[12] JM Chambers and TJ Hastie. Linear models. Chapter 4 of statistical models in S. *Wadsworth & Brooks/Cole*, 1992, 1992.

[13] Michael Chichignoud, Johannes Lederer, and Martin J 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.

[14] Thomas M Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, (3):326–334, 1965.

[15] Amit Dhurandhar and Marek Petrik. Efficient and accurate methods for updating generalized linear models with multiple feature additions. *The Journal of Machine Learning Research*, 15 (1):2607–2627, 2014.

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

[17] Richard M Dudley. The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967.

[18] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[19] Jianqing Fan and Jinchi Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101, 2010.

[20] Manuel Fernández-Delgado, Manisha Sanjay Sirsat, Eva Cernadas, Sadi Alawadi, Senén Barro, and Manuel Febrero-Bande. An extensive experimental survey of regression methods. *Neural Networks*, 111:11–34, 2019.

[21] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[22] Jean-Jacques Fuchs. Recovery of exact sparse representations in the presence of bounded noise. *IEEE Transactions on Information Theory*, 51(10):3601–3608, 2005.

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

[24] Rishi Gupta and Tim Roughgarden. A pac approach to application-specific algorithm selection. *SIAM Journal on Computing*, 46(3):992–1017, 2017.

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

[26] Arthur E Hoerl and Robert W Kennard. Ridge regression: applications to nonorthogonal problems. *Technometrics*, 12(1):69–82, 1970.

[27] 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.

[28] Johannes Lederer and Christian Müller. Don't fall for tuning parameters: tuning-free variable selection in high dimensions with the TREX. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[29] Qing Li and Nan Lin. The bayesian elastic net. *Bayesian analysis*, 5(1):151–170, 2010.

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

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[32] David Pollard. *Convergence of stochastic processes*. Springer Science & Business Media, 2012.

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

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

[35] Ryan J Tibshirani. The lasso problem and uniqueness. *Electronic Journal of statistics*, 7: 1456–1490, 2013.

[36] Andrey N Tikonov and Vasily Y Arsenin. Solutions of ill-posed problems. *New York: Winston*, 1977.

[37] Tong Zhang. Some sharp performance bounds for least squares regression with l1 regularization. *The Annals of Statistics*, 37(5A):2109–2144, 2009.

[38] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[39] Peng Zhao and Bin Yu. Stagewise lasso. *The Journal of Machine Learning Research*, 8: 2701–2726, 2007.

[40] 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.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We emphasize we are in a multiple instance setting instead of the standard setting.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]