Conformalization of Sparse Generalized Linear Models

Anonymous Author
Anonymous Institution

Abstract

Given a sequence of observable variables $(x_1, y_1), \ldots, (x_n, y_n)$, the conformal prediction method estimates a confidence set for y_{n+1} given x_{n+1} that is valid for any finite sample size by merely assuming that the distribution is permutation invariant. Although attractive, computing such a set turns out to be infeasible in most regression problems. Indeed, in these cases, the unknown variable y_{n+1} can take an infinite number of possible values, and generating conformal sets requires retraining a predictive model for each of them. In this paper, we focus on a sparse model where only a subset of variables is used for prediction, and we leverage numerical continuation techniques to efficiently approximate the solution path. The key property we exploit is that the set of selected variables is invariant under a small perturbation of the input data. Therefore, it is sufficient to enumerate and refit the model only at the change points of the set of active features and smoothly interpolate the rest of the solution via a predictor-corrector mechanism. We show how our path-following algorithm accurately approximates conformal prediction sets and illustrate its performance using synthetic and real data examples.

1 Introduction

Modern statistical learning algorithms achieve remarkable performance in predicting an object based on its observed characteristics. By design, practitioners train these algorithms on data sets that are assumed to be sampled from the same distribution, and it is important to quantify the uncertainty of their predictions. More precisely, after observing a finite sequence of data

$$\mathcal{D}_n = (x_1, y_1), \ldots, (x_n, y_n) \ ,$$

it is interesting to analyze to what extent one can build a confidence set for the next observation y_{n+1} given x_{n+1}. A classical approach is to adjust a prediction model on the observed data $\mu_{\mathcal{D}_n}$ and consider an interval centered around the prediction of y_{n+1} when the fitted model receives x_{n+1} as new input. We calibrate the confidence interval to satisfy a $100(1-\alpha)\%$ confidence, following as

$$\{z : |z - \mu_{\mathcal{D}_n}(x_{n+1})| \leq Q_n(1-\alpha)\} \ , \qquad (1)$$

where $Q_n(1-\alpha)$ is the $(1-\alpha)$-quantile of the empirical cumulative distribution function of the fitted residuals

$$|y_i - \mu_{\mathcal{D}_n}(x_i)|$$

for indices i in $\{1, \ldots, n\}$. This method is usually valid in an asymptotic regime and under the assumption that the fitted model is close to the ground truth, which requires some additional stringent regularity on both the prediction model and the data distribution.

Alternatively, conformal prediction is a versatile and simple method introduced in (Vovk et al., 2005; Shafer and Vovk, 2008) that provides a finite sample and distribution free $100(1-\alpha)\%$ confidence region for the predicted object based on past observations. The main idea is to follow the construction of the confidence set in Equation (1) by taking into account y_{n+1}. Since the latter is not given in the observed dataset $\mathcal{D}_n$, one learn a predictive model $\mu_{\mathcal{D}_{n+1}(z)}$ on an augmented database

$$\mathcal{D}_{n+1}(z) = \mathcal{D}_n \cup (x_{n+1}, z)$$

where z replaces the unknown response y_{n+1}. We can therefore define a prediction loss for each observation and rank them. A candidate z will be considered conformal or typical if the rank of its loss is sufficiently small. The conformal prediction set will merely collect the most typical z's as confidence set for y_{n+1}. Therefore, the model does not privilege observed over unobserved data and is immune to overfitting. Thus, the conformal prediction set is obtained as

$$\left\{z : |z - \mu_{\mathcal{D}_{n+1}(z)}(x_{n+1})| \leq Q_{n+1,z}(1-\alpha)\right\} \ , \qquad (2)$$

where $Q_{n+1,z}(1-\alpha)$ is the $(1-\alpha)$-quantile of the empirical cumulative distribution function of the re-fitted residuals

$$|y_i(z) - \mu_{\mathcal{D}_{n+1}(z)}(x_i)|$$

Preliminary work. Under review by AISTATS 2023. Do not distribute.

for indices i in $\{1, \dots, n+1\}$ and $y(z) = (y_1, \dots, y_n, z)$. As long as the sequence $\{(x_i, y_i)\}_{i=1}^{n+1}$ is exchangeable *i.e.,* their joint probability distribution is invariant with respect to permutation of the data, as well as the predictive model. This method benefits from a strong coverage guarantee without any assumption on the distribution and is valid for any finite sample size n; see Section 3. The conformal prediction approach has been applied for designing uncertainty sets in active learning (Ho and Wechsler, 2008), anomaly detection (Laxhammar and Falkman, 2015; Bates et al., 2021), few-shot learning (Fisch et al., 2021), time series (Chernozhukov et al., 2018; Xu and Xie, 2021; Chernozhukov et al., 2021), or to infer the performance guarantee for statistical learning algorithms (Holland, 2020; Cella and Ryan, 2020). We refer to the extensive reviews in (Balasubramanian et al., 2014) for other applications to artificial intelligence.

Despite its attractive properties, the computation of conformal prediction sets requires fitting a model on a dataset where a set of candidates replaces the unknown quantity. The set of candidates is infinite in a regression setting where an object can take an uncountable possible value. Therefore, the computation of conformal prediction is generally infeasible without additional structural assumptions on the underlying model fit. Otherwise, the calculation costs remain high or impossible. Nevertheless, we can take advantage of the stepwise behavior of the typicalness function that maps each candidate to the rank of its prediction loss. If we carefully manage to list all its transition points or where this typicalness function changes, we can find exactly where it is above the prescribed confidence level.

Furthermore, we can avoid the central issue of the model fit by using the structural assumptions given by the setting of General Linear Models with ℓ_1 regularization. For estimators with a closed-form formula (*e.g.,* Ridge or Lasso), this setting makes it possible to draw the solutions curve *w.r.t.* the input candidate z. They are often pieces of linear functions that enable the exhaustive listing of the change points of the rank function; see (Nouretdinov et al., 2001) and (Lei, 2019).

In this paper, we generalize linear homotopy approaches from quadratic loss to a broader class of nonlinear loss functions using numerical continuation to efficiently trace a piecewise smooth solution path. Overall, we propose a homotopy drawing algorithm that keeps track of the weights' sparsity over possible candidates' space to solve for the primal optimal solution efficiently. To do so, we propose to estimate the primal optimal solution by linearizing the dual constraint. This yield a Conformal Prediction algorithm for sparse generalized linear model. Additionally, using numerical continuation using the patterns in the sparsity of the weights, we relinquish the expensive necessity of retraining the model many times from random initialization. Furthermore, we provide a primal prediction step that significantly reduces the number of iterations needed to obtain an approximation at high precision. We illustrate the performance of our algorithm as a homotopy drawer and a conformal set generator using asymmetric, robust, and lasso loss functions with ℓ_1 regularization.

Notation. For a nonzero integer n, we denote $[n]$ to be the set $\{1, \cdots, n\}$. The dataset of size n is denoted $\mathcal{D}_n = (x_i, y_i)_{i\in[n]}$, the row-wise feature matrix $X = [x_1, \cdots, x_{n+1}]^\top$, and $X_{[n]}$ is its restriction to the n first rows. We denote the smallest integer no less than a real value r as $\lceil r \rceil$. We denote by $Q_{n+1}(1-\alpha)$, the $(1-\alpha)$-quantile of a real valued sequence $(U_i)_{i\in[n+1]}$, defined as the variable $Q_{n+1}(1-\alpha) = U_{(\lceil (n+1)(1-\alpha) \rceil)}$, where $U_{(i)}$ are the i-th order statistics. For j in $[n+1]$, the rank of U_j among $U_1, \cdots, U_{n+1}$ is defined as

$$\mathrm{Rank}(U_j) = \sum_{i=1}^{n+1} \mathbb{1}_{U_i \leq U_j} \quad .$$

Computing a set of predictions requires evaluating the rank of each candidate and thus fitting a model on the augmented data. Without additional assumptions, it is nontrivial to access the whole function $z \mapsto \beta(z)$. Parsimonious regularization brings out a particular regularity in the structure of the solution. Indeed, depending on the regularization parameter λ, the optimal solution contains coordinates equal to zeros, thus eliminating several features judged useless in predicting the observations. The main remark is that the set of selected variables is stable by slight perturbation of the optimization problem data. Thus, two close candidates will have the same support and thus select the same variables. We will identify the active set change points and re-fit the model only at those points. The intermediate models will be obtained by simple interpolation as the solution function is smooth with a computable formulation. We detail this procedure in the next section.

2 Sparse Generalized Linear Models

By definition of the conformal prediction set Equation (2), one needs to consider an augmented dataset $\mathcal{D}_{n+1}(z)$ for any possible replacement of the target variable y_{n+1} by a real value z. This implies the computation of the path $z \mapsto \mu_{\mathcal{D}_{n+1}(z)}(x_{n+1})$ as well as the path of scores and quantiles. However, doing this, in general, is difficult. We focus on the Generalized Linear Model (GLM) regularized with an ℓ_1 norm that promotes sparsity of the model parameter. For a fixed $z \in \mathbb{R}$, the latter is defined as a solution to the following optimization problem

$$\beta(z) \in \underset{\beta\in\mathbb{R}^p}{\arg\max}\, f(y(z), X\beta) + \lambda \left\|\beta\right\|_1 \quad . \tag{3}$$

where the data fitting term $f(y(z), \hat{y})$ is a non negative loss function between the prediction $\hat{y}$ and the augmented vector

of labels

$$y(z) = (y_1, \cdots, y_n, z).$$

For example, we parameterize a linear prediction as

$$\hat{y}_i = x_i^\top \beta$$

and the empirical loss reads

$$L(y(z), \hat{y}) = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \ell(z, \hat{y}_{n+1}) \ .$$

There are many examples of cost functions in the literature. A popular example is the power norm regression, where $\ell(a, b) = |a - b|^q$. When $q = 2$, this corresponds to the classical linear regression. The cases where $q = [1, 2)$ are frequent in robust statistics where the case $q = 1$ is known as the least absolute deviation. The loss `logcosh` $\ell(a, b) = \gamma \log(\cosh(a - b)/\gamma)$ is a differentiable alternative to the ℓ_∞ norm (Chebychev approximation). One can also have the loss function `Linex` (Gruber, 2010; Chang and Hung, 2007) which provides an `asymmetric` loss function $\ell(a, b) = \exp(\gamma(a-b)) - \gamma(a-b) - 1$, for $\gamma \neq 0$.

2.1 Efficient Computation of the Solution Path

In this section, we introduce tools to compute the solutions to the problem when the input data changes. While difficult for nonlinear models, for linear models, we can rely on the classical approach of evaluating the first-order optimal conditions and making sure to maintain them until they are violated. These points will therefore be the kinks i.e., the points of non-differentiability of the function $z \mapsto \beta(z)$, and characterize the active set changes along the path. Such continuation techniques have been previously used when the objective function is differentiable (Allgower and Georg, 2012), (Hastie et al., 2004) for support vector machine, least-squared loss and more general loss functions regularized with the ℓ_1 norm in (Rosset and Zhu, 2007; Park and Hastie, 2007; Tibshirani, 2013; Mairal and Yu, 2012). However, the latter focus on the regularization path and plot the solution curve as the regularization parameter λ varies. In the setting we consider, we recall that it is the response vector that is parameterized as $y(z) = (y_1, \ldots, y_n, z)$ for a real value z; for which (Garrigues and Ghaoui, 2009; Lei, 2019) proposed a homotopy algorithm when the loss function is quadratic. We extend such algorithms to more nonlinear loss functions.

We start by characterizing the properties of the optimal solutions and show how they can be sequentially approximated.

Lemma 1. *Assume that for any real value z, the function $f_z : \mathbb{R}^n \mapsto \mathbb{R}$ is strictly convex and differentiable. A vector $\beta(z) \in \mathbb{R}^p$ is optimal for Equation* (3) *if and only if*

$$- X^T \nabla_2 f(y(z), \hat{y}(z)) = \lambda v(z) \ , \tag{4}$$

where $v(z)$ belongs to the subdifferential of the ℓ_1 norm at $\beta(z)$ i.e., *$\forall j \in \{1, \ldots, p\}$, we have*

$$v_j(z) \in \begin{cases} \{\operatorname{sign}(\beta_j(z))\} & \text{if } \beta_j(z) \neq 0 \ , \\ [-1, 1] & \text{if } \beta_j(z) = 0 \ . \end{cases} \tag{5}$$

We define our active set at a point z as

$$A(z) = \left\{ j \in [p] : \ |X_j^\top \nabla_2 f(y(z), \hat{y}(z))| = \lambda \right\} \ . \tag{6}$$

The active set contains at least all the indices of the optimal solution that are guaranteed to be nonzero. We will denote $A = A(z)$ if there is no ambiguity.

Proof. The Fermat rule reads

$$0 \in \{X^T \nabla_2 f(y(z), \hat{y}(z)) + \lambda \partial \left\| \cdot \right\|_1 (\beta(z))\} \ ,$$

from which Equation (4) follows. From Equation (5), we have

$$\beta_j(z) \neq 0 \Longrightarrow |X^T \nabla_2 f(y(z), \hat{y}(z))| = |\lambda v_j(z)| = \lambda \ .$$

Hence, by contrapositive, we have $|v_j(z)| \neq 1$ implies $\beta_j(z) \neq 0$. Since the element of the subdifferential of the ℓ_1 norm is smaller or equal to 1, we conclude that the j-th coordinate of any optimal solution is zero when j is not in the active set defined in Equation (6). Combined with Equation (4) we have

$$-X_A^T \nabla_2 f(y(z), \hat{y}(z)) = \lambda v_A(z)$$

which is equivalent to (also from Fermat's rule)

$$\beta_A(z) \in \underset{w \in \mathbb{R}^{|A|}}{\arg\min} \, f(y(z), X_A w) + \lambda \left\| w \right\|_1 \ .$$

Since f is strictly convex and X_A is full rank, the objective function restricted on the active set is strictly convex and then has a unique minimum. □

Lemma 1 ensures that the mapping $z \mapsto \beta(z)$ is unique and well defined. At every point z, the computation can be restricted to the active set $A(z)$ since its component is equal to zero on the complementary.

2.2 Computation of $\beta_A(z)$

For the Lasso Case *i.e.,* the least square loss function $f(\hat{y}) = \frac{1}{2} \left\| y - \hat{y} \right\|_2^2$, the optimality condition Equation (4) implies that

$$\beta_A(z) = (X_A^\top X_A)^{-1} (X_A^\top y(z) - \lambda v_A(z)) \ . \tag{7}$$

Hence, given $z' \neq z$ such that the corresponding signs of the solutions are equal, $\beta_A(z')$ can be explicitly obtained from $\beta_A(z)$ by linear interpolation. Unfortunately, this does

not hold for general loss. However, a local linearization of the function $\beta_A(z')$ can be an efficient approximation:

$$\beta_A(z') \approx \beta_A(z) + \frac{\partial \beta_A}{\partial z}(z) \times (z' - z) \ . \tag{8}$$

We can have yet to understand the term $\frac{\partial \beta_A}{\partial z}(z)$. To do so, we follow similar work from (Park and Hastie, 2007) where we set a function

$$H(y(z), \beta_A(z)) \coloneqq X_A^T \nabla_2 f(y(z), \hat{y}(z)) + \lambda v_A$$

Across all possible z, due to the Optimality Condition, $H(y(z), \beta_A(z)) = 0$, implying $\frac{\partial H}{\partial z} = 0$. However, by using the Implicit Function Theorem, we yield

$$\begin{aligned}
\frac{\partial H}{\partial z} &= \frac{\partial H}{\partial y}\frac{\partial y}{\partial z} + \frac{\partial H}{\partial \beta}\frac{\partial \beta}{\partial z} \\
\frac{\partial \beta}{\partial z} &= -\left(\frac{\partial H}{\partial \beta}\right)^{-1} \frac{\partial H}{\partial y}\frac{\partial y}{\partial z}
\end{aligned}$$

Therefore, we can finally estimate

$$\beta_A(z') \approx \beta_A(z) - \left(\frac{\partial H}{\partial \beta}\right)^{-1} \frac{\partial H}{\partial y}\frac{\partial y}{\partial z} \times (z' - z) \tag{9}$$

where

$$\begin{aligned}
\frac{\partial H}{\partial \beta} &= X_A^\top \nabla_{2,2} f(\cdot, \hat{y}) X_A \\
\frac{\partial H}{\partial y} &= X_A^\top \nabla_{2,1} f(\cdot, \hat{y}) \\
\frac{\partial y}{\partial z} &= (0, \ldots, 0, 1) \ .
\end{aligned}$$

Notably, we get an equation for $\beta_A(z')$ where we can efficiently compute it in terms of $y(z')$. This linearization methodology will also prove helpful in determining an efficient method for computing the dual variable $\nabla_2 f(y(z), \hat{y}(z'))$.

2.3 Active Set Updates

Now we have to track sequentially the changes that may occur in the active sets along the path. There are only two possible cases for a change in the active set.

1. A variable j leaves the active set meaning $\beta_j(z) \neq 0$ and $\beta_j(z') = 0$.
2. A variable j joins the active set meaning $\beta_j(z) = 0$ and $\beta_j(z') \neq 0$.

Therefore, we aim to find the first z' at which any of these two events occur for any variable j.

Leaving the active set At the point where the first case occurs, we know that by Equation (9), we have a closed form approximation of $\beta_A(z')$ given $\beta_A(z)$. Therefore, for a dimension $j \in A$, we have an explicit formula for $\beta_j(z')$ in terms of z', which we can compute efficiently. Therefore, from Equation (9), j leaving the active set occurs at $\beta_j(z') = 0$ implies a kink occurs at z' when when $\mathcal{O}_j(z') \approx 0$ where

$$\mathcal{O}_j(z') := \left[\beta_A(z) - \left(\frac{\partial H}{\partial \beta}\right)^{-1} \frac{\partial H}{\partial y}\frac{\partial y}{\partial z} \times (z' - z)\right]_j \ ,$$

is the R.H.S. of Equation (9); which is easily solvable in closed-form.

Joining the active set At the point where the second case occurs, we know from Equation (4) that for any variable j that leaves the active set

$$|X_j^\top \nabla_2 f(y(z'), X_{A'}\beta_{A'}(z'))| = \lambda$$

where $A' = A \cup \{j\}$. However, given that we are searching for a point z' where the active sets shifts from A to A', at point z', $\beta_j(z')$ is roughly 0 since it is the first point where $\beta_j(z')$ becomes nonzero. Therefore, given this information, the prediction $X_j\beta_j(z') = 0$ where z' is a kink. Using this idea, we can provide the equivalence

$$X_{A'}\beta_{A'}(z') \approx X_A\beta_A(z') = \hat{y}(z') \ .$$

Using this approximation yields

$$|X_j^\top \nabla_2 f(y(z'), X_{A'}\beta_{A'}(z'))| \approx |X_j^\top \nabla_2 f(y(z'), \hat{y}(z'))|.$$

This equivalence is extremely useful as we know how to approximate $\beta_A(z')$ efficiently from Equation (10). Therefore, dimension j must join the active set at approximately the point z' where z' satisfies $|X_j^\top \nabla_2 f(y(z'), \hat{y}(z')| = \lambda$ From , we can approximate $\beta_A(z')$ efficiently. Therefore, setting the function

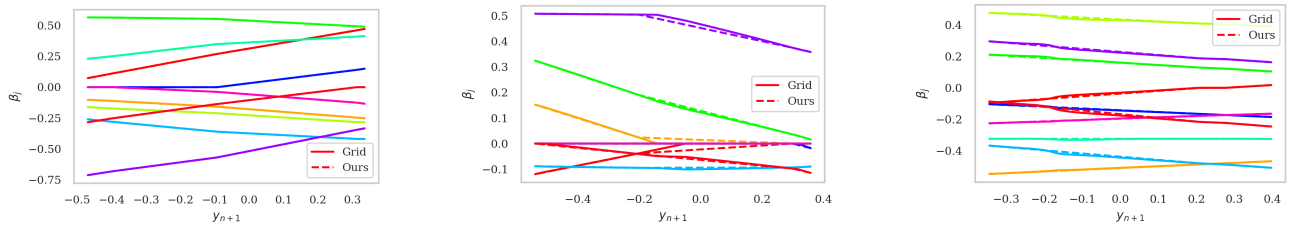
$$\mathcal{I}_j(z') \coloneqq |X_j \nabla_2 f(y(z'), \hat{y}(z'))| - \lambda \tag{10}$$

We know that whenever $\mathcal{I}_j(z')$ is equal to 0, variable j must join the active set. Again, we can use a root-finding function to efficiently find the roots of the function to find $\mathcal{I}_j(z')$ where the kink may lie.

2.4 Approximation of $\nabla_2 f(y(z'), \hat{y}(z'))$

While $\mathcal{O}_j(z')$ is linear in z', giving way for a explicit solution for z', this property does not hold for $\mathcal{I}_j(z')$. To achieve such a form, we need to further linearize $\nabla_2 f(y(z), \hat{y}(z'))$. For simplicity, we denote

$$g(z) \coloneqq f(y(z), \hat{y}(z))$$



(a) Lasso (b) Asymetric (c) Robust

Figure 1: We generate several example homotopies over Lasso, Asymmetric, and Robust loss functions. We plot using our algorithm and a discretized search space algorithm, where the space of potential z_{n+1} values is split into several points, and we solve for β using Proximal Gradient Descent at each point.

Using simple linearization yields,

$$\nabla_2 g(z') \approx \nabla_2 g(z) + \nabla_{2,1} g(z)^\top (y(z') - y(z)) \quad (10)$$
$$+ \nabla_{2,2} g(z)^\top (\hat{y}(z') - \hat{y}(z)) \ ,$$

from Equation (9). In Equation (10), only $\hat{y}(z')$ and $y(z')$ are parameterized by z'. By Equation (9), $\hat{y}(z')$ has an explicit formula linear in z' and $y(z')$ is already linear in z'. Therefore, the equation Equation (10) is linear in z'. Using this for the computation of $\mathcal{I}_j(z')$ yields an explicit equation in terms of z', meaning we can directly solve for which z' achieves $\mathcal{I}_j(z') = 0$.

2.5 Solution Updates

Our kink-finder finds kink z_{t+1} given kink z_t. However, to use our kink-finder again, we need to find $\beta(z_{t+1})$. We do so using Primal Prediction and Primal Correction.

Primal Predictor Given $\beta(z_t)$, the goal of warm starting is to provide an accurate estimate $\hat{\beta}(z_{t+1})$ of $\beta(z_{t+1})$. Giving Equation (9), we can provide an accurate prediction of $\beta(z_{t+1})$ efficiently.

Primal Correction The solution obtained in the Warm Start often has a reasonably small approximation error. For example, in the case of the Lasso, this warm start is exact and primal correction is not needed. However, it generally is an imprecise estimate of the exact solution. To overcome this, we use an additional primal corrector step using an iterative solver, such as proximal gradient descent initialized with the predictor output.

Combining all of these ideas, we can efficiently draw the homotopy. We first initialize the solution at $z_0 = \max(y)$, solving for $\beta(z_0)$, find the next kink z_t using functions $\mathcal{I}$ and $\mathcal{O}$, and repeat till the algorithm reaches a z such that $z < \min(y)$. We detail the full algorithm in Algorithm 1.

3 Conformal Algorithm for Sparse GLM

Given a homotopy for specific data and loss function, computing the Conformal Prediction only relies on the simple calculation from the homotopy. The primary tool is the rank of one variable among an exchangeable and identically distributed sequence following a (sub)-uniform distribution (Bröcker and Kantz, 2011).

This idea of rank helps construct distribution-free confidence intervals. We can estimate the conformity of a given candidate z by calculating its loss $|z - \hat{y}_{n+1}(z)|$ and compute its rank relevant to the losses of the other variables. The candidate will be considered conformal or typical if the rank of its loss is sufficiently small. Let us define the conformity measure for $\mathcal{D}_{n+1}(z)$ as

$$E_i(z) = |y_i - \hat{y}_i(z)|, \quad \forall i \in [n] \ , \quad (11)$$
$$E_{n+1}(z) = |z - \hat{y}_{n+1}(z)| \ . \quad (12)$$

The main idea for constructing a conformal confidence set is to consider the typicalness/conformity of a candidate point z measured as

$$\pi(z) := 1 - \frac{1}{n+1} \mathrm{Rank}(E_{n+1}(z)) \ . \quad (13)$$

The conformal prediction set will merely collect the most typical z as a confidence set for y_{n+1} *i.e.,* gathers all the real values z such that $\pi(z) \geq \alpha$, if and only if, the score $E_{n+1}(z)$ is ranked no higher than $\lceil (n+1)(1-\alpha) \rceil$, among the sequence $\{E_i(z)\}_{i \in [n+1]}$ *i.e.,*

$$\Gamma^{(\alpha)}(x_{n+1}) := \{z \in \mathbb{R} : \ \pi(z) \geq \alpha\}$$
$$= \{z \in \mathbb{R} : \ E_{n+1}(z) \leq Q_{n+1,z}(1-\alpha)\} \ ,$$

which is exactly the conformal set defined in Equation (2).

To compute a conformal set, we only need to calculate the piecewise constant function $z \mapsto \pi(z)$. Fortunately, our framework directly yields light on the computation of this value over the range space.

Algorithm 1 Full Homotopy Generation

Input Data: $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, x_{n+1}, $\lambda > 0$
$t = 0$, $z_0 = \max y$

$$y(z_0) = (y_1, \ldots, y_n, z_0)$$
$$\beta(z_0) \coloneqq \underset{\beta \in \mathbb{R}^p}{\arg\min}\, f(y(z_0), X\beta) + \lambda \|\beta\|_1$$
$$A(z_0) \coloneqq \left\{ j \in [p] : \; |X_j^\top \nabla_2 f(y(z_0), \hat{y}(z_0))| = \lambda \right\}$$

while $z_t > \min y$ **do**
$\hat{z}_{\mathcal{I}} \coloneqq \max(z')$ s.t. $\exists j \in A^c(z_t)$ where $\mathcal{I}_j(z') = 0$
$\hat{z}_{\mathcal{O}} \coloneqq \max(z')$ s.t. $\exists j \in A(z_t)$ where $\mathcal{O}_j(z') = 0$
if $\hat{z}_{\mathcal{I}} > \hat{z}_{\mathcal{O}}$ **then**
$z_{t+1} \coloneqq \hat{z}_{\mathcal{I}}$
$A(z_{t+1}) \coloneqq A(z_t) \cup \{j_{\mathcal{I}}\}$
else
$z_{t+1} \coloneqq \hat{z}_{\mathcal{O}}$
$A(z_{t+1}) \coloneqq A(z_t) \setminus \{j_{\mathcal{O}}\}$
end if
Primal Predictor

$$\hat{\beta}(z_{t+1}) \coloneqq \beta(z_t) - \left(\frac{\partial H}{\partial \beta} \right)^{-1} \frac{\partial H}{\partial y} \frac{\partial y}{\partial z} \times (z_{t+1} - z_t)$$

Primal Corrector warm started with $\hat{\beta}(z_{t+1})$

$$\beta(z_{t+1}) \coloneqq \underset{\beta \in \mathbb{R}^p}{\arg\min}\, f(y(z_{t+1}), X\beta) + \lambda \|\beta\|_1$$

$t \coloneqq t + 1$
end while
return $\beta(z), z$ for all time steps

4 Conformal Prediction using the homotopy

Access to the homotopy, as well as the kinks, yields an efficient methodology for calculating the function $\Gamma^{(\alpha)}(x_{n+1}))$ over the range space. We do so by tracking where changes in this function occur. Naturally, changes in the rank function only occur when the loss of one example surpasses or goes below that of the loss of the last example. Formally, this can be seen as when

$$|y_i(z) - \hat{y}_i(z)| = |y_{n+1}(z) - \hat{y}_{n+1}(z))|. \qquad (14)$$

At first, it is not immediately clear how to find points z that satisfy the above equation. To efficiently find such points, we will look between the two kinks. For a point z between two kinks, we can efficiently calculate $\hat{y}(z)$. Indeed, given a point z is between two kinks z_t and z_{t+1} with an active set A, we can use Equation (9) to estimate the quantity $y(z) - \hat{y}(z)$ as

$$y(z) - \hat{y}(z_t) - \left(\frac{\partial H}{\partial \beta} \right)^{-1} \frac{\partial H}{\partial y} \frac{\partial y}{\partial z} \times (z - z_t) \;\; ,$$

where $\beta_A(z_t)$ is stored from the primal corrector step at the kink z_t. Given that this value is linear in z, we can form a closed-form explicit equation for what z solves Equation (14) efficiently. Therefore, between every sequential pair of kinks, we can look for where the $\pi(z)$ value changes. To find the conformal set, we track the changes $\pi(z)$ and recompute it along each root of Equation (14), yielding an efficient methodology to compute $\pi(z)$, and, therefore, the conformal set along the space of possible y_{n+1} values.

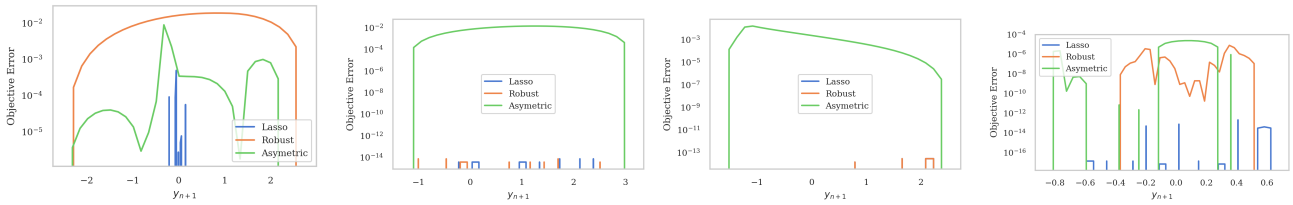
5 Numerical Experiments

Our main claim is twofold. Our method efficiently and accurately generates the homotopy over general loss functions. Our method efficiently and accurately generates conformal sets over general loss functions. We demonstrate these two claims over different datasets and loss functions.

Datasets We generate 4 different sets of data to test our claims. The first three datasets are real datasets of Diabetes data available from (Pedregosa et al., 2011), the multivariate regression dataset denoted Friedman1 from (Friedman, 1991), and the multivariate dataset denoted Friedman2 from (Breiman, 1996), demonstrating our algorithms efficacy on real data. We synthetically generate regression problems, generating random data and labels with a constant number of examples $n = 100$ and dimension $p = 100$. These datasets represent a reasonable range of regression problems, usable for our experiments.

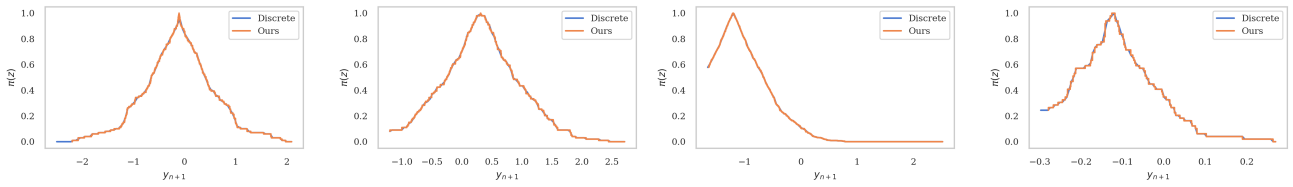
Homotopy Experiments

In order to test our algorithm in terms of homotopy generation, we measure both the accuracy and efficacy of our algorithm against different baselines across different loss functions. Specifically, we measure the negative logarithm of the gap between primal values of the calculated β values and a ground truth baseline. We measure this gap across many possible z values and take the average. The ground truth baseline is a Grid-based homotopy, where an iterative algorithm is applied with tolerance less than 10^{-10} at many discretized points z. Given that we apply the negative logarithm to the primal gap, the larger the value reported, the smaller the true error term and the better the algorithm's performance. Moreover, we report the time taken in seconds required to form the homotopy. As a baseline for our algorithm, we measure our algorithm against the Approximate Homotopy method from (Ndiaye and Takeuchi, 2019), a standard homotopy generation algorithm. Our experiments cover the Lasso, Robust, and Asymmetric functions across all the datasets from 5.



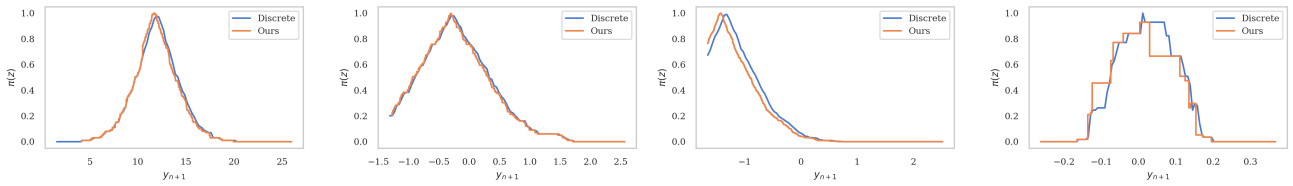
(a) Friedman1 (b) Friedman 2 (c) Diabetes (d) Synthetic

Figure 2: We demonstrate the objective error of our achieved homotopy over the space of possible y_{n+1} on all 4 datasets and loss functions.



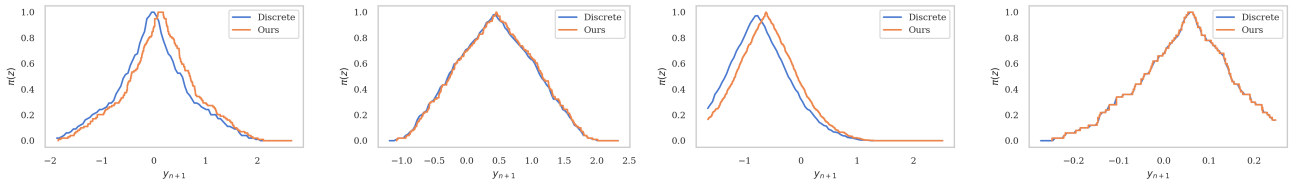
(a) Lasso on Friedman1 (b) Lasso on Friedman 2 (c) Lasso on Diabetes (d) Lasso on Synthetic

Figure 3: The $\pi(z)$ function as generated by a ground truth discretized searching algorithm and by our homotopy drawing algorithm for the Lasso loss function over all 4 datasets.



(a) Robust on Friedman1 (b) Robust on Friedman 2 (c) Robust on Diabetes (d) Robust on Synthetic

Figure 4: The $\pi(z)$ function as generated by a ground truth discretized searching algorithm and by our homotopy drawing algorithm for the Robust loss function over all 4 datasets.



(a) Asymmetric on Friedman1 (b) Asymmetric on Friedman 2 (c) Asymmetric Diabetes (d) Asymmetric on Synthetic

Figure 5: The $\pi(z)$ function as generated by a ground truth discretized searching algorithm and by our homotopy drawing algorithm for the Asymmetric loss function over all 4 datasets.

We report our results in Tables Section 5 and Section 5. As evident, we see a significant decrease in time used over Approximate Homotopy for all applications of the Lasso Loss with a significant increase in accuracy. Furthermore, we report similar primal gaps for both ours and the approximate homotopy algorithms on Robust and Asymmetric losses. However, we achieve significant time improvements. Notably, on the Diabetes dataset for Asymmetric losses and the Synthetic dataset for both Asymmetric and Robust losses, we report an almost 50% reduction in the time taken to achieve a similar error. Overall, across all loss types and datasets, we either achieve similar or better errors with the same or less time relative to the standard Approximate Homotopy, demonstrating the capability of our algorithm to efficiently and accurately generate the homotopy.

To further demonstrate the accuracy of our algorithm not only over an average of all possible candidates z_{n+1} but over the entire space, we plot the primal gap over the space of z_{n+1} for all 3 loss functions and 4 datasets. We report the figures in 2. Notably, we see that on 2d, we achieve all losses

better than 10^{-4}. On the other figures, we see all objective errors are bounded by 10^{-2}. Additionally, our application of Lasso and Robust over all datasets achieves near 0 objective error over the entire pass. This demonstrates our ability to accurately form the homotopy over all values of y_{n+1}.

Table 1: Average Time of Homotopy

	Dataset			
	Synthetic	**Friedman1**	**Diabetes**	**Freidman2**
Our Lasso	**1.706**	**1.945**	**1.0785**	**0.681**
Approx. Lasso	5.176	43.823	70.813	14.055
Our Robust	**27.156**	1.069	2.411	0.701
Approx. Robust	62.894	1.009	2.734	0.618
Our Asymmetric	**9.270**	3.147	**27.349**	2.454
Approx. Asymmetric	18.963	2.699	54.149	3.342

Table 2: Average Negative Logarithm of Primal Gap of Homotopy

	Dataset			
	Synthetic	**Friedman1**	**Diabetes**	**Freidman2**
Our Lasso	**12.498**	**15.844**	**16.001**	**15.241**
Approx. Lasso	6.597	6.469	7.554	6.702
Our Robust	5.137	2.317	3.819	2.778
Approx. Robust	5.990	3.561	3.712	4.434
Our Asymmetric	7.879	3.633	3.814	3.058
Approx. Asymmetric	6.939	3.208	4.032	2.795

Conformal Prediction Experiments

It is a natural question whether this improvement in the generation of the homotopy function yields a strong conformal set generation algorithm. The guarantees of coverage and length are immediate from the accuracy of the $\pi(z)$ function. Therefore, we need only demonstrate the accuracy of our algorithm in generating this function. Over all 4 datasets and 3 loss functions, we draw the $\pi(z)$ function using our algorithm. To form a comparison, we use the discretized searching algorithm that splits the space of possible z_{n+1} values into several discrete values and iteratively generates the homotopy at each of those points. This algorithm serves as a ground truth to which we compare our $\pi(z)$ function

We report the figures in 5. As is evident over all loss functions and datasets, our estimated $\pi(z)$ roughly traces the true $\pi(z)$ generated by the discretized searching algorithm. While on certain examples, notably Figures 4d, 5a, and 5c, the trace is not as accurate as the others. However, the error is within a reasonable range to achieve the desired coverage and length guarantees. We demonstrate that our homotopy drawing algorithm yields an efficient and accurate methodology for generating conformal sets for general loss functions as tested on several datasets.

6 Discussion

Our results demonstrate that we can efficiently and accurately draw the homotopy of the typicalness function of a model over several loss functions via exploiting the sparsity structure of the Linear Models with ℓ_1 regularization enables. Furthermore, we achieve explicit closed-form equations to model the behavior of this homotopy. This work can be seen as extending the regularization path algorithm of (Park and Hastie, 2007) to the typicalness function, and the Lasso conformalization algorithm of (Lei, 2019) to general loss functions.

Our contributions represent significant advancements over the existing conformalization methods for general loss functions. Previous results mainly focus on quadratic loss functions or ignore the structure of the regularization altogether. Our framework, instead, captures this information and uses it to improve the accuracy of our final results.

Several avenues for extending our research remain interesting. While we explored linearizing the computation of the optimal primal value and the gradient of the optimal primal value, it warrants interest whether using spline interpolation instead of linear interpolation may yield improved accuracy for different loss functions. Additionally, one may wish to achieve smoothing at the kinks so that our algorithm is not as sensitive to the primal corrector's results. Furthermore, we would like to expand our work to non-convex settings such as deep learning in future works.

References

Allgower, E. L. and Georg, K. (2012). *Numerical continuation methods: an introduction*. Springer Science & Business Media.

Balasubramanian, V., Ho, S.-S., and Vovk, V. (2014). *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Elsevier.

Bates, S., Candès, E., Lei, L., Romano, Y., and Sesia, M. (2021). Testing for outliers with conformal p-values. *arXiv preprint arXiv:2104.08279*.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Bröcker, J. and Kantz, H. (2011). The concept of exchangeability in ensemble forecasting. *Nonlinear Processes in Geophysics*.

Cella, L. and Ryan, R. (2020). Valid distribution-free inferential models for prediction. *arXiv preprint arXiv:2001.09225*.

Chang, Y.-C. and Hung, W.-L. (2007). Linex loss functions with applications to determining the optimum process parameters. *Quality & Quantity*.

Chernozhukov, V., Wüthrich, K., and Zhu, Y. (2018). Exact and robust conformal inference methods for predictive

machine learning with dependent data. *Conference On Learning Theory*.

Chernozhukov, V., Wüthrich, K., and Zhu, Y. (2021). An exact and robust conformal inference method for counterfactual and synthetic controls. *Journal of the American Statistical Association*.

Fisch, A., Schuster, T., Jaakkola, T., and Barzilay, R. (2021). Few-shot conformal prediction with auxiliary tasks. *ICML*.

Friedman, J. H. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1 – 67.

Garrigues, P. and Ghaoui, L. E. (2009). An homotopy algorithm for the lasso with online observations. In *Advances in neural information processing systems*, pages 489–496.

Gruber, M. (2010). *Regression estimators: A comparative study*. JHU Press.

Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. (2004). The entire regularization path for the support vector machine. *J. Mach. Learn. Res*.

Ho, S.-S. and Wechsler, H. (2008). Query by transduction. *IEEE transactions on pattern analysis and machine intelligence*.

Holland, M. J. (2020). Making learning more transparent using conformalized performance prediction. *arXiv preprint arXiv:2007.04486*.

Laxhammar, R. and Falkman, G. (2015). Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories. *Annals of Mathematics and Artificial Intelligence*.

Lei, J. (2019). Fast exact conformalization of lasso using piecewise linear homotopy. *Biometrika*.

Mairal, J. and Yu, B. (2012). Complexity analysis of the lasso regularization path. *ICML*.

Ndiaye, E. and Takeuchi, I. (2019). Computing full conformal prediction set with approximate homotopy. In *Advances in Neural Information Processing Systems*.

Nouretdinov, I., Melluish, T., and Vovk, V. (2001). Ridge regression confidence machine. *ICML*.

Park, M. Y. and Hastie, T. (2007). L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*.

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

Rosset, S. and Zhu, J. (2007). Piecewise linear regularized solution paths. *The Annals of Statistics*.

Shafer, G. and Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research*.

Tibshirani, R. J. (2013). The lasso problem and uniqueness. *Electronic Journal of Statistics*.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic learning in a random world*. Springer.

Xu, C. and Xie, Y. (2021). Conformal prediction interval for dynamic time-series. *ICML*.