Conformalization of Sparse Generalized Linear Models

Etash Kumar Guha [1] **Eugene Ndiaye** [2] **Xiaoming Huo** [3]

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 joint distribution of the data is permutation invariant. Although attractive, computing such a set is computationally infeasible in most regression problems. Indeed, in these cases, the unknown variable y_{n+1} can take an infinite number of possible candidate values, and generating conformal sets requires retraining a predictive model for each candidate. In this paper, we focus on a sparse linear model with only a subset of variables for prediction and use numerical continuation techniques to approximate the solution path efficiently. The critical 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 perform remarkably well in predicting an object based on its observed characteristics. In terms of AI safety, it is essential 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 $\mu_{\mathcal{D}_n}$ on the observed data $\mathcal{D}_n$ and consider an interval centered around the prediction of y_{n+1} when the fitted model receives x_{n+1} as new input, *i.e.,* using $\mu_{\mathcal{D}_n}(x_{n+1})$. We calibrate the confidence interval to satisfy a $100(1-\alpha)\%$ confidence by considering, for any level α in $(0, 1)$, the set

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

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 $\{1, \ldots, n\}$. If the fitted model is close to the exact value, this method is approximately valid as n goes to infinity.

Alternatively, conformal prediction is a versatile and simple method introduced in (Vovk et al., 2005; Shafer & 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 using candidate values for y_{n+1}. Since the true y_{n+1} is not given in the observed dataset $\mathcal{D}_n$, one can instead learn a predictive model $\mu_{\mathcal{D}_{n+1}(z)}$ on an augmented database

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

where a candidate 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 simply contain the most typical z as a confidence set for y_{n+1}. More formally, the conformal prediction set is obtained as

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

where $Q_{n+1}(1-\alpha, z)$ is the $(1-\alpha)$-quantile of the empirical cumulative distribution function of the refitted residuals, e.g., $|y_i(z) - \mu_{\mathcal{D}_{n+1}(z)}(x_i)|$ for indices i in $\{1, \ldots, n+1\}$ and $y(z) = (y_1, \ldots, y_n, z)$. This method benefits from a strong coverage guarantee without any assumption on the distribution, including finite sample size n; see Section 4. The conformal prediction approach has been applied for designing uncertainty sets in active learning (Ho & Wechsler, 2008), anomaly detection (Laxhammar & Falkman, 2015;

[1]College of Computing, Georgia Institute of Technology, Atlanta, GA, USA [2] Apple (Work partly done while at Georgia Tech) [3]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Correspondence to: Etash Guha <etash@gatech.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

Bates et al., 2021), few-shot learning (Fisch et al., 2021), time series (Chernozhukov et al., 2018; Xu & Xie, 2021; Chernozhukov et al., 2021; Lin et al., 2022), or to infer the performance guarantee for statistical learning algorithms (Holland, 2020; Cella & Ryan, 2020; Ndiaye, 2022). 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 traditionally requires fitting a model $\mu_{\mathcal{D}_{n+1}(z)}$ for each possible augmented dataset $\mathcal{D}_{n+1}(z)$ corresponding to each possible candidate z for y_{n+1}. The number of possible candidates is infinite in a regression setting where an object can take an uncountable number of possible values. 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. While many algorithms encounter this problem of fitting many models under alterations to the regularization parameter λ (Park & Hastie, 2007), to our knowledge, such algorithms do not exist for general loss functions under changes to the dataset without high computation cost. We can avoid the central issue of refitting the model many times by using the structural assumptions given by the setting of General Linear Models with ℓ_1 regularization.

Contributions 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 efficiently keeps track of the weights over the space of possible candidates using the sparsity induced by the ℓ_1 regularization. We develop an efficient Conformal Prediction algorithm for sparse generalized linear models from this homotopy algorithm. Additionally, using numerical continuation and 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 Quadratic, Asymmetric and Robust Loss functions with ℓ_1 regularization.

Related Works Our methodology uses numerical continuation (also called homotopy) to generate a path of solutions. Such continuation techniques have been previously used when the objective function is differentiable (Allgower & Georg, 2012), (Hastie et al., 2004) for support vector machine, (Bach et al., 2004) for logistic regression, and more general loss functions regularized with the ℓ_1 norm in (Rosset & Zhu, 2007; Park & Hastie, 2007; Tibshirani, 2013; Mairal & Yu, 2012). However, the latter focus on the regularization path and plot the solution curve as the regularization parameter λ varies. To our knowledge, there does not exist work generating the solution curve as the label z varies in $y(z)$ for general loss functions. In the setting we consider, we recall that it is the response vector that is parameterized as $y(z) = (y_1, \dots, y_n, z)$ for a real value z; for which Garrigues & Ghaoui (2009) and Lei (2019) proposed a homotopy algorithm when the loss function is quadratic. However, such algorithms do not work for general nonlinear loss functions; our algorithm extends these works to such nonlinear loss functions. For such loss functions, works such as Ndiaye & Takeuchi (2019) aim to approximate the homotopy only enough to generate the conformal prediction set. However, this work suffers much worse as increasing accuracy is required when drawing the homotopy and cannot, for example, recover the path with quadratic loss, for which an exact homotopy algorithm is known.

Notation For a nonzero integer n, we denote $[n]$ to be the set $\{1, \cdots, n\}$. Furthermore, the row-wise feature matrix is $X = [x_1, \cdots, x_{n+1}]^\top$ such that $X \in \mathbb{R}^{(n+1)\times p}$. We use the notation X_A to refer to the sub-matrix of X assembled from the columns with indices in A. If we need to do so for only one index j, where $j \in [p]$, we use X_j. For brevity, we will define $\sigma_{\max}(X_A)$ as the maximum singular value of X_A, i.e. $\sigma_{\max}(X_A) = \|X_A\|_2$. We also similarly define $\sigma_{\min}(X_A)$. If a function $\beta(z)$ returns a vector for some input z, we can index that output vector by $\beta_A(z)$, where $A \subset [p]$ or $\beta_j(z)$ where $j \in [p]$. Moreover, given a function $f(x_i, x_j)$ of two variables, we denote the gradient of that function as ∂f. Furthermore, we use the simple notation $\partial_{i,j,k} f = \frac{\partial^3 f}{\partial x_i \partial x_j \partial x_k}$ where $i, j, k \in [2]$. 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 k in $[n+1]$, the rank of U_k among $U_1, \cdots, U_{n+1}$ is defined as $\texttt{Rank}(U_k) = \sum_{i=1}^{n+1} \mathbb{1}_{U_i \leq U_k}$.

2. Sparse Generalized Linear Models

By definition of the conformal prediction set in 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 whole path $z \mapsto \mu_{\mathcal{D}_{n+1}(z)}(x_{n+1})$ as well as the path of scores and quantiles. However, it is generally difficult to achieve. 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 weight $\beta^\star(z)$ is defined as a solution to the following optimization problem

$$\beta^\star(z) \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} f(y(z), X\beta) + \lambda \|\beta\|_1 \quad . \tag{3}$$

where the data fitting term $f(y(z), y^\star(z))$ is a non negative loss function between a prediction $y^\star(z)$ and the augmented vector of labels $y(z) = (y_1, \cdots, y_n, z)$. We parameterize a linear prediction as $y_i^\star = x_i^\top \beta^\star(z)$ and the empirical loss is

$$f(y(z), y^\star(z)) = \sum_{i=1}^{n} \ell(y_i, y_i^\star(z)) + \ell(z, y_{n+1}^\star(z)) \ .$$

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. One can also consider the loss function `Linex` (Gruber, 2010; Chang & 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. Assumptions and Properties

We first describe the structure of the optimal solution $\beta^\star(z)$ for a candidate z. A solution to the optimization problem from Equation (3) must obey the first-order optimality condition. Analyzing the solution reveals a set of weights in $\beta^\star(z)$ whose value is 0 and, thus, does not contribute to the inference. This is a crucial property of ℓ_1 regularization.

Lemma 2.1. *A vector $\beta^\star(z) \in \mathbb{R}^p$ is optimal for Equation (3) if and only if for $y^\star(z) = X\beta^\star(z)$, it holds*

$$-X^\top \partial_2 f(y(z), y^\star(z)) = \lambda v(z) \ , \tag{4}$$

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

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

Within this lemma, we wish to formally distinguish between nonzero weights and zero weights, as this helps determine the value of $v_j(z)$, per Equation (5).

Definition 2.1. *We define our active set at a point z as*

$$A(z) = \left\{ j \in [p] : \ |X_j^\top \partial_2 f(y(z), y^\star(z))| = \lambda \right\} \ . \tag{6}$$

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.

The following result provides sufficient conditions to ensure *uniqueness of the solution path*, *i.e.,* for any z, there exists a single optimal solution $\beta^\star(z)$ for Problem 3.

Lemma 2.2. *For all z, we assume that the matrix $X_{A(z)}$ is full rank and that the loss function f is strictly convex. With these two assumptions, for all candidates z, only one unique optimal solution $\beta^\star(z)$ exists. Thus, the solution path $z \mapsto \beta^\star(z)$ is well defined.*

In the following, for simplicity of the presentation of the algorithms, we will add the classical qualification condition that the active set coincides with the support of the solution for any candidate z where the path is differentiable.

3. Efficient Computation of the Solution Path

We aim to finely approximate the function $\beta^\star(z)$ as $\hat{\beta}(z)$ across all candidates z. The initial and main observation is that the active set map (resp. solution path) is piecewise constant (resp. smooth). That is to say, That is to say, the variable selected by the ℓ_1 penalty is invariant with respect to small perturbation of the input data. Building on this, the path drawing algorithm is a combination of finding points where the active set changes occur and estimating the optimal solution, leveraging the regularity of the loss f.

We have two situations for a change in the active set:

- A nonzero variable becomes zero *i.e.,* $\exists j \in A(z)$ s.t.
$$\beta_j^\star(z) \neq 0 \text{ and } \beta_j^\star(z_j^{\text{out}}) = 0 \ .$$
- A zero variable becomes nonzero *i.e.,* $\exists j \in A^c(z)$ s.t.
$$|X_j^\top \partial_2 f(y(z_j^{\text{in}}), y^\star(z_j^{\text{in}}))| = \lambda \ .$$

Here, z_j^{out} and z_j^{in} are the estimated points where variable j could leave or join the active set, respectively. With decreasing input z, the next change point occurs at

$$z_{\text{next}}(z) = \max \left(\max_{j \in A(z)} z_j^{\text{out}}, \ \max_{j \in A^c(z)} z_j^{\text{in}} \right) \ . \tag{7}$$

Here, $z_{\text{next}}(z)$ is the function that finds where the active set changes after point z. The set of change points are called *kinks* of the path because they correspond to the non-differentiable points of the solution path $z \mapsto \beta^\star(z)$. Core difficulties are that f can be highly nonlinear, and the optimal weights $\beta^*(z^+)$ at an arbitrary point z^+ cannot be efficiently computed for many loss functions. To alleviate this, our algorithm sequentially creates a linearized version of $\beta_A^\star(z^+)$ called $\tilde{\beta}_A(z^+)$ (Section 3.1) in order to estimate the active set changes (Section 3.2 and Section 3.2). Given a point of active set change z_t, we can manually correct $\tilde{\beta}_A(z_t)$ into $\hat{\beta}_A(z_t)$ so that $\hat{\beta}_A(z_t) \approx \beta_A^\star(z_t)$ up to a negligible optimization error ϵ_{tol} using any appropriate solver (Section 3.3). It then approximates $\beta_{A^+}^\star(z^+)$, where A^+ is the new active set, repeating these steps until the stopping point is reached. We detail the entire pipeline in Algorithm 2 and illustrate how our approximated solution path deviates from the exact one for different loss functions in Appendix A.

3.1. Solution Estimation

We wish to approximate $\beta_A^\star(z^+)$ for a candidate z^+ smaller than the most recently found kink z_t where $A(z^+) = A(z_t)$.

To start, we will assume access to the corrected (up to negligible error) weights $\hat{\beta}_A(z_t)$ at the previous kink z_t. We can use a local linearization of the solution path as

$$\tilde{\beta}_A(z^+) = \hat{\beta}_A(z_t) + \hat{\beta}'_A(z_t) \times (z^+ - z_t) \ , \qquad (8)$$

where, $\hat{\beta}'_A(z_t)$ is our approximation of the true slope $\frac{\partial \beta^\star_A}{\partial z}(z_t)$, which we do not have access to. To understand this term, we follow Park & Hastie (2007) to define

$$H(y(z), \beta^\star_A(z)) = X_A^\top \partial_2 f(y(z), y^\star(z)) + \lambda v_A \ ,$$

From the Optimality Condition in Equation (4), it holds

$$H(y(z), \beta^\star_A(z)) = 0 \Longrightarrow \frac{\partial H}{\partial z} = 0 \ .$$

By the implicit function theorem and the chain rule, we have

$$\begin{aligned}
\frac{\partial \beta^\star_A}{\partial z} &= -\left(\frac{\partial H}{\partial \beta}\right)^{-1} \frac{\partial H}{\partial y} \frac{\partial y}{\partial z} \\
\frac{\partial H}{\partial \beta} &= X_A^\top \partial_{2,2} f(y(z), y^\star(z)) X_A \\
\frac{\partial H}{\partial y} &= X_A^\top \partial_{2,1} f(y(z), y^\star(z)) \\
\frac{\partial y}{\partial z} &= (0, \dots, 0, 1)^\top \ .
\end{aligned}$$

To compute an approximation of $\frac{\partial \beta^\star_A}{\partial z}(z_t)$, we use a plug-in approach and only replace the (unknown) exact value of $y^\star(z_t) = X\beta^\star(z_t)$ with the approximate $\hat{y}(z_t) = X\hat{\beta}(z_t)$, yielding $\hat{\beta}'_A(z_t)$. Notably, we get an equation for $\tilde{\beta}_A(z^+)$, which is efficient to compute given $y(z^+)$. As a reminder, the loss function f differentiates this algorithm from existing path-finding algorithms tailored for changes in the hyperparameter λ. If f is the Quadratic loss function, we recover the path-finding algorithm from Lei (2019). A completely different homotopy will be generated if it is another loss function.

3.2. Active Set Updates

We have to track the changes that may occur in the active sets along the path sequentially depending on whether the variable leaves or enters the active set. We will compute our path restricted in the interval $[z_{\min}, z_{\max}]$ where

$$z_{\min} = \min(y_1, \dots, y_n) \text{ and } z_{\max} = \max(y_1, \dots, y_n) \ .$$

For sufficiently large sample size n, any point z outside this interval has a very low probability of being in the conformal set since it is an outlier of a label; see justification in Lemma C.1. For simplicity, we reiterate that we know the corrected $\hat{\beta}(z_t)$ at the most recent kink z_t approximating $\beta^*(z_t)$ up to error ϵ_{tol} and the active set of weights $A(z_t)$. We estimate the kinks by following Equation (7) and replacing the exact solution $\beta^\star(z_t)$ by $\tilde{\beta}(z_t)$ in Equation (8).

As such, we will iteratively set $z_{t+1} = z_{\text{next}}(z_t)$ as the next change point following Equation (7).

Leaving the active set At the point, where a nonzero variable becomes zero, we know that by Equation (8), we have a closed form approximation of $\beta^\star_A(z^+)$ given $\beta_A(z_t)$. Therefore, for a feature index $j \in A$, we have a closed-form approximation for $\beta^\star_j(z^+)$ in terms of z^+, which we can compute efficiently. Thus, from Equation (8), j leaving the active set occurs at $\beta^\star_j(z^+) = 0$ implies a kink occurs at z^+ when $0 \approx \tilde{\beta}_j(z^+)$ defined in the R.H.S. of Equation (8); which is easily solvable in closed-form. Thus, for an active variable j with nonvanishing gradient $\hat{\beta}'_j(\hat{z}_t) \neq 0$, we define

$$z^{\text{out}}_{j,t+1} = z_t - \frac{\hat{\beta}_j(z_t)}{\hat{\beta}'_j(z_t)} \ ,$$

and define $z^{\text{out}}_{j,t+1} = -\infty$ otherwise. We remind the reader that $\hat{\beta}'_j(z_t)$ is our approximation of the true slope $\frac{\partial \beta^\star_j}{\partial z}(z_t)$ from Section 3.1.

Joining the active set At the point where a variable becomes nonzero, we know from Equation (4) that for any inactive variable $j \in A^c$ that joins the active set

$$|X_j^\top \partial_2 f(y(z^+), X_{A^+}\beta^\star_{A^+}(z^+))| = \lambda$$

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

$$X_{A^+}\beta^\star_{A^+}(z^+) = X_A\beta^\star_A(z^+) = y^\star(z^+) \ .$$

This equivalence is useful as we know how to approximate $\beta^\star_A(z^+)$, and therefore $y^\star(z^+)$, efficiently from Equation (8). Therefore, the j-th variable must join the active set at approximately z^+ such that $\mathcal{I}_j(z^+) = 0$ where

$$\mathcal{I}_j(z^+) = |X_j^\top \partial_2 f(y(z^+), y^\star(z^+))| - \lambda \ . \qquad (9)$$

We also leverage a plug-in estimate of Equation (9) by replacing $y^\star(\cdot)$ by $\hat{y}(\cdot)$. We could use a root-finding function to efficiently find the roots of the function $\mathcal{I}_j(z^+)$ where the kink may lie. However, we seek a closed form as in Equation (8) to make finding the roots of $\mathcal{I}_j(z^+)$ more efficient. We do this via linearization again.

Approximation of $\partial_2 f(y(z^+), y^\star(z^+))$

While $\tilde{\beta}_j(z^+)$ is linear in z^+, giving way to an explicit solution for z^+, this property does not hold for $\mathcal{I}_j(z^+)$ in Equation (9). To achieve such a form, we need to linearize further $\partial_2 f(y(z^+), y^\star(z^+))$. To simplify, we denote

$$f(y(z), y^\star(z)) = f \circ \zeta(z) \text{ where } \zeta(z) = (y(z), y^\star(z)) \ ,$$

and approximate its gradient $\partial_2 f \circ \zeta(z^+)$ as

$$\partial_2 f \circ \zeta(z) + \partial_{2,1} f \circ \zeta(z)^\top \Delta y + \partial_{2,2} f \circ \zeta(z)^\top \Delta y^\star \quad (10)$$

where $\Delta y = y(z^+) - y(z)$ and $\Delta y^\star = y^\star(z^+) - y^\star(z)$. We still have that Equation (10) can be nonlinear since $\Delta y^\star$ can be nonlinear in z^+. To alleviate this, we leverage the local approximation of the solution path in Equation (8) and the plug-in replacement of $\frac{\partial \beta_A^\star}{\partial z}$ with $\hat{\beta}'_A$. As such, we can estimate the root of $\mathcal{I}_j(z^+)$ and sequentially define the next point where the jth variable becomes active. To simplify the expression, we set $\hat{\zeta}(z) = (y(z), \hat{y}(z))$ and

$$g(z_t) = [\partial_{21} f \circ \hat{\zeta}(z_t)]_{n+1} + \partial_{2,2} f \circ \hat{\zeta}(z_t)^\top X_A \hat{\beta}'_A(z_t) \ .$$

A zero variable j is estimated to become nonzero at

$$z^{\text{in}}_{j,t+1} = z_t + \frac{-X_j^\top \partial_2 f \circ \zeta(z_t) \pm \lambda}{X_j^\top g(z_t)} \ ,$$

The detailed computations are provided in Appendix D. Note that when the denominator $g(z_t)$ is zero, we set $z^{\text{in}}_{j,t+1} = -\infty$. Finally, the next kink is estimated as

$$z_{t+1} = \max\left(\max_{j \in A(z_t)} z^{\text{out}}_{j,t+1}, \max_{j \in A^c(z_t)} z^{\text{in}}_{j,t+1} \right) \ .$$

3.3. Solution Updates

Our active set change point finder obtains the next kink z_{t+1} by tracking all variables in the optimal solution to see whether or not it cancels out after z_t. However, our kink-finding tool requires exact knowledge of $\hat{\beta}(z_t)$, as in Equation (8). To find the next kink, we, therefore, need to know $\hat{\beta}(z_{t+1})$. To ensure that our linearized version $\tilde{\beta}_A(z_{t+1})$ is close enough to the exact solution $\beta^\star_A(z_{t+1})$, we manually correct our linearized weights $\tilde{\beta}_A(z_{t+1})$, creating our $\hat{\beta}_A(z_{t+1})$. We use the Predictor-Corrector strategy described below (Allgower & Georg, 2012).

Predictor To initialize the solving process for $\hat{\beta}(z_{t+1})$, we first provide our linearized version $\tilde{\beta}(z_{t+1})$ from Equation (8) as a warm start initialization. This vastly improves the computation time of our corrector step here after.

Corrector The solution obtained in the warm start often has a reasonably small approximation error. For example, in the case of the Quadratic loss, this warm start is exact and correction is unnecessary. However, it generally is an imprecise estimate of the exact solution. To overcome this, we use an additional corrector step using an iterative solver, such as proximal gradient descent initialized with the predictor output, or more advanced solvers such as `CVXPY` (Diamond & Boyd, 2016) or `SKGLM` (Bertrand et al., 2022). This takes our linearized weight estimates of $\tilde{\beta}(z_{t+1})$ and outputs our approximate weights $\hat{\beta}(z_{t+1}) \approx \beta^*(z_{t+1})$ up to error ϵ_{tol} which is a hyperparameter for our corrector.

Finally, we can summarize our approximation of the homotopy as the following.

$$\hat{\beta}(z) = \begin{cases} \tilde{\beta}(z) \text{ if } z \notin \{z_1, \dots, z_t\} \\ \tilde{\beta}^\star(z) \text{ if } z \in \{z_1, \dots, z_t\} \text{ (output of corrector)} \end{cases}$$

For point z that is not a kink, we form our estiamte weights simply through the linearization. Otherwise, we can use the output of the corrector as our estimates.

Algorithm 1 Full Homotopy Generation

Input Data: $\{(x_1, y_1), \dots, (x_n, y_n)\}$, x_{n+1}, $\lambda > 0$
Initialization: $t = 0$
$z_0 = \max(y_1, \dots, y_n) \quad y(z_0) = (y_1, \dots, y_n, z_0)$
$\beta^\star(z_0) = \arg\min_{\beta \in \mathbb{R}^p} f(y(z_0), X\beta) + \lambda \|\beta\|_1$
$A(z_0) = \left\{ j \in [p] : \ |X_j^\top \partial_2 f(y(z_0), y^\star(z_0))| = \lambda \right\}$
while $z_t > \min(y_1, \dots, y_n)$ **do**
  $z_{\mathcal{I}} = \max_{j \in A^c(z_t)} z^{\text{in}}_{j,t+1}$ and $j_{\mathcal{I}} = \arg\max_{j \in A^c(z_t)} z^{\text{in}}_{j,t+1}$
  $z_{\mathcal{O}} = \max_{j \in A(z_t)} z^{\text{out}}_{j,t+1}$ and $j_{\mathcal{O}} = \arg\max_{j \in A(z_t)} z^{\text{out}}_{j,t+1}$
  if $z_{\mathcal{I}} > z_{\mathcal{O}}$ **then**
    $z_{t+1} = z_{\mathcal{I}}$
    $A(z_{t+1}) = A(z_t) \cup \{j_{\mathcal{I}}\}$
  else
    $z_{t+1} = z_{\mathcal{O}}$
    $A(z_{t+1}) = A(z_t) \setminus \{j_{\mathcal{O}}\}$
  end if
  Predictor

$$\tilde{\beta}(z_{t+1}) = \hat{\beta}(z_t) + \hat{\beta}'(z_t) \times (z_{t+1} - z_t)$$

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

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

  $t = t + 1$
end while
RETURN: $\hat{\beta}(z_t)$, z_t for all t

4. Conformal Prediction for Sparse GLM

Given a homotopy for specific data and loss function, computing the Conformal Prediction set relies on a simple calculation using the homotopy. Meanwhile, the primary tool for proving its validity is that the rank of one variable among an exchangeable and identically distributed sequence follows a (sub)-uniform distribution (Bröcker & 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 prediction loss $|z - y^\star_{n+1}(z)|$ and compute its rank relative to the losses of the other datapoints. The candidate will be considered conformal if the rank of its loss is sufficiently small. Let us define the conformity measure for $\mathcal{D}_{n+1}(z)$ as

$$E_i(z) = |y_i - y^\star_i(z)|, \quad \forall i \in [n] \ , \tag{11}$$
$$E_{n+1}(z) = |z - y^\star_{n+1}(z)| \ . \tag{12}$$

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

$$\pi(z) = 1 - \frac{1}{n+1}\texttt{Rank}(E_{n+1}(z)) \ . \tag{13}$$

The conformal prediction set will collect the most conformal z as a confidence set for y_{n+1}, *i.e.,* gathers all the real values z such that $\pi(z) \geq \alpha$. This condition occurs 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.,*

$$\{z \in \mathbb{R} : \ E_{n+1}(z) \leq Q_{n+1}(1-\alpha, z)\} \ ,$$

which is exactly the conformal set defined in Equation (2). We need to calculate the piecewise constant function $z \mapsto \pi(z)$ to compute a conformal set. Fortunately, our framework directly sheds light on the computation of this value over the range space.

Access to the homotopy, as well as the kinks, yields an efficient methodology for calculating the conformal prediction set over the range space. Once can readily use a root-finding approach (Ndiaye & Takeuchi, 2021) but it requires the assumption that the conformal set is an interval. Instead, we do so by tracking where changes in this set occur. Naturally, changes in the rank function only occur when the error of one example surpasses or goes below that of the error of the last example. Formally, this can be seen when

$$|y_i(z) - y^\star_i(z)| = |y_{n+1}(z) - y^\star_{n+1}(z))| \ . \tag{14}$$

We will look between the two kinks to efficiently find points satisfying Equation (14). For a point z between two kinks, we can efficiently estimate $y^\star(z)$. Indeed, given a point z is between two kinks z_t and z_{t+1} with an active set A, we can use Equation (8) to estimate the quantity $y(z) - y^\star(z)$ as

$$\mathcal{F}(z) = y(z) - \hat{y}(z_t) + X\hat{\beta}'_A(z_t) \times (z - z_t) \ ,$$

where $\hat{\beta}_A(z_t)$ is stored from the corrector step at the kink z_t. Given that this value is linear in z, we can form a closed-form explicit approximation for what z solves Equation (14). Therefore, we can look for where the $\pi(z)$ value changes between every sequential pair of kinks. 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.

Algorithm 2 Conformal Set Generation

Input Data: $\{z_t, \hat{\beta}(z_t)\}_{t\in[0:T]}, \alpha \in (0,1)$
`//Find where changes in π occur`
Set $\mathcal{C} = \emptyset$
for $t \in [T], i \in [n]$ **do**
  if $\exists z^+$ s.t. $\mathcal{F}(z^+)_i = \mathcal{F}(z^+)_{n+1}$ **then**
    if $|\mathcal{F}(z_t)_{n+1}| \geq |\mathcal{F}(z_t)_i|$ **then**
      $\mathcal{C} = \mathcal{C} \cup \{(z^+, -1)\}$
    else if $|\mathcal{F}(z_t)_{n+1}| \leq |\mathcal{F}(z_t)_i|$ **then**
      $\mathcal{C} = \mathcal{C} \cup \{(z^+, +1)\}$
    end if
  end if
end for
`//Get z s.t. Rank`$(E_{n+1}(z))$ `is small`
Set $\mathcal{E} = \emptyset$
$\mathcal{R} = \texttt{Rank}(E_{n+1}(z_0))$
`SORT`$(\mathcal{C})$ according to first argument z
for $z, c \in \mathcal{C}$ **do**
  $\mathcal{R} = \mathcal{R} + c$
  $\texttt{Rank}(E_{n+1}(z)) = \mathcal{R}$
  if $\texttt{Rank}(E_{n+1}(z)) \leq \lceil (n+1)(1-\alpha) \rceil$ **then**
    $\mathcal{E} = \mathcal{E} \cup \{z\}$
  end if
end for
RETURN: $(\min(\mathcal{E}), \max(\mathcal{E}))$

5. Theoretical Analysis

To understand where and how our algorithm fails, we provide an upper bound on the pointwise error of our algorithm. The error is mainly accumulated in the linearizations we use for estimating the solution and gradient of the loss. To form such bounds, we need assumptions on the regularity of the loss function f itself and on the sequence of design matrix restricted on the active sets along the path. Namely, we will see that the derivatives of the loss function is bounded.

Lemma 5.1. *The second derivatives, assumed to be continuous, of the loss function f are locally bounded by data-dependent constants. Indeed, for any $z \in [z_{\min}, z_{\max}]$, we have $\beta^\star(z) \in \mathcal{B}_{\|\cdot\|_1}(0, R/\lambda)$ where*

$$R = \max_{z\in[z_{\min}, z_{\max}]} f(y(z), \mathbf{0}) \ .$$

By Weierstrass theorem, for any $i, j \in [2]$, we have

$$\|\partial_{i,j} f \circ \zeta(z)\|_2 \leq \nu_f \ .$$

Lemma 5.2. *We assume that the loss f is μ_f-strongly convex* i.e., $\mu_f := \inf_{\|\zeta\|\leq B} \|\partial_{2,2} f \circ \zeta(z)\| > 0$, *where B is provided in the appendix. Thus, for any $z \in [z_{\min}, z_{\max}]$, the maximum singular value of the inverse of the matrix $\frac{\partial H}{\partial \beta} = X_A^\top \partial_{2,2} f \circ \zeta(z) X_A$ is upper bounded as*

$$\left\| \frac{\partial H}{\partial \beta}^{-1} \right\|_2 \leq \frac{1}{\sigma^2_{\min}(X_A) \times \mu_f} \ .$$

With these two lemmas, we can form our error bounds.

Theorem 5.1. *The error between our linearized weights $\tilde{\beta}(z^+)$ and the true weights $\beta^\star(z^+)$ is upper bounded by*

$$\left\|\tilde{\beta}(z^+) - \beta^\star(z^+)\right\|_2 \leq \epsilon_{\text{tol}} + L \times \frac{\nu_f}{\mu_f} \times |z^+ - z_t| \ .$$

where $L = \dfrac{\sigma_{\max}(X_{A(z_t)})}{\sigma^2_{\min}(X_{A(z_t)})} + \sup_{z \in [z^+, z_t]} \dfrac{\sigma_{\max}(X_{A(z)})}{\sigma^2_{\min}(X_{A(z)})}$,

and z_t is the prior kink of z^+.

Theorem 5.2. *The estimation error is upper bounded by*

$$\left\|\partial_2 f \circ \zeta(z^+) - \partial_2 f \circ \hat{\zeta}(z^+)\right\|_2 \leq K \left[\epsilon_{\text{tol}} + L\frac{\nu_f}{\mu_f}|z^+ - z_t|\right]$$

where $K = \nu_f \times \sigma_{\max}(X_A)$.

6. Numerical Experiments

Our central claim is twofold. Our method efficiently and accurately generates the homotopy over general loss functions. Our method also efficiently and accurately generates conformal sets over general loss functions. We demonstrate these two claims over different datasets and loss functions. For reproducibility, our implementation is at `github.com/EtashGuha/sparse_conformal`.

Datasets We use four datasets to illustrate the performance of our algorithm. The first three are real datasets sourced from (Pedregosa et al., 2011). The Diabetes dataset is a regression dataset with 20 features and 442 samples. Additionally, we use the well-known regression dataset from (H., 1991) denoted as Friedman1, which has 10 features and 100 samples. We also use the multivariate dataset denoted Friedman2 from (Breiman, 1996), which has 100 samples and 4 features. These datasets are used to demonstrate the capabilities of our algorithm on real datasets. We also generate regression problems synthetically. We sample the data and labels from a uniform distribution between $[-1, 1]$. We also divide by the standard deviation to normalize the dataset. We generate two different synthetic datasets, one normal-sized dataset, denoted `synthetic` with 100 samples and 100 features, and a larger dataset, denoted `large` with 1000 features and 20 samples. This larger dataset is intended to display our algorithm's complexity in terms of the number of features. These datasets represent a reasonable range of regression problems usable for our experiments.

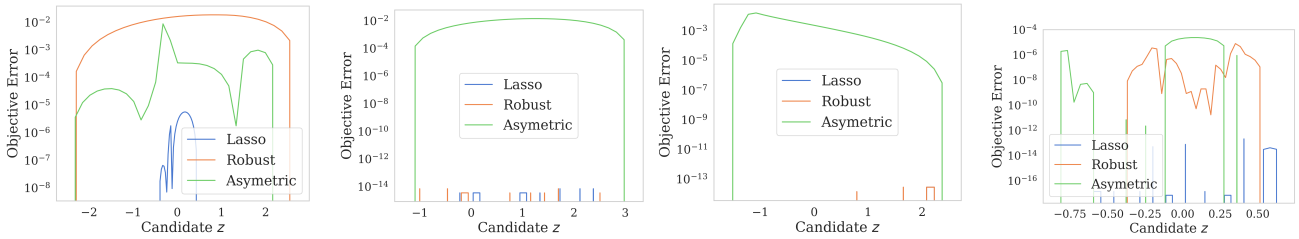
Baselines To form a baseline for our algorithm, we use several baselines. This baseline is the most naive conformal prediction algorithm. For Grid algorithms, the algorithm selects 100 potential candidates evenly across the range of possible candidates. It uses the primal corrector at each point to calculate the weights to form the homotopy. A more sophisticated conformal prediction and homotopy generating algorithm is the Approximate homotopy from (Ndiaye & Takeuchi, 2019), which leverages loss function smoothness to track violations (up to a prescribed error tolerance) of the optimality condition along the path.

6.1. Homotopy Experiments

To test our algorithm in terms of homotopy generation, we measure our algorithm's accuracy and efficacy against different baselines across different loss functions. For all baselines and our algorithm, we use Proximal Gradient Descent for Lasso Loss and `CVXPY` for Robust and Asymmetric as Primal Correctors. Precisely, we measure the negative logarithm of the gap between primal values of the calculated $\hat{\beta}$ 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 we compute the homotopy iteratively along a find grid of candidates. 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. Our experiments cover the Lasso, Robust, and Asymmetric functions across all the datasets.

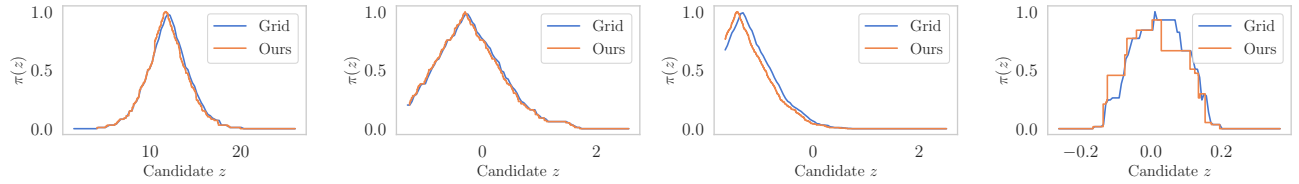
We report our results in Table 1 and Table 2. We shorten Synthetic to `Synth` and Approximate to `Appr` for brevity. As evident, we see a significant decrease in time used over Approximate Homotopy for most applications of the Lasso Loss with a significant increase in accuracy. On the largest dataset for Lasso Loss, our algorithm gets similar accuracy and is much more efficient. 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 and Large dataset for Asymmetric loss and the Synthetic and Large 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 illustrate the accuracy of our algorithm, we plot the optimization error gap over the space of all $z \in [z_{\min}, z_{\max}]$ for all three loss functions and four datasets. We report the figures in Figure 1. Notably, we see that on Figure 1d, we achieve all losses better than 10^{-4}. On other figures, all objective errors are bounded by 10^{-2}. Our application of Lasso and Robust over all datasets achieves near 0 objective error over the entire pass.



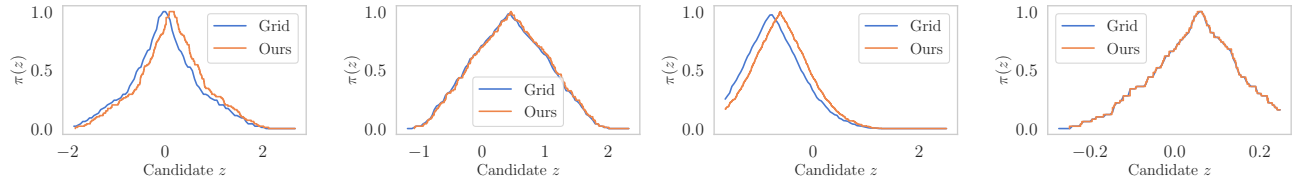
(a) Friedman1 (b) Friedman 2 (c) Diabetes (d) Synthetic

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



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

Figure 2: 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 3: 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.

Table 1: Average Time of Homotopy

	Dataset				
	Synth.	**Friedman1**	**Diabetes**	**Friedman 2**	**Large**
Our Lasso	**1.706**	**1.945**	**1.0785**	**0.681**	**150.012**
Appr. Lasso	5.176	43.823	70.813	14.055	500.820
Our Robust	**27.156**	1.069	2.411	0.701	**323.372**
Appr. Robust	62.894	1.009	2.734	0.618	607.203
Our Asym.	**9.270**	3.147	**27.349**	2.454	**41.269**
Appr. Asym.	18.963	2.699	54.149	3.342	82.857

Table 2: Average Negative Logarithm of Primal Gap of Homotopy

	Dataset				
	Synth.	**Friedman1**	**Diabetes**	**Friedman2**	**Large**
Our Lasso	**12.498**	**15.844**	**16.001**	**15.241**	7.933
Appr. Lasso	6.597	6.469	7.554	6.702	7.558
Our Robust	5.137	2.317	3.819	2.778	5.223
Appr. Robust	5.990	3.561	3.712	4.434	5.026
Our Asym.	7.879	3.633	3.814	3.058	6.101
Appr. Asym.	6.939	3.208	4.032	2.795	5.365

6.2. 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. We demonstrate this both visually and empirically. We draw the $\pi(z)$ function for visual verification over all four datasets and three loss functions using our algorithm. To form a baseline, we use the Grid algorithm. This algorithm is a ground truth to which we compare our $\pi(z)$ function. For empirical verification, we compare coverage, length, and the time of our method vs. several important baselines. Namely, we use the Grid method, Approximate homotopy from (Ndiaye & Takeuchi, 2019), the Oracle methodology, which has access to the true value of y_{n+1} to form its conformal interval, and the Split methodology, which uses a calibration dataset to calibrate the conformal values predicted but loses statistical validity.

	Diabetes Coverage			Friedman 1 Coverage			Friedman 2 Coverage			Synthetic Coverage		
	Lasso	**Robust**	**Asymmetric**	**Lasso**	**Robust**	**Asymmetric**	**Lasso**	**Robust**	**Asymmetric**	**Lasso**	**Robust**	**Asymmetric**
Ours	0.933	0.933	0.867	0.900	0.900	0.883	0.900	0.850	0.933	0.900	0.900	0.900
Approximate	0.933	0.933	0.867	0.850	0.900	0.833	0.900	0.800	0.900	0.850	0.900	0.850
Split	0.933	0.867	0.800	0.867	0.933	0.867	1.000	0.867	0.933	1.000	1.000	0.900
Grid	0.933	0.933	0.867	0.767	0.767	0.767	0.900	0.767	0.933	0.850	0.850	0.850
Oracle	0.933	0.933	0.867	0.867	0.967	0.933	0.900	0.867	0.933	1.000	0.900	1.000

(a) Coverage Results over Several Datasets

	Diabetes Length			Friedman 1 Length			Friedman 2 Length			Synthetic Length		
	Asymmetric	**Lasso**	**Robust**	**Lasso**	**Robust**	**Asymmetric**	**Lasso**	**Robust**	**Asymmetric**	**Lasso**	**Robust**	**Asymmetric**
Ours	0.867	2.234	2.230	2.340	2.570	2.875	2.004	2.191	2.621	0.632	0.714	0.702
Approximate	0.867	2.262	2.237	2.368	2.599	2.897	2.024	2.245	2.618	0.786	0.705	0.790
Split	0.800	2.409	2.429	2.589	2.837	3.219	2.361	2.448	2.888	0.831	0.831	0.831
Grid	0.867	2.286	2.255	2.475	2.782	2.982	2.108	2.338	2.741	0.872	0.903	0.651
Oracle	0.867	2.320	2.337	2.204	2.508	2.787	2.240	2.447	2.550	0.062	0.001	0.226

(b) Length Results over Several Datasets

	Diabetes Time (s)			Friedman 1 Time (s)			Friedman 2 Time (s)			Synthetic Time (s)		
	Lasso	**Robust**	**Asymmetric**	**Lasso**	**Robust**	**Asymmetric**	**Lasso**	**Robust**	**Asymmetric**	**Lasso**	**Robust**	**Asymmetric**
Ours	0.658	1.125	6.865	0.493	0.398	1.047	0.326	0.305	0.901	6.056	115.565	4.037
Approximate	4.012	33.858	213.995	5.127	4.234	17.111	1.538	2.991	13.178	30.124	332.152	9.142
Split	0.025	0.086	0.474	0.139	0.041	0.102	0.036	0.042	0.100	0.599	0.122	0.039
Grid	0.769	5.692	58.632	1.704	2.238	11.331	0.564	2.068	8.834	29.150	431.398	2.418
Oracle	0.049	0.188	1.032	0.116	0.034	0.212	0.040	0.033	0.193	0.429	0.256	0.063

(c) Time Results over Several Datasets

Figure 4: We demonstrate the performance of our Conformal Set Algorithm against several baselines across many datasets and loss functions. Our algorithm maintains strong coverage, length, and time metrics across many loss functions and datasets.

Visual Results We report the figures in Figure 3. 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 particular examples, notably Figures 2d, 3a, and 3c, the trace is less accurate than the others. However, the error is within a reasonable range to achieve the desired coverage and length guarantees. We also report similar experiments for the Lasso loss, but we mention these in Appendix A since our method is exact for the Lasso loss. 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.

Empirical Results We report our empirical results in Figure 4a, Figure 4b, and Figure 4c. We can see that most methods maintain strong coverage guarantees over all datasets. For our experiments, we used $\alpha = 0.1$, and most of our results hover around that level of coverage. Moreover, in Figure 4b, we see that except for Oracle, across several loss functions and datasets, our algorithm achieves the smallest length. The Oracle, however, consistently has the best length due to its knowledge of the true y_{n+1}. Also, our algorithm is the fastest over all homotopy methods but slower than Split and Oracle, as seen in Figure 4c. Therefore, our experiments indicate that our Conformal Prediction Algorithm is competitive in all coverage, length, and time measures.

7. Conclusion

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. Furthermore, we achieve explicit closed-form equations to model the behavior of this homotopy. 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. Spline instead of linear interpolation may yield improved accuracy for different loss functions. Additionally, smoothing at the kinks may reduce the algorithm's sensitivity 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. *Numerical continuation methods: an introduction*. Springer Science & Business Media, 2012.

Bach, F., Thibaux, R., and Jordan, M. Computing regularization paths for learning multiple kernels. *Advances in neural information processing systems*, 2004.

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

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

Bertrand, Q., Klopfenstein, Q., Bannier, P.-A., Gidel, G., and Massias, M. Beyond l1: Faster and better sparse models with skglm. In *NeurIPS*, 2022.

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

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

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

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

Chernozhukov, V., Wüthrich, K., and Zhu, Y. Exact and robust conformal inference methods for predictive machine learning with dependent data. *Conference On Learning Theory*, 2018.

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

Diamond, S. and Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *J. Mach. Learn. Res*, 2016.

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

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

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

H., F. J. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 1991.

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

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

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

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

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

Lin, Z., Trivedi, S., and Sun, J. Conformal prediction intervals with temporal dependence. *Transactions of Machine Learning Research*, 2022.

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

Ndiaye, E. Stable conformal prediction sets. In *International Conference on Machine Learning*. PMLR, 2022.

Ndiaye, E. and Takeuchi, I. Computing full conformal prediction set with approximate homotopy. *NeurIPS*, 2019.

Ndiaye, E. and Takeuchi, I. Root-finding approaches for computing conformal prediction set. *arXiv preprint arXiv:2104.06648*, 2021.

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

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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

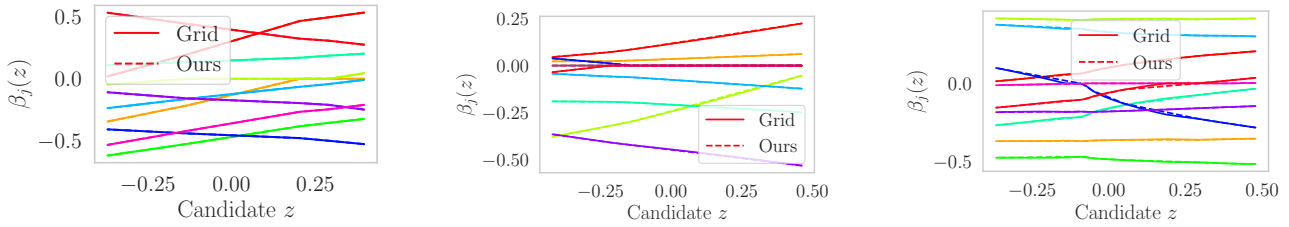
Rosset, S. and Zhu, J. Piecewise linear regularized solution paths. *The Annals of Statistics*, 2007.

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

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

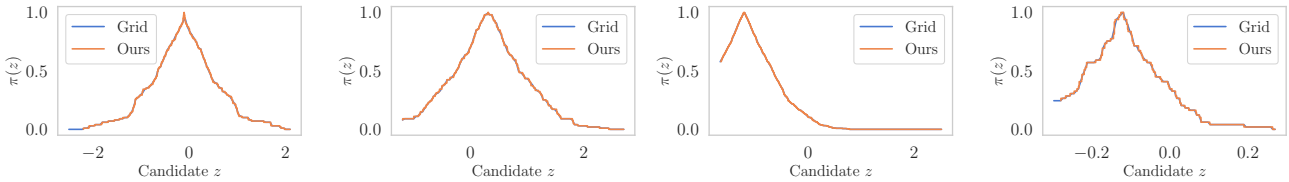
Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*. Springer, 2005.

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



(a) Lasso (b) Asymmetric (c) Robust

Figure 5: 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.



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

Figure 6: The $\pi(z)$ function as generated by a ground truth discretized searching algorithm and by our homotopy drawing algorithm for the Quadratic Loss function over all 4 datasets.

A. Additional Visualizations

We have provided two extra visualizations for the reader's understanding. We have provided figures of the homotopy for different loss functions and what the conformality function $\pi(z)$ looks like for the Quadratic Loss function on both our real and synthetic datasets.

Homotopy Visualizations We run our homotopy generation algorithm over Quadratic Loss, Robust, and Asymmetric Loss functions over several low-dimensional synthetic examples. As we can see in Figure 6a, for the Quadratic Loss, our homotopy algorithm perfectly matches that of the Grid baseline since our algorithm captures the Quadratic Loss homotopy from Lei (2019) exactly. Moreover, in Figure 6b and Figure 6c, we see that our algorithm very closely identifies the homotopy of the Grid algorithm. In the Robust and Asymmetric cases, the linearization causes a slight miss in the kink, but the difference is negligible. Across all dimensions, our homotopy generation algorithm closely tracks that of the baseline Grid algorithm. These visualizations verify visually that our homotopy generation algorithm is accurate.

Conformity Function for Quadratic Loss We also visualize what the conformality function $\pi(z)$ looks like across several datasets for the Quadratic Loss function. We do not include these in the main manuscript since our algorithm is exact on the Quadratic Loss function, and no visual verification is truly needed. Nevertheless, we provide such visuals in Figure 6. Our Conformal Prediction algorithm indeed matches precisely that of the Grid baseline algorithm. This confirms our claims that our algorithm is indeed exact on the Quadratic Loss function.

B. Proofs for Properties of GLM's

B.1. Proof of Lemma 2.1

Lemma 2.1. *A vector $\beta^\star(z) \in \mathbb{R}^p$ is optimal for Equation (3) if and only if for $y^\star(z) = X\beta^\star(z)$, it holds*

$$-X^\top \partial_2 f(y(z), y^\star(z)) = \lambda v(z) \ , \tag{4}$$

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

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

Proof. The Fermat rule reads

$$0 \in \{X^\top \partial_2 f(y(z), y^\star(z))\} + \lambda \partial \left\|\cdot\right\|_1 (\beta^\star(z)) \ .$$

Defining $v(z) \in \partial \left\|\cdot\right\|_1 (\beta^\star(z))$ yields Equation (4). To show Equation (5), we look at $v_j(z)$ for any index j. We remind that by separability of the ℓ_1 norm, we have $v_j(z) = \partial|\cdot|(\beta_j^\star(z))$. Hence, $v_j(z) = \text{sign}(\beta_j^\star(z))$ if $\beta_j^\star(z) \neq 0$ and $v_j(z) \in [-1, 1]$ otherwise. This proves the claim.

□

B.2. Proof of Lemma 2.2

Lemma 2.2. *For all z, we assume that the matrix $X_{A(z)}$ is full rank and that the loss function f is strictly convex. With these two assumptions, for all candidates z, only one unique optimal solution $\beta^\star(z)$ exists. Thus, the solution path $z \mapsto \beta^\star(z)$ is well defined.*

Proof. We first prove that $A(z)$ is unique. From the definition of the active set, we have

$$A(z) = \left\{j \in [p] : \ |X_j^\top \partial_2 f(y(z), X\beta^\star(z))| = \lambda\right\} \ ,$$

where we remind that

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

Since, from strict convexity of f, the prediction $X\beta^\star(z)$ is unique for any solution $\beta^\star(z)$ to the aforementioned optimization problem, we have $A(z)$ is uniquely defined. From the first order optimality condition, it exists $v(z) \in \partial \left\|\cdot\right\|_1 (\beta^\star(z))$

$$0 \in X^\top \partial_2 f(y(z), X\beta^\star(z)) + \lambda v(z) \ .$$

Restricted to the active set yields

$$0 \in X_A^\top \partial_2 f(y(z), X_A\beta_A^\star(z)) + \lambda v_A(z) \iff \beta_A^\star(z) \in \underset{w \in \mathbb{R}^{|A|}}{\arg\min}\, f(y(z), X_A w) + \lambda \left\|w\right\|_1 \ .$$

Since f is strictly convex and X_A is full rank, the latter optimization problem is strictly convex meaning $\beta_A^\star(z)$ is unique. □

B.3. Proof of Lemma B.1

Lemma B.1. *Let $\mathbf{0}$ be the vector of 0's, For all $z \in [y_{\min}, y_{\max}]$, we have that the optimal weights $\beta^\star(z)$ satisfy*

$$\{\beta^\star(z) : z \in [z_{\min}, z_{\max}]\} \subset \{\beta : \left\|\beta\right\|_1 \leq R/\lambda\} \text{ where } R = \sup_{z \in [z_{\min}, z_{\max}]} f(y(z), \mathbf{0}).$$

Proof. Let's denote the objective function as

$$P(\beta, z) = f(y(z), X\beta) + \lambda \left\|\beta\right\|_1 .$$

We remind the reader that the solution $\beta^\star(z)$ satisfies $\beta^\star(z) = \arg\min_\beta P(\beta, z)$. By optimality and assuming that f is non-negative, we have for any z

$$\lambda \left\|\beta^\star(z)\right\|_1 \leq P(\beta^\star(z), z) \leq P(\mathbf{0}, z) = f(y(z), \mathbf{0}) \ .$$

Here, $\mathbf{0}$ is the vector of 0's, the first step comes from the definition of P, and the second inequality comes from the fact that $\beta^\star(z)$ is a minimizer of P. Naturally, we then have that the ℓ_1 norm of the the solving weights $\beta^\star(z)$ is bounded by the value of $f(y(z), \mathbf{0})$. Any solution $\beta^\star(z)$ is inside the ℓ_1 ball centered at $\mathbf{0}$ with radius R/λ.

Since the path is truncated *i.e.,* $z \in [z_{\min}, z_{\max}]$, then the solution path is bounded *i.e.,*

$$\{\beta^\star(z) : z \in [z_{\min}, z_{\max}]\} \subset \left\{\beta : \|\beta\|_1 \leq \frac{R}{\lambda}\right\} \ ,$$

where

$$R = \sup_{z \in [z_{\min}, z_{\max}]} f(y(z), \mathbf{0}) \ .$$

□

Also, it is easy to see that, along the path

$$\begin{aligned}
\|y(z)\|_2 &\leq \max(\|y(z_{\min})\|_2, \|y(z_{\max})\|_2) \\
\|y^\star(z)\|_2 &= \|X_A \beta_A^\star\|_2 \leq \frac{\sigma_{\max}(X_A) \times R}{\lambda} \\
\|\zeta(z)\|_2 &= \sqrt{\|y(z)\|_2^2 + \|y^*(z)\|_2^2} \\
&\leq \sqrt{\max(\|y(z_{\min})\|_2, \|y(z_{\max})\|_2)^2 + \left(\frac{\sigma_{\max}(X_A) \times R}{\lambda}\right)^2} =: B \ .
\end{aligned}$$

Note that, for simplicity, we naturally suppose that the estimate $\hat{\beta}(z)$ is a better minimizer than the vector $\mathbf{0}$. Thus, the same bounds above hold for $\hat{\beta}(z)$, $\hat{y}(z)$ and $\hat{\zeta}(z)$.

C. Choice of the Range $[z_{\min}, z_{\max}]$

Lemma C.1. *Choosing $z_0 = z_{\max} = \max(y_1, \ldots, y_n)$ and stopping once $z_t \leq z_{\min} = \min(y_1, \ldots y_n)$ reduces the probability of coverage by at most $\frac{2}{n+1}$.*

Proof. Given our exchangability assumption, the probability that $y_{n+1} \geq z_{\max}$ is at most $\frac{1}{n+1}$. Similarly, the probability that $y_{n+1} \leq z_{\min}$ is at most $\frac{1}{n+1}$. Therefore, using the union bound, the probability that choosing the criteria we do in our algorithm affects coverage by at most $\frac{2}{n+1}$, which becomes negligible as n grows. □

D. Details on $\partial_2 f$

In the main text, we mentioned that we are approximating $\partial_2 f(y(z^+), y^\star(z^+))$. We can do this via linearization.

$$\partial_2 f \circ \zeta(z^+) \approx \partial_2 f \circ \zeta(z) + \partial_{2,1} f \circ \zeta(z)(y(z^+) - y(z)) + \partial_{2,2} f \circ \zeta(z)(y^\star(z^+) - y^\star(z))$$

Moreover,

$$\begin{aligned}
y(z^+) - y(z) &= (0, \ldots, 0, z^+ - z) \\
y^\star(z^+) - y^\star(z) &= X_A(\beta_A^\star(z^+) - \beta_A^\star(z)) \approx X_A \frac{\partial \beta_A^\star}{\partial z}(z) \times (z^+ - z)
\end{aligned}$$

Finally, with a plug-in approach, we approximate $\hat{\beta}'_A(z) \approx \frac{\partial \beta_A^\star}{\partial z}(z)$ and obtain

$$\partial_2 f \circ \zeta(z^+) \approx \partial_2 f \circ \zeta(z) + \left([\partial_{2,1} f \circ \zeta(z)]_{n+1} + [\partial_{2,2} f \circ \zeta(z)] \, X_A \hat{\beta}'_A(z) \right) \times (z^+ - z)$$

Here, the first equality come from definition, the second is from applying the chain rule to intermediate variables, the third is from a simple notational switch, and the final equality comes from using our estimators for $\beta^\star$. Now, we can find the roots of $\mathcal{I}$ from Equation (9) as the following

$$z^{\text{in}}_{j,t+1} = z_t + \frac{-X_j^\top \partial_2 f \circ \hat{\zeta}(z_t) \pm \lambda}{X_j^\top \left[[\partial_{2,1} f \circ \hat{\zeta}(z_t)]_{n+1} + \partial_{2,2} f \circ \hat{\zeta}(z_t)^\top X_A \hat{\beta}'_A(z_t) \right]} \ .$$

We can now present our desired theorem.

Lemma D.1. *The gradient of the solution path $\frac{\partial \beta^\star}{\partial z}(z)$, as well as its estimates $\hat{\beta}'(z)$ are bounded as follow*

$$\max \left(\left\| \frac{\partial \beta^\star}{\partial z}(z) \right\|, \left\| \hat{\beta}'(z) \right\| \right) \leq \frac{\sigma_{\max}(X_{A(z)})}{\sigma^2_{\min}(X_{A(z)})} \times \frac{\nu_f}{\mu_f} \ .$$

Proof. We remind that $\frac{\partial y}{\partial z} = (0, \dots, 0, 1)^\top$ and

$$\frac{\partial \beta^\star_A}{\partial z} = -\left(\frac{\partial H}{\partial \beta} \right)^{-1} \frac{\partial H}{\partial y} \frac{\partial y}{\partial z} \qquad \hat{\beta}'_A = -\left(\widehat{\frac{\partial H}{\partial \beta}} \right)^{-1} \widehat{\frac{\partial H}{\partial y}} \frac{\partial y}{\partial z}$$

$$\frac{\partial H}{\partial \beta} = X_A^\top \partial_{2,2} f \circ \zeta(z) X_A \qquad \widehat{\frac{\partial H}{\partial \beta}} = X_A^\top \partial_{2,2} f \circ \hat{\zeta}(z) X_A$$

$$\frac{\partial H}{\partial y} = X_A^\top \partial_{2,1} f \circ \zeta(z) \qquad \widehat{\frac{\partial H}{\partial y}} = X_A^\top \partial_{2,1} f \circ \hat{\zeta}(z)$$

Hence

$$\left\| \frac{\partial \beta^\star_A}{\partial z} \right\|_2 \leq \left\| \left(\frac{\partial H}{\partial \beta} \right)^{-1} \right\|_2 \left\| \frac{\partial H}{\partial y} \right\|_2$$

By definition, we have that for any $z \in [z_{\min}, z_{\max}]$,

$$\left\| \frac{\partial H}{\partial \beta}(y(z), \beta^\star_A(z)) \right\|_2 = \left\| X_A^\top \partial_{2,2} f \circ \zeta(z) X_A \right\|_2 \geq \sigma_{\min}(X_A^\top X_A) \times \inf_{\|\zeta\| \leq B} \sigma_{\min}(\partial_{2,2} f \circ \zeta(z)) \ .$$

Since f is assumed to be μ_f-strongly convex from Lemma 5.2, it holds

$$\inf_{\|\zeta\| \leq B} \sigma_{\min}(\partial_{2,2} f \circ \zeta(z)) \geq \mu_f > 0 \ ,$$

and then

$$\left\| \left(\frac{\partial H}{\partial \beta} \right)^{-1} \right\|_2 \leq \frac{1}{\sigma^2_{\min}(X_A) \times \mu_f} \ .$$

Similarly, given f is smooth with constant ν_f from Lemma 5.1, we have

$$\left\| \frac{\partial H}{\partial y}(y(z), \beta^\star_A(z)) \right\|_2 = \left\| X_A^\top \partial_{2,1} f \circ \zeta(z) \right\|_2 \leq \sigma_{\max}(X_A) \left\| \partial_{2,1} f \circ \zeta(z) \right\|_2 \leq \sigma_{\max}(X_A) \times \nu_f \ .$$

Hence the result. The proof for upper-bounding the estimated gradient norm follows the same line. □

Theorem 5.1. *The error between our linearized weights $\tilde{\beta}(z^+)$ and the true weights $\beta^\star(z^+)$ is upper bounded by*

$$\left\|\tilde{\beta}(z^+) - \beta^\star(z^+)\right\|_2 \leq \epsilon_{\text{tol}} + L \times \frac{\nu_f}{\mu_f} \times |z^+ - z_t| \ .$$

$$\textit{where } L = \frac{\sigma_{\max}(X_{A(z_t)})}{\sigma^2_{\min}(X_{A(z_t)})} + \sup_{z \in [z^+, z_t]} \frac{\sigma_{\max}(X_{A(z)})}{\sigma^2_{\min}(X_{A(z)})},$$

and z_t is the prior kink of z^+.

Proof. To analyze $\|\tilde{\beta}(z^+) - \beta^\star(z^+)\|_2$, we will use the definition from our algorithm that

$$\tilde{\beta}(z^+) = \hat{\beta}(z_t) + \hat{\beta}'(z_t)(z^+ - z_t).$$

Here, z_t is the last point at which we ran our primal corrector. Using this, we can decompose the error as the follows:

$$\begin{aligned}
\left\|\tilde{\beta}(z^+) - \beta^\star(z^+)\right\|_2 &= \left\|\hat{\beta}(z_t) + \hat{\beta}'(z_t)(z^+ - z_t) - \beta^\star(z^+)\right\|_2 \\
&= \left\|\hat{\beta}(z_t) + \hat{\beta}'(z_t)(z^+ - z_t) - \beta^\star(z^+) + \beta^\star(z_t) - \beta^\star(z_t)\right\|_2 \\
&= \left\|\hat{\beta}(z_t) - \beta^\star(z_t) + \int_{z_t}^{z^+} \left[\hat{\beta}'(z_t) - \frac{\partial \beta^\star(z)}{\partial z}\right] dz\right\|_2 \\
&\leq \left\|\hat{\beta}(z_t) - \beta^\star(z_t)\right\|_2 + \sup_{z \in [z^+, z_t]} \left\|\hat{\beta}'(z_t) - \frac{\partial \beta^\star(z)}{\partial z}\right\|_2 |z^+ - z_t|
\end{aligned}$$

Here, the third equality comes from the fact that $\beta^*(z^+) - \beta^*(z_t) = \int_{z_t}^{z^+} \frac{\partial \beta^\star(z)}{\partial z} dz$.

Now, from the Triangular Inequality, we have

$$\begin{aligned}
\sup_{z \in [z^+, z_t]} \left\|\hat{\beta}'(z_t) - \frac{\partial \beta^\star(z)}{\partial z}\right\|_2 &\leq \|\hat{\beta}'(z_t)\| + \sup_{z \in [z^+, z_t]} \left\|\frac{\partial \beta^\star(z)}{\partial z}\right\|_2 \\
&\leq \left[\frac{\sigma_{\max}(X_{A(z_t)})}{\sigma^2_{\min}(X_{A(z_t)})} + \sup_{z \in [z^+, z_t]} \frac{\sigma_{\max}(X_{A(z)})}{\sigma^2_{\min}(X_{A(z)})}\right] \times \frac{\nu_f}{\mu_f}
\end{aligned}$$

Here, the second inequality comes from Lemma D.1.

Now, if point z^+ is such that $z^+ \in (z_{t+1}, z_t]$, the active sets at point z^+ and z_t are constant. Then, we can simplify.

$$\left\|\tilde{\beta}(z^+) - \beta^\star(z^+)\right\|_2 \leq \epsilon_{\text{tol}} + \frac{2\,\sigma_{\max}(X_{A(z_t)})}{\sigma^2_{\min}(X_{A(z_t)})} \times \frac{\nu_f}{\mu_f} \times |z^+ - z_t| \ .$$

Note that if the candidate $z^+ = z_t$ is exactly a kink, the right-most term is zero. It only remains the corrector error. If it is not the case that z^+ and z_t have the same active set, we have

$$\left\|\tilde{\beta}(z^+) - \beta^\star(z^+)\right\|_2 \leq \epsilon_{\text{tol}} + L \times \frac{\nu_f}{\mu_f} \times |z^+ - z_t|$$

where

$$L := \left[\frac{\sigma_{\max}(X_{A(z_t)})}{\sigma^2_{\min}(X_{A(z_t)})} + \sup_{z \in [z^+, z_t]} \frac{\sigma_{\max}(X_{A(z)})}{\sigma^2_{\min}(X_{A(z)})}\right] \tag{15}$$

□

Theorem 5.2. *The estimation error is upper bounded by*

$$\left\|\partial_2 f \circ \zeta(z^+) - \partial_2 f \circ \hat{\zeta}(z^+)\right\|_2 \leq K \left[\epsilon_{\text{tol}} + L\frac{\nu_f}{\mu_f}|z^+ - z_t|\right]$$

where $K = \nu_f \times \sigma_{\max}(X_A)$.

Proof. Let us define, for $t \in [0,1]$, the function

$$\phi(t) = \partial_2 f(\hat{\zeta}(z^+) + t(\zeta(z^+) - \hat{\zeta}(z^+))) \ .$$

We have from the fundamental theorem of calculus, $\phi(1) - \phi(0) = \int_0^1 \frac{\partial\phi(t)}{\partial t}dt$ where

$$\phi(1) - \phi(0) = \partial_2 f(\zeta(z^+)) - \partial_2 f(\hat{\zeta}(z^+))$$

and

$$\frac{\partial\phi(t)}{\partial t} = \partial_{2,1} f(\hat{\zeta}(z^+) + t(\zeta(z^+) - \hat{\zeta}(z^+)))^\top [\zeta(z^+) - \hat{\zeta}(z^+)]_1 + \partial_{2,2} f(\hat{\zeta}(z^+) + t(\zeta(z^+) - \hat{\zeta}(z^+)))^\top [\zeta(z^+) - \hat{\zeta}(z^+)]_2 \ .$$

We remind the reader that, by definition, we have

$$\begin{aligned}
[\zeta(z^+) - \hat{\zeta}(z^+)]_1 &= y(z^+) - y(z^+) = \mathbf{0} \\
[\zeta(z^+) - \hat{\zeta}(z^+)]_2 &= y^\star(z^+) - \hat{y}(z^+)
\end{aligned}$$

and deduce that

$$\begin{aligned}
\|\partial_2 f \circ \zeta(z^+) - \partial_2 f \circ \hat{\zeta}(z^+)\|_2 &= \|\phi(1) - \phi(0)\|_2 \\
&= \left\|\int_0^1 \frac{\partial\phi(t)}{\partial t}dt\right\|_2 \\
&= \left\|\int_0^1 \partial_{2,2} f(\hat{\zeta}(z^+) + t(\zeta(z^+) - \hat{\zeta}(z^+)))^\top [\zeta(z^+) - \hat{\zeta}(z^+)]_2 dt\right\|_2 \\
&\leq \sup_{t\in[0,1]} \|\partial_{2,2} f(\hat{\zeta}(z^+) + t(\zeta(z^+) - \hat{\zeta}(z^+)))\|_2 \|y^\star(z^+) - \hat{y}(z^+)\|_2 \\
&\leq \nu_f \times \|X_A\beta_A^\star(z^+) - X_A\hat{\beta}_A(z^+)\|_2 \\
&\leq \nu_f \times \sigma_{\max}(X_A) \times \|\beta_A^\star(z^+) - \hat{\beta}_A(z^+)\|_2 \\
&\leq \nu_f \times \sigma_{\max}(X_A) \times \left[\epsilon_{\text{tol}} + L \times \frac{\nu_f}{\mu_f} \times |z^+ - z_t|\right] \ .
\end{aligned}$$

Here, the fourth inequality comes from our Lemma 5.1 and the final inequality comes from Theorem 5.1. □