Regression models using the LINEX loss to predict lower bounds for the number of points for approximating planar contour shapes

J. M. Thilini Jayasinghe\textsuperscript{a}, Leif Ellingson\textsuperscript{a} and Chalani Prematilake\textsuperscript{b}

\textsuperscript{a}Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX, USA; \textsuperscript{b}Department of Mathematics, East Carolina University, Greenville, NC, USA

\section*{ARTICLE HISTORY}
Received 15 June 2021
Accepted 19 September 2021

\section*{KEYWORDS}
Linear regression; LINEX loss function; lower bound prediction; planar contours; shape analysis

\section*{2020 MATHEMATICS SUBJECT CLASSIFICATIONS}
62H35; 62J05; 62J20

\section{1. Introduction}

With the technological developments in the digital age, digital images are constantly created, thus producing large amounts of data to be analyzed by both researchers and computers. These images arise from and are used in a variety of fields, including medical imaging, computer vision, biometrics, meteorology, and astronomy, among many others. While there are many characteristics of images that researchers may seek to understand and analyze, common features are the shapes of outlines of objects within the images. The study of these objects has developed considerably over the past 35 years.
Following from Kendall [10], the similarity shape, often referred to simply as shape, of an object is the geometrical information that remains in a configuration when location, scale, and rotation transformations are filtered out. In the early years of statistical shape analysis, due to technological limitations, researchers focused on analyzing shapes of finite dimensional configurations called \( k \)-ads, where each of the small number of \( k \) points represented the coordinates of a landmark that was carefully chosen by subject matter experts to represent a feature of particular interest. As computing technology improved, researchers began filling in more information along outlines of configurations by including pseudolandmarks, that were often equally spaced points between the landmarks, in the \( k \)-ads, thus resulting in higher-dimensional data objects. Details about landmark selections and pseudolandmarks can be found in Dryden and Mardia [5]. Within the past 15 years, researchers have largely shifted their focus to the study of shapes of outlines of objects, which we can view as contours in the plane, as continuous objects. Many of the models for shapes of these objects are described in [6,17].

Regardless of the choice of shape model used, the contours, while infinite dimensional in theory, must be discretized because they are not analytical functions. Despite this fact, a vast majority of the literature has treated this as a computational aside instead of as an integral part of the data analysis process. Indeed, researchers typically choose an arbitrary number of points \( k \), such as multiples of 50, to use to represent the contour without considering whether information is lost by doing so. In recent years, though, researchers have begun to look at the problem of how many points along the contours should be sampled to adequately approximate the contour while simultaneously aiming to keep the dimensionality low enough to prevent computational costs from growing too high. Ellingson et al. [6] described polygon approximations of contours and introduced criteria for describing adequacy of contour approximation. Prematilake and Ellingson [16] then expanded on this approach by evaluating the quality of approximations made using the approximation criteria using two different methods for sampling points along the contours. They then used linear regression models using ordinary least squares (OLS) to predict the lower bounds for the number of sampling points needed to approximate curves at given levels of approximation error. Strait et al. [18] solved a related problem of detecting a relatively small number of key landmarks of mathematical distinction among the sampling points along contours, but did not discuss how many sampling points should be selected to start with.

In this paper, we seek to improve upon the methodology and results of Prematilake and Ellingson [16] in several key areas. First, Prematilake and Ellingson [16] considers only two methods for obtaining sampling points, both of which rely on sampling at unit intervals with respect to either length or curvature and are quite restrictive as a result. We propose a third approximation scheme based on the approximation methods of Latecki and Lakamper [12] that can more effectively approximate shapes of contours with lower numbers of sampling points by taking both length between sampling points and the angle between successive points into consideration. Secondly, we have updated the approximation criteria for selecting the number of sampling points to ensure that the approximation error cannot unexpectedly rise above the threshold for \( k' > k \), the number of sampling points chosen.

Third, and most importantly, since the response variable for the regression model is the lower bound for the number of sampling points needed, OLS regression models are
improper to use. This is because the least squares criterion places equal weight on underestimating and over-estimating the response variable by identical amounts, but it is far worse to underestimate a lower bound than overestimate it. To correct for this issue, we propose using an asymmetric loss function to fit the regression model to place a higher penalty on underestimation of a certain amount than on overestimation of the same amount.

A reasonable candidate we found to do this is the LINEX loss function, which was introduced by Varian [19], and is defined as \( L(\Delta) = b[\exp(a\Delta) - a\Delta - 1] \), where \( \Delta = \hat{\theta} - \theta \). Here \( a \) and \( b \) are constant values that tune the loss function to obtain desired properties. The value of \( b \) changes the scale of the LINEX loss function and \( a \) works as the shape parameter of the loss function. As such, \( a \) determines how the loss function behaves with respect to controlling overestimation and underestimation. When \( a < 0 \), the loss rises exponentially for \( \Delta < 0 \) and almost linearly for \( \Delta > 0 \) and vice versa for \( a > 0 \). When \( a \) is close to 0, the LINEX loss function is almost symmetric and behaves similarly to the squared error loss function. Because \( b \) simply changes the scale of the entire loss function, it does not play a role in changing parameter estimates, so we will simply set \( b = 1 \) and consider the loss function as depending on a single tuning parameter.

Zellner [21] discussed more properties of LINEX loss function and suggested a Bayesian estimate for a linear combination of regression coefficients under LINEX loss by using a diffuse prior for \( \beta \). However, this method is not explored in detail and requires a normality assumption for errors, which may not be practical, especially in cases where the response is a lower or upper bound. Giles and Giles [7] discusses the fact that underestimation of that variance parameter has more severe consequences than overestimation and then used the LINEX loss function to estimate variance of two-sample models. Cain and Janssen [2] suggested a Bayesian approach that used three asymmetric loss functions, including the LINEX loss function, in real estate price prediction. Akdeniz [1] used the LINEX loss to compare risks of generalized Liu and ordinary least squares estimators. Wan [20], Chang and Hung [3], Hoque et al. [9], Hoque and Hossain [8], Chaturvedi et al. [4], Parsian [14], and Ohtani [13] used the LINEX loss for other problems.

One difficulty in using the LINEX loss function for fitting the regression models is in selecting the value for the shape parameter. We propose a data-driven diagnostic for choosing the value of this parameter that simplifies the process to one that is akin to selecting the number of principal components. An additional change from OLS regression is that residuals are not be the most appropriate quantities to analyze by themselves to evaluate how well models fit observed data. As such, we also present methods for analyzing the cost of each observation with respect to the LINEX loss function.

The remainder of this paper is structured as follows. In Section 2, we describe why it is necessary to determine lower bounds for the number of sampling points needed to approximate shapes of planar contours, methods with which we can select sampling points mathematically, and criteria we can use to judge approximation adequacy. In Section 3, we describe the data set and regression models we use throughout the study and the methodology we use to both fit and analyze the models using the LINEX loss function. Then, in Section 4, we present detailed results of our analysis. We conclude the main text of the paper in Section 5 with a discussion of our problem, methods, and results. In the Supplementary Information, we provide additional figures that illustrate our methods for data beyond what we can reasonably show in the main body of the text.
2. Approximating contours and their shapes

In this section, we describe the details of the overall problem. First, we describe contours, their digital representations and the shapes of both mathematically. We will then discuss methods for selecting sampling points to reduce dimensionality of the data. Following that, we will define methods for evaluating the quality of the approximations compared to the original data.

2.1. Shapes of contours and their digital representations

Following from Ellingson et al. [6], a contour \( \tilde{\gamma} \) is the range of a piecewise differentiable function \( \gamma \) that is parametrized by arclength, where \( \gamma : [0, L] \to \mathbb{C} \), such that \( \gamma(0) = \gamma(L) \) and \( \gamma \) is one-to-one on \([0, L)\). This last restriction ensures that the contour \( \tilde{\gamma} \) is non-self-intersecting. For simplicity, we will subsequently identify the contour with the function \( \gamma \). Following from Kendall [10], the direct similarity shape \( [\gamma] \) consists of all contours differing from \( \gamma \) by only rotation, translation, and/or scale transformations. As such, the shape of \( \gamma \) can be expressed symbolically as \( [\gamma] = \{ \lambda e^{i\theta} \gamma_0 : -\pi < \theta \leq \pi, \lambda > 0 \} \), where \( \lambda \) represents a scaling transformation, \( \theta \) denotes the angle of rotation, and \( \gamma_0 \) is the centered contour, having a center of mass at the origin. To quantify the difference between two shapes, Ellingson et al. [6] proposed using the following distance between \([\gamma_1] \) and \([\gamma_2] \):

\[
\rho^2([\gamma_1], [\gamma_2]) = \text{Tr}\left((j([\gamma_1]) - j([\gamma_2])) \otimes (j([\gamma_1]) - j([\gamma_2]))\right),
\]

where \( j([\gamma]) = \frac{1}{\|\gamma\|} \gamma \otimes \gamma_0 \).

In practice, a sample of \( n \) contours is obtained from digital images, so the digital representation of contour \( \gamma_s \) actually consists of \( K_s \) pixels, for \( s = 1, \ldots, n \) that make up the outline of an object from an image. This immediately leads to difficulties since, in general, \( K_s \neq K_{s'} \) for \( s \neq s' \), so each \( \gamma_s \) must be resampled in some fashion at the same number \( k \) of sampling times to have the same number of sampling points \( k \). This is necessary to ensure that the distance between shapes is well-defined.

As a result, the actual data object that is analyzed is not the original contour \( \gamma \), but the polygon \( z \), which is called a \( k \)-gon that is formed by the linear interpolation of the points between successive sampling points. The \( k \)-gon \( z \) can be expressed analytically under arc-length parameterization, for \( s \in (0, 1) \), as

\[
z(sL_k) = \begin{cases} 
(t_2 - sL_k)z(0) + sL_kz(t_2) & 0 < sL_k \leq t_2 \\
(t_j - sL_k)z(t_{j-1}) + (sL_k - t_{j-1})z(t_j) & t_{j-1} < sL_k \leq t_j \\
(L_k - sL_k)z(t_k) + (sL_k - t_k)z(0) & t_k < sL_k < L_k 
\end{cases}
\]

for \( j = 3, \ldots, k \). Here, \( z(t_j) \) is the \( j \)th ordered vertex, where \( t_j \in [0, L_k) \), and \( z(t_1) = z(0) = z(L_k) \), and \( L_k \) denotes the length of the \( k \)-gon,

\[
L_k = \sum_{j=2}^{k+1} \|z(t_j) - z(t_{j-1})\|,
\]

where \( z(t_{k+1}) = z(0) \). Since these \( k \)-gons can be evaluated at any time, the space of these approximations is dense in the space of contours, so all of the shape theory applies to them, as well.
2.2. Selecting sampling points

In the above discussion, we skipped over the details of how to select sampling points to use for constructing \( k \)-gons, but this is a non-trivial problem that we now turn our attention to. To begin with, we assume a digital contour consists of \( K \) ordered pixels that trace the contour when the adjacent points are connected. However, not all \( K \) of the points may contain relevant information about the shape of the contour. As such, we want to use a \( k \)-gon, for some \( 4 \leq k < K \), to approximate the original \( K \)-gon. The key to this, then, is determining an appropriate value of \( k \) so that the \( k \)-gon closely resembles the original contour.

To illustrate the importance of the choice for \( k \), Figure 1 shows polygon approximations with various values of \( k \) for the contour of a hand gesture.

Beyond choosing the number of sampling points, though, we also need to determine a procedure for selecting the sampling points. We first describe the methods considered in Prematilake and Ellingson [16]. Parametrizing the contour by arc-length, which amounts to traveling along the length of the contour at constant speed, provides the classical method to select sampling points, as we simply choose points equally spaced around the length of the contour. The second method used by Prematilake and Ellingson [16] is based on a curvature parametrization, where we traverse a constant amount of curvature per change in interval rather than a constant length.

While both of the above approaches allow us to select sampling points in a well-defined manner, both methods are extremely rigid in their methodology. The equally spaced approach is most commonly used method that researchers use to discretize contours, but, unfortunately, it does not take features of the contours into consideration and results in many sampling points being placed along regions of curves that may be relatively flat. The equal curvature approach addresses this somewhat by reducing the number of sampling points along straight portions of contours and instead placing more of them in regions with high degrees of curvature. Unfortunately, though, it can be overly sensitive to small details of contours because those often result in extreme values of curvature in small regions compared to more gently curving of areas which space out the sampling points further. An additional difficulty with these methods is that if it is decided to use \( k + 1 \) sampling points rather than \( k \), all sampling points besides the starting point will change.

To address these drawbacks of the above methods, we are also considering a third alternative approach, which we will refer to as the Latecki parametrization as it is an approximation scheme introduced in Latecki and Lakämper [12] for discrete contour evolution. The idea of this approach is as follows. In a given \( K \)-gon, each consecutive pair of points \( \{z_i, z_{i+1}\} \) forms a line segment, which is an edge of the polygon. Let \( s_1 = \{z_{i-1}, z_i\} \)
and $s_2 = \{z_i, z_{i+1}\}$ be two consecutive line segments. The cost of removing $z_i$ and replacing the arc, $s_1 \cup s_2$ by the line segment connecting $\{z_{i-1}, z_{i+1}\}$ can be calculated using $K(s_1, s_2) = \frac{\beta(s_1, s_2)l(s_1)l(s_2)}{l(s_1)+l(s_2)}$, where $l(s)$ is length of the line segment $s$ and $\beta(s_1, s_2)$ is the turning angle in the decomposition of arc $s_1 \cup s_2$. To remove a sampling point to go from a $K$-gon to a $(K - 1)$-gon, the vertex (sampling point) having the smallest $K(s_1, s_2)$ will be removed. This process is completed when only $k$ sampling points remain.

As with the first two parametrizations, the Latecki method allows us to construct a $k$-gon that can be directly compared with the original $K$-gon. However, this approach is completely data-driven since it removes the sampling points with the smallest cost of removal and does not impose any requirement for the $k$ sampling points to be spaced according to any criterion. Furthermore, the set of $k$ sampling points is a subset of the original $K$ sampling points, which means that if we were to decrease the number of points from $k$ to $k-1$, only one vertex would be different.

As an illustration of these methods, consider the example depicted in Figure 2. The 50 sampling points are selected according to the arc-length, equal curvature, and Latecki parametrizations, respectively, from left to right. Note that the leftmost $k$-gon misses a number of features, such as around the tip of the finger and the knuckles. This is in contrast to the figure in the middle which, though sampling points are clustered together where curvature is extreme, the $k$-gon better captures the details the equally-spaced sampling points do not. The right-most figure appears to show even fewer differences between the $k$-gon and the original contour. This suggests that the Latecki parametrization will typically result in better approximations to the original contours than the other two methods will with the same number of sampling points. Even though the first two methods may require more sampling points to be used, they are more widely used among the shape analysis community. As such, it is important for us to consider all three further in the analysis. Additionally, we must also carefully mathematically define the quality of an approximation.

2.3. Evaluating adequacy of approximations

Since a $k$-gon will not typically not fully coincide with the original contour, we must quantify differences between the contour and $k$-gon using various characteristics so that we can balance the loss of information due to the error in approximating the contour and the increased computational cost for the subsequent statistical analyzes associated with higher
values of $k$. To do this, we focused on choosing a value for $k$ by specifying a maximal level of approximation error in terms of the two characteristics. For our first criterion, we choose $k$ by fixing an upper bound for the relative difference of the lengths. Because of the linear interpolations used in obtaining the $k$-gons, for $k' > k$, $L_{k'} \geq L_k$, so the relative error with respect to $K$ will always be positive. To choose a lower bound for $k$ with respect to the relative error in lengths, we use the following criterion.

Definition 2.1: For a given $K$-gon and a prespecified, small error threshold, $0 < E < 1$, there exists a value $4 \leq k^* < K$ such that $\frac{L_K - L_{k^*}}{L_K} > E$ and $\frac{L_K - L_{k'}}{L_K} \leq E$ for all $k' > k^*$. $k_L$ is the smallest number of points needed to guarantee an approximation level of at most $E$ and is equal to $k^* + 1$.

Our second characteristic is the distance between the shapes of the original contour and the approximations in the shape space. While the relative difference in lengths in Definition 2.1 is bounded above by 1, the maximal distance between two shapes from (1) is $\sqrt{2}$, so we can normalize distances between contours with respect to this upper bound. To choose a lower bound for $k$ using the distances between shapes of contours, we use the following criterion.

Definition 2.2: For a given observation with $K$ points and a prespecified, small error threshold, $0 < E < 1$, there exists a value $4 \leq k^* < K$ such that $\rho(z_{k^*}, z_K) > \sqrt{2}E$ and $\rho(z_{k'}, z_K) \leq \sqrt{2}E$ for all $k' > k^*$. $k_D$ is the smallest number of points needed to guarantee an approximation level of at most $E$ and is equal to $k^* + 1$.

These criteria improve upon those of Prematilake and Ellingson [16], for which, in some cases, it was possible for the approximation level to be above $E$ for some $k'$ greater than the lower bound for $k$. This is because the approximation error as a function of $k$ is not monotonically non-increasing as a function of $k$ for the $k$-gons formed using the arclength and equal curvature parametrizations since all $k$ sampling points change for each increase of an additional point. For the Latecki parametrization, the updated criteria are identical to those of Prematilake and Ellingson [16] because these functions are monotonically non-increasing for this parametrization.

3. Data and methods

Unfortunately, finding the exact lower bound for $k$ for a dataset of contours under any given parametrization and approximation criterion can be quite time consuming. As a result, we wish to construct regression models using easily computable features of the original contours to predict the lower bounds for $k$ for a specified error threshold $E$. In this section, we will first describe the dataset that we utilized and then describe the methods that we used.

3.1. Data and regression model

The Kimia [11] dataset contains 238 observations that consist of multiple observations belonging to different categories of objects. There are 10 contours of dogs, a total of 120
fish, with 20 each of 6 subclasses, a total of 20 hand gestures, with 5 observations for each of the 4 hand gestures, and 88 contours of pears. Representatives of each of these categories and subclasses are shown in Figure 9 of Prematilake and Ellingson [16].

Since it showed promise in achieving the general goal of predicting lower bounds for \( k \), we use the same multiple linear regression model of Prematilake and Ellingson [16], which uses 3 predictor variables, which are defined as follows. The first predictor is the total absolute curvature of the original contour. This is denoted by \(|\kappa|\) and \(|\kappa| = \sum_{i=1}^{K} |\kappa_i|\), where \(|\kappa_i| = \frac{|x'^2+y'^2|}{(x^2+y^2)^{3/2}}\), for \( i = 1, \ldots, K \) and \( x \) and \( y \) respectively represent the real and imaginary coordinates at the \( i \)th infinitesimal time point. The absolute value of the curvature at each time must be used instead of the curvature, itself, because the total curvature for any closed curve is equal to 0. The second predictor is the length \( L_k \) of the original contour \( L_k = \sum_{j=2}^{K+1} \| z(t_j) - z(t_{j-1}) \| \), where \( z(t_{K+1}) = z(0) \). The third predictor is \( n_k \), the number of times the curvature’s sign changes along the contour. This is used as a surrogate for the number of ‘features’ the contour has. The response variable for each model is \( k \), the lower bound for the number of sampling points needed to approximate the contour under a given parametrization at threshold \( E \) for a chosen approximation criterion, as defined in Section 2.3. For the purposes of our study, we fixed \( E = 0.005 \). The resulting regression equation for predicting \( k \) is

\[
k = \beta_0 + \beta_1 |\kappa| + \beta_2 L_k + \beta_3 n_k + \epsilon,
\]

where we assume only that \( \epsilon \) are random variables that are independent across observations and have zero-mean. Please note that these assumptions are necessary for the cost function in the following subsection to be appropriate for fitting the model. Further distributional assumptions may be needed in general situations where variable selection is necessary and parameter inference is of primary concern.

We will now introduce some notation that will be used throughout the remainder of this paper. \( k_{LA}, k_{DA}, k_{LC}, k_{DC}, k_{LN}, \) and \( k_{DN} \) denote the models for different criteria and parametrizations. The first subscript represents the decision criterion: \( L \) for length and \( D \) for distance. The second subscript represents the parameterization: \( A \) for arclength, \( C \) for curvature, and \( N \) for Latecki.

### 3.2. Regression using the LINEX loss function

While parameter estimation is easy under the traditional squared-error loss function, it is not particularly appropriate to our problem because that loss function treats underestimation and overestimation of the response variable equally. However, since \( k \) is a lower bound, it is far worse to underpredict it than overpredict it. As such, it is preferable to use an asymmetric loss function to fit the regression model. A reasonable choice for this is the one-parameter LINEX loss function that was introduced by Varian [19], which results in the following cost function:

\[
L_a(\hat{k}, k) = \sum_{i=1}^{n} \left[ \exp \left( a \left( \hat{k}_i - k_i \right) \right) - a \left( \hat{k}_i - k_i \right) - 1 \right],
\]

where \( \hat{k}_i \) denotes the predicted value of \( k_i \) and \( a \) is the tuning parameter. Using a negative value for \( a \) places an approximately exponential loss on underestimation of \( k_i \) while
imposing just an approximately linear loss on overestimation. Note that we are using the one-parameter LINEX loss function as opposed to the two-parameter version mentioned in the introduction since the parameter $b$ simply scales the overall value of the loss, which does not impact parameter estimation. For our scenario, it is also worse to underestimate $k_i$ by some amount if $k_i$ is small compared to underestimating $k_j > k_i$. As such, it also makes sense to augment the LINEX loss by using relative, rather than absolute, residuals. This yields the cost function we use for fitting our regression model:

$$L_a(\hat{k}, k) = \sum_{i=1}^{n} \left[ \exp \left( a \left( \frac{\hat{k}_i}{k_i} - 1 \right) \right) - a \left( \frac{\hat{k}_i}{k_i} - 1 \right) - 1 \right]$$

For a fixed value of $a$, local optima for the loss function can be calculated numerically using standard multivariate optimization functions. We used least squares estimates for the initial conditions of the parameters and used the quasi-Newton method to estimate the regression parameters.

### 3.3. Choosing the LINEX tuning parameter

A primary challenge for fitting regression models using the LINEX loss function is in selecting the value for the parameter $a$ if a value for it is not known a priori. Following from Cain and Janssen [2], an intuitive approach is to fit the regression model using the cost function in (3) over a fine grid of values of $a$ and then select the values of $a$ that produce the minimal cost. Doing so for our data resulted in the plot shown in Figure 3(a).

As with the application in Cain and Janssen [2], this would suggest that $a$ should be as close to 0 as possible. Since a value of $a = 0$ coincides approximately with least squares, this makes it seem initially like there is not much to gain by using the LINEX loss. However, we propose that it is not appropriate, in general, to use the LINEX cost function to choose the value of $a$. This is because the values of the function for different values of $a$ will differ in orders of magnitude from each other due to the exponential terms in the cost function. Indeed, if $a_2 > a_1$, in order to keep the cost functions $L_{a_1}$ and $L_{a_2}$ at the same magnitude,
the $r_i = \frac{\hat{\theta}}{k} - 1$ obtained with $a_2$ must be approximately equal to or less than $\frac{a_1}{a_2}$ times the relative residuals obtained with $a_1$. As this may not easily be achieved even if increasing $a$ truly does result in a considerable reduction in the number of and/or magnitude of relative residuals due to underestimation as desired, minimizing the cost function to choose $a$ would typically result in choosing a value that penalizes residuals from underestimation too lightly.

Because our goal is to reduce both the number and the magnitude of (relative) residuals from underestimation, it is sensible that the criterion for choosing $a$ should take this information into account. However, it would be trivially easy to reduce the number of these residuals to 0 by simply inflating the intercept coefficient so that all values of $k$ are underestimated. As such, we also want to penalize choices of $a$ that result in the proportion and/or magnitude of residuals due to overestimation from being too high. To balance all of these considerations into account, we constructed a quantity that we will refer to as the RMSE power ratio (PR),

$$PR = (RMSE^+ \times R^+)^{R^-} \times (RMSE^- \times R^-)^{R^+}$$

where $R^+$ is the ratio of positive residuals, $RMSE^+$ gives the Root mean squared error (RMSE) of positive residuals, $R^-$ is the ratio of negative residuals, $RMSE^-$ gives the RMSE of negative residuals. To select $a$, we first compute the PR over a fine grid of values for $a$. Rather than trying to minimize this quantity, though, which will equal 0 if either $R^+$ or $R^-$ is 0, we look for an elbow in the graph, similarly to what is commonly done with scree plots for principal component analysis, since this marks the location where increasing the absolute value of $a$ results in less impact on reducing $R^+$ and the sum of squares of the positive residuals. This procedure would work similarly if overestimation is a bigger problem, but we would look for the elbow over the domain of negative values of $a$.

The plot of PR as a function of $a$ for the $k_{LA}$ model is shown in Figure 3(b). As often happens with scree plots, the large bend is not sharp, so the choice of $a$ is not immediately apparent, but the first flattening occurs at around $a = -32$, so we will use that. While an argument could be made that further flattening in the curve occurs at around $a = -40$, we also wish to keep the tuning parameter as small as possible while achieving our goals. As a result, we choose $a = -32$ for this model.

We performed this procedure similarly for the other regression models. The plots are shown in the Supplementary Information. For the $k_{LC}, k_{LN}$, and $k_{DN}$ models, we chose $a = -45$. For the $k_{DC}$ and $k_{DA}$ models, respectively, we set $a = -28$ and $a = -21$.

### 3.4. Evaluating the cost of each observation

In least-squares regression, residuals play a key role in both aiding our understanding of model fit and checking distributional assumptions. The latter use is inherently tied to the fact that residuals are used in practice as estimates of the model errors. However, LINEX residuals may not provide reasonable estimates for the errors of the original model because, as shown in Figure 4, the distribution of LINEX residuals may differ considerably from the distribution of LS residuals. As such, the LINEX residuals may best be used to help assess model fit rather than for checking model assumptions.
Figure 4. Histograms of residuals for $k_{DC}$ model using the specified loss function. (a) Least squares and (b) LINEX.

In practice, however, we know that for least-squares regression, it is the squared residuals that quantify how much each observation contributes to the total value of the minimized cost function, which is the sum of squared residuals. Thus, dividing the squared residuals by the sum of squares provides the relative cost of each observation compared to the whole sample. These quantities allow for quick diagnostics of which observations are explained the least by the model. While we should still directly examine residuals for some aspects of LINEX regression, understanding the contributions of each observation to the cost function requires slightly altered definitions of cost and relative cost.

The cost of observation $j$ is $C_j = \exp(a(\hat{k}_j - 1)) - a(\hat{k}_j - 1) - 1$ and the average cost for the entire sample is $\overline{C} = \frac{1}{n} \sum_{i=1}^{n} C_j$. The relative cost for observation $j$ is, therefore, $RC_j = C_j/(n\overline{C})$.

4. Results

We will now present the results for fitting the regression models using the LINEX cost function. To begin with, we will summarize some of the properties of all 6 models and describe differences between them and the comparable properties of the least squares models. After that, we will discuss model validation and analyze the results in-depth. For brevity, we will focus on the details for one model in the main text and then present the results for the other models in the Supplementary Information.

4.1. Model summaries

The fitted model for $k_{LA}$ under the squared error loss function is

$$\hat{k}_{LA} = -20.5694 + 5.4401|\kappa| - 0.0313L_k + 0.0217n_k,$$

while under the LINEX loss function, it becomes

$$\hat{k}_{LA} = -20.5636 + 5.3866|\kappa| - 0.0077L_k - 0.0057n_k.$$
Table 1. Parameter estimates under the LINEX loss and squared error loss functions.

|         | $k_LC$   | $k_{LN}$ | $k_{DA}$ | $k_{DC}$ | $k_{DN}$ |
|---------|----------|----------|----------|----------|----------|
| intercept | 8.5972   | 8.5949   | 5.0556   | -16.8324 | -16.8143 |
| $|k| | 3.7694   | 4.0489   | 1.4255   | 1.6052   | 6.8157   |
| $L_k$   | -0.0267  | -0.0468  | 0.0163   | -0.0457  | 0.0397   |
| $n_k$   | -0.1289  | -0.0253  | -0.1349  | -0.0915  | 0.255    |
| RMSE    | 7.81     | 11.27    | 5.34     | 7.38     | 41.99    |

The intercept of the model under LINEX loss function did not change a lot from least squares, which indicates that, with $a = -32$, the penalty on positive residuals does not simply inflate the intercept, as could be done naively to limit such residuals. Indeed, it is some of the other coefficients that changed more considerably. Most notably, the coefficient for $n_k$ flipped signs. The RMSE for the model under the squared error loss function is 14.9548, but climbs to 22.31 under the LINEX loss function. It is to be expected that RMSE will be higher under LINEX since MSE is minimized under the least squares framework while the LINEX loss trades higher negative residuals for a decrease in the magnitude and number of positive residuals, thus resulting in a higher RMSE. The estimated coefficients and RMSE for the other five models for both loss functions are shown in Table 1. The changes in the values of the estimated coefficients and RMSE follow similar behavior to what we saw with $k_{LA}$.

4.2. Cost analysis for each shape category

For a more detailed understanding of the performance of the models, we examined how well the use of the LINEX loss function achieved our goal of reducing the percentage of positive residuals and what effects doing so had on the root mean square error (RMSE). To do this, we used leave-one-category-out cross validation. More specifically, we broke our data set into 12 groups (of unequal sizes), with each composed of all observations belonging to one shape category, and fit the model using the observations from 11 of the groups as a training set and evaluated the model for the observations from the remaining group. The percentages of positive residuals for each shape category for all 6 models under both LINEX regression and least-squares estimation (LSE) are shown in Table 2. We can see that the percentage of positive residuals was indeed reduced considerably under LINEX loss compared to LSE, often considerably so, especially for the pear group.

There is by necessity, of course, a trade-off between the amount of positive residuals and the RMSE. As such, while we could easily reduce the percentage of positive residuals by naively increasing the intercept coefficient and thus causing a huge increase in RMSE, we want to make sure that our LINEX regression models are not doing the same. The RMSE for all 6 models under both loss functions are shown, by shape category, in Table 3. While the RMSE is, indeed, considerably higher under LINEX for some categories and models, such as Hand 1 for $k_{LC}$ and most categories for $k_{DA}$, it did not change much for some situations, such as the pears for $k_{LN}$ and $k_{DN}$. Furthermore, the RMSE actually decreased under the LINEX loss for some categories, such as Fish 4 using the $k_{DN}$ model. These results, then, show that using the LINEX loss function will not necessarily result in an explosion in RMSE.
Table 2. Percentages of positive residuals under the LINEX and squared error loss functions.

| Shape | $k_{LA}$ | $k_{LC}$ | $k_{LN}$ | $k_{DA}$ | $k_{DC}$ | $k_{DN}$ |
|-------|----------|----------|----------|----------|----------|----------|
|       | LSE      | LINEX    | LSE      | LINEX    | LSE      | LINEX    | LSE      | LINEX    | LSE      | LINEX    | LSE      | LINEX    | LSE      | LINEX    |
| Dog   | 20       | 0        | 80       | 50       | 90       | 70       | 10       | 0        | 40       | 0        | 100      | 40       | |
| Fish 1| 35       | 10       | 30       | 5        | 25       | 10       | 50       | 15       | 70       | 40       | 25       | 5        | |
| Fish 2| 40       | 15       | 20       | 5        | 40       | 15       | 45       | 5        | 35       | 0        | 40       | 15       | |
| Fish 3| 45       | 15       | 45       | 5        | 75       | 10       | 20       | 0        | 30       | 5        | 50       | 10       | |
| Fish 4| 55       | 15       | 55       | 15       | 80       | 25       | 55       | 5        | 30       | 0        | 80       | 55       | |
| Fish 5| 80       | 50       | 75       | 35       | 70       | 25       | 45       | 10       | 80       | 15       | 40       | 0        | |
| Fish 6| 35       | 10       | 65       | 20       | 65       | 30       | 10       | 0        | 15       | 0        | 45       | 0        | |
| Hand 1| 100      | 60       | 0        | 0        | 20       | 0        | 20       | 0        | 100      | 0        | 40       | 20       | |
| Hand 2| 40       | 0        | 20       | 0        | 20       | 0        | 40       | 0        | 40       | 0        | 40       | 0        | |
| Hand 3| 20       | 0        | 40       | 0        | 20       | 0        | 0        | 0        | 20       | 0        | 0        | 0        | |
| Hand 4| 80       | 0        | 80       | 0        | 60       | 0        | 100      | 20       | 0        | 0        | 0        | 0        | |
| Pears | 68       | 11       | 88       | 49       | 53       | 14       | 63       | 17       | 61       | 41       | 60       | 32       | |

Table 3. RMSE under LINEX Loss Function and squared error loss.

| Shape | $k_{LA}$ | $k_{LC}$ | $k_{LN}$ | $k_{DA}$ | $k_{DC}$ | $k_{DN}$ |
|-------|----------|----------|----------|----------|----------|----------|
|       | LSE      | LINEX    | LSE      | LINEX    | LSE      | LINEX    | LSE      | LINEX    | LSE      | LINEX    | LSE      | LINEX    |
| Dog   | 24.62    | 45.08    | 15.46    | 10.33    | 10.97    | 6.18     | 64.39    | 176      | 7.99     | 26.13    | 7.69     | 3.78     |
| Fish 1| 21.02    | 31.53    | 11.60    | 17.66    | 9.90     | 13.91    | 61.86    | 82.90    | 15.39    | 17.44    | 5.91     | 9.33     |
| Fish 2| 12.19    | 19.2     | 7.47     | 12.89    | 5.23     | 7.12     | 35.76    | 64.50    | 7.41     | 13.67    | 3.25     | 5.43     |
| Fish 3| 13.42    | 17.84    | 4.67     | 9.46     | 3.44     | 4.23     | 31.89    | 82.76    | 6.69     | 14.57    | 2.86     | 4.14     |
| Fish 4| 14.14    | 20.95    | 7.97     | 10.77    | 5.37     | 5.49     | 36.79    | 102.55   | 7.38     | 21.99    | 5.32     | 5.56     |
| Fish 5| 20.94    | 16.68    | 12.07    | 10.94    | 5.70     | 7.73     | 64.46    | 123.49   | 13.08    | 13.50    | 3.46     | 7.70     |
| Fish 6| 15.68    | 22.77    | 9.34     | 12.36    | 7.40     | 9.67     | 58.78    | 153.73   | 8.85     | 26.01    | 3.31     | 6.88     |
| Hand 1| 30.90    | 8.41     | 15.65    | 28.49    | 4.97     | 12.32    | 41.36    | 128.72   | 11.60    | 9.58     | 5.43     | 8.19     |
| Hand 2| 22.09    | 20.79    | 9.92     | 15.34    | 3.86     | 6.92     | 16.18    | 99.81    | 10.35    | 19.36    | 2.83     | 5.38     |
| Hand 3| 25.5     | 43.55    | 2.09     | 6.76     | 2.39     | 6.03     | 22.62    | 93.56    | 7.88     | 15.86    | 2.51     | 5.81     |
| Hand 4| 7.63     | 20.79    | 2.50     | 9.32     | 3.49     | 5.2     | 43.41    | 42.02    | 5.24     | 15.89    | 2.40     | 6.65     |
| Pears | 12.38    | 18.29    | 9.15     | 5.86     | 3.99     | 4.98     | 34.50    | 40.69    | 4.76     | 4.87     | 2.93     | 3.32     |

values. Indeed, using the power ratio from Section 3.3 to select $a$ prevents us from setting the tuning parameter in such a way that would result in arbitrarily high RMSE values.

We calculated the average cost for each shape group under the LINEX loss function for all 6 models to see the behavior of relative residuals. These results are shown in Table 4. Most of the shape groups have low average costs except for fish 3 and hand 3 under $k_{LA}$, the pears under $k_{LC}$, fish 2 under $k_{LN}$, pears under $k_{DA}$, fish 5 under $k_{DC}$ and fish 4 under $k_{DN}$. Most notably, the pears when using the $k_{LC}$ model have an extremely high average cost of 22,348.79.

To investigate why pears under $k_{LC}$ have such a high average cost, we examined the relative cost of each pear observation and how these relate to both the residuals and relative residuals. In Figure 5(a), we see that shape 153 has a relative cost of 0.99, so it is the observation responsible for the average cost being so high for this group. This shape, which is shown in Figure 5(b), is relatively simple, as its main feature is the stem, which is visible in the lower left region of the contour. To get a better understanding of why this shape has such a high cost, we examined the residuals in Figure 5(c). We see that the actual lower bound for $k$ for this scenario is 40, but the model predicted a lower bound of roughly 28, so not only is the residual negative, but the magnitude of the relative residual is quite high since the observed value of $k$ is so small, which is exactly the type of case that the relative
Table 4. Average cost under LINEX loss function.

| Shape  | $k_{LA}$ | $k_{LC}$ | $k_{LN}$ | $k_{DA}$ | $k_{DC}$ | $k_{DN}$ |
|--------|----------|----------|----------|----------|----------|----------|
| Dog    | 5.33     | 9.17     | 13.13    | 10.47    | 5.78     | 3.83     |
| Fish 1 | 6.14     | 5.63     | 9.83     | 16.03    | 11.04    | 7.76     |
| Fish 2 | 4.58     | 6.66     | 14.07    | 6.97     | 5.02     | 5.62     |
| Fish 3 | 11.38    | 4.38     | 2.83     | 8.93     | 5.39     | 3.96     |
| Fish 4 | 3.12     | 3.64     | 3.50     | 7.62     | 6.79     | 26.17    |
| Fish 5 | 3.58     | 5.90     | 3.62     | 8.02     | 18.54    | 6.21     |
| Fish 6 | 3.02     | 2.56     | 3.89     | 12.70    | 7.19     | 4.73     |
| Hand 1 | 0.89     | 9.19     | 6.64     | 6.20     | 1.33     | 6.36     |
| Hand 2 | 3.17     | 6.38     | 4.87     | 8.24     | 5.82     | 4.13     |
| Hand 3 | 11.84    | 2.58     | 4.97     | 9.14     | 5.39     | 5.24     |
| Hand 4 | 3.40     | 3.67     | 3.23     | 2.12     | 4.82     | 5.88     |
| Pears  | 7.50     | 22348.79 | 8.63     | 26.81    | 5.90     | 8.21     |

LINEX loss function places a large penalty on. This is in stark contrast to the observation having an observed value of 57 and a predicted value of roughly 89. Even though that prediction is substantially incorrect, since it predicts $k$ to be higher than it actually is, it has a considerably lower relative cost.

4.3. Evaluating predictive performance

To understand how well the LINEX models predict $k$ for new categories of observations, we again utilized leave-one-group out cross validation. Here, though, we want to compare the average cost for each group of $n$ shapes to that of many randomly selected groups of $n$ shapes. Doing so allows us to determine whether the predictive performance for a given group is higher than what would be expected when compared to arbitrarily selected observations that more closely resemble the entire data set.

To do this, we first left out the dogs, each type of fish, each type of hand gesture, and the pears, from the training set and then used them as the testing set one-by-one to obtain the average cost, along with other characteristics, of that group. We then left out 10, 20, 5, and 88 observations, respectively from the training set and evaluated the fit of these, as before. We then repeated this process 100,000 times to obtain the distributions of average cost.

While we did this for all 6 models, we will focus on analyzing the results for the $k_{LA}$ model in the main text and leave the results of the other models for the Supplementary Information, as they are quite similar. A histogram showing the average cost of the testing
The average cost for the group of dogs is shown as a red asterisk. From this graph, we can see that the probability of selecting a random sample with an average cost greater than or equal to that of the dogs is quite high, which suggests that this model performs reasonably well for predicting $k$ for the group of dog shapes.

However, it was unexpected that the distribution is trimodal instead of unimodal, which is what we typically encounter when looking at plots of sums of squares for least-squares regression. Our next goal, then, was to investigate this surprising behavior. After determining that there was no relationship between the cost of the observations and the predictor variables, themselves, we looked at the relationship between the average cost of each group and the maximum value of the relative cost of each testing set’s observations. A scatterplot showing this relationship is shown in Figure 6(b). The intervals of total cost used for the color coding correspond to the smaller peaks in Figure 6(a). Thus, this plot reveals that the unexpected peaks are due to testing sets where one observation accounts for at least 60 percent of the set’s cost.

Upon discovering this, we examined a number of testing sets having high cost and determined that at least one of shapes 62 and 233 appeared in each of these. If we re-label the testing sets based on the presence of these two shapes, we get the scatterplot shown in Figure 6(c). These new labels explain the clustering we saw in Figure 6(b). As such, it is clear that the unusual peaks in Figure 6(a) are explained by the presence of shapes 62 and 233.

We performed the same types of analyzes for the other shape categories for the $k_{LA}$ model, too. Histograms showing the distributions of average cost when the appropriately sized, randomly selected testing sets for comparisons to the average costs of the fish, hand gesture, and pear groups are shown in Figure 7. From Figure 7(a), we can see that fish groups 1 and 3 are not well predicted by the model since the upper tail probabilities are small. Likewise, Figure 7(b) suggests that hand gesture 3 is not well-predicted by the model. The plots suggest that the model was able to predict $k$ well for the other groups of shapes. Using the same procedure as we did for understanding the additional peaks in the histograms for the dog shapes, we determined that some combination of shapes 62 and 233...
Figure 7. The distribution of average cost for testing sets consisting of (a) 20, (b) 5, and (c) 88 randomly selected shapes. The average costs for the various groups of shapes are shown as red symbols.

Table 5. Errors of shapes with high cost for each model and the associated error with respect to the appropriate criterion.

| Shape | Method | kLA | kLC | kLN | kDA | kDC | kDN |
|-------|--------|-----|-----|-----|-----|-----|-----|
| Error |        | 0.0074 | 0.0066 | 0.0050 | 0.0053 | 0.0054 | 0.0053 | 0.008 | 0.0032 | 0.0032 | 0.0053 | 0.0046 |

once again are responsible for the peaks in these histograms. This tells us that shapes 62 and 233, specifically, are not well explained by the $k_{LA}$ model.

We also followed the same procedure described above for the other five models. While we leave the plots displaying the detailed results of these analyses to the Supplementary Information, we will provide a brief summary of the results here. For $k_{LC}$, the dog and hand gesture 1 groups are not well predicted by the model and shapes 95 and 153, in particular, have high costs. For the $k_{LN}$ model, the dogs, fish group 1, and fish group 2 are not well predicted and shapes 45 and 215 have particularly high costs. Unfortunately, most of the shape groups were not predicted well under the $k_{DA}$ model when similar shapes were not included in it, suggesting that this model may not be so useful for predicting for new categories of shapes. However, shapes 194 and 236, specifically, had particularly high costs. The $k_{DC}$ model does not appear to predict the dogs and all groups of fish well, but shapes 17 and 95 tended to have especially high costs associated with them. For the $k_{DN}$ model, fish group 1 and the pears are not explained well, but only shape 90 tends to have an exceptionally high cost.

4.4. Practical implications of high cost

While the above results allow us to understand how well the models perform for predicting $k$, they do not inform us directly about the associated error in approximating the original contours with the predicted number of sampling points under the appropriate parametrization. To better understand this, we further examined the shapes that had exceptionally high costs under any of the LINEX regression models. For each shape, we calculated the appropriate measurement of error for the parametrization associated with the model in which the shape had high cost. These results are shown in Table 5.

We see that, in all cases, the approximation error arising from using the predicted values of $k$ are near to the chosen threshold of 0.005. Note the interesting behavior for shapes 13
and 236, which have errors that are actually below the threshold of 0.005. To see why this is the case, we plot approximation error versus \( k \) for those two contours in Figure 8. From these plots, we see that the errors are not decreasing monotonically as \( k \) increases and, since Definitions 1 and 2 require us to choose \( k \) so that the error is never above the threshold for \( k' > k \), certain values of \( k \) below the lower bound can actually result in smaller amounts of error than specified.

To better understand the approximation error associated with these bounds, we plotted the original contours alongside these approximations to visualize how small the errors are. In Figure 9, we see that the only portion of the original contour for shape 233 that we can visually distinguish from its approximation is at the tip of the stem. Similar plots for the other shapes from Table 5, which are shown in the Supplementary Information, display similar levels of visual similarity. This suggests that even the cases that had the largest

**Figure 8.** Plot of \( K \) vs. error under \( k_{DA} \).

**Figure 9.** The originally observed contour for shape 233 and the approximation resulting from the lower bound predicted by the \( k_{LA} \) model.
degree of underestimation from the models still result in levels of approximation that are comparable to the desired level.

5. Conclusions and discussion

In this paper, we built upon the work of Prematilake and Ellingson [16] to develop regression models for predicting lower bounds of the number of sampling points necessary to approximate the shapes of planar contours sufficiently well. While the previous study provided helpful models for doing this using easy to calculate features of the original contours, since the response variable is a lower bound, OLS methods are not appropriate for fitting the models since it is far worse to underestimate a lower bound than to overestimate it. To solve this problem, we instead fit the regression models using the LINEX loss function, which allows us to more heavily penalize underestimation. As detailed in Table 2, our goal of limiting the number of under-predictions was easily satisfied using this approach.

However, since the literature involving the LINEX loss function is largely limited to comparing different estimators rather than fitting a model to data, we needed to develop methodology for adapting the loss function to the regression setting. To solve the key problem of determining how large of the penalty parameter $a$ should be, we introduced an RMSE power ratio. By choosing a value of $a$ near the elbow on a graph of this ratio versus $a$, we balance positive and negative residuals in such a way that we do not overly inflate over-predictions for just a small improvement in under-predictions. Additionally, in place of the traditional residual analysis, we introduce a cost analysis to help both evaluate the ability of the model to predict $k$ well for new categories of data in cross validation studies and determine which individual observations contribute to the total cost the most. This allowed us to gain a better understanding of what led to unexpected phenomena, such as multimodality in histograms of the total cost when performing leave-one-group-out cross validation.

The cross validation studies indicated that most of the 6 models we studied worked well to predict lower bounds for $k$ relatively well for new data. This was particularly true for the $k_{LA}$ and $k_{DN}$ models, for which each category of shapes had total costs that were not high compared to the costs associated with randomly selected groups of observations of the same sample size. Furthermore, after identifying the observations with the highest costs under each model, we determined that the amount of under-prediction of $k$ for these observations did not result in the approximation error levels being overly inflated. Indeed, the level of approximation error for these observations was either just slightly above the error threshold or, due to the updated definitions of the lower bounds in this study, even below the error threshold. In all cases, the contours and their approximations are nearly indistinguishable from each other. This suggests that the regression models fit with the LINEX loss function using relative residuals perform extremely well for predicting lower bounds for the number of sampling points needed.

We believe that the LINEX regression models would be useful in other applications where one of either underestimation or overestimation of the response variable would be more problematic, regardless of whether residuals or relative residuals are used in the loss function. However, because we had already chosen our models in this study and were not concerned with inference, there are a wealth of areas to explore. Perhaps most notably, it
remains to be seen what distributional assumptions are truly needed about the distributions of model errors. As noted previously, in this project, we assumed only that the errors are independent and have zero mean, both of which are important for the model-fitting process. Requiring the errors to all have a mean of 0 ensures that there is a common intercept for all observations so that the model is well-defined. Independence is needed so that all observations are treated equally when calculating the cost function.

At the moment, it is unclear what effect the other common distributional assumptions have on LINEX regression. In terms of model fitting, itself, we ideally would require that no outliers be present in the data, but, as described in Section 3.4, outlyingness may best be defined in terms of relative cost, which are functions of LINEX residuals but have an unclear relationship to the model errors. Additionally, the common LS assumption of error normality is largely used for modeling the distributions of the parameter estimates for variable selection and inference, but the sampling distributions of the LINEX parameter estimates may not be normal even when the assumption holds. Further, while it may be reasonable to desire homoskedasticity of the errors, it is unclear what effect the variance would have on parameter estimation and inference. As such, for generalizing the LINEX regression methodology for broader use, the effects of the distributional assumptions about the errors will need to be studied extensively through both theory and simulation studies to evaluate their impacts on parameter estimation, inference, and variable selection.

Additionally, we want to do what we can to ensure that the parameter estimates are global minimizers of the cost function, while we were satisfied with simply having local minimizers that achieved our goal of decreasing the amount and size of under-predictions, which may or may not be global optima. As part of solving this problem, in the future, we need to investigate what optimization methods and initializations tend to produce stable global solutions.

For our specific application, there also remains a considerable amount of work that remains to be done. First, both here and in Prematilake and Ellingson [16], we fixed the error threshold at a set amount, but Prematilake [15] shows that, as expected, the values of the parameters change with different error thresholds. An ideal model would allow for us to incorporate the error threshold directly in the model, but this is considerably more complicated since, even under OLS, residuals will be correlated in the classical regression framework. Developing a model that can incorporate the error threshold into the model directly as a predictor would help make these models more generalizable. Furthermore, in shape analysis, it is not sufficient to simply approximate the shape of a single contour, but rather requires us to approximate all shapes within a sample using the same number of landmarks to ensure that all observations lie in the same mathematical space.

**Disclosure statement**

No potential conflict of interest was reported by the author(s).

**References**

[1] F. Akdeniz, *New biased estimators under the LINEX loss function*, Statist. Papers 45 (2004), pp. 175–190.

[2] M. Cain and C. Janssen, *Real estate price prediction under asymmetric loss*, Ann. Inst. Stat. Math. 47 (1995), pp. 401–414.
[3] Y.C. Chang and W.L. Hung, *LINEX loss functions with applications to determining the optimum process parameters*, Qual. Quant. 41 (2007), pp. 291–301.

[4] A. Chaturvedi, M.I. Bhatti, and K. Kumar, *Bayesian analysis of disturbances variance in the linear regression model under asymmetric loss functions*, Appl. Math. Comput. 114 (2000), pp. 149–153.

[5] I.L. Dryden and K.V. Mardia, *Statistical Shape Analysis: With Applications in R*, Vol. 995, John Wiley & Sons, Chichester, 2016.

[6] L. Ellingson, V. Patrangenaru, and F. Ruymgaart, *Nonparametric estimation of means on Hilbert manifolds and extrinsic analysis of mean shapes of contours*, J. Multivar. Anal. 122 (2013), pp. 317–333.

[7] J.A. Giles and D.E. Giles, *Risk of a homoscedasticity pre-test estimator of the regression scale under LINEX loss*, J. Stat. Plan. Inference 50 (1996), pp. 21–35.

[8] Z. Hoque and S. Hossain, *Improved estimation in regression with varying penalty*, J. Stat. Theory Pract. 6 (2012), pp. 260–273.

[9] Z. Hoque, J. Wesolowski, and S. Hossain, *Shrinkage estimator of regression model under asymmetric loss*, Comm. Statist. Theory Methods 47 (2018), pp. 5547–5557.

[10] D.G. Kendall, *Shape manifolds, procrustean metrics, and complex projective spaces*, Bull. London Math. Soc. 16 (1984), pp. 81–121.

[11] B. Kimia, *A Large Binary Image Database*, Spring, 2009. Available at [http://www.lems.brown.edu/dmc/](http://www.lems.brown.edu/dmc/).

[12] L.J. Latecki and R. Lakämper, *Convexity rule for shape decomposition based on discrete contour evolution*, Comput. Vis. Image Underst. 73 (1999), pp. 441–454.

[13] K. Ohtani, *Generalized ridge regression estimators under the LINEX loss function*, Statist. Papers 36 (1995), pp. 99–110.

[14] A. Parsian, *On the admissibility of an estimator of a normal mean vector under a LINEX loss function*, Ann. Inst. Stat. Math. 42 (1990), pp. 657–669.

[15] C.C. Prematilake, *Prediction of lower bounds of the number of sampling points for approximating shapes of planar contours*, Ph.D. diss., Texas Tech University, 2016.

[16] C. Prematilake and L. Ellingson, *Evaluation and prediction of polygon approximations of planar contours for shape analysis*, J. Appl. Stat. 45 (2018), pp. 1227–1246.

[17] A. Srivastava and E.P. Klassen, *Functional and Shape Data Analysis*, Vol. 1, Springer, New York, 2016.

[18] J. Strait, O. Chkrebtii, and S. Kurtek, *Automatic detection and uncertainty quantification of landmarks on elastic curves*, J. Am. Stat. Assoc. 114 (2019), pp. 1002–1017.

[19] H.R. Varian, *A Bayesian approach to real estate assessment*, in *Studies in Bayesian Econometric and Statistics in Honor of Leonard J. Savage*, North-Holland Pub. Co., Amsterdam, 1975, pp. 195–208.

[20] A.T. Wan, *A note on almost unbiased generalized ridge regression estimator under asymmetric loss*, J. Stat. Comput. Simul. 62 (1999), pp. 411–421.

[21] A. Zellner, *Bayesian estimation and prediction using asymmetric loss functions*, J. Am. Stat. Assoc. 81 (1986), pp. 446–451.