Elastic Handling of Predictor Phase in Functional Regression Models

Kyungmin Ahn
Department of Statistics
Florida State University
Tallahassee, FL
k.ahn@stat.fsu.edu

J. Derek Tucker
Sandia National Laboratories
Albuquerque, NM
jdtuck@sandia.gov

Wei Wu
Department of Statistics
Florida State University
Tallahassee, FL
wwu@stat.fsu.edu

Anuj Srivastava
Department of Statistics
Florida State University
Tallahassee, FL
anuj@stat.fsu.edu

Abstract

Functional variables serve important roles as predictors in a variety of pattern recognition and vision applications. Focusing on a specific subproblem, termed scalar-on-function regression, most current approaches adopt the standard $\mathbb{L}^2$ inner product to form a link between functional predictors and scalar responses. These methods may perform poorly when predictor functions contain nuisance phase variability, i.e., predictors are temporally misaligned due to noise. While a simple solution could be to pre-align predictors as a pre-processing step, before applying a regression model, this alignment is seldom optimal from the perspective of regression. We propose a new approach, termed elastic functional regression, where alignment is included in the regression model itself, and is performed in conjunction with the estimation of other model parameters. This model is based on a norm-preserving warping of predictors, not the standard time warping of functions, and provides better prediction in situations where the shape or the amplitude of the predictor is more useful than its phase. We demonstrate the effectiveness of this framework using simulated and stock market data.

1. Introduction

Functional data analysis (FDA) [10] is a branch of statistics that deals with functional variables, i.e. variables that are real- or vector-valued functions on a fixed domain. FDA has gained prominence in recent years because of the centrality of functional data in many applications. In fact, many problems involving so-called *big data* often consists of time-varying measurements, leading to functional data. Such data can be found in all branches of science, including vision, pattern recognition, biology, medical imaging, bioinformatics, social science, and so on. FDA deals with summarizing, clustering, modeling, and predicting variables involving functional data.

One of the fast growing subtopics in FDA is the problem of *regression* involving functional variables, either as predictors or responses or both. Depending whether the predictor or the response or both are functional variables, Morris [8] categorizes functional regression problems into three types: (1) functional predictor regression (scalar-on-function), (2) functional response regression (function-on-scalar) and (3) function-on-function regression. In this paper, we are interested in the problem of scalar-on-function regression [2, 11, 3], where the predictor $\{f\}$ is real-valued function over a fixed interval, element of a pre-specified functional space $\mathcal{F}$, and the response variable $\{y\}$ is a scalar. A simple baseline model used frequently in the literature is called the *functional linear model* (FLM) given by:

$$y_i = \alpha + \left(\int_0^T f_i(t)\beta(t)\,dt \right) + \epsilon_i, \;\; i = 1, \ldots, n\,. \quad (1)$$

Here $\beta \in \mathcal{F}$ plays the role of regression coefficient, f_is are observations of the predictor, y_is are observations of the response, and ϵ_is are the measurement errors. Various advancements beyond this simple model have been proposed but the inner-product based mapping from $\mathcal{F}$ to $\mathbb{R}$ remains central to this framework.

While the use of functional data has grown in recent years, there has also been a growing awareness of a prob-

lem/issue that is specific to functional data. Functional data most often comes with a **phase variability**, i.e., *a lack of registration between peaks and valleys across functions*. This situation arises, for example, when using bio-signals for diagnosing medical conditions, and where the measurements of signals across subjects lack synchronizations. Different subjects have different temporal evolutions of their bio-signals, introducing an intrinsic phase variability in the data. In mathematical terms, the predictor data is not $\{f_i\}$ but rather $\{(f_i \circ \gamma_i)\}$, where γ_is are unknown time warping functions that misalign predictor functions. (There are other forms of phase variability also, such $(f \circ \gamma_i)\dot{\gamma}_i$, $(f \circ \gamma_i)\sqrt{\dot{\gamma}_i}$, and so on, depending on if the time warping affects the amplitudes or not. The nature and expression of time warping is ultimately dependent on the application.) In some cases these misalignments or phase variability are simply linear or affine shifts, while in other cases the misalignments are nonlinear. One consequence of phase variability is the inflation of variance in the functional data that renders classical statistical model (Eqn. 1) ineffective. If we use such functions in Eqn. 1, then the estimates of model parameters and predicted values are adversely affected by the presence of γ_is. One needs to account for the phase variability in predictor measurements to improve performance.

One approach to handling phase variability is to separate the phase and amplitude components in the predictor functions, as described in [7, 14] and others. The separation results in alignment of peaks and valleys across functions using nonlinear time warpings. These warpings correspond to the phase components and the aligned functions correspond to the shape or amplitude components. One may envision regression models where both these components – phase and shape – are useful as predictors in the model. However, there are some other situations where only one of them, most notably the shape of the function, that may be of interest in predicting a response variable. This situation arises, for instance, in cases where the response depends primarily on the number of modes of the predictor functions. The locations of these modes and anti-modes are less important and, therefore, phase components are considered nuisance variables. Motivated by such situations, we develop a regression model where only the shape (or amplitude) of a function is considered as a predictor and its phase is removed from the consideration.

While phase-amplitude separation as a pre-processing step is possible, it seems better to separate these two components inside the regression model. The important question is how? One may consider a model of the type:

$$y_i = \alpha + \sup_{\gamma_i} \left(\int_0^T f_i(\gamma_i(t))\beta(t)\, dt \right) + \epsilon_i, \;\; i = 1, \ldots, n \,. \tag{2}$$

While this is a simple extension of the so-called functional linear model, it has a major shortcoming in that the alignment is based on standard warping under the standard $\mathbb{L}^2$ metric. As described in several places, see e.g. Marron [7], Ramsay-Silverman [12], and Srivastava-Klassen [14], the alignment of functions under the $\mathbb{L}^2$ leads to a degeneracy termed the *pinching effect*. Some authors avoid or minimize pinching by restricting warpings to a much smaller set, in a pre-determined manner. This restriction is unnatural as it is mostly impossible to pre-determine the optimal subset of warpings needed to align future data. Another possibility is to assume the presence of additional information in form of landmarks, points that need to be registered across functions, to help avoid pinching and misalignment. However, the availability of landmarks is rare in real data.

Notationally, we will use $\langle \cdot, \cdot \rangle$ to denote the $\mathbb{L}^2$ inner product and $\| \cdot \|$ for the $\mathbb{L}^2$ norm. In this paper, we present a novel solution that is motivated by the use of the Fisher-Rao metric in functional data alignment [15, 14]. In fact, this elastic functional data analysis (EFDA) framework suggests several ideas, only one of which is pursued in this paper. EFDA is based on replacing the use of $\mathbb{L}^2$ inner product and the $\mathbb{L}^2$ distance between functions by the Fisher-Rao Riemannian metric and the Fisher-Rao distance between these functions. The latter provides better mathematical and numerical properties, and indeed leads to a superior registration between functions. The challenge in using the original Fisher-Rao metric comes from its complicated expression, but that is overcome using the square root velocity function (SRVF), as described in Srivastava et al. [15]. One works with the SRVFs q_is instead of the predictors f_is and the Fisher-Rao metric becomes the standard $\mathbb{L}^2$ metric. In this framework, the time warpings of q_is, given by $(q_i \circ \gamma_i)\sqrt{\dot{\gamma}_i}$, are norm preserving. That is, $\|q_i\| = \|(q_i \circ \gamma_i)\sqrt{\dot{\gamma}_i}\|$ for $q_i \in \mathbb{L}^2$ and all warpings γ_i, and thus pinching is no longer possible. This, in turn, suggests two ways of fixing the problem in Eqn. 2:

1. **Use Fisher-Rao Metric and SRVF Representation**: One can compute SRVFs of the given predictors, and then simply replace the term $\sup_{\gamma_i} \langle f_i \circ \gamma_i, \beta \rangle$ in Eqn. 2 by the Fisher-Rao based inner product: $\sup_{\gamma_i} \left\langle (q_i \circ \gamma_i)\sqrt{\dot{\gamma}_i}, \beta \right\rangle$. Since any warping of q_i in this way preserves its norm, the pinching problem is avoided.

2. **Use a Norm-Preserve Warping and $\mathbb{L}^2$ Metric**: The other option is to change the basic definition of the warping itself, from the classical composition $(f_i \circ \gamma_i)$ to the norm-preserving action $((f_i \circ \gamma_i)\sqrt{\dot{\gamma}_i})$. In the new definition, a warping changes both the location and the height of a function value. This suggests using $\sup_{\gamma_i} \left\langle (f_i \circ \gamma_i)\sqrt{\dot{\gamma}_i}, \beta \right\rangle$ in Eqn. 2. That is, we assume that f_is are already in SRVF space and use them as such. This process may be useful when the data is

noisy and a further SRVF transformation enhances this noise due to the presence of a derivative. By treating f_is as SRVFs, one obtains the nice properties of this framework and avoids enhancing the noise. On the other hand, this warping is different from the typical warping $f \circ \gamma_i$ used in the alignment literature.

Each of these models avoid the pinching effect, and have their own pros and cons. Ultimately, the choice of a model depends on the nature of the data and the goals of the application. The response variable in both these models is invariant to the action of the time warping group on the predictor functions.

In this paper, we will develop the second approach and will call this the *elastic functional regression* model. In Section 2, we develop the resulting elastic functional regression model and present the parameter estimation technique in Section 3. In Section 4, we demonstrate this model using some simulated and real functional data, and compare its performance against some current ideas in the literature.

2. Proposed Elastic Framework

In this section we layout an elastic functional regression model for *scalar-on-function* problem with the property that the response variable is invariant to the phase component of the predictor. This framework is based on ideas used previously for alignment of functional data, or phase-amplitude separation, using the Fisher-Rao metric and the SRVF representation of functions. We start by briefly introducing those concepts and refer the reader to [15] for details.

As mentioned earlier, the use of $\mathbb{L}^2$ inner-product or $\mathbb{L}^2$ norm for alignment of functions leads to a well-known problem called the pinching effect. While some papers avoid this problem using a combination of external penalties and search space reductions, a superior solution comes from using an elastic Riemannian metric with appropriate invariance properties. This metric, called the *Fisher-Rao metric*, avoids the pinching effect without any external constraint and results in better alignment results. Let f be a real-valued function on the interval $[0, 1]$ (with appropriate smoothness) and let $\mathcal{F}$ denote the set of all such functions. For the purpose of alignment, one represents it using a square-root velocity function (SRVF) defined as $q(t) = \dot{f}(t)/\sqrt{|\dot{f}(t)|}$ or $q(t) = \text{sign}(\dot{f}(t))\sqrt{|\dot{f}(t)|}$. These two expressions are algebraically equivalent. One of the advantages of using SRVF is that under the transformation $f \mapsto q$, a complicated Fisher-Rao Riemannian metric and the Fisher-Rao distance into much simpler expressions. That is:

$$\langle\langle f_1, f_2\rangle\rangle_{FR} = \langle q_1, q_2\rangle\,, \quad \text{and} \quad d_{FR}(f_1, f_2) = \|q_1 - q_2\|\,.$$

If we warp a function f by a time warping γ, i.e., map $f \mapsto (f \circ \gamma)$, then its SRVF changes by $q \mapsto (q \circ \gamma)\sqrt{\dot{\gamma}}$. The latter is often denoted by (q, γ). The invariance property of the Fisher-Rao metric implies that for any $q_1, q_2 \in \mathbb{L}^2$ and $\gamma \in \Gamma$, we have: $\|(q_1, \gamma) - (q_2, \gamma)\| = \|q_1 - q_2\|$. In other words, the action of Γ on $\mathbb{L}^2$ is by isometries. A special case of this equation is that $\|(q, \gamma)\| = \|q\|$ for all q and γ. Thus, this action preserves the $\mathbb{L}^2$ norm of the SRVF and, therefore, avoids any pinching effect.

This framework motivates several solutions for avoiding the pinching problem associated with the inner-product term in Eqn. 2. While one can work with the SRVFs of the given predictor functions, they are prone to noise in the original data due to the involvement of a time derivative in the definition of SRVF. In case the original data is noisy, this noise gets enhanced by the derivative. As a workaround to this problem, we treat the given predictor functions to be in the SRVF space already. That is, we assume the action of warping γ_i on an f_is is given by $(f_i \circ \gamma_i)\sqrt{\dot{\gamma}_i}$ and not $f_i \circ \gamma_i$. With this action, we have that $\|(f_i, \gamma_i)\| = \|(f_i \circ \gamma_i)\sqrt{\dot{\gamma}_i}\| = \|f_i\|$.

Based on this argument, the inner-product term in Eqn. 2 can be replace by the term: $\sup_{\gamma_i} \langle \beta, (f_i, \gamma_i)\rangle$. This is a scalar quantity and represents a modified linear relationship between the predictor and the response. One can impose a more sophisticated single-index model on top of this construction as follows. Such single-index models have been used commonly in conjunction with the Functional Linear Model (FLM), see e.g. [16, 1, 13, 4]. Let $h : \mathbb{R} \to \mathbb{R}$ be any smooth function defined on the real line, and define the model:

$$y_i = h\Big(\sup_{\gamma_i} \langle \beta, (f_i, \gamma_i)\rangle\Big) + \epsilon_i, i = 1, \ldots, n \qquad (3)$$

The inclusion of h allows the model to capture nonlinear relationships between the predictor and the response variables. This single-index model (SIM) is generally the same as a generalized functional linear model (GFLM), but in SIM the link function h is unknown.

3. Parameter Estimation and Prediction

Next we consider the problem of estimating model parameters under the model given in Eqn. 3. The list of parameters, include the link function, h, and the coefficient of regression β. We take an iterative approach given in [5], where one updates the estimates of h or β while keeping the other fixed. Thus, we first focus on the techniques for updating 1) the estimation of β and 2) the estimation of single-index model h separately, and then we propose an iterative process for joint-estimation.

3.1. Estimation of β keeping h fixed

Given a set of observations $\{(f_i, y_i)\}$, the goal here is to solve for the coefficient of regression β, while keeping h fixed, using maximum-likelihood estimation. In order to

reduce the search space to a finite-dimensional set, we will assume that $\beta \in \{\sum_{j=1}^{J} c_j b_j | c_j \in \mathbb{R}\}$ for a fixed basis set $\mathcal{B} = \{b_j, j = 1, 2, \dots\}$ of $\mathbb{L}^2([0,1], \mathbb{R})$. The estimation problem is now given by:

$$\hat{c} = \underset{c \in \mathbb{R}^J}{\operatorname{argmin}} H(c), \text{ where } H : \mathbb{R}^d \to \mathbb{R},$$

$$H(c) = \left(\sum_{i=1}^{J} (y_i - h(\sup_{\gamma_i} \left\langle \sum_{j=1}^{J} c_j b_j, (f_i, \gamma_i) \right\rangle)^2 \right) .$$

This optimization can be summarized into an algorithm as follows.

Algorithm 1 Estimation of β keeping h fixed

1: Initialization Step. Choose an initial $c \in \mathbb{R}^J$ and compute $\beta(t) = \sum_{j=1}^{J} c_j b_j(t)$.
2: Find $\{\gamma_i^*\}$ using Dynamic Programming, $\gamma_i^* = \operatorname{argmin}_{\gamma \in \Gamma} \|\beta - (f_i, \gamma_i)\|^2$, for each $i = 1, \dots, n$.
3: Compute the aligned functions $\tilde{f}_i \leftarrow (f_i \circ \gamma_i^*)\sqrt{\dot{\gamma}_i^*}$.
4: Use an optimization code (such as `fminunc` in matlab) to find $\hat{c}$ that minimize the cost function H function.
5: Update $\beta(t) \mapsto \sum_{j=1}^{J} \hat{c}_j b_j(t)$. If the $|H(\hat{c})|$ is small, then stop. Else return to step 2.

3.2. Estimation of h keeping β fixed

Next we consider the problem of estimating the link function h given the data and the estimated β. The reason for introducing this single-index model is to capture nonlinear relationship between the predicted responses and observed responses. While there are many nonparametric estimators for handling h, we keep the model simple by restricting to lower-order polynomials. Hence, this link function can either be linear, quadratic, cubic, and so on: $h(x) = ax+b$, $h(x) = ax^2+bx+c$, and $h(x) = ax^3+bx^2+cx+d$, etc. In our experiment, we use the first three polynomial functions for h.

In terms of estimating h, we use the current estimate $\hat{\beta}$ to predict the responses according to: $\hat{y}_i^{(train)} = \sup_{\gamma_i} \left\langle \hat{\beta}, (f_i^{(train)}, \gamma_i) \right\rangle$, Then, we fit a polynomial function h between the predicted responses $\hat{y}_i^{(train)}$ and the observed responses $y_i^{(train)}$ using the least squares error criterion.

The full parameter estimation procedure is as presented in Algorithm 2.

3.3. Prediction of Response Under the Elastic Regression Model

One of the main goals of a regression model is to predict values of the response variable for the future predictor observations. We describe that prediction process in the elastic functional regression model. This process involves aligning the predictors to the coefficient $\hat{\beta}$ using dynamic programming algorithm. For a given $f^{(test)}$, the predicted value of y is:

$$\hat{y} = \hat{h}\Big(\sup_{\gamma_i} \left\langle \sum_{j=1}^{J} \hat{c}_j b_j, (f^{(test)}, \gamma_i) \right\rangle \Big). \tag{4}$$

We will use this process to evaluate the prediction performance of our proposed model, and other current models, using both simulated data and real data.

Algorithm 2 Elastic Scalar-on-Function Regression Model

1: Initialize h as the identity function ($h(x) = x$).
2: Given h, use Algorithm 1 to estimate $\hat{\beta}$.
3: For a given $\hat{\beta}$, fit the single-index model using the least squares criterion and update h.
4: If the stopping criterion is met, then stop. Else, return to step 2.

4. Experimental Illustration

In this section, we compare our method with four models that are natural alternatives to the proposed model. Either these models are commonly used in the literature or they are simple modifications of the current models for handling the phase variability in the predictors. These models are: Functional Linear Model (FLM); Pre-Aligned Functional Linear Model (PAFLM); Nonparametric regression model (NP) using a Gaussian kernel function and two different metrics: $\mathbb{L}^2$ distance and elastic distance. We briefly summarize and introduce these models.

Functional Linear Model (FLM) Functional Linear Model is first introduced by Ramsay and Dalzell [11] and the regression model can be expressed as in Eqn. 2. This model ignores phase variability in the predictor data and is quite vulnerable to the variability.

Pre-Aligned Functional Linear Model (PAFLM) We also implemented regression model called Pre-Aligned Functional Linear Model (PAFLM). PAFLM is the model which pre-aligns the training data and the test data using one of several existing alignment algorithms and then performs standard FLM. For example, the registration can be implemented by using Square-Root Velocity Functions (SRVFs) and *template function or karcher mean* from the *"Complete Alignment Algorithm"* [15]. This alignment is naturally suboptimal from the perspective of regression, since the response variable is not used in phase separation.

Nonparametric Regression Model Nonparametric regression is the model that does not require any predeter-

mined form but driven from the observed data. It has been developed and studied by Ferraty and Vieu [6] and several other authors, and takes the form: $y_i = r(f_i(t)) + \epsilon_i$, where r the unknown smooth map from $\mathcal{F}$ to $\mathbb{R}$, and is estimated by the functional Nadaraya-Watson (NW) estimator [9]. For the given data (f_i, y_i) for $i = 1, 2, \ldots, n$, the estimator is given by:

$$G(f) = \frac{\sum_{i=1}^{n} y_i K(d(f_i, f)/m^*)}{\sum_{i=1}^{n} K(d(f_i, f)/m^*)}, \tag{5}$$

where: K is a Gaussian kernel, and d is a chosen distance on the predictor space. The choice of d is critically important in kernel estimators. In the experiments presented later, we use:

- The standard $\mathbb{L}^2$ norm between predictors.
- The elastic distance in SRVF space we define, $d_s(f, f_i) = \lambda d_a + (1 - \lambda) d_p$, where λ is a proportion parameter, $d_a = \text{argmin}_{\gamma \in \Gamma} ||f - (f_i, \gamma_i)||$ and $d_p = ||\sqrt{\dot{\gamma}^*} - \sqrt{\dot{\gamma}_I^*}||$.

The scalar quantity m^* is the optimal bandwidth by a cross-validation procedure. $m^* = \text{argmin}_m \sum_{i=1}^{n} (y_i - G_{(-i)}(f_i))^2$, with $G_{(-i)}(f) = \frac{\sum_{j=1, j\neq i}^{n} y_j K(d(f_j, f))/m)}{\sum_{j=1, j\neq i}^{n} K(d(f_j, f))/m)}$ using the training data.

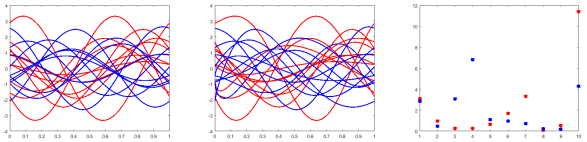
Next, we present experimental results from these methods on different data sets.

4.1. Simulated Data: Illustration of Method

As the first study, we simulate data for evaluating estimation and prediction performance of our elastic functional regression model.

Simulation:
In this experiment, we simulate $n = 20$ observations using the model stated in Eqn. 3. For the predictors, we use a truncated Fourier basis and random coefficients to form the functions $\{f_i\}$. Given these functions, we perturb them using random time warpings $\{\gamma_i\}$ to obtain the predictors $\{(f_i, \gamma_i)\}$. We also simulate the coefficient function β using the same Fourier basis but with a fixed coefficient vector $c_0 = [1, 1, \ldots, 1]$. We plug these quantities in the model and add independent observation noise, $\epsilon_i \sim N(0, 0.01^2)$, to obtain the responses $\{y_i\}$. These simulated quantities are shown in Fig. 1. In the following evaluations, we randomly divide this set in half training and half testing.

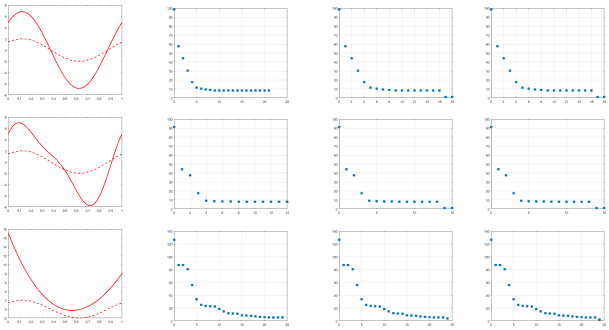
Model Estimation:
Using the training data, we estimate the model parameters h and β, as described in Algorithm 2. In order to evaluate this algorithm we actually use three different bases when fitting the model: 1) Fourier basis with only two elements,



$\{f_i\}$ $\{(f_i, \gamma_i)\}$ $\{y_i\}$

Figure 1: Simulated Data

2) Fourier basis with four elements, and 3) B-spline basis with four elements. The reason for using different bases for the estimation problem is to study the effects of bases on the model performance. We also use three different link functions, $h(\cdot)$: linear, quadratic, and cubic polynomial functions in the estimation setup. .



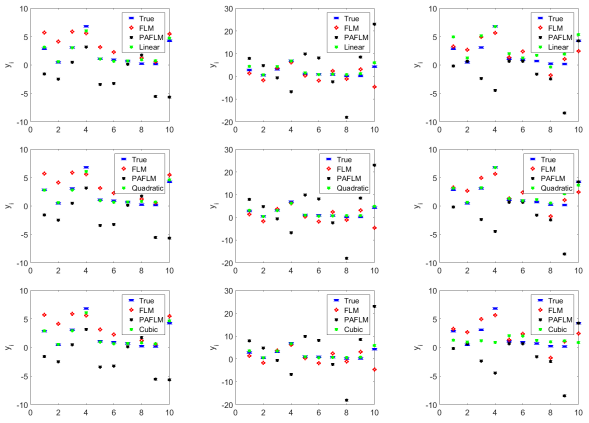
(a) $\hat{\beta}$ vs. β_0 (b) h: linear (c) h: quad (d) h: cubic

Figure 2: $\hat{\beta}$ (solid curve) vs. β_0 (dashed curve) and the evolution of cost function H for each link function, $h(\cdot)$

Fig. 2 presents results of the estimation process for different setting. Each row corresponds to a different choice of basis set for estimating β, i.e. Fourier basis with two elements ($J = 2$) (first row), Fourier basis with four elements ($J = 4$) (second row), and B-spline basis with 4 elements (third row). The last three columns correspond to the three choices of the link function h In Fig. 2a, we display estimated coefficient functions $\hat{\beta}$ (dashed red line) versus actual function β_0 (solid red line) for the different setting. We also plot the evolution of cost function H in Algorithm 2 for each of link functions: linear, quadratic, and cubic polynomial functions, in 2b, 2c, and 2d, respectively. These plots show that the cost function H goes down in all cases and the optimization algorithm provides at least local solutions.

Prediction Performance:
To evaluate model performance, we use the model parameters estimated using the training data for predicting response variable in the test data. This prediction followed the pro-

cedure laid out in described in Eqn. 4. The predicted responses are then compared with the true responses to quantity the prediction error.



(a) Fourier basis with $J = 2$ (b) Fourier basis with $J = 4$ (c) B-spline basis with 4 elements

Figure 3: Prediction of Regression Models using Fourier basis with $J = 2$, Fourier basis with $J = 4$, and B-spline basis with 4 elements

Fig. 3 shows the predicted values of the response variable using different models, along with the true values of the response. In addition to our elastic functional regression model, we study the standard Functional Linear Model (*FLM*) and the Pre-Aligned Functional Linear Model (*PAFLM*). Similar to the estimation problem, we study predictions using three different basis functions: Fourier basis with $J = 2$, Fourier basis with $J = 4$, and B-spline basis with 4 elements. These corresponding results are shown in 3a, 3b, and 3c, respectively. As earlier, we continue to use three different link functions $h(\cdot)$: linear (top), quadratic(middle), and cubic (bottom) polynomial functions. These prediction plots show that the predictions of our elastic functional regression are closer to the actual response variables compared to two traditional models.

To evaluate the performance of the models more precisely, we randomly iterate this process 5 times to ensure how elastic functional regression performs compared to other methods. Then we compute the average and the standard deviation of Mean Squared Prediction Errors (MSPE = $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$) from 5 different sets and use these quantities to compare different models. In addition to FLM and PAFLM mentioned above, we also implement and compute MSPEs of two nonparametric regression models – NP using the $\mathbb{L}^2$ norm and $\mathbb{L}^2$ using elastic distance (described in section 4). These last two approaches are model free and do not depend on estimating any regression coefficients. The numerical results for the average of the five MSPEs and corresponding standard deviation on simulated data are shown in Table 1 and Table 2. As these results show, the proposed elastic functional regression model is able provide a better prediction performance than the competing models despite using very simple models. In addition, the predictions from PAFLM are less accurate since this method is aligning the functional predictors without considering response variables, $\{y_i\}$. Also, the nonparametric regression model cannot perform well since this model captures all its information about data. This can be a problem since it captures all errors.

Basis	Fourier2	Fourier4	Bspline4
FLM	21.70 (20.79)	21.15 (19.16)	27.98 (33.47)
PAFLM	22.66(11.68)	147.12 (141.10)	29.89 (37.74)
h: Linear	1.99 (0.90)	2.16 (1.04)	**3.76** (3.17)
h: Quadratic	**0.87** (0.96)	**0.96** (1.14)	5.47 (4.38)
h: Cubic	4.93 (4.32)	4.14 (4.06)	11.56 (7.88)

Table 1: The average and the standard deviation (in parentheses) of the five MSPEs for three model-based methods on simulated test data.

Model	NP-$\mathbb{L}^2$	NP-elastic
MSPE	10.39 (9.12)	17.73 (14.81)

Table 2: The average and the standard deviation (in parentheses) of the five MSPEs for nonparametric regression model under the two distances.

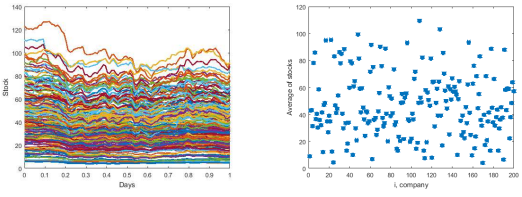
4.2. Application to Real Data

Next, we study the proposed model on some real data extracted from the *Historical Stock* data. The goal of this study is to use historical stock data, in form of functional predictors, and predict average stock value over a future time interval.

4.2.1 Description of the Data

QuantQuote has large amount of free historical stock data that is freely available for download from their website. There are total of 200 companies and each company has total 3,926 stock entries during the interval 1/2/1998 to 8/9/2013. For each company's stock, we exported stock prices from 7/8/2011 to 11/28/2011 to form functional predictors. So there are 100 time points over the selected interval for describing the predictor functions. Then, we compute the average value of stock prices over a future interval, namely 11/29/2011 to 8/9/2013, to form the scalar response variable.

The Fig. 4 shows the example of this stock data. The 200 functional predictors are shown in Fig. 4a and scalar



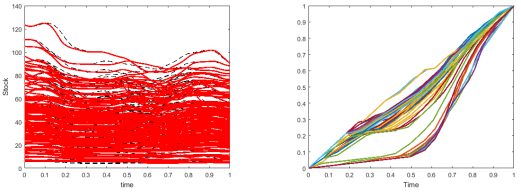
(a) Stock Prices, $\{f_i\}$ (b) $\{y_i\}$

Figure 4: Historical Stock Data

response variables are shown in Fig. 4b. We use first 140 curves to fit the model and remaining 40 curves as test.

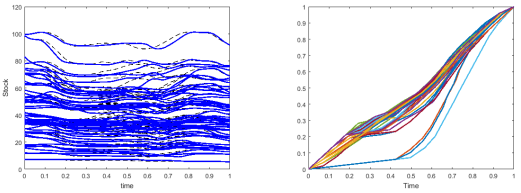
4.2.2 Analysis of the Data

For representing the coefficient function β, we use a B-spline basis with 20 elements and estimate the parameters using Algorithm 2.



(a) $\{f_i\}$ vs. Warped $\{f_i\}$ (b) $\{\gamma_i^*\}$

Figure 5: Training Set



(a) $\{f_i\}$ vs. Warped $\{f_i\}$ (b) $\{\gamma_i^*\}$

Figure 6: Test Set

Fig. 5 and 6 show the aligned functional predictors obtained by warping during the estimation and the prediction stages of the method. The original functions are drawn in black dashed curves and the warped functions are drawn using the red/blue solid curves. Fig. 5a shows the curves for the training data and Fig. 6a shows the curves for the test data. The corresponding optimal time warping functions, $\{\gamma_i^*\}$ on the training set and the test set are shown in 5b and 6b, respectively. Since the predictor functions look more aligned after the algorithm than before, we can assume that the data contains phase variations that are detrimental to the prediction performance. By handling these phase variations, we can expect higher prediction accuracy as shown next.

4.2.3 Prediction Results

The values of the predicted response under the different models are shown in Fig. 7. Red marks correspond to the actual response variables on the test set. In this figure, we compare our elastic functional regression model (*h: linear (first panel) or h: quadratic (second panel) or h: cubic (third panel)*, cyan) with four methods similar to the simulated data analysis: Functional Linear Model (*FLM*: green), Pre-Aligned Functional Linear Model (*PAFLM*: blue), Nonparametric regression model on $\mathbb{L}^2$ space, (*NP-*$\mathbb{L}^2$: magenta) and elastic distance (*NP-elastic*: yellow).

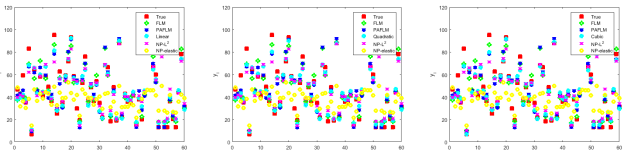


Figure 7: Predicted vs. Actual

In Fig. 7, we can see that predictions of elastic functional regression model are closer to the actual responses compared to the predictions of other methods. To investigate our results numerically of the real data, we compute MSPE and R^2. R^2 is a coefficient of determination that gives a measure of goodness-of-fit of the least squares model. We can compute R^2 by following equation:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \text{ where } i = 1, \ldots, n$$

where y_i is the actual response variable on the test set, $\hat{y}_i$ is the prediction, $\bar{y}$ is the mean of actual response variables on the test set. n is the number of predictors on the test set.

Model	MSPE (R^2)
FLM	72.1350 (0.9156)
PAFLM	77.0300 (0.9099)
h : Linear	54.7425 (0.9359)
h : Quadratic	**54.2575 (0.9365)**
h : Cubic	55.1075 (0.9355)
NP-$\mathbb{L}^2$	96.5150 (0.8870)
NP-elastic	79.8980 (0.9065)

Table 3: MSPE and R^2 (in parentheses) of each model

Table 3 presents MSPE of each model and corresponding R^2 in parentheses. It shows that the predictions of elastic

functional regression model outperformed most compare to other predictions of the functional regression models.

Predictions from the kernel regression model performed less accurately. This might be due to the observed functions having all different heights (relatively) and different starting points. Functional predictors in each training data and test data have different shapes (different heights and starting points) so nonparametric method cannot handle this problem.

5. Summary

The statistical functional regression model with phase variability is a well-known challenging problem. We have proposed a new elastic approach for handling predictor phase in functional regression models which is based on a norm-preserving warping of the predictors and handling the nuisance phase variability by optimizing the $\mathbb{L}^2$ inner product over the warping group in the model. We compare MSPE and R^2 of the model with several existing methods to demonstrate the effectiveness of this technique using simulated data and historical stock data.

References

[1] A. Ait-Sadi, F. Ferraty, R. Kassa, and P. Vieu. Cross-validated estimations in the single-functional index model. *Statistics*, 42(6):475–494, 2008.

[2] T. T. Cai, P. Hall, et al. Prediction in functional linear regression. *The Annals of Statistics*, 34(5):2159–2179, 2006.

[3] H. Cardot, F. Ferraty, and P. Sarda. Functional linear model. *Statistics & Probability Letters*, 45(1):11–22, 1999.

[4] H. C. Eilers and B. D. Marx. Flexible smoothing with b-splines and penalties. 11, 06 1996.

[5] P. H. C. Eilers, B. Li, and B. D. Marx. Multivariate calibration with single-index signal regression. *Chemometrics and Intelligent Laboratory Systems*, 96(2):196–202, 2009.

[6] F. Ferraty and P. Vieu. *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media, 2006.

[7] J. S. Marron, J. O. Ramsay, L. M. Sangalli, A. Srivastava, et al. Statistics of time warpings and phase variations. *Electronic Journal of Statistics*, 8(2):1697–1702, 2014.

[8] J. S. Morris. Functional regression. *Annual Review of Statistics and Its Application*, 2:321–359, 2015.

[9] E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

[10] J. O. Ramsay. When the data are functions. *Psychometrika*, 47(4):379–396, 1982.

[11] J. O. Ramsay and C. J. Dalzell. Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):539–572, 1991.

[12] J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, 2nd edition, 2005.

[13] P. T. Reiss, J. Goldsmith, H. L. Shang, and R. T. Ogden. Methods for scalar-on-function regression. *International Statistical Review*, 85(2):228–249, 2017.

[14] A. Srivastava and E. Klassen. *Functional and shape data analysis*. Springer, 2016.

[15] A. Srivastava, W. Wu, S. Kurtek, E. Klassen, and J. S. Marron. Registration of functional data using fisher-rao metric. *arXiv*, arXiv:1103.3817, 2011.

[16] T. M. Stoker. Consistent estimation of scaled coefficients. *Econometrica: Journal of the Econometric Society*, pages 1461–1481, 1986.