Regression-Based Elastic Metric Learning on Shape Spaces of Cell Curves

Adele Myers
University of California, Santa Barbara
Santa Barbara, CA 93106
adele@ucsb.edu

Nina Miolane
University of California, Santa Barbara
Santa Barbara, CA 93106
ninamiolane@ucsb.edu

Abstract

We propose a metric learning paradigm, Regression-based Elastic Metric Learning (REML), which optimizes the elastic metric for geodesic regression on the manifold of discrete curves. Geodesic regression is most accurate when the chosen metric models the data trajectory close to a geodesic on the discrete curve manifold. When tested on cell shape trajectories, regression with REML's learned metric has better predictive power than with the conventionally used square-root-velocity (SRV) metric. The code is publicly available here.

1 Introduction

Cell shape is strongly representative of cell function and can help diagnose many conditions including cancer [14]. However, most cell analyses do not consider the cell's shape in its entirety and instead only consider coarse global attributes like area, perimeter, or convexity [6]. Analyses that do use the cell's entire shape are often "static", as they disregard cell shape as it evolves over time [1, 13]. Because advances in live imaging are poised to provide an increasing amount of cell video data, statistical tools that describe the time evolution of cell shape are timely and necessary to precisely assess cell health and potential pathological conditions.

We propose a *metric learning* paradigm, Regression-based Elastic Metric Learning (REML). REML learns the optimal elastic metric for geodesic regression on the manifold of discrete curves: here, the manifold of cell shapes. Our work expands on the framework [3, 10] which parameterizes cell shapes from microscopy images with a 2D array of discrete points that trace a cell's outline. Then, we use the elastic metric implemented in Geomstats [11] to analyze these shapes on the manifold of discrete curves. The outline of each cell is itself a point on the manifold of discrete curves $\mathcal{M}$, and changes in cell shape over time form a trajectory on $\mathcal{M}$. Our method analytically solves the geodesic regression problem using different metrics and selects the metric that maximizes the coefficient of determination R^2 on the validation set Fig. 1. The elastic metric is particularly meaningful when analyzing cell shape because it quantifies "stretching" and "bending" between curves, which provides a biological measure of stretching and bending properties of the cells in a cell shape trajectory. We validate our approach on synthetic trajectories between real osteosarcoma cells. The experimental results confirm that our paradigm (i) learns biological parameters of cell shape evolution, and (ii) provides a mathematically-grounded approach to enhanced characterization of cell shape evolution.

2 Background

This section reviews the tools of Riemannian geometry that will support our metric learning method. Additional background can be found in [12, 2]. We model cell outlines as elements of the space of

Published at the Workshop *Learning Meaningful Representations of Life* (LMRL 2022) of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

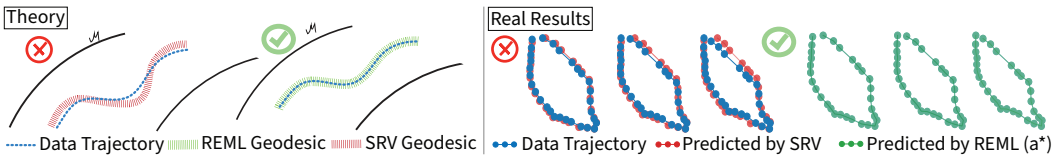


Figure 1: Summary and results of the proposed approach. Left: A trajectory may follow a geodesic as calculated by one metric but not follow a geodesic as calculated by another metric. Our paradigm learns the elastic metric (parameterized by a) that best represents the data trajectory as a geodesic on the manifold of discrete curves. Right: true cell trajectory overlaid with 1) cells predicted by a regression which utilizes our paradigm's learned metric parameter ($a*$) 2) cells predicted by a square-root-velocity (SRV) regression. Regression predictions using the SRV metric (red) do not match the data trajectory (blue), but our algorithm's $a*$ predicts the data trajectory perfectly: our prediction (green) perfectly overlays the data trajectory (blue).

regular planar curves $\mathcal{C} = \mathcal{C}^1([0,1], \mathbb{R}^2)$. Our metric learning approach will require us to calculate distances between cell shapes in $\mathcal{C}$. To do so, we introduce the concept of elastic metrics.

Elastic metrics A *Riemannian metric* g on a manifold $\mathcal{C}$ is a set of smoothly varying inner products defined on tangent spaces of $\mathcal{C}$. These define geometric measurements on $\mathcal{C}$, including distances, geodesics and exponential maps. Geodesics on manifolds generalize straight lines on vector spaces: they are curves that locally minimize the distance between points. Exponential maps generalize addition on vector spaces.

We equip $\mathcal{C}$ with a family of *elastic metrics* $g^{a,b}$ parameterized by $a, b > 0$ [9]:

$$g_c^{a,b}(h,k) = a^2 \int_0^1 \langle D_s h, N\rangle \langle D_s k, N\rangle ds + b^2 \int_0^1 \langle D_s h, T\rangle \langle D_s k, T\rangle ds, \quad \forall c \in \mathcal{C}, h, k \in T_c\mathcal{C}, \quad (1)$$

where c is a cell outline (a point in $\mathcal{C}$), h, k are infinitesimal deformations of c (tangent vectors in $T_c\mathcal{C}$), D_s is the derivative with respect to arc-length s along the curve c, (T, N) denote the unit tangent and the unit normal to c, respectively, and $<,>$ is the canonical Euclidean inner-product of $\mathbb{R}^2$. An elastic metric $g^{a,b}$ depends on two parameters: a "bending" parameter b and a "stretching" parameter a, which respectively evaluate whether two curves are "bent" or "stretched" compared to one another and then define how far apart these curves should be on $\mathcal{C}$. For example, when a is large, two curves that differ by a stretching operation will be far apart. When b is large, two curves that differ by a bending operation will be far apart. Elastic metrics are invariant under shape-preserving transformations, i.e., elements in the group $\mathcal{G}$ of 2D translations, 2D rotations, and re-scaling of cell outlines [12]. As such, elastic metrics are well-defined on the *space of curve shapes*, which is formally given by the quotient $\mathcal{M} = \mathcal{C}^1([0,1], \mathbb{R}^2)/\mathcal{G}$. The elastic metric with $a = 1$ and $b = 0.5$ is called the square-root-velocity (SRV) metric, and is often used as the "default metric" on the manifold of discrete curves. For this reason, we will consider the statistical analysis with the SRV metric as a baseline in our experiments.

Geodesic regression Because $\mathcal{M}$ is a manifold, we consider geodesic regression, which is the linear regression equivalent for manifolds. *Geodesic regression* on $(\mathcal{M}, g^{a,b})$ solves a least-square fitting problem [5]:

$$\min_{(p,v)\in T\mathcal{M}} \sum_{i=1}^{T} d^2\left(\mathrm{Exp}_p(t_i v), c_i\right), \quad (2)$$

where d and Exp are the Riemannian distance and exponential maps associated with the metric $g^{a,b}$. When the metric is Euclidean, this expression simplifies to the usual linear regression with intercept p and slope v. In geodesic regression, choice of metric affects the goodness of fit. Because metrics define distances between points in shape space, a data trajectory may follow the geodesic calculated by one metric but not another. We design an algorithm that finds the "optimal" metric for regression of a particular trajectory —the metric where the trajectory is closest to a geodesic, as judged by the value of the coefficient of determination R^2. This optimal metric gives the regression fit more predictive power, which in the case of microscopy image analysis in cell biology can thus provide a more accurate prediction of future cell shapes. We will perform metric learning across all possible elastic metrics and compare our results to the commonly used SRV metric.

Related Works (i) *Shape Spaces:* Cell biology has only recently started to use tools of statistical shape analysis. Like us, Philip *et al.* study cell shapes, but unlike us, they use the Kendall metric, which is a metric designed to compare shapes defined as sets of landmarks, while the elastic metric is meant to study shapes of curves [3]. Cho *et al.* use the SRV metric to study biological shapes, but they do not generalize their study to all elastic metrics, and they do not study cell shape as we do [4]. (ii) *Metric Learning:* The problem of metric learning has received considerable attention in machine learning literature for many years (see [7] for a survey). However, no existing paradigm optimizes the elastic metric for geodesic regression on cell shape trajectories. Riemannian metric learning has been studied by [8, 15], but these works do not learn elastic metrics. In research concurrent to ours, Bauer *et al.* [2] proposed an *elastic metric learning* scheme. However, they optimize the elastic metric in the context of shape classification, while we optimize it here for geodesic regression. Additionally, they use a random search on the elastic metrics' parameter, whereas we propose a gradient-ascent scheme instead.

3 Methods

In what follows, we consider synthetic trajectories of osteosarcoma cell shapes changing through time. The synthetic trajectory draws a geodesic between two real osteosarcoma cells, and thus, the synthetic shapes in the trajectory mimic realistic cell outlines. We give each trajectory a predetermined (i) number of time points, (ii) number of sampling points on each cell outline, and (iii) amount of measurement noise on each sampling point. These discrete cell outlines lie on the manifold of discrete curves, which we equip with an elastic metric. We divide the time trajectory of cell shapes $c_1, ..., c_T$ into three sequential datasets: a train set (60%), a validation set (30%) and a test set (10%), the size of which depends on the size of the synthetic trajectory. Our paradigm uses the train set to learn regression fit parameters for any metric g_{ab} being tested by the validation set. The validation set allows us to learn the "optimal" metric, chosen as the metric that maximizes the value of the coefficient of determination R^2. We denote $a*$ the parameter of this optimal elastic metric. We use the test data set to evaluate the regression performance of the learned $a*$ to that of the SRV metric.

1. Training: Geodesic regression with $F_{a,b}$-transform Our training procedure simplifies regression fit calculations by harnessing a key property of elastic metric $g^{a,b}$. Riemannian operations on cell shapes c on the manifold $(\mathcal{M}, g^{a,b})$ are equivalent to Euclidean operations in a linear "F_{ab} space", which is accessible through the F_{ab} transform defined as [12]:

$$F_{ab}(c) = 2b\|c'\|^{1/2}\left(\frac{c'}{\|c'\|}\right)^{\frac{a}{2b}}, \tag{3}$$

where c' is the derivative of the curve w.r.t. its parameter $s \in [0, 1]$. The F_{ab}-transform maps a curve c, which lies on the Riemannian manifold $\mathcal{M}$, to a point in the Euclidean F_{ab}-space. The F_{ab}-transform also has an inverse, which maps elements of the F_{ab}-space back to curve shapes on $\mathcal{M}$. This allows us to perform Euclidean operations in F_{ab} space and then transform the results back into shape space. Using the F_{ab}-transform, the geodesic regression of Eq. 2 becomes a linear regression on the train set $F_{ab}(c_1), ..., F_{ab}(c_{T_{\text{train}}})$.

2. Validation: Optimization of the elastic parameter a We use the validation set to perform elastic metric learning. The definition of the elastic metric in Eq. 1 shows that the parameter b is only a rescaling factor, as only the ratio $\frac{a}{b}$ substantially affects the metric. Different choices of b only change units and do not affect whether a trajectory is a geodesic or not. Thus, we fix the "units of the calculation" by imposing $b = 0.5$ and perform elastic metric learning by optimizing only a. We evaluate how close a trajectory is to a geodesic by calculating the R^2 value of the fitted geodesic regression model:

$$R^2 = 1 - \frac{\sum_{i=i_{train}}^{i_{val}} \|y_i - \hat{y}_i\|}{\sum_{i_{train}}^{i_{val}} \|y_i - \bar{y}\|}, \tag{4}$$

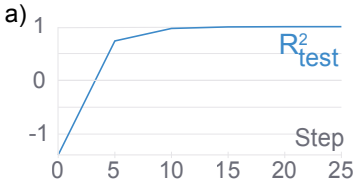
where $y_i = F_a(c_i)$ and $\bar{y} = \frac{1}{n_{\text{val}}} \sum_{i=i_{train}}^{i_{val}} y_i$. $\hat{y}_i$ is the i^{th} cell shape predicted by the regression model parameters derived from the "training" data set. Note that R^2 depends explicitly on the elastic parameter a, which allows us to compute its gradient with respect to a (see analytical expression in the appendices).

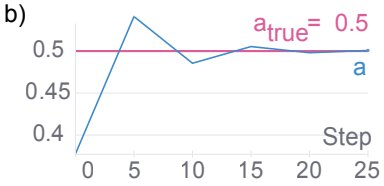
We perform gradient ascent on the coefficient of determination R^2 of the validation set to find the optimal a, the $a*$ that brings R^2 closest to 1. Importantly, we derive the gradient of R^2 with respect to a (see appendices for derivation).

3. Evaluation and Test We evaluate our metric learning method based on 1) whether the $a*$ calculated by the validation dataset is close to a_{true} (which is the metric used to create the synthetic data trajectory), 2) whether regression on the test dataset using $a*$ provides a good fit and R^2 value, and 3) whether the regression fit on the test dataset using $a*$ is better or worse than the regression fit using the SRV metric.

4 Experiments

Datasets We simulate the evolution of a cancer cell through time by generating an extensive set of synthetic geodesics between two cancer cells extracted from real microscopy images [10]. We generate geodesics with all combinations of the following parameters: $a_{true} \in \{0.2, 0.5, 0.7, 0.9, 1.0\}$ (a used to generate synthetic geodesic trajectory i.e., the metric space where the trajectory follows a geodesic), number of time points T (number of cells on the geodesic): $T \in \{20, 50, 100, 200\}$, number of sampling points n_s along each cell outline $n_s \in \{30, 50, 100\}$, and standard deviation of a centered Gaussian noise $\sigma \in \{0.0, 0.001, 0.01\}$ added on the 2D coordinates of the sampling points. Our synthetic experiments are directly relevant to biological researchers as they help inform the characteristics of their experimental cell shape dataset. The number of time points T corresponds to a choice of length of biological experiment video recording; the number of sampling points n_s is a biologist's choice upon extracting a discrete cell outline from a microscopy image, and the noise level σ represents the uncertainty during the cell outline extraction process from images.





c)

a_{true}	% of runs where a* metric outperforms SRV metric
0.2	41/44 = 93%
0.5	30/32 = 94%
0.7	28/32 = 88%
0.9	17/30 = 56%

Figure 2: Experimental results on synthetic trajectories modeling the evolution of real osteosarcoma cells. (a-b) shows our method simultaneously converging upon $R^2 = 1$ and $a_{true} = 0.5$. c) When the number of cell sampling points is between $30 - 50$, the number of cells in the trajectory is $100 - 200$, and the noise level is < 0.005, the $a*$ metric predicted by our algorithm outperforms the SRV metric with a better R^2 at every a_{true} (excluding $a_{true} = 1$, which is the SRV metric itself).

Results Our experimental results demonstrate that (i) our algorithm successfully converges $a*$ to a_{true} and R^2_{test} to 1 and (ii) our method outperforms the SRV metric at every a_{true} excluding $a_{true} = SRV$ when the noise level is < 0.005, the cells are sampled between $30 - 50$ times, and there are $100 - 200$ cells in the trajectory. Our results show that our method (i) correctly learns the metric a_{true} that was used to generate the synthetic cell trajectory and (ii) provides better predictions of future cell shapes. Learning a is biologically relevant because a quantifies how much a cell bends and stretches over a given amount of time.

5 Conclusion and Future Work

Our method, Regression-based Elastic Metric learning, describes an automated way for finding the elastic metric that brings a trajectory closest to a geodesic on the manifold of discrete curves. Our method increases the predictive power of geodesic regression, which allows for more accurate prediction of future cell shapes. In the long term, such methodology could be used for early detection of diseases or for early diagnosis of chemo-resistant cells based on their early evolution. Currently, most analyses on the manifold of discrete curves use the SRV metric by default. However, our results show that the SRV metric does not produce the best regression fit for most trajectories. Our method recognizes that the "optimal" metric is trajectory-dependent and thus creates an opportunity for improved analysis on all cell data trajectories.

References

[1] Elaheh Alizadeh et al. "Measuring systematic changes in invasive cancer cell shape using Zernike moments". In: *Integrative biology : quantitative biosciences from nano to macro* (8) (Oct. 2016).

[2] Martin Bauer et al. "Elastic Metrics on Spaces of Euclidean Curves: Theory and Algorithms". In: *arXiv preprint arXiv:2209.09862* (2022).

[3] Karthik Bharath and Sebastian Kurtek. "Analysis of shape data: From landmarks to elastic curves". en. In: *Wiley Interdiscip. Rev. Comput. Stat.* 12.3 (May 2020).

[4] Min Ho Cho, Amir Asiaee, and Sebastian Kurtek. "Elastic statistical shape analysis of biological structures with case studies: A tutorial". en. In: *Bull. Math. Biol.* 81.7 (July 2019), pp. 2052–2073.

[5] Thomas Fletcher. "Geodesic Regression on Riemannian Manifolds". In: *Proceedings of the Third International Workshop on Mathematical Foundations of Computational Anatomy - Geometrical and Statistical Methods for Modelling Biological Shape Variability*. Ed. by Pennec et al. Toronto, Canada, Sept. 2011, pp. 75–86. URL: https://hal.inria.fr/inria-00623920.

[6] Allison M Gabbert et al. "Septins are essential for protrusion and detachment in collective border cell migration". Apr. 2021.

[7] Brian Kulis et al. "Metric learning: A survey". In: *Foundations and Trends® in Machine Learning* 5.4 (2013), pp. 287–364.

[8] Guy Lebanon. "Learning riemannian metrics". In: *arXiv preprint arXiv:1212.2474* (2012).

[9] Washington Mio, Anuj Srivastava, and Shantanu Joshi. "On shape of plane elastic curves". In: *International Journal of Computer Vision* 73.3 (2007), pp. 307–324.

[10] Nina Miolane et al. *A Riemannian elastic metric captures cellular shape heterogeneity and enhances morphological discrimination. (Personal communication)*. 2022.

[11] Nina Miolane et al. "Geomstats: A python package for riemannian geometry in machine learning". In: *Journal of Machine Learning Research* 21 (2020), pp. 1–9. ISSN: 15337928. arXiv: 2004.04667.

[12] Tom Needham and Sebastian Kurtek. "Simplifying transforms for general elastic metrics on the space of plane curves". In: *SIAM Journal on Imaging Sciences* 13.1 (2020), pp. 445–473.

[13] Jude M. Phillip et al. "A robust unsupervised machine-learning method to quantify the morphological heterogeneity of cells and nuclei". In: *Nature Protocols* 16.2 (Jan. 2021), pp. 754–774.

[14] Ashok Prasad and Elaheh Alizadeh. "Cell Form and Function: Interpreting and Controlling the Shape of Adherent Cells". In: *Trends in Biotechnology* 37.4 (Apr. 2019), pp. 347–357. DOI: 10.1016/j.tibtech.2018.09.007. URL: https://doi.org/10.1016/j.tibtech.2018.09.007.

[15] Benoit Sauty and Stanley Durrleman. "Riemannian metric learning for progression modeling of longitudinal datasets". In: *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2022, pp. 1–5.

6 Appendix

6.1 Derivation of R^2 in terms of a

This derivation finds R^2 in terms of a. We are considering regression on curve trajectories, so R^2 is given by

$$R^2 = 1 - \frac{\sum_{n_{train}}^{n_{val}} d(c_i, \hat{c}_i)^2}{\sum_{n_{train}}^{n_{val}} d(c_i, \bar{c})^2} \tag{5}$$

.

where c_i is an arbitrary curve in the trajectory, $\hat{c}_i$ is a curve predicted by the regression model, $\bar{c}$ is the mean curve in the data set curve trajectory, n_{train} is the number of curves in the data set trajectory that are used to train the regression model, and $n_{val} - n_{train}$ is the number of curves in the validation data set trajectory that we use to calculate R^2 (and thus, the number of curves that we use to optimize R^2 to find $a*$). The numerator of the second term is the (un-normalized) mean squared error, which shows how much variation exists between the data set curve trajectory and the regression-predicted trajectory. The denominator of the second term is the (un-normalized) variance, which shows how much variation exists between the curves in the data set trajectory.

We will begin by finding the mean squared error (MSE) in terms of a.

$$\begin{aligned} MSE &= \sum_{i=n_{train}}^{n_{val}} \|\vec{r_i}\|^2 \\ &= \sum_{i=n_{train}}^{n_{val}} d(c_i, \hat{c}_i)^2. \end{aligned}$$

We choose to do this analysis in $F_a - space$, and use q's to denote curves which have been transformed to $F_a - space$. Thus, the derivation continues as

$$\begin{aligned} &= \sum_{i=n_{train}}^{n_{val}} \|q_i - \hat{q}_i\|^2 \\ &= \sum_{i=n_{train}}^{n_{val}} \|q_i - \hat{\beta}_0 - \hat{\beta}_1 t_i\|^2 \end{aligned}$$

To continue our derivation of the MSE in terms of a, we must derive expressions for the β parameters in terms of a. Consider the linear regression in F-space:

$$q_i = \beta_0 + \beta_1 t_i. \tag{6}$$

We will learn/estimate the coefficients β_0, β_1 using the training set of cells. We will learn the optimal a factor using the validation set. The size of the training set is denoted n_{train}, the size of the validation set is denoted $n_{val} - n_{train}$.

The estimated coefficients of the regressions are given by the normal equations, i.e.

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{7}$$

where $\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix}$ are the estimated coefficients, $X = \begin{bmatrix} 1 & t_1 \\ \vdots & \vdots \\ 1 & t_n \end{bmatrix}$ is the design matrix, $y = [y_1 \quad \dots \quad y_n]$ is the response. We have

$$X^T X = \begin{bmatrix} n_{train} & \sum_{i=1}^{n_{train}} t_i \\ \sum_{i=1}^{n_{train}} t_i & \sum_{i=1}^{n_{train}} t_i^2 \end{bmatrix} = n_{train} \begin{bmatrix} 1 & \bar{t} \\ \bar{t} & \bar{t^2} \end{bmatrix}, \tag{8}$$

where we have introduced the notations: $\bar{t} = \sum_{i=1}^{n_{train}} t_i$ and $\bar{t^2} = \sum_{i=1}^{n_{train}} t_i^2$.

The inverse gives:

$$(X^T X)^{-1} = \frac{1}{n_{train}(\bar{t^2} - (\bar{t})^2)} \begin{bmatrix} \bar{t^2} & -\bar{t} \\ -\bar{t} & \bar{1} \end{bmatrix}. \tag{9}$$

Multiplying by X^T:

$$(X^T X)^{-1} X^T = \frac{1}{n_{train}(\bar{t^2} - (\bar{t})^2)} \begin{bmatrix} \bar{t^2} - \bar{t}.t_1 & \dots & \bar{t^2} - \bar{t}.t_{n_{train}} \\ -\bar{t} + t_1 & \dots & -\bar{t} + t_{n_{train}} \end{bmatrix}. \tag{10}$$

Multiplying by y:

$$\hat{\beta} = (X^T X)^{-1} X^T y = \frac{1}{n_{train}(\bar{t^2} - (\bar{t})^2)} \begin{bmatrix} \sum_{i=1}^{n_{train}} (\bar{t^2} - \bar{t}.t_i) q_i \\ \sum_{i=1}^{n_{train}} (-\bar{t} + t_i) q_i \end{bmatrix}. \tag{11}$$

We introduce the notations:

$$\tau_{i0} = \frac{(\bar{t^2} - \bar{t}.t_i)}{n_{train}(\bar{t^2} - (\bar{t})^2)}; \qquad \tau_{i1} = \frac{(-\bar{t} + t_i)}{n_{train}(\bar{t^2} - (\bar{t})^2)} \tag{12}$$

so that:

$$\hat{\beta} = \begin{bmatrix} \sum_{i=1}^{n_{train}} \tau_{i0} q_i \\ \sum_{i=1}^{n_{train}} \tau_{i1} q_i \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n_{train}} \tau_{i0} F_{ab}(c_i) \\ \sum_{i=1}^{n_{train}} \tau_{i1} F_{ab}(c_i) \end{bmatrix}, \tag{13}$$

where the last expression highlights the dependency of this quantity in a, b.

Thus, we can continue our MSE derivation as:

$$\begin{aligned}
&= \sum_{i=n_{train}}^{n_{val}} \|q_i - \sum_{j=1}^{n} \tau_{j0} q_j - \sum_{j=1}^{n} \tau_{j1} q_j t_i\|^2 \\
&= \sum_{i=n_{train}}^{n_{val}} \|q_i - \sum_{j=1}^{n} (\tau_{j0} + \tau_{j1} t_i) q_j\|^2 \\
&= \sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \sum_{j=1}^{n} (\tau_{j0} + \tau_{j1} t_i) F_{ab}(c_j)\|^2 \\
&= \sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \sum_{j=1}^{n} \tau_{ij} F_{ab}(c_j)\|^2,
\end{aligned}$$

where we denote $n = n_{train}$ for convenience of notations.

Now, let's begin our derivation of the (un-normalized) variance (VAR) of the data set curve trajectory.

$$\begin{aligned}
VAR = &= \sum_{i=n_{train}}^{n_{val}} d(c_i, \bar{c})^2 \\
&= \sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \frac{1}{n_{val}} \sum_{j=1}^{n_{train}} F_{ab}(c_j)\|^2,
\end{aligned}$$

where the "f-transform" explicitly depends on a.

Thus, in its entirety, R^2 w.r.t. a is given by:

$$R^2 = 1 - \frac{\sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \sum_{j=1}^{n} \tau_{ij} F_{ab}(c_j)\|^2}{\sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \frac{1}{n_{val}} \sum_{j=1}^{n_{train}} F_{ab}(c_j)\|^2} \tag{14}$$

The next section expands the norm in this result.

6.2 Expansion of the norm $\|\vec{r_i}\|^2$.

The previous section derives R^2 in terms of a, and comes to the result:

$$R^2 = 1 - \frac{\sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \sum_{j=1}^{n} \tau_{ij} F_{ab}(c_j)\|^2}{\sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \frac{1}{n_{val}} \sum_{j=1}^{n_{train}} F_{ab}(c_j)\|^2} \tag{15}$$

This section will show how we expand the norm in the equation above. We will use the MSE norm as an example. $\vec{r_i}$ is a vector which indicates the difference between two curves: c_i, which is the i^{th} curve in the data set trajectory, and $\hat{c}_i$, which is the corresponding i^{th} curve in the regression-predicted trajectory. Because $\vec{r_i}$ indicates a difference between two curves, it must be a function of the parameter s, which parameterized all curves. Thus, its long-form notation is $r_i\vec{(}s)$. Conceptually, if all curves are deformed circles, then the parameter s tells you where a point on a curve maps to on an equivalent circle. s ranges from 0 to 1 by construction, where $s = 0$ is the start point of the curve and $s = 1$ is the end point of the curve. The squared vector norm of the function $r_i\vec{(}s)$ is given by

$$\sum_{i=n_{train}}^{n_{val}} \|\vec{r_i}\|^2 = \sum_{i=n_{train}}^{n_{val}} \int_0^1 \|r_i\vec{(}s)\|^2 \, ds = \sum_{i=n_{train}}^{n_{val}} \int_0^1 r_i\vec{(}s) \cdot r_i\vec{(}s) \, ds$$

Now, we will describe how we compute this integral. In our code, we take discrete samples of the points in each curve. Therefore, in our code, a curve is represented by a 2D array containing the (x, y) coordinates of these samples. Because c_i's in curve space are 2D discrete curves, q_i's are also discrete curves (in the linear $F_a - space$), and $\vec{r_i}$'s are discrete vectors comprised of the coordinate differences between two curves. Because $\vec{r_i}$'s are discrete, we will compute the integral above using a Riemann sum.

$$\sum_{i=n_{train}}^{n_{val}} \|\vec{r_i}\|^2 = \sum_{i=n_{train}}^{n_{val}} \sum_{s=0}^{1} r_i\vec{(}s) \cdot r_i\vec{(}s) \, \Delta s$$

Here, $\Delta s = \frac{\text{length of curve}}{\text{number of segments}} = \frac{1}{\text{number of sampling points} - 1}$ since in $F_a - space$, the transformed curve q_i has one fewer sampling point than c_i.

6.3 Derivation of R^2's derivative with respect to a

In the previous appendix sections, we derived R^2 w.r.t. a. Now, we will compute the gradient of R^2 w.r.t. a.

$$R^2 = 1 - \frac{\sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \sum_{j=1}^{n} \tau_{ij} F_{ab}(c_j)\|^2}{\sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \frac{1}{n_{val}} \sum_{j=1}^{n_{train}} F_{ab}(c_j)\|^2}$$

$$\begin{aligned}
\frac{d}{da} R^2 &= -\frac{d}{da} \frac{\sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \sum_{j=1}^{n} \tau_{ij} F_{ab}(c_j)\|^2}{\sum_{i=n_{train}}^{n_{val}} \|F_{ab}(c_i) - \frac{1}{n_{val}} \sum_{j=1}^{n_{train}} F_{ab}(c_j)\|^2} \\
&= -\frac{d}{da} \frac{MSE}{VAR} \\
&= -\frac{\frac{d}{da} MSE * VAR - MSE * \frac{d}{da} VAR}{VAR^2}
\end{aligned}$$

We now calculate $\frac{d}{da} MSE$ and $\frac{d}{da} VAR$.

Let's start with $\frac{d}{da}MSE$. Using information from the previous section, we see

$$\begin{aligned}\frac{d}{da}MSE &= \frac{d}{da}\sum_{i=n_{train}}^{n_{val}}\sum_{s=0}^{1} r_i(\vec{s})\cdot r_i(\vec{s})\,\Delta s \\ &= 2\sum_{i=n_{train}}^{n_{val}}\sum_{s=0}^{1}\left[\frac{d}{da}r_i(\vec{s})\right]\cdot r_i(\vec{s})\,\Delta s\end{aligned}$$

where for the MSE, $r_i(s) = F_{ab}(c_i(s)) - \sum_{j=1}^{n}\tau_{ij}F_{ab}(c_j(s))$. Calculating $\frac{d}{da}r_i(\vec{s})$ is the trickiest part. Halfway through the calculation, we switch into polar coordinates because polar coordinates allow us to easily take derivatives of vectors raised to the power of the differentiated variable (in this case, a).

$$\begin{aligned}\frac{d}{da}r_i(\vec{s}) &= \frac{d}{da}\left[|c_i'|^{1/2}\frac{c_i'}{|c_i'|}^{a} - \sum_{j=1}^{n}\tau_{ij}|c_j'|^{1/2}\frac{c_j'}{|c_j'|}^{a}\right] \\ &= |c_i'|^{1/2}\frac{d}{da}\frac{c_i'}{|c_i'|}^{a} - \sum_{j=1}^{n}\tau_{ij}|c_j'|^{1/2}\frac{d}{da}\frac{c_j'}{|c_j'|}^{a} \\ &= |c_i'|^{1/2}\frac{d}{da}e^{iarg(c_i')a} - \sum_{j=1}^{n}\tau_{ij}|c_j'|^{1/2}\frac{d}{da}e^{iarg(c_j')a} \\ &= |c_i'|^{1/2}(iarg(c_i'))e^{iarg(c_i')a} - \sum_{j=1}^{n}\tau_{ij}|c_j'|^{1/2}(iarg(c_j'))e^{iarg(c_j')a} \\ &= |c_i'|^{1/2}(e^{i\pi/2}arg(c_i'))e^{iarg(c_i')a} - \sum_{j=1}^{n}\tau_{ij}|c_j'|^{1/2}(e^{i\pi/2}arg(c_j'))e^{iarg(c_j')a} \\ &= |c_i'|^{1/2}(arg(c_i'))e^{i(\pi/2+arg(c_i')a)} - \sum_{j=1}^{n}\tau_{ij}|c_j'|^{1/2}(arg(c_j'))e^{i(\pi/2+arg(c_j')a)}\end{aligned}$$

where we have used the facts that $e^{iarg(c_i')} = \frac{c_i'}{|c_i|}$ and $i = e^{i\pi/2}$. This result can now be plugged into the equation for $\frac{d}{da}MSE$.

The calculation for $\frac{d}{da}VAR$ is similar, except for the VAR, $r_i(s) = F_{ab}(c_i) - \frac{1}{n_{val}}\sum_{j=1}^{n_{train}}F_{ab}(c_j)$.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This work instead encourages ethically responsible computing as it focuses on differential privacy. We thereby do not see any potential for negative social impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Will be part of supplementary material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] computations done locally on a Macbook Pro
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] No data was curated. All the datasets were publicly available
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]