Fast Approximation of Distance Between Elastic Curves using Kernels

Hedi Tabia[1]
http://perso-etis.ensea.fr/tabia
David Picard[1]
http://perso-etis.ensea.fr/~picard
Hamid Laga[2]
http://people.unisa.edu.au/Hamid.Laga
Philippe-Henri Gosselin[1,3]
http://perso-etis.ensea.fr/~gosselin/

[1] ETIS/ENSEA, University of Cergy-Pontoise, CNRS, UMR 8051, France
[2] Phenomics and Bioinformatics Research Centre, University of South Australia, Australia
[3] INRIA Rennes Bretagne Atlantique, France

Elastic shape analysis on non-linear Riemannian manifolds provides an efficient and elegant way for simultaneous comparison and registration of non-rigid shapes. In such formulation, shapes become points on some high dimensional shape space. A geodesic between two points corresponds to the optimal deformation needed to register one shape onto another. The length of the geodesic provides a proper metric for shape comparison [2, 3, 4]. Joshi et al. [2] and Srivastava et al. [7] proposed the Square Root Velocity Function parametrization (SRVF) that allow to compute geodesic distances between closed curves $\mathbb{R}^n$. The distance is invariant to different shape preserving geometric transformation including translation, rotation and re-parametrization. The computation of geodesics, and therefore the metric, is computationally very expensive as it involves a search over the space of all possible rotations and re-parameterizations. This problem is even more important in shape retrieval scenarios where the query shape is compared to every element in the collection to search.

In this paper, we propose a fast approximation of this metric, using Kernel functions, while keeping its nice properties such as the invariance to geometric transformations. We search a subspace, equipped with the standard dot product, and in which the discriminative power of the original distance between two curves is retained. The key idea is to design a kernel function $k(\cdot,\cdot)$ associated with the elastic metric, and build a mapping function such that the dot product between mapped elements is as close as possible to the original kernel $k(x,y)$. This mapping is based on the Nyström method for kernel approximation [1]. The advantage of this formulation is that the heavy-computational metric becomes now a dot product in the new subspace with very low dimensions. This reduces significantly the computation time needed to compare one shape to all the elements of the collection to search. We further show that the approximated distance preserves the invariance properties and achieves retrieval performance that is competitive with the original metric.

Given the elastic distance between two curves $d(\beta_1,\beta_2)$, we consider a kernel function $k(\beta_1,\beta_2)$ associated with it. The approximation we propose finds a mapping function P to some specific space such that the dot product between mapped elements $P(\beta_1)^\top P(\beta_2)$ is as close as possible to the original kernel $k(\beta_1,\beta_2)$. We first consider the expression of the triangular kernel [5], $k(\beta_1,\beta_2)$ associated to $d(\beta_1,\beta_2)$ as follows:

$$k(\beta_1,\beta_2) = 1 - \frac{d(\beta_1,\beta_2)^2}{2} \tag{1}$$

With this expression of k, we derive the projection into a lower dimensional space that preserves most of the metric proprieties.

Let us consider a training set $\mathcal{A} = \{\beta_i\}$ used for the training of the projection $P(\cdot)$. We want P such that $\forall \beta_i,\beta_j \in \mathcal{A}$, $P(\beta_i)^\top P(\beta_j) = k(\beta_i,\beta_j)$. The Gram matrix of the kernel k on $\mathcal{A}$ is given by $K = [k(\beta_i,\beta_j)]_{\beta_i,\beta_j\in\mathcal{A}}$. We propose to approximate the matrix K by a low rank version and to compute the corresponding projection. This is known as the Nyström approximation for kernels, and has been used to speed up large scale kernel based classifiers in machine learning [1, 8]. Since k is a Mercer kernel and thus positive semi-definite, we can compute its eigen-decomposition: $K = V\Lambda V^\top$. The non-linear projection $P(\cdot)$ is then given by:

$$P(\beta) = \Lambda^{-\frac{1}{2}} V^\top [k(\beta_i,\beta)]_{\beta_i\in\mathcal{A}} \tag{2}$$

Where $[k(\beta_i,\beta)]_{\beta_i\in\mathcal{A}}$ is the vector of entry-wise computation of the kernel between β and the elements of the training set. Let us consider the matrix Y of the projected elements of $\mathcal{A}$: $Y = [P(\beta)]_{\beta\in\mathcal{A}}$. The Gram matrix in the projection space using the linear kernel is thus:

$$Y^\top Y = (\Lambda^{-\frac{1}{2}} V^\top K)^\top \Lambda^{-\frac{1}{2}} V^\top K = KV\Lambda^{-1}V^\top K = KK^{-1}K = K \tag{3}$$

Furthermore, the Euclidean distance between mapped elements perfectly fits to the original distance on the training set, $\forall \beta_i,\beta_j \in \mathcal{A}$:

$$\begin{aligned}
\|P(\beta_i) - P(\beta_j)\|^2 &= P(\beta_i)^\top P(\beta_i) + P(\beta_i)^\top P(\beta_i) - 2P(\beta_i)^\top P(\beta_j)\\
&= k(\beta_i,\beta_i) + k(\beta_j,\beta_j) - 2k(\beta_i,\beta_j)\\
&= 1 - \frac{d(\beta_i,\beta_i)^2}{2} + 1 - \frac{d(\beta_j,\beta_j)^2}{2} - 2 + 2\frac{d(\beta_i,\beta_j)^2}{2}\\
&= d(\beta_i,\beta_j)^2.
\end{aligned}$$

The more $\mathcal{A}$ is a good sampling of the original space, the closer the linearization P along with the dot product are to the original kernel k and thus the better the distance $d(\cdot,\cdot)$ is approximated. Some of the eigenvalues of the Kernel matrix are often very small compared to the other ones and can be safely discarded to further gain in computational efficiency. In this case, the reconstruction of the Gram matrix (respectively the distance matrix) on the training set is not perfect. However, the associated projectors often encode noise in the data, and discarding them can act as a de-noising process, as a PCA would do with linear projections. Since we are using a non-linear mapping, this process is analog to Kernel PCA [6].

We illustrate the effectiveness of the proposed method using three different experiments: shape retrieval from two databases and 3D face recognition application. The experiments show that approximated distances proposed in this paper achieves a very similar performance to the original elastic metric. The most important advantage is that the distance computing is reduced to a simple dot product on some projection space. This drastically reduces the running time of the computing process, since the non-linear computations have a complexity increasing linearly with the size of the collection (instead of quadratic with the original distance). Several applications can benefit from this work. This includes 3D face recognition, 3D biometrics and 3D shape retrieval and matching.

[1] Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research*, 6:2153–2175, 2005.

[2] S.H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn. Removing shape-preserving transformations in square-root elastic (sre) framework for shape analysis of curves. In *EMMCVPR*, pages 387–398, 2007.

[3] Y. Laurent. Computable elastic distance between shapes. *SIAM Journal of Applied Mathematics*, 58(2):565–586, 1998.

[4] P. W. Michor and D. Mumford. Riemannian geometries on spaces of plane curves. *J. Eur. Math. Soc.*, 8:1–48, 2006.

[5] B Schölkopf. The kernel trick for distances. TR MSR 2000-51, Microsoft Research, Redmond, WA, 2000. Advances in Neural Information Processing Systems, 2001.

[6] Bernhard Scholkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Advances in kernek methods - Support vector learning*, pages 327–352. MIT Press, 1999.

[7] Anuj Srivastava, Eric Klassen, Shantanu H. Joshi, and Ian H. Jermyn. Shape analysis of elastic curves in euclidean spaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(7), 2011. ISSN 0162-8828.

[8] Chris Williams and Matthias Seeger. Using the nystroem method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.