

Note

Click [here](#) to download the full example code

Elastic registration

Shows the usage of the elastic registration to perform a groupwise alignment.

```
# Author: Pablo Marcos Manchón
# License: MIT

# sphinx_gallery_thumbnail_number = 5

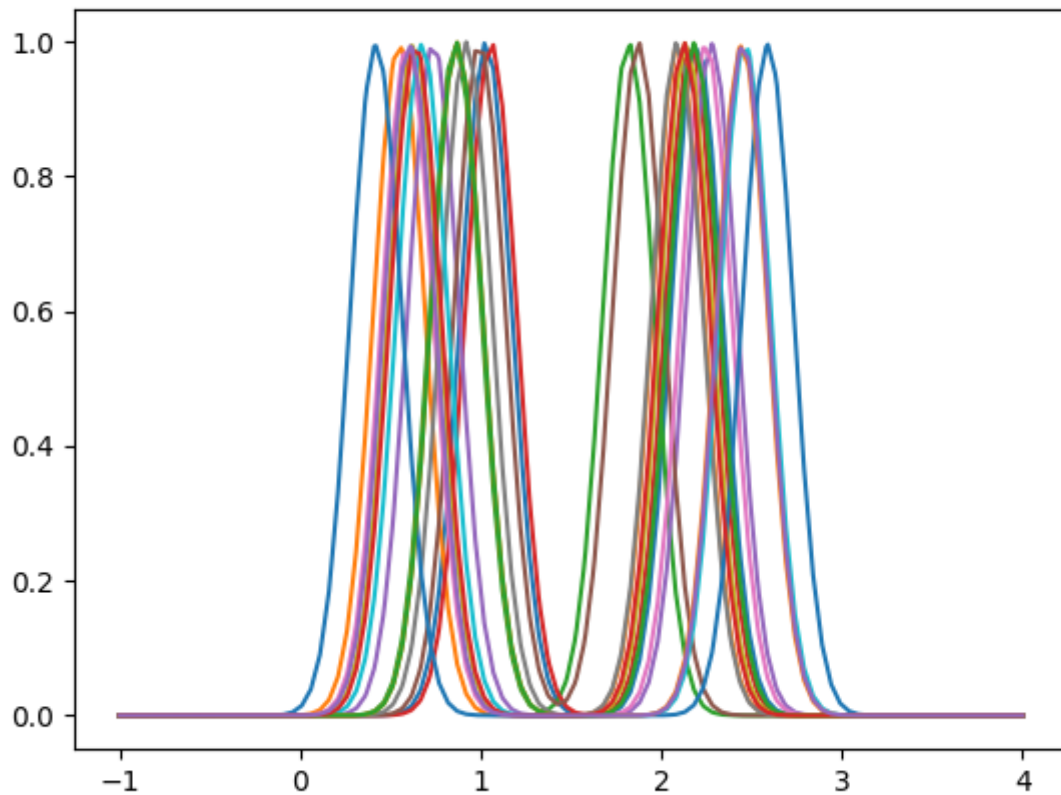
import skfda
import matplotlib.pyplot as plt
import numpy as np
```

In the example of pairwise alignment was shown the usage of `elastic_registration` to align a set of functional observations to a given template or a set of templates.

In the groupwise alignment all the samples are aligned to the same templated, constructed to minimise some distance, generally a mean or a median. In the case of the elastic registration, due to the use of the elastic distance in the alignment, one of the most suitable templates is the karcher mean under this metric.

We will create a synthetic dataset to show the basic usage of the registration.

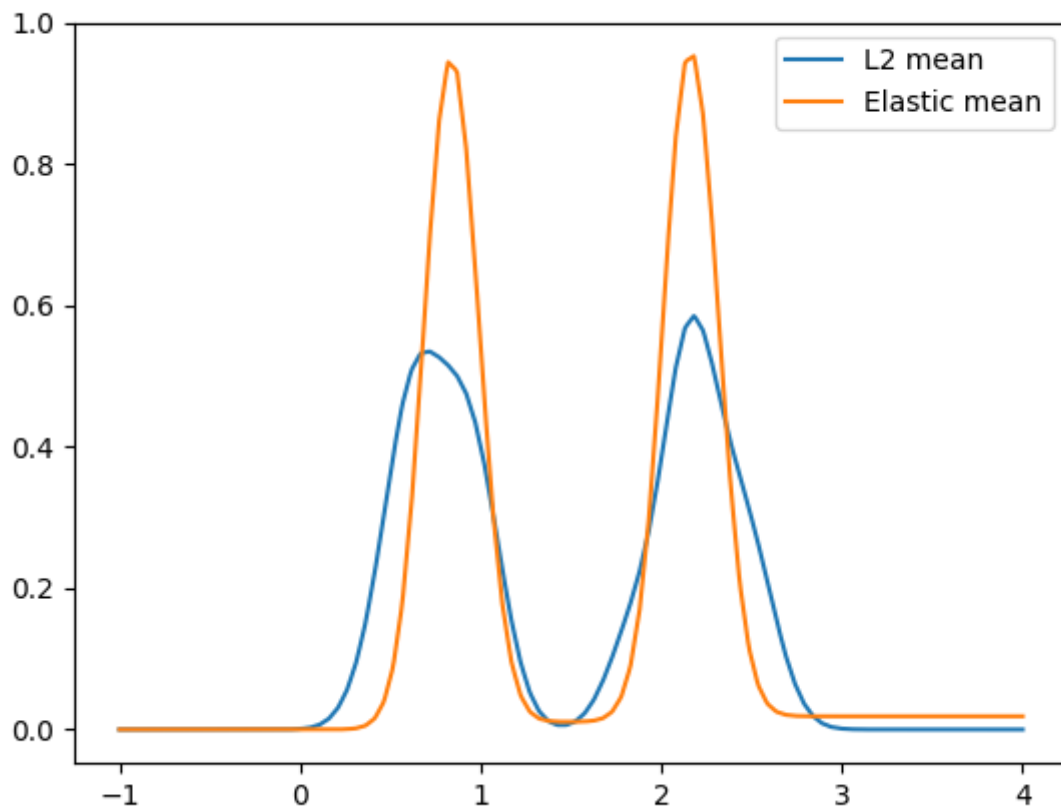
```
fd = skfda.datasets.make_multimodal_samples(n_modes=2, stop=4, random_state=1)
fd.plot()
```



The following figure shows the `elastic mean` of the dataset and the cross-sectional mean, which correspond to the karcher-mean under the \mathbb{L}^2 distance.

It can be seen how the elastic mean better captures the geometry of the curves compared to the standard mean, since it is not affected by the deformations of the curves.

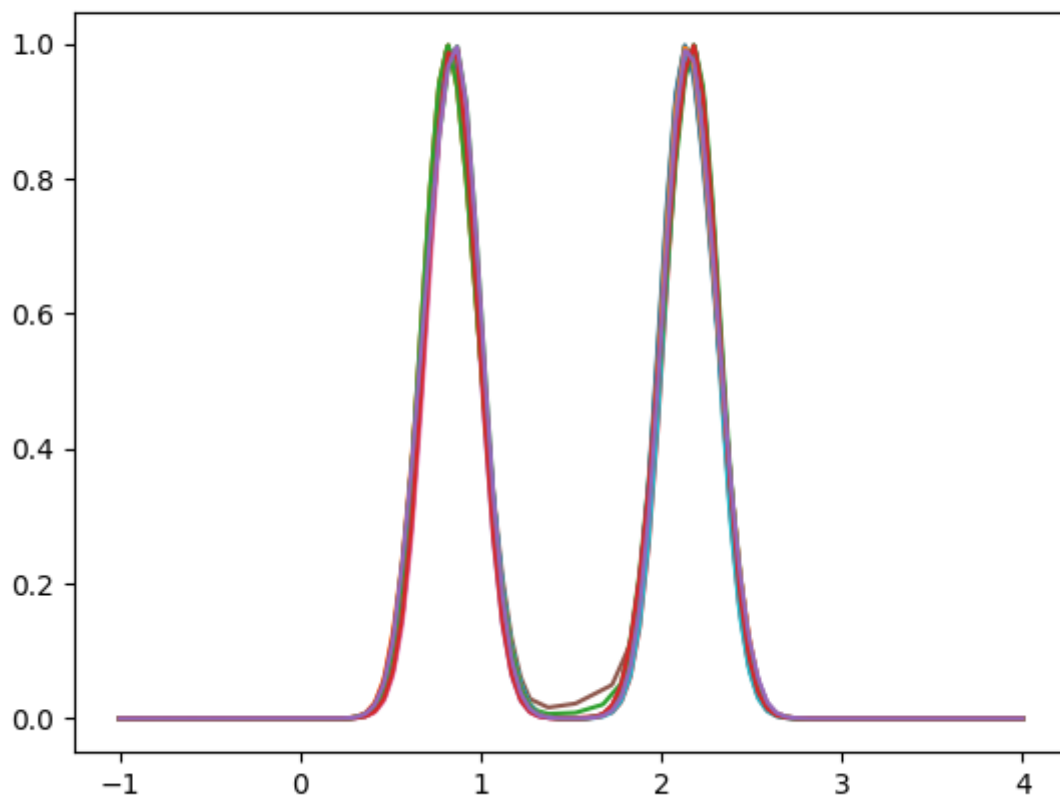
```
plt.figure()
fd.mean().plot(label="L2 mean")
skfda.preprocessing.registration.elastic_mean(fd).plot(label="Elastic mean")
plt.legend()
```



In this case, the alignment completely reduces the amplitude variability between the samples, aligning the maximum points correctly.

```
fd_align = skfda.preprocessing.registration.elastic_registration(fd)

plt.figure()
fd_align.plot()
```



In general these type of alignments are not possible, in the following figure it is shown how it works with a real dataset. The `berkeley growth dataset` contains the growth curves of a set of children, in this case will be used only the males. The growth curves will be resampled using cubic interpolation and derived to obtain the velocity curves.

```
growth = skfda.datasets.fetch_growth()

# Select only one sex
fd = growth['data'][growth['target'] == 0]

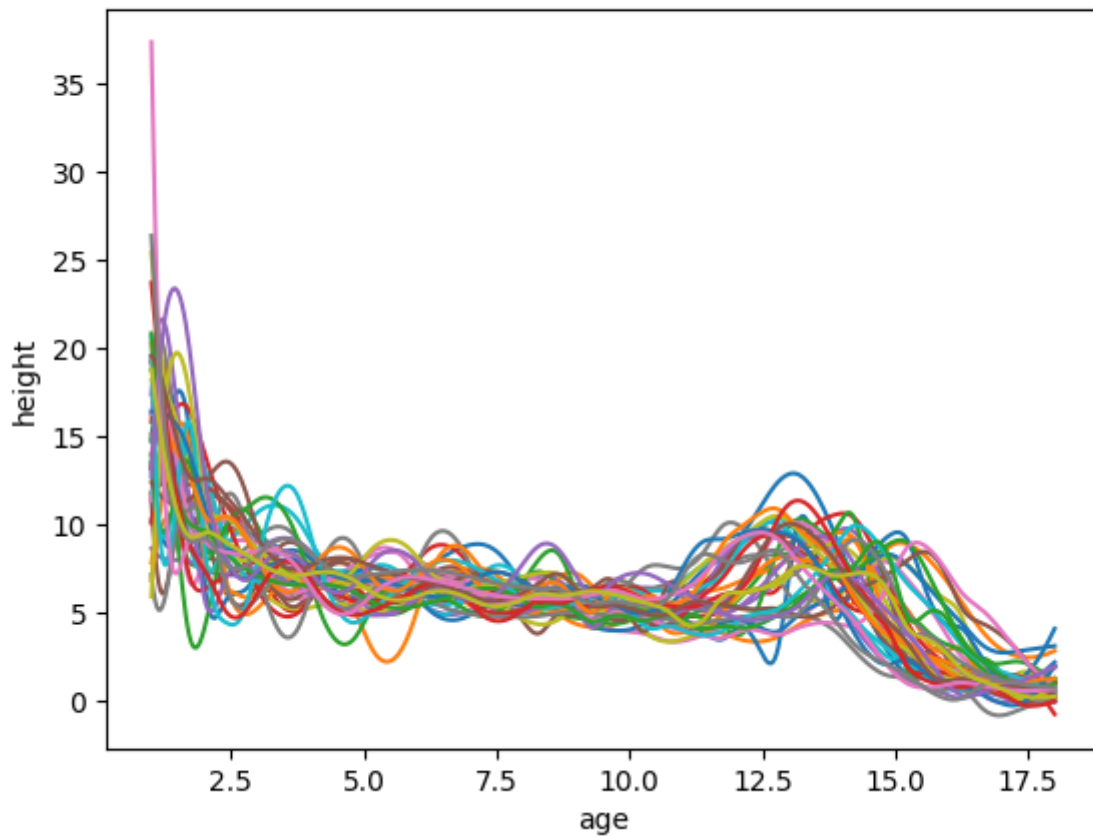
# Obtain velocity curves
fd.interpolator = skfda.representation.interpolation.SplineInterpolator(3)
fd = fd.to_grid(np.linspace(*fd.domain_range[0], 200)).derivative()
fd = fd.to_grid(np.linspace(*fd.domain_range[0], 50))
fd.plot()

plt.figure()
fd_align = skfda.preprocessing.registration.elastic_registration(fd)
fd_align.dataset_label += " - aligned"

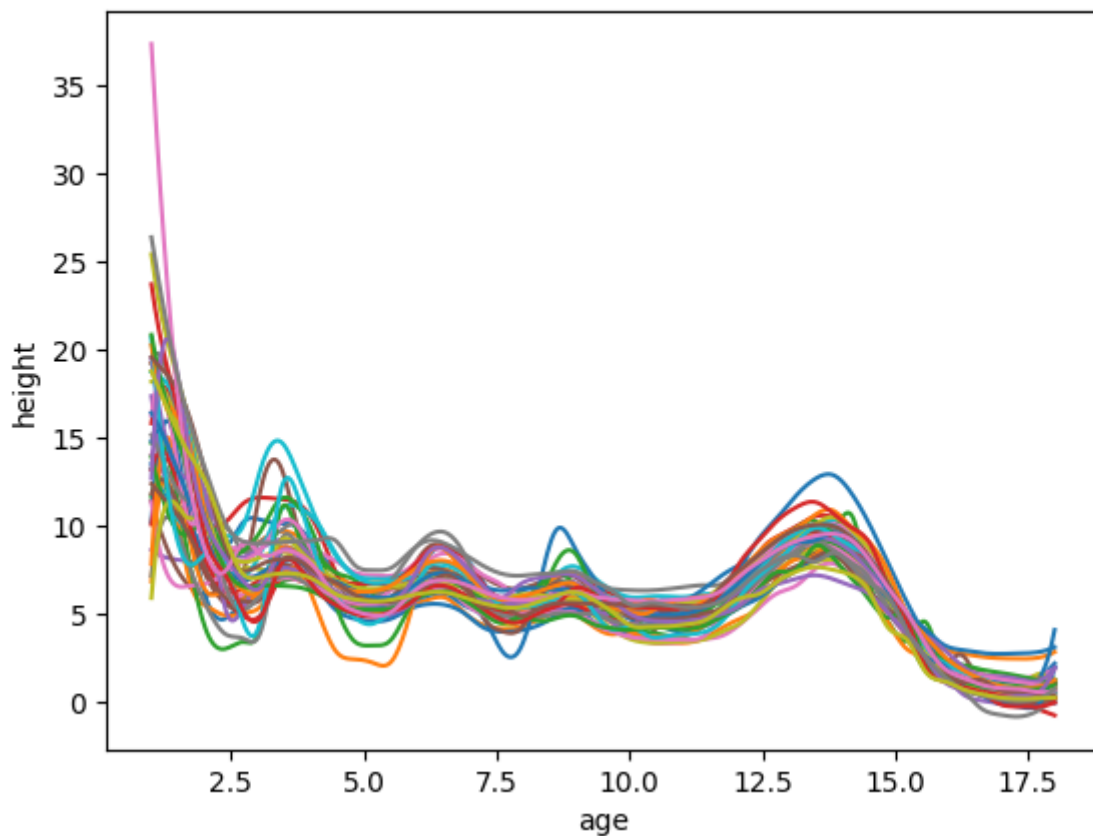
fd_align.plot()

plt.show()
```

Berkeley Growth Study - 1 derivative





Berkeley Growth Study - 1 derivative - aligned



- Srivastava, Anuj & Klassen, Eric P. (2016). Functional and shape data analysis. In *Functional Data and Elastic Registration* (pp. 73-122). Springer.
- J. S. Marron, James O. Ramsay, Laura M. Sangalli and Anuj Srivastava (2015). Functional Data Analysis of Amplitude and Phase Variation. *Statistical Science* 2015, Vol. 30, No. 4

Total running time of the script: (0 minutes 5.055 seconds)

 Download Python source code: `plot_elastic_registration.py`

 Download Jupyter notebook: `plot_elastic_registration.ipynb`

Gallery generated by Sphinx-Gallery