Note

Click here to download the full example code

Landmark shift

This example shows how to shift functional data objects to align its samples with a particular reference point.

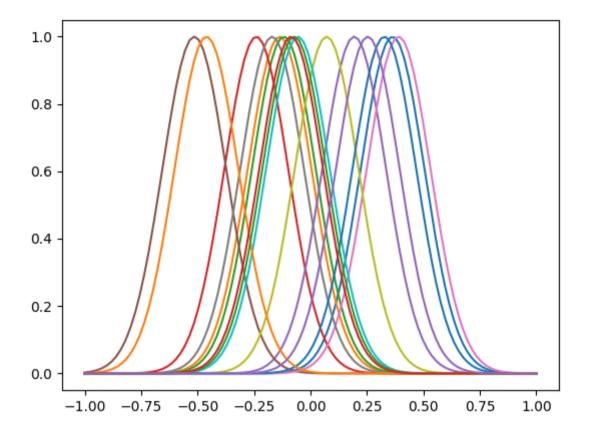
```
# Author: Pablo Marcos Manchón
# License: MIT

# sphinx_gallery_thumbnail_number = 2

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

We will use an example dataset synthetically generated by <code>make_multimodal_samples</code>, wich in this case will be used to generate gaussian-like samples with a mode near to 0. Each sample will be shifted to align their modes to a reference point using the function <code>landmark_shift</code>.

```
fd = skfda.datasets.make_multimodal_samples(random_state=1)
fd.extrapolation = 'bounds' # See extrapolation for a detailed explanation.
fd.plot()
```

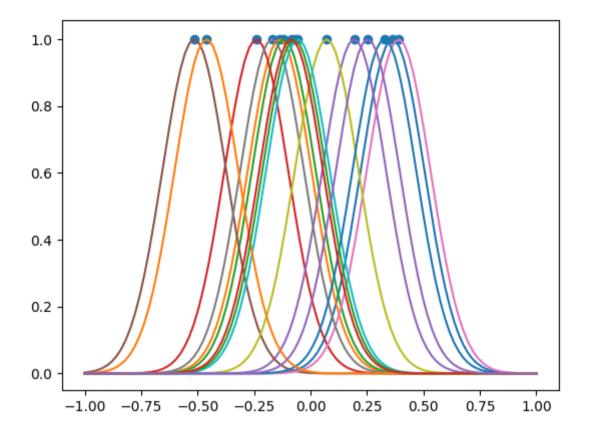


A landmark or a feature of a curve is some characteristic that one can associate with a specific argument value t. These are typically maxima, minima, or zero crossings of curves, and may be identified at the level of some derivatives as well as at the level of the curves themselves. [1]

For alignment we need to know in advance the location of the landmark of each of the samples, in our case it will correspond to the maxima of each sample. Because our dataset has been generated synthetically we can obtain the value of the landmarks using the function <code>make_multimodal_landmarks</code>, which is used by <code>make_multimodal_samples</code> to set the location of the modes.

In general it will be necessary to use numerical or other methods to determine the location of the landmarks.

```
landmarks = skfda.datasets.make_multimodal_landmarks(random_state=1).squeeze()
plt.figure()
plt.scatter(landmarks, np.repeat(1, fd.nsamples))
fd.plot()
```



Location of the landmarks:

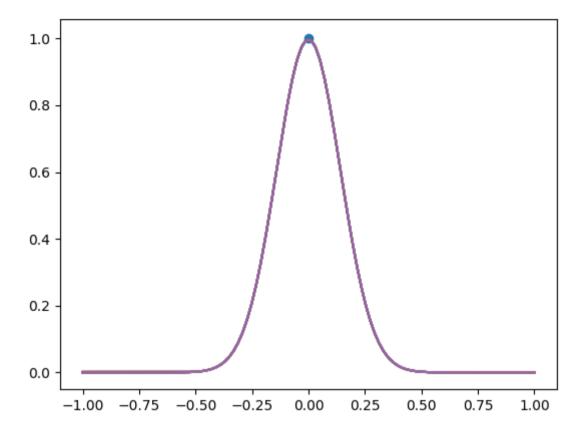
```
print(landmarks)
```

Out:

The following figure shows the result of shifting the curves to align their landmarks at 0.

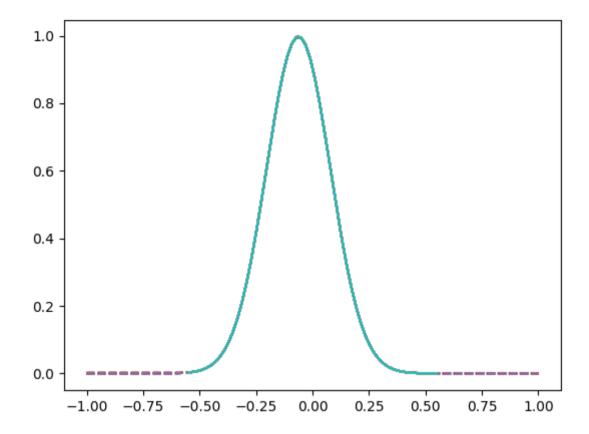
```
fd_registered = skfda.preprocessing.registration.landmark_shift(fd, landmarks,
location=0)

plt.figure()
fd_registered.plot()
plt.scatter(0, 1)
```

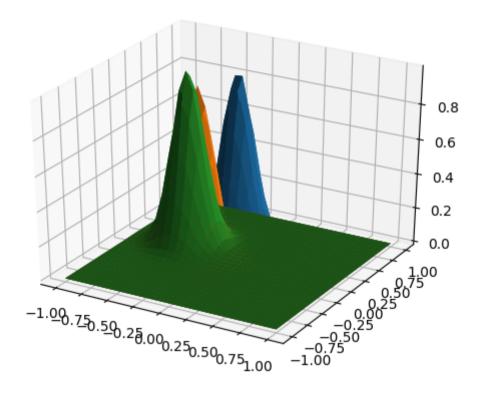


In many circumstances it is possible that we could not apply extrapolation, in these cases it is possible to restrict the domain to avoid evaluating points outside where our curves are defined.

If the location of the new reference point is not specified it is choosen the point that minimizes the maximum amount of shift.



The previous method is also applicable for multidimensional objects, without limitation of the domain or image dimension. As an example we are going to create a datset with surfaces, in a similar way to the previous case.



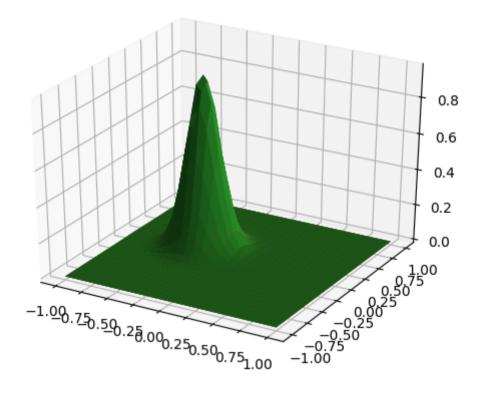
In this case the landmarks will be defined by tuples with 2 coordinates.

Out:

```
[[ 0.36321467 -0.13679289]
[-0.11810279 -0.23992308]
[ 0.19351103 -0.5146397 ]]
```

As in the previous case, we can align the curves to a specific point, or by default will be chosen the point that minimizes the maximum amount of displacement.

```
fd_registered = skfda.preprocessing.registration.landmark_shift(fd, landmarks)
fd_registered.plot()
plt.show()
```



[1] Ramsay, J., Silverman, B. W. (2005). Functional Data Analysis. Springer.

Total running time of the script: (0 minutes 1.899 seconds)

▲ Download Python source code: plot_landmark_shift.py

▲ Download Jupyter notebook: plot_landmark_shift.ipynb

Gallery generated by Sphinx-Gallery