skfda.preprocessing.registration.landmark_registration

skfda.preprocessing.registration.landmark_registration(*fd, landmarks, *, location=None, eval_points=None*) [source]

Perform landmark registration of the curves.

Let t_{ij} the time where the sample i has the feature j and t_j^* the new time for the feature. The registered samples will have their features aligned, i.e., $x_i^*(t_j^*) = x_i(t_{ij})$.

See [RS05-7-3] for a detailed explanation.

Parameters:
- **fd** (`FData`) – Functional data object.
- **landmarks** (*array_like*) – List containing landmarks for each samples.
- **location** (*array_like*, *optional*) – Defines where the landmarks will be alligned. By default it will be used as location the mean of the landmarks.
- **eval_points** (*array_like*, *optional*) – Set of points where the functions are evaluated to obtain a discrete representation of the object. In case of objects with multidimensional domain a list axis with points of evaluation for each dimension.

Returns: FData with the functional data object registered.

Return type: `FData`

References:

[RS05-7-3] Ramsay, J., Silverman, B. W. (2005). Feature or landmark registration. In *Functional Data Analysis* (pp. 132-136). Springer.

Examples

```
>>> from skfda.datasets import make_multimodal_landmarks
>>> from skfda.datasets import make_multimodal_samples
>>> from skfda.preprocessing.registration import landmark_registration
>>> from skfda.representation.basis import BSpline
```

We will create a data with landmarks as example

```
>>> fd = make_multimodal_samples(n_samples=3, n_modes=2, random_state=9)
>>> landmarks = make_multimodal_landmarks(n_samples=3, n_modes=2,
...                                       random_state=9)
>>> landmarks = landmarks.squeeze()
```

The function will return the registered curves

```
>>> landmark_registration(fd, landmarks)
FDataGrid(...)
```

This method will work for FDataBasis as for FDataGrids

```
>>> fd = fd.to_basis(BSpline(nbasis=12, domain_range=(-1,1)))
>>> landmark_registration(fd, landmarks)
FDataBasis(...)
```