skfda.preprocessing.registration.shift_registration_deltas

skfda.preprocessing.registration.shift_registration_deltas(*fd*, ***, *maxiter=5*, *tol=0.01*, *restrict_domain=False*, *extrapolation=None*, *step_size=1*, *initial=None*, *eval_points=None*) [source]

Return the lists of shifts used in the shift registration procedure.

Realizes a registration of the curves, using shift aligment, as is defined in [RS05-7-2-1]. Calculates δ_i for each sample such that $x_i(t + \delta_i)$ minimizes the least squares criterion:

$$\text{REGSSE} = \sum_{i=1}^{N} \int_{\mathcal{T}} [x_i(t + \delta_i) - \hat{\mu}(t)]^2 ds$$

Estimates the shift parameter δ_i iteratively by using a modified Newton-Raphson algorithm, updating the mean in each iteration, as is described in detail in [RS05-7-9-1-1].

Method only implemented for Funtional objects with domain and image dimension equal to 1.

Parameters:

- **fd** (`FData`) – Functional data object to be registered.
- **maxiter** (*int*, *optional*) – Maximun number of iterations. Defaults to 5.
- **tol** (*float*, *optional*) – Tolerance allowable. The process will stop if $\max_i |\delta_i^{(\nu)} - \delta_i^{(\nu-1)}| < tol$. Default sets to 1e-2.
- **restrict_domain** (*bool*, *optional*) – If True restricts the domain to avoid evaluate points outside the domain using extrapolation. Defaults uses extrapolation.
- **extrapolation** (str or `Extrapolation` , optional) – Controls the extrapolation mode for elements outside the domain range. By default uses the method defined in fd. See :module: *extrapolation* to obtain more information.
- **step_size** (*int or float*, *optional*) – Parameter to adjust the rate of convergence in the Newton-Raphson algorithm, see [RS05-7-9-1-1]. Defaults to 1.
- **initial** (*array_like*, *optional*) – Initial estimation of shifts. Default uses a list of zeros for the initial shifts.
- **eval_points** (*array_like*, *optional*) – Set of points where the functions are evaluated to obtain the discrete representation of the object to integrate. If None is passed it calls numpy.linspace in FDataBasis and uses the *sample_points* in FDataGrids.

Returns: list with the shifts.

Return type: `numpy.ndarray`

Raises: `ValueError` – If the initial array has different length than the number of samples.

Examples

```
>>> from skfda.datasets import make_sinusoidal_process
>>> from skfda.representation.basis import Fourier
>>> from skfda.preprocessing.registration import shift_registration_deltas
>>> fd = make_sinusoidal_process(n_samples=2, error_std=0, random_state=1)
```

Registration of data in discretized form:

```
>>> shift_registration_deltas(fd).round(3)
array([-0.022,  0.03 ])
```

Registration of data in basis form:

```
>>> fd = fd.to_basis(Fourier())
>>> shift_registration_deltas(fd).round(3)
array([-0.022,  0.03 ])
```

References

[RS05-7-2-1] Ramsay, J., Silverman, B. W. (2005). Shift registration. In *Functional Data Analysis* (pp. 129-132). Springer.

[RS05-7-9-1-1] *(1, 2)* Ramsay, J., Silverman, B. W. (2005). Shift registration by the Newton-Raphson algorithm. In *Functional Data Analysis* (pp. 142-144). Springer.