EFDA_warps_visual_continuous_asymm(varargin)

Description

This function visualizes the impact of various asymmetric warping functions on a one-dimensional signal. The warping functions are generated using a modified version of the approach proposed by Tucker (2014), with parameters determined empirically. The function focuses on asymmetric warps, meaning they reside entirely below or above the identity line (t=t) in the warping function plot. Such “asymmetric” warp may be relevant for the data transformation skewed towards earlier or later times.

Input Arguments

- varargin (optional):
  - 'CloseFig': Closes the figure window before plotting (default: false).
  - 'SaveFig': Saves the generated figure (default: false).

Output

The function generates a figure containing several subplots that visualize:

- The original two-peaked Gaussian signal.
- The warping functions (psi) for each case.
- The warped versions of the original signal.
- The derivatives of the original and warped signals (optional).
- The integrals of the original and warped signals (optional).
- The timespeed (warping function derivative) and timeshift (warping function difference from identity) for each case (optional).

Notes

- The original signal is a bimodal Gaussian distribution with user-defined parameters for the means and standard deviation.
- The warping function generation utilizes empirically determined parameters for the genwarp function, which might be further refined for specific applications.
- The function calculates and displays the L2 norm (integral of squared values) and area (integral) of the original and warped signals.

EFDA_warps_visual_continuous_symm(varargin)

Description

This function visualizes the impact of various symmetric warping functions on a one-dimensional signal. The warping functions are generated using a modified version of the approach proposed by Tucker (2014), with parameters determined empirically. The function focuses on warping functions that reside both above and below the identity line (t=t) in the warping function plot. Such "symmetric" warp may be relevant for the data with a centered peak or a localized compression/expansion effect around the center.

Input Arguments

- varargin (optional):
  - 'CloseFig': Closes the figure window before plotting (default: false).
  - 'SaveFig': Saves the generated figure (default: false).

Output

The function generates a figure containing several subplots that visualize:

- The original two-peaked Gaussian signal.
- The warping functions (psi) for each case.
- The warped versions of the original signal.
- The derivatives of the original and warped signals (optional).
- The integrals of the original and warped signals (optional).
- The timespeed (warping function derivative) and timeshift (warping function difference from identity) for each case (optional).

Notes

- The original signal is a bimodal Gaussian distribution with user-defined parameters for the means and standard deviation.
- The warping function generation utilizes empirically determined parameters for the genwarp function, which might be further refined for specific applications.
- The function calculates and displays the L2 norm (integral of squared values) and area (integral) of the original and warped signals.

EFDA_warps_visual_piecewise(varargin)

Description

This function demonstrates the impact of various piecewise linear warping functions on a one-dimensional signal. The warps are constructed using three to four knots positioned at predefined locations, with linear segments connecting them. This visualization approach aids in understanding the concept of warping, particularly for those new to the technique. This technique is conceptually similar to piecewise time-normalization (see e.g. Helwig et al (2011) in *J Biomech* or Russo et al (2020) in *Neuron*), allowing manual selection of landmarks and avoiding non-linear time-warping, but producing discontinuities.

Input Arguments

- varargin (optional):
  - 'CloseFig': Closes the figure window before plotting (default: false).
  - 'SaveFig': Saves the generated figure (default: false).

Output

The function generates a figure containing several subplots:

- The original two-peaked Gaussian signal.
- The warping functions for each case (except for compression and stretching, which are visualized through interpolation).
- The warped versions of the original signal.
- The derivatives of the original and warped signals (for piecewise warps with internal time shifts).
- The integrals of the original and warped signals (for piecewise warps with internal time shifts).
- The timespeed (warping function derivative) and timeshift (warping function difference from identity) for each case with internal time shifts.

Notes

- The original signal is a bimodal Gaussian distribution with user-defined parameters for the means and standard deviation.
- The warping functions are constructed with predefined knot positions and linear interpolation between them. Parameters for the slopes and intercepts of each linear segment are specified within the function.
- The function includes examples of warps that compress, stretch, and introduce localized time shifts within the signal.

[psii, gam, shvec] = genwarp(A, f, time, varargin)

Description

Generates a warping function based on specified parameters. The function utilizes a shooting vector approach from Tucker (2013) in *Comp Stat and Data Anal* to create the warping function. The shooting vector, when no noise is specified, follows a sinusoidal profile.

Input Arguments

- A: Amplitude of the shooting vector, controlling the overall magnitude of the warp. Values range from 0 (identity warp) to 1 (maximum warp).
- f: Frequency of the shooting vector, influencing the number of oscillations in the warping function.
- time: Time vector for the warping function.
- varargin (optional):
    - 'InverseFlag': If set to 1, inverts the warping function by adding pi to the phase shift.
    - 'Shift': Phase shift of the shooting vector. Recommended 0, and ±π/2
    - 'PsiNoiseSigma': Standard deviation of additive Gaussian noise applied to the psi function.
    - 'PsiNoiseSigmaMulti': Standard deviation of multiplicative Gaussian noise applied to the psi function.

Output Arguments

- psii: The generated warping function.
- gam: The cumulative integral of psii^2, representing the warping path.
- shvec: The shooting vector used to generate the warping function.

Details

The function generates a warping function by:

1. Creating a shooting vector using sinusoidal function with specified amplitude, frequency, phase shift, and potential inversion.
2. Applying the exponential map to the shooting vector to obtain the psi function.
3. Adding optional Gaussian noise to the psi function.
4. Calculating the cumulative integral of psi^2 to obtain the warping path gam.

Notes

- The function provides a flexible way to generate various warping function shapes by adjusting the A, f, and shift parameters.
- A combination of arguments InverseFlag and Shift results in the warping function’s location under, below, or around the identity (t=t) and its behavior at the boundaries, specifically whether it is parallel to the identity.
- The PsiNoiseSigma and PsiNoiseSigmaMulti parameters introduce noise to the warping function for potential regularization or simulation of variability.

EFDA_conceptDemos()

Description

Generates two double-Gaussian signals and demonstrates time-warping alignment using the toolbox *fdasrvf* of Srivastava, Tucker and others. Displays the original signals, and produces three automatic alignments, where one signal is aligned to the other one, or both signals are aligned to an emerging Karcher mean. Displays Euclidean and Fisher-Rao distances, the latter also for the warping functions, between the signals before and after alignment. This function was used for some of the figures in the manuscript submitted by Krotov, S. Razavian, Sadeghi, and Sternad (2024).

Notes

Use the flag ShowSRRF to plot SRRF instead of the original signals and observe their change in both amplitude and temporal domains when the original signal is warped.

Use the flag EmulateL2 to use EFDA alignment as if L2-distance between the original signals was minimized, instead of Fisher-Rao distance.

Use the flag EmulateDTW to use MATLAB’s function dtw to align the signals, which likely results in strong discontinuities after warping.

EFDA_GaussModel()

Description

A comprehensive tool for generating and analyzing synthetic data to evaluate the performance of different signal processing methods, specifically alignment and resampling techniques to extract meaningful information from noisy data. A triple-Gaussian signal is used as an example, with the peak parameters specified in the second sub-function, and optional noise values (also a Gaussian noise, added to one or few of the parameters) are specified in the third sub-function. This function was used for some of the figures in the manuscript submitted by Krotov, S. Razavian, Sadeghi, and Sternad (2024).

1) N (default 500) noisy realizations of the original triple-Gaussian signal are generated; Each signal is trimmed with on/offset at 2% of the signal’s maximum imitating an experimental measurement. Peaks are found and characterized (peak value, peak location, peak width at half-maximum).

2) Three approaches are then used on the whole ensemble to extract time-series of mean and standard deviation. Time-padding approach aligns all the signals at their start (trimmed onset) and pads all but the longest signal with NaNs. Time-normalizing approach resamples all signals to the same number of samples, effectively rescaling time to a relative scale. Time-warping approach performs similar time-normalization, and then performs simultaneous time-warping alignment via fdasrvf.time_warp method by Tucker (2014).

3) After each of the three approaches, mean and SD are extracted. The results are displayed. Peaks are identified and characterized. Their properties are compared against “ground truth” obtained from the distributions from (1). These distributions and the estimate after alignment are displayed as histograms. The resulting differences (errors of estimation from the mean) are additionally displayed as horizontal and vertical barcharts.

4) For the time-warping approach, estimated warping functions are plotted, against with “TimeShift” where identity was subtracted, and “TimeSpeed”, a derivative of warping function. Additionally, the extracted mean warped with every warping function is plotted as another ensemble demonstrating contribution of the temporal (warp-exposed) variability to the “spatial”, signal domain.

5) Variabilities are displayed as RMSE across aligned ensembles from (2), with additional temporal and temporal-to-spatial components from the time-warping approach.

Notes

Explore the NoiseOption parameter and combinations of noise imposed on the noisy signal ensemble.

Explore the rangeSet parameter to scale the amount of corresponding noise imposed and relate that noise to post-alignment parameter estimation error and variabilities.

EFDA_Whip_Average_ShowVar(data,SubjType,AverMethod)

Description

Processes and analyzes hand speed data from a motor neuroscience experiment of hitting a target with the bullwhip (Krotov & Russo 2022 in *RSOS* and Henrot B.S. MIT Thesis 2018). It takes experimental data provided along, a subject type (Novice or Expert), and an averaging method as input. The function aligns time-series data using one of four methods (TimePadLeft, TimePadRight, TimeNorm, EFDA), calculates mean and standard deviation, compares peak hand speed estimates, and visualizes spatial and temporal variability. This function was used for some of the figures in the manuscript submitted by Krotov, S. Razavian, Sadeghi, and Sternad (2024).

Notes

Change Nresamp to a lower of a higher value to specifically monitor the effect of sampling rate to time-warping alignment

Toggle the flag PreserveIntegral for preserving the total distance travelled (the original signals are speed).

Toggle flag NormByPeakHS for further reducing amplitude differences between the subjects.

EFDA_alignmentPublishing(X,varargin)

Description

Performs time-warping alignment of multiple time series using the time-warping alignment (EFDA) algorithm. It offers various options for preprocessing, alignment, and post-processing, including filtering, resampling, outlier removal, and visualization.

Inputs

X: A cell array of time series data. Each cell contains a Tx2 or Tx1 matrix, where T is the number of time points and the second column (if present) represents the time stamps.

varargin: Optional input arguments for specifying function parameters. Can be passed as a structure or as individual arguments.

Outputs

EFDAResult: A structure containing the alignment results, including warped time series, warping functions, and other relevant information.

opts: the options structure detailing customizable options. (if no opts or keywords supplied, use that output to see the default settings).

Details:

Pass a structure opts to modify the default settings, or pass keywords for enabling binary-flag settings.

EFDAQuick: If set to 1, performs alignment on a subset of the data, while also downsampling it, to estimate the emerging mean; Then the remaining warping functions are found. Expected time on a single thread 2 seconds. Use for large datasets where M * N^2 > 10^6, for M signals of N samples.

EFDAGraphics: If set to 1, visualizes results. Default: 1.

PreserveIntegral: If set to 1, rescales aligned signals to preserve the integral. Experimental. Default: 0.

FSResampleTarget: Target sampling frequency for resampling. Default: 20 Hz.

NSamplesTarget: Target number of samples for resampling (overrides FSResampleTarget). Default: NaN.

RemoveLongOutliers: If set to 1, removes trials with duration significantly different from the mean. Default: 1.

Filter: Specifies the filter type (e.g., 'Butter', 'SavGol') and parameters. Default: empty.

Quiet: If set to 1, suppresses output messages and graphics. Default: 0.

EFDAMethod: EFDA method to use (e.g., 'DP'). Default: 'DP'.

EFDALambda: Warp roughness penalty for EFDA. Default: 0.01.

EFDAParallel: If set to 1, uses parallel computation. Default: 1.

EFDAPreserving: Specifies whether to preserve values, derivatives, or integrals during warping. Experimental. Default: 'values'.

EFDAMaxIter: Maximum number of EFDA iterations. Default: 2.

EFDAWarpsWithDurs: If set to 1, resamples normalized time to match the mean duration. Default: 0.

MaxNanInterp: Maximum fraction of NaN values allowed for interpolation. Default: 0.1 (10%) per signal

NSampPlot: Number of samples to plot. Default: 150.

NLinesPlotMax: Maximum number of lines to plot. Default: 90.

EstimateExecutionTime: If set to 1, estimates the execution time of EFDA. Default: 1.