Kernel Smoothing Package

Jeffrey Hetherly

June 12th 2015

Keywords

statistics, smoothing

# Overview

This package is an implementation of a local polynomial kernel estimator (LPKE) along with utilities for smoothing a CDI file and performing an eigen-variation analysis. The package contains a generic implementation of an LPKE that can be customized to any polynomial order, domain dimensionality, numeric type, kernel function, and coordinate scaling. It also provides a convenience class for loading 1-D ROOT data from a TH1 or TGraphErrors.  
This implementation of an LPKE is customized for differently-weighted data. Specifically, the LPKE here expects each data point to have an associated (symmetric) error, . This is used to multiply the kernel by .  
The utilities for smoothing a CDI file and performing the eigen-variation (EV) analysis are meant to aid in production of the smoothed CDI calibrations file and perform a complete EV analysis to help decide the appropriate number of EV to keep, respectively. They should be run from the command line as standalone executables and are not ROOT macros.  
These utilities allow for some customizability, but in general will require modifications to their respective source codes for more realistic scenarios. The utility for smoothing a CDI file allows for command-line specification of the file to smooth, the number of bins to store, and the (global) bandwidth. To specify which data sets to smooth, order of LPKE, etc. one needs to look at smoothCalibrations.cxx. The utility for the EV analysis can be customized at the command line to specify the required files, number of desired EV, and which data set to look analyze.

# Mathematical Detail

## Local Polynomial Kernel Estimator

A general, multivariate local polynomial kernel estimator of order , with bandwidth matrix , domain dimension , and constructed from data points has the form:

where,

For this implementation of the LPKE we use a simplified bandwidth matrix:

Thus, the kernel here is a function of the "scaled" coordinates:

As mentioned in the [overview](#OV) this package provides a convenience class for interfacing with data stored in a ROOT TH1 and TGraphErrors. This class is restricted to order 1 and a domain dimension of 1 (1-D histograms only). This form of the estimator is called a local linear kernel estimator and has better behavior at the ends of the data than a local constant kernel estimator (order 0). (All odd orders have better performance at the end points than even powers)

Lastly, note that the form of the estimator has the following, seemingly insignificant, property. Let be associated with . Now shift the by so that is the estimator associated with the "shifted" data. Notice that . That is to say, the estimator of the "shifted" data less the estimator for the original data is simply the estimator associated with the shifts themselves, . This is useful in nuisance variations where the nominal distribution is shifted and normally one would need to perform two LPKEs and a subtraction. This can introduce undesirable numerical effects. Fortunately, one can skip that procedure altogether and simply perform an LPKE on the 's themselves.

## Eigen-Variation Reduction

Each scale factor distribution in a CDI data set has several (~40) systematic variations as well as statistical variations that need to be taken into account in order to properly access the uncertainty of the distribution. For a physics analysis this can lead to a large number of nuisance parameter when it comes time to asses the affects of various systematics. Therefore, it is highly desirable to somehow reduce the number of systematics under consideration if possible.  
One method of reducing the number systematics that preserves the bin-to-bin correlations and total error is to perform an eigenvalue decomposition on the covariance matrix of systematic and statistical variations. It is clear that the resulting number of variations will simply be the number of bins in the scale factor distribution. This is already a huge reduction of systematics to consider if your distribution has at most 10 bins. However, after smoothing the number of bins can far exceed the original number of systematics. Initially, this can seem counterproductive and another method should be pursued. Fortunately most of the EV are very small in magnitude and can be thrown out without fear of losing correlations or total error. In principle one should recover the original number of "significant" EV after this pruning. The remaining EV can be further reduced by summing up the least significant EV (ranked by eigenvalue). Whatever summing technique is used, the over goal is to try to satisfy the following constraint on the newly summed histogram ():

This condition may only be satisfied if one of two criteria are met:

1. The are vectors with elements
2. The are scalars but the conspire in such a way that makes this equality hold

Unfortunately, neither of these criteria are applicable for our case. Thus, our essential problem is that of how one can store the information of a vector in a single scalar. Using the (false) equality above one can set and take the positive square root to yield:

Using this equation () preserves the total errors in the resulting covariance matrix. However, because all the information about the sign of the original systematics is lost this method of merging severely distorts the correlations. Another manipulation of the (false) equality is to first sum over , sum over , then finally back substitute to solve for :

Thus,

In principle this should be equal to the first method of merging. But, because the original equation is false they give different results. This alternative method of summing () the lowest eigen-variations is found empirically to better preserve the correlations over the method. However, doesn't preserve the total error as does. Thus, a trade off must be made as to how much total error can be sacrificed for extra gains in correlations.  
As a final note on EV reduction, another way of attempting to satisfy the (false) equality is to perform a multidimensional minimization against some metric treating each bin of the merged histogram () as the variables to minimize. An implementation of this using the Amoeba routine was explored, but yielded results similar to using . Thus, it is not currently being used.

# Implementation

## LPKE

The LPKE implemented in this package (LocalPolyKernelEstimator) attempts to be as generic as possible. It is written in idiomatic C++11 and requires a compatible ROOT version (version 6 or version 5 with the c++11 flag enabled). The LocalPolyKernelEstimator class allows for any domain dimension, order of estimator, coordinate scaling, or kernel function (albeit the kernel must be a function of an array of normalized coordinates). It is a straightforward and naive implementation with respect to computational efficiency and numerical stability. No effort is made to reduce the number of kernel evaluations with a nearest-neighbors-based pruning or ensure that the minimum number of numerically unstable operations are performed. However, for our applications of this LPKE this doesn't appear to be an issue.  
The coordinate

## Eigen-Variation Analysis

# Usage

## LPKE

## Eigen-Variation Analysis