Minimal Empirical Density Estimation

Matthew Leonawicz

December 24, 2014

1 Introduction

Here I explore basic kernel density estimation in **R** as a way to empirically estimate densities for data extracted from common SNAP data sets. This is just a toy example where I use simple simulations to investigate properties I am most interested in for the estimated distributions.

1.1 Motivation

I often face the real-world problem of needing to rapidly summarize large amounts of data in an efficient manner without losing, skewing, or obscuring too much information. The context for this simulation is to show that it is possible to estimate an empirical distribution of a variable by a small set of points roughly defining the density curve, which can then be used in conjunction with linear approximation and bootstrap resampling to simulate new draws from the estimated distribution. The process can stop here if it is a sample that is required, or, in turn, an arbitrarily large sample drawn can be used to re-estimate the density more smoothly.

For some applications this is not useful, as the original data are already available. Their distribution can be estimated in the most appropriate manner the first time around. However, this delay in the chain of data propagation is extremely helpful when I am trying to provide real-time summaries of large data sets on demand, the quintessential example being my **R** Shiny web applications.

Particularly, in the case of trying to rapidly summarize and graph full distributional information, as opposed to, say, a time series of mean values, some form of data reduction must take place upstream from the web application. At SNAP, we have long time series of high spatial resolution data that are too much to dump into a simple web application. Data can easily be externalized and specific data sets sourced by an app on demand, but it is especially helpful to avoid the crippling load times associated with forcing massive amounts of data into an app while a user is trying to interact with it.

Instead of loading a dataset, e.g., an R workspace file, containing a huge sample of data for some variable, it is much easier to:

* store a small, efficient, and hopefully, accurate and precise representation of that data, * load only that into the Shiny app, * and then have **R** quickly explode that representation into a new simulated data set.

This is especially an effective approach in the context of Shiny apps where the goal is to visually explore patterns and present information, and not to have tunnel vision for an exact value buried deep in a massive data set which is nevertheless riddled with uncertainty.

1.2 Details

Currently a simple simulation is shown, followed by two typical use cases in which application of such a strategy proves very beneficial. For the use cases, **R** code is provided to show how this is done in practice with real data inthe context of actual projects.

1.2.1 Limitations

The simulation is obviously not a robust analysis. Rather it is intended as a simple illustration of the process being utilized for data reduction in specific applications.

2 Related items

2.1 Files and Data

There are no files or data related to the simulation itself. It is self-contained and reproducible. For the use cases, references are made to other projects which can be explored further.