Location-dependent Weighted Average of Temperature

Jared Callaham

June 28, 2017

1 Continuous Case

This is the easiest to handle mathematically, even though the data isn't actually set up like this. Define a coordinate system with the recorded collection location at the origin (so when I say "origin" I mean "reported coordinates of collection"). Then there is a radius σ of uncertainty associated with this location. Suppose we also have temperature $T(\vec{r})$ as a continuous function of location $\vec{r}$.

Next we have to specify some probability distribution function $\rho(\vec{r})$ for the location of collection as a function of location. In other words, we decide how to formally interpret "coordinate uncertainty in meters." Note that we don't have to make any assumptions about the underlying distribution of temperatures, only of collection location. So without any reason to think otherwise, let's say the probability of collection at some distance from the origin is radially symmetric, so $\rho(\vec{r}) = \rho(r)$. For the form of the function, two simple choices are:

1. Uniform distribution with radius σ - specimen could have been collected anywhere within σ meters of the origin with equal likelihood
2. Normal distribution with standard deviation σ - specimen was most likely collected at the reported location, but there is uncertainty on the scale of σ meters.

Practically, I doubt it will have much of an impact on the results. But either is easy to implement, so we can try both.

Then the estimated temperature can be found by integrating the probability distribution:

$$\langle T \rangle = \int_0^{R_{\max}} T(\vec{r})\rho(r)d^2r, \tag{1}$$

where $R_{\max}$ indicates a relevant area (e.g. $R_{\max} = 2\sigma$).

2 Discrete Case

In reality the temperature measurements are the raster from NOAA. So let's say that instead of knowing the temperature as a continuous function of location, we have measurements on a grid of points $\{\vec{r}_i\}$, where i indexes the measurements, for instance by latitude and longitude. Then the probability distribution function is also a discrete function $p(r_i) \propto \rho(r)$, where the proportionality is determined by the normalization condition that

$$\sum_i p(r_i) = 1. \tag{2}$$

So the estimated temperature is now

$$\langle T \rangle \approx \sum_i p(r_i)T(\vec{r}_i), \qquad i \in (r_i < R_{\max}). \tag{3}$$

3 Implementation

For example, if we choose the uniformly distributed pdf, all locations for which $r < \sigma$ have equal probability. We also need to choose how we decide whether to include pixels, so for now I'll include a pixel if the center is less than σ meters of the origin. For a given σ there are N such pixels, so the pdf is just

$$p(r) = \frac{1}{N}. \tag{4}$$

This reduces to a normal average over all pixels whose centers are within σ meters of the origin.

For the normally distributed pdf

$$p(r) = \frac{1}{N} e^{(-r^2/2\sigma^2)} \tag{5}$$

the normalization N is given by

$$N = \sum_i e^{(-r_i^2/2\sigma^2)}, \qquad i \in (r_i < \sigma). \tag{6}$$

I haven't been careful about defining pdfs here; the definitions imply but don't say the pdf is zero for $r > \sigma$. This should be reflected in the code.

For calculating distance between pixels, note that the distance encompassed by one degree of longitude is actually a function of latitude, while the distance encompassed by one degree of latitude is not. In other words, the circumference of longitude lines gets smaller as you get closer to the poles.