Estimation of Daily Water Temperature using Random Forest

B Steele

September 17, 2023

Checkpoint Issues:

- is this a defensible way to do train/val/test? Should I leave an additional lake out for validation?
- how do I establish a baseline for this? better than yesterday-is-today? (when I don't have a daily timeseries, can I do this?)
-

Scientific motivation and problem statement:

Water temperature is often a reliable indicator of general water quality (cite). Active monitoring of lakes, especially those that are difficult to access by monitoring personnel, is difficult. Additionally, manual monitoring of waterbodies (by physically visiting a site) and sensor networks to monitor water temperature, are costly endeavors (cite).

In this example, I will use Random Forest to estimate water surface temperature for reservoirs with long manual monitoring data from Northern Water. The features that I will be using to estimate surface temperature include summary NLDAS meteorological data (air temperature, precipitation, solar radiation, and wind) as well as static values for each of the reservoirs (elevation, surface area, maximum depth, volume, and shoreline distance).

In addition to the manual sampling record that is maintained by Northern water, I will be leveraging surface temperature estimates from the Landsat constellation, Landsat 4-9. These thermal estimates are well-aligned with the manual monitoring data for the 7 reservoirs. 'Surface temperature' in the manual sampling record for this example is any measured temperature at >= 1m depth. I retain only the top-most value for temperature. Static variables are only available for 6 of 7 reservoirs, so Windy Gap reservoir has been dropped from this analysis.

[[add units to table]]

Table 1: Static variables used in the Random Forest algorithm. Windy Gap Reservoir has incomplete data and has been dropped from this analysis.

feature	elevation	area	shoreline_length	max_depth	volume
Carter Lake	5760	1100	12.0	180	112230
Grand Lake	8370	507	4.5	389	68621
Granby Reservoir	8280	7256	40.0	221	530000
Horsetooth Reservoir	5430	1850	25.0	200	150000
Shadow Mountain Reservoir	8367	1346	8.0	24	16800
Willow Creek Reservoir	5400	303	7.0	124	10500
Windy Gap Reservoir	NA	NA	NA	NA	445

Ideally, implementation of this algorithm will include application to lakes that have only Landsat-derived temperature estimates and that are outside of this dataset. Because I want this algorithm to perform well on new lakes, I want to take steps to make sure that it is not overfit to these specific lakes.

Training/Validation/Testing

It's clear that there are site-level differences in temperature range and general seasonal response (fig?). These differences are likely due to static variables that differentiate these lakes. That said, if I add in site-level information, the algorithm will quickly learn those key attributes and likely overfit to the data, not allowing for generalization beyond these lakes.

Due to this, my test set will be comprised of data from a lake that was never used in the train/validate set (Carter Lake).

For validation, I'll be using timeseries-aware k-fold training and validation sets, using n = 5 splits

Write up contents:

- description of any data pre-processing performed and why you did it
- machine learning setup and reasons for hyperparameter choices when relevant
- results (e.g. testing accuracy)
- a detailed discussion of why you don't think you have overfit

- a detailed discussion of why you think the results are better (or worse if that is the case) than a baseline approach of your choice (e.g. random chance, linear regression, climatology, etc)
- concluding thoughts including any insights gained from your efforts