The Journal of Open Source Software

climpred: Verification of weather and climate forecasts

Riley X. Brady[1] **and Aaron Spring**[2, 3]

1 Department of Atmospheric and Oceanic Sciences and Institute of Arctic and Alpine Research, University of Colorado Boulder, Boulder, Colorado USA **2** Max Planck Institute for Meteorology, Hamburg, Germany **3** International Max Planck Research School on Earth System Modelling, Hamburg, Germany

DOI: 10.21105/joss.02781

Software

- Review
- Repository
- Archive

Editor: David Hagan

Reviewers:

- @neerajdhanraj
- @darothen

Submitted: 16 October 2020
Published: 25 October 2020

Summary

Predicting extreme events and variations in weather and climate provides crucial information for economic, social, and environmental decision-making (Merryfield et al., 2020). However, quantifying prediction skill for multi-dimensional geospatial model output is computationally expensive and a difficult coding challenge. The large datasets (order gigabytes to terabytes) require parallel and out-of-memory computing to be analyzed efficiently. Further, aligning the many forecast initializations with differing observational products is a straight-forward, but exhausting and error-prone exercise for researchers.

To simplify and standardize forecast verification across scales from hourly weather to decadal climate forecasts, we built `climpred`: a community-driven python package for computationally efficient and methodologically consistent verification of ensemble prediction models. The code base is maintained through open-source development. It leverages `xarray` (Hoyer & Hamman, 2017) to anticipate core prediction ensemble dimensions (ensemble `member`, `initialization` date and `lead` time) and `dask` (Dask Development Team, 2016; Rocklin, 2015) to perform out-of-memory and parallelized computations on large datasets.

`climpred` aims to offer a comprehensive set of analysis tools for assessing the quality of dynamical forecasts relative to verification products (*e.g.*, observations, reanalysis products, control simulations). The package includes a suite of deterministic and probabilistic verification metrics that are constantly expanded by the community and are generally organized in our companion package, `xskillscore`.

Statement of Need

While other climate verification packages exist (*e.g.*, `s2dverification` (Manubens et al., 2018) written in R and `MurCSS` (Illing, Kadow, Oliver, & Cubasch, 2014) written with python-based `CDO`-bindings (Schulzweida, 2019)), `climpred` is unique for many reasons. (1) It spans broad temporal scales of prediction, supporting the weather, subseasonal-to-seasonal (S2S), and seasonal-to-decadal (S2D) communities. (2) `climpred` supports `dask` (Dask Development Team, 2016; Rocklin, 2015) and thus works across all computational scales, from personal laptops to supercomputers (HPC). This leads to verification of a global 5° x 5° resolution climate prediction in a few seconds, compared to the 8 minutes required by `MurCSS`. This allows for a truly interactive analysis experience. However, note that benchmarking is inherently biased and `MurCSS` is valuable for their rigorous replication of decadal climate prediction metrics. (3) `climpred` is highly modular and supports the research process from end-to-end, from loading in model output, to interactive pre-processing and analysis, to visualization. (4) `climpred` is part of the wider scientific python community, `pangeo` (Eynard-Bontemps Guillaume, 2019). A wide adoption of `climpred` could standardize prediction model evaluation and make verification reproducible (Irving, 2015). (5) The `climpred` documentation serves as a repository of unified analysis methods through `jupyter` notebook (Kluyver et al., 2016) examples and collects references and literature.

Prediction Simulation Types

Weather and climate modeling institutions typically run so-called "hindcasts," where dynamical models are retrospectively initialized from many past observed climate states (Meehl et al., 2009). Initializations are then slightly perturbed to generate an ensemble of forecasts that diverge solely due to their sensitive dependence on initial conditions (Lorenz, 1963). Hindcasts are evaluated by using some statistical metric to score their performance against historical observations. "Skill" is established by comparing these results to the performance of some "reference" forecast (Jolliffe & Stephenson (2012); *e.g.*, a persistence forecast). The main assumption is that the skill established relative to the past will propagate to forecasts of the future.

A more idealized approach is the so-called "perfect-model" framework, which is ideal for investigating processes leading to potentially exploitable predictability (Bushuk et al., 2018; Griffies & Bryan, 1997; Séférian, Berthet, & Chevallier, 2018; Spring & Ilyina, 2020). Ensemble members are spun off an individual model (by slightly perturbing its state) to predict its own evolution. This avoids initialization shocks (Kröger et al., 2017), since the framework is self-contained. However, it cannot predict the real world. The perfect-model setup rather estimates the theoretical upper limit timescale after which the value of dynamical initialization is lost due to chaos in the Earth system, assuming that the model perfectly replicates the dynamics of the real world. Skill quantification is accomplished by considering one ensemble member as the verification data and the remaining members as the forecasts (Griffies & Bryan, 1997).

Climpred Classes and Object-Oriented Verification

`climpred` supports both prediction system formats, offering `HindcastEnsemble` and `PerfectModelEnsemble` objects. `HindcastEnsemble` is instantiated with an `initialized` hindcast ensemble dataset and requires an `observational` dataset against which to verify. `PerfectModelEnsemble` is instantiated with an `initialized` perfect-model ensemble dataset and also accepts a `control` dataset against which to evaluate forecasts. Both objects can also track an `uninitialized` dataset, which represents a historical simulation that evolves solely due to random internal climate variability or can be used to isolate the influence of external forcing (e.g., Kay et al., 2014).

Assessing skill for `PredictionEnsemble` objects (the parent class to `HindcastEnsemble` and `PerfectModelEnsemble`) is standardized into a one-liner:

```
PredictionEnsemble.verify(
    # Score forecast using the Anomaly Correlation Coefficient.
    metric='acc',
    # Compare the ensemble mean to observations.
    comparison='e2o',
    # Keep the same set of initializations at each lead time.
    alignment='same_inits',
    # Reduce the verification over the initialization dimension.
    dim='init',
    # Score performance of a persistence forecast as well.
```

```
    reference='persistence',
)
```

Each keyword argument allows flexibility from the user's end—one can select from a library of metrics, comparison types, alignment strategies, dimensional reductions, and reference forecasts. The most unique feature to `climpred`, however, is the ability for users to choose the alignment strategy to pair initialization dates with verification dates over numerous lead times. In other words, initialization dates need to be converted to target forecast dates by shifting them using the lead time coordinate. This is tedious, since one must remedy disparities in calendar types between the model and observations and account for the time span of or gaps in observations relative to the time span of the model.

There is seemingly no unified approach to how hindcast initialization dates are aligned with observational dates in the academic literature. The authors of `climpred` thus identified three techniques, which can be selected by the user: (1) Maximize the degrees of freedom by selecting all initialization dates that verify with the available observations at each lead. In turn, initializations and verification dates are not held constant for each lead. (2) Use the identical set of initializations that can verify over the given observational window at all leads. However, the verification dates change at each lead. (3) Use the identical verification window at each lead, while allowing the set of initializations used at each lead to change. These strategies are shown graphically and explained in more detail in the documentation. Note that `climpred` offers extensive analysis functionality in addition to forecast verification, such as spatiotemporal smoothing (Goddard et al., 2013), bias removal (Boer et al., 2016), significance testing (Boer et al., 2016; DelSole & Tippett, 2016; Goddard et al., 2013), and a graphics library.

Use in Academic Literature

`climpred` has been used to drive analysis in three academic papers so far. Brady, Lovenduski, Yeager, Long, & Lindsay (2020) used the `HindcastEnsemble` class to highlight multi-year predictability of ocean acidification in the California Current; Spring & Ilyina (2020) used the `PerfectModelEnsemble` class to highlight predictability horizons in the global carbon cycle; and Krumhardt et al. (2020) used the `HindcastEnsemble` class to illuminate multi-year predictability in marine Net Primary Productivity.

Acknolwedgements

We thank Andrew Huang for early stage refactoring and continued feedback on `climpred`. We also thank Kathy Pegion for pioneering the seasonal, monthly, and subseasonal time resolutions. Thanks in addition to Ray Bell for initiating and maintaining `xskillscore`, which serves to host the majority of metrics used in `climpred`.

References

Boer, G. J., Smith, D. M., Cassou, C., Doblas-Reyes, F., Danabasoglu, G., Kirtman, B., Kushnir, Y., et al. (2016). The Decadal Climate Prediction Project (DCPP) contribution to CMIP6. *Geosci. Model Dev.*, *9*(10), 3751–3777. doi:10/f89qdf

Brady, R. X., Lovenduski, N. S., Yeager, S. G., Long, M. C., & Lindsay, K. (2020). Skillful multiyear predictions of ocean acidification in the California Current System. *Nature Communications*, *11*(1), 1–9. doi:10/ggtpks

Bushuk, M., Msadek, R., Winton, M., Vecchi, G., Yang, X., Rosati, A., & Gudgel, R. (2018). Regional Arctic sealce prediction: Potential versus operational seasonal forecast skill. *Climate Dynamics*. doi:10/gd7hfq

Dask Development Team. (2016). Dask: Library for dynamic task scheduling. Retrieved from https://dask.org

DelSole, T., & Tippett, M. K. (2016). Forecast comparison based on random walks. *Monthly Weather Review*, *144*(2), 615–626. doi:10/f782pf

Eynard-Bontemps Guillaume, H. J., Abernathey Ryan. (2019). The Pangeo big data ecosystem and its use at CNES. Proceedings paper, France, USA, Germany. Retrieved from https://archimer.ifremer.fr/doc/00503/61441/

Goddard, L., Kumar, A., Solomon, A., Smith, D., Boer, G., Gonzalez, P., Kharin, V., et al. (2013). A verification framework for interannual-to-decadal predictions experiments. *Climate Dynamics*, *40*(1-2), 245–272. doi:10/f4jjvf

Griffies, S. M., & Bryan, K. (1997). Predictability of North Atlantic multidecadal climate variability. *science*, *275*(5297), 181–184. doi:10/dp65gs

Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in python. *Journal of Open Research Software*, *5*(1). doi:10/gdqdmw

Illing, S., Kadow, C., Oliver, K., & Cubasch, U. (2014). MurCSS: A tool for standardized evaluation of decadal hindcast systems. *Journal of Open Research Software*, *2*(1), e24. doi:10/gfxr7x

Irving, D. (2015). A Minimum Standard for Publishing Computational Results in the Weather and Climate Sciences. *Bulletin of the American Meteorological Society*, *97*(7), 1149–1158. doi:10/gf4wzh

Jolliffe, I. T., & Stephenson, D. B. (2012). *Forecast verification: A practitioner's guide in atmospheric science*. John Wiley & Sons.

Kay, J. E., Deser, C., Phillips, A., Mai, A., Hannay, C., Strand, G., Arblaster, J. M., et al. (2014). The Community Earth System Model (CESM) large ensemble project: A community resource for studying climate change in the presence of internal climate variability. *Bulletin of the American Meteorological Society*, *96*(8), 1333–1349. doi:10/f7r9st

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., et al. (2016). Jupyter notebooks - a publishing format for reproducible computational workflows. In *Elpub*. doi:10/gf48c9

Kröger, J., Pohlmann, H., Sienz, F., Marotzke, J., Baehr, J., Köhl, A., Modali, K., et al. (2017). Full-field initialized decadal predictions with the MPI earth system model: An initial shock in the North Atlantic. *Climate Dynamics*. doi:10/gdsnf8

Krumhardt, K. M., Lovenduski, N. S., Long, M. C., Luo, J. Y., Lindsay, K., Yeager, S., & Harrison, C. (2020). Potential predictability of net primary production in the ocean. *Global Biogeochemical Cycles*, *34*(6), e2020GB006531. doi:10/gg9ss8

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, *20*(2), 130–141.

Manubens, N., Caron, L.-P., Hunter, A., Bellprat, O., Exarchou, E., Fučkar, N. S., Garcia-Serrano, J., et al. (2018). An R package for climate forecast verification. *Environmental Modelling & Software*, *103*, 29–42. doi:10/gc9wzk

Meehl, G. A., Goddard, L., Murphy, J., Stouffer, R. J., Boer, G., Danabasoglu, G., Dixon, K., et al. (2009). Decadal prediction: Can it be skillful? *Bulletin of the American Meteorological Society*, *90*(10), 1467–1486. doi:10/dpsjbp

Merryfield, W. J., Baehr, J., Batté, L., Becker, E. J., Butler, A. H., Coelho, C. A. S., Danabasoglu, G., et al. (2020). Current and Emerging Developments in Subseasonal to Decadal Prediction. *Bulletin of the American Meteorological Society*, *101*(6), E869–E896. doi:10/ggvcqv

Rocklin, M. (2015). Dask: Parallel Computation with Blocked algorithms and Task Scheduling. In *Python in Science Conference* (pp. 126–132). Austin, Texas. doi:10/gfz6s5

Schulzweida, U. (2019). CDO: Climate Data Operators. Retrieved from https://doi.org/10.5281/zenodo.2558193

Séférian, R., Berthet, S., & Chevallier, M. (2018). Assessing the decadal predictability of land and ocean carbon uptake. *Geophysical Research Letters*. doi:10/gdb424

Spring, A., & Ilyina, T. (2020). Predictability horizons in the global carbon cycle inferred from a perfect-model framework. *Geophysical Research Letters*, *47*(9), e2019GL085311. doi:10/ggtbv2