Bootstrap aggregation and cross-validation methods to reduce overfitting in reservoir policy search

Zachary Paul Brodeur[1], Scott Steinschneider[1], and Jonathan D Herman[2]

[1]Cornell University
[2]University of California, Davis

November 24, 2022

Abstract

Policy search methods provide a heuristic mapping between observations and decisions and have been widely used in reservoir control studies. However, recent studies have observed a tendency for policy search methods to overfit to the hydrologic data used in training, particularly the sequence of flood and drought events. This technical note develops an extension of bootstrap aggregation (bagging) and cross-validation techniques, inspired by the machine learning literature, to improve control policy performance on out-of-sample hydrology. We explore these methods using a case study of Folsom Reservoir, California using control policies structured as binary trees and daily streamflow resampling based on the paleo-inflow record. Results show that calibration-validation strategies for policy selection and certain ensemble aggregation methods can improve out-of-sample tradeoffs between water supply and flood risk objectives over baseline performance given fixed computational costs. These results highlight the potential to improve policy search methodologies by leveraging well-established model training strategies from machine learning.

Bootstrap aggregation and cross-validation methods to reduce overfitting in reservoir policy search

Zachary P. Brodeur[1], Jonathan D. Herman[2], Scott Steinschneider[3]
[1, 3] Department of Biological and Environmental Engineering, Cornell University, Ithaca, NY
[2] Department of Civil and Environmental Engineering, University of California-Davis, CA

1. Graduate Research Assistant, 111 Wing Drive, Riley-Robb Hall, Department of Biological and Environmental Engineering, Cornell University, Ithaca, NY, 14853. Email: zpb4@cornell.edu, Phone: 607-255-2155 (Corresponding Author).

2. Assistant Professor, 3138 Ghausi Hall, Department of Civil and Environmental Engineering, University of California-Davis, Davis, CA, 95616. Email: jdherman@ucdavis.edu, Phone: 530-752-8870.

3. Assistant Professor, 111 Wing Drive, Riley-Robb Hall, Department of Biological and Environmental Engineering, Cornell University, Ithaca, NY, 14853. Email: ss3378@cornell.edu, Phone: 607-255-2155.

Key Points:
1) We apply machine learning techniques of bootstrap aggregation (bagging) and cross-validation to improve policy search
2) Block bootstrapping of available hydrology using paleo inflow data provides an efficient calibration-validation-testing dataset
3) We find greatest improvement in out of sample policy performance by leveraging bootstrap validation to choose policies

Abstract: Policy search methods provide a heuristic mapping between observations and decisions and have been widely used in reservoir control studies. However, recent studies have observed a tendency for policy search methods to overfit to the hydrologic data used in training, particularly the sequence of flood and drought events. This technical note develops an extension of bootstrap aggregation (bagging) and cross-validation techniques, inspired by the machine learning literature, to improve control policy performance on out-of-sample hydrology. We explore these methods using a case study of Folsom Reservoir, California using control policies structured as binary trees and daily streamflow resampling based on the paleo-inflow record. Results show that calibration-validation strategies for policy selection and certain ensemble aggregation methods can improve out-of-sample tradeoffs between water supply and flood risk objectives over baseline performance given fixed computational costs. These results highlight the potential to improve policy search methodologies by leveraging well-established model training strategies from machine learning.

1. Introduction

Efficient policy search methods are becoming increasingly important to identify water system management strategies that provide satisfactory performance across a range of objectives and plausible climate, hydrologic, and regulatory scenarios. Heuristic optimization algorithms have grown in popularity for this purpose (e.g., Nicklow et al., 2010; Reed et al., 2013; Maier et al., 2014), using parameterized functions such as neural networks, binary trees, or radial basis functions to map observed and projected information directly to actions (Raman & Chandramouli, 1996; Koutsoyiannis and Economou, 2003; Giuliani et al., 2014). The parameters and structure of these functions are the decision variables to be optimized, given a training sequence of hydrologic data. A key challenge in this process is the tendency for optimized policies to overfit to the training data, particularly when system performance is driven by infrequent extreme events (e.g., Herman and Giuliani, 2018). In this case, policies may fail to generalize to out-of-sample hydrology even assuming a stationary climate, let alone a nonstationary one.

Several approaches have been explored to address this issue. One common approach is to optimize policies based on several random initializations (seeds) of the heuristic search, and select or combine solutions from those with the highest within-sample performance either over the historical hydrologic record (Salazar et al. 2016; Herman and Giuliani, 2018; Nayak et al., 2018) or a synthetically generated scenario ensemble (Giuliani et al. 2014; Salazar et al. 2017; Giuliani et al. 2018). Other work has re-evaluated policy performance over other synthetic traces not used in training, but sampled from the same uncertainty characterization (Quinn et al., 2017),

or over other traces modified by additional uncertain scenario factors not considered during training (Quinn et al., 2019). These strategies help to reduce policy overfitting to a single trace or synthetic ensemble, though both cases are constrained by the variability observed in the historical record. There remains an opportunity to extend policy search experiments to reduce overfitting by making best use of limited hydrologic data. The machine learning literature offers several promising approaches: namely, bootstrap aggregation (bagging) techniques and calibration-validation-testing frameworks.

Bootstrap aggregation (bagging) is an ensemble method that consists of two primary steps. First, samples are bootstrapped from the training data and used to train an ensemble of models, each fit to a different sample. Second, the model ensemble is applied to out-of-sample data and the classification/prediction of each model is aggregated into a single output (Breiman, 1996a). The bootstrapping scheme is a simple way to approximate independent and identically distributed samples from the underlying population, which increases the diversity of models within the ensemble and significantly reduces classification/prediction variance and overfitting in the final aggregated output. A number of modifications and competitors to this approach have been devised, such as boosting (Freund & Schapire, 1996; Breiman, 1996b); stacking (Wolpert, 1993); and random forests (Breiman, 2001). Each of these leverages ensemble training to improve overall performance, and several aim to achieve both diversity and strength (i.e., low bias) in the ensemble of fitted models.

In calibration-validation-testing frameworks, some portion of the data is used for model fitting, while the remainder is withheld for testing to assess out-of-sample performance. The data that is

retained for model fitting is further divided into training and validation sets. While the training data is used directly to fit the model (i.e., used to calculate objective function values that drive the optimization of parameters), the validation data is used to approximate out-of-sample behavior and guide the training process. For instance, in training Artificial Neural Networks (ANN) it is common to periodically assess network performance on the validation set to initiate a stopping rule to avoid overfitting (Shalev-Shwartz & Ben-David, 2014). Bagging is inherently an efficient validation method, as each model trained on a randomly selected portion of the data can be evaluated against the remainder of the data that it did not see. This allows for a reasonably accurate characterization of out-of-sample performance with no additional cost (Breiman, 1996a) and can be used to weight outputs in the final model aggregation.

In this technical note, we extend bagging techniques and calibration-validation-testing frameworks to reservoir policy design in a case study of the Folsom Reservoir in Northern California. We build upon the previous work of Nayak et al. (2018) and Herman and Giuliani (2018), in which an evolutionary algorithm was used to train binary policy trees that determine reservoir releases balancing water supply and flood control objectives. This work contributes to a developing set of methods to reduce policy overfitting (Giuliani et al. 2014; Salazar et al. 2017; Quinn et al., 2017; Giuliani et al. 2018) by forwarding an experimental design to systematically test how out-of-sample policy performance varies using different combinations of bagging and calibration-validation techniques, which could be applied to the design and testing of any policy structure. These methods are adapted for the highly auto-correlated nature of hydrologic flows using a simple block bootstrapping approach based on a paleo-reconstruction of reservoir inflows, similar to methods in Prairie et al. (2008). We conclude with recommendations of

machine learning methods that hold particular promise for improved reservoir policy design, and possible avenues for future research.

2. Data and Methods

2.1. Case Study: Folsom Reservoir and Policy Trees

These ideas are tested on a case study of Folsom Reservoir, California, following Nayak et al. (2018). We use daily inflow data for the period of 1922 – 2016, split into a training set (1982-2016) and a testing set (1922-1981) based on water years beginning in October. In addition, we use a policy tree formulation of reservoir control rules, which was originally proposed in Herman & Giuliani (2018) (Figure 1). In form, these policy trees are equivalent to Classification and Regression Tree (CART) models (Breiman, 1984), where each node of the tree produces a binary response based on a thresholding feature. In our application, these features are state variables within the reservoir simulation model, i.e., available water at time *t*, equal to the sum of previous storage and current inflow ($S_{t-1} + Q_t$), and day of water year (d_t). The terminal nodes of the tree relate to specific target release actions (u_t), including the release of demand (D_t), varying degrees of water supply hedging, and flood control releases. Target releases are then adjusted for hard constraints (e.g., ramping, maximum channel capacity r_{max} = 130,000 cfs) to produce final releases (r_t).

Indicators	*Description*	*Range (Units)*
	Available Water $S_{t-1} + Q_t$	0-975 (TAF)
	Day of Water Year d_t	0-366 (days)
Actions	*Description*	*Target Release*
	Release Demand	$u_t = D_t$
	Hedge p	$u_t = (p/100)D_t$
	Flood Control	$u_t = Q_t$

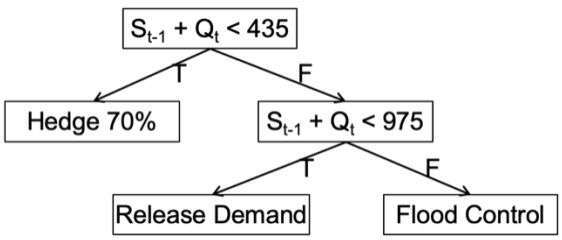

Figure 1: Set of indicator variables and actions for the Folsom Reservoir case study (left); an example policy tree using these components (right).

The policy tree is optimized to minimize the following objective function calculated over the simulation horizon (*N*), which is composed of a squared water supply shortage cost relative to daily demand (first term) and a large penalty that discourages releases that exceed the daily maximum channel capacity r_{max} (second term):

$$J = \frac{1}{N}\sum_{t=0}^{N} \max(D_t - r_t, 0)^2 + \sum_{t=0}^{N} 10^9 \times \max(r_t - r_{max}, 0) \quad \text{(Eq. 1)}$$

A key difference between the policy tree and the original CART model for classification and regression is that the policy tree is evaluated by running a simulation model rather than fitting observed data. It is therefore optimized using random permutations of its structure via an evolutionary algorithm (Herman & Giuliani, 2018), whereas a CART tree is optimized using a recursive partitioning framework. Nonetheless, policy trees have exhibited very similar performance limitations to CART models, including a tendency to overfit to the training data, which has also been observed in other types of policy search methods.

2.2. Experimental Design

The baseline experiment, taken from Nayak et al. (2018), is to train a set of policy trees to a single inflow time-series across a number of random seeds and choose the best policy based on its performance in training. We then devise a set of experiments to test alternate methods of training and selecting policies based on the two components of bagging (bootstrapping and

model aggregation), as well as calibration-validation procedures that are also based on bootstrapped data from the hydrologic record. We employ a $2 \times 2 \times 2$ factorial design, described below and shown in Table 1. Importantly, in all experiments a total of 30 policies are trained, each to a relatively short (35 year) hydrologic trace, which constrains the computational expense to be roughly equivalent in each of the proposed procedures. All experiments were performed on a Dell OptiPlex Desktop with an 8-core, 3.00 GHz i7 processor in a non-parallel configuration, which required approximately 24 hours to create each of the ensembles of 30 policies using 10,000 function evaluations per policy.

Table 1: Summary of Factors and Treatments tested in experimental design.

	Factor 1 – Treatment 1 *(policies fit to original data)*			**Factor 1 – Treatment 2** *(policies fit to bootstrapped data)*	
	Factor 3– Treatment 1 *(aggregate on calibration period statistics)*	**Factor 3 – Treatment 2** *(aggregate on validation period statistics)*		**Factor 3 – Treatment 1** *(aggregate on calibration period statistics)*	**Factor 3 – Treatment 2** *(aggregate on validation period statistics)*
Factor 2 - Treatment 1 *(aggregate via best tree)*	Fit many trees to 1980-2016, pick best tree for 1980-2016 (*best.cal.hist*)	Fit many trees to 1980-2016, pick best tree for resampled hydrology (*best.val.hist*)		Fit one tree to each sequence of resampled hydrology, pick best tree against its own sequence (*best.cal.paleo*)	Fit one tree to each sequence of resampled hydrology, pick best tree across resampled hydrology (*best.val.paleo*)
Factor 2 - Treatment 2 *(aggregate via ensemble mode)*	Fit many trees to 1980-2016, use ensemble mode approach (*ens.mode.hist*)			Fit one tree to each sequence of resampled hydrology, use ensemble mode approach (*ens.mode.paleo*)	

Within Factor 1, we assess the utility of bootstrapping as a method to train multiple policies. Two treatments are considered. Under Treatment 1, bootstrapping is not used when training the policies, and 30 policy trees (i.e., 30 random seeds) are fit to the same historical sequence of 1982-2016 hydrology. Under Treatment 2, 1 policy tree (i.e., 1 seed) is trained to 30 different hydrologic sequences, each of which is developed by block bootstrapping the 1982-2016 data. The block bootstrapping procedure is described in Section 2.3. The primary hypothesis related to Factor 1 is that policies fit to bootstrapped data will exhibit more diversity, providing a wider range of policies that when aggregated will increase system performance.

Regardless of whether policies are fit to the original hydrologic sequence or bootstrapped samples of that sequence, each approach produces 30 policy trees that need to be aggregated to produce a single policy. Factor 2 relates to the aggregation method and considers two different approaches. The first treatment employs a strategy where a single tree is selected from the 30 candidate trees based on a measure of the objective function over a subset of data (Factor 3 relates to the data used for this purpose). Treatment 2 employ a voting-based approach where the decisions of all 30 trees are considered at each time step of the simulation and the decision with the most votes (arithmetic mode) is selected. This aggregation technique is referred to as the ensemble mode approach. If a tie exists in the mode solution, a random selection is made between the two candidate decisions.

For the first of the aggregation strategies (selecting the best tree), each tree needs to be assigned a score that can be used to compare performance across trees. This score is related to, but not equal to, the objective function of that policy evaluated over a subset of data determined by

Factor 3. Under Treatment 1, the data used to optimize the policy tree are also used to calculate the performance score. Under Treatment 2, the score is averaged over the ensemble members of the bootstrapped dataset that were not used to train the policy (i.e., validation traces). The final score used to weight each policy (C_{final}) is calculated by summing scaled (between 0-1) values of water supply cost (first term in Eq. 1, C_{supply}) and total flood volume (second term in Eq. 1, C_{flood}):

$$C_{final} = C_{supply,scaled} + C_{flood,scaled} \quad \text{(Eq. 2)}$$

$$where \quad C_x = \frac{1}{M}\sum_{n=1}^{M} C_{x,n} \quad and \quad C_{x,scaled} = \frac{(C_x - C_{best})}{(C_{worst} - C_{best})}$$

Here, water supply and flood overage costs are first averaged across the M traces being used to develop the weighting score (M equals 1 for Treatment 1; M equals the number of validation traces for Treatment 2) and then scaled by the best (C_{best}) and worst (C_{worst}) costs in the ensemble. We used scaled versions of both water supply and flood overage costs instead of the original objective function so that the weighting of the two costs would be equivalent. In both treatments, the policy with the lowest final score is chosen as the best. No policies evaluated in their respective calibration period (Treatment 1) exhibited flood failures owing to the high penalty imposed by the objective function in training the policy. Therefore, the score used to choose policies under Treatment 1 was based solely on water supply cost.

2.3. Paleo-based Streamflow Bootstrap

This work proposes a simple block bootstrapping approach to support the bagging and calibration-validation scoring procedures discussed above (see Figure 2). In this approach, we

use a 1,113 year (900 – 2012 CE) tree-ring based annual inflow series for the American River upstream of Folsom Lake (Meko & Touchan, 2014). We partition the tree-ring derived series into 35-year periods from 900 – 1915 AD, resulting in 29 annual flow sequences matching the length of our training period (1982-2016). The resampling period ends in 1915 to prevent a policy from being trained on information from the 1922-1981 period, which is being used as an out-of-sample test period.

A K-nearest neighbor (KNN) resampling approach is used to populate each 35-year paleo-period with daily inflow data. For each year in a paleo-period, we first select K=6 years from the training period (1982-2016) that are closest in annual flow to that paleo year. Then, we randomly sample with replacement the monthly flows from those 6 years to reconstruct a daily flow sequence for the paleo-year. The months of October – March are sampled individually, whereas the months of April – September are sampled as one continuous 6-month flow sequence in order to preserve persistence related to snowpack and melt. As an example, if we are seeking to generate a daily flow sequence for the paleo-year 900 CE, we pick the 6 years in the training period closest in annual flow to 900 CE, randomly select one of these 6 years, retrieve the daily October flow sequence from that year, and use those resampled data for October in 900 CE. This process is repeated to fill the remaining months through March, and then again for April-September, but as a single 6-month block.

The proposed block bootstrap preserves realistic sequences of daily inflow but allows for inter-monthly and inter-annual variability not experienced in the training period. Although there are likely some discontinuities between individual months in the cold season, these discontinuities

are less problematic because of the high daily flow variability in this season. Conversely, by maintaining the continuity of daily flows in the April – September dry season, we preserve the persistence of slow hydrologic processes (snowmelt, groundwater discharge) that would generally not change drastically from month to month and may be influenced by inter-monthly factors.

The proposed bootstrapping procedure ultimately produces 29 resampled daily flow sequences, in addition to the original 1982 – 2016 time-series. These sequences form the 30-member ensemble that are the basis for the bagging and validation strategies discussed above. While other approaches could also be used to develop this ensemble (Giuliani et al. 2014; Quinn et al., 2017), the proposed bootstrap procedure requires little effort to develop and can sample from a diverse space of inter-monthly to inter-annual flow sequences.

Finally, a 30-member ensemble is also generated using the same bootstrap procedure described above but using data from the testing period (1922 – 1981) as the basis for resampling. This ensemble is strictly used to assess the performance of different policies developed under the experimental design and is never used in policy training (i.e., all policies are trained on the 1982-2016 data and then tested on the 30-member ensemble based on 1922-1981 data).

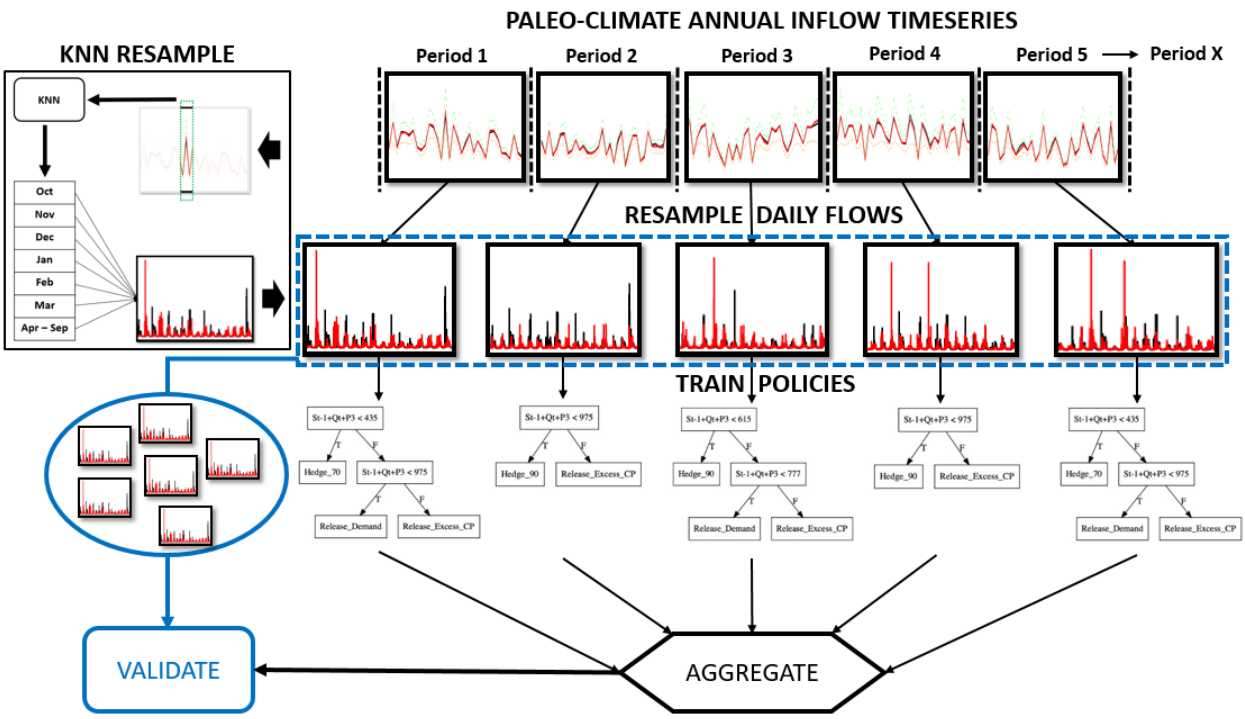

Figure 2: Workflow of hydrological resampling and policy training framework. The top layer shows partitioning of the paleo annual inflow time-series into the desired number and length of periods. The second layer depicts the process to create resampled daily inflows based on the paleo annual inflows with the KNN resampling sub-process depicted in the upper left. The third layer shows the training of policies to each of the resampled daily inflows (Factor 1 – Treatment 2), and the final layer shows the aggregation strategy (Factor 2), which in the case of choosing the best tree (Factor 2 – Treatment 1) may involve comparing policy performance to validation data (blue, Factor 3 – Treatment 2)).

Results

To assess the performance of each factor and treatment, we tested each framework against all resampled inflow sequences from both the training period (1982-2016) and the test period (1922-1981). For each inflow sequence, we simulated the water supply cost and the total flood overage

of each policy. Importantly, the results for the training period are developed in a leave-one-out framework, in which each trace is removed from the 30-member ensemble, and policies from the other 29 traces are aggregated to simulate performance over the left-out trace. In this way, the results shown for the training period reflect some degree of out-of-sample performance, as policies used to simulate reservoir operations over each trace were trained to reordered versions of the 1982-2016 data not experienced in that trace.

Figure 3 shows the distribution of water supply costs and the total flood overage (summed across traces) for each combination of treatments and both the training and testing period. A single value rather than distribution of flood overage values is presented because of the large number of traces with zero flood overages. In each period, we compared the water supply cost distributions for each pair of frameworks using a Tukey multiple pairwise comparison test. In both the upper and lower panels of Figure 3, the 'ens.mode.hist' water supply costs are well above the cost range for other policies (median cost 1982-2016: 1.61, 1922-1981: 3.34). Similarly, the total flood overage for the 'best.cal.paleo' in 1982-2016 (9998 TAF) is much higher than that for other policies. The axis range in Figure 3 is designed to highlight with greater precision the differences between the policy frameworks with more competitive water supply and flood overage costs.

We note that the test period of 1922-1981 is drier than 1982-2016, and so is more prone to higher water supply costs and lower flood overage costs. The histograms in Figure 3 demonstrate this point, showing the observed mean annual flow in the training period near the center of the paleo-resampled distribution, whereas in the test period the observed flow is on the drier end of its

respective paleo-resampled distribution. Despite these differences, the relative water supply costs between formulations are consistent across the training and testing periods, suggesting that these comparative results are robust across a range of climate conditions.

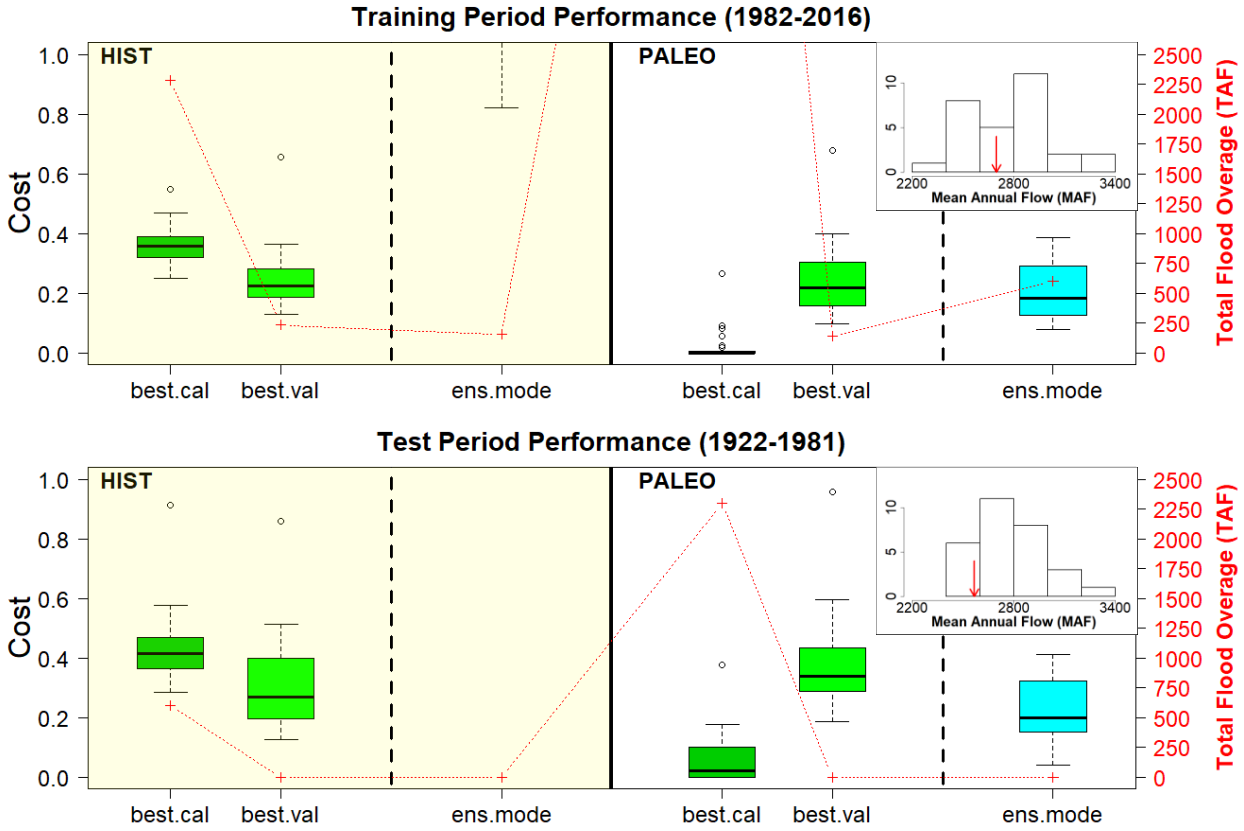

Figure 3: Policy framework performance across factors and treatments from Table 1. Boxplots display the distribution of water supply costs (left y-axis) while red dashed line and '+' symbols show the total flood overage (right y-axis) across all 30 resampled inflows. The upper plot is for the training period of 1982-2016 whereas the lower plot is for the test period of 1922-1981. The pale-yellow background highlights policy frameworks based on the historical sequence of inflow (Factor 1 - Treatment 1) while the white background highlights policy frameworks based on resampled traces based on the paleo-record (Factor 1 – Treatment 2). Histograms in the upper

right corner display the distribution of mean annual flow for 29 resampled 35-year inflow sequences and the red arrow shows mean annual flow from the observed record.

One of the most apparent signals that emerges from Figure 3 is the superiority of using validation data to select a policy (Factor 3). This insight only applies to the aggregation method that chooses the best policy among the ensemble of 30 candidates (best.cal vs. best.val), and holds regardless of the period under consideration (1982-2016 or 1922-1981) or the sequencing of data used to fit the policies (historical sequence or paleo-based bootstrapped sequences). The use of a validation set (Treatment 2) to choose a policy is uniformly better than the use of the calibration set (Treatment 1) with respect to flood overages. When polices are fit to the historic sequence of inflow (best.cal.hist vs. best.val.hist), this performance enhancement is matched by a similar improvement in water supply cost, although the difference in means is not significant. When policies are fit to paleo-based bootstrapped sequences, the water supply cost performance of the best.cal.paleo framework is significantly lower than best.val.paleo ($p < 0.01$ by a Tukey multiple pairwise comparison test), but this comes at a tremendous cost to flood risk (nearly an order of magnitude greater than that of the other policies for the 1982-2016 period). In any conceivable water management scenario, this risk/reward relationship would be unacceptable. Therefore, we focus on the validation-based policies (best.val.hist and best.val.paleo) when comparing policy performance across other factors.

Figure 3 also highlights significant differences in policy performance when policies are fit to the historical sequence versus bootstrapped data (Factor 1), at least for some aggregation techniques. If polices are aggregated by selecting the best policy among a candidate ensemble, then policy

performance does not different significantly for either water supply or flood control costs when using historical or bootstrapped data (best.val.hist vs. best.val.paleo). However, when using the ensemble mode aggregation strategy, there is a large disparity between policy performance (ens.mode.hist vs. ens.mode.paleo). For both the 1982-2016 and 1922-1981 periods, the ens.mode.hist framework (Treatment 1) performs significantly worse ($p << 0.01$) in water supply cost than all the other policies, including the ens.mode.paleo (treatment 2) framework. While ens.mode.hist consistently shows very low flood overage values, many of the other policies can obtain water supply costs that are four or more times lower than the median cost of ens.mode.hist while having similar or only moderately elevated flood risk. This indicates a substantial deficiency in the ens.mode.hist framework. In contrast, the ens.mode.paleo framework exhibits water supply costs that are much more competitive, while maintaining an overall flood risk comparable in magnitude to those of other high-performing policy frameworks.

These results provide evidence that policy diversity can influence ensemble mode performance. Many of the 30 different policies fit to the historical sequence of 1982-2016 data exhibited very similar rule structures and feature thresholds, while the 30 policies each fit to a separate bootstrapped sequence of the 1982-2016 data exhibited a wider range of tree designs. For instance, within the 30 policies fit to historical data, there were three groups (of 5, 4, and 2 policies, respectively) with nearly identical replicate structures. There were no such replicates in the bootstrapped policies. In addition, the distribution of values for the decision threshold on the day of water year feature (*dowy*) was far more variable across policies in the bootstrapped case than the historical case, likely because the timing and magnitude of high flows was more varied in the bootstrapped traces. Given the results in Figure 3, the greater diversity in the policy

ensemble enabled by bootstrapping appears to be necessary to maintain reasonable water supply and flood risk performance when aggregating policies using the ensemble mode approach.

When comparing aggregation methods (Factor 2), the differences between policies are less pronounced. For the purposes of this comparison, we focus on the three most competitive policies, two using the aggregation strategy that selects a single, best policy (best.val.hist and best.val.paleo) and one that uses the ensemble mode (ens.mod.paleo). When using both 1982-2016 and 1922-1981 data, median cost performance for the ensemble mode aggregation is slightly (albeit insignificantly) lower than the policies based on selecting a single best policy. However, flood overages are moderately higher, at least in the 1982-2016 period.

Overall, best.val.hist and best.val.paleo appear to be the policies that provide the best overall performance for both water supply and flood control costs. When using 1982-2016 data, best.val.paleo has near equal water supply costs and slightly better (i.e., lower) flood overages, but when using the 1922-1981 data neither policy has any flood overages and best.val.hist has the lower water supply cost. In addition, in the 1922-1981 test period, the ens.mod.paleo approach does provide non-trivial improvements in water supply costs without any flood increase, but again this only occurs in the drier climate of the 1922-1981 period that is less prone to significant flooding.

Conclusions

Heuristic policy search methods are becoming increasingly popular for designing reservoir control policies, but these methods can suffer from overfitting to training data and reduced policy

performance on out-of-sample streamflow sequences. New frameworks are needed to better exploit the limited hydrologic record to design reservoir policies that can generalize to flow sequences previously unseen in the historical record, which may occur either due to long-term change in the distribution of floods and droughts or limited samples of natural hydrologic variability (Herman et al., 2020). In the latter case, the machine learning literature is rich with novel approaches for this purpose. This technical note draws from that literature and contributes a systematic experimental design to test whether bagging techniques and calibration-validation-testing frameworks can be used to reduce overfitting in reservoir policy design and improve out-of-sample policy performance. Three policy design parameters were considered, including the use of bootstrapped data for policy fitting, aggregation strategies to combine multiple candidate policies into a single decision, and the use of validation data to select policies. These methods were implemented in a case study using policies structured as binary trees, but insights are generalizable to other policy fitting techniques. Importantly, all methods were implemented under a fixed computational cost that was feasible within 24 hours (per method) using relatively standard personal computing resources.

The primary conclusions of this work are as follows:

1. When selecting a single best policy from a candidate set, the use of validation (rather than calibration) performance provides a more robust metric that can help prevent policy performance degradation on out-of-sample hydrology.
2. Validation data based on a simple and fast block bootstrap of the available hydrology can be used for this purpose.

3. When training a policy, the use of bootstrapped data based on a paleo-record did not lead to significant improvements in out-of-sample policy performance (particularly when selecting a single best policy from a candidate set). However, the use of the bootstrapped data for validation and policy selection did lead to significant improvements (see #1 and #2 above).
4. An ensemble mode approach to policy aggregation requires diversity among the candidate policies to be competitive with other approaches. However, even with a diverse set of policies, the ensemble mode approach did not provide significant benefits over the simpler approach of selecting a single, best policy based on validation performance.

Overall, the results of this work suggest that there is potential to improve policy optimization and selection processes by incorporating some methods used in classic machine learning constructs. As stated above, the calibration-validation-testing framework showed the most potential in this study. However, there are many other machine learning methods to control overfitting that could be considered in future experiments. For instance, the ensemble mode approach used in this study (ens.mode.paleo) produced results on par with the other high-performing techniques (and somewhat better for the dry climate of 1922-1981), and is a relatively simple bagging formulation analogous to the early scheme presented in Breiman (1996a). Modifications to this technique to include a weighted voting scheme could prove effective in improving performance of an ensemble based solely on historical inflow data. In addition, alternative choices could be explored for developing the trees included in the ensemble mode, such as a Random Forest formulation in which the underlying policy tree algorithm would include random feature selection. Finally, a common critique of ensemble methods is their lack of policy interpretability.

Future work could explore approaches to address this issue by deriving a simple policy structure from ensemble mode-based decisions. For instance, in the case of policy trees (the method used in this work), a single policy tree could be derived from the input/output sequence of the ensemble mode via another CART regression.

Our experiments were conducted on a relatively simple case study with two objectives (water supply cost and flood overage) that were combined into a single objective function using a weighting approach. It is possible that the benefits of some of the approaches considered in this work could become more or less apparent in a higher-dimensional multi-objective formulation, or a more complex case study of a multi-reservoir system. In addition, we leveraged an existing paleo-based inflow reconstruction in our bootstrapping approach. In instances where such reconstructions are unavailable, a time series model of the annual inflow record (possibly designed to capture low-frequency variability) could be used as an alternative source of data on which to base the block bootstrap (see Steinschneider and Brown, 2013). Both of these issues are left as potential avenues of future work.

Acknowledgements:

This work was supported by the U.S. National Science Foundation grant EnvS-1803563. The data used in this manuscript are in this repository and cited in the references: HydroShare, http://www.hydroshare.org/resource/b8f87a7b680d44cebfb4b3f4f4a6a447.

Reference:

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees.* Belmont, CA: Wadsworth.

Breiman, L. (1996a). Bagging predictors. *Machine Learning*, *24*(2), 123–140. https://doi.org/10.1007/bf00058655

Breiman, L. (1996b). Bias, variance, and arcing classifiers, *Technical Report 460*, Statistics Department, University of California-Berkeley, CA.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1007/9781441993267_5

Brodeur, Z. P., S. S. Steinschneider, J. D. Herman (2020). Data repository for 'Bootstrap aggregation and cross-validation methods to reduce overfitting in reservoir policy search', HydroShare, http://www.hydroshare.org/resource/b8f87a7b680d44cebfb4b3f4f4a6a447

Freund, Y., & Schapire, R. R. E. (1996). Experiments with a New Boosting Algorithm. *International Conference on Machine Learning*, 148–156. https://doi.org/10.1.1.133.1040

Giuliani, M., Herman, J. D., Castelletti, A., & Reed, P. (2014). Many-objective reservoir policy identification and refinement to reduce policy inertia and myopia in water management. *Water Resources Research*, *50*(4), 3355–3377. https://doi.org/10.1002/2013WR014700

Giuliani, M., Quinn, J. D., Herman, J. D., Castelletti, A., & Reed, P. M. (2018). Scalable Multiobjective Control for Large-Scale Water Resources Systems under Uncertainty. *IEEE Transactions on Control Systems Technology*, *26*(4), 1492–1499. https://doi.org/10.1109/TCST.2017.2705162

Herman, J. D., & Giuliani, M. (2018). Policy tree optimization for threshold-based water resources management over multiple timescales. *Environmental Modelling and Software*, *99*, 39–51. https://doi.org/10.1016/j.envsoft.2017.09.016

Herman, J. D., Quinn, J. D., Steinschneider, S., Giuliani, M., & Fletcher, S. (2020). Climate adaptation as a control problem: Review and perspectives on dynamic water resources planning under uncertainty. Water Resources Research, 56, e24389. https://doi.org/10.1029/2019WR025502

Kirsch, B. R., Characklis, G. W., & Zeff, H. B. (2013). Evaluating the impact of alternative hydro-climate scenarios on transfer agreements: Practical improvement for generating synthetic streamflows. *Journal of Water Resources Planning and Management*, *139*(4), 396–406. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000287

Nowak, K., Prairie, J., Rajagopalan, B., & Lall, U. (2010). A nonparametric stochastic approach for multisite disaggregation of annual to daily streamflow. *Water Resources Research*, *46*(8). https://doi.org/10.1029/2009WR008530

Prairie, J., Nowak, K., Rajagopalan, B., Lall, U., & Fulp, T. (2008). A stochastic nonparametric approach for streamflow generation combining observational and paleoreconstructed data. *Water Resources Research*, *44*(6), 1–11. https://doi.org/10.1029/2007WR006684

Knutti, R. (2010). The end of model democracy? *Climatic Change*, *102*(3), 395–404. https://doi.org/10.1007/s10584-010-9800-2

Koutsoyiannis, D., & Economou, A. (2003). Evaluation of the parameterization-simulation-optimization approach for the control of reservoir systems. *Water Resources Research*, *39*(6). https://doi.org/10.1029/2003WR002148

Maier, H. R., Kapelan, Z., Kasprzyk, J., Kollat, J., Matott, L. S., Cunha, M. C., et al. (2014). Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. *Environmental Modelling and Software*, *62*, 271–299. https://doi.org/10.1016/j.envsoft.2014.09.013

Markus, M., Angel, J., Byard, G., McConkey, S., Zhang, C., Cai, X, et al. (2018). Communicating the impacts of projected climate change on heavy rainfall using a weighted ensemble approach. *Journal of Hydrologic Engineering*, *23*(4). https://doi.org/10.1061/(ASCE)HE.1943-5584.0001614

Meko, D. and Touchan, R. (2014). American River Inflow to Folsom Lake, CA (900-2012 AD), TreeFlow Streamflow Reconstructions from Tree Rings, California Department of Water Resources. https://www.treeflow.info/content/american-river-inflow-folsom-lake-ca. Accessed 2019-10-01.

Nicklow, J., Reed, P., Savic, D., Dessalegne, T., Harrell, L., Chan-Hilton, et al. (2009). State of the Art for Genetic Algorithms and Beyond in Water Resources Planning and Management. *Journal of Water Resources Planning and Management*, *136*(4), 412–432. https://doi.org/10.1061/(asce)wr.1943-5452.0000053

Nayak, M. A., Herman, J. D., & Steinschneider, S. (2018). Balancing flood risk and water supply in California: Policy search integrating short-term forecast ensembles with conjunctive use. *Water Resources Research*, 1–20. https://doi.org/10.1029/2018WR023177

Quinn, J. D., Reed, P. M., Giuliani, M., & Castelletti, A. (2017). Rival framings: A framework for discovering how problem formulation uncertainties shape risk management trade-offs in water resources systems. *Water Resources Research*, *53*(8), 7208–7233. https://doi.org/10.1002/2017WR020524

Quinn, J. D., Reed, P. M., Giuliani, M., & Castelletti, A. (2019). What Is Controlling Our Control Rules? Opening the Black Box of Multireservoir Operating Policies Using Time-Varying Sensitivity Analysis. *Water Resources Research*, 5962–5984. https://doi.org/10.1029/2018wr024177

Raman, H., & Chandramouli, V. (1996). Deriving a general operating policy for reservoirs using neural network. *Journal of Water Resources Planning and Management*, *122*(5), 342–347.

Reed, P. M., Hadka, D., Herman, J. D., Kasprzyk, J. R., & Kollat, J. B. (2013). Evolutionary multiobjective optimization in water resources: The past, present, and future. *Advances in Water Resources*, *51*, 438–456. https://doi.org/10.1016/j.advwatres.2012.01.005

Zatarain Salazar, J., Reed, P. M., Herman, J. D., Giuliani, M., & Castelletti, A. (2016). A diagnostic assessment of evolutionary algorithms for multi-objective surface water reservoir control. *Advances in Water Resources*, *92*, 172–185. https://doi.org/10.1016/j.advwatres.2016.04.006

Salazar, J. Z., Reed, P. M., Quinn, J. D., Giuliani, M., & Castelletti, A. (2017). Balancing exploration, uncertainty and computational demands in many objective reservoir optimization. *Advances in Water Resources*, *109*(September), 196–210. https://doi.org/10.1016/j.advwatres.2017.09.014

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge, UK: Cambridge University Press. Retrieved from https://doi.org/10.1017/CBO9781107298019

Steinschneider, S., and C. Brown (2013), A semiparametric multivariate, multi-site weather generator with low-frequency variability for use in climate risk assessments, Water Resour. Res., 49, 7205-7220, doi: 10.1002/wrcr.20528.

Strobl, C., Malley, J., & Gerhard, T. (2009). An Introduction to Recursive Partitioning: Rationale, Application Psychol Methods. *Psychological Methods*, *14*(4), 323–348. https://doi.org/10.1037/a0016973

Wolpert, D. (1992). Stacked Generalization (Stacking). *Neural Networks*, *5*, 241–259.