Machine Learning in Predicting Ungauged Basin Flows

By

Elaheh White
B.S. University of Kentucky 2015
M.S. University of California, Davis 2017

Dissertation

Submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

in

Civil and Environmental Engineering

in the

Office of Graduate Studies

of the

University of California

Davis

Approved:

Jay R. Lund, Chair

Robert J. Hijmans

Jonathan D. Herman

Committee in Charge

2020

To Jim White,...

I hope you were right, and that we meet again.

Contents

List of Figures

List of Tables

Abstract

Machine Learning in Predicting Ungauged Basin Flows

All science is the search for unity in hidden likeness (Bronowski, 1988). There are two reasons to approximate the processes that produces such hidden likeness: (1) *prediction* for interpolation or extrapolation to unknown (often future) situations; and (2) *inference* to understand how variables are connected or how change in one affects others. Statistical learning tools aid both prediction and inference. In recent years, rapidly growing computing power, the advent of machine learning algorithms, and user-friendly programming languages (e.g., R and Python) allow for the application of statistical learning methods to broader societal problems.

This dissertation developed statistical learning models, generally simpler than mechanistic models, to predict the unimpaired flows of California basins from available data. Unimpaired flow is the flow produced by the basin in its current state, but, without dams and diversions (California Department of Water Resources, Bay-Delta Office, 2016). The models predict these type of flows for ungauged basins, an International Association of Hydrological Sciences "grand challenge" in hydrology. In Predicting Ungauged Basin (PUB) flows, the models learn from the information available at gauged points on a river and extrapolate to points that are ungauged.

Multiple issues arise when solving this prediction problem: (1) the way we view hydrology, and what we define as a observational unit, determines how data gets fed into statistical learning methods. Therefore, one issue is in deciding the organizational form of the data (i.e., aggregate vs. incremental basins). This concept of data transformation or pre-processing strategies are explored in chapter two; (2) often, in water resources, we are not concerned with accurately predicting the expectation (or the mean) of a distribution; instead, we care to accurately predict the extreme values of the distribution (i.e., floods and droughts). This problem can be tackled with defining asymmetric loss functions presented in chapter three; (3) hydrologic data is structured, meaning there are dependencies and correlation structures inherent in the data (temporal and spatial autocorrelation exist) and rivers forming a network that flow into one another (hierarchical autocorrelation exists). These characteristics

require a careful construction of resampling techniques for model error estimation, which are discussed in chapter four; (4) non-stationarity due to climate change requires adjustments to statistical models, especially, if they are meant to aid long-term decision-making. Chapter five develops models from datasets representing future hydrology.

These issues make PUB a non-trivial problem for statistical learning methods that operate with no a priori knowledge of the system. In juxtaposition with physical or semi-physical models, statistical learning models learn from the data itself, with no assumptions on the underlying process. Their advantages lie in their fast and easy development, simplicity of use, lesser data requirements, good performance, and flexibility in model structure and parameter specifications. In the past two decades, more sophisticated statistical learning models have been applied to rainfall-runoff modeling.

This dissertation starts with an introduction to the PUB problem, along with a literature review, and a heuristic guide for model selection (chapter one), and ends with a sociological examination of the implications of big data, statistical learning, and artificial intelligence in the larger context of our society and political economy (chapter six). It is intended to provide insights in using statistical learning techniques to civil engineers. Since, with these techniques, we now have the potential to get to wrong answers faster, we should take care in each step of model selection, development, and prediction error estimation.

Keywords: predicting ungauged basins (PUB); rainfall-runoff modeling; asymmetric loss functions; structured data; blocking resampling methods; climate change; McDonaldization; water resources; hydrology; statistical learning.

Acknowledgments

UPDATE THIS WHEN STUDY IS DONE!!!!

All data processing and model development for this study was done in `R version 3.5.2 (2018-12-20)`, a statistical programming language, (R Core Team, 2018) on platform: x86_64-w64-mingw32/x64 (64-bit), running under: Windows >= 8 x64 (build 9200). Programs were written in `RStudio`, an integrated development environment, (RStudio Team, 2016), and used the following packages: `statmod` (Giner & Smyth, 2016) `sp` (Pebesma & Bivand, 2005; R. S. Bivand, Pebesma, & Gomez-Rubio, 2013), `raster` (Hijmans, 2019), `rgeos` (R. Bivand & Rundel, 2018), `rgdal` (R. Bivand, Keitt, & Rowlingson, 2018), `dismo` (Hijmans, Phillips, Leathwick, & Elith, 2017), `geojsonio` (Chamberlain & Teucher, 2018), `hydroGOF` (Mauricio Zambrano-Bigiarini, 2017), `RColorBrewer` (Neuwirth, 2014), `fBasics` (Wuertz, Setz, & Chalabi, 2017), `lmtest` (Zeileis & Hothorn, 2002), and `reshape2` (Wickham, 2007).

The code used for analysis is provided in a repository at:
https://github.com/whiteellie/predicting-ungauged-basins.

This material is based upon work partly supported by the NSF GRFP under Grant No. 1650042 and the Climate Change, Water, and Society NSF IGERT, to UC Davis DGE No. 1069333. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Acronyms & Abbreviations

TO BE UPDATED FOR FINAL DRAFT!!!!!

AF: Acre-feet

AF/m: Acre-feet/Month

CDWR: California Department of Water Resources

CDEC: California Data Exchange Center

GLM: Generalized Linear Multivariate Regression

LM: Linear Multivariate Regression

UF: Unimpaired Flow

ME: Mean Error

MAE: Mean Absolute Error

MSE: Mean Squared Error

RMSE: Root Mean Squared Error

NRMSE: Normalized Root Mean Squared Error

PBIAS: Percent Bias

RSR: RMSE to Standard Deviation of Observations Ratio

NSE: Nash-Sutcliffe Efficiency

rSD: Ratio of Standard Deviations

mNSE: Modified Nash-Sutcliffe Efficiency

rNSE: Relative Nash-Sutcliffe Efficiency

d: Index of Agreement

md: Modified Index of Agreement

rd: Relative Index of Agreement

cp: Persistence Index

r: Pearson Correlation coefficient

R2: Coefficient of Determination

bR2: Bias-Corrected Coefficient of Determination

KGE: Kling-Gupta Efficiency

VE: Volumetric Efficiency

Chapter 1

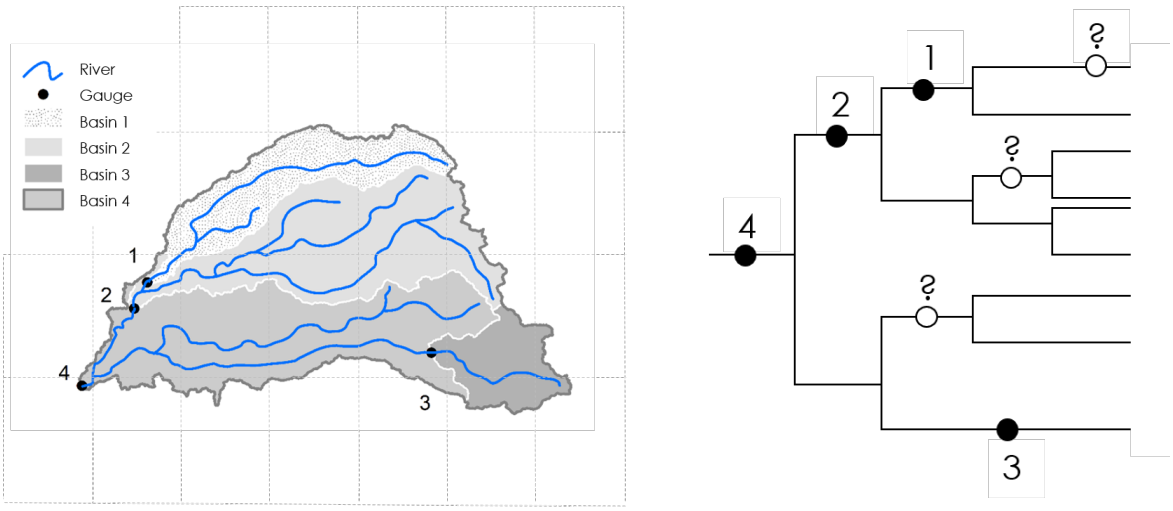
Introduction & Literature Review

> Life must be lived forwards, but it can only be understood backwards.
>
> Sören Kierkegaard, *"The Journals of Sören Kierkegaard"*, 1844

1.1 Introduction

Our ability to extract insights from large diverse data sets has rapidly improved with growing computing power and sophisticated algorithms. The field of *statistical learning* has emerged as a framework that ranges from simple linear regression and complex algorithmic methods (James, Witten, Hastie, & Tibshirani, 2013). A main contribution of this field is the development of modeling techniques that allow for the semi-automatic creation of complex models, with many interacting predictor variables, which are not overfit, and predict well. These developments allow for more accurate and flexible empirical models to manage complex systems. For example, in hydrology, runoff formation processes are highly variable, non-linear, and spatially heterogeneous, which creates a challenge for predicting processes such as streamflow (Dooge, 1986). The International Association of Hydrological Sciences (IAHS) dubbed the 2003-2012 years the decade on Predictions in Ungauged Basins (PUB) (Sivapalan et al., 2003). The PUB initiative has aimed the scientific community, in a coordinated manner, towards achieving major advances in the capacity to make predictions in ungauged basins (Figure 1.1).

Predicting and forecasting hydrology at ungauged sites promotes better management of water and the environment (Sivapalan et al., 2003). It is important for the sustainable



(a) PUB Problem

(b) PUB Schematic

Figure 1.1: The predicting ungauged basins (PUB) problem. This dissertation focuses on predicting unimpaired flows at ungauged locations from other gauges on the network. Predictor variables include climate and basin characteristics.

management of river basins, integrating economic, social and environmental perspectives (Sivapalan, 2003), flood protection, water supply and drought management, solving water quality issues (Hrachowitz et al., 2013), and to serve as inputs for other models.

1.2 Terms & Definitions

This dissertation investigates the relationships between the response variable: "unimpaired flow" and the predictor variables: climate and basin characteristics. Before continuing, it is useful to know how unimpaired flow is defined. Unimpaired flow is the flow that is produced by the basin in its current state, but, without dams and diversions (California Department of Water Resources, Bay-Delta Office, 2016). Unimpaired flow calculations are used mostly in places where dams have created major changes to the natural flow regime. It is often calculated by a simple accounting of water in the system (Figure 1.2 and Equation 1.1).

$$q_{uf} = q_{out} - q_{imp} + q_{div} + \Delta S + q_{evap} \tag{1.1}$$

Where q_{uf} is unimpaired flow, q_{out} is observed gauge data, q_{imp} is imported flows, q_{div} is diverted flows, ΔS is the change in storage, and q_{evap} is the evaporation out of the system.

In contrast, "natural flow" is the runoff produced by a basin in its pre-development state

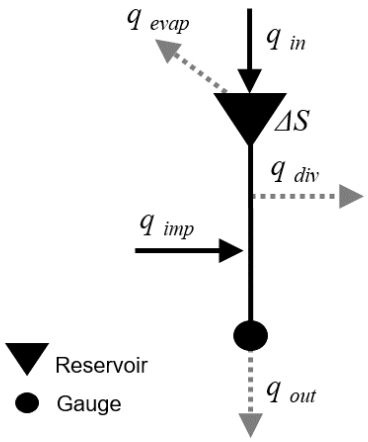


Figure 1.2: Calculating unimpaired flow. Unimpaired flow is calculated by adding back in diversions, subtracting imports, accounting for change in storage and evaporation caused by the reservoir.

prior to any human alterations (Poff et al., 1997). The differences between unimpaired flow and natural flows are usually driven by effects of levees, upland land use, wetlands, and groundwater. This study, however, was only concerned with unimpaired flow; the models were built with unimpaired flow data from the California Data Exchange Center (CDEC), and the predictor variables are taken from various sources discussed in appendix A. See appendix B for terms and concepts used in statistical learning.

1.3 Literature Review

1.3.1 Hydrologic Modeling

Hydrologic models in PUB can be classified as *mechanistic* (physical process-based, causal) or *empirical* (statistical, purely stochastic) (Guisan & Zimmermann, 2000) (Figure 1.3). All modeling techniques assume that the past is a reasonable guide to the future, and that data from one basin is a useful guide to understanding hydrological responses at another basin (Sivapalan, 2003). But, each approach sacrifices some generality, realism, cost, and precision for better understanding, predicting, and managing natural resources (Levins, 1966; Klemes, 1982).

Hydrologists have used both mechanistic and empirical models to capture complex runoff processes; since the mid-19$^{\text{th}}$ century, with the employment of the *rational method*, empirical relationships have been used in rainfall-runoff modeling (Beven, 2011). Engineers developed

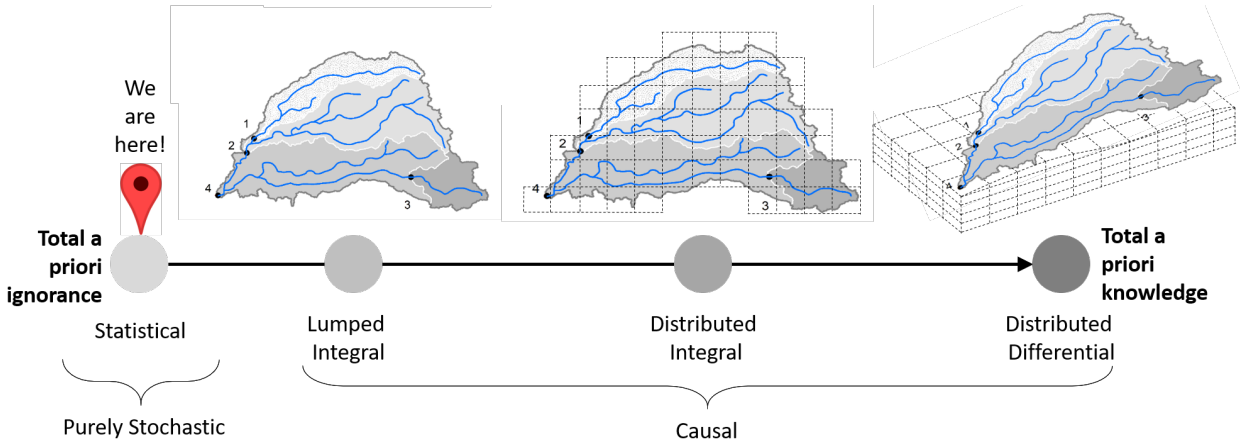


Figure 1.3: The different classes of hydrologic models. The hydrologic modeling field has been moving from total a priori ignorance to total a priori knowledge of the system. With the increase in computing power and the development of statistical learning methods, hydrologist can now re-visit predicting hydrologic conditions with purely stochastic methods.

the rational method in response to problems in which the design discharge was of major concern (i.e., urban sewer, land reclamation drainage systems, and reservoir spillway design) (Todini, 1988). This method, based on the concept of concentration time, calculates runoff by simply multiplying a runoff coefficient by rainfall intensity and the basin's drainage area. It is applicable only to small or mountainous catchments where the rainfall duration normally exceeds the basin's concentration time, the time it takes for the entire basin area precipitation to reach the basin's outlet as discharge.

To address more complexities in rainfall duration, basin size, and non-uniform characteristics, other methods emerged. In the 1930s, the *unit hydrograph* method was developed (Sherman, 1932). In the 1950s, mathematical techniques such as Z, Laplace or Fourier transforms led to the derivation of response functions from the analysis of input and output data (Dooge, 1973). In the 1960s, grander approaches emerged to model the physical processes of the hydrologic cycle. Models increased in complexity over time and often lacked realistic parameter estimates, leading researchers to other ambitious mechanistic modeling efforts (Todini, 1988). These models require considerable field input data collection and model calibration to obtain basin-specific parameters (Singh & Frevert, 2005). Unfortunately, as mechanistic models increase in complexity, it is unclear if hydrologic predictions improve commensurately (Beven, 2011).

Our incomplete understanding of the process (Hrachowitz et al., 2013), poor understanding of where water goes when it rains, what flow path it takes to the stream, and the age of the water that emerges in the channel (Sivapalan, 2003) make PUB a difficult problem to model. Spatio-temporal heterogeneity of climate and basin characteristics create a *uniqueness-of-place* issue, and there is a lack of agreement on what is a suitable regionalization technique for this problem (Hrachowitz et al., 2013).

Without a unifying approach, and considering the increasing availability of environmental data, in the past two decades, more sophisticated statistical learning models have been applied to rainfall-runoff modeling. In juxtaposition with physical or semi-physical models, machine learning models learn from the data itself, with no assumptions as to the underlying process.

1.3.2 Statistical Learning

Artificial intelligence has gone through the ages of speculation (1940s), dawn, business, and bulldozer age (Winston, 2010). In the bulldozer age, with seemingly unlimited computing capacity, machines process more abundant data much like a bulldozer processes soil. Recent advances in reinforcement learning, one-hot learning (where machines learn from the first example), learning in sparse spaces, and the integration of thinking, perception, and action (rather than viewing them separately) are moving us away from the bulldozer era (Winston, 2010). See appendix C for a brief history of statistical learning. However, the application of these newer techniques to water resources problems is slow.

We can group the statistical learning methods developed in the bulldozer age into seven main categories: *supervised machine learning*, *regression family*, *time series analysis*, *geostatistics*, *multi-variate analysis*, *unsupervised machine learning*, and other methods. Supervised machine learning methods are more generally used for predicting a variable in the past where no equation is needed to represent the model. In contrast, the regression family of methods are used when the purpose is more *inference* than *prediction*, and equations, or more specifically the coefficients of the variables in the equations, are of interest. Time series analysis is most suited to prediction problems where the time component is of interest (e.g., problem of extrapolating to the future), as opposed to geostatistics, which is mainly concerned with the spatial component of the data. Under pattern recognition problems sets,

multi-variate analysis and unsupervised machine learning methods find natural groupings in the data. Other methods handle networks, text, patterns caused by latent factors, and relationships between variables, which generally don't apply to problems in water resources. Lastly, in descriptive methods, we use measures of center (e.g., mean and median), measures of spread (e.g., range, standard deviation, and quantiles), and the distribution of a variable as a way to describe the data.

Taxonomies, like the one shown in Figure 1.4, help guide the user through a discovery process. Its goal is for the user to be able to identify an object of interest without prior knowledge of its existence. In statistical learning, method or model selection is iterative and should follow the *generate-and-test* approach. So, any guide to model selection should be considered only a heuristic, meaning as a general rule it will recommend appropriate methods, but may fail sometimes.

The hydro-informatics literature shows that the techniques presented in the heuristic guide are aiding civil engineers in the fields of: (1) hydrology: e.g., rainfall-runoff modeling and model calibration; (2) hydraulics: e.g., water levels in channels and reservoirs; (3) environmental water quality: e.g., temperature and groundwater heads; (4) urban water supply: e.g., water demand and water distribution networks; and (5) general data cleaning and anomaly detection. In the following sections, we will discuss the models suitable to the PUB problem.

1.3.3 Suitable Statistical Modeling for Hydrologic Data

Precipitation feeding into a stream needs to satisfy soil moisture deficits along its flow path before it produces runoff. In other words, the soil needs to "fill" up to a certain threshold before it can "spill" (Spence & Woo, 2006). Therefore, a *threshold behavior* is frequently discussed as influencing local, hillslope and catchment scale runoff generation processes (Zehe & Sivapalan, 2008). This physical phenomenon may be why most successful machine learning techniques currently applied to the rainfall-runoff problem use artificial neural networks (e.g., Minns & Hall, 1996; Dawson & Wilby, 1998; Tokar & Johnson, 1999; Hsu, Gupta, Gao, Sorooshian, & Imam, 2002; Hu, Wu, & Zhang, 2007; Abrahart, Heppenstall, & See, 2007; Govindaraju & Rao, 2013). In artificial neural networks, at each node the weighted sum of all inputs are passed through a non-linear activation function. Much like the neurons in our

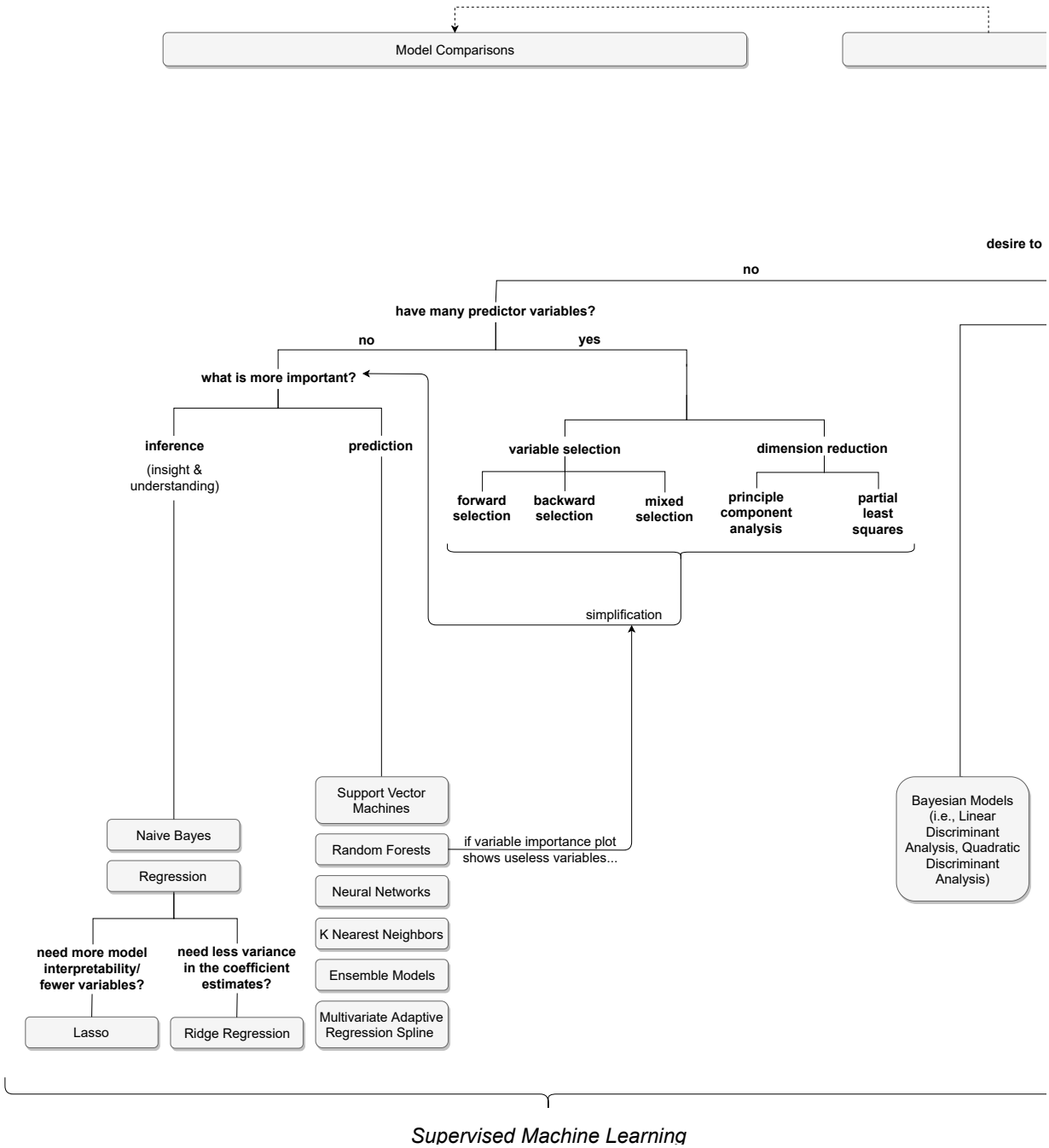


Figure 1.4: What are you trying to do? Heuristic guide for model selection.

brains, there is a threshold that determines if the neuron will "fire".

The same effect can be replicated with tree based algorithms where models are built with a series of binary splits on the predictor variables (e.g., Iorgulescu & Beven, 2004; Galelli & Castelletti, 2013; Magnuson-Skeels, 2016; Worland, Farmer, & Kiang, 2018). Papers in which the number of basins in the study are fairly small suffer when forming the test/train or calibration/validation split. Usually, in these studies the data for one whole basin is not held out when training; in other words, the models are able to learn from a partial record from the basin of interest. Although this approach seems to be valid for rainfall-runoff modeling in the current literature, it does not comply by the test set requirements in the PUB problem where no data from the basin in the test set is available to the model. In contrast, when the datasets are large (e.g., studies done on the the GAGESII dataset, a massive USGS hydrologic data set), this problem is less pronounced. Some studies employ a random test/train split which is not appropriate when the dataset is correlated. We will discuss this concept further in chapter 4. These studies also employ a pre-modeling split on the dataset when classifying basins as "impaired" vs. "reference" basin. This imposes a subjective top split in the data and homogenizes the basins in the study; the reference basins are usually smaller headwater basins with low flows. As such, and rightly so, these models fail to make accurate predictions when extrapolating to basins lower in the network, with higher flows, since the model was denied information such information.

More recently, studies have turned to support vector machines (SVM) (Asefa, Kemblowski, McKee, & Khalil, 2006; Lin, Cheng, & Chau, 2006), which initially were only applied to classification problems and have now been modified to accommodate regression problems (e.g., Han, Chan, & Zhu, 2007; Yu, Liong, & Babovic, 2004; Bray & Han, 2004 applied to flood forecasting). Such studies show that advances are putting SVMs generally on par with artificial neural networks in terms of model performance. However, the applications of SVM in time-series regression is still in its infancy; one study showed a peculiar behavior of the SVM where lighter rainfall would generate unrealistic hydrographs that would increase to an equilibrium point rather than having the characteristic skewed bell shape (Han et al., 2007). Of course, this contradicts the physical principle that limited rainfall cannot generate an unlimited flow.

The difficulty in modeling lower flows is not unique to SVMs. Other modeling techniques (e.g., LM, GLMs, ...) suffer from the same problem given that the response, unimpaired flows, is a *semi-continuous* variable. Semi-continuous data take non-negative values but have a substantial proportion of values at zero. The modeling of such "clumped-at-zero" or "zero-inflated" data is challenging (Min & Agresti, 2002). The following methods have been developed to solve this issue:

- Censored regression model: A censored regression, or Tobit, model assumes that the data comes from a single underlying Normal distribution, but that negative values are censored and stacked on zero (Tobin, 1958).

- Two-part models: As opposed to the Tobit model that allows the same underlying stochastic process to determine whether the response is zero or positive as well as the value of a positive response, two-part models allow the two components to have different parameters. Without assuming an underlying distribution, Duan, Manning, Morris, and Newhouse (1983) proposed a two-part model that uses two equations to separate the modeling into two stages. The first stage refers to whether the response outcome is positive (e.g., a binomial model). Conditional on its being positive, the second stage refers to its level (e.g., linear model).

- Compound Poisson exponential dispersion models: A model that uses a single distribution from the exponential dispersion family (i.e., Tweedie distribution) to analyze semi-continuous data. The distributions in this family have a given range of shape parameters ($1 < \alpha < 2$) which define a point mass at zero and a skewed positive distribution for positive values.

As Min and Agresti (2002) explain, other modeling methods exist to solve the problem of inflated zeros or other inflated boundaries (e.g., ordinal threshold, finite mixture, Neyman type A models). Unfortunately, these methods may require groupings that necessitate information loss, may overestimate the number of components when there is a lack of model fit, or employ methods where the mathematical and inferential advantages associated with the family of distributions are not available and are simply difficult to fit. As such, we will not discuss them here.

In this thesis, we will be developing Linear Multivariate Regression (LM) as a first pass model. We will then develop Generalized Linear Regression (GLM), Tobit Regression (TR), Random Forest (RF), and Neural Network (NN) models.

1.4 Limitations & Assumptions of Statistical Modeling

Many hydrologists are skeptical of statistical modeling. Klemes (1982) warns modelers of the general limitations of empirical modeling, some of which are discussed here.

In search of "better calculus", the modeler may be in danger of **overfitting** (i.e., regarding noise in the data as information) (Klemes, 1982). However, resampling methods, when correctly applied, can illuminate differences between training and testing set performances (J. Friedman, Hastie, & Tibshirani, 2001).

Furthermore, empirical models must be regarded as **interpolation formulas**, and so, lack justification outside the range of underlying data sets (Klemes, 1982). The models in this study were fitted with data on the California Sierra Nevada mountainous basins, and some coastal, and southern California basins (Figure A.1). These training data sets mostly span the same hydrologic region (i.e., the United States Geological Survey Region No. 18). As such, the model may not be applicable to basins outside this spatial range as other hydrologic processes may dominate other basins. We can demonstrate this by observing the spatial variability (i.e., the coefficient of variation) in precipitation across the United States (Figure 1.5; Dettinger et al., 2011).

In addition to concerns with spatial extrapolation, there is temporal extrapolation. Climate change emposes **non-stationarity** in environmental variables like precipitation and temperature. Empirical models for flow should not be used to extrapolate beyond the limits of the variables the model observes or it will risk large errors. However, many advances in time-series analysis handle non-stationarity in data; one can reduce the process to a stationary one (i.e., trend seasonality and noise can be decomposed) or consider these processes as stochastic.

Another downside is the **complexity in model structure**, especially in ensemble statistical learning methods, sometimes referred to as black-box models. If inference, or model parameters, are of interest, complex models introduce challenges. Dimensionality reduction

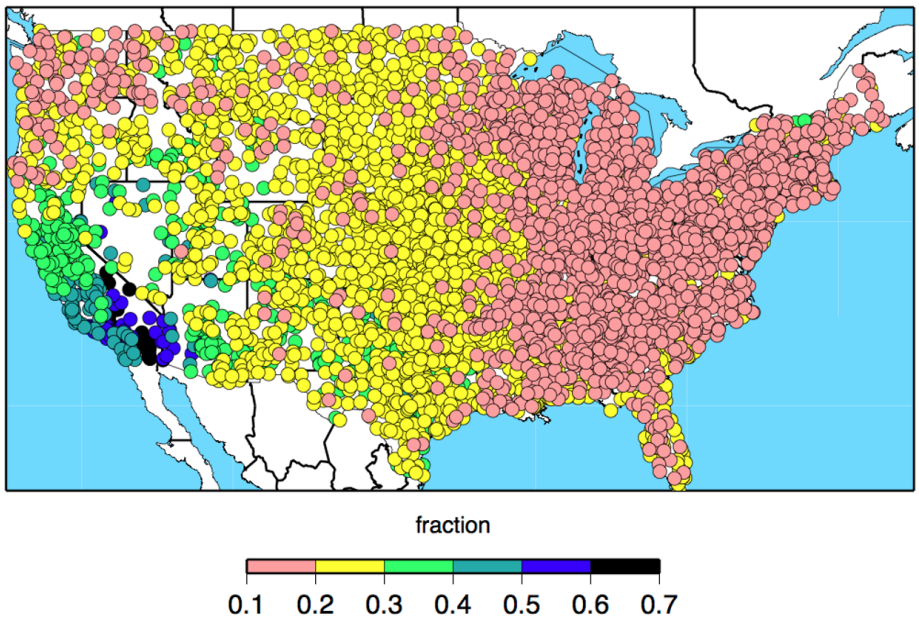


Figure 1.5: Coefficient of variation in total precipitation from 1951-2008. Reprinted from Dettinger et al., 2011

.

methods (e.g., principle component analysis, partial least squares) and regularization techniques in regression (e.g., ridge, lasso, and elastic net) can help reduce the number of model parameters, and systematically produce simpler models (J. Friedman et al., 2001).

The essential **arbitrariness in the selection of the form** of an empirical model is another drawback (Klemes, 1982). Most studies report using one modeling method, which perhaps suggests that researchers are not employing more than one modeling method. Such a study could provide insights into the system by revealing the sensitivity of results to the algorithms employed. Therefore, the application and comparison of different machine learning models to the PUB problem was considered in this study.

Lastly, some limitations are caused by the **nature of the algorithms** deployed. For example, regression-based random forest models make predictions by averaging predictions made by multiple regression trees. Therefore, the ensemble model limits the predictions it makes to the range seen in the training data; the predictions do not extrapolate to ranges not seen in the training data. In fact, averaging dampens the density function when we compare observed to predicted data. This is especially problematic where the extreme tails

of the distribution (i.e., floods and droughts) are of interest. Another example is that of the SVMs mentioned before that seem to perform poorly on low rainfall data.

1.5 Conclusion

Generally, in statistical learning, applications lag behind advances in theory; the application of statistical learning theory to water resources problems is still in the bulldozer era, so, most models are computationally expensive. In the past two decades, in hydrology, statistical learning methods have been applied to modeling rainfall-runoff processes, predicting streamflow temperatures, sediment and nutrient loadings, forecasting the groundwater heads in an aquifer, or water demand among many others.

This chapter's main contribution is a heuristic guide to empirical model selection. Like a flowchart, it guides in selecting methods tailored to general purposes and limitations of various empirical modeling approaches. This guide should help in selecting from range of possible methods that are well suited to a problem at hand and give some comparative insights on these diverse methods. As a heuristic it works in most cases, but it is not comprehensive or applicable to all problems.

In some cases, a wide range of empirical models can be employed, suggesting that no one single modeling method is useful across all locations, timescales, and problems. Also, despite their other limitations discussed in this chapter, these methods are much easier, faster, and less expensive to apply and study than mechanistic models. They are well suited to dynamic, non-linear and sometimes noisy data, especially when underlying physical processes are complex or not fully understood. In addition, the purpose of modeling is often to inform decision makers with adequate timing. For example, models need to be run during and just before flood events. Real-time applications require rapid computation, which statistical methods provide. The merits of statistical learning techniques, as a subset of empirical models, motivate their study in the following chapters.

This dissertation follows steps outlined in Figure 1.6. Chapter two compares two different data transformations that reflect our philosophical view of hydrology. Chapter three explores the effect of the loss functions on the model parameters and estimates. Chapter four compares different resampling methods for test error approximation. Chapter five discusses model

development for non-stationary hydrologic processes.

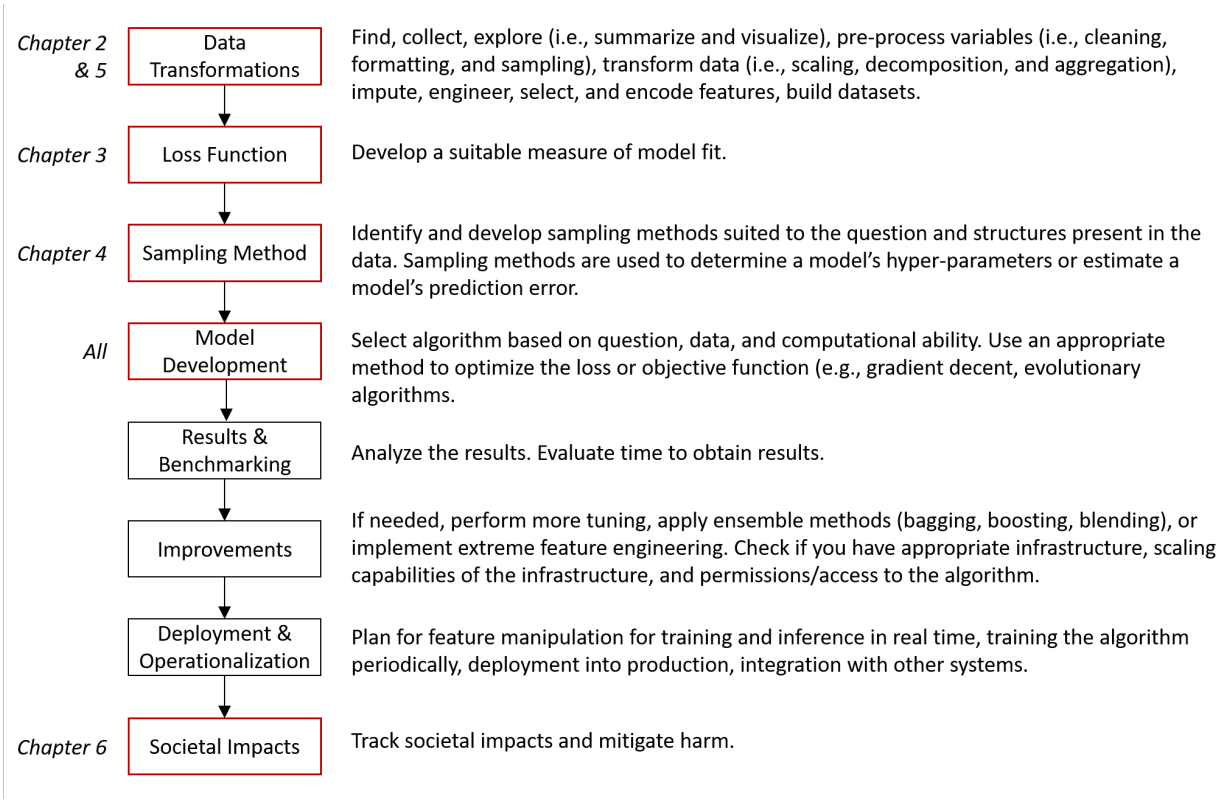


Figure 1.6: Statistical learning steps. Adapted from Brownlee, 2014; Ingle, 2017. Each chapter of this dissertation discusses a unique step.

Chapter 2

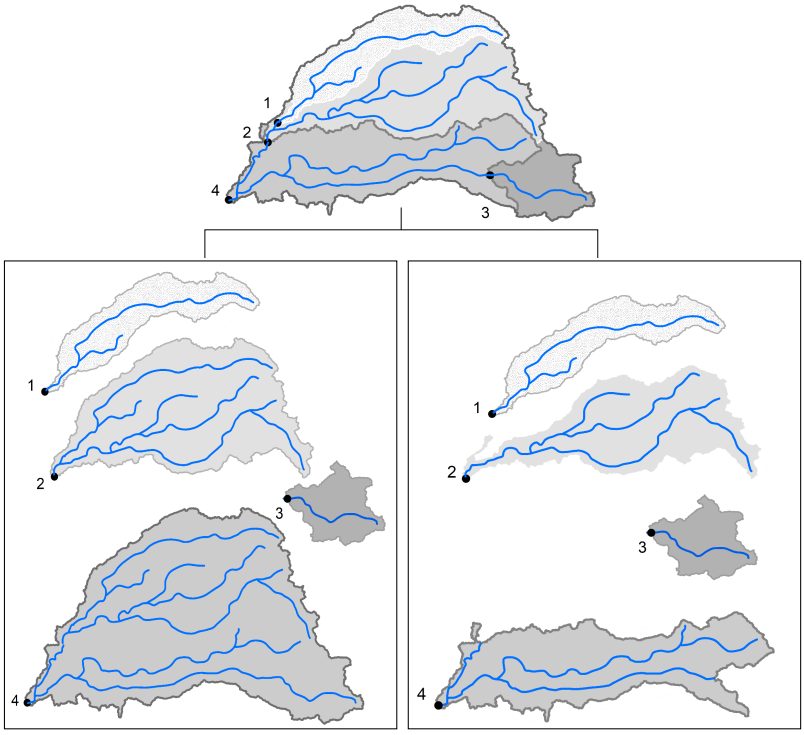
Data Transformations: Two Philosophies on Hydrologic Processes

> Science is what we know, and philosophy is what we don't know.
>
> Bertrand Russell, *"Unpopular Essays"*, 1950

Summary

This chapter developed linear multivariate regression (LM), generalized linear regression (GLM), random forest (RF), and Neural Network (NN) models with a typical least squares loss function, of monthly unimpaired flows in 67 California basins. The best overall error (Bias-Corrected Coefficient of Determination, bR^2=0.92, Nash-Sutcliffe Efficiency, NSE=0.97) reflects the model's ability to capture monthly variations in flow. The NN with "incremental basins" performed the best in the NSE criterion. Incremental basins are segments of the basin that have not been gauged, a concept discussed in this chapter.

The test set error from "leave one group out" (LOGO) cross-validation shows that model quality in predicting unimpaired flow is very spatially variable. LOGO cross validation and other resampling strategies are discussed in the next chapter. A comparison of different models concludes that the incremental basin approach to hydrologic modeling provides increasing benefits as the outlet of interest moves downstream in the gauge network.



(a) Aggregate Basins (b) Incremental Basins

Figure 2.1: A basin's hydrologic response can be interpreted in two fundamentally different ways: (a) aggregate basins, where each basin's response is a function of all the land above the outlet that drains to the outlet, or (b) incremental basins, where each piece of land below an outlet incrementally alters the observed flows from gauges above it.

2.1 Introduction

Unimpaired flows can be presented in two fundamentally different ways: (1) we can imagine each basin as a separate function that transforms its inputs (precipitation and snow) into runoff (or unimpaired stream flow). Here, we define each basin to be an "aggregate" basin. Flows for these basins are simply the observed gauge values (Figure 2.1a); (2) we can imagine the basins as interconnected, and overlapping. One stream flows into another, like in a network, and so, some basins overlap. Here, we can define "incremental" basins to be segments of basins that do not overlap. Flows for these basins are the amount that has not been observed by gauges in the network above the outlet of interest (Figure 2.1b). Therefore, when modeling incremental flows, the network information is being preserved.

In this chapter, we have two types of data pre-processing: aggregate and incremental basins. Each data transformation method reflects our way of viewing hydrologic processes.

Neither is "right" as they are merely philosophical views of hydrologic processes.

2.2 Methods

2.2.1 Model Types and Loss Functions

The models considered and their parameters are explained below (Table 2.1).

Table 2.1: Model types and their parameters.

Model type	R package	Parameters defined in model formulation	Parameters selected through cross validation
LM	stats	not applicable	not applicable
GLM	stats statmod	family=Tweedie link.power=0 maxit=1000	var.power=1.1
RF	randomForest	ntree=500 sampsize=length(training set) nodesize=5	mtry=20
NN	keras	batch_size=25 validation_split=0.2	epochs=100

Linear Multivariate Regression Models

In 1805, Adrien Marie Legendre introduced the least squares method of estimating parameters as an appendix to his book on the paths of comets. A few years later, Carl Freidrich Gauss also published the method (Stigler, 1981). The method is brought to perfection with its application to linear regression and curve fitting.

Linear Multivariate Regression models (LM) are customarily made of systematic and random error components, where the errors are usually assumed to have Normal distribution (Equation 2.1).

$$\begin{aligned} &Y \sim N(\mu, \sigma^2)\text{: random} \\ &\mu = X\beta\text{: systematic} \end{aligned} \tag{2.1}$$

Given the model, the fitted values can be estimated by Equation 2.2.

$$Y_i^{sim} = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_i X_{ki} \tag{2.2}$$

The unknown parameters in Equation 2.2 are: β_0 (the overall mean) and β_k (the regression coefficients). To find the best fit, much like simple linear regression, we need to estimate the unknown parameters by minimizing a loss function, customarily the residual sum of squares (RSS) (Equation 2.3).

$$\begin{aligned} RSS &= \sum_{i=1}^{n} e_i^2 \\ &= \sum_{i=1}^{n} (Y_i^{obs} - y_i^{sim})^2 \\ &= \sum_{i=1}^{n} (Y_i^{obs} - \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_i X_{ki}) \end{aligned} \tag{2.3}$$

The `lm()` function in R constructs LMs. They are easy to understand and interpret, which makes them a great first cut at predictive modeling. However, they over simplify reality (hydrologic processes are not linear) and lack precision (as demonstrated by the goodness-of-fit measures). Another major flaw is that a linear predictor can give predictions that are physically impossible (e.g., negative flows). Here, the variance cannot be considered constant since there is a boundary on the response. These shortcomings can be overcome with generalized linear models.

Generalized Linear Regression Models

In 1972, Nelder and Wedderburn introduced Generalized Linear Regression models (GLM). This work allowed for a unified fitting procedure, despite the type of error distribution, based on likelihood (Nelder & Wedderburn, 1972). Therefore, unlike LMs, GLMs can accommodate non-Normal distributions of error. However, except for Normal distributions most other distributions do not have a closed-form solution.

In GLMs, the linear model is related to the response variable via a link function. This function allows the magnitude of the variance of each measurement to be a function of its

predicted value. Therefore a GLMs components are (Equation 2.4):

$$\begin{aligned} &Y \sim P(\mu, \phi)\text{: random} \\ &g(\mu) = X\beta\text{: systematic} \end{aligned} \tag{2.4}$$

Where P is the distribution of random errors, and $g(\mu)$ is the link function. P and g can be specified by the user.

The `glm()` function in R constructs GLMs. The GLMs developed here are characterized by the *Tweedie distribution*, since the outcome (i.e., unimpaired flow) is continuous, non-negative, skewed, and unbalanced with exact zeros. Tweedie distributions are a special case of exponential dispersion models where the variance function is a power function (Equation 2.5), and the link, or the function used to explain how the expectation of the outcome is related to the linear predictor can be specified in terms of Box-Cox transformations (Jorgensen, 1997).

$$var(Y) = V(\mu)\phi = \mu^{\alpha}\phi \tag{2.5}$$

The power, alpha, can be set by the user, or determined through cross-validation. Special cases include Normal (α=0), Poisson (α=1), Gamma (α=2), and inverse-Gaussian (α=3) GLMs . Here, we set the power α to be 1.1. The link g can be specified as log or identity. Here, we used the log link.

Therefore, the above model assumes that $y_i \sim Tweedie_{\alpha}(\mu_i, \phi)$ where

$$var(Y_i) = \mu_i^{1.1}\phi$$

and

$$log(\mu_i) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_i X_{ki}$$

The regression coefficients, β_j, were estimated by maximum likelihood. The dispersion parameter, ϕ, was estimated using the residual sum of squared residuals, otherwise called the Pearson estimator.

Both LMs and GLMs are parametric models. For prediction purposes non-parametric methods are proven to perform better since their form is shaped by the data and not fixed a priori. Therefore, next, we consider a non-parametric modeling method, random forests.

Tree Building Algorithms

Classification and Regression Trees (CARTs) involve stratifying or segmenting the predictor space, into a number of regions, using a series of if-then statements. At each internal node in the tree, a test is made to one of the inputs. Depending on the outcome of the test (or split rule), the algorithm goes to either the left or the right sub-branch of the tree. Eventually the algorithm arrives at a terminal node, which contains a prediction. The prediction for a given observation is the mean or the mode of the training observations in the region to which it belongs (Breiman, Friedman, Stone, & Olshen, 1984).

In essence, each tree is a series of split rules. The split rule is found using a greedy top-down search for recursively splitting of the data into binary partitions. It is greedy, because, the split rule at each internal node is selected to maximize the homogeneity of its child nodes, without consideration of nodes further down the tree, yielding only locally optimal trees (Grubinger, Zeileis, Pfeiffer, et al., 2011). For regression trees, the mean of all the observation points that fall within a branch is considered the prediction of that branch in the tree. The best tree is one which has the minimum test error rate calculated by the RSS.

Since trees have a finite number of terminal nodes (CARTs are pruned based on a complexity parameter, α), the prediction of these methods are discrete, and therefore, not particularly suited to modeling a continuous variable. In addition, CARTs suffer from high variance; trees grown on different subsets of the training set will produce different predictions. This phenomenon is one of the major drawbacks of CARTs. Methods such as *bagging* (Breiman, 1996), *random forests* (Breiman, 2001), *boosting* (J. H. Friedman, 2001) and *bumping* (Grubinger, Kobel, & Pfeiffer, 2010) attempt to improve the prediction accuracy of trees with the idea that combining and averaging trees reduces variance.

A Random Forest (RF) consists of an assemblage of unpruned CART models. Each CART model in an RF is different because it is grown using: (1) a new training set: in each bootstrapped training set, about one-third of the instances are left out; and (2) random feature selection: each time a split in a tree is considered, a random sample of predictors is chosen as split candidates from the full set of predictors. This process de-correlates the trees. This strategy, using a random selection of features to split each node, introduces some

randomness that improves the accuracy of the predictions of the trees as a whole and yields error rates that are robust with respect to noise (Breiman, 2001).

The `randomForest()` function in the `randomForest` library (Liaw & Wiener, 2002), constructs RF models. This function takes in tuning parameters such as mtry, ntree, sampsize, and maxnodes:

`mtry`: In RFs, internal estimates monitor error, strength and correlation, which are used to show the response to increasing the number of features used in the splitting. Here, this parameter was set to 20 out of the full 25 predictor variables available found through cross-validation.

`ntree`: The generalization error of a forest of trees depends on the strength of the individual trees in the forest and the correlation between them (Breiman, 2001). This error converges to a limit as the number of trees in the forest increases. Here, the number of trees was set to the default 500.

`sampsize`: In RFs, the trees are built on a bootstrap sample of the training data, a sample equal in size to the original dataset, but selected with replacement. Therefore, some observations are not selected, and others are selected more than once. Here, the sample size is set to the default value, the length of the training set.

`maxnodes`: Using the maximum number of terminal nodes, the user can "prune" the trees back to a smaller version of itself. Here, we used the default value, which is a function of `nodesize` or the allowed minimum number of observations in each node. The default value for nodesize is 5.

Like LMs, RFs also typically use the RSS loss function to find the optimal split value. For more information about loss functions see Chapter 3.

Neural Networks

In 1951, Marvin Minsky and graduate student Dean Edmonds built the first neural network (NN) machine. This machine was a randomly connected network of capacitors that have a finite amount of memory and time to keep or remember that memory. The memory holds the probability that a signal will come in one input and another signal will come out of the output. This machine, modeled after the Hebbian theory of learning in the human brain, was one of the first pioneering attempts at artificial intelligence. Shortly after, in

1957, Frank Rosenblatt invents the perceptron, the first **neural network** for computers.

[INSERT neural network graph, equation, chain rule for finding optimal parameters, insert discussion on model variables and their specifications]

2.2.2 Test Error Approximation

Blocking cross-validation is used to approximate the test set error. Here, all data for the basin to be modeled is left out of the training data and becomes the test set (i.e., leave one group out cross validation). Therefore, the training data is the data from all the other basins. This process was repeated for all basins in the study, and so, for model evaluation, one LM, GLM, RF, and NN model exists for each basin. For more information about resampling methods see Chapter 4.

With the developed model's predictions and the observations in the test set, we can calculate the desired model goodness-of-fit: Bias-Corrected Coefficient of Determination (bR^2) and Nash-Sutcliffe Efficiency factor (NSE) (Equations 2.6, 2.7, and 2.8). See appendix D for more model measures-of-fit.

$$R^2 = \left(\frac{\sum_{i=1}^{n} \left(Y_i^{obs} - \overline{Y^{obs}} \right) \left(Y_i^{sim} - \overline{Y^{sim}} \right)}{\sqrt{\sum_{i=1}^{n} \left(Y_i^{obs} - \overline{Y^{obs}} \right)^2} \sqrt{\sum_{i=1}^{n} \left(Y_i^{sim} - \overline{Y^{sim}} \right)^2}} \right)^2 \quad R^2 \in [0, 1] \tag{2.6}$$

R^2 is insensitive to additive and proportional difference between model simulation and observations. One can simply show that for a non zero value of β_0 and β_1, if the predictions follow a linear form, $Y^{sim} = \beta_0 + \beta_1 Y^{obs}$, the R^2 equals one (Legates & McCabe Jr, 1999). Therefore, for a proper model assessment, it is recommended that the slope of the predicted vs. observed graph be reported or systematically included as in Equation 2.7.

$$bR^2 = \begin{cases} |b| \, R^2 & \text{for b} \leq 1 \\ |b|^{-1} \, R^2 & \text{for b} > 1 \quad bR^2 \in [0, 1] \end{cases} \tag{2.7}$$

By weighting R^2, under and over predictions are quantified together with the model dynamics which results in a more comprehensive reflection of model results.

Another commonly used model goodness-of-fit is the the Nash-Sutcliffe efficiency factor (Equation D.11).

$$NSE = 1 - \frac{\sum_{i=1}^{n} \left(Y_i^{sim} - Y_i^{obs}\right)^2}{\sum_{i=1}^{n} \left(Y_i^{obs} - \overline{Y^{obs}}\right)^2} \quad NSE \in (-\infty, 1] \tag{2.8}$$

A Nash-Sutcliffe efficiency factor of lower than zero indicates that the mean value of the observed time series would have been a better predictor than the model. Like the bR^2, the largest disadvantage of the Nash-Sutcliffe efficiency factor is the fact that the differences between the observed and predicted values are calculated as squared values. As a result, larger values in a time series are strongly overestimated whereas lower values are neglected (Legates & McCabe Jr, 1999). For the quantification of runoff predictions this leads to an overestimation of the model performance during peak flows and an underestimation during low flow conditions (Krause, Boyle, & Bäse, 2005).

2.2.3 Post-Processing

In post-processing all model predictions are modified so as to be comparable to original gauge flows; all cumulative values are back-transformed into non-cumulative forms, and all incremental basins are back-transformed into aggregate forms. In other words, the steps used in pre-processing are reversed so as to fairly compare the goodness-of-fit across all models.

2.3 Results

2.3.1 Model Evaluation

Figure 2.2 shows the predicted unimpaired flows versus the observed for each model type in order of increasing NSE. A perfect model will follow the $y = x$ line. The regression line shows a tendency for the LM and RF aggregate and incremental models to under predict and for the GLM aggregate and incremental models to slightly over predict the unimpaired flows. Under predicting flows are generally bad in times of floods and over predicting flows are generally bad in times of droughts, since both lead to decisions that are both inaccurate and not conservative. The NN outperforms other models and it is somewhat insensitive to the input data pre-processing.

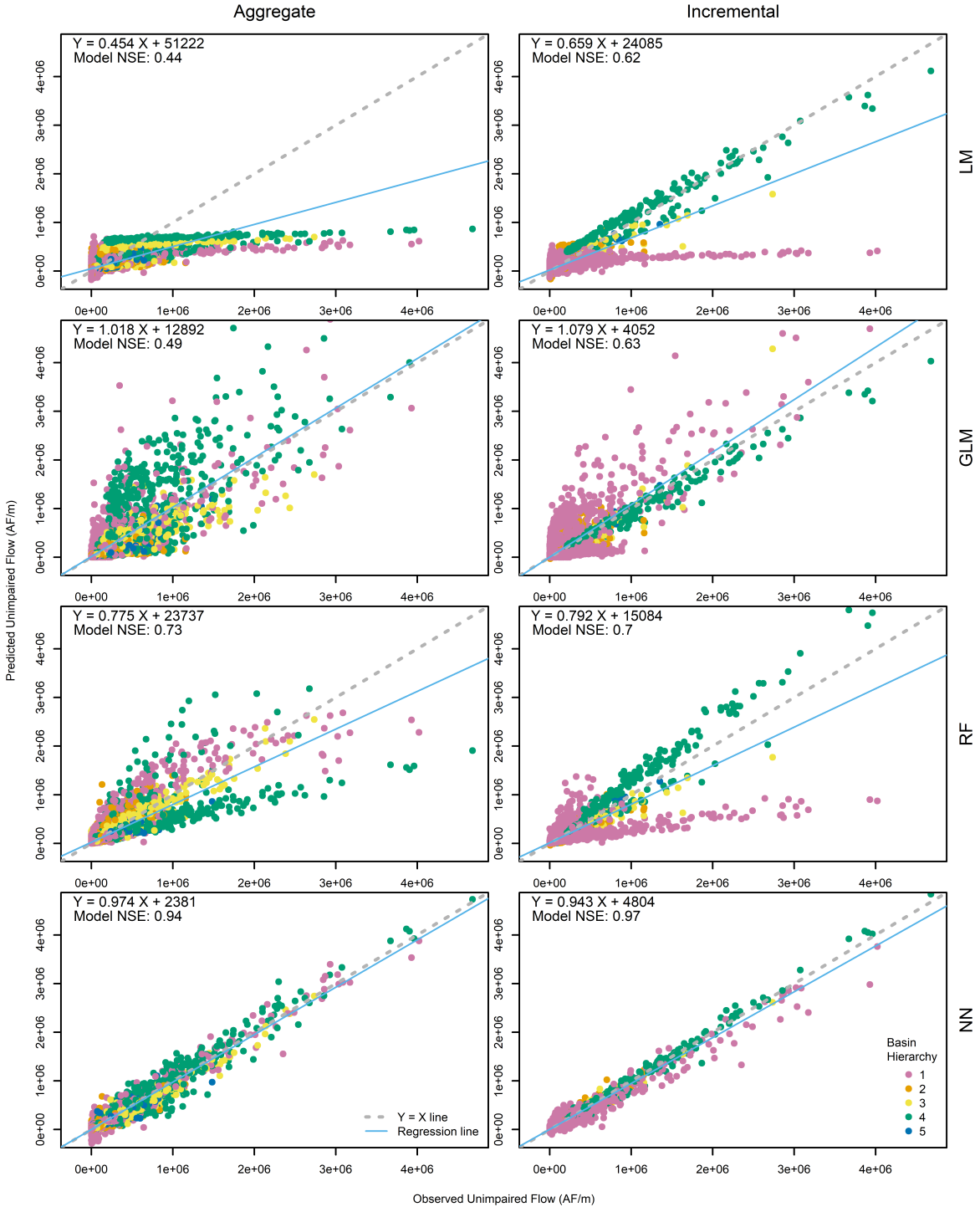
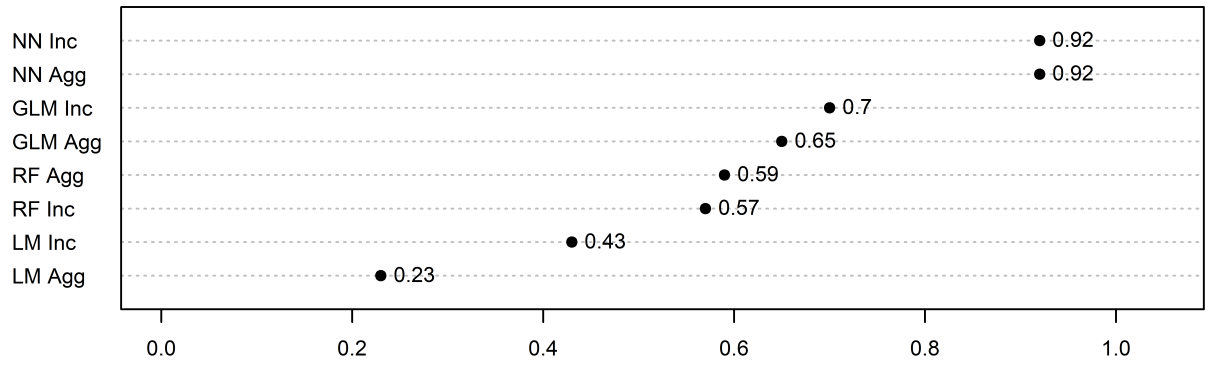
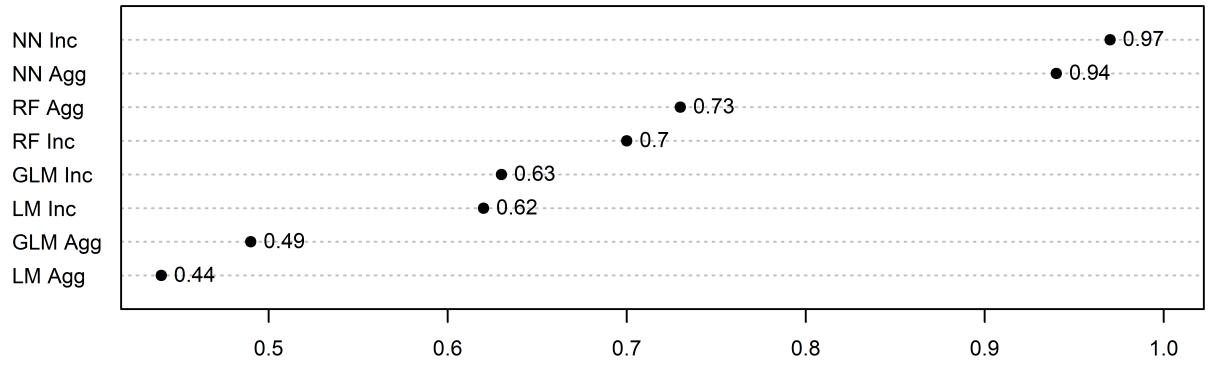


Figure 2.2: The models trained on the four types of data (i.e., aggregate, incremental, cumulative aggregate, and cumulative incremental). The LM generally under predict unimpaired flows and show a bad fit. The GLMs slightly over predict unimpaired flows, but show a better fit. The RF generally over predict unimpaired flows. The RFs, non-parametric models, are a big improvement compared to the linear models, parametric models. Lastly, the NNs, out perform all models.



(a) bR^2



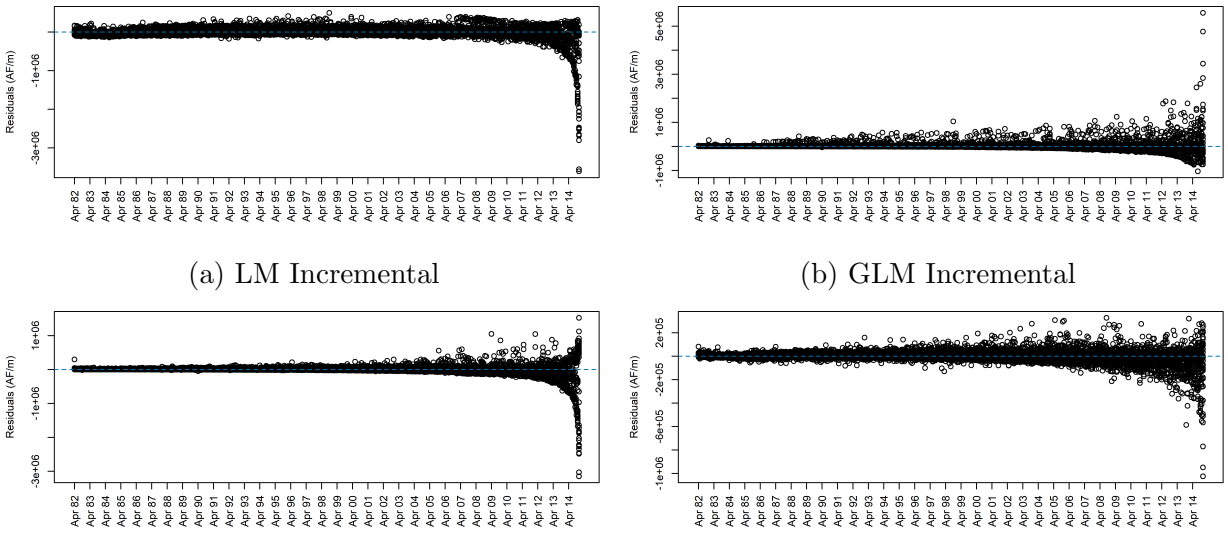
(b) NSE

Figure 2.3: The goodness-of-fit of models trained on the two types of data (i.e., aggregate and incremental) as measured by the Coefficient-of-Determination (bR^2) and Nash-Sutcliffe Efficiency (NSE). The NN aggregate and incremental model provides the best model performance in the bR^2 and NSE respectively.

Figure 2.3 shows how each model scores as to the bR^2 and NSE. In the LM and GLM the incremental modeling method performs better than the aggregate. In the RF and NN their performances are very similar.

Figure 2.4 shows that the models all decrease in accuracy over time. We can hypothesize that this is due to climate change imposing non-stationarity on the hydrologic process. This is further discussed in chapter 5.

In the NSE, the NN incremental model provides the best model performance, therefore, we will now abandon the comparative analysis and examine the spatial distribution of this model.



(a) LM Incremental

(b) GLM Incremental

(c) RF Incremental

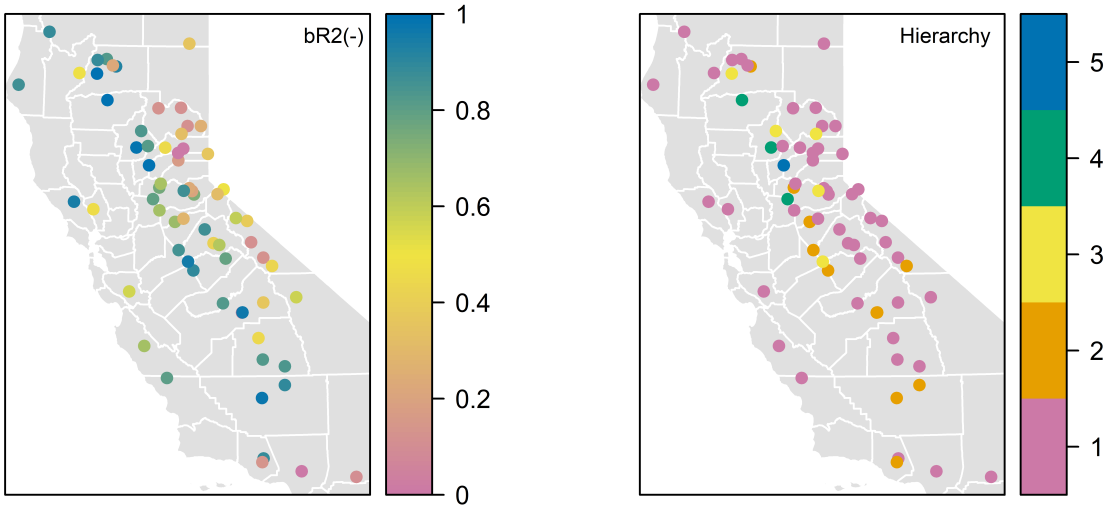
(d) NN Incremental

Figure 2.4

2.3.2 Spatial Distribution of Errors

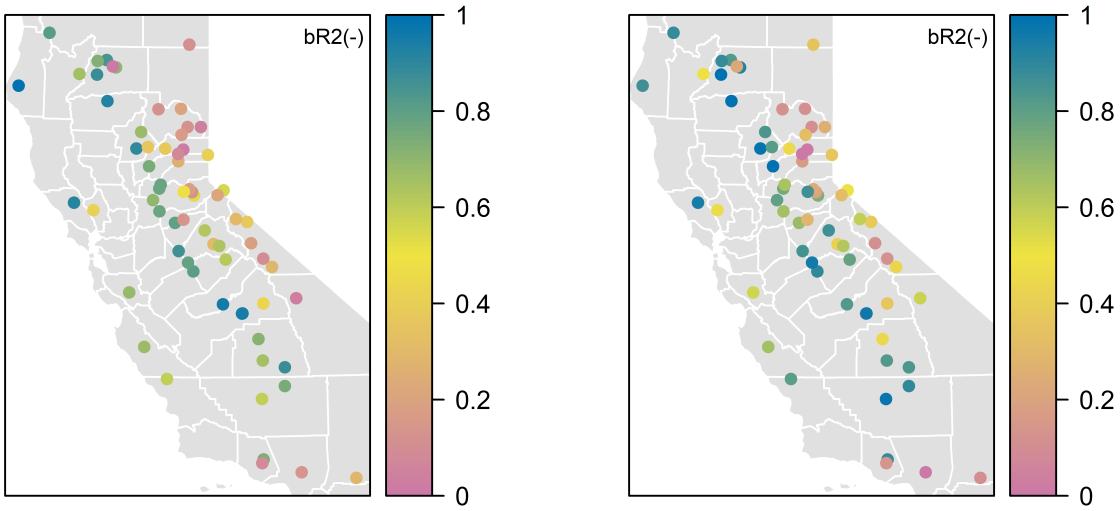
Figures 2.5 and 2.6 show the bR^2 values for the 67 basins in this study. As expected, the model's ability to predict unimpaired flow varies across California. The model performs better at basins lower in the network (i.e., have a higher hierarchy). This could be due to: (1) the basins with higher hierarchies generally have larger flows and the model is trained with a squared error loss that penalizes large errors more harshly; or (2) there was substantial value in having flow information upstream (i.e, the decline is error is due to having incremental basins).

Figures 2.7 and 2.8 show that when there is no flow information upstream (i.e., hierarchy=1) there is not much difference in the performance of incremental and aggregate models. However, when we introduce increasingly more information upstream (i.e., hierarchy=2,3,4, and 5) the NN incremental model can perform much better than the NN aggregate model. Even though the data set is smaller in the basins lower in the network, we can conclude that, the model is performing better at these basins due to information upstream and not just due to its higher flows and the loss function.



(a) Test Set Error (b) Basin Hierarchies

Figure 2.5: The spatial distribution of errors. (a) The bR^2 error is not random and follows a line down the middle of California, and it somewhat follows the basin hierarchies. (b) The basins are not evenly distributed between the hierarchies; the lower the hierarchy the more basins in this study. Altogether, the lower the basin is in the network (i.e., the higher its hierarchy), the better the model performs.



(a) NN Aggregate (b) NN Incremental

Figure 2.6: The aggregate and incremental basins perform very similarly when there isn't any information upstream (i.e., hierarchy=1). However, when we introduce information upstream (i.e., hierarchy=2,3,4, and 5) the incremental basins can perform much better than the aggregate.

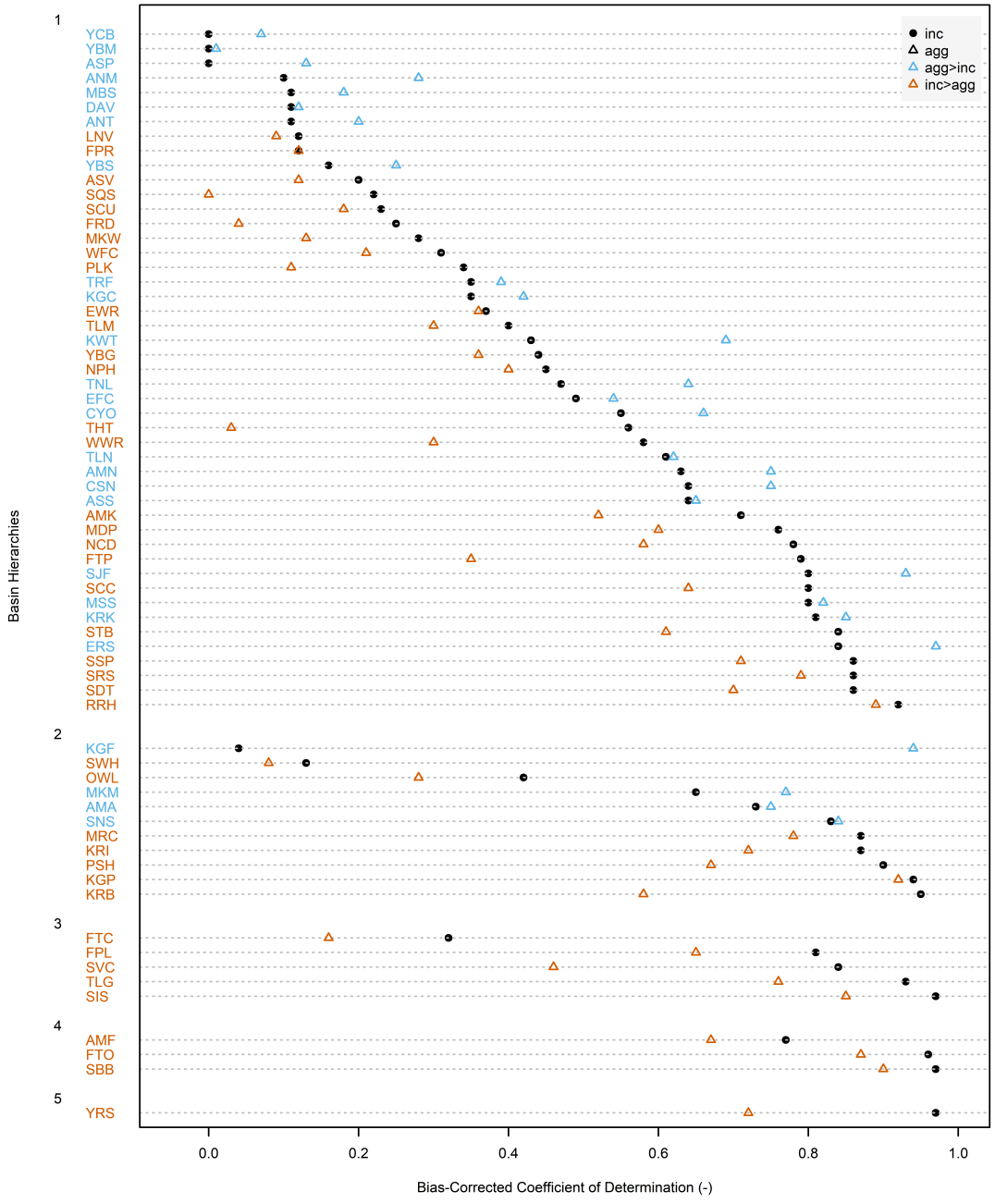


Figure 2.7: Basin bR^2 performance in order.

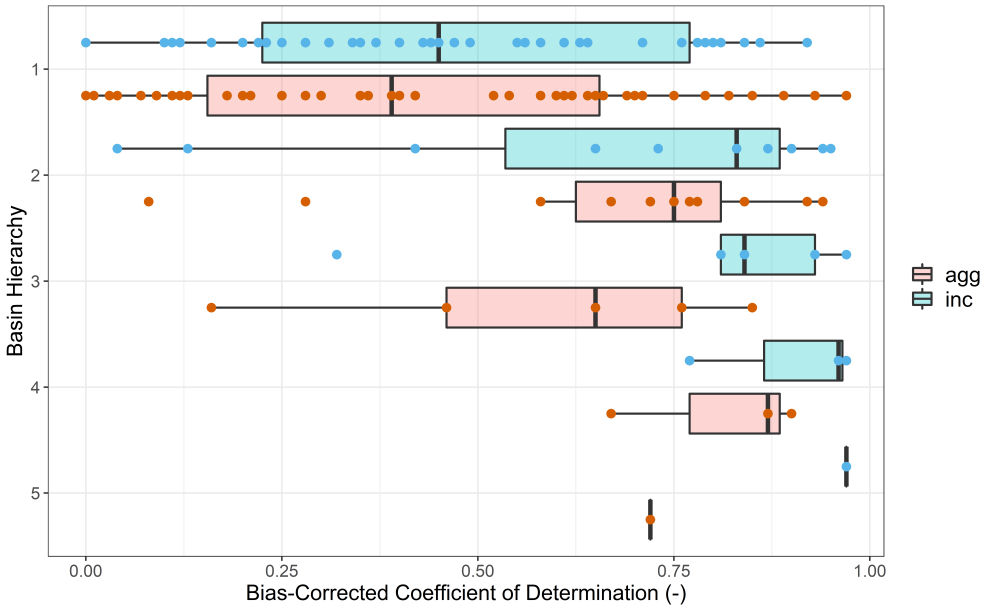


Figure 2.8: The incremental and aggregate basins perform very similarly when there is no information upstream (i.e., hierarchy=1). However, when we introduce information upstream (i.e., hierarchy=2,3,4, and 5) the incremental basins can perform much better than the aggregate.

2.4 Conclusion

Incremental basin modeling provides an easy way to include network information in statistical models and the results show that it is valuable for modeling hydrology with parametric models especially those that have a few parameters like the LM and GLM. As the results showed, the LM and GLM prefer the incremental modeling approach, whereas the RF and the NN are somewhat insensitive to it.

On this data set, and according to the performance ratings provided by Moriasi et al. (2007), the GLM and RF provide a "good" prediction for unimpaired flows, and the NN provides a "very good" one (Table 2.2). We can hypothesize that the RF performed well due to the nature of non-parametric methods where the model form is determined by the data and hence the data is "honored" and prediction becomes easier. The NN performance is the best and proves why these methods are so popular in studies in hydroinformatics.

In another experiment, the models were trained on their cumulative flows and cumulative rainfall. Given that snow-melt driven hydrology dominates the Sierra-Nevada basins, and processing the data to its cumulative forms would have given the model a "memory" effect,

Table 2.2: Model performance ratings. Criteria are given by Moriasi et al., 2007 (Appendix D).

Model	Aggregate	Incremental
LM	Unsatisfactory	Satisfactory
GLM	Unsatisfactory	Satisfactory
RF	Good	Good
NN	Very Good	Very Good

we repeated the experiment in this chapter with cumulative values. However, surprisingly, none of the models were able to provide satisfactory results, and therefore, we have therefore left those results out of this chapter.

The next chapter will explore one flaw we pointed out here: the squared loss function forcing better predictions at flood levels at the expense of drought level data. We will be training models using different asymmetric loss functions to penalize the under predicting of floods and over predicting of droughts at a higher cost (i.e., forcing the model to reach the peaks and valleys of the hydrograph).

Chapter 3

New Loss Functions: Comparing Asymmetric and Symmetric Methods of Evaluating Error

> Maybe all one can do is hope to end up with the right regrets.
>
> Arthur Miller, *"The Ride Down Mount Morgan"*, 1991

Summary

In practice, the loss function for a chosen statistical learning method is the translation of a informal philosophical objective into the formal language of mathematics (Hennig & Kutlukaya, 2007). Therefore, the choice of a loss function in estimation is somewhat subjective and depends on the specific application of the model or the decisions being made when using it. Some loss functions have already been established in and are common in hydrology.

This chapter will look at differences in performance of already established and new functions in hydrology when they are embedded in the machine learning algorithm rather than using them as only an evaluation step after the model is built.

3.1 Introduction

Mechanistic models in hydrology simulate conditions based on available input parameters, modeled processes, and calibration to specific locations. *Measures-of-fit* or the similarity of the simulations to the observations help in assessing model performance. Visual similarity

is recommended first (i.e., the plot of observed and simulated time series), and calculated measures-of-fit are recommended next. In model calibration, these measures can help guide better fits of simulations to observations.

In statistical learning, the same process can be used by estimating the model using a pre-defined loss function, solve the simulation problem as well as is possible, and then calculate the model measures-of-fit of interest. However, improving a model trained on a different loss function than that which is desired can be quite tricky. Since the machine learning algorithm requires a loss function to begin with, we can directly define the custom loss function as the measure-of-fit of interest before deploying the learning algorithm. This section performs statistical learning with different loss functions, and then examines the differences in predictions.

Typical loss functions in statistical learning are the $\ell 1 - norm$ and $\ell 2 - norm$ (See Equation 3.1 and 3.2). The $\ell 2 - norm$ is the familiar objective function in simple least-squares regression, a convex function, emphasizing points distant from the bulk of the data.

$$\ell_1(y_i, \hat{f}(x_i)) = ||y_i - \hat{f}(x_i)||_1 = \left| y_i - \hat{f}(x_i) \right| \tag{3.1}$$

$$\ell_2(y_i, \hat{f}(x_i)) = ||y_i - \hat{f}(x_i)||_2^2 = \left(y_i - \hat{f}(x_i) \right)^2 \tag{3.2}$$

Risk, or cost, is defined as the expectation of the loss function. For example, the risk of over predicting the severity of a drought can be defined as *how much* it was over predicted on average. This distance can be defined as the absolute value of the difference or the difference squared as in Equations 3.3 and 3.4, the empirical risks associated with the $\ell 1 - norm$ and $\ell 2 - norm$. The expectation of the $\ell 2 - norm$ will produce a model that regresses to the mean, and the $\ell 1 - norm$ regresses to the median. That is, the $\ell 2 - norm$ is more sensitive to outliers than the $\ell 1 - norm$.

$$L_1(y_i, \hat{f}(x_i)) = E\left[\ell_1(y_i, \hat{f}(x_i)) \right] = \frac{1}{n} \sum_{i=1}^{n} \left| y_i - \hat{f}(x_i) \right| \tag{3.3}$$

$$L_2(y_i, \hat{f}(x_i)) = E\left[\ell_2(y_i, \hat{f}(x_i))\right] = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 \tag{3.4}$$

On the other hand, *Regret* is the difference between the consequences of a sub optimal decision and the optimal decision. Often, in reinforcement learning, the objective is to minimize regret, which is equivalent to maximizing the highest accumulated reward (Sutton & Barto, 2018). For example, maybe over predicting the severity of a drought this year will lead to better management of resources and fewer regrets in later years.

In this research, to avoid developing a mathematical representation for regret, we leave this discussion and proceed with the much simpler *risk-minimization framework* (See Equation 3.5).

$$\hat{f}(x_i) = \underset{\tilde{f}}{\operatorname{argmin}}\ E\left[L(y, \tilde{f}(x))\right] \tag{3.5}$$

3.2 Research Design

This study used the monthly unimpaired flows data set developed and maintained by the California Data Exchange Center (CDEC). Unimpaired flow is the flow produced by the basin in its current state, but, without dams and diversions (California Department of Water Resources, Bay-Delta Office, 2016).

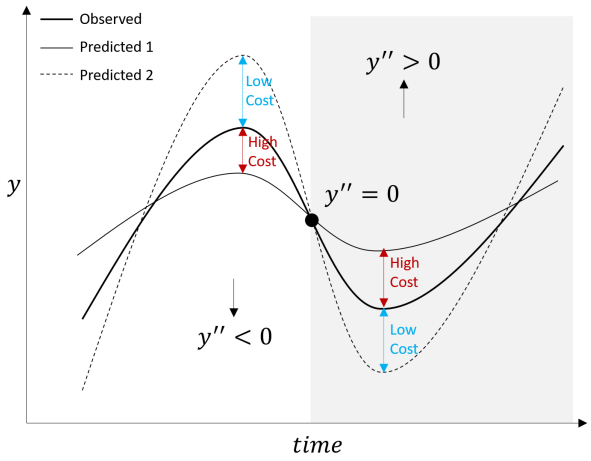
The data spans 69 California basins (See appendix **??**, Figure A.1) from 1982 to 2014. It can be downloaded with the `sharpshootR` package in R (Beaudette, 2016). 28 predictor attributes were calculated for each observation point based on the knowledge of basin characteristics and processes that influence a watershed's response to precipitation: evaporation (temperature); snowfall (cumulative sum of precipitation below 2°C); storage in soil (with soil and land cover parameters); antecedent conditions (with lagged precipitation and temperature parameters); and groundwater processes (with geology and depth to a restricted layer) (See Table A.1). The dataframe has approximately 18,500 monthly unimpaired flow observations in acre-feet (AF) and as a continuous variable can be used for regression type studies.

Typical measures-of-fit developed in hydrologic modeling are the Mean Absolute Error (MAE), Relative Standard Deviation (RSD), Relative Mean (RMU), Mean Squared Error (MSE), Root Mean Square Error (RMSE), normalized RMSE (nRMSE), RMSE standard deviation ratio (RSR), Percent Bias (PBIAS), Coefficient of Determination (R^2), Nash-Sutcliffe Efficiency (NSE), Index of Agreement (d), Modified NSE, Modified d, Relative NSE, Relative d, King-Gupta Efficiency (KGE), and Volumetric Efficiency (VE). appendix **??** presents their equations, strengths, and weaknesses. First, let us consider a list of characteristics that the loss function, in its application to hydrologic prediction, should fulfill:

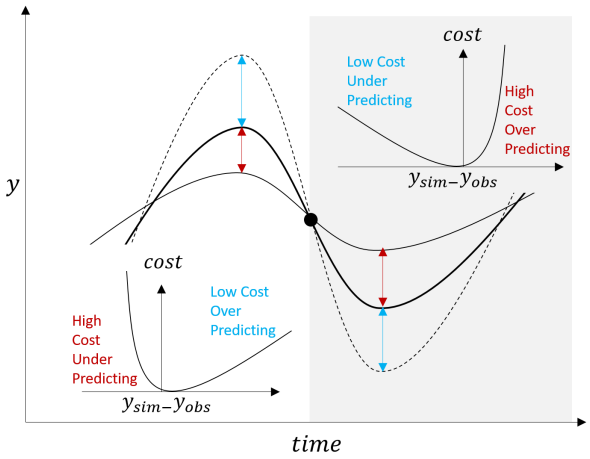
(1) **Should the loss function be symmetric?** In symmetric functions, under predicting produces the same loss as over predicting of the same absolute error. However, a conservative loss function applies a different penalty to the different directions of loss. That is, an asymmetric loss function can force the model to over predict the severity of floods and droughts rather than under predict them. This approach requires the labelling of all instances of the data as either a peak, normal, or drought point, requiring a labeling mechanism (i.e., a classification model) before running the predictive regression model.

Great care should be taken not to introduce "data leakage", or the leakage of information from the response variable into the final predictive model; the classification model will have to either be trained on the predictor variables only, or use a portion of the data that is thrown away for the rest of the study. A simple classification model can be defined by a fit of a thin plate spline to the precipitation data with a predefined degree of smoothness (i.e., degree of freedom). Next, we find the points at which the direction of curvature (the second derivative) in the time series changes. Areas where the curvature is upward can be labelled drought and downward labelled flood.

After all data points are labeled, we define two asymmetric loss functions for each peak and valley section (See Figure 3.1). Such loss functions can be defined as linear exponential (LINEX) loss if smoothness is desired (See Equation 3.6). However, current subgradient-based and derivative-free methods of optimization in convex programming can easily handle non-differentiability at the origin of the loss function. In fact, many asymmetric loss functions in machine learning have a simple *kink* in them, which makes them otherwise entirely differentiable. Figure 3.2 and Equation 3.7 explain such a function.



(a) Asymmetric loss shown around peaks and valleys.



(b) Asymmetric loss can be defined with LINEX functions.

Figure 3.1: Asymmetric loss functions define different losses to over predicting and under predicting a value.

$$LINEX(y_i, \hat{f}(x_i)) = e^{\phi(y_i - \hat{f}(x_i))} - \phi(y_i - \hat{f}(x_i)) - 1, \quad \phi \in \mathbb{R} \tag{3.6}$$

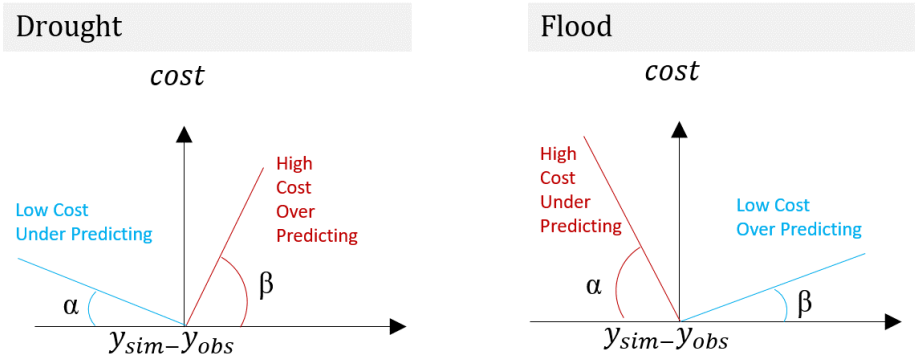


Figure 3.2: Asymmetric loss functions define different losses to over predicting and under predicting a value.

$$Hinge(y_i, \hat{f}(x_i)) = \alpha * min(0, \hat{f}(x_i) - y_i) + \beta * max(0, \hat{f}(x_i) - y_i) \tag{3.7}$$

The squared error loss penalizes larger errors more than smaller error (i.e., the function

is steeper in the tails than in the middle). To preserve this feature we can combine the concepts above and define a weighted $\ell 2 - norm$ (See Equation 3.8).

$$Weighted\,Squared\,Error(y_i, \hat{f}(x_i)) = \alpha * \left[min\left(0, (\hat{f}(x_i) - y_i)\right)\right]^2 + \beta * \left[max\left(0, (\hat{f}(x_i) - y_i)\right)\right]^2 \quad (3.8)$$

(2) **Should we be concerned with relative errors or absolute errors?** In hydrology, both manual and automatic attempts aimed at minimizing absolute errors often lead to fitting the higher portions of the hydrograph (i.e., peak flows) at the expense of the lower portions (i.e., baseflow) (Krause et al., 2005). Relative errors are generally more important than absolute errors. For example, a 10 TAF error in 1,800 TAF (monthly annual average of the Sacramento River) is less extreme of an error than in 15 TAF (monthly annual average of the Colorado River). Relative error loss functions or a simple log transformation of the data can help in this regard.

(3) **Should losses be a continuous function, or stepwise?** Although, most outcomes may follow a discontinuous step function (e.g., a neuron firing or not), many decisions in water resources (e.g., releases from a reservoir) are continuous. Continuity and differentiability make continuous math more convenient. One major development in neural networks was doing away with the concept of thresholds in the step function (representing the collective influence of all the inputs) and replacing it with a smoother *sigmoid* function. As with neural networks, many optimization algorithms require continuity and differentiability (e.g., gradient decent). However, advances in these methods now allow for piece-wise differentiability in the loss function.

(4) **Should losses be homogeneous or heterogeneous (i.e., weighted based on geographic region)?** The cost of incorrectly managing a densely populated urban basin may be very different than a desert or a headwater basin; the importance of having accurate flow estimates is not completely homogeneous especially across a big and diverse region as California. However, to avoid making those judgements, we will use a single loss function across all regions.

Legates and McCabe Jr (1999) suggests that a complete assessment of model performance

should include at least one *goodness-of-fit* or relative error measure (e.g., Modified NSE or Modified d defined in Equation D.13 & D.14, with $j = 1$) and at least one absolute error measure (e.g., RMSE or MAE defined in Equation D.3 & D.1) with additional supporting information (e.g., a comparison between the observed and simulated mean and standard deviations such as those defined in Equation D.6 & D.7) (Legates & McCabe Jr, 1999).

Therefore, along with the four characteristics discussed above, we propose to use only the following three selected measures-of-fit: the Modified NSE (as a relative error measure), the MSE and weighted MSE (as an absolute error measure), and the RSD (as an additional supporting measure).

3.3 Conclusion

This chapter will follow a risk minimization framework in developing a loss function. Contrary to other studies, we *are putting the horse before the cart*. That is, the loss function is developed before performing the learning, not just as an evaluation step after. The different performances of the models will be compared against the loss functions applied. Next, we will compare the shape of the predictions in the time-series compared to the observations. In squared error loss functions (i.e., MSE, NSE) the peaks get fitted at the expense of the low flows (i.e., high leverage points). However, the proposed wighted squared error asymmetric loss may be able to force a fit to both tails of the distribution. A comparison of the results from these losses will determine whether the aforementioned problem is mitigated with asymmetric losses. A dissertation chapter will include a comparison between the results of various loss functions applied. It will investigate the differences in the general shape of predictions obtained across the various loss functions, and discuss the effects of different weights in the asymmetric weighted MSE function.

Chapter 4

Rethinking Resampling Methods: Random Is Not Unbiased

> If two things are similar, the thought of one will tend to trigger the thought of the other.
>
> Aristotle, *"Laws of Association"*, 300 B.C.

Summary

After a statistical learning method is chosen and applied, the model needs to be tested. Most statistical learning techniques used in water resources modeling employ a randomized splitting of data into k-folds to estimate model error. In each iteration, one fold is held out as a test set and others are designated as a training set. Such random cross-validation methods ignore structures in the data, which underestimate model error (Roberts et al., 2017).

A more accurate estimate of the model error can come from techniques that block training sets in time, space, or unique structure (e.g., by hydrologic basins). The difficulty here lies in specifying block sizes in time and space. Blocking potentially reduces the range of parameters seen by the model, or may exclude a particular meaningful combination of predictor variables in the training data set. Too small of a block size and the cross-validation method more closely mimics the randomized method and runs the risk of under estimating model error. Large block sizes force too much model extrapolation and risk over estimating model errors.

This chapter compares resubstitution, Monte Carlo (i.e., randomized), leave one group

out (LOGO), and leave multiple groups out (LMGO) cross-validation strategies, as well as, resubstitution, Monte Carlo, blocked by group (BBG), blocked by multiple group (BBMG), and blocked by hierarchy (BBH) bootstrapping strategies for modeling synthetic unimpaired flows. This chapter intends to assess the sensitivity of the estimated uncertainty to the aforementioned resampling methods.

4.1 Introduction

Most, if not all, geographic data have internal correlation and dependence structures (Legendre, 1993): (1) temporal autocorrelation: nature responds to changes gradually. For example, today's precipitation is correlated with yesterday's precipitation; (2) spatial autocorrelation: nearby things tend to be more related than those far from one another. For example, two points close together on a topographic map are likely to have similar elevations; and (3) hierarchical structures: the network of streams flowing into one another (or more formally, the stream order) provides a hierarchical structure. That is, basin topology provides a spatial structure more complicated than merely proximity of river gauges. For example, two points on a river may be close in proximity but depending on which side of the watershed divide they fall on they can be fed by two different basins, in different hierarchies in the network, with different governing hydrologic processes, and therefore, have different measured flows (See Figure 4.1).

In predictive modeling, the goal is to accurately predict data, with structures mentioned above, at unmeasured locations either in the past in locations not gauged, or in the future where observations do not yet exist. Therefore, the predictive accuracy of the *training set*, the data the model is trained on, is of little consequence. The *test set* error, the error of a set of data not seen by the model, is the true measure of model accuracy.

Moreover, the predictive accuracy of the model on the training set can severely differ from the test set. With increasing model complexity (e.g., adding parameters to the model), clearly the model will fit the training data increasingly well. However, errors in the test set behave differently as is evident in the characteristic U-shape of the bias-variance tradeoff curve (J. Friedman et al., 2001). The expected error in the test set is a polynomial of power two (See Equation B.1) and is comprised of variance, bias, and a constant term. In the first

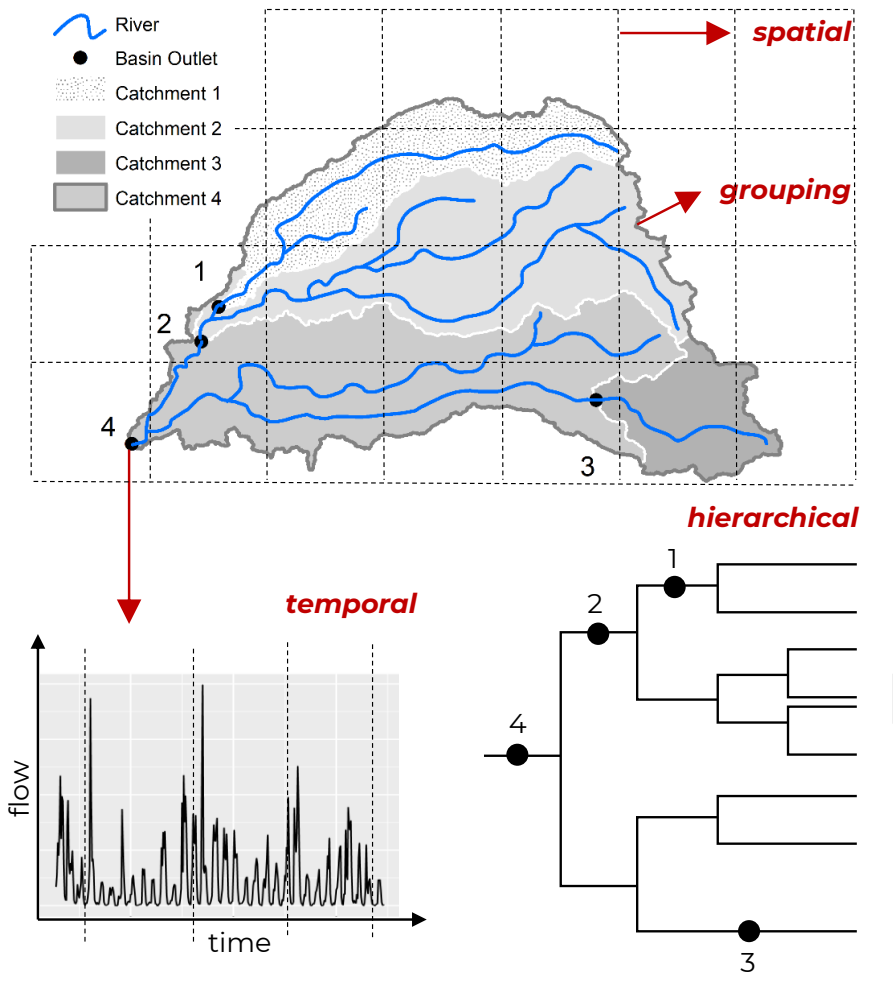


Figure 4.1: The four types of dependence structures in gauged data and blocking strategies

portion of the U, bias will decrease more than the gain in variance, however, past some point, we are now overfitting and the gain in variance is too much to be offset by the decrease in bias. Therefore, depending on how we specify the model, we will lie somewhere along this U shape and cannot substitute training error rates for the true predictive capability of the model.

The test set error can be easily calculated if such a data set exists, or, it can be estimated by holding out a subset of the training data. The holding-out is achieved by resampling strategies, to effectively creating a test set. Two resampling methods are: *cross-validation* and *bootstrapping*. In cross-validation, the data set is split into testing and training data sets where each observational unit gets a chance at being in the test set once. In bootstrapping, sampling is done with replacement where each observational unit gets an equal chance at

being selected and being selected more than once. In this case, on average $1/3^{\text{rd}}$ of the data set will end up not being selected at all, in other words these observations are out-of-bag (Efron & Tibshirani, 1997).

With the test set, that is held out observed data, and our model's results, we can conveniently apply any statistical measure of fit desired as a proxy for model accuracy (e.g., Nash-Sutcliffe Efficiency (NSE)).

Most studies, in water resources, ignore dependence structures in the data when devising a resampling strategy. When testing data are randomly selected from the entire spatial domain, training and testing data from nearby locations will be dependent due to *spatial autocorrelation*. Therefore, if the objective is to project outside the spatial structure of the training data (e.g., to an ungauged basin), error estimates from random cross-validations or the bootstrap statistic, will be overly optimistic (Roberts et al., 2017).

In essence, a correlation structure points to a pseudo replication problem (See Figure 4.2). For x^d at a distance, Δd, from $x^{d+\Delta d}$, where x^d and $x^{d+\Delta d}$ are autocorrelated. The distance Δd can be defined in time, space, or hierarchy. In random resampling, either of the autocorrelated values may lie in the bag of samples given to the model, or it may be left out-of-bag. Therefore, the model can easily predict one, given that the other is likely in the bag. However, in blocking resampling the two observations are connected and will both end up in the bag or out-of-bag. Here, the model is forced to predict a phenomenon from other observations.

While correlation structures may not be as problematic in conventional statistics, combined with high-dimensions and low sample sizes, predictive methods suffer. Compounding the problem can be low computational abilities (See Figure 4.3).

4.2 Research Design

Resampling methods are generally used to either: (1) estimate the accuracy of sample statistics (e.g., the standard deviation of an α parameter in a linear model $y = \alpha x + \beta$); (2) estimate the accuracy of significance tests (e.g., p-values); or (3) test the models. In hydrology, when examining observed data, the true population (i.e., the set of all possible hydrologic states) is unknown and the true error in the model or a sample statistic is un-

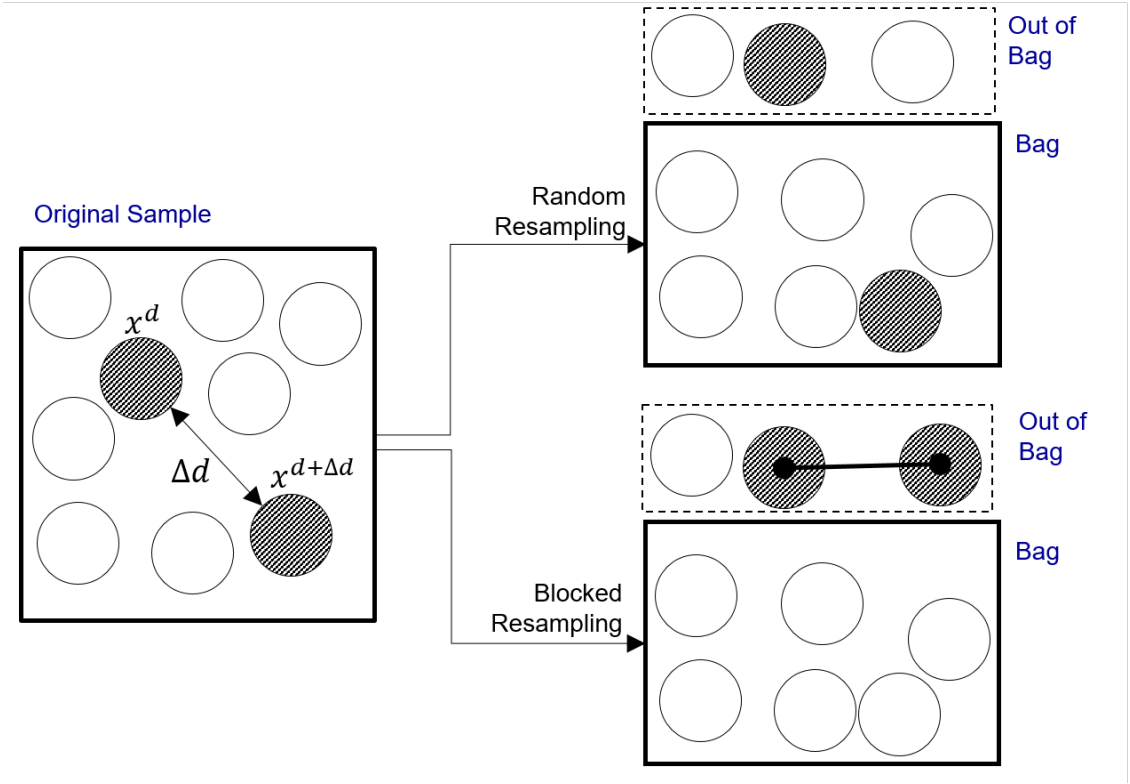


Figure 4.2: Autocorrelation is a pseudo replication problem. The two grey marbles are autocorrelated. A model that uses random resampling will be able to easily predict one grey marble since it has seen the other. When blocking, the observations move in and out of the bag together.

knowable. Therefore, we commonly use resampling methods to test our model's predictions.

In this study, data from a mechanistic simulation model, the Basin Characterization Model (BCM), with fitted values are considered as "true" unimpaired flows. The BCM approximates California hydrology well. It estimates monthly unimpaired flows and is developed and maintained by the U.S. Geological Survey (USGS). The data spans California at 270m x 270m resolution. The recharge and runoff estimates from the BCM are attained from physically based equations that calculate potential evapotranspiration, snow, excess water, and actual evapotranspiration. Depending on soil properties and the permeability of underlying bedrock, surface water can be classified for each cell as either recharge or runoff (Flint & Flint, 2014).

The recharge and runoff rasters can be aggregated to any given basin. Here, the machine learning model will be trained on the simulated runoff values from the BCM aggregated to the CDEC basins (See Figure A.1). The developed data set has approximately 18,500

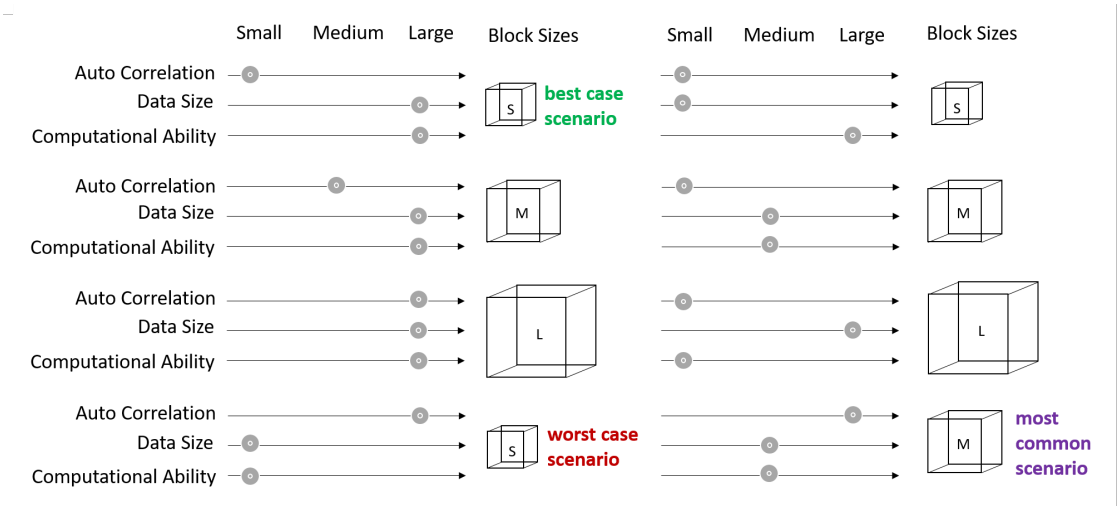


Figure 4.3: The block size in resampling methods is a function of the autocorrelation, data size, and computational ability.

monthly unimpaired flow observations in acre-feet (AF) (See appendix **??** for more info). The data spans from 1895 to 2018 at a monthly time step. As mentioned in Chapter **??**, we will develop a GLM, RNN, and TMARS model. That said, the focus of this study is to demonstrate the differences between the cross-validation methods, not on the data or the machine learning method themselves; the purpose is to see which resampling strategy used in the machine learning algorithm gives the closest estimate to the true model error. That is, we want to see which cross-validation or bootstrap method is most appropriate for the machine learning model predicting values of a known model. In this chapter, we considered MSE as the loss function and the model measure of fit (See Equation D.2).

To find the MSE of the machine learning technique: (1) simulate n landscapes of the data by resampling the original data set using the bootstrap method (resampling with substitution); (2) separate the data into training and testing sets (use the CV or BS methods discussed below); (3) for each simulation feed the training data into the desired machine learning algorithm (i.e., a GLM, RNN and TMARS); (4) calculate the desired model measure of fit for each of the simulations; and (5) compare the Probability Density Function's (PDFs) of the model measures of fit with an "ideal" one (See Figure 4.4 &4.5).

In both the cross-validation and bootstrap, the resampling results are compared to an "ideal" MSE, which was calculated by: (1) producing one model for each simulated landscape; (2) using said model to predict to the other $n-1$ landscapes; (3) using the predicted

and observed values to calculate the MSE for each $n-1$ landscape; (4) averaging the MSE of the $n-1$ landscapes; (5) repeating the process for all n landscapes; and (6) resulting in n MSEs, one for each landscape, which can be plotted as a PDF (See Figure 4.6).

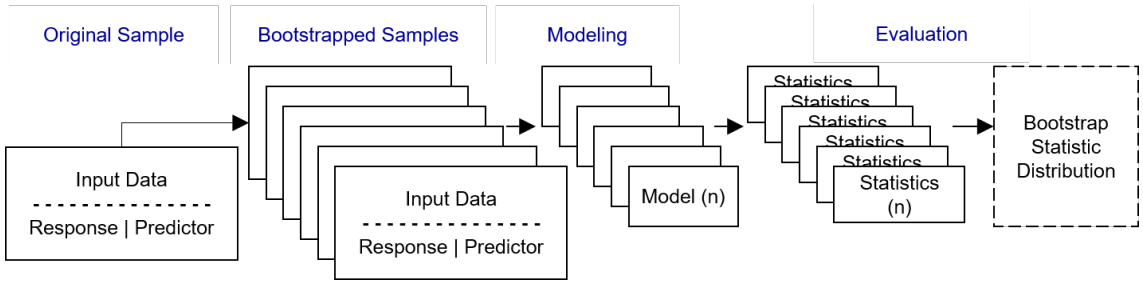


Figure 4.4: Research design: We employ the bootstrap method to find the distribution of the bootstrap statistic.

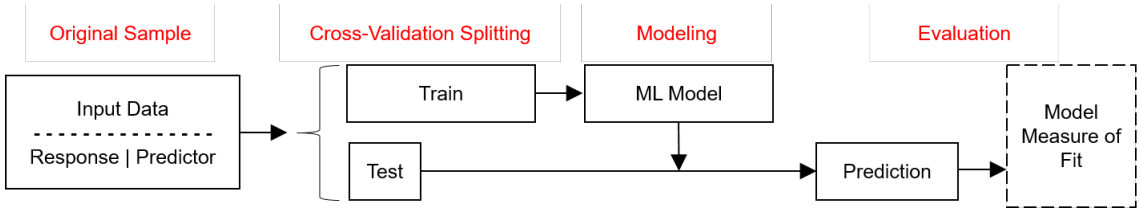


Figure 4.5: Research design: We employ the cross-validation method to find the model error estimate.

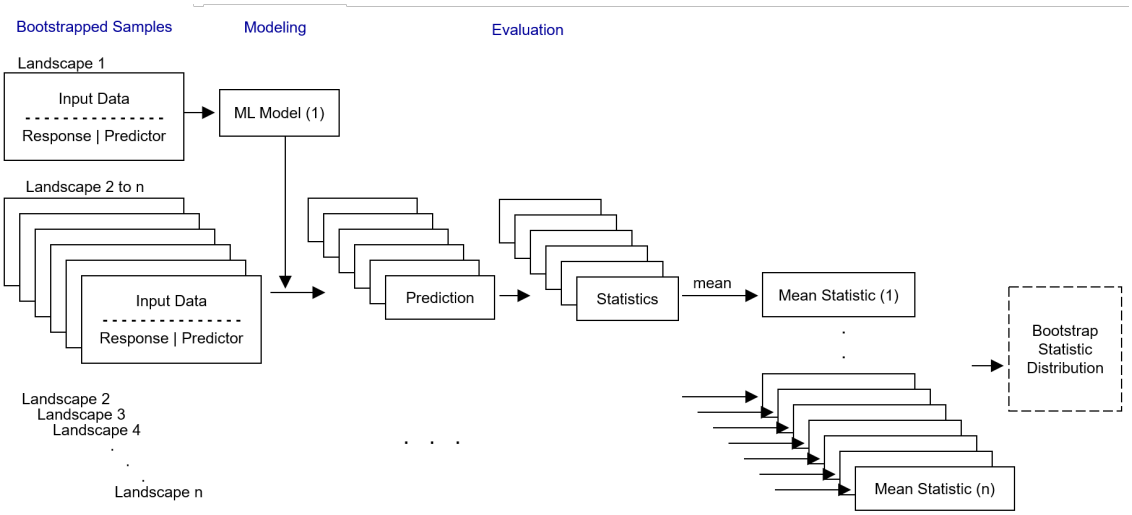


Figure 4.6: Research design: compare the errors with that of an "ideal" model.

Resampling Methods

The second step in the methods mentioned above, separating the data in to training and testing sets, can be accomplished by one of the following methods:

Cross-Validation

- Resubstitution: the test set is the training set. Here, the model is evaluated against the same data it has already seen. We expect the model to perform the best here and the PDF of the residuals to be closest to 0.

- Randomized or Monte-Carlo:cross-validation

  Validation Set: the test set is a random 1/2 of the full data set. the training set is the other 1/2. This method is run twice, once with the first half as a test set and next with the second half being the test set.

  5-Fold: the data is split into 5 folds. In each iteration each fold is considered the test set and the other 4 folds the training set.

  10-Fold: the data is split into 10 folds. In each iteration each fold is considered the test set and the other 9 folds the training set.

- Leave One Out (LOO): in each iteration, one instance of the data is help out, and the rest of the data set is the training set. most intense computationally.

- Leave One Group Out (LOGO): in each iteration, one basin's data is held out as a whole and the rest of the basins become the training set. The process is repeated for each basin.

- Leave Multiple Groups Out (LMGO): in each iteration, $1/5^{\text{th}}$ or $1/10^{\text{th}}$ of the basin's are held out and the rest of the basins become the training set. The process is repeated for each fold.

- Leave Hierarchies Out (LHO): blocking is design across similar stream orders.

Bootstrap Methods

- Resubstitution: same as above, the test set is the training set.

- Randomized or Monte-Carlo: the most popular form of bootstrapping where a new data set is built from randomly resampling the original sample with substitution. The length of the data set is the original length of the data set.

- Blocked By Group (BBG): the data set is blocked by unique basins. The basins are randomly resampled with substitution. Since the basins may have differing record lengths, the length of the data set may not match the original data set. However, the data set will have the same number of basins as in the original data set.

- Blocked By Multiple Groups (BBMG): the data set is blocked by multiple basins. The grouped basins are randomly resampled with substitution. As the group sizes become larger the blocking size becomes larger.

- Blocked By Hierarchy (BBH): the data set is blocked by stream order. The grouped basins are randomly resampled with substitution. Some stream orders have a chance of occurring in the data set twice where some are left out.

4.3 Conclusion

This chapter presents various blocking resampling techniques where the observations in a block are bonded together. The idea behind blocked resampling is simple: *birds of a feather flock together*, or more accurately birds of a feather *should* flock together. That is, if two observations are autocorrelated they should be both included in the bag, or training set, or both be out-of-bag, or in the test set.

These blocking methods should show how much random methods underestimate the model error. That is, models evaluated with random methods may actually perform worse than we expect due to the pseudo-replication problem that autocorrelation presents. This isn't to say that, in hydrology, random resampling is never useful; the studies, in which a random test-train split is considered, are most appropriate for predicting flow for a sparsely incomplete gauge record, and the studies, in which holding out blocks of data in time is considered the resampling strategy, are most appropriate for predicting streamflow in time for that location. One should not expect to use these resampling strategies and get the same predictive accuracy in a purely ungauged basin problem, where blocks are supposed to be designed across geographic space (or hierarchical structure).

This chapter proposes applying multiple resampling strategies to the ungauged basins problem and comparing the resultant PDFs to that of an ideal model. We can then visualize

how far errors predicted with random resampling strategies are from the ideal scenario.

Chapter 5

Climate Change: Including Non-Stationarity into Statistical Analysis

> From where we stand the rain seems random. If we could stand somewhere else, we would see the order in it.
>
> Tony Hillerman, *"Coyote Waits"*, 2009

Summary

5.1 Introcuction

Appendix A

Model Data

This section introduces the data used in the statistical learning model.

Study Area

This study used the monthly unimpaired flows dataset developed and maintained by the California Data Exchange Center (CDEC). The data spans 67 California basins (See Figures A.1 and A.2) from 1982 to 2014. It can be downloaded with a simple webscraping script available on GitHub. It has approximately 19,000 monthly streamflow observations in acre-feet (AF) and as a continuous variable can be used for regression type studies (See Figure A.3).

Predictor Variables

Predictor attributes were calculated for each observation point (See Table A.1). A total of 24 predictor variables were selected based on the knowledge of basin characteristics and processes that influence a watershed's response to precipitation: evaporation (temperature); snowfall (cumulative sum of precipitation below 2°C); storage in soil (with soil and land cover parameters); antecedent conditions (with lagged precipitation and temperature parameters); and groundwater processes (with depth to restricted layer).

The climate data were derived from the Parameter elevation Regression on Independent Slopes Model (PRISM) dataset, which contains gridded rasters for the continental United States at $4km$ resolution from 1891 to 2014. The **temperature** variable and its lagged forms are the basin averaged PRISM *tmean* variable, which in turn was calculated by the

Figure A.1: The 67 California basins under study are the CDEC unimpaired flow basins.

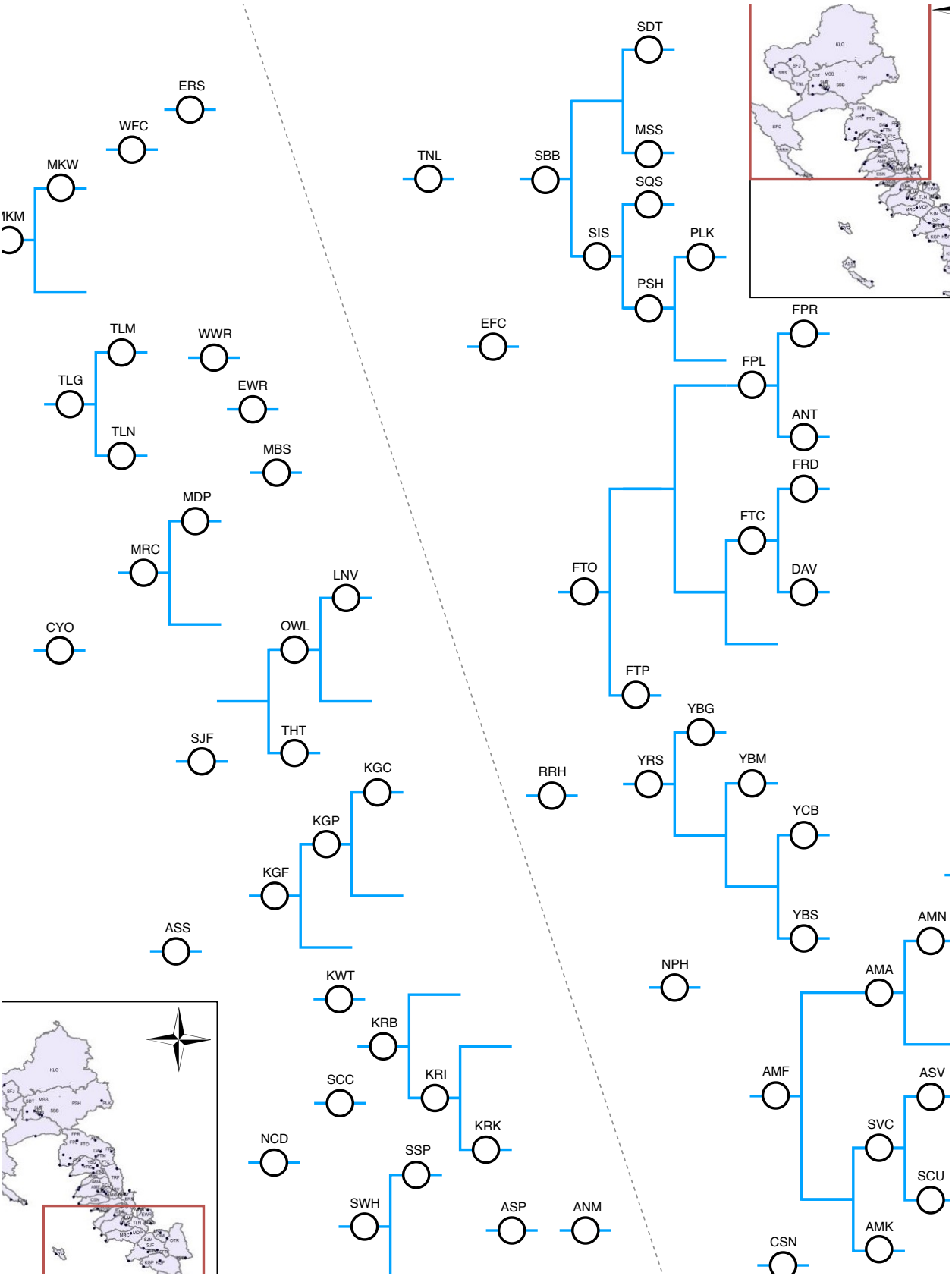
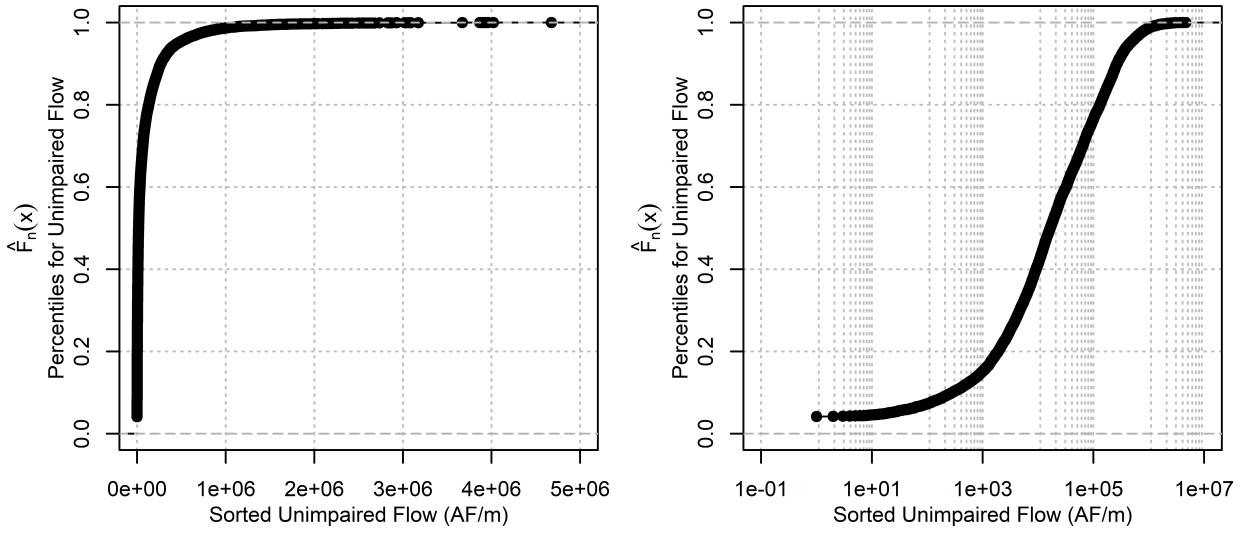


Figure A.2: Network schematic.



(a) The cumulative distribution function.

(b) The cumulative distribution function in log space.

Figure A.3: Distributions of the response variable. Approximately 19,000 unimpaired flows in acre-feet/month.

mean of the monthly minimum temperatures and the monthly maximum temperatures. The **precipitation** variable and its lagged forms are the basin averaged PRISM *ppt* variable, which is a measure of total precipitation (rain and snow).

Low flows in some Sierra Nevada basins exhibit a "memory" effect in which they depend on the current and previous year's snowpack (Godsey, Kirchner, & Tague, 2014). Since we did not want to include 24 lagged precipitation parameters in the model, we developed a snow variable. The **snow** variable was the cumulative sum of precipitation, starting in October of each water year, for temperatures under 2°C.

Basin shape can affect the peak discharge; peak discharge for a circular basin arrives sooner than for an elongated basin of the same area. Because of how the tributary network in a circular basin is organized, the flows in a circular basin enter the main stem at roughly the same time, so more runoff is delivered to the outlet together, sooner. In an elongated basin, because of the mismatch in timing, peak runoff is more attenuated, except for some slow moving streams. The **shape** parameter, calculated by basin length divided by basin width, and the **compactness** parameter, calculated by basin area divided by (basin perimeter)^2, account for this phenomenon. Although, this phenomenon is more pronounced in runoff on

a smaller time step, we included these parameters in the final model to see their importance.

Basin hypsometric information was derived from the Shuttle Radar Topography Mission (SRTM) $90m$ model, which is a gridded raster of static elevation at a $3arc-second$ resolution. The vertical error of the model is reported to be less than $16m$. The **mean basin elevation** and **basin relief ratio** parameters (Pike & Wilson, 1971) were calculated from this dataset. Basin relief ratio is calculated by the difference in maximum and minimum elevations divided by basin length.

Soil properties were derived from the POLARIS dataset, a Soil Survey Geographic Database (SSURGO) processed dataset at a $3arc - second$ resolution. Percent **clay**, **silt**, and **sand**, **saturated hydraulic conductivity**, **lambda** and **n** pore size, **available water content**, and **depth to restricted layer** information was averaged for each basin.

Table A.1: Summary of the variables used in the implementation of the model.

Type	Variable	Description	Source
Response	Unimpaired Flow	monthly estimated unimpaired flows, in AF	CDEC (Beaudette, 2016)
Time	Ordinal Month	numerical distance till October	
	Water Year	numeric year starting from the October of previous Gregorian year	
Climate	Temperature, Lag 1, 2 and 3 Months	temperature and lagged monthly temperature, in °C	PRISM (Hart & Bell, 2015)
	Precipitation, Lag 1, 2 and 3 Months	precipitation and lagged monthly precipitation, in mm	
	Snow	cumulative precipitation of the same water year for temperatures bellow 2 °C, in mm	
Hypsometric	Relief Ratio	(max(elev) - min(elev))/ basin length in, m/m	SRTM90 (Jarvis, Reuter, Nelson, Guevara, et al., 2008)
	Mean Elevation	mean basin elevation, in m	
Basin Boundaries	Area	basin drainage area, in m^2	NHD2PLUS (McKay et al., 2012)
	Shape	basin length/basin width, in m/m	
	Compactness	basin area/(basin perimeter)^2, in m^2/m^2	
Soil	% Clay	percent clay in surface layer, in %	POLARIS (Chaney et al., 2016)
	% Silt	percent silt in surface layer, in %	

Type	Variable	Description	Source
	% Sand	percent sand in surface layer, in %	
	Sat. Hydraulic Conductivity	hydrologic conductivity of surface layer, in cm/hr	
	Lambda	pore size distribution index (brooks-corey)	
	N	measure of the pore size distribution (van genuchten)	
	Available water content	available water content, in m^3/m^3	
Land Cover	Vegetated	Percent of area in the basin vegetated in %	CALVEG (Forest Service, USDA, Pacific Southwest Region, 2006)
Ground Water	Depth to Restricted Layer	depth to aquitard, in cm	POLARIS (Chaney et al., 2016)

Other Descriptive Variables

Some variables are included in the dataset, but not in the modeling; these variables define the location of the guages, and consist of the following: Longitude and Latitude (definite location), Hierarchy or the number of guages that exist above (relative location in the network), river basin, county, and guage operator. These variables are only used for plotting purposes (See Figure A.4, A.5, and A.6).

Correlations

A simple examination of the partial correlations of predictor variables with flow shows that most of the information content lies within drainage area, precipitation, and some measures of infiltration (i.e., lambda pore size, n pore size, and saturated hydraulic conductivity). The correlated variables were not removed from the model (See FigureA.7). For a more complete correlation plot see Figure A.8.

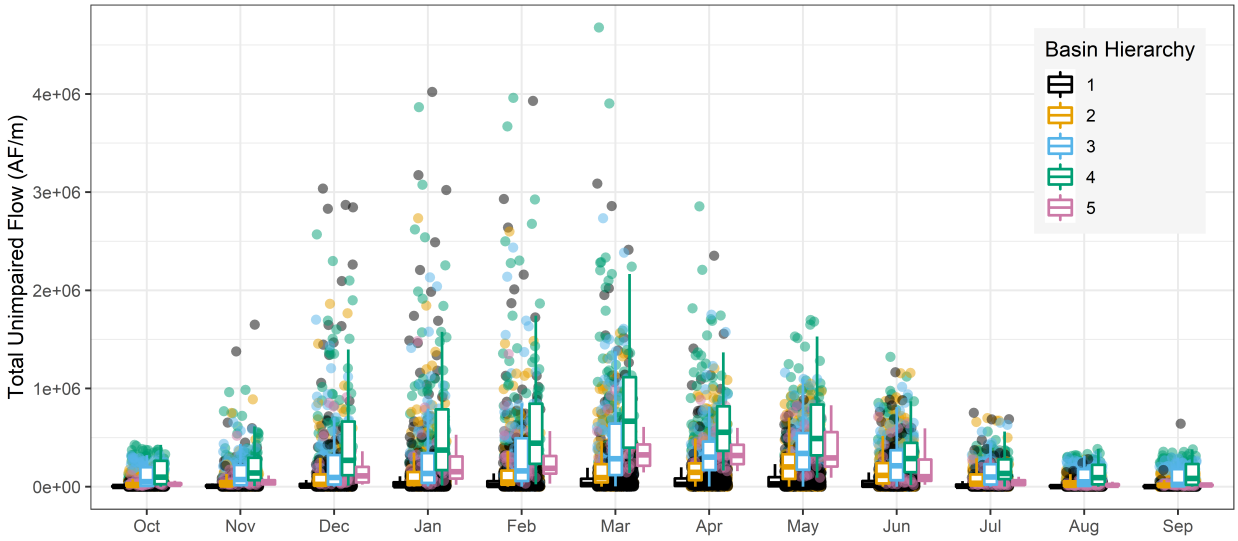


Figure A.4: The cyclical behavoir of total monthly unimpaired flows. The flows start to rise in October, the start of the "water year". The boxplots also show that given a higher hierarchy (i.e., being lower in the network of guages) the monthly distribution of flows becomes larger. The only expection to this is basin hierarchy number 5, and that is due to the fact that this data set only had one basin in that hierarchy. Had there been more basins, its distribution would be wider showing that the lower you are in the network, the higher the flows and the bigger the distribution of flows.

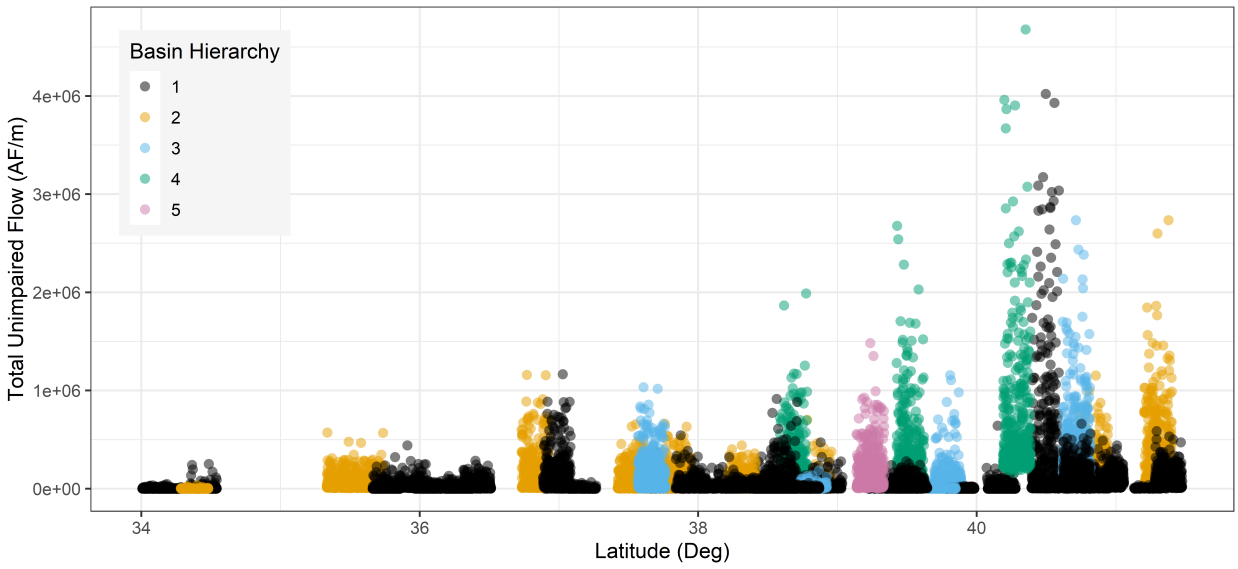


Figure A.5: Total monthly unimpaired flow vs. latitude. Total monthly unimpaierd flow increases at higher latitudes in California. Note that each of the basins were at a unique latitude, for illustration purposes the latitude variable was jittered.

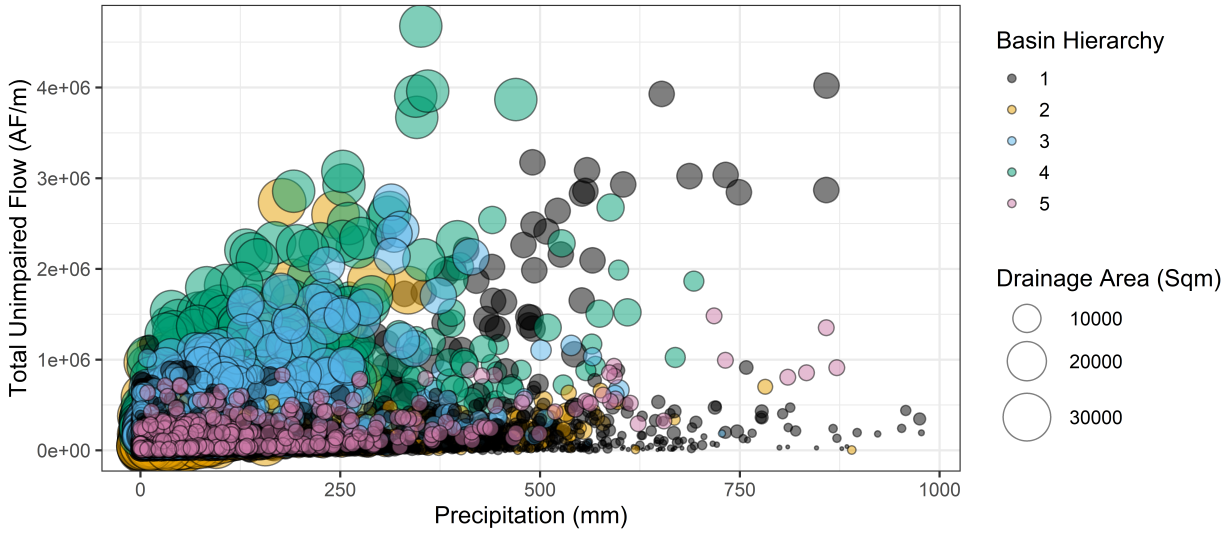
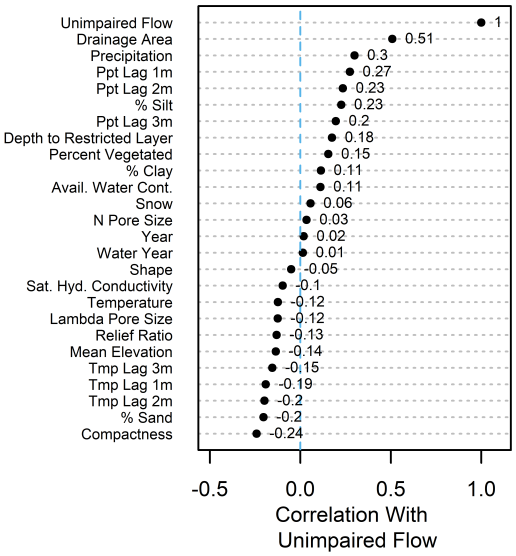
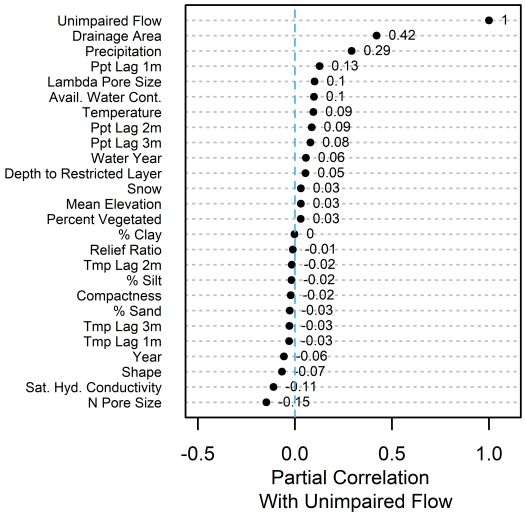


Figure A.6: Ttoal monthly unimpaired flow vs. precipitation. The total monthly unimpaired flow increases with increasing precipitation. This is also drainage area dependent, as the smaller drainage areas that happen to have high amounts of precipitation still produce low flows. Basin hierarchies also show that the larger basins are lower in the network. The only exepction being hierarchy number 5, and that is due to the fact that this data set only had one basin in that hierarchy.



(a) Pearson's Correlation



(b) Partial Correlations

Figure A.7: Correlation of predictor variables with monthly flow volumes. Drainage area and precipitation correlate the most with flow.

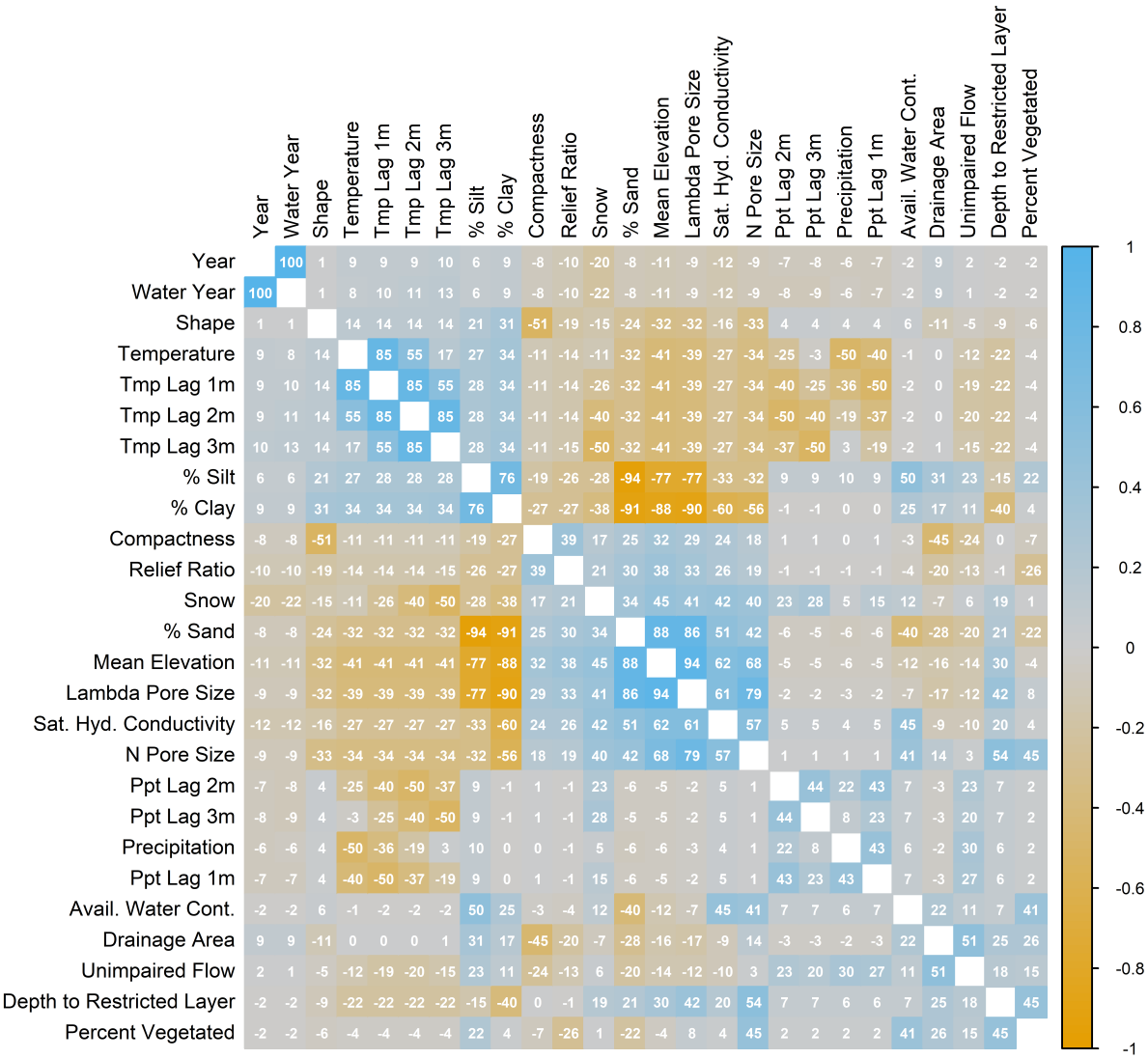


Figure A.8: Correlation plot. Patterns can arise in correlations especially when some variables are calculated from or are directly related to others. For example, the percentage of sand silt and clay in a basin adds to one. Therefore, these variables are negatively correlated. Also, lag variables calculated from precipitation and temperature will tend to correlate with one another. However, the snow variable that was calculated from precipitation does not significantly correlate with precipitation.

Appendix B

Terms & Concepts in Machine Learning

This section introduces common terms and concepts used in statistical learning and in this paper.

Terminology

Variables: Predictors, independent variables (sometimes just variables), or features all are the inputs into a model that we believe in some way will inform us about another variable we are interested in. The response, output, or dependent variable, is the output of the model we are interested in.

Training and Test sets: Data sets used for training the model and testing the model's predictive capabilities respectively.

Bias Variance Trade-off: Bias and variance make up part of the expected test set squared error (See Equation B.1).

$$\begin{aligned} &E[(y-\hat{f})^2] = Var[\hat{f}] + (Bias[\hat{f}])^2 + Var(\epsilon) \\ &Var[\hat{f}] = E[\hat{f}^2] - (E[\hat{f}])^2 \\ &Bias[\hat{f}] = E[\hat{f} - E[y]] \\ &Var[\epsilon] = \sigma^2 \end{aligned} \tag{B.1}$$

where y is the observed response variable, x is the observed predictor variable and $y = f(x) + \epsilon$, $\hat{f}(x)$ is the modeled or predicted response variable, and ϵ is the irreducible error in

the response variable.

That is, variance and bias make up the reducible error in the response variable. It is reducible because we can modify it by changing the training data (e.g., adding more data), which effects the variance component, or changing the model type (e.g., going from linear to nonlinear), which effects the bias component of the bias variance trade-off.

Resampling: These methods create "extra" data from the same data set. This data set, different from the whole sample, is sometimes needed for nuisance parameter estimation (usually achieved with cross-validation) or model error estimation (usually achieved with the bootstrap). We will discuss the importance of resampling methods in Chapter **??**.

Loss or Objective Function: The expectation of the loss function, $L(y_i, \hat{y}_i)$ is the function that is minimized (or maximized) in a statistical learning algorithm. Figure **??** depicts typical loss functions used in machine learning. In essence, a loss function is a statement of priorities; what we want the model to get right and what are we willing to trade for it. For example, what is the true cost of getting low flows predicted incorrectly (drought damage cost)? What is the cost for predicting high flows incorrectly (flood damage cost)? Therefore, to some extent the choice of a loss function requires informal subjective decisions. We will examine some loss function in Chapter **??**.

Convex Optimization Problems: Optimization problems that are convex in the objective function and constraints have a special property; if a solution is found to the minimization or maximization it is guaranteed to be a global solution.

Gradient-Based Optimization Methods: These methods find the local minima or maxima of an objective function by searching along the gradient of the objective function. For example, in a minimization problem using the steepest gradient search methods, the decent direction and step size is found in one iteration. Gradient-based methods require the loss function to be differentiable. However, variations such as subgradient methods have been developed that allow for the minimization of convex problems given kinks in the loss function.

Derivative-Free Optimization Methods: These methods do not require gradient calculations and are well suited to problems where a loss function is not explicit. For example, evolutionary algorithms find local minima or maxima by evaluating the loss function on a

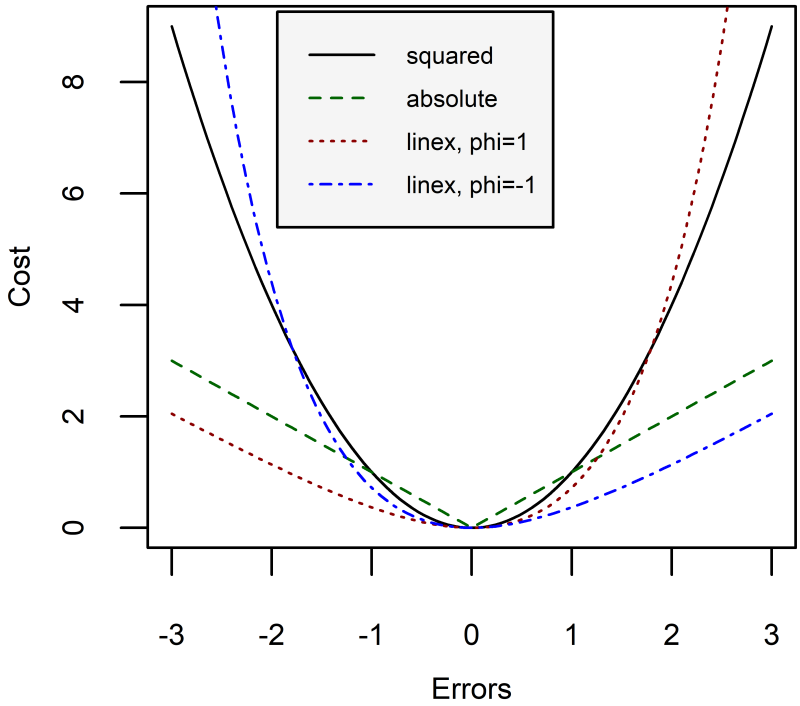


Figure B.1: Typical loss functions in statistical learning.

population of solutions, and letting them evolve in each iteration.

Learning Scenarios

Supervised vs. Unsupervised: In supervised settings, we have a variable of interest, y, that we believe follows a functional form: $y = f(x) + \epsilon$, where $f(x)$ provides systematic information about y, and ϵ is the error term. In modeling we try to approximate this functional form (i.e., $\hat{f}(x)$) with the observations (i.e., $\hat{y}$). We also can try to estimate y from the data itself, without assuming a functional form (See next section on Parametric vs. Non-Parametric).

In unsupervised learning, we do not have a variable of interest, y, to model. Instead, we have observations of many variables that we can still study for their natural groupings, patterns, or relationships between variables.

Prediction vs. Inference: The two major goals of statistical analysis is either prediction or inference. In prediction, we are interested in getting the simulated value to closely resemble the observed values (e.g., can we accurately predict the value of a house). That is,

we are concerned with accuracy.

However, in inference, we are interested in the relationship of the predictor variables to the response variable (e.g., how much extra will a house be worth given a scenic view). That is, we are concerned with model interpretability, which implies that a simpler (i.e., fewer variables, less flexible) model is preferred even at a little cost to prediction accuracy (James et al., 2013).

Parametric vs. Non-Parametric: Parametric models assume a functional form. For example, from Ohm's law ($V = IR$), we can safely assume that given an unknown resistor, voltage and current have a linear relationship ($y = \beta_1 x + \epsilon$), where y is the voltmeter readings and x is the ammeter readings. By assuming this functional form errors in observations can be due to the measurement device (the voltmeter or ammeter) or human error. Now, we can estimate the parameters of the model from the observations. In this case, we are estimating R, resistance, from fitting $\hat{y} = \hat{\beta}_1 x$. We have effectively reduced the problem of finding $\hat{f}(x)$ to finding $\hat{\beta}_1$.

However, in non-parametric models, we do not assume a functional form and try to get the model as close to the data points as possible without being too "rough". For example, Kriging interpolators are known to be exact interpolators where the predictions at each observation point goes through the exact observation. Therefore, this approach is highly dependent on the observation and suffers from high variance in the bias-variance trade-off. Smoothing techniques, such as thin plate splines, relax this constraint, and depending on the degrees of freedom or flexibility we allow, the prediction can get close to or far from the observations. This approach is data intense and usually performs better where prediction, rather than inference, is concerned, because, after all, it is more or less honoring the data.

Regression vs. Classification: Variables can be classified as quantitative or qualitative. Quantitative variables take on numerical values and a quantitative response variable is used in what we refer to as regression models. In contrast, qualitative variables take on classes, categories or levels and a categorical response variable is used in classification models. The predictors may take either form and are generally less important (James et al., 2013).

Appendix C

Brief History of Statistical Learning

This section explains how some of the ideas organized in Chapter **??**'s heuristic guide developed over time.

In 1763, Thomas Bayes's *An Essay towards solving a Problem in the Doctrine of Chances* is published posthumously. In it, Bayes explained that "given the number of times in which an unknown event has happened and failed, the chance that the probability of its happening in a single trial lies somewhere between any two degrees of probability that can be named" (Bayes & Price, 1763). This work later underpins **Bayes Theorem**.

In 1805, Adrien Marie Legendre introduced the least squares method of estimating parameters as an appendix to his book on the paths of comets. Carl Freidrich Gauss also publishes the method a few years later but claimed he had been using it since 1795 (Stigler, 1981). Regardless of the original inventor, the method is brought to perfection with its application to **linear regression** and curve fitting.

In 1812, Pierre-Simon Laplace, expanding on the work of Bayes, introduced methods of finding probabilities of compound events when the probabilities of their simple components are known, and he defined what is now known as **Bayes' Theorem** (O'Connor & Robertson, 2000).

In 1913, Andrey Markov founded a new branch of probability theory by applying mathematics to poetry. Later called **Markov chains**, the method went beyond coin-flipping (where each event is independent of all others) to chains of linked events (where what happens next depends on the current state of the system) (Hayes et al., 2013).

In 1936, Ronald Fisher introduced a method to find a linear combination of features that separates (or discriminates between) two or more classes of events. Fisher's discriminant is later slightly modified to add the assumptions of normally distributed classes or equal class covariances, and became the more famous **linear discriminant analysis (LDA)** (Härdle & Simar, 2007).

In the 1958, David Cox developed **logistic regression** for situations where it is not reasonable to assume that the independent variables are normally distributed as in LDA.

In 1951, Marvin Minsky and graduate student Dean Edmonds built the first **neural network machine**. This machine was a randomly connected network of capacitors that have a finite amount of memory and time to keep or remember that memory. The memory holds the probability that a signal will come in one input and another signal will come out of the output. This machine, modeled after the Hebbian theory of learning in the human brain, was one of the first pioneering attempts at artificial intelligence. Shortly after, in 1957, Frank Rosenblatt invents the perceptron, the first **neural network** for computers.

In 1967, the Thomas Cover and Peter Hart invent the **nearest neighbor** algorithm, which kickstarted basic pattern recognition (Cover & Hart, 1967). The algorithm was used to map a route for the *traveling salesmen problem*, starting at a random city but ensuring a visit to all cities during the shortest tour (Marr, 2016).

In 1972, Nelder and Wedderburn introduced **generalized linear models**. Linear models are customarily made of systematic and random error components, with the errors usually assumed to have normal distribution. This work allowed for a unified fitting procedure, despite the type of error distribution, based on likelihood (Nelder & Wedderburn, 1972).

In 1980, Kunihiko Fukushima developed the neocognitron, a type of **artificial neural network** (Fukushima & Miyake, 1982). This work later inspired the development of **convolutional neural networks**.

In 1981, Gerald Dejong introduced **explanation based learning**, where a computer algorithm analyzes data, creates a general rule it can follow, and discards unimportant data (Marr, 2016). The new knowledge structure is not constructed by noticing the similarities and differences among a large number of inputs, nor is it constructed from a more general one already existing within the system. The system is capable of learning from just one example.

The knowledge structure can be expanded later but is already a viable new schema capable of adding future processing (DeJong, 1981).

In 1982, John Hopfield developed Hopfield networks, a type of **recurrent neural network** that can serve as content-addressable memory systems (Hopfield, 1982). Based on aspects of neurobiology, the content-addressable memory can yield an entire memory from any subpart of sufficient size. The recurrent aspect of RNNs make it a breakthrough for processes that are driven by lagged parameters. For example, in hydrology, runoff processes are effected by time-lagged precipitation; depending on the size of the watershed, precipitation at the headwaters may take days to get to the outlet, or, snowfall in the winter will take months to melt and turn into baseflow. In 1997, Sepp Hochreiter and Jorgen Schmidhuber invent **long short-term memory (LSTM) recurrent neural networks**. This method greatly improved the efficiency of neural networks (i.e., more successful runs, at a higher learning rate) and it solved complex (i.e., artificial long-time-lag) tasks that have never been solved by previous recurrent network algorithms (Hochreiter & Schmidhuber, 1997).

In 1984, Brieman, Friedman, Olshen, and Stone introduced **classification and regression trees (CART)** (Breiman, Friedman, Olshen, & Stone, 1993), a method of recursively partitioning the feature space. In 1995, Tin Kam Ho fixes the issue of high variance in the CART with his proposed **random forest** algorithm (Ho, 1995).

In 1986, Hastie and Tibshirani developed the **generalized additive model**, a non-parametric extension to the generalized linear models where the linear predictor is replaced by an additive predictor (Hastie & Tibshirani, 1990). This means the model is fit on multiple predictors and the fit on each predictor is updated by holding the others fixed (i.e., fit to a partial residual).

In 1995, Corinna Cortes and Vladimir Vapnik published their work on **support vector machines**. Originally applied to only two-group classification problems, this procedure constructs a linear decision surface in high dimensions with corresponding "support vectors" at a margin, M, from the decision surface. The purpose of the method is to maximize the margin, M (Cortes & Vapnik, 1995).

Until the 1990's, statistical learning was a purely theoretical analysis of the problem of function estimation from a given collection of data (Vapnik, 1999). Since then, with

the commercialization of software programs, these methods can be applied to "real-world" data and therefore used in fields outside of statistics and computer science. Work on these methods has also shifted from knowledge-driven approaches to a data-driven approaches; we are letting the computer analyze large amounts of data and "learn" from the results. As Winston (2010) puts it, "the computer is learning much like a bulldozer processing gravel (Winston, 2010)."

In 2006, Geoffrey Hinton developed *deep learning* techniques that let computers "see" and distinguish text in images (using the famous MNIST database of hand-written digits). These methods make inference easier in densely connected belief nets that have many hidden layers and scale poorly to increases in the number of parameters (Hinton, Osindero, & Teh, 2006). **Deep convolutional networks** have brought about breakthroughs in processing images, video, speech and audio (Marr, 2016).

In 2010, the Microsoft **Kinect** was launched. The devise could track 20 human features at a rate of 30 times per second (Marr, 2016), allowing people to interact with the computer (or more pointedly, the console) via movements and gestures. Microsoft's vision was to incorporate motion into gaming, eliminating the need for controllers you would have to charge or could accidentally fling into your TV (Cranz, 2018).

In 2012, **Google Brain** started. Led by Andrew Ng and Jeff Dean, its deep neural network can learn to discover and categorize objects. Despite the fact that the network had never been told what a cat was, nor was it given even a single image labeled as a cat, it "discovered" what a cat looked like from unlabeled YouTube images (Dean & Ng, 2015).

In 2014, Facebook developed **DeepFace**, a software algorithm that is able to recognize that two images show the same face (i.e., facial verification). It employs a nine-layer neural net with over 120 million connection weights, and was trained on four million images uploaded by Facebook users (Simonite, 2014). This algorithm raised some privacy concerns and their recent Cambridge Analytica scandal didn't help Facebook with the heightened scrutiny either.

In 2014, Google researchers presented their work on **Sibyl**. This proprietary platform started off by recommending YouTube videos to users. Now it can predict spam and a user's ad preferences. In general, its goal is to predict how Google users will behave in the future,

based on what they did in the past (Woodie, 2014).

In 2015, Amazon launched its own machine learning platform, **SageMaker**. This platform was designed to help developers and data scientists from the data acquisition step to full model deployment (Amazon Web Services, 2018).

In 2015, Microsoft created the **Distributed Machine Learning Toolkit**, which makes machine learning tasks on big data highly scalable, efficient, and flexible. The toolkit employs a special sampling techniques to create and distribute training data throughout the cluster (Rolle, 2015).

In 2015, over 3,000 AI and Robotics researchers, endorsed by Stephen Hawking, Elon Musk, and Steve Wozniak (among many others), signed an open letter calling for a ban on offensive autonomous weapons beyond meaningful human control. The letter warns us that "Artificial Intelligence technology has reached a point where the deployment of such systems is–practically if not legally–feasible within years (Hawking, Musk, Wozniak, et al., 2015)."

In August of 2018, artificial intelligence bots beat five human players at the video game Dota 2. OpenAI, an independent research institute cofounded by Elon Musk developed the bots, and used reinforcement learning to train for the match. In contrast to to chess or go, it is especially difficult to train machiness to play videogames, because the action takes place on a much larger board, where not all your opponent's moves are visible, and it requires that players make decisions quickly.

Appendix D

Model Measures of Fit

Typical model measures-of-fit(MOF) developed in hydrologic modeling is listed in Table D.1 and explained here.

Table D.1: Summary of the variables used in the implementation of loss functions.

MOF	Name	Type	Ideal Value	Range
MAE	Mean Absolute Error	absolute measure	0	$[0, \infty)$
MSE	Mean Squared Error	absolute measure	0	$[0, \infty)$
RMSE	Root Mean Squared Error	absolute measure	0	$[0, \infty)$
nRMSE	Normalized RMSE	absolute measure	0	$[0, \infty)$
RSR	RMSE standard deviation ratio	absolute measure	0	$[0, \infty)$
RSD	Relative Standard Deviation	supporting measure	1	$(-\infty, \infty)$
RMU	Relative Mean	supporting measure	1	$(-\infty, \infty)$
PBIAS	Percent Bias	supporting measure	0	$(-100\%, 100\%)$
R^2	Coefficient of Determination	measure of linearity in simulated vs. predicted	1	$[0, 1]$
bR^2	Weighted R^2	bias corrected R^2	1	$[0, 1]$
NSE	Nash-Sutcliffe Efficiency	square difference measure of fit	1	$(-\infty, 1]$
d	Index of Agreement	square difference measure of fit	1	$[0, 1]$
mNSE	Modified NSE	sensitivity to peaks can be modified	1	$(-\infty, 1]$
md	Modified d	sensitivity to peaks can be modified	1	$[0, 1]$
rNSE	Relative NSE	sensitivity to peaks eliminated	1	$(-\infty, 1]$
rd	Relative d	sensitivity to peaks eliminated	1	$[0, 1]$
KGE	Kling-Gupta Efficiency	relative importance of error component made explicit	1	$(-\infty, 1]$

MOF	Name	Type	Ideal Value	Range
VE	Volumetric Efficiency	volumes made important no matter if it is in a peak or recession	1	$(-\infty, 1]$

See Equations D.1 to D.8 where Y_i^{obs} are the observed unimpaired flows, and Y_i^{sim} are the predicted or simulated unimpaired flows, and n is the number of observations.

$$MAE = \frac{\sum_{i=1}^{n} \left| Y_i^{sim} - Y_i^{obs} \right|}{n} \tag{D.1}$$

$$MSE = \frac{\sum_{i=1}^{n} \left(Y_i^{sim} - Y_i^{obs} \right)^2}{n} \tag{D.2}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(Y_i^{sim} - Y_i^{obs} \right)^2}{n}} \tag{D.3}$$

$$nRMSE = \frac{RMSE}{MU_{obs}} = \frac{\sqrt{\sum_{i=1}^{n} \left(Y_i^{sim} - Y_i^{obs} \right)^2}}{\overline{Y^{obs}}} \tag{D.4}$$

$$RSR = \frac{RMSE}{\sigma_{obs}} = \frac{\sqrt{\sum_{i=1}^{n} \left(Y_i^{sim} - Y_i^{obs} \right)^2}}{\sqrt{\sum_{i=1}^{n} \left(Y_i^{obs} - \overline{Y^{obs}} \right)^2}} \tag{D.5}$$

The MAE, MSE, RMSE, nRMSE, RSR, are absolute measures of error.

$$RSD = \frac{\sigma_{sim}}{\sigma_{obs}} = \frac{\sqrt{\sum_{i=1}^{n} \left(Y_i^{sim} - \overline{Y^{sim}} \right)^2}}{\sqrt{\sum_{i=1}^{n} \left(Y_i^{obs} - \overline{Y^{obs}} \right)^2}} \tag{D.6}$$

$$RMU = \frac{\overline{Y^{sim}}}{\overline{Y_i^{obs}}} = \frac{\sum_{i=1}^{n} Y^{sim}}{\sum_{i=1}^{n} Y^{obs}} \tag{D.7}$$

$$PBIAS = \frac{\sum_{i=1}^{n}\left(Y_i^{obs} - Y_i^{sim}\right) * 100}{n\sum_{i=1}^{n}\left(Y_i^{obs}\right)} \tag{D.8}$$

The RSD, RMU, and PBIAS are additional supporting measures of error.

$$R^2 = \left(\frac{\sum_{i=1}^{n}\left(Y_i^{obs} - \overline{Y^{obs}}\right)\left(Y_i^{sim} - \overline{Y^{sim}}\right)}{\sqrt{\sum_{i=1}^{n}\left(Y_i^{obs} - \overline{Y^{obs}}\right)^2}\sqrt{\sum_{i=1}^{n}\left(Y_i^{sim} - \overline{Y^{sim}}\right)^2}}\right)^2 \tag{D.9}$$

R^2 is insensitive to additive and proportional difference between model simulation and observations. One can simply show that for a non zero value of β_0 and β_1, if the predictions follow a linear form, $Y^{sim} = \beta_0 + \beta_1 Y^{obs}$, the R^2 equals one (Legates & McCabe Jr, 1999). Therefore, for a proper model assessment, it is recommended that the slope of the predicted vs. observed graph be reported or systematically included as in Equation D.10.

$$bR^2 = \begin{cases} |b|\, R^2 & \text{for b} \leq 1 \\ |b|^{-1}\, R^2 & \text{for b} > 1 \end{cases} \tag{D.10}$$

By weighting R^2 under or over predictions are quantified together with the dynamics which results in a more comprehensive reflection of model results.

$$NSE = 1 - \frac{\sum_{i=1}^{n}\left(Y_i^{sim} - Y_i^{obs}\right)^2}{\sum_{i=1}^{n}\left(Y_i^{obs} - \overline{Y^{obs}}\right)^2} \tag{D.11}$$

A Nash-Sutcliffe efficiency factor of lower than zero indicates that the mean value of the observed time series would have been a better predictor than the model. The largest disadvantage of the Nash-Sutcliffe efficiency factor is the fact that the differences between the observed and predicted values are calculated as squared values. As a result, larger values in a time series are strongly overestimated whereas lower values are neglected (Legates & McCabe Jr, 1999). For the quantification of runoff predictions this leads to an overestimation of the model performance during peak flows and an underestimation during low flow conditions (Krause et al., 2005).

To reduce the problem of the squared differences and the resulting sensitivity to extreme values the Nash-Sutcliffe efficiency factor is often calculated with logarithmic values of Y_i^{sim} and Y_i^{obs}. Through the logarithmic transformation of the runoff values the peaks are flattened and the low flows are kept more or less at the same level. As a result the influence of the low flow values is increased in comparison to the flood peaks resulting in an increase in sensitivity of $ln(NSE)$ to systematic model over or under prediction (Krause et al., 2005).

$$d = 1 - \frac{\sum_{i=1}^{n} \left(Y_i^{sim} - Y_i^{obs}\right)^2}{\sum_{i=1}^{n} \left(\left|Y_i^{sim} - \overline{Y^{obs}}\right| + \left|Y_i^{obs} - \overline{Y^{obs}}\right|\right)^2} \tag{D.12}$$

$$mNSE = 1 - \frac{\sum_{i=1}^{n} \left|Y_i^{sim} - Y_i^{obs}\right|^j}{\sum_{i=1}^{n} \left|Y_i^{obs} - \overline{Y^{obs}}\right|^j}, \quad j \in \mathbb{N} \tag{D.13}$$

$$md = 1 - \frac{\sum_{i=1}^{n} \left|Y_i^{sim} - Y_i^{obs}\right|^j}{\sum_{i=1}^{n} \left(\left|Y_i^{sim} - \overline{Y^{obs}}\right| + \left|Y_i^{obs} - \overline{Y^{obs}}\right|\right)^j}, \quad j \in \mathbb{N} \tag{D.14}$$

For j=1, the overestimation of the flood peaks in regular NSE is reduced significantly resulting in a better overall evaluation. j=3 is best for flood modeling.

$$rNSE = 1 - \frac{\sum_{i=1}^{n} \left(\frac{Y_i^{sim} - Y_i^{obs}}{Y_i^{obs}}\right)^2}{\sum_{i=1}^{n} \left(\frac{Y_i^{obs} - \overline{Y^{obs}}}{\overline{Y^{obs}}}\right)^2} \tag{D.15}$$

$$rd = 1 - \frac{\sum_{i=1}^{n} \left(\frac{Y_i^{sim} - Y_i^{obs}}{Y_i^{obs}}\right)^2}{\sum_{i=1}^{n} \left(\frac{\left|Y_i^{sim} - \overline{Y^{obs}}\right| + \left|Y_i^{obs} - \overline{Y^{obs}}\right|}{\overline{Y_i^{obs}}}\right)^2} \tag{D.16}$$

As a result, an over or under prediction of higher values (i.e., peaks) has, in general, a greater influence than those of lower values. Therefore, we can use relative values in the regular NSE equations. These equations will not be sensitive to peaks at all.

$$\begin{aligned} KGE = & \ 1 - \sqrt{(r-1)^2 + (\beta-1)^2 + (\gamma-1)^2}, \\ r = & \ Pearson's \ r, \\ \beta = & \ \frac{\overline{Y^{sim}}}{\overline{Y^{obs}}}, \\ \gamma = & \ \frac{C_v^{sim}}{C_v^{obs}} = \frac{\frac{\sigma_{sim}}{\overline{Y^{sim}}}}{\frac{\sigma_{obs}}{\overline{Y^{obs}}}} \end{aligned} \tag{D.17}$$

The Kling Gupta Efficiency (KGE) factor facilitates the analysis of the relative importance of its different components: r, correlation and timing; β: magnitude and bias; and γ: variability).

$$VE = \ 1 - \frac{\sum_{i=1}^{n} \left| Y_i^{sim} - Y_i^{obs} \right|}{\sum_{i=1}^{n} \left(Y_i^{obs} \right)} \tag{D.18}$$

To solve the problems presented with reporting bias in hydrologic models, the Volumetric Efficiency (VE) can be used. It is easy to calculate, and of treats every unit volume of water the same as any other unit volume, whether it be delivered during slow recession or during peak flow (?, ?).

In conclusion, the optimal benchmark will differ for different applications, which is why so many benchmarks have been proposed in hydrology. It is especially critical when the model measure of fit it to be used as a loss function in a machine learning algorithm. These discretionary choices tend to disappear when complex modeling is concerned. Therefore, the criteria for decisions should be made explicit and known before modeling begins.

Moriasi et al. (2007) provides a table of gneral performance ratings for recommended statistics for a monthly time step, useful for the modeling done in this dissertation D.3.

Table D.3: General performance ratings for recommended statistics for a monthly time step. Reprinted from Moriasi et al., 2007.

Performance Rating	RSR	NSE	PBIAS
Very good	$[0.0, 0.5]$	$(0.75, 1.00]$	$(-\infty, \pm 10)$
Good	$(0.5, 0.6]$	$(0.65, 0.75]$	$[\pm 10, \pm 15)$
Satisfactory	$(0.6, 0.7]$	$(0.50, 0.65]$	$[\pm 15, \pm 25)$
Unsatisfactory	$(0.7, \infty)$	$(-\infty, 0.50]$	$[\pm 25, \infty)$

References

Abrahart, R. J., Heppenstall, A. J., & See, L. M. (2007). Timing error correction procedure applied to neural network rainfall–runoff modelling. *Hydrological sciences journal*, *52*(3), 414–431.

Amazon Web Services. (2018). Amazon sagemaker: Build, train, and deploy machine learning models at scale. *Amazon*. Retrieved from https://aws.amazon.com/sagemaker/features/

Asefa, T., Kemblowski, M., McKee, M., & Khalil, A. (2006). Multi-time scale stream flow predictions: the support vector machines approach. *Journal of Hydrology*, *318*(1), 7–16.

Bayes, M., & Price, M. (1763). An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfrs. *Philosophical Transactions (1683-1775)*, 370–418.

Beaudette, M. D. (2016). Package 'sharpshootr'.

Beven, K. J. (2011). *Rainfall-runoff modelling: the primer*. John Wiley & Sons.

Bivand, R., Keitt, T., & Rowlingson, B. (2018). rgdal: Bindings for the 'geospatial' data abstraction library [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=rgdal (R package version 1.3-6)

Bivand, R., & Rundel, C. (2018). rgeos: Interface to geometry engine - open source ('geos') [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=rgeos (R package version 0.4-2)

Bivand, R. S., Pebesma, E., & Gomez-Rubio, V. (2013). *Applied spatial data analysis with R, second edition*. Springer, NY. Retrieved from http://www.asdar-book.org/

Bray, M., & Han, D. (2004). Identification of support vector machines for runoff modelling. *Journal of Hydroinformatics*, *6*(4), 265–280.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (1993). Classification and regression trees. wadsworth, 1984. *Google Scholar*.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Bronowski, J. (1988). The nature of scientific reasoning. *Occasions for Writing*, 443–45.

Brownlee, J. (2014). 4-steps to get started in machine learning: The top-down strategy for beginners to start and practice. *ML Mastery*. Retrieved from https://machinelearningmastery.com/4-steps-to-get-started-in-machine-learning/

California Department of Water Resources, Bay-Delta Office. (2016). Estimates of natural and unimpaired flows for the central valley of california: Water years 1922-2014.

Chamberlain, S., & Teucher, A. (2018). geojsonio: Convert data from and to 'geojson' or 'topojson' [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=geojsonio (R package version 0.6.0)

Chaney, N. W., Wood, E. F., McBratney, A. B., Hempel, J. W., Nauman, T. W., Brungard, C. W., & Odgers, N. P. (2016). Polaris: A 30-meter probabilistic soil series map of the contiguous united states. *Geoderma*, *274*, 54–67.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21–27.

Cranz, A. (2018). Microsoft kinect refuses to die. *Gizmodo*. Retrieved from https://gizmodo.com/microsoft-kinect-refuses-to-die-1825847023

Dawson, C. W., & Wilby, R. (1998). An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal*, *43*(1), 47–66.

Dean, J., & Ng, A. (2015). Using large-scale brain simulations for machine learning and ai. *Official Google Blog*, *26*. Retrieved from https://www.blog.google/technology/ai/using-large-scale-brain-simulations-for/

DeJong, G. (1981). Generalizations based on explanations. *Urbana*, *51*(61,801).

Dettinger, M. D., Ralph, F. M., Das, T., Neiman, P. J., & Cayan, D. R. (2011). Atmospheric rivers, floods and the water resources of california. *Water*, *3*(2), 445–478.

Dooge, J. C. (1973). *Linear theory of hydrologic systems* (No. 1468). Agricultural Research Service, US Department of Agriculture.

Dooge, J. C. (1986). Looking for hydrologic laws. *Water Resources Research*, *22*(9S).

Duan, N., Manning, W. G., Morris, C. N., & Newhouse, J. P. (1983). A comparison of alternative models for the demand for medical care. *Journal of business & economic statistics*, *1*(2), 115–126.

Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, *92*(438), 548–560.

Flint, L., & Flint, A. (2014). California basin characterization model: a dataset of historical and future hydrologic response to climate change. *US Geological Survey Data Release doi*, *10*, F76T0JPB.

Forest Service, USDA, Pacific Southwest Region. (2006). Existing vegetation–vegetation classification and mapping for region 5.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer series in statistics Springer, Berlin.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267–285). Springer.

Galelli, S., & Castelletti, A. (2013). Tree-based iterative input variable selection for hydrological modeling. *Water Resources Research*, *49*(7), 4295–4310.

Giner, G., & Smyth, G. K. (2016). statmod: probability calculations for the inverse gaussian distribution. *R Journal*, *8*(1), 339-351.

Godsey, S. E., Kirchner, J. W., & Tague, C. L. (2014). Effects of changes in winter snowpacks on summer low flows: case studies in the sierra nevada, california, usa. *Hydrological Processes*, *28*(19), 5048–5064.

Govindaraju, R. S., & Rao, A. R. (2013). *Artificial neural networks in hydrology* (Vol. 36). Springer Science & Business Media.

Grubinger, T., Kobel, C., & Pfeiffer, K.-P. (2010). Regression tree construction by bootstrap: Model search for drg-systems applied to austrian health-data. *BMC Medical Informatics and Decision Making*, *10*(1), 1.

Grubinger, T., Zeileis, A., Pfeiffer, K.-P., et al. (2011). *evtree: Evolutionary learning of globally optimal classification and regression trees in r*. Department of Economics (Inst. für Wirtschaftstheorie und Wirtschaftsgeschichte).

Guisan, A., & Zimmermann, N. E. (2000). Predictive habitat distribution models in ecology. *Ecological modelling*, *135*(2), 147–186.

Han, D., Chan, L., & Zhu, N. (2007). Flood forecasting using support vector machines. *Journal of hydroinformatics*, *9*(4), 267–276.

Härdle, W., & Simar, L. (2007). *Applied multivariate statistical analysis* (Vol. 22007). Springer.

Hart, E. M., & Bell, K. (2015). prism: Download data from the oregon prism project [Computer software manual]. Retrieved from http://github.com/ropensci/prism (R package version 0.0.6) doi: 10.5281/zenodo.33663

Hastie, T., & Tibshirani, R. (1990). *Generalized additive models* (Vol. 43). CRC Press.

Hawking, S., Musk, E., Wozniak, S., et al. (2015). *Autonomous weapons: an open letter from ai & robotics researchers. future of life institute.*

Hayes, B., et al. (2013). First links in the markov chain. *American Scientist*, *101*(2), 252.

Hennig, C., & Kutlukaya, M. (2007). Some thoughts about the design of loss functions. *REVSTAT–Statistical Journal*, *5*(1), 19–39.

Hijmans, R. J. (2019). raster: Geographic data analysis and modeling [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=raster (R package version 2.8-19)

Hijmans, R. J., Phillips, S., Leathwick, J., & Elith, J. (2017). dismo: Species distribution modeling [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=dismo (R package version 1.1-4)

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527–1554.

Ho, T. K. (1995). Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on* (Vol. 1, pp. 278–282).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, *79*(8), 2554–2558.

Hrachowitz, M., Savenije, H., Blöschl, G., McDonnell, J., Sivapalan, M., Pomeroy, J., ... others (2013). A decade of predictions in ungauged basins (pub)—a review. *Hydrological sciences journal*, *58*(6), 1198–1255.

Hsu, K.-l., Gupta, H. V., Gao, X., Sorooshian, S., & Imam, B. (2002). Self-organizing linear output map (solo): An artificial neural network suitable for hydrologic modeling and analysis. *Water Resources Research*, *38*(12).

Hu, T., Wu, F., & Zhang, X. (2007). Rainfall–runoff modeling using principal component analysis and neural network. *Hydrology Research*, *38*(3), 235–248.

Ingle, K. (2017). Machine learning–mind map cheatsheet. *Medium*. Retrieved from https://medium.com/@karan.ingle/machine-learning-mind-map-cheatsheet-cb200b2246fe

Iorgulescu, I., & Beven, K. J. (2004). Nonparametric direct mapping of rainfall-runoff relationships: An alternative approach to data analysis and modeling? *Water Resources Research*, *40*(8).

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 6). Springer.

Jarvis, A., Reuter, H. I., Nelson, A., Guevara, E., et al. (2008). Hole-filled srtm for the globe version 4. *available from the CGIAR-CSI SRTM 90m Database*. Retrieved from http://srtm.csi.cgiar.org

Jorgensen, B. (1997). *The theory of dispersion models*. CRC Press.

Klemes, V. (1982). Empirical and causal models in hydrology.

Krause, P., Boyle, D., & Bäse, F. (2005). Comparison of different efficiency criteria for hydrological model assessment. *Advances in geosciences*, *5*, 89–97.

Legates, D. R., & McCabe Jr, G. J. (1999). Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation. *Water resources research*, *35*(1), 233–241.

Legendre, P. (1993). Spatial autocorrelation: trouble or new paradigm? *Ecology*, *74*(6),

1659–1673.

Levins, R. (1966). The strategy of model building in population biology. *American scientist*, *54*(4), 421–431.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R news*, *2*(3), 18–22.

Lin, J.-Y., Cheng, C.-T., & Chau, K.-W. (2006). Using support vector machines for long-term discharge prediction. *Hydrological Sciences Journal*, *51*(4), 599–612.

Magnuson-Skeels, B. (2016). *Using machine learning to statistically predict natural flow.* MS Thesis.

Marr, B. (2016). A short history of machine learning every manager should read. *Forbes*. Retrieved from http://tinyurl.com/gslvr6k

Mauricio Zambrano-Bigiarini. (2017). hydrogof: Goodness-of-fit functions for comparison of simulated and observed hydrological time series [Computer software manual]. Retrieved from http://hzambran.github.io/hydroGOF/ (R package version 0.3-10) doi: 10.5281/zenodo.840087

McKay, L., Bondelid, T., Dewald, T., Johnston, J., Moore, R., & Rea, A. (2012). Nhdplus version 2: user guide. *National Operational Hydrologic Remote Sensing Center, Washington, DC*.

Min, Y., & Agresti, A. (2002). Modeling nonnegative data with clumping at zero: a survey. *Journal of the Iranian Statistical Society*, *1*(1), 7–33.

Minns, A., & Hall, M. (1996). Artificial neural networks as rainfall-runoff models. *Hydrological sciences journal*, *41*(3), 399–417.

Moriasi, D., Arnold, J., Van Liew, M., Bingner, R., Harmel, R., & Veith, T. (2007). Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, *50*(3), 885-900. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-34447500396&partnerID=40&md5=50b5724614f28257edef46d43db96018 (cited By 2311)

Nelder, J. A., & Wedderburn, R. W. M. (1972). *Generalized linear models*. Wiley Online Library.

Neuwirth, E. (2014). Rcolorbrewer: Colorbrewer palettes [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=RColorBrewer (R package version 1.1-2)

O'Connor, J., & Robertson, E. (2000). Biography of pierre-simon laplace and article on orbits and gravitation. *Published by School of Mathematics and Statistics, University of St Andrews, Scotland.*. Retrieved from http://www-history.mcs.standrews.ac.uk/history/Mathematicians/Laplace.html

Pebesma, E. J., & Bivand, R. S. (2005, November). Classes and methods for spatial data in R. *R News*, *5*(2), 9–13. Retrieved from https://CRAN.R-project.org/doc/Rnews/

Pike, R. J., & Wilson, S. E. (1971). Elevation-relief ratio, hypsometric integral, and geomorphic area-altitude analysis. *Geological Society of America Bulletin*, *82*(4), 1079–1084.

Poff, N. L., Allan, J. D., Bain, M. B., Karr, J. R., Prestegaard, K. L., Richter, B. D., ... Stromberg, J. C. (1997). The natural flow regime. *BioScience*, *47*(11), 769–784.

R Core Team. (2018). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from https://www.R-project.org/

Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., ... others (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*.

Rolle, O. (2015). Googles tensorflow and microsofts dmtk goes open source. *PosiDev Blog*. Retrieved from http://posidev.com/blog/2015/11/14/googles-tensorflow-and-microsofts-dmtk-goes-open-source/

RStudio Team. (2016). Rstudio: Integrated development environment for r [Computer software manual]. Boston, MA. Retrieved from http://www.rstudio.com/

Sherman, L. K. (1932). Streamflow from rainfall by the unit-graph method. *Eng. News Record*, *108*, 501–505.

Simonite, T. (2014). Software that matches faces almost as well as you do. *Technology Review*, *117*(3), 19–19.

Singh, V. P., & Frevert, D. K. (2005). *Watershed models*. CRC Press.

Sivapalan, M. (2003). Prediction in ungauged basins: a grand challenge for theoretical hydrology. *Hydrological Processes*, *17*(15), 3163–3170.

Sivapalan, M., Takeuchi, K., Franks, S., Gupta, V., Karambiri, H., Lakshmi, V., ... others (2003). Iahs decade on predictions in ungauged basins (pub), 2003–2012: Shaping an exciting future for the hydrological sciences. *Hydrological sciences journal*, *48*(6), 857–880.

Spence, C., & Woo, M.-k. (2006). Hydrology of subarctic canadian shield: heterogeneous headwater basins. *Journal of Hydrology*, *317*(1-2), 138–154.

Stigler, S. M. (1981). Gauss and the invention of least squares. *The Annals of Statistics*, 465–474.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Tobin, J. (1958). Estimation of relationships for limited dependent variables. *Econometrica: journal of the Econometric Society*, 24–36.

Todini, E. (1988). Rainfall-runoff modeling past, present and future. *Journal of Hydrology*, *100*(1), 341–352.

Tokar, A. S., & Johnson, P. A. (1999). Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, *4*(3), 232–239.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*(5), 988–999.

Wickham, H. (2007). Reshaping data with the reshape package. *Journal of Statistical Software*, *21*(12), 1–20. Retrieved from http://www.jstatsoft.org/v21/i12/

Winston, P. (2010). *6.034 artificial intelligence, fall 2010. massachusetts institute of technology: MIT opencourseware.* Retrieved from https://ocw.mit.edu.License: CreativeCommonsBY-NC-SA

Woodie, A. (2014). Inside sibyl, google's massively parallel machine learning platform. *Datanami*. Retrieved from https://www.datanami.com/2014/07/17/inside-sibyl-googles-massively-parallel-machine-learning-platform/

Worland, S. C., Farmer, W. H., & Kiang, J. E. (2018). Improving predictions of hydrological low-flow indices in ungaged basins using machine learning. *Environmental Modelling & Software*, *101*, 169–182.

Wuertz, D., Setz, T., & Chalabi, Y. (2017). fbasics: Rmetrics - markets and basic statistics [Computer software manual]. Retrieved from https://CRAN.R-project.org/

package=fBasics (R package version 3042.89)

Yu, X., Liong, S.-Y., & Babovic, V. (2004). Ec-svm approach for real-time hydrologic forecasting. *Journal of Hydroinformatics*, *6*(3), 209–223.

Zehe, E., & Sivapalan, M. (2008). Threshold behavior in hydrological systems and geo-ecosystems: manifestations, controls and implications for predictability. *Hydrology & Earth System Sciences Discussions*, *5*(6).

Zeileis, A., & Hothorn, T. (2002). Diagnostic checking in regression relationships. *R News*, *2*(3), 7–10. Retrieved from https://CRAN.R-project.org/doc/Rnews/

Elaheh White
September 2020
Civil and Environmental Engineering

Machine Learning in Predicting Ungauged Basin Flows

Abstract

COPY PASTE FROM ABOVE WHEN DONE!!!