Machine Learning in Predicting Ungauged Basin Flows

Proposal for Doctoral Research

Ellie White

Department of Civil & Environmental Engineering

University of California, Davis

Advisor: Jay R. Lund

December 7th, 2018

Abstract

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) allow for the application of statistical learning methods to broader societal problems. The advantages of statistical learning models lie in their fast and easy development, simplicity of use, lesser data requirements, good performance, and flexibility in model structure and parameter specifications.

This proposal’s overall goal is to demonstrate the pitfalls and potential of statistical learning techniques in water resources, especially when aimed at Predicting Ungauged Basin (PUB) flows. Chapter one explains this study’s motivation and research questions. Chapter two discusses methods and knowledge gaps in the field of statistical learning for hydrologic modeling. Chapter three provides a heuristic guide for model selection. Chapters four and five present research in the development of appropriate loss functions, and the application of appropriate resampling methods for model prediction error estimation (i.e., cross-validation and bootstrap). Chapter six examines the implications of big data, statistical learning, and artificial intelligence in the larger context of a society and political economy.

The thesis 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*** Statistical Learning, Water Resources, Model Selection, Loss Functions, Cross-Validation, Bootstrap, McDonaldization

Contents

## Objectives & Contributions to the Field

The overall objective of this thesis is to contribute to understanding the appropriate application of statistical learning techniques and their methods of prediction error estimation (i.e., cross-validation and bootstrap). Statistical learning techniques are generally simpler modeling alternatives than traditional physically-based models in water resources. Expected contributions of this work include: (1) a heuristic guide to empirical model selection; (2) appropriate new loss functions for statistical learning methods in hydrology; (3) methods for more accurate estimation of model prediction error when using resampling methods (i.e., cross-validation and bootstrapping); and (4) putting the pursuit of the next-best algorithm in the context of our society and political economy.

## 1 Introduction & Motivation

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). Terms and concepts used in this paper are defined in Appendix A.

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 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.

Hydrologic models in PUB can be classified as *mechanistic* (physical process-based) or *empirical* (statistical) (Guisan & Zimmermann, 2000). 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-19th century, with the employment of the *rational method*, empirical relationships have been used in rainfall-runoff modeling (Beven, 2011). Engineers developed 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).

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.

As Solomatine and Ostfeld (2008) explain, most machine learning techniques applied to the rainfall-runoff problem use neural networks (Govindaraju & Rao, 2013). Other studies use support vector machines (Lin, Cheng, & Chau, 2006) and tree based algorithms (Galelli & Castelletti, 2013). Despite the wide range of empirical models that could be employed–suggesting that no one single modeling method is useful across all locations, timescales, and problems–these methods are much easier to apply and study than mechanistic models, but it is often harder to understand and explain why they perform well. The merits of statistical learning techniques, as a subset of empirical models, motivates this study and its goals are outlined below.

### 1.2 Scientific Motivation & Research Questions

This research is to develop statistical models, generally simpler than mechanistic models, that use statistical learning techniques to predict the unimpaired flow of California basins from available data. The objectives are as follows:

*Chapter 3* - Heuristic Guide for Model Selection

**Objective**: Contribute to the understanding and appropriate application of statistical learning methods. These methods should not be viewed as black-box.

*Chapter 4* - New Loss Functions

**Objective**: Encourage examination of off-the-shelf machine learning methods, specifically their default objective function. Identify some characteristics to consider when designing a loss function for machine learning algorithms. Examine how different loss functions perform when used to model hydrology.

*Chapter 5* - Rethinking Resampling Methods

**Objective**: Caution against the possibly inappropriate use of easy-to-apply random resampling methods in the model’s prediction error estimation. Describe how to more accurately estimate a model’s prediction error with resampling methods when the data is structured. Structured data has internal correlation structures that need to be accounted for when resampling.

*Chapter 6* - Beyond McDonaldization: The “Robotanization" of Agriculture

**Objective**: Look to the impacts of statistical learning on our society and economy. Discuss how statistical learning is changing our society and what “irrationalities" may it produce or take away that make it distinct from past mechanization.

## 2 Statistical Learning in Hydrology

Our responsibility is to do what we can, learn what we can, improve the solutions, and pass them on.

----------------------------------------------------------------------------------------------

Richard P. Feynman, *“What Do You Care What Other People Think?”*, 1988

### 2.1 Summary

Statistical learning techniques 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.

This chapter discusses some of these applications and presents some weaknesses and strengths of empirical modeling.

### 2.2 Introduction

Artificial intelligence has gone through the ages of speculation (1940s), dawn, business, bulldozer age (Winston, 2010). See Appendix B for a brief history of statistical learning. In the bulldozer age, with seemingly unlimited computing capacity we can 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). However, the application of these newer techniques to water resources problems is slow. Therefore, the literature reviewed below is on computationally-intensive methods.

### 2.3 Literature Review

Possible applications of statistical learning methods in water resources include the following:

• Rainfall-runoff modeling: streamflow dis-aggregation or forecasting

• Building an assisting surrogate model in calibration of a rainfallÐrunoff model

• Auto-calibration and uncertainty estimation methods

• Modeling river stage-discharge relationships

• Cleaning up anomalies in data: (cite Nate Chaney’s work with SSURGO, step like patterns in temperature from rounding observed data.)

• Simulating runoff, sediment and nutrient loadings from a watershed

• Modeling rainfallÐrunoff processes intelligent controller for realtime management

• Control of water levels in channels

• Model-based optimal control of a reservoir

• Forecasting the groundwater heads in an aquifer

• River temperature prediction

• Rainfall forecasting

• Water demand forecasting

• Flood forecasting

• Predicting scour depth at culverts

• Pipe deterioration models

• Optimizing operations of a water distribution network

Some applications listed above are expanded on here. As Solomatine and Ostfeld (2008) explain, most machine learning techniques applied to the rainfall-runoff problem use neural networks (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). Other studies use support vector machines (Asefa, Kemblowski, McKee, & Khalil, 2006, Lin et al., 2006), and tree based algorithms (Iorgulescu & Beven, 2004, Galelli & Castelletti, 2013, Magnuson-Skeels, 2016).

Han, Chan, and Zhu (2007), Yu, Liong, and Babovic (2004), and Bray and Han (2004) used *support vector machines* (SVMs) for flood forecasting. The applications of SVM in regression of time series is still in its infancy. Such studies show that advances are putting SVMs generally on par with neural networks in terms of model performance.

Han et al. (2007) shows 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. This contradicts the physical principle that limited rainfall cannot generate an unlimited flow. However, the model performs well given the data it has trained and was tested on (Han et al., 2007).

Yu et al. (2004) used SVMs for real-time hydrologic forecasting. In this study, 16 years of daily data informs the learning or phase space reconstruction and the last year of record is used for a 1-lead day prediction. One problem with SVMs is their difficulty in dealing with a large training record, which is required in chaotic time series analysis. To make SVMs more suitable to such problems, a decomposition method can be applied, which is demonstrated (Yu et al., 2004).

Bray and Han (2004) explored the relationships among various model structures, kernel functions (e.g., linear, polynomial, radial basis, and sigmoid), scaling factor, model parameters (i.e., cost C and epsilon) and composition of input vectors in the development of the SVM.

Gautam and Holz (2001) proposes an adaptive neuro-fuzzy system with autoregressive exogenous input (ARX) structure which was capable of reasonably producing the hydrograph shape and was able to maintain a good representation of the overall water balance as well as the general flow pattern. Its excellent performance for one-step prediction shows that it will be very valuable for real-time forecasting and control of floods (Gautam & Holz, 2001).

Ames, Neilson, Stevens, and Lall (2005) used *Bayesian networks* to model water management decisions in complying with phosphorous loadings in a river. This study estimated the probability of meeting legal water quality requirements for phosphorus in a creek under several management scenarios and estimated the probability of increased recreational use of the reservoir on the creek and subsequent revenue under these scenarios. The Bayesian networks framework served as a structured means for capturing the probability that management activities will have the desired effect (Ames et al., 2005).

Shrestha and Solomatine (2009) used the UNcertainty Estimation based on Local Errors and Clustering (UNEEC) method for assessing total model uncertainty (i.e., model error in reproducing observed historical river flow data). The UNEEC method is a data driven technique that consists of clustering, estimation of the probability distribution of error, and building the model of the probability distribution of error (Shrestha & Solomatine, 2009).

Alvisi, Franchini, and Marinelli (2007) used an auto-regressive moving average (ARMA) model to forecast water demand. Since operational decisions have to be based on the expected future demands for water, rather than just the present known requirements, it is necessary to develop a short-term, demand-forecasting procedure. Trends, seasonality, and general periodicities are usually found in these type of time-series, which make them ideal candidates for ARMA models.

### 2.4 Limitations & Assumptions

Many hydrologists are skeptical of statistical modeling. Klemes (1982) warns modelers of the general limitations of empirical modeling, the most important are as follows:

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 (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 (See Figure C.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 (See Figure 2.1; Dettinger, Ralph, Das, Neiman, and Cayan, 2011).

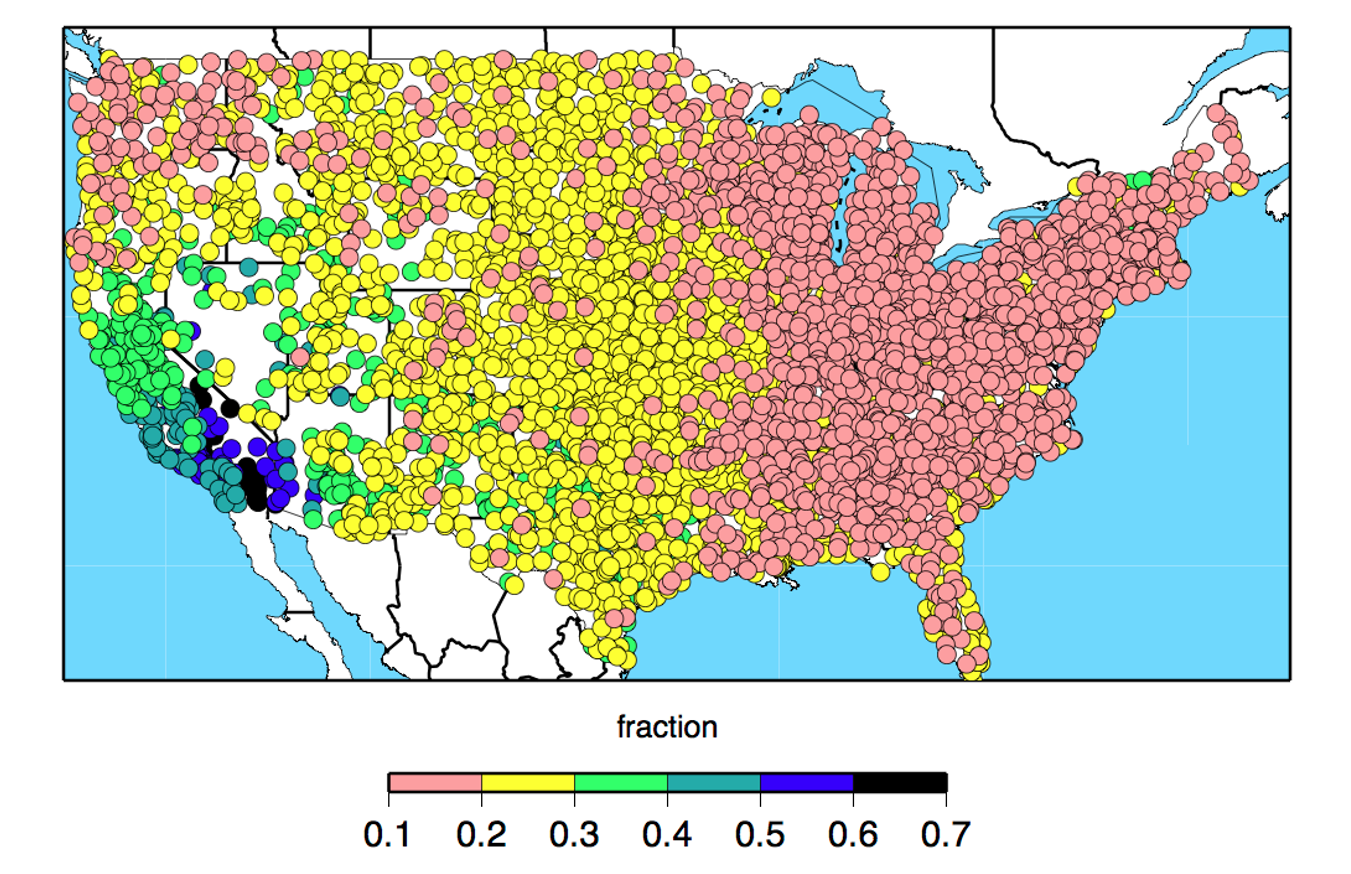


Figure 1: Coefficient of variation in total precipitation from 1951-2008, Dettinger et al. (2011)

In addition to concerns with spatial extrapolation, there is temporal extrapolation. Climate change brings **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 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 (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; in other words, 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.

### 2.5 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, everything seems to be dirt to push. 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. Despite the wide range of empirical models that could be employed, suggesting that no one single modeling method is useful across all locations, timescales, and problems, these methods are much easier, faster, and less expensive to apply and study than mechanistic models. These methods are well suited to dynamic, non-linear and sometimes noisy data, especially when underlying physical processes are complex or not fully understood. Also, the purpose of modeling is often to inform decision makers with adequate timing. For example, models need to be run during and just before floods. 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.

## 3 Heuristic Guide for Model Selection

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.

### 3.1 Summary

Theorists in statistical learning have made progress in developing methods, especially in supervised machine learning, for a range of problems. However, if a statistical learning method works particularly well for one data set, it may be that the problem was defined in a space rich with solutions (Winston, 2010). That is, other statistical learning methods also could produce good results. Therefore, the merits of finding a solution may be due to either the statistical learning method or the problem’s characteristics. Researcher can apply other statistical learning methods.

This chapter presents a guide for finding suitable statistical methods, particularly in supervised machine learning. As a heuristic guide, it will work for many problems, but may fail to recommend a good method in some cases.

### 3.2 Introduction

Given a research question, statistical learning follows steps outlined in Figure 3.1. This chapter looks at step two (model selection), and the next chapter examines step three (loss or objective function definition). However, 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.

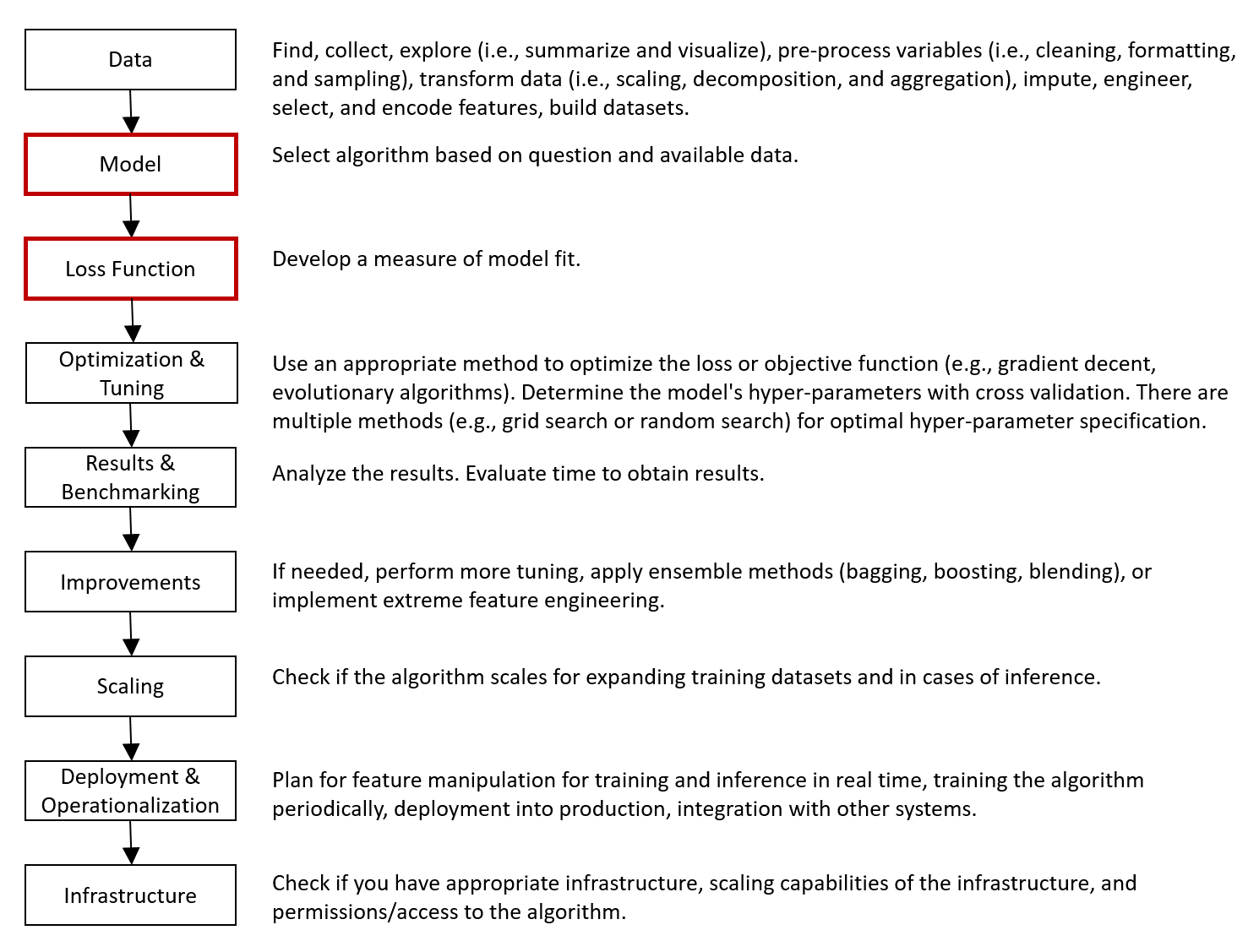


Figure 2: Statistical learning application follows several steps, adapted from Brownlee (2014), Ingle (2017). Steps two and three will be examined here.

### 3.3 Statistical Learning Methods

The guide presented in Figure 3.2 groups statistical methods into six main categories: supervised machine learning, regression family, time series analysis, 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 that need extrapolation into the future. 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.

Figure 3.2 expands on the six main categories explained above. Appendix A explains the terms and concepts used here.

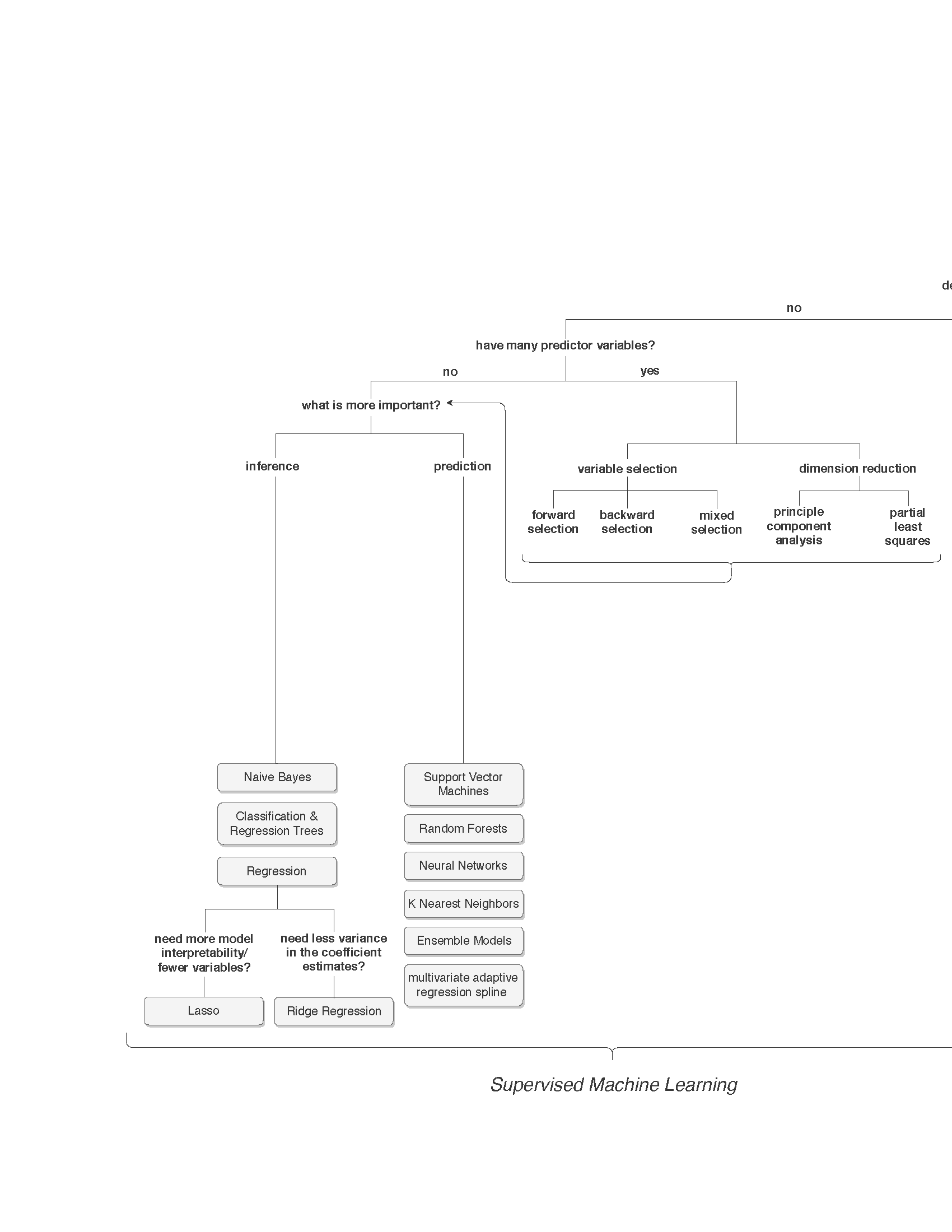


Figure 3: What are you trying to do? Heuristic guide for model selection.

Given the guide mentioned above, we can develop the following model types for the PUB problem:

(1) A multi-variate linear regression model: or to be precise a generalized linear regression (GLM) model where the *random component* has a known distribution form. The *systematic component*, or the explanatory variables, are mixed continuous and categorical. The *link* between the random and systematic components is a function describing how the expected value of the response relates to the linear predictor of explanatory variables. In this case, the link is logarithmic, because, we often assume that the probability distribution of unimpaired flow to follow a Log-Pearson Type III, a lognormal distribution. Log-Pearson Type III is a gamma distribution used for continuous, positive, right-skewed data, and where variance is near-constant on the log-scale. We have to account for this non-normality in the residuals, stemming from non-normality in the response variable, when using regression methods. The GLM removes some restrictions and assumptions normal linear models make (Penn State Department of Statistics Online Programs, 2018).

(2) A recurrent neural network (RNN): a special case of neural networks for sequential data. A traditional neural network assumes that all inputs (and outputs) are independent of each other, so, it uses different parameters in each layer. A RNN shares the same parameters across all connected steps, and performs the same task just on different inputs. This feature greatly reduces the total number of parameters the model needs to learn. In chapters 4 and 5, we will develop a special type of RNN, a long short-term memory (LSTM), which is much better at capturing long-term dependencies.

(3) A multivariate adaptive regression spline (MARS): a non-parametric regression technique that uses recursive partitioning. In chapters 4 and 5, we will develop a special type of MARS model, a time series MARS (TMARS), where the predictors are lagged time series values resulting in autoregressive spline models.

### 3.4 Model Selection From Developed Alternatives

Once a set of suitable modeling methods are applied, statistical analysis of the results can hep in choosing the “best" model. What is meant by best is subjective. In prediction, when the model is an estimator of a physical quantity, the better model more closely predicts the observed data. This highlights the importance of first defining a loss function (See Chapter 4). Also, *Bias* and *variance* are important measures of the estimator’s quality.

In addition, a good model selection will balance simplicity with prediction accuracy. This principle is known as *Occam’s Razor*: given models with similar predictive power, the model with the fewest variables (i.e., simpler) is preferable. Along the same lines, we can select a model using the *one-standard-error rule* (Friedman et al., 2001). Here, various models are compared based on their test set error. We can then select the simplest model within one standard error of the model with the lowest test set error. Therefore, efficiency is considered as well as prediction accuracy.

### 3.5 Conclusion

This chapter introduces 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. A dissertation chapter on this topic would expand and document these methods and selection steps. It will further distinguish between the methods presented at the terminal nodes of the tree in a comparative table. Other taxonomies and classifications will be presented and compared to the one developed here.

## 4 New Loss Functions

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

### 4.1 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.

### 4.2 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 ℓ1−*norm* and ℓ2−*norm* (See Equation 4.1 and 4.2). The ℓ2−*norm* is the familiar objective function in simple least-squares regression, a convex function, emphasizing points distant from the bulk of the data.

 (1)



(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 4.3 and 4.4, the empirical risks associated with the ℓ1−*norm* and ℓ2−*norm*. The expectation of the ℓ2−*norm* will produce a model that regresses to the mean, and the ℓ1−*norm* regresses to the median. That is, the ℓ2−*norm* is more sensitive to outliers than the ℓ1−*norm*.

(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 4.5).

(5)



### 4.3 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 C, Figure C.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 2C); 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 C.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), 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 D 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 4.1). Such loss functions can be defined as linear exponential (LINEX) loss if smoothness is desired (See Equation 4.6). 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 4.2 and Equation 4.7 explain such a function.

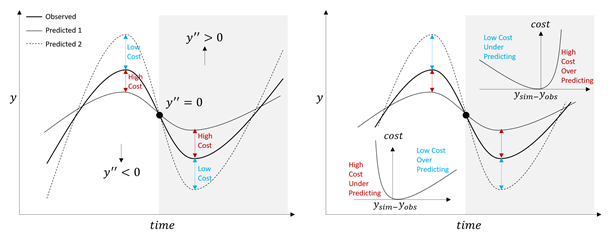


Figure 4: Asymmetric loss functions define different losses to over predicting and under predicting a value.

(6)

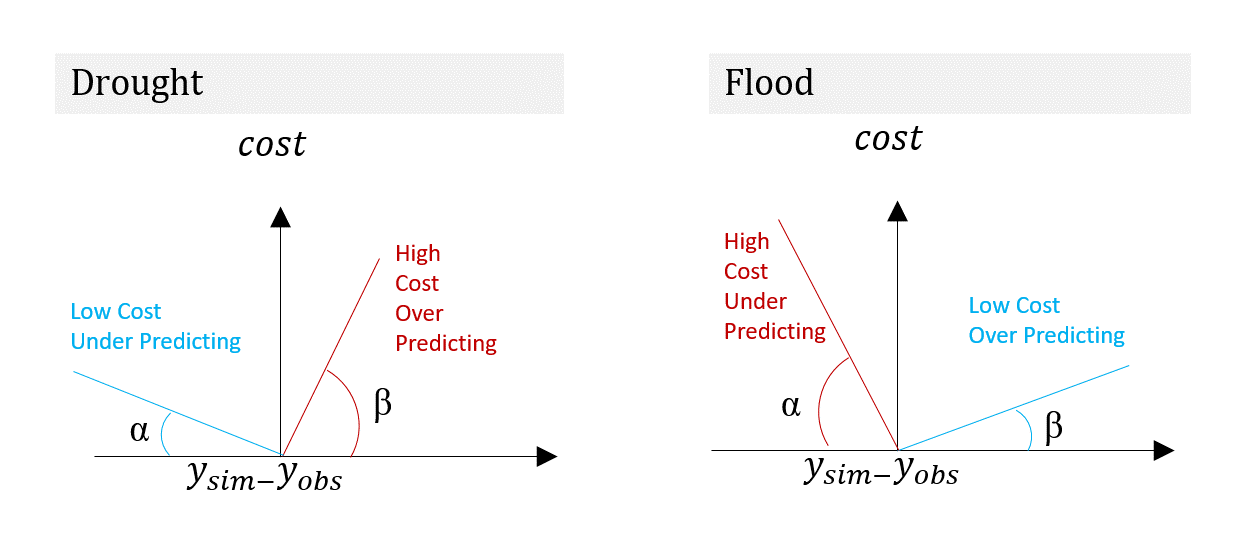


Figure 5: Asymmetric loss functions define different losses to over predicting and under predicting a value.

(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 ℓ2−*norm* (See Equation 4.8).

(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, Boyle, & Bäse, 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).

### 4.4 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.

## 5 Rethinking Resampling Methods

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

### 5.1 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.

### 5.2 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 5.1).

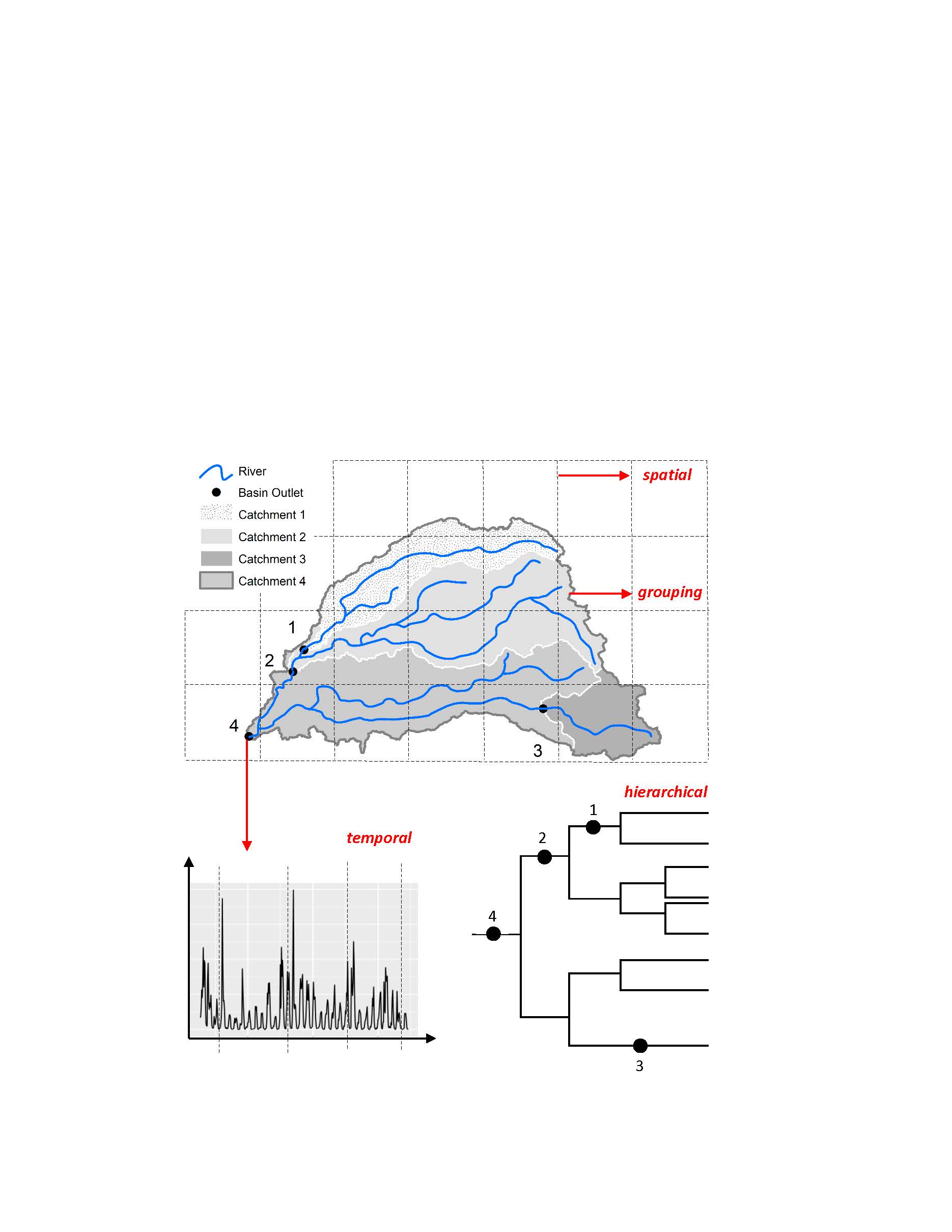


Figure 6: The four types of dependence structures in gauged data and blocking strategies

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 (Friedman et al., 2001). The expected error in the test set is a polynomial of power two (See Equation A.1) and is comprised of variance, bias, and a constant term. In the first 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 1/3rd 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 5.2). For at a distance, Δ*d*, from , where and are autocorrelated. The distance Δ*d* 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.

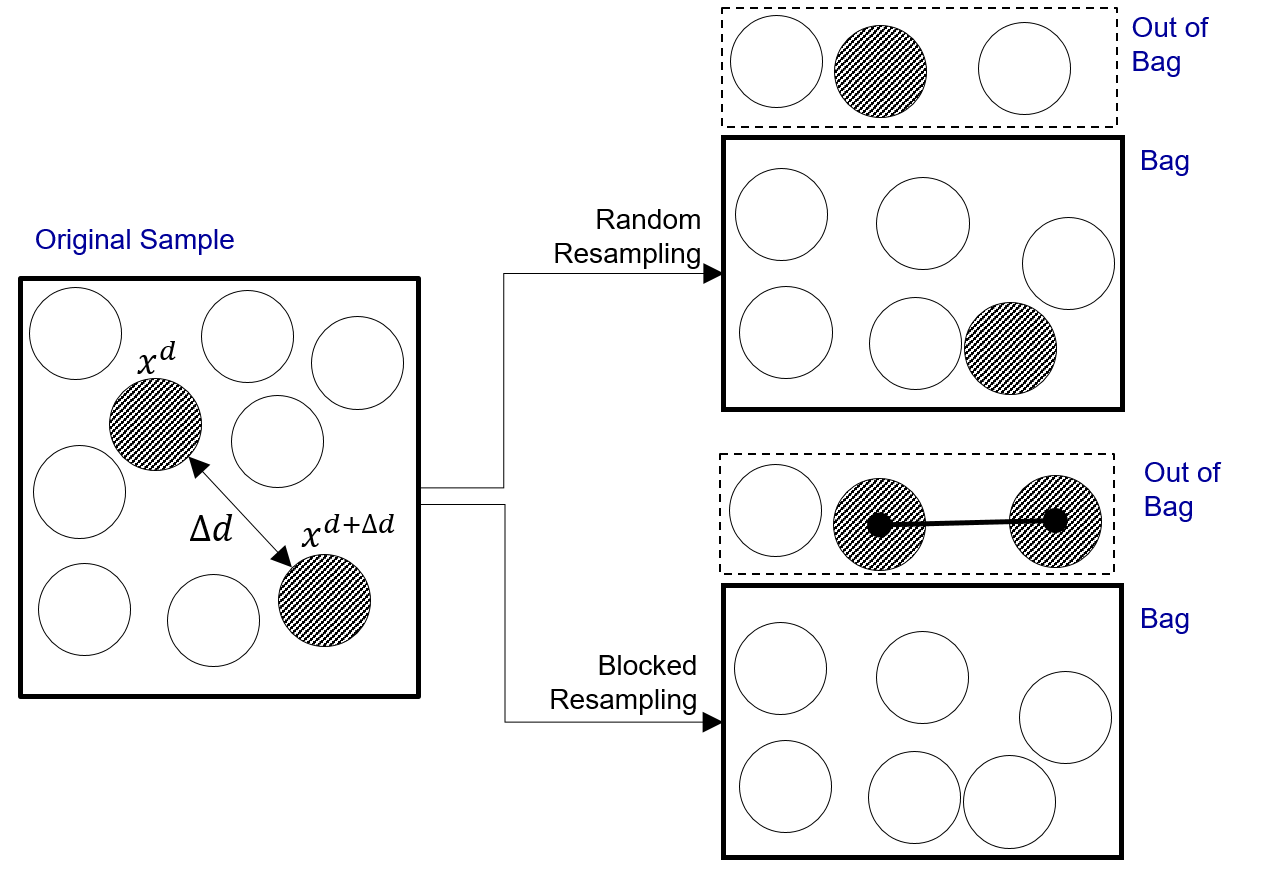


Figure 7: Autocorrelation is a pseudo replication problem. The two grey marbles are autocorrelated. A model that uses random resampling will be able to easily predict one grey marble since it has seen the other. When blocking, the observations move in and out of the bag together.

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 5.3).

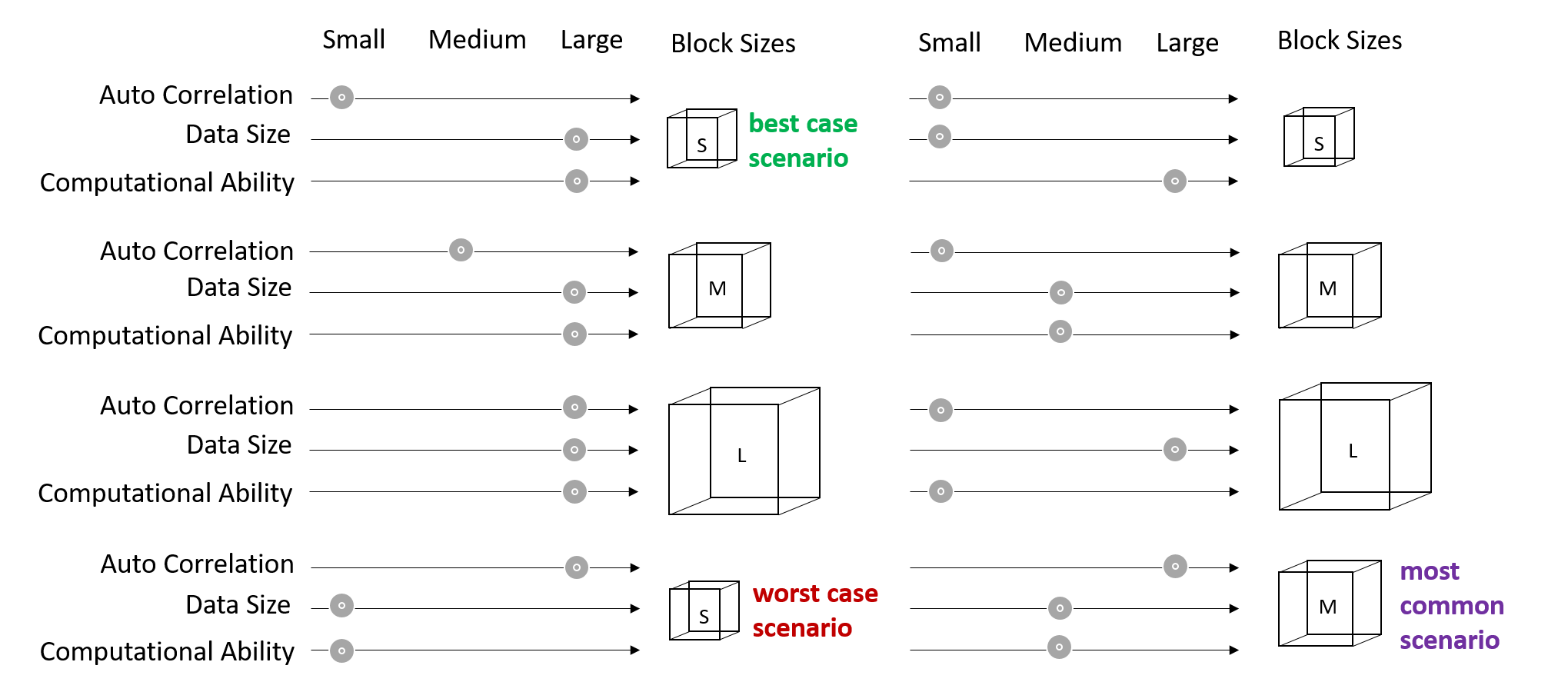


Figure 8: The block size in resampling methods is a function of the autocorrelation, data size, and computational ability.

### 5.3 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*=α*x*+β); (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 unknowable. 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 C.1). The developed data set has approximately 18,500 monthly unimpaired flow observations in acre-feet (AF) (See Appendix E for more info). The data spans from 1895 to 2018 at a monthly time step. As mentioned in Chapter 3, 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 5.4 &5.5).

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 5.6).

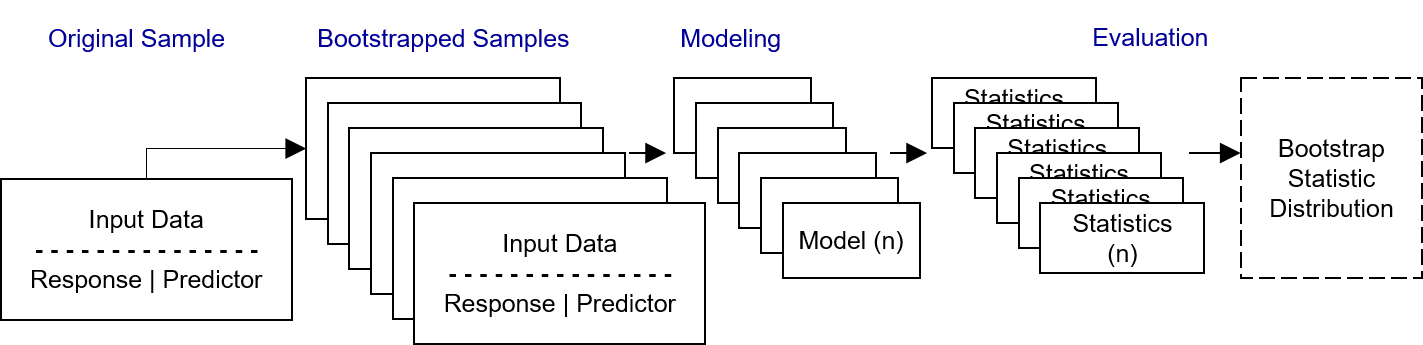


Figure 9: Research design: We employ the bootstrap method to find the distribution of the bootstrap statistic.

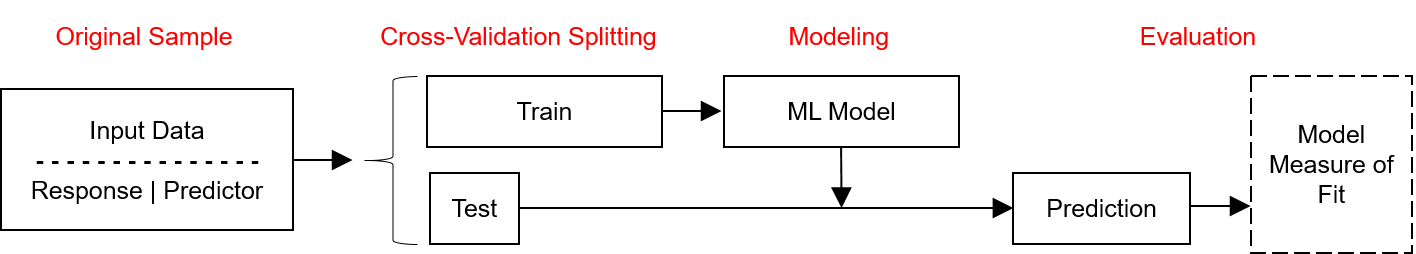


Figure 10: Research design: We employ the cross-validation method to find the model error estimate.

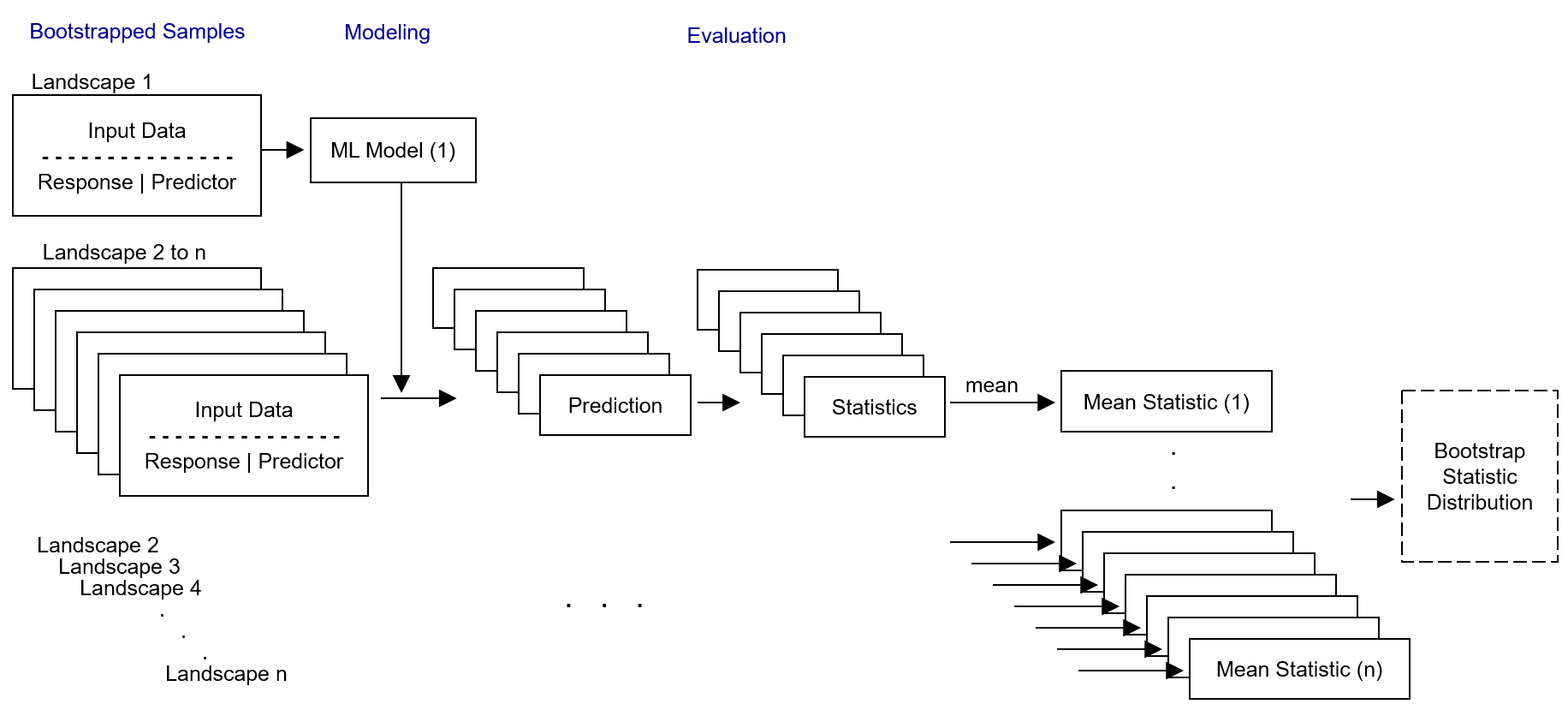


Figure 11: Research design: compare the errors with that of an “ideal" model.

#### 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, 1/5th or 1/10th 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.

### 5.4 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.

## 6 Beyond McDonaldization: The “Robotanization" of Agriculture

No increase in material wealth will compensate for arrangements which insult people’s self-respect and impair their freedom.

-------------------------------------------------------------------------------

Richard H. Tawney, *“Religion and the Rise of Capitalism”*, 1926

### 6.1 Summary

Given the increasingly popular statistical learning methods and their resampling techniques, the sociological effects of these methods become important fro for both scholarship and application. In 2013, Monsanto paid $930 Million USD to acquire The Climate Corporation (Upbin, 2013). Monsanto is a large publicly traded agricultural multinational corporation, and The Climate Corporation is a digital agriculture company that examines field data. In other words, Òbig-farmaÓ has learned the potential of Òbig data.Ó Recent technological advances in big data is impacting the nature of agricultural (i.e., arable, livestock, horticulture, and fishery farming) work in various and complex ways.

This chapter will explore the impacts of big data (BD), machine learning (ML), and artificial intelligence (AI) on arable farming and studies of farming. BD refers to data sets that are too large or complex for traditional data-processing application software to adequately deal with (Wikipedia contributors, 2018). ML is the development of computer programs that analyzes data and automatically learns and improves from experience without being explicitly programmed (Koza, Bennett, Andre, & Keane, 1996). ML is an application in the larger field of AI. BD, ML, and/or AI in farming, hereinafter referred to as “robotany", manifest as a combination of data-heavy decision making and precision agriculture.

Part of my research will be preparing a literature review on historical and recent mechanization trends in agriculture (e.g., the tomato harvester ÒsavingÓ the California tomato) and the sociological process of McDonaldization. My research will cover these topics in the Science and Technology Studies discipline and the Cybernetics sub-discipline.

### 6.2 Literature Review

The major influences in the development of the first tomato picking harvester (See Figure 6.1) were technologies such as, effective machines, specially bred tomatoes, careful irrigation and fertilization, and particular planting techniques, another major influence was a societal phenomenon: the fear of a lack of labor to handle the tomato crop, as the Brarcero migrant-worker program had ended (Rasmussen, 1968).

The two major hurdles to developing a mechanical tomato picker were the susceptibility of the tomato to bruising, thus making it a tough candidate for machine harvesting, and the fact that tomatoes did not all ripen at the same time, that is, they usually required multiple passes through the field. In the 1940s, UC Davis researchers successfully removed both obstacles; they developed a pear-shaped tomato particularly adapted to machine harvesting that would also ripen around the same time (Rasmussen, 1968). Uniformity in shape and ripening time, brings the predictability needed for mechanization. Predictability is one aspect of McDonaldization.



Figure 12: Blawelder tomato harvester makes fundamental changes to the way scientists think about plants; the science was then less about the delicious tomato but about the tomato that can pass through the machine and get to the market Fell, Bailey, Kerlin, and Griffith (2016)

The more recent tools continuing the tradition of mechanization (i.e., robotany) can be categorized based on their designed functions: (a) crop management, including applications on yield prediction, disease detection, weed detection, crop quality, and species recognition; (b) livestock management, including applications on animal welfare and livestock production; (c) water management; and (d) soil management (Liakos, Busato, Moshou, Pearson, & Bochtis, 2018). The author claims that tools developed in these areas are helping farmers enhance yields, improve efficiency, and manage their risk. Efficiency is another dimension of McDonaldization.

McDonaldization has four dimensions: efficiency, calculability, predictability, and control (Ritzer, 2002). Ritzer (2009) uses the meat industry to describe the four dimensions of this process. In this paper, the organic movement is introduced as an alternative to the industrial model and a Òrebellion against McDonaldization.Ó After discussing the irrationalities of McDonaldization, the author acknowledged the positive outcomes of this process: abundance of cheap food and the availability of products year round. The author did not speculate on whether McDonaldization will win over the organic food movement or if organic food will ever become mainstream but ends with acknowledging the risks (i.e., externalities or irrationalities) that are being ignored in the process and asks: Òcould it be that the long arm of McDonaldization is reaching too far? (Ritzer, 2009)

### 6.3 Two Case Studies

Large agribusinesses deploy and operationalize most technologies in robotany. These technologies are usually proprietary and patented. For example, IntelinAir, with one headquarter in San Jose, California, has developed drones and airplanes with MRI-like imaging, called Ag-MRIª, to help identify anomalies within a field. Therefore, interventions (e.g., the application of chemicals and water) can be applied discerningly.

There are much fewer examples of small farmers using robotany. In one example, Makoto Koike, a former designer in the Japanese automobile industry and the son of cucumber farmers, used GoogleÕs open source ML algorithm to build a machine that sorts cucumbers by size, shape, color, quality and other features (Sato, 2017).

### 6.4 Research Objectives

An examination of the two case studies above (i.e., robotany used by IntelinAir and Makoto Koike) can help determine weather robotany has and will be distinctly different from past processes of McDonaldization and mechanization. This chapter will discuss the effects of the open-source movement and the availability of rental cloud computing services, that allow for small farmers to use this technology. Finally, it will discuss the irrationalities that robotany may produce or quell for our society.

### 6.5 Tentative Hypothesis

The difference between past mechanization and robotany is perhaps the extent to which we are using robotany to control nature and therefore the extent to which it can transform farming practices. For the farmer, the specialization of labor (e.g., no multi-cropping, or crop rotations) leading into the mechanization of labor spelled the loss of control over workday, when to plant, when to weed, how to cultivate, etc. Robotany isn’t necessarily a new and distinctly different process, but perhaps is exacerbating the loss of control through non-human means.

In many instances, because the machines will learn to optimize farming of one crop at a time (called Òweak AIÓ), it will likely reproduce certain types of agricultural systems that are less environmentally sustainable and characterized by growing skill gaps between on-the-ground laborers, who are still going to be needed at least a little, and those who are the tech gurus behind the systems. This process is racialized; people of color will more likely be in laborers the field than behind the system. However, Òstrong AIÓ or Ògeneral AIÓ aims to replace human laborers as a whole; the goal is to mimic the entirety of human intelligence. If AI successfully gets here, it will spell non-specialization of machine labor. Here, humans are entirely superfluous to the farming process and thus the questions concerning on-the-ground human farm labor disappear.

### 6.6 Conclusion

According to Ritzer (2002), the irrationality of rationality is the fifth dimension of McDonaldization. Robotany may produce irrationalities in production (e.g., genetic modification and reduced genetic diversity of seeds, data privacy and security), harvesting (e.g., loss of jobs, waste, and environmental degradation), and consumption (e.g., consumer appetite for uniformity). These irrationalities have important implications for public policy. As it stands, we have two choices: the possibility of a better life with less labor and more leisure time to be creative, or to face mass unemployment and continued wealth and income inequality.

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.

Alvisi, S., Franchini, M., & Marinelli, A. (2007). A short-term, pattern-based model for water- demand forecasting. *Journal of hydroinformatics* , *9* (1), 39–50.

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

Ames, D. P., Neilson, B. T., Stevens, D. K., & Lall, U. (2005). Using bayesian networks to model watershed management decisions: an east canyon creek case study. *Journal of hydroinfor- matics* , *7* (4), 267–282.

Analytics, R., & Weston, S. (2017). iterators: Provides iterator construct for r [Computer software manual]. Retrieved from [https://CRAN.R-project.org/package=iterators](https://CRAN.R-project.org/package%3Diterators) (R package version 1.0.9)

Asefa, T., Kemblowski, M., McKee, M., & Khalil, A. (2006). Multi-time scale stream flow predic- tions: 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.

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

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

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.

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.

Corporation, M., & Weston, S. (2017). doparallel: Foreach parallel adaptor for the ’parallel’ pack- age [Computer software manual]. Retrieved from [https://CRAN.R-project.org/package= doParallel](https://CRAN.R-project.org/package%3DdoParallel) (R package version 1.0.11)

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:](https://gizmodo.com/microsoft-kinect-refuses-to-die-1825847023)

[//gizmodo.com/microsoft-kinect-refuses-to-die-1825847023](https://gizmodo.com/microsoft-kinect-refuses-to-die-1825847023)

Criss, R. E., & Winston, W. E. (2008). Do nash values have value? discussion and alternate proposals. *Hydrological Processes: An International Journal* , *22* (14), 2723–2725.

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](https://www.blog.google/technology/ai/using-large-scale-brain-simulations-for/)

[-scale-brain-simulations-for/](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). Edmund, H. (2015). Package ‘prism’.

Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: the 632+ bootstrap method.

*Journal of the American Statistical Association* , *92* (438), 548–560.

Fell, A., Bailey, P., Kerlin, K., & Griffith, D. (2016). 13 uc davis discoveries that changed our world. *UC Davis News* . Retrieved from [https://www.ucdavis.edu/news/13-discoveries](https://www.ucdavis.edu/news/13-discoveries-changed-our-world/)

[-changed-our-world/](https://www.ucdavis.edu/news/13-discoveries-changed-our-world/)

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 classifi- cation 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.

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.

Gautam, D., & Holz, K.-P. (2001). Rainfall-runoff modelling using adaptive neuro-fuzzy systems.

*Journal of Hydroinformatics* , *3* (1), 3–10.

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

Science & Business Media.

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.

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

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., proceed- ings 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.

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:](https://medium.com/%40karan.ingle/machine-learning-mind-map-cheatsheet-cb200b2246fe)

[//medium.com/@karan.ingle/machine-learning-mind-map-cheatsheet-cb200b2246fe](https://medium.com/%40karan.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:](http://srtm.csi.cgiar.org/)

[//srtm.csi.cgiar.org](http://srtm.csi.cgiar.org/)

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

Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In *Artificial intelligence in design96* (pp. 151–170). Springer.

Krause, P., Boyle, D., & B¨ase, 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.

Liakos, K., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. *Sensors* , *18* (8), 2674.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R News* , *2* (3),

18-22. Retrieved from <http://CRAN.R-project.org/doc/Rnews/>

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 learningevery manager should read. *Forbes* . Retrieved from <http://tinyurl.com/gslvr6k>

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* .

Microsoft, & Weston, S. (2017). foreach: Provides foreach looping construct for r [Computer software manual]. Retrieved from [https://CRAN.R-project.org/package=foreach](https://CRAN.R-project.org/package%3Dforeach) (R package version 1.4.4)

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

Nelder, J. A., & Wedderburn, R. W. M. (1972). *Generalized linear models*. Wiley Online Library. NRCS, USDA. (2006). Land resource regions and major land resource areas of the united states,

the caribbean, and the pacific basin. *US Department of Agriculture Handbook* , *296*

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](http://www-history.mcs.standrews.ac.uk/history/Mathematicians/Laplace.html)

[-history.mcs.standrews.ac.uk/history/Mathematicians/Laplace.html](http://www-history.mcs.standrews.ac.uk/history/Mathematicians/Laplace.html)

Penn State Department of Statistics Online Programs. (2018). *Stat 504 analysis of dis- crete data, fall 2018. penn state: Eberly college of science.* Retrieved from [https:](https://onlinecourses.science.psu.edu/stat504/node/49/)

[//onlinecourses.science.psu.edu/stat504/node/49/](https://onlinecourses.science.psu.edu/stat504/node/49/)

R Core Team. (2017a). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <https://www.R-project.org/>

R Core Team. (2017b). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <https://www.R-project.org/>

R Core Team. (2017c). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <https://www.R-project.org/>

R Core Team. (2017d). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <https://www.R-project.org/>

R Core Team. (2017e). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from <https://www.R-project.org/>

Rasmussen, W. D. (1968). Advances in american agriculture: The mechanical tomato harvester as a case study. *Technology and Culture* , *9* (4), 531–543.

Ritzer, G. (2002). An introduction to mcdonaldization. *McDonaldization: The Reader* , *2* , 4–25. Ritzer, G. (2009). *Mcdonaldization: the reader*. Pine Forge Press.

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 struc- ture. *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](http://posidev.com/blog/2015/11/14/googles-tensorflow-and-microsofts-dmtk-goes-open-source/)

[-microsofts-dmtk-goes-open-source/](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/>

Sato, K. (2017). Tensorflow brings machine learning to the masses. *opensource.com* . Retrieved from <https://opensource.com/article/17/9/tensorflow>

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

*108* , 501–505.

Shrestha, D., & Solomatine, D. (2009). Assessing uncertainty in rainfall-runoff models: Application of data-driven models.

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., 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.

Solomatine, D. P., & Ostfeld, A. (2008). Data-driven modelling: some past experiences and new approaches. *Journal of hydroinformatics* , *10* (1), 3–22.

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. Todini, E. (1988). Rainfall-runoff modelingpast, 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.

Upbin, B. (2013, October). *Monsanto buys climate corp for $930 million.* Retrieved from [https://www.forbes.com/sites/bruceupbin/2013/10/02/monsanto-buys-climate](https://www.forbes.com/sites/bruceupbin/2013/10/02/monsanto-buys-climate-corp-for-930-million/#12adc25c177a)

[-corp-for-930-million/#12adc25c177a](https://www.forbes.com/sites/bruceupbin/2013/10/02/monsanto-buys-climate-corp-for-930-million/#12adc25c177a)

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

Wikipedia contributors. (2018). *Big data — Wikipedia, the free encyclopedia.* Retrieved from [https://en.wikipedia.org/w/index.php?title=Big data&oldid=865688908](https://en.wikipedia.org/w/index.php?title=Big_data&amp;oldid=865688908) ([Online; accessed 2-November-2018])

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

[-NC-SA](https://ocw.mit.edu.License:CreativeCommonsBY-NC-SA/)

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

[-googles-massively-parallel-machine-learning-platform/](https://www.datanami.com/2014/07/17/inside-sibyl-googles-massively-parallel-machine-learning-platform/)

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

## A Terms & Concepts in Machine Learning

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

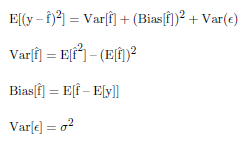
### Terminology

**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 A.1).

(9)



where *y* is the observed response variable, *x* is the observed predictor variable and *y*=*f*(*x*)+ε, ̂*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 5.

**Loss or Objective Function**: The expectation of the loss function, is the function that is minimized (or maximized) in a statistical learning algorithm. Figure

Table 1: @

xref fig:complossfuncs on input line 29 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 4.

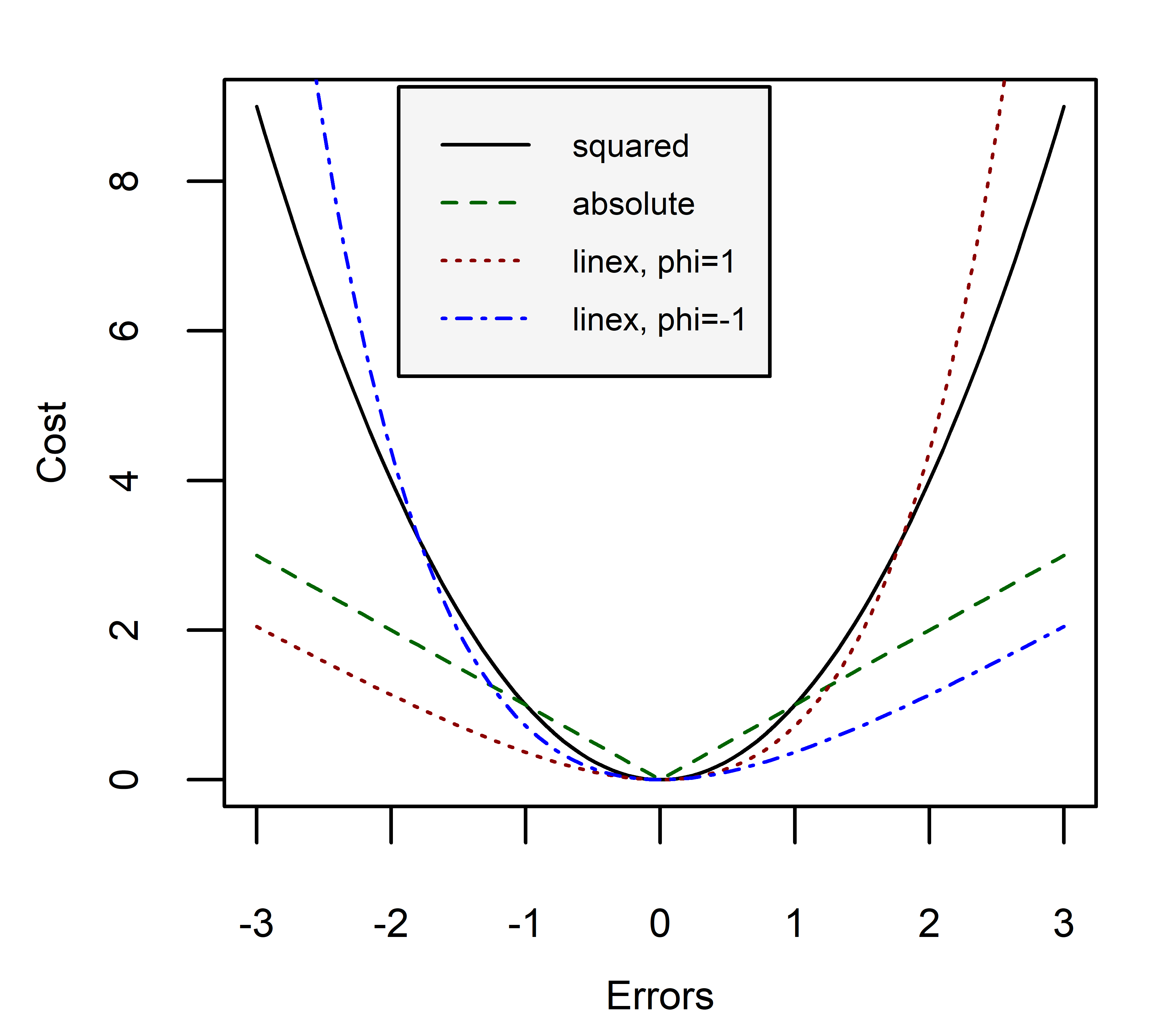


Figure 13: Typical loss functions in statistical learning.

**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 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*)+ε, 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., ̂*f*(*x*)) with the observations (i.e., ŷ). 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 (), 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 . We have effectively reduced the problem of finding ̂*f*(*x*) to finding .

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).

## B Brief History of Statistical Learning

This section explains how some of the ideas organized in Chapter 3’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 JŸrgen 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 1990Õs, 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 ÒseeÓ 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 opponentÕs moves are visible, and it requires that players make decisions quickly.

## C Unimpaired Flow Data set

The data set developed for Chapter 4 is described here. The watersheds in the study are depicted in Figure C.1, and the response and predictor variables in the dataframe are described in Table C.1.

Figure 14: The 69 California basins under study are the CDEC unimpaired flow basins.

1.0

tableSummary of the variables used in the implementation of loss functions.

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Variable | Description | Source |
| Response | Unimpaired Flow | monthly estimated unimpaired flows, in *AF* | CDEC (Beaudette, 2016) |
| Time | Month | categorical: Jan, Feb, ..., Dec | - |
|  | Ordinal Month | numerical distance till June: Jan:6, Feb:5, ..., Dec:6 |  |
|  | Season | categorical: Fall, Winter, Spring, Summer |  |
|  | Year | numeric |  |
| Climate | Temperature, Lag 1, 2 and 3 Months | temperature and lagged monthly temperature, in C | PRISM (Edmund, 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 | NHD2PLUS (McKay et al., 2012) |
|  | Shape | basin length/basin width, in *m*/*m* |  |
|  | Compactness | basin area/(basin perimeter), in |  |
| Soil | % Clay | percent clay in surface layer, in % | POLARIS (Chaney et al., 2016) |
|  | % Silt | percent silt in surface layer, in % |  |
|  | % 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 | , in |  |
| Land Cover | Vegetated | Percent of area in the basin vegetated in % | CALVEG (Forest Service, USDA, Pacific Southwest Region, 2006) |
| Ground Water | Dominant Geology | dominant rock type in basin, categorical | NRCS (NRCS, USDA, 2006) |
|  | Depth to Restricted Layer | in *cm* | POLARIS (Chaney et al., 2016) |

### Summary Stats

INSERT TABLE OR IMAGE.

## D Model Measures of Fit

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

1.0

tableSummary of the variables used in the implementation of loss functions. R>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MOF | Name | Type | Ideal Value | Range |
| MAE | Mean Absolute Error | absolute measure | 0 | [0,∞) |
| MSE | Mean Squared Error | absolute measure | 0 | [0,∞) |
| RMSE | Root Mean Squared Error | absolute measure | 0 | [0,∞) |
| nRMSE | Normalized RMSE | absolute measure | 0 | [0,∞) |
| RSR | RMSE standard deviation ratio | absolute measure | 0 | [0,∞) |
| RSD | Relative Standard Deviation | supporting measure | 1 | (−∞,∞) |
| RMU | Relative Mean | supporting measure | 1 | (−∞,∞) |
| PBIAS | Percent Bias | supporting measure | 0 | (−100%,100%) |
|  | Coefficient of Determination | measure of linearity in simulated vs. predicted | 1 | [0,1] |
|  | Weighted | bias corrected | 1 | [0,1] |
| NSE | Nash-Sutcliffe Efficiency | square difference measure of fit | 1 | (−∞,1] |
| d | Index of Agreement | square difference measure of fit | 1 | [0,1] |
| mNSE | Modified NSE | sensitivity to peaks can be modified | 1 | (−∞,1] |
| md | Modified d | sensitivity to peaks can be modified | 1 | [0,1] |
| rNSE | Relative NSE | sensitivity to peaks eliminated | 1 | (−∞,1] |
| rd | Relative d | sensitivity to peaks eliminated | 1 | [0,1] |
| KGE | Kling-Gupta Efficiency | relative importance of error component made explicit | 1 | (−∞,1] |
| VE | Volumetric Efficiency | volumes made important no matter if it is in a peak or recession | 1 | (−∞,1] |

See Equations D.1 to D.8 where are the observed unimpaired flows, and are the predicted or simulated unimpaired flows, and *n* is the number of observations.

(10)

(11)

(12)

(13)

(14)

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

(15)

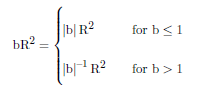
(16)

(17)

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

(18)

is insensitive to additive and proportional difference between model simulation and observations. One can simply show that for a non zero value of and , if the predictions follow a linear form, , the 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.



(19)

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

(20)

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 and . 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).

(21)

(22)

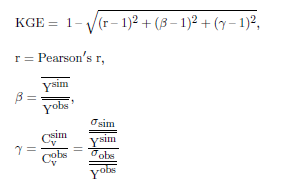
(23)

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.

(24)

(25)

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.



(26)

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).

(27)

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 (Criss & Winston, 2008).

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.

## E BCM Simulated Hydrology

The data set developed for Chapter 5 is described here. The watersheds in the study are depicted in Figure C.1, and the response and predictor variables in the dataframe are described in Table E.1.

1.0

tableSummary of the variables used in the implementation of the Random Forest model.

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Variable | Description | Source |
| Response | Runoff | monthly simulated unimpaired flows, in | USGS (Flint & Flint, 2014) |
| Time | Month | categorical: Jan, Feb, ..., Dec | - |
|  | Ordinal Month | numerical distance till June: Jan:6, Feb:5, ..., Dec:6 |  |
|  | Season | categorical: Fall, Winter, Spring, Summer |  |
|  | Year | numeric |  |
| Climate | Temperature, Lag 1, 2 and 3 Months | temperature and lagged monthly temperature, in C | PRISM (Edmund, 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 et al., 2008) |
|  | Mean Elevation | mean basin elevation, in *m* |  |
| Basin Boundaries | Area | basin drainage area, in | NHD2PLUS (McKay et al., 2012) |
|  | Shape | basin length/basin width, in *m*/*m* |  |
|  | Compactness | basin area/(basin perimeter), in |  |
| Soil | % Clay | percent clay in surface layer, in % | POLARIS (Chaney et al., 2016) |
|  | % Silt | percent silt in surface layer, in % |  |
|  | % 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 | , in |  |
| Land Cover | Vegetated | Percent of area in the basin vegetated in % | CALVEG (Forest Service, USDA, Pacific Southwest Region, 2006) |
| Ground Water | Dominant Geology | dominant rock type in basin, categorical | NRCS (NRCS, USDA, 2006) |
|  | Depth to Restricted Layer | in *cm* | POLARIS (Chaney et al., 2016) |

### Summary Stats

TO BE COMPLETED

## Acknowledgements

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.106933s3. 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.

All data processing and model development for Chapters 4 and 5 was done in R, a statistical programming language, (R Core Team, 2017a), RStudio (RStudio Team, 2016), and used the following packages: doParallel (Corporation & Weston, 2017), parallel (R Core Team, 2017b), iterators (Analytics & Weston, 2017), foreach (Microsoft & Weston, 2017), stats (R Core Team, 2017c), grDevices (R Core Team, 2017d), utils (R Core Team, 2017e), and randomForest (Liaw & Wiener, 2002). TO BE UPDATED!