Leveraging Classification Models for River Forecasting

Ruizhou Ding
Carnegie Mellon University
Pittsburgh, PA 15213
rding@andrew.cmu.edu

Diana Marculescu
Carnegie Mellon University
Pittsburgh, PA 15213
dianam@cmu.edu

ABSTRACT

Prior work in river forecasting has focused on applying regression models to gage and discharge prediction since these are naturally continuous dynamical functions. On the other hand, with discretized data, classifiers can be adopted to solve this problem by predicting a conditional probability distribution. Predicting this distribution is important in at least two ways: (1) the variance of the distribution can indicate the confidence of the predicted expected values, and (2) the distribution can be used for computing the probability that the gage or discharge exceeds or falls below some threshold. This paper presents a concrete river forecasting framework with classifiers including probabilistic graphical models (PGMs) and artificial neural network classifiers (ANNCs). The proposed framework is applied on real data for the Guadalupe river basin (Texas) thereby enabling a detailed comparison among various manners of forecasting studied, along with a set of guidelines for their best use.

CCS CONCEPTS

•Information systems →Geographic information systems; •Computing methodologies →Machine learning;

KEYWORDS

River forecasting, spatial-temporal modeling

1 INTRODUCTION

River forecasting is essential to human life and subsistence, especially for flood or drought prediction, ever since early agriculture has started to spread. More recently, the advent of hydropower generation, especially in the context of run-of-river hydropower projects [12], requires fine grain forecasting capabilities for potential energy availability. However, progress in forecasting river behavior has stalled, mainly due to several challenges coming from its application domain. First, the hydrologic eco-system is characterized by many inter-related factors with highly non-linear dynamical dependencies [10]. Second, the metrics used for assessing certain models must rely on their application domain. For example, to forecast the dynamical behavior of rivers, the expected value of future gage or discharge is produced as prediction, while for flood avoidance and hydropower availability, the probability that discharge exceeds or is below a certain threshold is needed. Third, the observed data are noisy, making complex forecasting models prone to overfitting.

Much attention in river forecasting was placed on regression models, ranging from linear models, to artificial neural network regression (ANNR) [1]. However, by quantizing the output variable to multiple discrete levels, the problem may be transformed to a classification problem, and the continuous predicted value can be computed based on the classification results. The advantages of using classification models are twofold: (1) Clustering continuous values to discrete levels can alleviate the measurement noise, thereby reducing the danger of overfitting. (2) Classifiers usually produce a probability distribution for the predicted variable instead of just an expected value. In our work, we implement the river forecasting problem on probabilistic graphical models (PGMs) and artificial neural network classifiers (ANNCs). A neighbor smoothing (NBS) strategy is proposed to address the inherent overfitting problem of PGMs due to their larger number of parameters.

The rest of paper is organized as follows. In section 2, we introduce prior work on river modeling. In section 3, the Guadalupe river dataset is introduced. In section 4, implementation of regression and classification models is described. Finally, in section 5, we describe our experiment setup and results on Guadalupe river data.

2 RELATED WORK

Past decades have witnessed great progress in river modeling and forecasting. Approaches used can be divided into two types: (1) conceptual models and (2) black-box models. Conceptual models aim at simulating the physical processes and transforming inputs to outputs guided by prior knowledge of the natural systems [11]. The development of conceptual models started early and have achieved good performance over the years. An example is SWAT (Soil and Water Assessment Tool), formed by combining and improving many well-performed river models, such as CREAMS, SWRRB, etc. [4].

Black-box approaches, on the other hand, rely more on data instead of knowledge of physical procedure. With much more data available, the data-driven black-box models are witnessing an increased popularity for river modeling. Before 1990s, traditional linear models including autoregressive integrated moving average (ARIMA) were most widely used [15]. However, river flow forecasting is believed to be highly non-linear and not easily described by simple models [10]. Hsu *et al.* used a neural network to model rainfall-runoff process, and achieved better performance than linear models [10]. Asadi *et al.* proposed a hybrid ANNR by combining GA and Levenberg-Marquardt (LM) algorithm for learning feed forward neural networks [5].

As the predicted variables (*e.g.* water flow) are continuous, regression models have been naturally adopted by almost all the

[1]For clarity, "ANNR" and "ANNC" are adopted to differentiate between ANNs used for regression and classification, respectively.

SIGSPATIAL'17, Los Angeles Area, CA, USA
 978-1-4503-5490-5/17/11…$15.00
DOI: 10.1145/3139958.3140048

previous work, while little work was done in the realm of classification models for predicting a probability distribution. The idea of using classification models for regression problems can be dated back to 1984, when Breiman *et al.* performed inference using regression trees that partition the range of continuous variables into multiple sections [7]. However, this idea was explored by only a few researchers, since most forecasting problems are formulated as regression problems where the target is a single value, *i.e.*, the conditional expected value. River forecasting, however, is concerned with predicting both the expected value and the conditional probability distribution [13], which can be further used to answer questions like: What is the probability that the river gage exceeds the flooding level or falls below the drought level? Or what is the probability that the river discharge exceeds the threshold required by hydropower generation? To the best of our knowledge, our work is the first to employ classification models including PGMs and ANNC for regression problems in river forecasting. This paper compares regression and classification models, as a guideline of model selection for river forecasting.

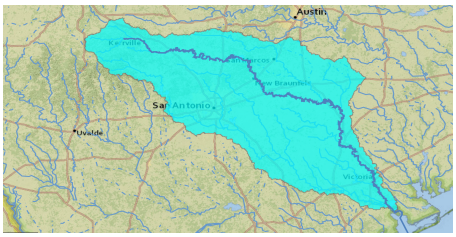

Figure 1: Guadalupe river basin is shown by the light blue area. The dark blue stream is the trace of main Guadalupe river. Figure generated from ArcGIS [3].

3 DATASET

Guadalupe river (shown in Figure 1) is located in the southeast of U.S., contains abundant hydropower resources, and is also prone to fluctuations. Guadalupe basin has a 3256 km^2 catchment area. Its length is 370 km, starting from ($30°05'17''N$, $99°38'32''W$), and flowing into Gull of Mexico at ($28°24'07''N$, $96°46'57''W$). The average discharge of Guadalupe river is 34 m^3/s [2]. Data ranging from April to late July in 2016 are studied, as drastic fluctuations of river flow happen in this period, making it harder to forecast. The data, provided by United States Geological Survey (USGS) [1], include gages, discharges, and precipitations at 15-minute intervals. Therefore, each time series has 11520 time steps. Gages and discharges of six big nodes are predicted, four of which are joint nodes with parent nodes from multiple branches. They are selected because the prediction for joint nodes exhibits more challenges. Features of eleven nodes, including the six nodes to predict, are collected in total, serving as inputs for forecasting. These settings can scale to any number, and we are showing these joint nodes because they are potentially more suitable for hydropower generation due to the large discharges.

4 METHODOLOGY

In this section, we first provide an overview of river models by formulating the forecasting problem, and introducing regression and classification models for river modeling. We then show how two classifiers, PGM and ANNC, can be used for the regression problems. Finally, we discuss metrics for assessing prediction performances.

4.1 Model Overview

4.1.1 Problem Formulation. We aim at forecasting two features characterizing river dynamics: gage and discharge. Gage is the water level, also called *water stage*. Discharge is the volume of water running per unit time, also called *water flow* or *runoff*. Along the river, there are several stations, also called *nodes*, measuring gage, discharge and precipitation at their own locations in real time. Suppose N nodes on a river are studied. We denote the gage at the n-th node ($n \in \{1, ..., N\}$) as time series $G^{(n)} = \{G_1^{(n)}, ..., G_t^{(n)}, ...\}$, its discharge as $D^{(n)} = \{D_1^{(n)}, ..., D_t^{(n)}, ...\}$, and precipitation as $P^{(n)} = \{P_1^{(n)}, ..., P_t^{(n)}, ...\}$. Then river forecasting has (1) input (predictors): $\{G^{(\mathbb{N})}_{t_c-t_h+1}, ..., G^{(\mathbb{N})}_{t_c}\}$, $\{D^{(\mathbb{N})}_{t_c-t_h+1}, ..., D^{(\mathbb{N})}_{t_c}\}$ and $\{P^{(\mathbb{N})}_{t_c-t_h+1}, ..., P^{(\mathbb{N})}_{t_c}\}$, where $\mathbb{N} = \{1, 2, ..., N\}$, t_c is current time, and t_h is history window size; and (2) output (targets): $\mathrm{E}(G^{(n)}_{t_c+t_l}|\mathcal{F}^{(\mathbb{N})}_{t_c})$ and $\mathrm{E}(D^{(n)}_{t_c+t_l}|\mathcal{F}^{(\mathbb{N})}_{t_c})$, where $n \in \mathbb{N}$, t_l is *lead time*, *i.e.* how long in the future we are predicting, and $\mathcal{F}^{(\mathbb{N})}_{t_c}$ is the accumulated information up to and including time t_c for nodes $\mathbb{N}$.

4.1.2 Classification Models. Classifiers solve the problem: $argmax_{y \in \mathbb{C}} P(y^{(d)}|\vec{x})$, where $\vec{x}$ is the predictor, $y^{(d)}$ is a discrete variable denoting a class, the superscript d indicates that the target y is a discrete variable, and $\mathbb{C}$ is the set of classes.

A regression problem can be transformed into a classification problem by discretizing the continuous target variable y into multiple levels. Let us denote the number of levels as K. Then, the range of y is split into K bins, with K centroids. Classification models produce an estimation of $P(y^{(d)}|\vec{x})$. Discriminative models, including ANNC, Multinomial Logistic Regression (MLR) and Random Forest (RF), first compute K scores for the K classes, and then output the class with the highest score. The normalized K scores estimate $P(y^{(d)}|\vec{x})$ over $y^{(d)} \in \mathbb{C}$ where $|\mathbb{C}| = K$. Generative models including PGMs first compute $\hat{P}(y^{(d)}, \vec{x})$ for all $y^{(d)} \in \mathbb{C}$, and then output $argmax_{y^{(d)} \in \mathbb{C}} \hat{P}(y^{(d)}, \vec{x})$. Since $\hat{P}(y^{(d)}|\vec{x}) = \frac{\hat{P}(y^{(d)}, \vec{x})}{P(\vec{x})}$, the normalized $\hat{P}(y^{(d)}, \vec{x})$ is an estimation of $P(y^{(d)}|\vec{x})$.

To predict $\mathrm{E}(y|\vec{x})$, instead of simply using the centroid of a bin [20][6], we leverage the estimated $\hat{P}(y^{(d)}|\vec{x})$, and compute the weighted average of bin centroids: $\hat{\mathrm{E}}(y|\vec{x}) = \sum_{i=1}^{K} \hat{P}(y^{(d)} = i|\vec{x}) B_i$, where B_i is the centroid of the i-th bin.

To predict the $p(y|\vec{x})$, two approaches are adopted: (1) maximum likelihood estimation (MLE) by fitting a pre-assumed type of distribution [17], and (2) kernel smoothing (KNS) which predicts $p(y|\vec{x})$ by: $\hat{p}(y|\vec{x}) = \frac{1}{h} \sum_{i=1}^{K} \hat{P}(y^{(d)} = i|\vec{x}) \mathrm{F}(\frac{y - B_i}{h})$, where h is a factor determining smoothness, and $\mathrm{F}(\cdot)$ is the kernel function with constraint $\mathrm{E}_y(\mathrm{F}(y)) = 0$ and $\int_{-\infty}^{\infty} \mathrm{F}(y)dy = 1$. KNS works like kernel density estimation, but uses a discrete distribution instead of data samples to predict the continuous distribution [17].

4.2 Probabilistic Graphical Models

4.2.1 Graph Structure. A PGM uses a graph structure to describe the dependency of different variables. Since the river topology naturally defines the node dependencies, the graph structure

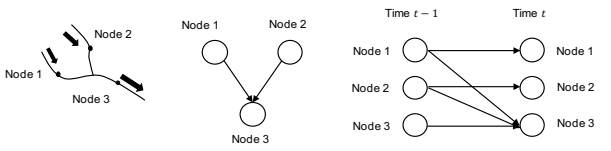

Figure 2: Left: river topology for node 1-3. Middle: a concise structure of node dependency. Right: node dependency extended with a time dimension.

simply follows this topology. Note that for almost all the rivers, there is a constant direction of the flows, and there are no cycles, which makes a directed acyclic graph (DAG) assumption reasonable. In this paper, a directed probabilistic graphical model is considered. A simple example is shown in Figure 2.

4.2.2 Neighbor Smoothing. Training PGMs requires estimating the conditional probability distribution (CPD) of target variables, given each combination of the predictor values. The CPD is estimated by counting the number of corresponding instances in the training set, and storing them in a look-up table. The look-up table can be very sparse yet in the worst case, its size grows exponentially with predictors and discrete levels.

To address the sparsity-based overfitting, we propose a neighbor smoothing (NBS) approach which adds to each row with its "neighbor" rows in the inference phase. To define neighbors, we first define the distance between two condition vectors in the look-up table $C_1 : (X_1 = x_1, X_2 = x_2, ..., X_m = x_m)$ and $C_2 : (X_1 = x'_1, X_2 = x'_2, ..., X_m = x'_m)$ as $D(C_1, C_2) = \sum_{i=1}^{m} |x_i - x'_i|$, where m is the number of predictors. Two conditions are defined as h-hop neighbors if their distance is h. NBS estimates CPD by: $P(Y|\vec{X} = \vec{x}) = \frac{1}{Z} \sum_{h=0}^{H} \sum_{\vec{x'} \in \mathbb{V}_h(\vec{x})} \alpha_h P(Y|\vec{X} = \vec{x'})$, where $\mathbb{V}_h(\vec{x})$ is the set of h-hop neighbors of $\vec{x}$, H is the pre-defined maximum number of hops, α_h is a decay factor for h-hop neighbors ($0 < \alpha_{h+1} < \alpha_h < 1$), and Z is a normalizing coefficient used to ensure that the sum of $P(Y|\vec{X} = \vec{x})$ over all Y values is 1.

The intuition behind NBS is that it allows to increase the size of training data by Gibbs sampling with a small probability of distortion. By increasing the ratio of training size over parameter amount, overfitting is alleviated.

4.3 ANNC and ANNR

After discretizing targets of the training set, an ANNC is trained using the backpropagation algorithm, with K output neurons where K is the number of bins of the target variable. The label of each instance in the training set is a vector of length K, where only one of the K values is 1, indicating the correct bin. Since the outputs of softmax function are considered as a proxy of probability distribution of the target variable [19][9], we select the softmax function as the activation function of the output layer. In the inference phase, the K values of the output neurons are used as an estimation of the CPD of the K classes, i.e. $\hat{P}(y^{(d)}|\vec{x})$. Different from ANNC, the architecture of an ANNR has only one output neuron, and no activation function for the output layer (or equivalently, a linear activation function). ANNR is also trained with backpropagation algorithm [18].

4.4 Metrics

River models can help with two types of questions: (1) what is the expected target value, and (2) what is the confidence interval or the probability that a given target feature falls in some range. Correspondingly, R squared (R^2) and mean log-likelihood (MLL) are adopted to assess the prediction performance.

R^2 measures the relative distance between true and predicted values. $R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - f(\vec{x_i}))^2}{\sum_{i=1}^{N}(y_i - \bar{y_i})^2}$, where $f(\cdot)$ is the model function that predicts $\mathrm{E}(y_i|\vec{x_i})$, $\vec{x_i}$ is the input of i-th instance in the testing set, y_i is the true target value of i-th instance, and N is the size of testing set.

MLL assesses the quality of predicted probability distribution. $MLL = \frac{1}{N} \sum_{i=1}^{N} \log \hat{p}(y_i|\vec{x_i})$, where N is the size of testing set, $\hat{p}(y|\vec{x_i})$ is the predicted CPD for the target value of the i-th instance, and y_i is the true target value of the i-th instance.

5 EXPERIMENT

5.1 Setup

The experiment is conducted on Guadalupe river described in section 3. The workflow of experiments follows the commonly accepted procedure: (1) data preprocessing, (2) model calibration, and (3) model validation [8].

Five models are compared: last-value forward (LVF), stepwise multiple linear regression (SWMLR), ANNR, PGM, and ANNC. Their setup configurations are shown in Figure 3. LVF simply uses current value as a prediction for future. SWMLR is a linear model with varying history steps [14]. ANNR and ANNC both have one hidden layer with sigmoid activation function, and hidden neurons twice the number of input neurons [16]. ANNR has one output neuron with linear activation function, while ANNC has K output neurons with softmax function where K is the discrete levels. Five-fold cross validation is adopted. The average values for R^2 and MLL over five folds are reported for each model.

Model	Input	Output	History Steps[1] t_h	Linearity	Number of Parameters
LVF	$G_{t_c}^{(n)}, D_{t_c}^{(n)}$	$G_{t_c+t_l}^{(n)}$, $D_{t_c+t_l}^{(n)}$	1	Linear	0
SWMLR	$\{G_{t_c-t_h+1'}^{(\mathbb{N}_n)}, ..., G_{t_c}^{(\mathbb{N}_n)}\}$, $\{D_{t_c-t_h+1'}^{(\mathbb{N}_n)}, ..., D_{t_c}^{(\mathbb{N}_n)}\}$, $\{P_{t_c-t_h+1'}^{(\mathbb{N}_n)}, ..., P_{t_c}^{(\mathbb{N}_n)}\}$		3	Linear	16-25
ANNR			5	Non-linear	1351-3361
PGM			1	Non-linear	>1M[2]
ANNC			5	Non-linear	1810-4090

Figure 3: Model setup.[1]History steps are selected by stepwise testing for SWMLR, ANNR and ANNC. It is set to 1 for PGM due to large memory requirements.[2]PGMs have an exponential number of parameters, but more than 99.9% are zero. Parameters for each target variable vary with the number of neighbor nodes and available features.

5.2 Results

Different models are compared in Figure 4. The models are shown in the format "(model name)-(t_h)", where t_h is the history window size. For SWMLR, ANNR, and ANNC, t_h is selected by stepwise testing, while t_h is set to 1 for PGM due to the exponentially increase in number of parameters. KNS is used to estimate distribution.

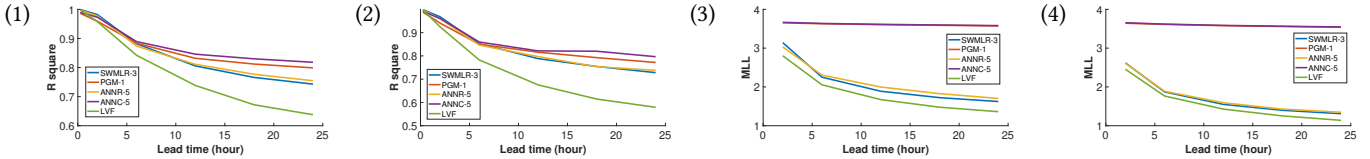

Figure 4: R square and MLL results of five models. (1) and (3) are results of gage forecasting; (2) and (4) are results of discharge forecasting.

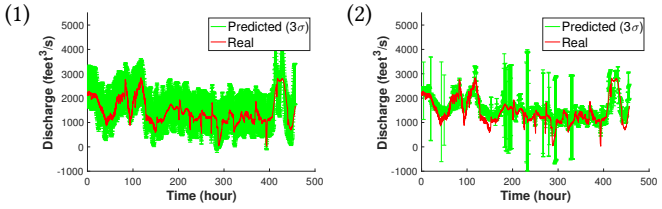

Figure 5: Predicted result vs. true values for one node with (1) SWMLR and (2) PGM. The green bar shows the three-standard-deviation range.

Classification models are better for long term forecasting than regression. (*i*) In terms of R^2, classification models are better at long-term forecasting while regression models work better in immediate future, with a cutoff of six hours. Since longer-term forecasting models more complicated relation between predictors and targets, classification models can capture the causal effects better. Besides, classification models can mitigate effects of noise by discretization, while regression models are more prone to extreme noise. However, for shorter-term gage and discharge prediction, even the LVF model has better performance than classification models which lose accuracy due to quantization. (*ii*) When MLL is considered, classification models always work better than regression models. The assumption made by regression models that noise should have stationary mean and variance does not always hold. However, classification models estimate distributions according to the inputs provided. Figure 5 shows the three-standard-deviation range of the prediction results for a node's discharge using SWMLR and PGM. Except for a few points, PGM captures the true value with higher confidence than SWMLR. Since PGM predicts the variance based on inputs while SWMLR predicts a constant variance, in general PGM works better than SWNLR. The few outlier points with high variance in the PGM graph correspond to time steps for which the input training data also has very high variance.

6 CONCLUSIONS

In this paper, we compare regression and classification models for river gage and discharge forecasting. MLL is introduced as a metric to assess the estimated probability distribution, while R^2 is used to assess the expected prediction. Experiment results on real data for Guadalupe river (Texas) show that classification models always work better in terms of MLL. For R^2, regression models work better for shorter-term predictions, while classification models are better for longer-term predictions. To discretize continuous variables, K-means works better than linear and density-based approaches. To estimate continuous distribution using a discrete one, KNS is superior to MLE. The methodology can be extended to include additional predictors relevant for multi-modal renewable energy generation besides hydropower (*e.g.*, wind or solar) or to other domains where time series forecasting is of interest.

7 ACKNOWLEDGEMENT

This work was supported in part by the US National Science Foundation (NSF) under CyberSEES Grant CCF-1331804.

REFERENCES

[1] *http://maps.waterdata.usgs.gov/mapper/index.html.*
[2] *https://en.wikipedia.org/wiki/Guadalupe_River_(Texas).*
[3] *https://www.arcgis.com/.*
[4] Jeffrey G Arnold, Daniel N Moriasi, Philip W Gassman, Karim C Abbaspour, Michael J White, Raghavan Srinivasan, Chinnasamy Santhi, RD Harmel, Ann Van Griensven, Michael W Van Liew, et al. 2012. SWAT: Model use, calibration, and validation. *Transactions of the ASABE* 55, 4 (2012), 1491–1508.
[5] Shahrokh Asadi, Jamal Shahrabi, Peyman Abbaszadeh, and Shabnam Tabanmehr. 2013. A new hybrid artificial neural networks for rainfall–runoff process modeling. *Neurocomputing* 121 (2013), 470–480.
[6] Stamatia Bibi, Grigorios Tsoumakas, Ioannis Stamelos, and I Vlahavas. 2008. Regression via Classification applied on software defect estimation. *Expert Systems with Applications* 34, 3 (2008), 2091–2101.
[7] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees.* CRC press.
[8] CW Dawson and RL Wilby. 2001. Hydrological modelling using artificial neural networks. *Progress in physical Geography* 25, 1 (2001), 80–108.
[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
[10] Kuo-lin Hsu, Hoshin Vijai Gupta, and Soroosh Sorooshian. 1995. Artificial neural network modeling of the rainfall-runoff process. *Water resources research* 31, 10 (1995), 2517–2530.
[11] RK Kachroo. 1992. River flow forecasting. Part 1. A discussion of the principles. *Journal of Hydrology* 133, 1-2 (1992), 1–15.
[12] Jordan D Kern, Gregory W Characklis, Martin W Doyle, Seth Blumsack, and Richard B Whisnant. 2011. Influence of deregulated electricity markets on hydropower generation and downstream flow regime. *Journal of Water Resources Planning and Management* 138, 4 (2011), 342–355.
[13] Roman Krzysztofowicz. 2001. The case for probabilistic forecasting in hydrology. *Journal of hydrology* 249, 1 (2001), 2–9.
[14] William M Mendenhall and Terry L Sincich. 2016. *Statistics for Engineering and the Sciences.* CRC Press.
[15] JD Salas, JR Delleur, V Yevjevich, and WL Lane. 1980. Applied modeling of hydrologic time series, Water Resor. *Pub., Littleton, CO, USA* (1980).
[16] K Gnana Sheela and Subramaniam N Deepa. 2013. Review on methods to fix number of hidden neurons in neural networks. *Mathematical Problems in Engineering* 2013 (2013).
[17] Bernard W Silverman. 1986. *Density estimation for statistics and data analysis.* Vol. 26. CRC press.
[18] Donald F Specht. 1991. A general regression neural network. *IEEE transactions on neural networks* 2, 6 (1991), 568–576.
[19] Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction.* Vol. 1. MIT press Cambridge.
[20] Luís Torgo and Joao Gama. 1996. Regression by classification. *Advances in artificial intelligence* (1996), 51–60.