
Nian Zhang\(^1\), Roussel Kamaha\(^2\), and Pradeep Behera\(^3\)

\(^1\)University of the District of Columbia, Department of Electrical and Computer Engineering
4200 Connecticut Ave. NW, Washington, DC, 20008, USA
nzhang@udc.edu, roussel.kamaha@udc.edu

\(^2\)University of the District of Columbia, Department of Civil Engineering
4200 Connecticut Ave. NW, Washington, DC, 20008, USA
pbehera@udc.edu

Abstract

In this research, we developed a predictive model based on least squares support vector machine (LS-SVM) that forecasts the future streamflow discharge using the past streamflow discharge data. A Gaussian Radial Basis Function (RBF) kernel framework was built on the data set to tune the kernel parameters and regularization constants of the model with respect to the given performance measure. The 10-fold cross-validation is used as a cost function for estimating the performance of the model. The training process of LS-SVM was designed to train the support values and the bias term of an LS-SVM for function approximation. After the network has been well trained, we test the prediction performance on the new testing samples, as well as the training samples. The USGS real-time streamflow data were used as time series input. The experimental results showed that the proposed LS-SVM algorithm is a reliable and efficient method for streamflow prediction, which has an important impact to the water resource management field.

Keywords: Water Quantity Prediction, Least Squares Support vector Machine.

1. Introduction

Land development activities inevitably change watershed conditions, primarily due to an increase in the impervious area through paving, construction, drainage systems and removal or alteration of vegetation which results in water quantity and quality problems for local receiving bodies. The examples of water quantity problems include flooding and stream bank erosion, while examples of water quality problems include pollution loading and receiving water impairments. The impact of this type of activity is more pronounced for highly urbanized areas and the associated receiving waters such as the Potomac River and Anacostia River within the Chesapeake Bay Area watershed. In addition, it has been recognized that climate change can have severe impacts on our streams and rivers due to extreme weather events such as frequent flooding. In this regard, reliable estimation of streamflow at various locations is very important from the water resources management viewpoint. Engineers, water resources professionals, and regulatory authorities need this streamflow information for planning, analysis, design, and operation & maintenance of water resources systems (e.g., water supply systems, dams, and hydraulic structures). Currently USGS provides the streamflow data at various locations in the form of gage height and discharge volume at specific locations, and we used this input to design a reliable prediction model.

The study area is focused on the Potomac River watershed, as shown in Fig. 1. The Potomac River is one of the least dam-regulated large river systems in the eastern United States [1]. The Potomac River has the highest level of nitrogen and the third highest level of phosphorus loading of all the major rivers in the Chesapeake Bay watershed. These nutrients can limit the growth of submerged aquatic vegetation, cause low oxygen conditions and create dead zones.

Fig. 1. Potomac River Watersheds. Of approximately 10,000 stream miles assessed in the watershed, more than 3,800 miles were deemed “threatened” or “impaired”.
of LS-SVM predictions on the water data are demonstrated. In Section 4, the conclusions are given.

2. GENERAL METHODOLOGY

2.1 Modeling and Prediction with NARX and NAR Model with Time-Delay

In time series modeling, a nonlinear autoregressive exogenous model (NARX) is a nonlinear autoregressive model which has exogenous inputs. Extending backward from time \( t \), we have time series \((x(t), x(t-1), x(t-2), \ldots)\) and time series \((y(t), y(t-1), y(t-2), \ldots)\). The predictive model can be represented mathematically by predicting future values of the time series \( y(t) \) from past values of that time series and past values of the precipitation time series \( x(t) \) that influence \( y(t) \), as shown in Fig. 2. In the model we can observe the exogenous variables \( x(t) \), which influence the values of the time series \( y(t) \). The time series \( y(t) \) is the one that we want to predict.

This form of prediction can be written as follows:

\[
y(t + s) = f(y(t), y(t - 1), \ldots, y(t - d), x(t), x(t - 1), \ldots, x(t - d))
\]

where \( s \) is called the horizon of prediction. Assume we predict one step ahead, \( s \) will be \( 1 \). \( y(t) \) are the past predicted values by the model, \( d \) is the time delay, \( x(t) \) are the exogenous variables and \( f \) is a nonlinear function.

This paper is organized as follows. In Section 2, the modeling and prediction with NARX and NAR Model with time-delay is briefly introduced. The Least Squares Support Vector Machines (LS-SVM) method is illustrated in detail. The practical implementation is introduced. In Section 3, the training data and time delays are depicted. The training method is designed. The model predict future values on the testing data, as well as the training data. The experimental results

...
exogenous variables are time series as well. There could be none, one, or more exogenous variables. The y are the past predicted values. Because we want to predict the value at the current time t, we can use values starting from \(t - 1\) to \(t - d\), where \(d\) is the number of past predictions fed into the model.

2.2 Least Squares Support Vector Machine Regression

Support Vector Machines (SVMs) are a powerful kernel based statistical learning methodology for solving problems of nonlinear classification, pattern recognition and function estimation [5]. Least Squares Support Vector Machines (LS-SVM) are an advanced version of the standard SVMs which incorporates unsupervised learning and recurrent networks. Recent developments of LS-SVM are especially relevant to the fields of time series prediction, kernel spectral clustering, and data visualization [6]-[16]. The preliminary results show that the LS-SVM modeling method is a promising method for time series prediction, and because time series prediction is a generalized form of runoff quantity prediction, we expect the LS_SVM method will also work the best for the runoff prediction problem. The following are a brief introduction to the Support Vector Machines and Least Squares Support Vector Machines.

Support Vector Machines are a new and potential data classification and regression method. The basic idea of SVM is based on Mercer core expansion theorem which maps sample space to a high dimension or even unlimited dimension feature space (Hilbert space) via nonlinear mapping \(\phi\). And it will boil algorithm which searches for optimal linear regression hyper plane down to a convex programming problem of solution of a convex restriction condition. And it will also obtain overall situation optimum solution so as to use the method of linear learning machine in feature space to solve the problem of high-degree nonlinear regression in sample space [17].

The principles of SVM can be summarized by Fig. 3 as follows:

In Fig. 3, N input support vectors are in the first layer and the second layer is nonlinear operation of N support vectors, that is, the core operation. For nonlinear problems, assume sample to be n-dimension vector, then in one certain domain, N samples and their values can be expressed as:

\[
(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \in \mathbb{R}^n \times \mathbb{R}
\]  

(1)

Firstly, a nonlinear mapping \(\psi(\cdot)\) is used to map samples from former space \(\mathbb{R}^n\) to feature space:

\[
\psi(x) = (\phi(x_1), \phi(x_2), \ldots, \phi(x_n))
\]  

(2)

Then, in this high-dimension feature space, optimal decision function:

\[
y(x) = w\phi(x) + b
\]  

(3)

is established. In this function, \(w\) is a weighed value vector and \(b\) is a threshold value. In this way, nonlinear prediction function is transformed to linear prediction function in high-dimension feature space.

As development and improvement of classical SVM, Least Squares Support Vector Machine (LSSVM) defines a cost function which is different from classical SVM and changes its inequation restriction to equation restriction. As a result, the solution process becomes a solution of a group of equations which greatly accelerates the solution speed [18]. In Least Squares Support Vector Machines, the problem of optimization is described as follows:

\[
\min_{w, b, \varepsilon} L(w, b, \varepsilon) = \frac{1}{2} \|w\|^2 + \frac{\varepsilon}{2} \sum_{i=1}^{t} \varepsilon_i^2
\]  

(4)

Such that: \(y_i = w^T\phi(x_i) + b + \varepsilon_i (i=1, 2, \ldots, t)\)

The extreme point of \(Q\) is a saddle point, and differentiating \(Q\) can provide the formulas as follows, using Lagrangian multiplier method to solve the formulas:

\[
\frac{\partial Q}{\partial w} = w - \sum_{i=1}^{t} \alpha_i \phi(x_i) = 0
\]  

(5)

\[
\frac{\partial Q}{\partial b} = -\sum_{i=1}^{t} \alpha_i = 0
\]

\[
\frac{\partial Q}{\partial \varepsilon} = w^T \phi(x_i) + b + \varepsilon_i - y_i = 0
\]
\[
\frac{\partial Q}{\partial \epsilon_i} = C \epsilon_i - \alpha_i = 0
\]

From formulas above:
\[
\frac{1}{2} \sum_{i=1}^{l} \alpha_i \phi(x_i) \sum_{j=1}^{l} \alpha_j \phi(x_j) + \frac{1}{2C} \sum_{i=1}^{l} \alpha_i^2 + b \sum_{i=1}^{l} \alpha_i = \sum_{i=1}^{l} \alpha_i y_i
\]

The formula above can be expressed in matrix form:
\[
\begin{bmatrix}
0 \\
e^T
\end{bmatrix}
(1 + 1)(1 + 1)
\begin{bmatrix}
b \\
\alpha
\end{bmatrix} =
\begin{bmatrix}
0 \\
y
\end{bmatrix}
\]

In this equation,
\[
e = [1, \ldots, 1]^T
\]
\[
\Omega_{ij} = K(x_i, x_j) = \phi(x_i)^T \phi(x_j)
\]

Formula (7) is a linear equation set corresponding to the optimization problem and can provide us with \(a\) and \(b\). Thus, the prediction output decision function is:
\[
\hat{y}(x) = \sum_{i=1}^{l} \alpha_i K(x_i, x) + b
\]

where \(K(x_i, x)\) is the core function.

We are ultimately using the LS-SVM method to calculate and predict the USGS water data, specifically using time-series data prediction. After loading the data into Matlab, we first build the training and testing sets from the data. Next we cross-validate based upon a feed-forward simulation on the validation set using a feed-forwardly trained model. This will supply us with the tuning parameters: \(\gamma\) (gam) which is the regularization parameter and \(\sigma^2\) (sig2) or the squared bandwidth. The tuning parameters were found by using a combination of coupled simulated annealing (CSA) and a standard simplex method. The CSA finds good starting values and these were passed to the simplex method in order to fine tune the result. One of the parameters, \(\gamma\) is the regularization parameter, determining the trade-off between the training error minimization and smoothness. The other parameter, \(\sigma\) represents the squared bandwidth. Once the parameters are calculated, we are able to plot the function estimation or use the prediction function to predict future values of the data. By using only a subset of the total data available, we can compare the predictions against real values to see how accurate the prediction is.

2.3 Practical Implementation

The training process of LS-SVM involves the selection of kernel parameter, sig2 and the regularization constant, gam. A good choice of these parameters is crucial for the performance of the estimator. In this paper, we use 10-fold cross-validation for selecting these parameters. Another important choice is the selection of regressors, i.e., which lags of inputs and outputs are going to be included in the regression vector. This selection is done by using a large number of initial components and then performing a greedy search to prune non-informative lags on a cross-validation basis. Therefore an initial model containing all regressors is estimated and optimal choices for the parameters are made. On each stage of the greedy backwards elimination process, a regressor is removed if the cross-validation mean absolute error or mean squared error improves. The final set of regressors is then used for the final predictions. For the purpose of model estimation, all series are normalized to zero mean and unit variance.

3. EXPERIMENTAL RESULTS

3.1 USGS Time Series Data

The study area will focus on the Four Mile Run at Alexandria, VA. The Four Mile Run is 9.2 miles long, and is a direct tributary of the Potomac River, which ultimately carries the water flowing from Four Mile Run to the Chesapeake Bay. The stream passes from the Piedmont through the fall line to the Atlantic Coastal Plain, and eventually empties out into the Potomac River. Potomac River was determined to be one of the most polluted water bodies in the nation mainly due to the CSOs and stormwater discharges and wastewater treatment plant discharges. In addition, because of the highly urbanized nature of the Four Mile Run watershed, the neighborhoods and businesses adjacent to this portion of the run were subjected to repeated flooding, beginning in the 1940s. Therefore, the flood-control solutions are the major concern. Runoff prediction would provide a promising solution for flood-control.

The real-time USGS data for the Four Mile Run station include the discharge data, which is useful for investigating its impact to the long-run discharge forecast. The discharge is the volume of water flowing past a certain point in a water-flow. For example, the amount of cubic feet passing through a drain per second is a measure of discharge. The discharge data was retrieved for 120 days between August 28, 2010 and December 4, 2010. Because the real-time data typically are recorded at 15-minute intervals, the runoff discharge (cubic feet per second) data plots 34721 data during the 120 days, as shown in Fig. 8. The discharge will be presented to the
system as an input. It is a 34721x1 vector, representing dynamic data, i.e. 34721 time steps. It is challenging that these discharge values vary significantly over time. As shown in Fig. 8, the baseline is at around 4 on the Y-axis, with peaks reaching 8, with very little repetition to the pattern, making it more difficult to predict future values.

At this stage, we are only looking to input one variable into the LS-SVM algorithm, but in the future it would probably prove to increase prediction accuracy to include the use of additional variables at once. For example, gage height is one of the useful variables [20]. Gage height is simply the height of water at a certain point, like the level of the Potomac River measured at Key Bridge. The more data input into the system often translates into better results. The gage height is an even more varied set, as seen in Fig. 9.

The gage height plot contains peaks in similar timeframes as the discharge plot, likely due to large rainfall events or local flooding. The discharge data and gage height data have strong correlation with each other, as shown in Fig. 10.

The discharge data is multiplied by 100 so it is visible on the same scale. This is likely due to the patterns in local weather, specifically precipitation. Sending only one of these variables to the LS-SVM function will produce good predictions, but if we can go further and implement a two input algorithm, which will analyze both discharge and gage height at the same time, this will definitely increase the accuracy of predictions. Again at this point, we only use the discharge data as the input to the model.

3.2 Training Data and Time Delays

The first 500 time series data from the original sample of about 34,721 were used for our analysis. To determine an appropriate time delay or lag, we increase the number of delays lags until the network performed well. After a number of experiments, 80 is determined to be the smallest lag number that ensures a good performance. That means the model will use the past 80 input data to predict a future data.

Before parameter tuning and network training, we should use the function windowize to convert the time-series into a Hankel matrix useful for training a nonlinear function approximation. For example, assume there is a matrix A which is defined below.

\[
A = \begin{bmatrix}
a_1 & a_2 & a_3 \\
b_1 & b_2 & b_3 \\
c_1 & c_2 & c_3 \\
d_1 & d_2 & d_3 \\
e_1 & e_2 & e_3 \\
f_1 & f_2 & f_3 \\
g_1 & g_2 & g_3
\end{bmatrix}
\]

Now we want to convert matrix A to a new matrix W by running the Matlab command: \( W = \text{windowize}(A, \)}
In the form of Hankel matrix after the conversion. We define \( W \) as \[
W = \begin{bmatrix}
a_1 & a_2 & a_3 & b_1 & b_2 & b_3 & c_1 & c_2 & c_3 \\
b_1 & b_2 & b_3 & c_1 & c_2 & c_3 & d_1 & d_2 & d_3 \\
c_1 & c_2 & c_3 & d_1 & d_2 & d_3 & e_1 & e_2 & e_3 \\
d_1 & d_2 & d_3 & e_1 & e_2 & e_3 & f_1 & f_2 & f_3 \\
e_1 & e_2 & e_3 & f_1 & f_2 & f_3 & g_1 & g_2 & g_3
\end{bmatrix}
\]
In our case, \( Xu = \text{windowize}(X,1:\text{lag}+1) \) will convert the discharge data set into a new input including the past measurements and the future output by \text{windowize}. For the 500 data points and 80 lags, it will generate 420 rows and 81 columns. The last column of the resulting matrix \( Xu \) contains the future values of the time-series, and the previous 80 columns contain the past inputs. Fig. 11 shows the new data set in the form of Hankel matrix after the conversion.

![Fig. 11 New data set in the form of Hankel matrix after the windowize conversion.](image)

Next we separate the data set into training and testing set. The first 340 data points will be used as training data, and the remaining 160 data will be testing data. \( Xtra = Xu(1:\text{end-lag},1:\text{lag}) \) will generate 80 past inputs, i.e. \( x(t-1), x(t-2), \ldots, x(t-80) \), while \( Ytra = Xu(1:\text{end-lag},\text{end}) \) contains their actual future value, \( x(t) \). \( Ytra \) will be used as the target for those past inputs.

### 3.3 Tuning the Parameters

In order to build an LS-SVM model, we need to tune the regularization constant, \( gam \) and the kernel parameter, \( sig2 \). \( gam \) determines the trade-off between the training error minimization and smoothness. In the common case of the Gaussian RBF kernel, the kernel parameter, \( sig2 \) is the squared bandwidth. We use \( \text{[gam,sig2]} = \text{tunelsvm}([Xtra,Ytra,'f',[],[],'RBF_kernel'],... 'simplex','crossvalidatelsvm','[10,'mae']) \) to tune these parameters. ‘F’ stands for function estimation. The Kernel type is chosen to be the default RBF kernel. The optimization function is specified as simplex. The simplex is a multidimensional unconstrained non-linear optimization method. Simplex finds a local minimum of a function starting from an initial point \( X \). The local minimum is located via the Nelder-Mead simplex algorithm [21]. The model adopts \text{crossvalidatelsvm} as the cost function. It estimates the generalization performance of the model. It is based upon feedforward simulation on the validation set using the feedforwardly trained model. In addition, 10 means 10-fold. We use 10-fold cross-validation because the input size is greater than 300 points. Otherwise, leave-one-out cross-validation will be used when the input size is less or equal than 300 points. The 10-fold cross-validation method will break data (the size of the data is assumed to be \( n \)) into 10 sets of size \( n/10 \), then train on 9 datasets and test on 1, and then repeat 10 times and take a mean accuracy. \( mae \) is the mean absolute error and is used in combination with the 10-fold cross-validation method. It is the absolute value of the difference between the forecasted value and the actual value. It tells us how big of an error we can expect from the forecast on average.

The tuning of the parameters is conducted in two steps. First, a state-of-the-art global optimization technique, Coupled Simulated Annealing (CSA) [22], determines suitable parameters according to some criterion. Second, these parameters are then given to a second optimization procedure \text{simplex} to perform a fine-tuning step. The parameter tuning results are shown in Fig. 11. Coupled Simulated Annealing chosen the initial \( gam \) to be 1364.706, and \( sig2 \) to be 13.989. They serve as the starting values for the \text{simplex} optimization routine. After 11 iterations, the \( gam \) and \( sig2 \) are optimized to be 83.2188 and 15.298, respectively.

Copyright (c) 2015 Advances in Computer Science: an International Journal. All Rights Reserved.
The first 340 samples are training data, and the remaining 160 samples are testing data. As shown in Fig. 13, the prediction on the training data matches the actual values perfectly. This makes sense because these training samples have been seen by the network during training. The prediction on these data should have already been trained to be very close to the actual value. In addition, when we test the new data from time step 341 to 500, we find the predicted values match very well with the actual values. This demonstrated that the LSSVM model has an excellent prediction ability.

### 4. Conclusions

In this paper, the least squares support vector machine (LS-SVM) based algorithm is developed to forecast the future streamflow based on the previous streamflow. The first 340 data points are used as training data, and the remaining 160 data are testing data. First we convert the time-series into a Hankel matrix useful for training a nonlinear function approximation. Next we build an LS-SVM model by tuning the regularization constant, \( \text{gam} \) and the kernel parameter, \( \text{sig}^2 \). A Gaussian Radial Basis Function (RBF) kernel framework was built on the data set to optimize the tuning parameters. The 10-fold cross-validation method is used to estimate the generalization performance of the model. Then we train the LSSVM network. It trains the support values and the bias term of an LS-SVM for function approximation. We developed an effective training scheme. After the network has been well trained, we test the prediction performance by predicting new values on the testing samples, as well as the training samples. The excellent experimental results demonstrated that the proposed LS-SVM based...
predictive model has superior prediction performance on not only the training samples, but also the testing samples. In addition, the proposed parameter tuning method and the training scheme worked effectively, which ensure an accurate prediction of streamflow.

Moreover, the proposed approach provides an excellent prediction method for any time series data, and if correctly implemented can be an invaluable tool in predicting natural weather events. Even outside of storm-water, this algorithm could be very useful to researcher or engineers who wish to develop a resource efficient prediction model for any quantifiable data set, i.e. climate change, solar radiation, global warming, glacier melting, and more.

Acknowledgments

The authors would like to express thanks to the University of the District of Columbia STEM Center (NSF/HBCU-UP/HRD-0928444) grant and DC Water Resources Research Institute (WRRI) Grant.

References


Dr. Nian Zhang received her B.S. in Electrical Engineering at the Wuhan University of Technology, M.S. in Electrical Engineering from Huazhong University of Science and
Technology, and Ph.D. in Computer Engineering from Missouri University of Science and Technology. She is an Assistant Professor of the Department of Electrical and Computer Engineering at the University of the District of Columbia, Washington DC. Dr. Zhang's research expertise and interests include support vector machines, neural networks, fuzzy logic, computational intelligence methods, and their applications on pattern recognition, signal and image processing, time series prediction, renewable energy, biomedical engineering, and autonomous robot navigation.

Roussel Kamaha is a senior student at the University of the District of Columbia in the Department of Electrical and Computer Engineering. He has a strong interest in renewable energy, time series prediction, biomedical engineering, and machine learning.

Dr. Pradeep Behera received his Ph.D. degree from the Civil & Environmental Engineering, University of Toronto, Canada. He is the Professor and Chair of the Department of Civil Engineering at the University of the District of Columbia, Washington DC. His research interests include Urban Stormwater Management, Non-point Source Pollution, Water Resources Engineering, Erosion and Sediment Control, Sustainable Urban Water Systems, Environmental Systems, and Spatio-Temporal Informatics.