Postprint: Runoff Forecasting Model Using PSO-SA Hybrid Optimized Support Vector Regression

Authors: Jiang Linli, Li Jie, Wu Jiansheng

Date: 2018-05-24T00:00:00+00:00

Abstract

To effectively improve the accuracy of runoff forecasting, this paper proposes an effective fusion optimization strategy that employs a hybrid method combining particle swarm and simulated annealing algorithms to simultaneously optimize the kernel function type and kernel parameters of support vector regression, thereby establishing an effective hybrid-optimized support vector regression runoff forecasting model. The proposed method provides an effective approach for kernel function selection and parameter optimization. Through case analysis of the Liujiang River runoff in Liuzhou, Guangxi, and comparison with the pure support vector regression model, the research results demonstrate that the model exhibits stable prediction performance, possesses high generalization capability and prediction accuracy, and provides an effective predictive method for runoff forecasting.

Full Text

Preamble

Title: Runoff Forecasting Based on Hybrid Optimized Support Vector Regression Using PSO-SA

Authors: Jiang Linli[1], Li Jie[1], Wu Jiansheng[1,2]

Affiliations: 1. School of Mathematics and Computer Science, Guangxi Science & Technology Normal University, Laibin, Guangxi 546199, China 2. School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China

Abstract: To effectively improve the accuracy of runoff forecasting, this paper proposes an effective hybrid optimization strategy that simultaneously optimizes both the kernel function type and kernel parameters of Support Vector

Regression (SVR) using a hybrid method combining Particle Swarm Optimization (PSO) and Simulated Annealing (SA). This establishes an effective hybrid optimization SVR runoff forecasting model. The proposed method provides an effective approach for kernel function selection and parameter optimization. Through case analysis of the Liujiang River runoff in Liuzhou, Guangxi, and comparison with pure SVR models, the results demonstrate that the proposed model exhibits stable prediction performance with high generalization capability and accuracy, offering an effective predictive method for runoff forecasting.

Keywords: support vector regression; particle swarm optimization; simulated annealing; fusion improvement; runoff forecast model

0 Introduction

Accurate and timely rainfall-runoff forecasting is critically important for water resource management, as effective predictions enable advanced preparedness and help prevent catastrophic accidents involving loss of life and economic damage [1]. However, developing a robust and accurate rainfall-runoff model remains a formidable challenge in hydrology due to various nonlinear influencing factors such as rainfall characteristics, watershed morphology, water levels, and soil moisture content [2]. Furthermore, issues including missing data, noisy measurements, and invalid records exacerbate data scarcity in rainfall-runoff modeling research [3], presenting additional challenges for model construction.

Neural Networks (NNs) [4] have proven valuable for rainfall-runoff forecasting by discovering subtle functional relationships in data even when underlying relationships are unknown or difficult to describe [5]. However, NNs suffer from several limitations, including difficulty in obtaining stable solutions, risks of model overfitting, and challenges in acquiring appropriate control parameters.

In recent years, Support Vector Machines (SVM), developed from statistical learning theory, have gained increasing attention. SVM adheres to structural risk minimization principles by seeking to minimize the upper bound of generalization error rather than training error [6]. With Vapnik' s introduction of the insensitive loss function, the SVM regression model—Support Vector Regression (SVR)—has attracted growing interest for solving nonlinear estimation problems. SVR has become a powerful method for classification problems on large-scale datasets, and due to its attractive characteristics and excellent generalization performance, has achieved significant success in both academic and industrial applications.

Nevertheless, SVR faces two critical problems: selecting the appropriate kernel function type and determining optimal kernel function parameters. These issues are crucial because the kernel function type influences SVR parameters, while parameter optimization determines kernel function performance; these two aspects interact and must be considered synergistically. SVR performance heavily depends on both kernel function type selection and parameter settings. Appropriate choices can improve regression accuracy, while improper parameters lead to overfitting or underfitting [7]. In conventional SVR models, effectiveness largely depends on operator experience, with practitioners often using trial-and-error methods to obtain optimal parameters—a process that risks overfitting and produces models that may predict historical events well but fail to forecast future events reliably.

Meanwhile, Particle Swarm Optimization (PSO), a swarm intelligence algorithm, has been widely applied to optimize kernel function parameters. Proposed by Kennedy in 1995, PSO converges rapidly by continuously updating particle positions and velocities to search for optimal solutions. While simple optimization algorithms may yield satisfactory results for relatively straightforward problems, PSO often suffers from premature convergence when addressing multidimensional nonlinear optimization problems, causing it to become trapped in local optima. To overcome this limitation, researchers have proposed various improvement methods. For instance, Wu Jiansheng et al. employed BP algorithms to avoid convergence stagnation in PSO’ s local search process [8]. Simulated Annealing (SA), conversely, is an effective global optimization algorithm that can avoid local minima and find global optimal solutions, thus effectively compensating for PSO’ s deficiencies.

Recent studies have proposed algorithms for optimizing Gaussian kernel function parameters using Genetic Algorithms (GA), PSO, and Immune Algorithms (IA) [9-11], but no existing work addresses the simultaneous selection of SVR kernel function types. This research gap motivates our proposed approach.

1.2 Support Vector Regression Parameters

Support Vector Regression has emerged as an efficient method for solving nonlinear regression problems. Designing an effective SVR model requires careful selection of four fundamental parameters, as different parameter settings significantly impact SVR performance:

a) **Kernel function type**: Constructs a nonlinear mapping between input space and feature space.

b) **Kernel parameters**: Include the intercept parameter for polynomial kernels, degree coefficient for polynomial kernels, bandwidth for Gaussian kernels, and parameters for sigmoid kernels.

c) **Regularization parameter C**: Balances the trade-off between minimizing training error and reducing model complexity.

d) **-insensitive loss function**: Determines the approximation accuracy at training data points.

The objective of SVM is to ensure the -insensitive loss function remains as small as possible during training. According to literature [12], the SVM function is expressed as:

where K(x_i, x_j) represents the kernel function. In machine learning theory, four commonly used kernel functions exist:

- Linear kernel: K(x_i, x_j) = x_i^T x_j
- Polynomial kernel: K(x_i, x_j) = (t x_i^T x_j + d)^p
- Gaussian (RBF) kernel: K(x_i, x_j) = exp(-||x_i - x_j||^2 / 2 ^2)
- Sigmoid kernel: K(x_i, x_j) = tanh(x_i^T x_j +)

2 HPSOSA Hybrid Optimization Strategy for Support Vector Regression

This paper proposes a hybrid optimization algorithm combining Simulated Annealing and Particle Swarm Optimization to simultaneously optimize SVR kernel function types and parameters, designated as SVR-HPSOSA. This algorithm can determine appropriate kernel function types and optimal kernel parameters for SVR models. The SVR-HPSOSA algorithm prevents PSO from prematurely converging to local optima and enables discovery of global optimal solutions. Applied to Liujiang River runoff data in Liuzhou, Guangxi, the method establishes a runoff forecasting model based on hybrid PSO-SA evolved SVR. Results demonstrate that the SVR-HPSOSA model achieves high prediction accuracy and generalization capability for daily rainfall-runoff forecasting.

2.1 Particle Swarm Optimization Algorithm

PSO, invented by Kennedy and Eberhart in 1995, is an evolutionary computation technique that has successfully solved complex optimization problems. Its flexibility and effectiveness in simple simulation systems have attracted significant research interest. PSO originates from studies of bird flock foraging behavior, with the fundamental concept that collaboration and information sharing among individuals in a population enable optimal solution discovery.

The primary advantage of PSO lies in its simplicity and ease of implementation without requiring extensive parameter tuning. It has been widely applied to function optimization, neural network training, fuzzy system control, and other domains traditionally served by genetic algorithms. However, recent research has identified limitations, particularly premature convergence that causes the swarm' s optimal solution to become trapped in local minima during the search process.

The core idea of SVR regression analysis assumes training data where x_i are input vectors, y_i are output vectors, and N represents the total data dimensionality. The linear regression function is defined as:

f(x) = ^T (x) + b

where and b are correlation coefficients, and (x) represents the nonlinear mapping from input space to high-dimensional feature space.

In PSO, particle i's position in N-dimensional space is represented by vector X_i = (x_i1, x_i2, ···, x_iN), and its flight velocity by vector V_i = (v_i1, v_i2, ···, v_iN). Each particle possesses a fitness value determined by the objective function and maintains knowledge of its personal best position (Pbest_i) and current position X_i, representing its individual flight experience. Additionally, each particle knows the global best position (Gbest) discovered by the entire swarm, representing collective experience. Particles combine personal and social knowledge to determine subsequent movements.

PSO initializes as a swarm of random particles (random solutions) and iteratively searches for optimal solutions. During each iteration, particles update themselves by tracking two "extremes" (Pbest_i and Gbest). After identifying these optimal values, the standard PSO algorithm updates velocity and position using:

V_i = × V_i + c1 × rand() × (Pbest_i - X_i) + c2 × rand() × (Gbest - X_i)
X_i = X_i + V_i

where M is the total number of particles in the swarm; is a non-negative inertia factor; V_i represents particle velocity; Pbest_i and Gbest are as previously defined; rand() generates random numbers between 0 and 1; X_i is the particle's current position; and c1 and c2 are learning factors typically set to c1 = c2 = 2. In each dimension, particles have a maximum velocity limit V_max. If velocity in any dimension exceeds V_max, it is clamped to V_max.

2.3 HPSOSA Hybrid Optimization for SVR Parameters

To obtain global or near-global optimal solutions, this paper proposes a hybrid optimization algorithm (HPSOSA) that integrates PSO and SA, leveraging SA's ability to avoid local optima while obtaining global solutions and PSO's advantages of speed and ease of implementation. The HPSOSA algorithm is straightforward to implement and combines the strengths of both approaches.

Generally, when selecting parameters based on a small yet appropriate feature subset, most researchers still follow a trial-and-error approach: constructing several SVR models with different parameter sets and testing them on validation sets to identify optimal parameters. However, this process is time-consuming and relies heavily on chance. Beyond selecting appropriate kernel function types, optimal parameter settings can also improve SVR model accuracy. Since kernel function types affect kernel parameters, both aspects should be considered simultaneously to obtain proper SVR kernel functions and optimal parameters.

This study employs the HPSOSA algorithm to simultaneously optimize kernel function types and parameters, establishing an SVR model based on HPSOSA optimization. The improved SVR-HPSOSA model is applied to daily rainfall-runoff data from the Liujiang River in Liuzhou, Guangxi.

2.4 Chromosome Representation

summarizes the various kernel function types, effective kernel parameters, and SVR parameters requiring optimization. The chromosome is therefore designed with three components: kernel function type (integer), kernel parameters (real numbers), and SVR parameters (real numbers). The table illustrates a chromosome representation with d dimensions, where d represents the number of features varying across different datasets. Chromosome segments 00, 01, 10, and 11 encode kernel function types, corresponding to linear, polynomial, Gaussian, and sigmoid kernels, respectively. Chromosome segments x_{i1}, x_{i2}, x_{i3} represent kernel parameter values, while segments x_{i4}, x_{i5} denote penalty parameter and insensitive loss function values.

2.5 SVR-HPSOSA Model

[Figure 2: see original paper] illustrates the SVR–HPSOSA optimization process. The algorithm proceeds as follows:

a) **Particle initialization and parameter setting**: Generate initial particle combinations for SVR kernel types and parameters. Set $k = 0$ and randomly initialize particle swarm positions X_i^0 and velocities V_i^0.

b) **Fitness calculation**: Input training data and compute each chromosome' s fitness value using Equation (11). The fitness function is defined as the mean absolute percentage error (MAPE) from 5-fold cross-validation on the training dataset:

$$\text{Fitness} = \text{MAPE\_cross-validation} = (1/n) \times \Sigma|(y_i - \hat{y}_i)/y_i| \times 100\%$$

where n is the number of training samples, y_i represents actual values, and $\hat{y}_i$ represents predicted values. In this study, 5-fold cross-validation provides an optimal compromise between computational cost and effective parameter estimation for fitness calculation on the training dataset. For each particle, compute its fitness value using Equation (11).

c) **Particle evaluation**: Assess all particle fitness values and determine Gbest and Pbest through simple comparison of fitness values. Set average training accuracy variation as the global value and validation accuracy as individual best.

d) **PSO execution**: Update global and individual best values based on fitness evaluation results.

e) **SA acceptance criterion**: Generate a random number U [0,1]. If $U > \exp(-\Delta\text{Fitness}/T_k)$, where $\Delta\text{Fitness} = \text{Fitness}(s_k) - \text{Fitness}(s_{k+1}) > 0$ (meaning fitness improvement), accept the new position. The new position s_{k+1} is set according to the following principle: if all particle velocities have been determined, proceed to step f); otherwise, return to step b) to generate new velocities for unaccepted particles. In our study of 100 examples, identical particles forced acceptance of final velocities while considering computational time to prevent endless loops. This potential failure did not occur during our research and simulations.

f) **Position update**: Update each particle' s new position through simple fitness value comparison and modify Pbest and Gbest values.

g) **Termination check**: If the evolutionary process has reached a satisfactory state (or maximum regression count), proceed to step 3. Otherwise, adjust inertia weight and SA temperature T_k, set k = k + 1, and return to step b).

h) **Output**: Return the best solution Gbest and its fitness value. Retrieve the kernel function type and parameter values at iteration termination.

i) **Model construction**: Based on selected kernel function types and parameters, repeatedly train and construct SVR on a larger training set. Finally, test the trained SVR model on the test set.

3 Experimental Results and Analysis

The improved SVR-HPSOSA method was implemented on an Intel Celeron M 1.86 GHz processor with 1.5 GB RAM, Windows XP operating system, and MATLAB development environment. PSO parameters were configured as: 100 iterations, population size of 100, minimum inertia weight 0.1, maximum inertia weight 0.9, and learning rate 2. SA parameters included: initial temperature 5000, termination temperature 0.9, and temperature reduction coefficient 0.001.

3.1 Experimental Data

The Liujiang River, part of the Pearl River system, has a main stream length of 773.3 km and a drainage area of 57,173 km^2 across 30 counties and cities in Guizhou, Guangxi, and Hunan provinces. With a natural drop of 1,306 meters, average gradient of 1.68‰, and annual average flow of 1,865 m^3/s, the river experiences a flood season from April to September. During flood seasons, the Liujiang River section experiences frequent flooding. Over the past two decades since 1988, major floods have occurred repeatedly with severe consequences, including one catastrophic flood exceeding the 100-year return period ("1996.7"), three major floods with 20-50 year return periods ("1988.8" , "1994.6" , and "2004.7"), and one significant flood with a 10-20 year return period ("2000.6"). Establishing an accurate flood season runoff forecasting model for the Liujiang River is therefore crucial for disaster prevention and mitigation.

This study utilizes daily water level data at 12:00 from the Laojiaokou hydrological station on the Liujiang River in Guangxi, comprising 1,004 data points from January 1, 2008, to September 30, 2010. The dataset is divided into two parts: 821 data points from January 1, 2008, to March 31, 2010, are used for model training, while the remaining 183 samples from April 1, 2010, to September 30, 2010 (the primary flood season) serve as test data.

3.2 Input Variables

Rainfall-runoff causal relationships typically involve variables such as rainfall amount, pre-rainfall water level, evaporation, and temperature. Most studies employ rainfall and antecedent flow (or water level) as input variables. According to stepwise regression methods, considering both antecedent rainfall and runoff time series shows stronger correlation with current runoff. Therefore, these two optimal input combinations are selected as model inputs. Following literature [15], the values of runoff time series at time points t-1, t-2, t-3, and t-4, along with rainfall at time point t-1, are used as model inputs through stepwise regression analysis.

3.3 SVR-HPSOSA Model Training

Using identical predictors, the SVR-HPSOSA model predictions are compared with standard SVR model results. This study employs simple parameter settings with Gaussian kernel bandwidth = 0.75. The remaining two parameters—regularization constant C and loss function —are selected through trial-and-error. Parameters yielding the minimum Root Mean Square Error (RMSE) on test data are considered optimal, resulting in the best testing and prediction performance.

3.4 Performance Evaluation

Numerous performance metrics exist for rainfall-runoff forecasting models. This study employs the following evaluation criteria: Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Coefficient of Efficiency (CE). Additionally, since peak points are crucial in rainfall-runoff modeling, Equation (12) compares different models' capabilities in capturing peak flow trends throughout the runoff time series.

Performance metrics RMSE and MAPE primarily evaluate model performance numerically, where smaller values indicate better predictive performance. The CE metric assesses whether the model correctly identifies runoff trends, with larger values indicating better capability to accurately predict future runoff patterns. Equation (12) specifically evaluates flood peak trend prediction, which is of utmost practical concern for disaster prevention—whether the model can accurately predict catastrophic floods when peak flows occur. In this study, flow rates exceeding 4,000 m^3/s are considered flood peaks, where n_peak in Equation (12) represents the number of samples exceeding 4,000 m^3/s within the prediction time interval.

presents comparative performance metrics for model fitting accuracy, prediction accuracy, and model efficiency between pure SVR (Gaussian kernel) and SVR-HPSOSA models. As shown, for training data, SVR achieves an RMSE of 167.863, while SVR-HPSOSA reduces this error to only 74.218. Similarly, for test data, SVR yields an RMSE of 909.564, whereas SVR-HPSOSA achieves 201.723. The SVR-HPSOSA model also exhibits lower MAPE values for both training and test data. These results indicate minimal deviation between observed and predicted values. Furthermore, the SVR-HPSOSA model achieves the highest CE efficiency coefficient among all models, demonstrating its capability to capture daily runoff variation trends.

Experimental results demonstrate that the SVR-HPSOSA model is highly promising for runoff prediction, exhibiting both excellent fitting performance and effective predictive capability. The consistency between training and testing results shows that SVR-HPSOSA outperforms SVR across all three performance metrics (RMSE, MAPE, and CE) for runoff prediction with identical inputs.

Compared with other models, SVR-HPSOSA demonstrates superior peak prediction capability—a critical reference for flood prevention and mitigation plans in practical rainfall-runoff applications. reveals that SVR-HPSOSA provides better predictions for extreme runoff data, substantially outperforming the standard SVR model. The SVR-HPSOSA model achieves training data CE__peak of 0.984 and test data CE__peak of 0.977, demonstrating high processing capability for peak data. Therefore, the proposed model is suitable for establishing SVR prediction models for Liujiang River flood season runoff with high accuracy.

Further analysis of indicates that the optimal kernel function type for runoff modeling is the Sigmoid kernel, with optimal parameter settings: = 0.325, = 0.0264, C = 18.60, and = 0.1059.

3.5 Results Analysis

[Figure 3: see original paper] compares predicted and observed runoff values for flows below 4,000 m^3/s from April 1 to September 30, 2010, for both SVR-HPSOSA and standard SVR (Gaussian kernel) models. [Figure 4: see original paper] presents a similar comparison for runoff exceeding 4,000 m^3/s during the same period. The figures clearly show that SVR-HPSOSA predictions align more closely with actual observed water levels than SVR model predictions, particularly for peak flow prediction where SVR-HPSOSA generates superior results while other models tend to overestimate or underestimate peaks.

4 Conclusion

The rainfall-runoff system represents one of the most dynamic meteorological systems. The key contribution of this paper lies in fusing the respective advantages of PSO and SA to evolve SVR kernel function types and parameters, establishing the SVR-HPSOSA runoff forecasting model. Experiments on real rainfall-runoff data demonstrate that the SVR-HPSOSA model successfully identifies appropriate kernel function types and parameter values, showing relatively high prediction accuracy compared to pure SVR models.

Based on our research findings, the SVR-HPSOSA method exhibits the following characteristics:

a) The SVR-HPSOSA algorithm avoids local optima by hybridizing SA' s acceptance criteria with PSO' s random acceptance principles, reducing oscillation deviations near local search endpoints. By applying rules to accept new positions or recalculate alternative credible positions based on fitness differences between old and new positions, the model can escape local optima and reduce oscillation near search endpoints, thereby improving solution quality and convergence speed.

b) The SVR-HPSOSA algorithm combines PSO' s operational mechanisms and parallel processing interference capabilities with SA' s search path characteristics, enabling rapid global optimization with fast computation speed.

c) Empirical results demonstrate that Sigmoid kernel SVR models can serve as an effective tool for practical runoff prediction applications, achieving favorable forecasting accuracy and high prediction quality.

d) When using Sigmoid kernels, SVR implements a feedforward neural network. However, through the SVR methodology, both hidden layer node counts and connection weights are automatically determined during design, and SVR theory ensures global rather than local optimal solutions, guaranteeing good generalization capability for unknown samples without overfitting. This explains why Sigmoid kernel SVR models are particularly suitable for runoff prediction and can be extended to other regions, representing a robust runoff forecasting tool for hydrological research and system management.

References

[1] Wu Jiansheng, Liu Mingzhe, Jin Long. A hybrid support vector regression approach for rainfall forecasting using particle swarm optimization and projection pursuit technology [J]. International Journal of Computational Intelligence and Applications, 2010, 9(2): 87-104.

[2] Teng Shaohua, Fan Jihui, Chen Xiao, et al. Application of SVM-based multi-classifiers in mining cooperatively local area meteorological data [J]. Journal of Guangxi University: Natural Science Edition, 2014, 39(5): 1131-1137.

[3] Wu Jiansheng. An effective hybrid semi-parametric regression strategy for rainfall forecasting combining linear and nonlinear regression [J]. International Journal of Applied Evolutionary Computation, 2011, 2(4): 50-63.

[4] Talei A, Chua L H C, Quek C. A novel application of a neuro-fuzzy computational technique in event-based rainfall–runoff modeling [J]. Expert Systems with Applications, 2010, 37(12): 7456-7468.

[5] Zhang Chenggang, Song Jiazhi, Jiang Jingqing, et al. Imbalanced data classification algorithm of improved de-noising auto-encoder neural network [J]. Application Research of Computers, 2017, 34(5): 1329-1332.

[6] Kisi O, Shiri J, Tombul M. Modeling rainfall-runoff process using soft computing techniques [J]. Computers & Geosciences, 2013, 51(2): 108-117.

[7] Hong Weichang, Dong Yucheng, Zheng Feifeng, et al. Hybrid evolutionary algorithms in a SVR traffic flow forecasting model [J]. Applied Mathematics & Computation, 2011, 217(15): 6733-6747.

[8] Zhang Shujuan, Deng Xiuqin, Liu Bo. Research on tax forecasting model based on PSO and least squares support vector machine [J]. Computer Science, 2017, 44(s1): 119-122.

[9] Trelea I C. The particle swarm optimization algorithm: convergence analysis and parameter selection [J]. Information Processing Letters, 2016, 85(6): 317-325.

[10] Chakrapani B, Varadarajan V. Genetic algorithm based demand side management for smart grid [J]. Wireless Personal Communications, 2017, 93(2): 481-502.

[11] Li Xuejun, Xu Jia, Zhu Erzhou, et al. A novel computation method for adaptive inertia weight of task scheduling algorithm [J]. Journal of Computer Research and Development, 2016, 53(9): 1990-1999.

[12] Ireneusz G. Discussion on semi-immune algorithm behaviour based on fractal analysis [J]. Soft Computing, 2017, 21(14): 3945-3956.

[13] Bermolen P, Rossi D. Support vector regression for link load prediction [J]. Computer Networks, 2009, 53(2): 191-201.

[14] Wu Yile, He Qing. WSN path optimization algorithm based on improved genetic simulated annealing algorithm [J]. Application Research of Computers, 2016, 33(10): 2959-2962.

[15] Asadi S, Shahrabi J, Abbaszadeh P, et al. A new hybrid artificial neural networks for rainfall-runoff process modeling [J]. Neurocomputing, 2013, 121(18): 470-480.

Note: Figure translations are in progress. See original paper for figures.