Short-time traffic flow prediction based on improved LSSVM

K. Ueda
S. Abe
Z. Shen (shenzhou.research@gmail.com)

Research Article

Keywords: Traffic flow prediction, improved LSSVM, data-driven approach

Posted Date: July 12th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-702558/v1

Short-time traffic flow prediction based on improved LSSVM

Kazuki Natsuko Ueda, Soma Kouji Abe, and Zhou Shen*

Abstract

In order to improve the accuracy of short-time traffic flow prediction, an improved LSSVM-based short-time traffic flow prediction model is proposed. To address the problem that the traditional hybrid frog-jumping algorithm (SFLA) easily falls into local optimum, an improved hybrid frog-jumping algorithm (ISFLA) based on a new local update strategy is proposed, which is combined with the least squares support vector machine (LSSVM) to improve the prediction capability of LSSVM by using this algorithm to optimize the key parameters of LSSVM. The model and algorithm are simulated and analyzed with examples to prove the feasibility of the model and the effectiveness of the algorithm.

1 Introduction

Accurate prediction of traffic flow is an important basis for alleviating traffic congestion and is a core part of realizing intelligent transportation systems [1–4]. Traffic flow prediction refers to the prediction of future traffic flow based on current and past traffic flow information using a suitable prediction model [5–8]. Among them, short-time traffic flow forecasting refers to the prediction of time range within 30 minutes [2, 9–11]. Due to the randomness, uncertainty and suddenness of short-time traffic flow, it is a difficult problem in the field of intelligent traffic management to forecast short-time traffic flow effectively and accurately.

Many prediction methods have been proposed by domestic and foreign scholars for short-term traffic flow prediction. One type is the traditional mathematical methods, such as Kalman filter and autoregressive sliding average. These methods are simple to implement and have good results for predicting slowly changing traffic flow; the disadvantage is that they cannot reflect the uncertainty and nonlinearity in the traffic flow, and it is difficult to predict the short-time traffic flow accurately. To address this problem, scholars have proposed another class of artificial intelligence-based methods, such as neural network methods

*The authors declare no competing interests.

and support vector machine method [2,6,11–17]. Although neural network models have outstanding performance in describing nonlinear systems, they emphasize too much on learning error, which can lead to overfitting, and, since neural network methods are based on the principle of empirical risk minimization, they are ineffective in dealing with small sample data [18–22]. Support vector machines overcome these inherent drawbacks of neural network models, where the least squares support vector machine (LSSVM), which is an improvement of the standard support vector machine, is widely used in short-time traffic flow prediction [23]. The prediction results of this algorithm are closely related to the selection of its relevant parameters. Unsuitable parameter selection can lead to the degradation of LSSVM prediction ability. Genetic algorithms, particle swarm algorithms, and grid search algorithms have been used to optimize the relevant parameters of LSSVM. However, these algorithms have disadvantages such as easily falling into local optima with too many initial parameter settings [24, 25].

The Shuffled Frog Leaping Algorithm (SFLA) is a new population intelligence optimization algorithm proposed by Muzaffar Eusuff and Kevin Lansey in 2003, which can be used to solve many complex, such as nonlinear, multidimensional optimization problems. This algorithm combines the advantages of particle swarm algorithm (PSO) and modulo algorithm (MA), and has the features of easy to understand, less setup parameters, fast computation, and better global search and optimization capability.

Like most intelligent optimization algorithms, the SFLA algorithm is also prone to fall into local optima. To address this problem, this paper firstly improves the SFLA algorithm by proposing a new local individual update strategy to make the algorithm jump out of the local optimum and obtain better optimization results to some extent. Based on this, we propose an improved hybrid frog-jumping algorithm to optimize the short-time traffic flow prediction model (ISFLA-LSSVM) by least-squares support vector machine, and improve the prediction accuracy of short-time traffic flow by optimizing the regularization parameter γ and kernel parameter σ in LSSVM. And the effectiveness of ISFLA-LSSVM is verified by simulation experiments.

2 Methodology

For a given training data sample set, $D = \{(x_1, y_1), ..., (x_k, y_k), ..., (x_n, y_n)\}, x_k \in R, y_k \in R$, where x_i is the input of time series samples, y_i is the corresponding output,and n is the number of training samples. Through the nonlinear function $\Phi(x)$, the LSSVM method maps the sample space into the high-dimensional feature space for regression, and the regression function is shown below:

$$f(x) = w^T \varphi(x) + b \tag{1}$$

where w is the weight, b is the deviation value, and $f(x)$ denotes the predicted value. According to the principle of structural risk minimization, the regression

problem of LSSVM can be expressed as an optimization problem as follows:

$$\begin{aligned} &\min \emptyset(w,e) = \frac{1}{2}w^T w + \frac{1}{2}\gamma \sum_{i=1}^{n} e_i^2, \gamma > 0 \\ &\text{s.t. } y_i = w^T \emptyset\left(x_i\right) + b + e_i, i = 1, 2, \cdots, n \end{aligned} \tag{2}$$

where e_i is the error variable, which represents the error between the actual and predicted values, and γ is the regularization parameter, which is used to balance the training error and complexity of the model. In order to solve the above optimization problem, a Lagrangian function is established:

$$\begin{gathered} L(w,b,e,\alpha) = \\ \frac{1}{2}w^T w + \frac{1}{2}\gamma \sum_{i=1}^{n} {e_i}^2 - \sum_{i=1}^{n} \alpha_i \left(w^T \emptyset(x) + b + e_i - y_i\right) \end{gathered} \tag{3}$$

Where α_i denotes the Lagrangian operator. The partial differentiation of w, b, e, and α are set to zero, resp

$$\begin{aligned} &\frac{\partial \mathrm{L}}{\partial \mathrm{w}} = w - \sum_{i=1}^{n} \alpha_i \emptyset\left(x_i\right) = 0 \\ &\frac{\partial \mathrm{L}}{\partial \mathrm{b}} = -\sum_{i=1}^{n} \alpha_i = 0 \\ &\frac{\partial \mathrm{L}}{\partial e_i} = \gamma e_i - \alpha_i - 0, i = 1, \cdots, n \\ &\frac{\partial \mathrm{L}}{\partial \alpha_i} = \mathrm{w}^{\mathrm{T}} \emptyset\left(x_i\right) + b + e_i - y_i = 0, i = 1, 2, \cdots, n \end{aligned} \tag{4}$$

After eliminating w and e from the above set of equations, we can obtain:

$$\mathrm{f(x)} = \sum_{i=1}^{n} \alpha k\left(x, x_i\right) + b \tag{5}$$

where $y = [y_1; ...y_i; ...y_n]$, $I_i = [1; ...1]$, $\alpha = [\alpha_1; ...; \alpha_n] k(x_i, x_j) = \Phi(x_i)\Phi(x_j)^T$, and the fitted function is:

$$\mathrm{f(x)} = \sum_{i=1}^{n} \alpha k\left(x, x_i\right) + b \tag{6}$$

The kernel function used in this paper is the radial basis function, $k(x_i, x_j) = ex(-||x_i - x_j||^2/\delta^2)$, where *sigma* denotes the width of the kernel function. After the kernel function is selected, the appropriate regularization parameter γ and kernel parameter σ need to be chosen. These two parameters affect the learning ability and generalization performance of LSSVM. In this study, the improved frog-jumping algorithm is used to optimize the above two parameters to improve the prediction accuracy of short-time traffic flow.

3 ISFLA-LSSVM-based traffic flow prediction

3.1 Traditional SFLA algorithm

The SFLA algorithm is a new intelligent optimization algorithm that mimics the process of searching for food in a frog population. The algorithm consists of two core processes: the global information exchange process among frog subpopulations and the local update process within subpopulations.

3.1.1 Global information exchange between subpopulations

1. Parameter initialization: L frogs are randomly generated to form the initial frog population $G = X_1, X_2, ..., X_L$, and each frog individual X represents a feasible solution to the problem, i.e., as follows: $X = (x_1, x_2, ..., x_d)$, where represents the dimension of the frog individual.

2. Calculate the fitness value $f(X)$ of all frogs in the population and sort them in descending order according to the fitness value. The sorted population is divided into m subpopulations, each with n frogs, where m and n satisfy $L = m \times n$. The partitioning rule is as follows: the first frog belongs to the first subpopulation, the second frog belongs to the second subpopulation, and so on until the nth frog belongs to the nth subpopulation, and then the $n+1$th frog belongs to the first subpopulation until all frogs are assigned. For the frog population, we denote by G_b the global solution with the maximum fitness value. For each subpopulation, we use Z_b and Z_w to denote the solutions with the maximum and minimum fitness values, respectively.

3. Local iterative search within subgroups: The position of the frog individual with the minimum fitness value in each subgroup is updated according to the search procedure in 2.

4. All the frogs in each subpopulation are mixed into a new population and reordered in descending order by fitness value.

5. The end condition of the algorithm: judge whether the number of iterations t reaches the specified maximum, if not, return to step 2, otherwise output the optimal solution.

3.1.2 Subgroup internal local update

Local update within a subgroup is to update the position of the frog individual with the smallest fitness value in each subgroup.

1. k denotes the number of iterations within the subpopulation, and the initial value is set to 0.

2. Add 1 to k, and then get the frog with the largest and smallest fitness value in the subpopulation according to the fitness value.

3. The frog with the smallest fitness value in the subpopulation is updated, and the update strategy is shown below:

$$\begin{aligned} s_i &= \text{rand}() \times (z_b - z_w) \\ x_{\text{neu}} &= x_{\alpha l} + s_i \left(-D_{\text{mux}} \leqslant s_i \leqslant D_{\max}\right) \end{aligned} \tag{7}$$

   The rand() function generates a random number in the range of 0 to 1, S denotes the movement step of the frog, and $D_{\max}$ denotes the maximum distance the frog is allowed to move.

4. Adjust the corresponding update method according to the update result. If the above process can produce a better solution, i.e., an individual with a larger fitness value, then the frog with the smallest fitness value in the subpopulation is replaced by the frog in the new position, otherwise, F_b is replaced by G_b, i.e., the step size formula becomes:

$$s_i = \text{rand}() \times (\boldsymbol{G}_b - z_w) \tag{8}$$

   Repeat the above update process for the worst individual. If a better solution still cannot be generated, then a frog is randomly generated to replace the frog with the smallest fitness value in the subpopulation.

5. Judgment of the end condition of the subpopulation local update. If the number of iterations k meets the maximum, return to step 3 in (1), otherwise return to step 3 in (2).

3.2 Improved SFLA algorithm

The local update process of the basic frog hopping algorithm is based on the way that the best frog individual influences the worst frog individual to update the worst individual in the subpopulation. If the resulting new individual is not a better solution, i.e., the fitness value is larger, then we update the frog with the worst position based on the maximum population fitness. If the new individual is still not the optimal solution, then we randomly generate a new individual to replace the individual with the lowest fitness value within the original subpopulation.

Based on this, we improve the traditional hybrid frog jumping algorithm in two ways: Firstly, we consider updating the smallest individual in each subpopulation based on the largest individual in the subpopulation and the whole population in the process of local updating, so as to avoid falling into local optimum. In other words, a new formula for calculating the step size is used:

$$s_i = \frac{1}{2} \times (F_b - F_w) + \frac{1}{2} \times (G_b - F_w) \tag{9}$$

Secondly, we propose a new frog individual update formula to update the position of the worst frog individual in each subpopulation. The new formula is shown below:

$$x_{ncw} = x_{al} + w(k) \times s_i \left(-D_{\max} \leqslant s_i \leqslant D_{\max}\right) \tag{10}$$

Let $w(k)$ be a function that decreases as the number of iterations k increases within the subpopulation. $w(k) = (kmax - k + rand())$, from which it can be seen that a large value of $w(k)$ in the early iterations can expand the search range of the solution space, i.e., increase the diversity of solutions, while a small value of $w(k)$ in the later iterations allows the algorithm to search for better solutions in the local range, i.e., further improve the The smaller $w(k)$ values in the later iterations allow the algorithm to search for better solutions in the local range, i.e., further improve the local search capability.

3.3 ISFLA Optimized LSSVM for Short-Term Traffic Forecasting Model

The steps to optimize the parameters in the LSSVM model with the improved hybrid frog-jumping algorithm (ISFLA) in this paper are shown below:

1. Initialize the basic parameters, such as the number of frogs L, the number of subpopulations m, the number of iterations k within the subpopulation, the global iteration number t, the maximum iteration number k_{max}, the global maximum iteration number t_{max}, and the maximum distance D_{max} that an individual frog is allowed to move.

2. The position of each individual frog corresponds to a set of regularization parameters and kernel parameters (γ, σ) of LSSVM, and the fitness values of all frogs are calculated and the global optimal solutions are recorded. We set the precision PRE as the fitness function, and each individual frog is ranked according to the value of the fitness function and divided into subgroups.

3. For each subpopulation, the frog with the smallest fitness value is first updated according to Eqs. (10) and (11). If the fitness value of the frog in the new position is larger than the original fitness value, the updated frog replaces the frog with the smallest fitness value, otherwise a new frog is still randomly generated to replace the frog with the smallest fitness value in the atomic group, and the above process is repeated until the maximum number of iterations within the subpopulation is satisfied.

4. Mix all subpopulations and re-sort all frogs in descending order by fitness value.

5. If the termination condition is not satisfied, return to step (2), otherwise output the position of the frog with the largest fitness value, i.e., the optimal LSSVM parameter pair (γ, σ).

4 Case studies

In this paper, the data were collected from a section of highway in Sichuan province in 2012, and the traffic volume was collected every 15 minutes from

9:00 am to 8:00 pm every day from July 15 to 22, and $48 \times 8 = 384$ data were obtained. The data of the first 7 days were used as the training set to train the model, and the data of the last 1 day were used as the test set to verify the prediction effect of the model.

Obviously, there is a clear connection between current traffic flow and historical traffic flow. $v(t)$ denotes the traffic flow at time t, and $v(t-1)$ denotes the traffic flow at time $t-1$. We use the traffic flow at current time and historical time to predict the traffic flow at the next time as follows:

$$X_i = \{v(t-n), \cdots, v(t-1), v(t)\}, Y_i = v(t+1) \tag{11}$$

Where X_i denotes the input of the system and Y_i denotes the output of the system. In this paper, we set n to be 6.

4.1 Parameter Setting

In order to verify the effectiveness of ISFLA-LSSVM model in short-time traffic flow prediction, three experiments were conducted as follows: (1) LSSVM model optimized by improved hybrid frog-hopping algorithm (ISFLA) for prediction; (2) LSSVM model optimized by traditional SFLA algorithm for prediction; (3) LSSVM model optimized by grid search algorithm (GS) for prediction. LSSVM model for prediction. The parameters of the algorithm in experiment (1) are set as follows: the range of γ is $(0, 1000)$, the range of σ is $(0, 100)$, the number of frogs is set to 100, the number of subgroups is set to 5, the number of iterations within the subgroups is set to 10, the total number of iterations is set to 20, and the maximum move step is set to 50. The parameters of experiment (2) are set as in experiment (1). The parameters of experiment (3) are set as follows: the iteration cutoff error is 10^{-3}, the search range of γ and σ is the same as above, and the step size is set to 10.

4.2 Evaluation indexes

In order to verify the effectiveness and superiority of the proposed model, the following metrics are used to evaluate the prediction performance of the model: MRE, RMSE, and precision PRE:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{\sum_{k=1}^{n} \left(x_k - \overline{x_k}\right)^2}{n}} \\ \text{MRE} &= \frac{1}{n} \sum_{k=1}^{n} \frac{\mid x_k - \overline{x_k} \mid}{x_k} \\ \text{PRE} &= 1 - \frac{1}{n} \sum_{k=1}^{n} \frac{|x_k - \overline{x_k}|}{x_k} \end{aligned} \tag{12}$$

where x_k is the predicted value is the actual value. The smaller the value of RMSE and MRE, the larger the value of precision PRE, the stronger the prediction performance of the model.

4.3 Experimental results

By optimizing the LSSVM parameters with the ISFLA algorithm, we can construct an ISFLA-LSSVM-based short-time traffic flow prediction model. From Table 1, we can see that the MRE and RMSE of the ISFLA-LSSVM model are the smallest, 0.058 and 39.60, respectively, which are significantly lower than those of the SFLA-LSSVM model and GS-LSSVM model. Therefore, compared with the other two models, the ISFLA-LSSVM model proposed in this paper is the optimal one.

Model	MRE	RMSE	PRE
ISFLA-LSSVM	0.0587	39.60	94.13
SFLA-LSSVM	0.0803	49.85	91.97
GS-LSSVM	0.1231	77.72	87.66

Table 1: Result Comparisons

5 Conclusions

The traditional hybrid frog-jumping algorithm is easy to fall into local optimum, and this paper proposes an improved hybrid frog-jumping algorithm (ISFLA) by modifying the local update strategy of this algorithm. The improved algorithm has a better balance between global search ability and local update ability, and is less likely to fall into local optimality. In addition, to address the problem of difficult parameter selection when using the least squares support vector machine model for short-time traffic forecasting, the ISFLA algorithm is used to optimize the parameters in this paper, and the ISFLA-LSSVM short-time traffic forecasting model is proposed. It is demonstrated that the improved model has higher prediction accuracy compared with the reference model, and is suitable for short-time traffic prediction.

References

[1] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.

[2] J. J. Q. Yu, "Citywide traffic speed prediction: a geometric deep learning approach," *Knowledge-Based Systems*, p. 106592, Nov. 2020.

[3] S. Clark, "Traffic prediction using multivariate nonparametric regression," *Journal of transportation engineering*, vol. 129, no. 2, pp. 161–168, 2003.

[4] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: a federated learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7751–7763, Aug. 2020.

[5] C. Van Hinsbergen, J. Van Lint, and F. Sanders, "Short term traffic prediction models," in *PROCEEDINGS OF THE 14TH WORLD CONGRESS ON INTELLIGENT TRANSPORT SYSTEMS (ITS), HELD BEIJING, OCTOBER 2007*, 2007.

[6] J. J. Q. Yu, C. Markos, and S. Zhang, "Long-term urban traffic speed prediction with deep learning on graphs," *IEEE Transactions on Intelligent Transportation Systems*, 2021, in press.

[7] S. Ishak and H. Al-Deek, "Performance evaluation of short-term time-series traffic prediction model," *Journal of transportation engineering*, vol. 128, no. 6, pp. 490–498, 2002.

[8] C. Zhang, S. Zhang, J. J. Q. Yu, and S. Yu, "FASTGNN: A topological information protected federated learning approach for traffic speed forecasting," *IEEE Transactions on Industrial Informatics*, 2021, in press.

[9] Y. Gu, W. Lu, X. Xu, L. Qin, Z. Shao, and H. Zhang, "An improved bayesian combination model for short-term traffic prediction with deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1332–1342, 2019.

[10] H. Tan, Y. Wu, B. Shen, P. J. Jin, and B. Ran, "Short-term traffic prediction based on dynamic tensor completion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2123–2133, 2016.

[11] J. J. Q. Yu and J. Gu, "Real-time traffic speed estimation with graph convolutional generative autoencoder," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3940–3951, Oct. 2019.

[12] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning," *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1688–1711, 2019.

[13] D. Kang, Y. Lv, and Y.-y. Chen, "Short-term traffic flow prediction with lstm recurrent neural network," in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*. IEEE, 2017, pp. 1–6.

[14] J. J. Q. Yu, "Sybil attack identification for crowdsourced navigation: a self-supervised deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, 2020, in press.

[15] Q. Liu, B. Wang, and Y. Zhu, "Short-term traffic speed forecasting based on attention convolutional neural network for arterials," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 11, pp. 999–1016, 2018.

[16] J. J. Q. Yu, "Travel mode identification with GPS trajectories using wavelet transform and deep learning," *IEEE Transactions on Intelligent Transportation Systems*, 2020, in press.

[17] ——, "Semi-supervised deep ensemble learning for travel mode identification," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 120–135, Mar. 2020.

[18] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel svm with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2018.

[19] J. J. Q. Yu, "Online traffic speed estimation for urban road networks with few data: a transfer learning approach," in *Proc. IEEE Intelligent Transportation Systems Conference*. Auckland, New Zealand: IEEE, 2019, pp. 4024–4029.

[20] C. Luo, C. Huang, J. Cao, J. Lu, W. Huang, J. Guo, and Y. Wei, "Short-term traffic flow prediction based on least square support vector machine with hybrid optimization algorithm," *Neural processing letters*, vol. 50, no. 3, pp. 2305–2322, 2019.

[21] J. J. Q. Yu and A. Y. S. Lam, "Autonomous vehicle logistic system: Joint routing and charging strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2175–2187, Jul. 2018.

[22] J. J. Q. Yu, "Two-stage request scheduling for autonomous vehicle logistic system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1917–1929, May 2019.

[23] H. Wang and D. Hu, "Comparison of svm and ls-svm for regression," in *2005 International Conference on Neural Networks and Brain*, vol. 1. IEEE, 2005, pp. 279–283.

[24] J. J. Q. Yu and V. O. K. Li, "A social spider algorithm for global optimization," *Applied Soft Computing*, vol. 30, pp. 614–627, May 2015.

[25] ——, "A social spider algorithm for solving the non-convex economic load dispatch problem," *Neurocomputing*, vol. 171, pp. 955–965, Jan. 2016.