Analysis of Nonlinear Problems with Distributed Least Squares Algorithms

Puze Ying

School of Shanghai Maritime University, Shanghai 200135, China.

yingpuze@yeah.com

Abstract

Distributed network structure is an algorithm processing structure, which is widely used in various fields. The function of this structure is to connect each neighbor node and modify its own behavior according to the data obtained from these nodes. The traditional distributed network structure algorithm is not ideal for dealing with nonlinear problems. The purpose of this paper is to optimize the effect of parameter approximation by adjusting and improving the algorithm, so as to ensure the convergence of the algorithm and maximize the steady-state error. First of all, this paper briefly describes the principle of the classic distributed dynamic programming network structure; secondly, on this basis, it puts forward problems and improvement measures for the algorithm of the distributed network structure; then draws the simulation conclusion and proves that the algorithm is better.

Keywords

Distributed Algorithm; Nonlinear; Adaptive Dynamic Programming; Network Topology; Kernel Learning.

1. Introduction

With the development of adaptive algorithm, Internet technology makes the data collection, analysis and processing more simple and stable. As a research frontier, intelligent information processing involves not only machine learning, artificial intelligence, but also pattern recognition and optimization theory. It has a high application prospect and research value.

Distributed adaptive algorithm is an important branch of intelligent information processing. Based on the traditional adaptive algorithm theory, the concept of wireless sensor network is introduced. Compared with the traditional adaptive filtering, the distributed filter can estimate the location parameters by analyzing the data collected from each neighborhood in the distributed network. Compared with the traditional single node adaptive algorithm, the distributed algorithm has faster convergence performance and smaller steady-state error.

Due to the characteristics of intelligent information network, the distributed algorithm itself has scalability, which can be easily combined with other methods. Both the distributed diffusion proportion algorithm in non Gaussian environment and the rc-dlms algorithm for multi hop neighbor communication to improve the estimation accuracy is the expansion and derivation of the distributed dynamic planning network .

Compared with the traditional adaptive algorithm, distributed algorithm can collect and analyze the surrounding data because of its own network structure. It can handle more types of data, and the density of data is lower. The data collected from neighbor nodes is processed adaptively by iterative analysis. In the case of sufficient data, by sharing the data of each node, we can also effectively mine the important information under the big data, which is an important part of dynamic optimization algorithm.

Distributed algorithm is mainly used to deal with linear data.it is difficult to obtain accurate mathematical model in nonlinear mathematical model. In the process of data iteration, the distributed algorithm processes the nonlinear data slowly, and inevitably produces some errors in the process of signal extraction.

This paper will optimize the distributed algorithm in nonlinear problems. Distributed algorithms are mainly divided into incremental and diffusion, and this paper will focus on incremental scheme.

2. Adaptive filter theory and its algorithm

2.1 Adaptive filter theory

The main function of adaptive filter is to adjust the current time parameters iteratively to obtain the best filtering effect. This feature is called adaptability. It can achieve the ideal state by constantly adjusting the coefficients of the filter when the signal is unchanged; and it can detect and adjust the filter coefficient in time when the signal changes, so as to track the signal. This change requires a cost function to determine the weight of the data at this time.[1]

The input and output signals are determined before the cost function is determined. Let $[x(1), x(2), \cdots x(i)]^T$ be the nonlinear input signal at each time.[2] At the same time, there is a corresponding time I and a corresponding weight function w in these times. At each time, by iterating the input signal, the error signal generated in the process converges continuously, which makes the error signal approach to zero continuously, so as to obtain the optimal solution of the output signal. These weighted signals can form a corresponding set $[w_1(i), w_2(i), \cdots w_M(i)]^T$。[3]

2.2 Diffusion LMS algorithm

The traditional diffusion LMS algorithm estimates the unknown parameters by exchanging data between nodes.[4] First proposed by Federico S. Cattivelli. It updates the real-time estimates according to the adaptive diffusion algorithm of the distributed network.[5]

Node collaborates with neighbor nodes at i-1 time, so that neighbor nodes estimate $\{\varphi_{k,i-1}; k \in N_k\}$ at i-1 time for target parameter w^0 to be done estimated, where N_k represents nodes connected to K node including their own nodes. [6] These estimates are fused at i-1 time, that is, at i-1 time.

$$w_{k,i-1} = f_k(\varphi_{k,i-1}), k \in N_k \tag{1}$$

Where $f_k()$is to collect data from neighboring nodes to produce a fusion estimate $w_{k,i-1}$ based on the merge policy of node k.[7]

Based on the obtained $w_{k,i-1}$, the local estimate $\varphi_{k,i-1}$ is updated by the least mean square adaptive filter algorithm.

Formula 1 can be extended to include:

$$w_{k,i-1} = \sum_{l \in N_k} c_{l,k} \varphi_{k,i-1} \tag{2}$$

Where $c_{l,k}$ represents the fusion weight of node i to node k, where node k and node i must communicate with each other, which is determined by the network structure of the diffusion strategy.[8]

The resulting estimate is expressed as a column vector. That is:

$$\varphi_{N_k} = col\{\varphi_{l,i-1}\}_{l \in N_k} \tag{3}$$

This make the problem turn into a local mean square estimation problem:

$$\min_{w} \left\| \varphi_{N_k} - Q w_k \right\|_{C_k}^2 \tag{4}$$

Where $Q = \text{col}\{I_M, \ldots, I_M\}$,$I_M$ is the identity matrix of$M \times M$. According to the weight matrix, the solution of equation (4) can be obtained as follows:

$$w_{k,i-1} = \sum_{l \in N_k} \frac{c_{l,k}}{\sum_{r \in N_k} c_{r,k}} \varphi_{l,i-1} \tag{5}$$

Obviously, the fusion weight is redefined as:

$$c_{l,k} \longleftarrow \frac{c_{l,k}}{\sum_{r \in N_k} c_{r,k}} \tag{6}$$

The sum of weight $c_{l,k}$ is equal to 1.

Based on the above deduction process, it is known that in the network, node K first fuses the estimates of adjacent node l, then adaptively updates the iteration data according to the data obtained from each node according to the LMS algorithm, and then passes the good results to the adjacent nodes. Therefore, the iteration process of the diffuse LMS algorithm is obtained:

$$\begin{cases} w_{k,i-1} = \sum_{l \in N_k} c_{l,k} \varphi_{k,i-1} \\ e_{k,i} = d_{k,i} - u_{k,i} w_{k,i-1} \\ \varphi_{k,i} = w_{k,i-1} + \mu_k u_{k,i}^T e_{k,i} \end{cases} \tag{7}$$

The distributed least-mean-square algorithm for the entire diffusion strategy consists of three phases: adaptive phase, exchange phase and fusion phase[9].

In the adaptive phase, by updating the LMS algorithm between the expected signal $d_{k,i}$ and the input vector $u_{k,i}$, a new estimator $\varphi_{k,i}$ is obtained based on the local estimator $w_{k,i-1}$ of node k at i-1 time, and the update equation is:

$$\varphi_{k,i} = w_{k,j-1} + \mu_k u_{k,j}^T (d_{k,i} - u_{k,i} w_{k,i-1}) \tag{8}$$

μ_k represents the step of node k.

The swap phase swaps the estimated value $\varphi_{k,i}$ between all nodes k and their neighbors.

In the fusion phase, the estimated value $\varphi_{k,i}$ of neighbor node L is fused according to the weight to get the fused estimated value $w_{k,j}$. The update equation in the fusion phase is: $w_{k,j} = \sum_{l=1}^{N} c_{l,k} \varphi_{l,i}$

Diffusion LMS algorithm has two implementations, ATCDLMS and CTADMLMS, of which the order of adaptive phase and fusion phase is different. The difference is that the CTADLMS algorithm updates the fusion estimates by selecting the iteration results of the nodes themselves, while the ATCDLMS algorithm updates the fusion estimates based on the merged results. The difference is that ATCDDLMS has lower steady-state error and faster convergence than CTADLMS. Therefore, the algorithm in this paper will be implemented by ATC.

2.3 Distributed LMS algorithm

For non-linear problems, by introducing a kernel function, the features are mapped (usually with a higher dimension) and the sample points become linear separable in the mapped feature space. The purpose of this paper is also to make the parameter estimates obtained from the diffusion LMS algorithm linear separable. In SVM, its final classification model is writable.

$$f(x) = \sum_{i=1}^{m} \alpha_i \, y_i \phi(x_i)^T \phi(x) + b \tag{9}$$

Set H as the feature space, if there is a mapping:

$$\phi(x) : U \longrightarrow H \tag{10}$$

For all$x, z \in U$, function $K(x, z)$ satisfies the condition:

$$K(x, z) = \langle \phi(x), \phi(z) \rangle \tag{11}$$

K is called a kernel function. Where $\phi(x)$ is a mapping function and $\langle \cdot, \cdot \rangle$ is an inner product. That is, the kernel function inputs two vectors and returns the same value as two vectors mapped to ϕ .

The kernel technique is a technique that uses the kernel function to calculate $\langle \phi(x), \phi(z) \rangle$ directly and avoid calculating $\phi(x), \phi(z)$ separately, thus speeding up the operation of the kernel method. Based on the core technique, its final classification model can be written:

$$f(x) = \sum_{i=1}^{m} \alpha_i \, y_i K(x_i, x) + b \tag{12}$$

Generally, selecting a kernel function is the most important part of the kernel method. If the kernel function is not selected properly, a will not be able to map the input space to ϕ linearly separable feature space. Because we want to achieve the non-linear mapping, we use the Gaussian kernel function to improve the algorithm.[10]

2.4 The model and hypothesis of KDLMS

In the exchange phase and fusion stage, the diffusion LMS algorithm using kernel method is the same as the conventional diffusion LMS algorithm, but in the adaptive phase, I improve the LMS algorithm by the corresponding kernel method, so that it can accelerate the operation speed when accepting the nonlinear estimation value.[11]

Firstly, the data around node k are collected to obtain the fusion estimation sequence:

$$w_{k,i-1}(j) = \sum_{l \in N_k} c_{l,k} \varphi_{k,i-1}(j) \tag{13}$$

According to the fusion estimates, we can calculate the corresponding error series:

$$e_{k,i}(j) = d_{k,i}(j) - u_{k,i}(j) w_{k,i-1}(j) \tag{14}$$

According to the error sequence and fusion estimation sequence, the fusion estimation sequence is brought into the kernel function, where the kernel function is Gaussian kernel function, which is:

$$k(x, x') = e^{-\frac{\|x - x'\|^2}{2\sigma^2}} \tag{15}$$

The estimated value group at the next moment can be obtained:

$$\varphi_{k,i}(j) = \varphi_{k,i-1}(j) + \mu_k e_{k,i}(j) k(w_{k,i-1}(j), \cdot) \tag{16}$$

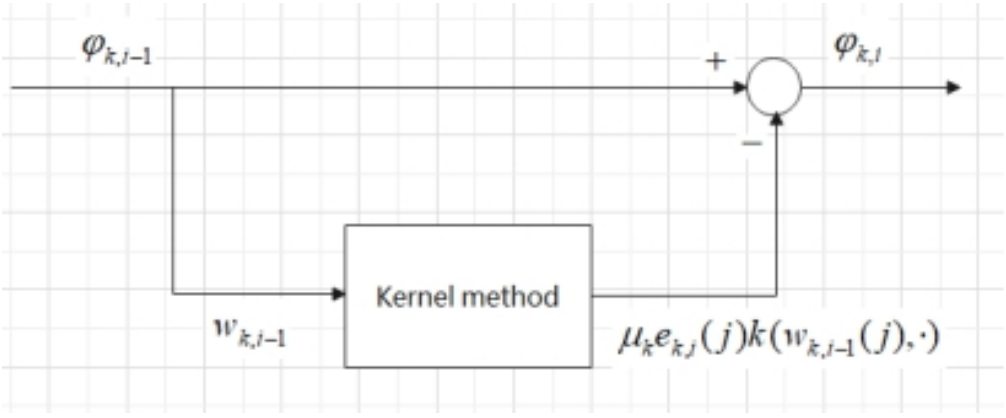

Fig. 1 Kernel incremental distribute strategy

According to Fig1.The linear model is improved by several algorithms to estimate an unknown parameter$ $\varphi_{k,i}$ which involves dealing with the dimension of input and output signals $\{u_{k,i}(j), d_{k,i}(j)\}$ under different nodes.[12] Generally speaking, k represents the spatial dimension of nodes, i represents the time dimension of nodes, and the signal $\{u_{k,i}(j), d_{k,i}(j)\}$ can be simply regarded as the excitation and corresponding of each local node. We assume that k, i are independent of each other to complete the following experimental analysis.

3. Simulation

In this section, we provide computer simulations to compare the theoretical performance with the simulation results. Although the analysis relies on the assumption of independence, all simulations use regression functions with shift structure to deal with the actual situation. Therefore, the regression coefficient is filled with: $u_{k,i}$=col{$u_k(i)$,u_k(i-1),...,u_k(i-M+1)}

$$u_{k,i}=\mathrm{col}\{u_k(i),u_k(i-1),\ldots,u_k(i-M+1)\} \quad (17)$$

In order to generate the performance curve, we conducted 100 independent experiments and averaged them. The steady-state curve is generated by 5000 iterations in the network learning process. We were interested in the amount of MSD, EMSE and MSE, and then averaged the last 1000 samples of the corresponding learning curve.

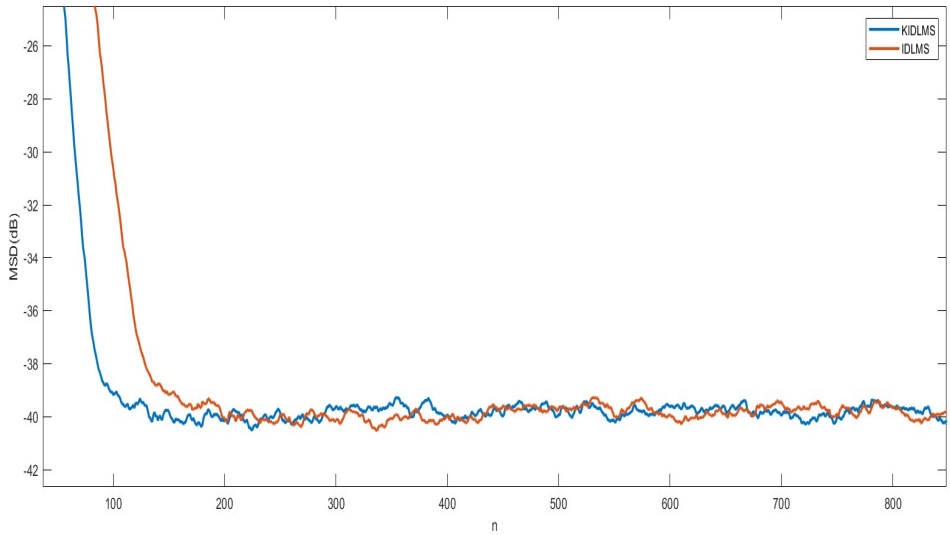

Fig. 2 The simulation of IDLMS and KIDLMS in MSD

According to figure 3, We can see that the incremental distributed least squares algorithm converges at the 150th iteration. The incremental distributed least squares algorithm based on kernel method can converge at the 100th iteration.

The same performance can be achieved when the entire network converges. A good algorithm design can use the kernel method to modify the original algorithm to obtain a new algorithm. By properly adjusting the weight processing algorithm for each node, a good performance balance can be achieved throughout the network. Higher accuracy can be achieved in a shorter time, which improves the utilization of each node and enhances the robustness of the data.

4. Conclusion

As a random gradient method used to solve nonlinear problems, KLMS algorithm can greatly improve its computational effect by transforming the update process into an internal product form. In the process of internal product transformation, it has good approximation ability because it retains infinite number of data features. In view of the characteristics of distributed LMS algorithm, it is combined with KLMS algorithm, which improves its convergence performance and prediction accuracy, and reduces the complexity of the algorithm.

The distributed minimum square algorithm has the characteristics of robustness, high flexibility, simple structure, easy to implement and stable performance, and the algorithm itself has been widely used in wireless sensor network, adaptive control and other systems, but there are still some problems. In this paper, the original distributed diffusion minimum square (DLMS) algorithm is kernel-methodized, so as to improve its sparse accordingly.

This paper deeply studies the principle process of adaptive filter, and supplements the deficiency of traditional DLMS algorithm.

In order to give full play to its performance in kernel method, the new algorithm in this paper. Detailed algorithm derivation and convergence analysis of distributed algorithm are carried out. Through detailed formula derivation, the convergence conditions and feasibility of the algorithm are proved from two aspects of mean and mean square.

Finally, matlab simulation software is used to compare the new algorithm with the traditional algorithm in noise environment; when the input signal is white signal, the convergence speed and convergence performance of various algorithms are compared. At the same time, the simulation results show that the new algorithm is effective. Compared with the distributed diffusion algorithm based on the mean square error criterion, the method not only solves the contradiction between steady-state maladjustment and convergence speed, but also increases the convergence speed of the system, which verifies the robustness and effectiveness of the algorithm.

References

[1] Xue, L.; Wang, J.L.; Li, J.; Wang, Y.L.; Guan, X. “Precoding design for energy effificiency maximization in MIMO half-duplex wireless sensor networks with SWIPT.“ Sensors 2019,19, 923.

[2] Chen, J.; Richard, C.;“ Sayed, A.H. Multitask diffusion adaptation over networks with common latent representations. “IEEEJ. Sel. Top. Signal Process. 2017, 11, 563C579.

[3] Ribeiro, A.; Schizas, I.D.; “ Roumeliotis, S.I.; Giannakis, G.Kalman fifiltering in wireless sensor networks. “IEEE ControlSyst. 2010, 30, 66C68.

[4] Mu, C.X.; Ni, Z.; Sun, C.Y.; He, H.B. “Data-driven tracking control with adaptive dynamic programming for a classof continuous-time nonlinear systems.“ IEEE Trans. Cybern.2017, 47, 1460C1470.

[5] Fernndez-Carams, Tiago M. and Froiz-Młguez, Ivn and Blanco-Novoa, Oscar and Fraga-Lamas, Paula “Enabling the Internet of Mobile Crowdsourcing Health Things: A Mobile Fog Computing, Blockchain and IoT Based Continuous Glucose Monitoring System for Diabetes Mellitus Research and Care“Sensors.2019 1424-8220

[6] Avetisov, Viacheslav and Bjoroey, Ove and Wang, Junyang and Geiser, Peter and Paulsen, Ketil Gorm “Hydrogen Sensor Based on Tunable Diode Laser Absorption Spec-troscopy “Sensors. 2019 1424-8220

[7] Grimble, M. Nonlinear minimum variance estimation for state dependent discrete-time systems. IET Signal Process. 2012,6, 379C391.

[8] Specht, D.F. Probabilistic neural networks. Neur. Netw. 1990,3, 109-118.

[9] Li, C.; Huang, S.; Liu, Y. Distributed TLS over multitask networks with adaptive intertask cooperation. IEEE Trans. Aerosp. Electron. Syst. 2016, 52, 3036C3052.

[10] Hua, J.H.; Li, C. Distributed jointly sparse Bayesian learning with quantized communication. IEEE Trans. Signal Inf. Process. Netw. 2018, 4, 769C782.

[11] M. Yukawa, Multikernel adaptive fifiltering, IEEE Trans. Signal Process. 60 (9) (2012) 4672C4682.

[12] W. Liu, P.P. Pokharel, J.C. Principe, The kernel least mean square algorithm, IEEE Trans. Signal Process. 56 (2) (2008)543C554.