A new mutton logistics tracking algorithm for Internet of things based on PSO and neural network

Minning Wu[1*], Fei You[1], Feng Zhang[1, 2]

[1]*School of Information Engineering, Yulin University, Yulin 719000, China*

[2]*School of automation, Northwestern Polytechnical University, Xi'an 710072, China*

Received 3 March 2014, www.cmnt.lv

Abstract

In order to improve the particle filtering precision and reduce the required number of particles, to solve the neural network training algorithm has slow convergence speed, easily falling into local optimal solution, proposed a target tracking algorithm based on PSO particle filter, using of Bayesian method to sample the prior information and coupled PSO algorithm. For the existence of intelligent wireless sensor network energy constrained sensor nodes, limited communication features, the PSO optimization is introduced into the distributed particle filter algorithm to solve the existing distributed particle filter network traffic load is heavy and node energy consumption of high disadvantage. Then, we propose a new particle filter algorithm based on PSO and neural integration the algorithm makes full use filter tracking historical information, combined with predictions of particle filter, the detection signal of the sensor nodes were isolated, thus achieving the target tracking. Simulation results show that the target tracking algorithm based on particle filter PSO and neural integration can use a smaller computational cost, multi-target tracking problem solving, and practical system to meet the demand.

Keywords: particle swarm optimization, particle filter, neural network, tracking algorithm, internet of things

1 Introduction

Particle filter (PF) is an estimate based on statistical filtering theory and Bayesian Monte Carlo idea [1]. It is based law of large numbers, using the Monte Carlo method to solve the integral Bayesian estimation problem, namely the sequential Monte Carlo methods. Theoretically applicable to any nonlinear stochastic systems under non-Gaussian background model represents the state space available, its accuracy can approach the optimal estimation. In recent years, it has been successfully applied in targeting [4], the Internet [5] and other fields. Target tracking is one of the typical applications of wireless sensor networks, target tracking can be used for cargo warehouse logistics tracking, traffic planning, monitoring wildlife and intruder monitoring. Since the sensor nodes with small size, low price, easy to deploy, such as communications and computing capabilities, while the network has self-organization, concealment and robust features, compared with the traditional tracking methods, wireless sensor networks for its good characteristics make up the deficiency of traditional tracking methods, making the wireless sensor network more suitable for locating and tracking moving targets. Therefore, research on target tracking based on WSN is of great theoretical and practical significance.

Wireless sensor network target tracking system consists of target detection, target location and target prediction and notification class into [6]. By activation of the target nodes of a dynamic near the tracking area, the node information by voice, vibration information, the image information of the target detection and location [7]. Then sends the position information to the dynamic tracking of the target area of the data processing centre, and the status of the completion of updating the target position in the data processing centre and the prediction.

Particle filter observation system as an effective way to solve the nonlinear, non-Gaussian problems, the target location and tracking system widely used. Since the particle filter complexity and computational requirements of data storage capacity, the particle filter algorithm must consider the impact the computational complexity, computational power nodes and node energy consumption and other factors in wireless sensor network based target tracking environment [8]. The traditional particle filter does not need to consider the energy consumption, computation complexity and data storage capacity and other issues in the operation of Sink node or terminal user centre. When the particle filter run-time dynamic tracking of distributed nodes in the cluster, you need to consider limiting the node energy and computing power, find a new particle filter operating mode, reducing energy consumption under the premise of protecting the target tracking accuracy, balance tracking accuracy and network energy consumption.

In this paper, from the basic theory of target tracking, in-depth analysis of the particle filter principle, on the basis of dynamic clustering, we propose a parallel

***Corresponding author** e-mail: 178578051@qq.com

extended Kalman particle filter algorithm (PEPF) and based on PSO particle filter (PSO-PF) and neural networks particle filter (PSO-PFNN).

2 Targeting particle filter algorithm

Motion model and measurement model is not only the design basis of objective filtering algorithm , and appropriate model can effectively reduce the computational complexity [9], improve tracking accuracy. For manoeuvring target tracking issues, we can use the following motion model Equation (1) and measurement model Equation (2) description:

$$x_t = f(x_{t-1}, v_{t-1}), \quad (1)$$

$$z_t = h(x_t, w_t), \quad (2)$$

where x_t represents the system state at time t, for target tracking, it is usually the target position, velocity or acceleration information. z_t represents the measured value of the target, v and w denote the process noise and measurement noise. $f()$ represents manoeuvrings target motion model, function $h()$ represents a measurement model, used to indicate the relationship between the measured value and the target states. General Equation (1) is established in the Cartesian coordinates, while Equation (2) was established in the spherical coordinates, in the solving process usually take the measurement equation is transformed to Cartesian coordinates. So, the target tracking problem is usually considered a highly nonlinear problem.

Measurement model used to describe the relationship between the target measurement data and status variables. It is generally refers to the measured data from the sensor output at all viewing measurement collection. These observations are usually attribute information (type, status, etc.) and a set of goals, motion information (position, velocity, etc.) as well as environmental information (time) correlation measurements. In the target tracking problem, the measured data is the main signal and the target motion state. As for the sensors, typically measured value and the target cannot be obtained directly related to the state, and the measured value contains mostly noise and clutter, so the general model to the measured data using the measured data for filtering, and the original measured value and the target the motion state information associated with it.

2.1 KALMAN FILTER ALGORITHM

Tracking Kalman filter for linear systems, and are assuming a Gaussian posterior probability distribution at any time can be determined by the mean and variance of the distribution. Kalman filter algorithm that is based on the following assumptions:

1) Target transfer noise from the v_{i-1} and measurement noise n_i is Gauss distribution with known parameters.

2) The state transition equation $f_i(x_{i-1}, v_{i-1})$ is a linear function of x_{i-1} and v_{i-1}.

3) The measurement equation $h_i(x_i, n_i)$ is a linear function of x_i and n_i.

Then, the Equations (1) and (2) can be written as:

$$x_t = Fx_{t-1} + v_{t-1}, \quad (3)$$

$$z_t = Hx_t + n_i, \quad (4)$$

where F and H are matrix definition of linear function, v_{t-1} and n_i are independent of each other, and the mean value is 0, Q and R are variance of a Gaussian distribution. Then it is according to Bayesian estimation theory, we can get a Kalman filter algorithm.

2.1.1 State Prediction

a) Target state and variance forecast:

$$x_{t|t-1} = Fx_{t-1|t-1}, \quad (5)$$

$$P^x_{t|t-1} = Q + FP^x_{t|t-1}F^T. \quad (6)$$

b) Predict the measured value and variance:

$$z_{t|t-1} = Hx_{t|t-1}, \quad (7)$$

$$P^z_{t|t-1} = R + FP^z_{t|t-1}H^T_t. \quad (8)$$

c) Correlation matrix is a priori predictions:

$$P^{xz}_{t|t-1} = P^x_{t|t-1}H^T_t. \quad (9)$$

2.1.2 Measurement update

a) Calculating the Kalman gain:

$$K_t = P^{xz}_{t|t-1}[P^z_{t|t-1}]^{-1} = P^x_{t|t-1}H^T_t[R + HP^{xz}_{t|t-1}P^x_{t|t-1}H^T_t]^{-1}. \quad (10)$$

b) Using of time t measured values, update the target state prediction and estimation variance:

$$x_{t|t} = x_{t|t-1} + K_t[z_t - z_{t|t-1}], \quad (11)$$

$$P^x_{t|t} = P^x_{t|t-1} - K_t P^z_{t|t-1} K^T_t = (1 - K_t H_t)P^x_{t|t-1}. \quad (12)$$

Since the Kalman filter algorithm is simple, small amount of calculation has been widely used in target tracking. Kalman filter algorithm is a Bayesian filter theory of linear, Gaussian distribution analytical results obtained under the assumption, therefore, is the best estimate of the Kalman filter algorithm for linear Gaussian filter under the concept.

2.2 PARTICLE SWARM OPTIMIZATION ALGORITHM

PSO algorithm [10] is an operation based on the fitness of particles, Swarm intelligence generated by the cooperation and competition between particles to guide the optimization search. The particle as a no weight and volume of the particles. In the *n-dimensional* space, it is keeping a certain speed flight, flight speed dynamically adjusted by the flying experience of individuals and groups of flying experience. PSO algorithm is the use of the substance of the information itself, the next iteration of individual extreme position information and global information on these three kinds of extreme value information to guide particles.

Standard PSO algorithm is to calculate the fitness of each particle *i* in iterative process, by tracking individual extreme p_i and p_g to update their global minimum. Particle *i* is to update its velocity and position values according to the following two equations:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 [P_i(t) - x_i(t)] + c_2 r_2 [p_g(t) - x_i(t)], \quad (13)$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \quad (14)$$

where $V_i(t)$ is the particle *i* velocity at time *t*, ω is the inertia weight, $x_i(t)$ is a position of the particle at time *t*, r_1, r_2 is a random number between (0, 1), c_1, c_2 is a learning factor, usually $c_1 = c_2 = 2$. It can significantly improve the performance of the PSO algorithm, it is given:

$$\omega = \omega_{\min} + (iter_{\max} - iter) \times (\omega_{\max} - \omega_{\min}) / iter_{\max}, \quad (15)$$

where $\omega_{\min}$, $\omega_{\max}$ respectively the maximum and minimum weighting factor, *iter* is the current iteration number, $iter_{\max}$ is the total number of iterations. $\omega_{\max}$ is the initial inertia weight; $\omega_{\min}$ is the last inertia weight; $t_{\max}$ is the maximum number of iterations. Flight speed is $v_i \in [-V_{\max}, V_{\max}]$, the constraint conditions to prevent particle speed missed optimal solutions, through the improvement of the algorithm further improves the global searching ability of particle swarm.

3 Particle filtering algorithm based on PSO and neural networks

3.1 PSO PARTICLE FILTER ALGORITHM

PSOPF algorithm can be expressed as follows: First, the introduction of the new measured values of the sampling process and the fitness function is defined as:

$$z_k \sim fitness = \exp\left[-\frac{1}{2R_k}(z_k - z_k^i)^2\right]. \quad (16)$$

Particle initialization: PSO $p(x_0)$ generated from the a priori probability density $\{x_0^i\}_{i=1}^N$, it all particle value is $\frac{1}{N}$, calculate the importance weights:

$$w_k^i = w_{k-1}^i \exp\left[-\frac{1}{2R_k}(z_k - z_{k|k-1})^2\right]. \quad (17)$$

Using the PSO algorithm according to the speed and position of each particle to update formula, the particles closer to the true state of constantly, it is given

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}[p_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}[p_{gj}(t) - x_{ij}(t)], \quad (18)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1). \quad (19)$$

Flow chart of particle PSO particle filter algorithm as shown in Figure 1.

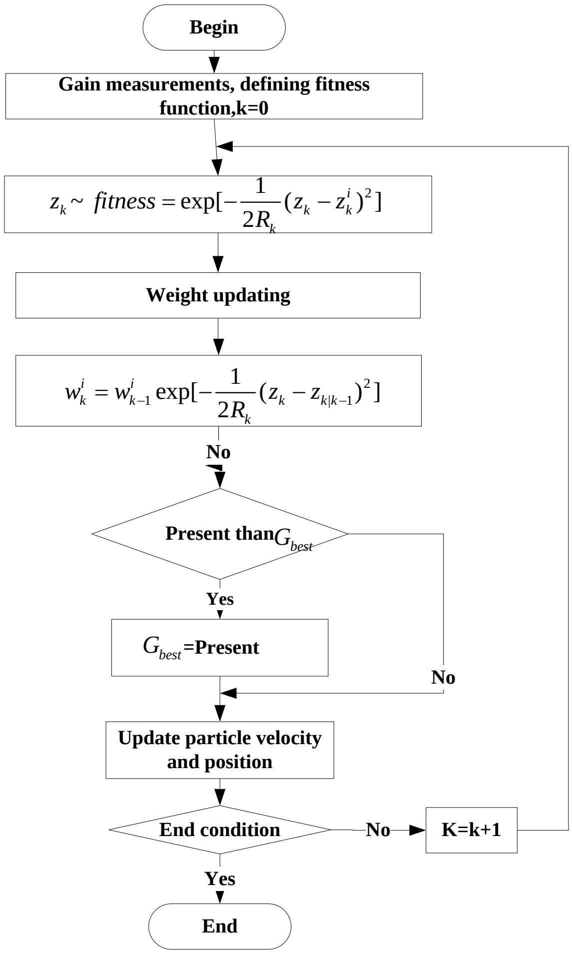

FIGURE 1 Flow chart of PSO particle filter algorithm

3.2 PSO NEURAL NETWORK OPTIMIZATION ALGORITHM

In recent years, with the increasing depth research, many researchers use swarm intelligence optimization algorithm to optimize the training process of neural networks [11]. Because of swarm intelligence optimization algorithm has strong global convergence ability and robustness, and the characteristics of the problem without the help of information, such as derivatives and other information, therefore, will not only be able to play a combination of both the generalization ability of neural network mapping, but also can improve the convergence speed and learning ability. Among them, PSO algorithm is used as a simple and effective study based on a random search algorithm is shown to have great potential in terms of optimization of neural networks.

PSO neural network optimization (PSO-NN) algorithm is used to replace the traditional parameters of PSO training algorithms such as BP to optimize NN. Each particle is a vector that represents a group of parameters, process the process for the global optimal value is to obtain the optimal parameters.

The training error is used to calculate the fitness value $f(x)$, it is given:

$$f(x) = \frac{1}{1 + \frac{1}{2n}\sum_{k=1}^{n}(y_k - t_k)}, \tag{20}$$

where, k is the number of samples, y_k is the actual output value, t_k is the output value. When it reach the maximum number of iterations or target error, the program terminates, and get the global optimum value that a group of optimal parameters. Algorithm is detailed as shown in Figure 2.

The initial particle swarm PSO algorithm standard position is randomly generated, causing the initial search space of uncertainty. Set the priori information coupling problem inherent in the object into the PSO, it can narrow the search space. For a large sample of the data set, the posterior probability is calculated first plurality of sets of values of weights in the particulate composition that is under the given conditions. Then initializes the value according to the probability distribution of particle position, thereby reducing the value of the learning process to modify the magnitude of particulates, to improve the learning speed.

Get the priori information is calculated as follows: set the neural network weights is $W = (w_1, w_2, ..., w_n)$. Sample points are $D = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$, where n and m are respectively the weight and the number of sample points, The position of a particle of W value in the sample points of conditional probability is:

$$P\left(\frac{W}{D}\right) = \frac{P(D/W)P(W)}{P(D)}. \tag{21}$$

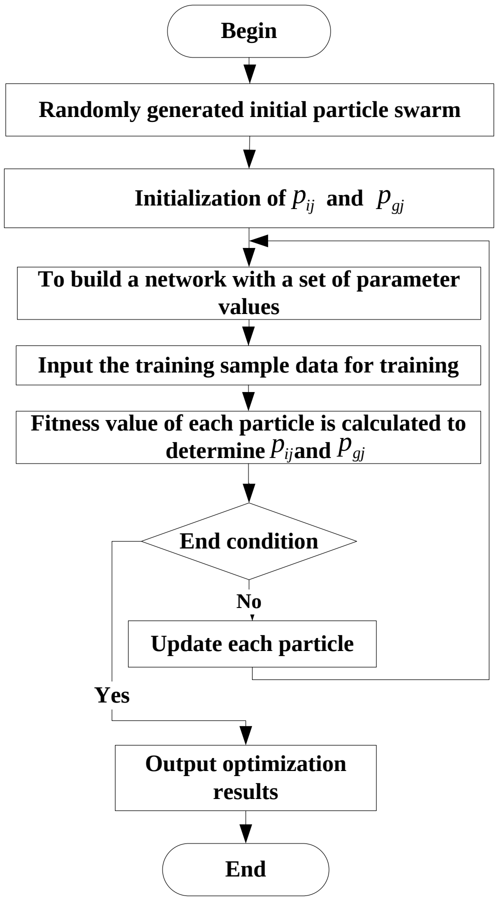

FIGURE 2 PSO neural network optimization algorithm

$P(W)$ is the particle position value of W prior distribution. $P(D/W)$ is the likelihood function, $P(D) = \sum_i P(D/w_i)P(w_i) = 1$ is a normalization factor.

In the absence of a priori information, if the W is too small, then all the incentive function are almost in a linear part, will also reduce the speed of convergence. The general consensus is that W obeys the exponential distribution, the equation is as follows:

$$P(W) = \frac{1}{Zw(\alpha)}\exp(-aEw). \tag{22}$$

The $Zw(\alpha)$ is a normalization factor, to ensure $\int P(W)dW = 1$, $Zw = \int \exp(-\alpha Ew)dW$, a is a control parameter of weight distribution form, Ew is a kind of error function. General Ew as shown in the following equation:

$$Ew = \frac{1}{2}\|W\|^2 = \frac{1}{2}\sum_{i=1}^{n} w_i^2. \tag{23}$$

Coupled Bayesian prior information using PSO algorithm combines the former BP algorithm to train feed forward neural networks (PSO-BPNN), the algorithm detailed steps can be divided into four stages.

3.2.1 PSO-BPNN initialization step

Step 1: generate training samples

$D = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$;

Step 2: determine the number and network target error each neuron;
Step 3: determine the relevant parameters of PSO: the number of particles, the dimensions of each particle, inertia weight, acceleration factor, the maximum number of steps, such as iteration.

3.2.2 Coupling priori information

Step 1: randomly initialize each particle velocity and position;
Step 2: according to the formula (22) modify each particle's position, that is used to calculate the $P(D/W)$ update particle position value.

3.2.3 Using a priori information coupled PSO training Neural Network Model

Step 1: generate particles using a priori information to adjust the position of the initial value $X_i(0)$;
Step 2: calculate the fitness value $f(X_i)$ of each particle;
Step 3: comparing each individual extreme fitness value and the size of the particles, if the former than the latter, making it a new personal best;
Step 4: If the particle swarm best individual extreme due to the current global optimum value, write down the number of particles, adjust its position, then the optimum value as the new global;
Step 5: Update the velocity and position of each particle;
Step 6: determine whether the PSO stop iteration condition. If yes, stop the search, note the global optimum value, go to the fourth stage; otherwise go to step2.

3.2.4 BP neural network re-training

Step 1: PSO iteration of the global optimal value obtained corresponds to the BP neural network weights and thresholds;
Step 2: Use BP neural network algorithm to continue training until it reaches the target error.

4 Experiment and analysis

In order to prove that the improved classification algorithm results, using three kinds of algorithm PEPF, PSO-PF and PSO-PFNN in MATLAB 13.0 environment classified simulation.

Selected data sets are the breeding process, the production process and sales process for experimental details as shown in Table 1.

TABLE 1 The selected data set

Dataset	Samples number	Attributes number
breeding process	500	12
production process	1240	8
sales process	560	24

Based on the experience and experimental test case, the relevant parameters are set as follows: The number of particles $n = 100$, inertia weight $\omega = 0.92$. Neural network output layer weights in the range [-1, 1], the maximum iteration number is 5000.

Preclude the use of tri-fold cross-validation method to select the training and test samples. Before classification, the use of SNR method attributes data set dimensionality reduction. Relationship dimensions and classification error rate shown in Figure 3. It can be seen that the PSO-PFNN error rate compared to other three algorithms in most dimensions are the lowest.

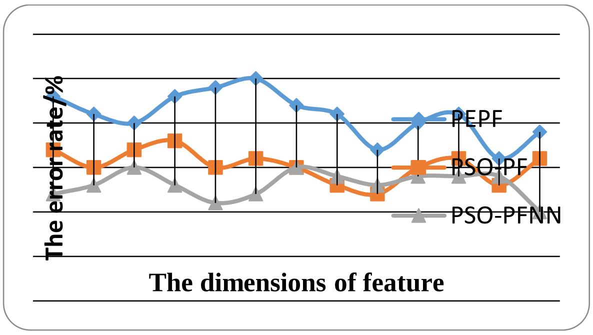

FIGURE 3 The relationship dimensions error rate

Three algorithms average classification accuracy on three data sets shown in Table2.

TABLE 2 Three algorithms average classification accuracy

Dataset	PEPF	PSO-PF	PSO-PFNN
breeding process	93.34%	95.12%	96.52%
production process	81.61%	83.31%	88.51%
sales process	73.12%	75.22%	79.68%

As can be seen from Table 2, both of which are experimental data sets for, PSO-PFNN average classification accuracy rate is the highest in the four algorithms, proved coupling prior information PSO-BPNN algorithm effectively improve the classification accuracy. These results prove that the use of a priori information Bayesian large sample extracted from a centralized data to improve search efficiency PSO algorithm is effective. Because PSO can more accurately search the global optimum value, so that the accuracy of the classification model trained network has been improved.

5 Conclusions

In this paper, the actual demand for the background, expand the research work for the lamb traceability network target tracking problem. Mainly is based on particle filter algorithm as the centre, research on distributed address tracing target tracking. Analysed the characteristics of a large sample and small samples, respectively, combined with Bayesian theory is to obtain the corresponding representation of a priori information, and then called together into the PSO algorithm. The experimental results show that for any type of sample set, the guiding of prior information, which accelerates the convergence speed and improve the classification accuracy rate of neural network model.

Acknowledgments

This work is partially supported by Natural Science and Technology Project Plan in Yulin of China **(Ny13-10)**, Natural Science Basic Research Plan in Shaanxi Province of China **(2011KW-27)**, and Thanks for the help.

References

[1] Jain A K, Duin R P W, Mao J 2000 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(1) 4-37

[2] Zhang Y-H, Zhang F 2013 A New Time Synchronization Algorithm for Wireless Sensor Networks Based on Internet of Things *Sensors and Transducers* **151**(4) 95-100

[3] Ji Y, Zhang F 2013 Wireless Sensor Traceability Algorithm Based on Internet of Things in the Area of Agriculture *Sensors and Transducers* **151**(4) 101-6

[4] Sperduti A, Starita A 1993 Speed up learning and network optimization with extended back propagation *Neural Networks* **6** 365-83

[5] Lengelie R, Denceux T 1996 Training MLPs layer-by-layer using an objective function for internal representations *Neural Networks* **9**(1) 83-97

[6] He S, Wu Q H, Wen J. Y, Saunders J R, Paton R C 2004 A particle swarm optimizer with passive congregation *Biosystems* **78**(1-3) 135-47

[7] Monson C K, Seppi K D 2004 The Kalman Swarm: A New Approach to Particle Motion in Swarm Optimization *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004)* Seattle Washington 140-50

[8] van den Bergh F, Engelbrecht A P 2006 A study of particle swarm optimization particle trajectories *Information Sciences* **176**(8) 937-71

[9] Ratnaweera A, Halgamuge S, Watson H C 2004 *IEEE Transactions on Evolutionary Computation Conference* **8**(3) 240-55

[10] van den Bergh F, Engelbrecht A P 2004 *IEEE Transactions on Evolutionary Computation Conference* **8**(3) 225-39

[11] Zahara E, Hu C-H 2008 Solving constrained optimization problems with hybrid particle swarm optimization *Engineering Optimization* **40**(11) 1031-49

Authors

Minning Wu, born in July, 1984, China

Current position, grades: Lecturer in Yulin University.
University studies: MS degree in Computer Application at Xidian University in 2010.
Scientific interests: computer application in agriculture, mobile internet technology.

Fei You, born in October, 1970, Yulin, Shanxi, China

Current position, grades: Professor at Yulin University.
University studies: PhD degree in Mathematical science from Beijing Normal University in 2005.
Scientific interest: fuzzy mathematics and artificial intelligence.

Feng Zhang, born in June, 1980, Yulin, Shanxi, China

Current position, grades: associate professor in Yulin University.
University studies: MS degree in Computer science from Xidian University in 2009.
Scientific interest: cloud integrated manufacturing technology, the modeling of complex systems, internet of things applications.