International Journal of Control and Automation
Vol. 8, No. 9 (2015), pp. 219-228
http://dx.doi.org/10.14257/ijca.2015.8.9.22

Bayesian Network Structure Learning Method with Insufficient Data Based on Cuckoo Search Algorithm with Cauchy Mutation

Wang Bo[1*], Zhang Jian-fei[2], Du Xiao-xin[2] and Liu Yan-ju[1]

[1]Computer Center, Qiqihar University, Qiqihar Heilongjiang 161006, China
[2]College of Computer and Control Engineering, Qiqihar University, Qiqihar Heilongjiang 161006, China
wangbo_theone@163.com

Abstract

Aiming at the cuckoo search algorithm (CSA) with disadvantages of slow convergence speed, getting into local extremum easily and low accuracy, we put forward cuckoo search algorithm with cauchy mutation(CCSA). For Bayesian networks(BNs) structure learning with insufficient data, we propose data completion method and Bayesian network structure learning with insufficient data based on CCSA(BNSL-ID-CCSA). In BNSL-ID-CCSA, firstly, we adopt K_2 metric as evaluation measure for learning Bayesian networks from data. Secondly, we use expectation maximization(EM) algorithm and CCSA to make BNSL-ID-CCSA quickly and accurately converge to the global optimal solution. The experimental results show that BNSL-ID-CCSA has strong learning ability and good stability.

Keywords: *cuckoo search; insufficient data; Bayesian networks; structure learning; EM algorithm*

1. Introduction

Bayesian networks(BNs)[1-2]are important tool for uncertainty knowledge representation and inference, Bayesian network structure learning is a research hotspot in the field of data mining in recent years. At present, most of the existing algorithms for Bayesian network structures learning can only be carried out under the sufficient data, but data is often lost or missing in the real world, the Bayesian network structure learning with insufficient data has more important theoretical significance and practical value. In recent years, the development and maturity of intelligent bionic algorithm greatly promotes the Bayesian network structure learning, Bayesian network structure learning algorithms based on artificial bee colony algorithm(ABCA), ant colony algorithm(ACA) and artificial fish swarm algorithm(AFSA) have been proposed, these algorithms can only make structure learning under sufficient data, the study on Bayesian network structure learning with insufficient data are little.

In this paper, we use cauchy mutation [3]to improve the cuckoo search algorithm (CSA)[4], and then put forward cuckoo search algorithm based on cauchy mutation (CCSA). CCSA can speed up the later evolution process, improve the precision and avoid falling into local extremum. For Bayesian network structure learning with insufficient data, we come up with Bayesian network structure learning method with insufficient data based on CCSA(BNSL-ID-CCSA). Firstly, we adopt K_2 metric as evaluation measure. Secondly, we use expectation maximization(EM) algorithm to get mathematical expectation of each

ISSN: 2005-4297 IJCA

statistical factor based on the current optimal BN structure. Thirdly, we use posterior probability getting by reasoning to execute random sampling, and insert the relevant data, so insufficient data is converted into sufficient data. Finally, we use CCSA to continuously optimize structure of BN, the BNSL-ID-CCSA eventually converge to global optimal solution.

2. Cauchy Mutation Suitable for CSA

Definition 1 (*mutation control factor* $\Re$) $\Re$ indicates the degree of mutation effect, as you see in formula(1).

$$\Re = \frac{-i^2 + I + 1}{I} \qquad i = (1, 2, \ldots, \sqrt{I}) \tag{1}$$

Among them, I is max iterations, at first iteration the value of $\Re$ is 1, and at last iteration the value is $1/I$, intermediate values decrease successively in the value of $1/I$. When the value of $\Re$ is 1, mutation plays the biggest role, however, when the value of $\Re$ is $1/I$, mutation plays the smallest role.

Definition 2 (*cauchy mutation*) Apply cauchy mutation on cuckoo's nest position (See formula (2)).

$$X_i^{(i)M} = X_i^{(i)} + X_i^{(i)} \times Cauchy(0,1) \times \phi \tag{2}$$

Among them, $Cauchy(0,1)$ is standard cauchy distribution, ξ is a random variable that obeys uniform distribution in [0,1]. The generating function of random variable of $Cauchy(0,1)$ is $\eta = \tan[(\xi - 0.5)\pi]$, n is the number of cuckoo's nest and ϕ is coefficient of mutation effect, usually value is 0.618.

In this paper we add random disturbance item based on cauchy distribution to formula (2) on the basis of $X_i^{(i)}$, we make full use of interference of the current population information, the degree of interference decays gradually with the attenuation of mutation control factor $\Re$. So formula (2) can make optimal location of nest jump out of local extremum point and converge to the global extremum point, at the same time the convergence speed is improved. In addition, cauchy mutation has stronger ability of disturbance than gaussian mutation, through cauchy mutation we can obtain the global optimal solution of nest's position more quickly than gaussian mutation.

3. Cuckoo Search Algorithm with Cauchy Mutation

3.1. Description of Cauchy Mutation on CCSA

CSA uses Lévy flight mode to generate new position of nest, but this method is easy to cause some defects, such as slow convergence speed, falling into local extremum and low precision of optimization. So after the ith iteration completes, we should not let the nest position directly into the next iteration, but should determine whether the nest position meets mutation condition. When the optimal location of nest changes little for two consecutive times, we consider it satisfy the mutation condition and then perform cauchy mutation, otherwise directly enter into the next iteration. When the logical value of formula (3) is true, start the cauchy mutation, otherwise does not perform.

$$\frac{\left|Fit(X_i^{(t)}) - Fit(X_i^{(t-1)})\right|}{Fit(X_i^{(t)})} \le \Omega \quad \text{and} \quad \frac{\left|Fit(X_i^{(t-1)}) - Fit(X_i^{(t-2)})\right|}{Fit(X_i^{(t-1)})} \le \Omega \tag{3}$$

In formula(3), $Fit(X_i^{(t)})$ is the fitness value of optimal nest position in the *t*th iteration, $Fit(X_i^{(t-1)})$ is the fitness value of optimal nest position in the (*t-1)*th iteration, $Fit(X_i^{(t-2)})$ is the fitness value of optimal nest position in the (*t-2)*th iteration, Ω is threshold(in this paper the value is 0.008).

3.2. Description of CCSA

Step 1 Initialize the nests, randomly generate the position of nests $X_i^{(i)} = (X_1^{(i)}, X_2^{(i)}, \ldots, X_n^{(i)})$, n is the number of nests.

Step 2 According to fitness function $Fit()$ to calculate initial fitness value of each nest.

Step 3 The owner of each nest improves their own nest through Lévy flight mode, calculates the fitness value of each improved nest, compares to the former fitness value, and retains better quality nest according to greedy heuristics. The owner of nest improves the nest according to the formula (4).

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus s \tag{4}$$

Among them, t indicates the current number of iterations, α is control parameter of step length, the value obeys the standard normal distribution, $\oplus$ indicates the point to point multiplication, s is search path of lévy flight, that is the step length of flight, see formula(5).

$$S = 0.01 \times \frac{\mu}{|\upsilon|^{1/\beta}} \times (g_{best} - X_i^{(t)}) \tag{5}$$

In formula(5), coefficient of 0.01 is typical flight scale in the Lévy fight, μ and υ obey uniform distribution, that is $\mu \sim N(0, \delta_\mu^2)$, $\upsilon \sim N(0, \delta_\upsilon^2)$.

$$\delta_{\mu} = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\times\beta\times 2^{(\beta-1)/2}} \right\}^{1/\beta}$$

Among them, Γ expresses standard gamma function, $\delta_{\upsilon} = 1$, in this paper we set the value of β is 2/3, g_{best} indicates the nest with the current optimal fitness value.

Step 4 Improve the nest with poor fitness value according to formula(6) with probability P (see Table 2).

$$X_i^{(t+1)} = X_i^{(t)} + rand \times (X_j^{(t)} - X_i^{(t)}) \tag{6}$$

Among them, *rand* is a random number between [0, 1], $X_j^{(t)}$ is a nest near the $X_i^{(t)}$.

Step 5 Compare the improved nest with the nest which possesses the current optimal fitness value.

Step 6 Determine whether meet mutation start condition as shown in formula (3), if meet turn to step 7, otherwise, turn to step 8.

Step 7 Perform cauchy mutation according to formula (2).

Step 8 Perform step2~step7 until the maximum number of iterations, output the nest of optimum fitness value.

3.3. CCSA Performance Analysis

In order to verify the performance of the proposed CCSA, we choose four typical multimodal functions[5-7](see Table 1) as test cases. The parameter settings of CCSA are shown in Table 3. Experiments are performed on the computer with Intel Core i3, 4 GB of memory, Windows 7 operating system, and implemented by MATLAB programming.

Table 1. Test Functions

	Function name	Test function	Dimension	Domain of variable	Optimum value	Objective precision
f_1	Schaffer	$f_1 = 0.5 + \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2}$	2	$-100 \le x_1, x_2 \le 100$	0	0.002
f_2	Griewank	$f_2 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\frac{x_i}{\sqrt{i}} + 1$	30	$-100 \le x_i \le 100$	0	200
f_3	Rastigrin	$f_3 = \sum_{i=1}^{n}(x_i^2 - 10\cos 2\pi x_i + 10)$	10	$-5.12 \le x_i \le 5.12$	0	20
f_4	Rosenbrock	$f_4 = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	30	$-30 \le x_i \le 30$	0	7000

Figure1-4 are evolutionary curves about the logarithm of average fitness of CSA and CCSA performing 20 times on the functions in Table1. From Figure1-4 we can see, CCSA has faster convergence speed and higher precision than CSA. So CCSA has advantages in accuracy, convergence speed and robustness.

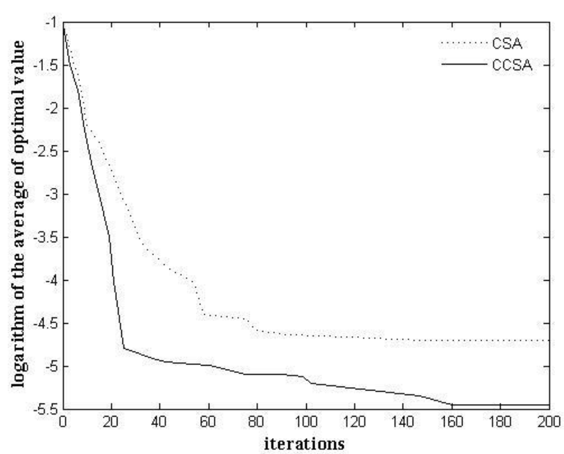

Figure 1. Evolutionary Curve of Function Schaffer Figure

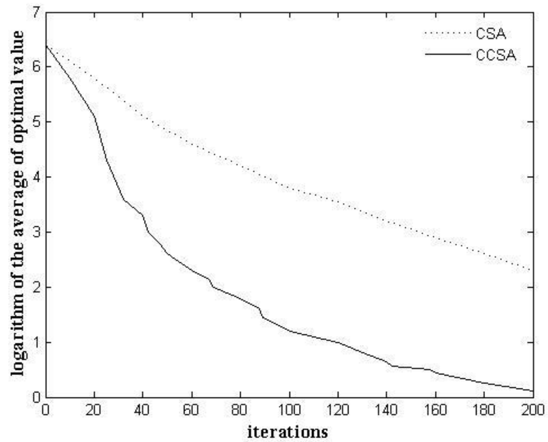

Figure 2. Evolutionary Curve of Function Griewank

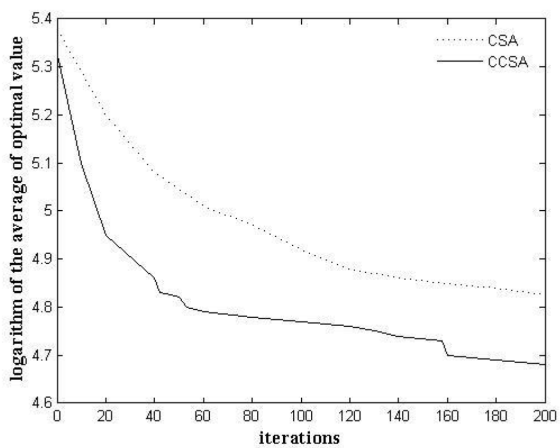

Figure 3. Evolutionary Curve of Function Rastigrin

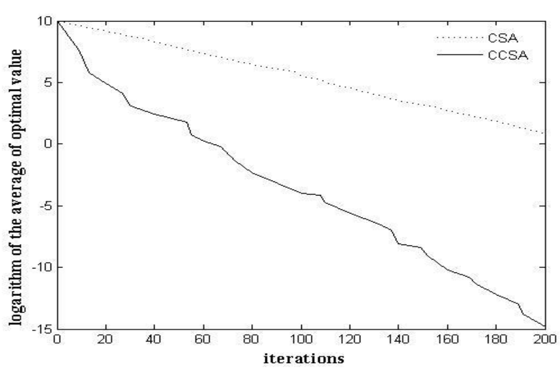

Figure 4. Evolutionary Curve of Function Rosenbrock

4. Bayesian Network Structure Learning with Insufficient Data Based on CCSA

In the BNSL-ID-CCSA, we preset parameters as follows: $I = < I^{obs}, I^{mis} >$ is the original insufficient data set, I^{obs} indicates observation data, I^{mis} indicates missing data, S^{optimal} is the current optimal structure, η^{optimal} is current best network parameters getting by using EM algorithm to estimate I; I_{com} is sufficient data obtained after filling the missing values.

4.1. Data Completion in BNSL-ID-CCSA

Achieving data completion is through EM statistical parameters and random sampling interpolation. Firstly, we use the EM algorithm to estimate the parameters of the current optimal BN structure. Secondly, we complete data completion through random sampling interpolation. EM algorithm contains expectation (E-step) and estimation (M-step) two steps.

In the E-step, we estimate the sufficient statistical factor of unobserved data using probabilistic inference. Sufficient statistical factor is shown in formula(7).

$$E(Q_{ijk} \left| I^{obs}, \eta^{\text{optimal}}, S^{\text{optimal}} \right.) = \sum_{l=1}^{Q} p(X_i = k, \prod(X_i) = j \left| i_l^{obs}, \eta^{\text{optimal}}, S^{\text{optimal}} \right.) \quad (7)$$

In formula(7), i_l^{obs} indicates variable observed from the l th instance, and it meets formula(8).

$$p(X_i = k, \prod(X_i) = j \left| i_l^{obs}, \eta^{\text{optimal}}, S^{\text{optimal}} \right.) = \begin{cases} 1, & X_i, \prod(X_i) \text{observable and } X_i = k \ \& \prod(X_i) = j \\ 0, & X_i, \prod(X_i) \text{observable and } X_i \neq k \ || \prod(X_i) \neq j \\ p_{<\eta^{\text{optimal}}, S^{\text{optimal}}>}(X_i = k, \prod(X_i) = j \left| i_l^{ons} \right.), & other \end{cases} \quad (8)$$

In formula(8), $p_{<\eta^{\text{optimal}}, S^{\text{optimal}}>}(X_i = k, \prod(X_i) = j \left| i_l^{ons} \right.)$ is obtained by BN probability reasoning based on the current network structure and parameters.

In M-step, Parameter η_{ijk} is shown in formula (9).

$$\eta_{ijk} = \frac{E(Q_{ijk} \left| I^{obs}, \eta^{\text{optimal}}, S^{\text{optimal}} \right.)}{\sum_k E(Q_{ijk} \left| I^{obs}, \eta^{\text{optimal}}, S^{\text{optimal}} \right.)} \quad (9)$$

EM algorithm iterative executes E-step and M-step based on the current optimal network structure S^{optimal}, finally the algorithm converges to an optimal parameters of the network structure.

4.2. Description of BNSL-ID-CCSA

The description of BNSL-ID-CCSA is as follows.

Step 1 Use EM algorithm to estimate I to get the current best network parameters $\eta^{optimal}$;

Step 2 Use the conditional probability to execute random sampling, and then fill the missing value to get sufficient data I_{com};

Step 3 Use CCSA to get the optimal solution of the current data completion process, if meet the end condition(optimal score no longer changes in continuous ω generations), turn to step 5; else turn to step 4;

Step 4 Select current optimal Bayesian network for the next data completion, CCSA optimization process. Turn to step 2;

Step 5 Output the optimal Bayesian network structure.

4.3. BNSL-ID-CCSA Performance Analysis

The experiment parameter setting is shown in Table 2. In this paper in order to test the performance of BNSL-ID-CCSA we adopt standard ALARM network generator (http://genie.sis.pitt.edu/) [8]to generate sufficient data samples containing 1000, 2000, 3000 cases respectively. In each sufficient data sample we use MCAR mechanism to delete the value of some variables randomly, missing data respectively accounts for 10%, 20%, 30% of the total sample, thereby we obtain the inadequate training data set. Perform BNSL-ID-CCSA five times on inadequate training data set, the study results are shown in Table 3. In Table 3 A is the number of additional arcs, D is the number of accidental removal arcs, R is mis-directed arcs, these three indicators indicate structure differences between the optimal network structure we learned and the standard structure. From Table 3 we can see, BNSL-ID-CCSA has stronger learning ability, in addition, structure differences and K_2 metric has no fixed relationship. In the case of the lower loss rate there are little changes in structure differences, this reflects the BNSL-ID-CCSA has good stability.

Table 2. Parameter Setting

Problem size n	The largest number of iterations I	Parameter p	Parameter ω
20	200	0.25	5

Table 3. Learning Effect of BNSL-ID-CCSA

sample size	statistical item	information loss rate		
		10%	20%	30%
1000	K_2	-4 566.40±61.23	-4 563±38.21	-4 722.68±16.86
	A.	8.01±1.36	7.42±0.30	8.01±1.36
	D.	1.63±0.24	1.42±0.20	2.01±0.00
	R.	3.06±1.94	1.81±0.48	3.25±0.66
2000	K_2	-8 598.32±49.68	-8 706.32±35.32	-8 930.26±28.48
	A.	5.19±0.79	4.18±0.20	7.90±1.20
	D.	1.00±0.00	1.58±0.22	2.03±0.30
	R.	1.21±0.48	0.82±0.36	2.25±0.82
3000	K_2	-12 735.90±12.68	-12 928±28.01	-13 432±22.46
	A.	3.78±0.96	3.28±1.07	6.68±0.72
	D.	1.00±0.03	1.00±0.04	2.00±0.00
	R.	1.50±0.89	1.25±0.46	9.56±0.86

5. Conclusions

In this paper, aiming at the disadvantages of basic CSA, we put forward cuckoo search algorithm with cauchy mutation(CCSA). For Bayesian network structure learning with insufficient data, we use EM algorithm and random sampling interpolation to realize data completion and use CCSA to optimize Bayesian network structure. Finally we put forward Bayesian network structure learning method with insufficient data based on CCSA(BNSL-ID-CCSA). The experimental results show that CCSA has good solving accuracy and speed and BNSL-ID-CCSA has good learning ability and stability.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No.61403222), the Natural Science Foundation of Heilongjiang Province of China (Grant No.F201333, F201439) and the Ministry of Education Research of Humanities & Social Science Youth fund in China (14YJC630188). The authors would like to thank the editors and reviewers of this manuscript for their very helpful suggestions.

References

[1] Z. Cai and S. S. Suns, "Identifying product failure rate based on a conditional Bayesian network classifier", Expert Systems with Applications, vol.38, no.5, **(2011)**, pp.5036-5043.

[2] L. P. H. Hung, "A data driven ensemble classifier for credit scoring analysis", Expert Systems with Applications, vol.37, no.1, **(2010)**, pp.534-545.

[3] K. T. Lan and C. H. Lan, "Notes on the distinction of Gaussian and Cauchy mutations", Eighth International Conference on Intelligent Systems Design and Applications, **(2008)**.

[4] X. S. Yang and S. Deb, "Cuckoo Search via Lévy flights", Proc.World Congress on Nature&Biologically Inspired Computing, IEEE Publications, **(2009)**; India.

[5] A. Layeb and S. R. Boussalia, "A novel quantum inspired cuckoo search algorithm for bin packing problem", Int'l Journal of Information Technology and Computer Science, vol.4, no.5, **(2012)**, pp.58-67.
[6] H. Q. Zhang and Y. Q. Zhou, "Self-adaptive step cuckoo search algorithm", Computer Engineering and Applica -tions, vol.49, no.10, **(2013)**, pp.68-71.
[7] E. Valian, S. Mohanna and S. Tavakoli, "Improved cuckoo search algorithm for global optimization", Int'l Journal of Communications and Information Technology, vol.1, no.1, **(2011)**, pp.31-44.
[8] C. H. Lo, Y. K. Wong and A. B. Rad, "Bond graph based Bayesian network for fault diagnosis", Applied Soft Computing, vol.11, no.2, **(2011)**, pp.1208-1212.