Feature Combination Method Integrating Denoising Autoencoder and BPSO and Its Application in Traditional Chinese Medicine (Postprint)

Authors: Canyi Huang, Du Jianqiang, Nie Bin, Zeng Qingxia, Zhipeng Zhu, Yu Fang

Date: 2018-06-19T00:00:00+00:00

Abstract

Discrete Binary Particle Swarm Optimization (BPSO) exhibits numerous advantages in various discrete optimization problems; however, it is susceptible to falling into local optima due to nonlinear problems, preventing it from obtaining the optimal feature subset. In contrast, Denoising Autoencoders can perform mapping and reconstruction through multi-layer nonlinear networks, demonstrating favorable processing efficacy for Traditional Chinese Medicine data. Therefore, a feature combination method fusing Denoising Autoencoders and BPSO is proposed. This method primarily employs Denoising Autoencoders to conduct nonlinear mapping of features, thereby forming an overcomplete basis, within which BPSO is subsequently utilized to search for the optimal feature subset. Clinical diabetes datasets and UCI datasets are respectively adopted for analysis and processing. Experimental results indicate that the feature combination method fusing Denoising Autoencoders and BPSO possesses favorable adaptability to Traditional Chinese Medicine clinical experimental data.

Full Text

A Feature Combination Method Fusing Denoising Autoencoder with BPSO and Its Application in Traditional Chinese Medicine

Huang Canyi, Du Jianqiang†, Nie Bin, Zeng Qingxia, Zhu Zhipeng, Yu Fang (School of Computer Science, Jiangxi University of Traditional Chinese Medicine, Nanchang 330004, China)

Abstract: The Discrete Binary Particle Swarm Optimization (BPSO) algorithm offers many advantages for various discrete optimization problems, but it

easily falls into local optimal solutions due to nonlinear issues, preventing it from obtaining the optimal feature subset. Denoising Autoencoders, however, can perform mapping and reconstruction through multi-layer nonlinear networks, demonstrating good effectiveness for Traditional Chinese Medicine data. Therefore, this paper proposes a feature combination method that fuses Denoising Autoencoder with BPSO. This method primarily utilizes the Denoising Autoencoder to perform nonlinear mapping of features to form a super-complete basis, then searches within this super-complete basis through BPSO, thereby obtaining the optimal feature subset. Clinical diabetes datasets and UCI datasets were used for analysis and processing. Experimental results demonstrate that the feature combination method fusing Denoising Autoencoder with BPSO exhibits good adaptability to clinical experimental data in Traditional Chinese Medicine.

Key words: Denoising Autoencoder; BPSO; Nonlinear; TCM

0 Introduction

Clinical experimental data in the field of Traditional Chinese Medicine typically exhibit characteristics of multiple components, multiple targets, and nonlinearity [1]. Due to the complexity of the data, strong correlations and redundancies exist among features. Therefore, traditional statistical analysis methods cannot be employed to elucidate the dose-effect relationships within the data, necessitating a data analysis method capable of addressing multivariate and nonlinear problems to provide technical support for researchers.

The Discrete Binary Particle Swarm Optimization (BPSO) algorithm, proposed by Kennedy and Eberhart [2,3], is an extended particle swarm optimization method widely applied in bioinformatics, knapsack problems, and image processing. However, BPSO easily falls into local optimal values during the feature selection process [4], preventing the algorithm from selecting the optimal feature subset.

The Denoising Autoencoder (DA), proposed by Vincent [5,6], is an improved method that adds noise to original data based on autoencoders and performs multi-layer nonlinear network learning. Through unsupervised layer-wise greedy training and systematic parameter optimization, it can extract and encode features with good robustness. Reference [7] proposes a self-adaptive discrete particle swarm algorithm that introduces a repulsion process to overcome premature convergence, yet the parameter settings for attraction and repulsion are difficult to determine. Reference [8] proposes a feature selection method based on the SVM-RFE-BPSO algorithm, which uses SVM-RFE to quickly remove some irrelevant features and then continues searching for the optimal feature subset with particle swarm optimization, but its parameter settings are also difficult to determine. Reference [9] proposes a hybrid particle swarm algorithm with Gaussian white noise perturbation that introduces a threshold for adaptively adjusting population diversity to prevent falling into local optima, yet the new

constraint conditions also make the optimal value extremely unstable.

Therefore, this paper combines and optimizes the Denoising Autoencoder with the Discrete Binary Particle Swarm Optimization algorithm. Through the three-layer network structure of DA, features undergo nonlinear mapping to make the input layer data as similar as possible to the output layer data [10,11], thereby forming a super-complete basis. BPSO is then used to search within this super-complete basis until the optimal feature combination is found. This algorithm can not only effectively remove redundant features but also prevent local optima, thus establishing an analysis model suitable for Traditional Chinese Medicine data.

1 DA-BPSO: A Discrete Binary Particle Swarm Algorithm Fused with Denoising Autoencoder

The Denoising Autoencoder [12] is a method that modifies autoencoders by combining robustness with corrupted inputs. Its basic idea is to first perform corruption processing, i.e., randomly set each value in the original input to 0, causing partial feature loss in some data, such as $E' = Binom \cdot E$. The corrupted data is then mapped through a mapping method: $Y = f_\theta(E')$, into a hidden layer representation Y. The hidden layer data Y is reconstructed using a reconstruction method: $Z = g_{\theta'}(Y)$, to form the output layer data Z. Through iterative training, the error function $L(E, Z)$ is minimized, thereby ensuring that Z approximates X as closely as possible.

The specific steps are as follows:

a) Data standardization preprocessing: raw data $X \rightarrow$ standardized data E.

b) Corruption operation: First, generate a corruption matrix $Binom$ based on binomial distribution (0-1), then use the corruption matrix to corrupt the input data, i.e., $E' = Binom \cdot E$.

c) Mapping and reconstruction: Use the Sigmoid activation function (Equation (1)) to map the corrupted data E' into hidden layer data Y, then reconstruct data Y to obtain output layer data Z.

d) Calculate cost function and optimize parameters: Use output layer data Z and input data E to calculate the cost function $L(E, Z)$, and optimize parameters by minimizing the cost function [14] using minimum mean square error. Through repeated iterative training, parameters in the network are updated to minimize the error function, thereby obtaining well-reconstructed data and forming a super-complete basis. The minimized mean square error is:

$$L(E, Z) = \frac{1}{n} \sum_{i=1}^{n} \|e_i - z_i\|^2$$

The loss function is:

$$L(E,Z)=\frac{1}{d}\sum_{k=1}^{d}[-e_k\log(z_k)-(1-e_k)\log(1-z_k)]$$

The weight update matrix is:

$$\begin{cases}W_1\leftarrow W_1-\varphi\cdot\frac{\partial L(E,Z)}{\partial W_1}\\B_1\leftarrow B_1-\varphi\cdot\frac{\partial L(E,Z)}{\partial B_1}\\W_2\leftarrow W_2-\varphi\cdot\frac{\partial L(E,Z)}{\partial W_2}\\B_2\leftarrow B_2-\varphi\cdot\frac{\partial L(E,Z)}{\partial B_2}\end{cases}$$

e) Initialize particle positions and velocities: Define the number of particles in the super-complete basis as N with dimension M, then the particle swarm Q can be represented as $Q=(Q^{(1)},Q^{(2)},Q^{(3)},\dots,Q^{(N)})$, where $Q^{(i)}=(q_i^1,q_i^2,\dots,q_i^M)$ represents each particle' s position. Particle initial values are randomly generated as 0 or 1 using binomial distribution. Each particle' s corresponding velocity V is defined as $V=(V^{(1)},V^{(2)},V^{(3)},\dots,V^{(N)})$, where $V^{(i)}=(v_i^1,v_i^2,\dots,v_i^M)$. Particle initial velocities are randomly initialized between [0,1].

f) Calculate fitness function values: For the features selected in each particle, and based on the data extracted from the super-complete basis, compose a new dataset. Use a classifier to classify this dataset and calculate accuracy $accuracy(Q^{(i)})$, thereby obtaining each particle' s fitness function value:

$$f(Q^{(i)})=A\cdot accuracy(Q^{(i)})-B\cdot n\_features(Q^{(i)})$$

where $n\_features(Q^{(i)})$ is the number of feature subsets (i.e., the number of 1s in each particle), and A and B are weight parameters (adjusted according to different datasets to achieve a trade-off between classification accuracy and feature subset size) with values in the range (0,1).

g) Record each particle' s historical optimal fitness function value $f_h(Q^{(i)})$ and the corresponding individual best position $Q_h^{(i)}=(q_{h,i}^1,q_{h,i}^2,\dots,q_{h,i}^M)$. Record all particles' optimal fitness function value $f_{wh}(Q^{(i)})$ and the corresponding global best position $Q_{wh}^{(i)}=(q_{wh,i}^1,q_{wh,i}^2,\dots,q_{wh,i}^M)$.

h) Update each particle' s position and velocity using the following formulas:

$$\begin{cases}V_i^{k+1}=w\cdot V_i^k+c_1\cdot\xi\cdot(Q_h^{(i)}-Q_i^k)+c_2\cdot\eta\cdot(Q_{wh}^{(i)}-Q_i^k)\\Q_i^{k+1}=\begin{cases}1, & \text{if } r\le \mathrm{Sig}(V_i^{k+1})\\0, & \text{otherwise}\end{cases}\end{cases}$$

where w is the coefficient maintaining original velocity, called inertia weight; c_1 and c_2 are weight coefficients for particles tracking their own historical optimum and the swarm's global optimum, respectively, typically set between [1,2] (set to 2 after multiple experiments); ξ and η are uniformly distributed random numbers in [0,1]; Q_i^k represents the value of the i-th particle at step k; t is the iteration number; r is a uniformly distributed random number; and $\mathrm{Sig}(v) = \frac{1}{1+\exp(-v)}$ is the activation function.

i) Repeat the optimization process until the optimal position is found, which represents the optimal feature combination, and the algorithm terminates.

The DA-BPSO construction process is shown in Figure 1 [Figure 1: see original paper].

2 Experiments

2.1 Data Description

The experimental data in this paper primarily come from clinical diabetes data from the Key Laboratory of Jiangxi University of Traditional Chinese Medicine and the Wine Quality and CASP datasets from UCI. The Wine Quality dataset contains 11 features and 1,600 samples; CASP has 9 features and 45,730 samples; the clinical diabetes experimental data comprises 16 features and 284 samples. The main features include: BMI (indicating overweight, obese, normal, and thin), enrollment HDLc, enrollment cholesterol, enrollment triglycerides (TG), enrollment insulin (oh), enrollment glycated hemoglobin, enrollment systolic blood pressure, enrollment fasting venous blood glucose, coptis grouping number, etc. Partial experimental data are shown in Table 1 .

2.2 Experimental Process and Results Analysis

To validate the effectiveness of DA-BPSO, we compared using all original features (hereinafter referred to as original features) with four strategies: feature combinations searched by traditional BPSO, feature combinations searched by Hybrid Particle Swarm Optimization (HDPSO), feature combinations searched by Self-Adaptive Discrete Particle Swarm Optimization (SADPSO), and feature combinations searched by DA-BPSO. The feature analysis results for the three datasets using these different strategies are as follows:

a) Clinical diabetes data: BPSO selected 8 features, HDPSO selected 11 features, SADPSO selected 8 features, while DA-BPSO selected 11 features after mapping to 30 dimensions during the DA operation (the mapped feature number was 30), as shown in Figure 2 [Figure 2: see original paper].

b) Wine Quality dataset: BPSO selected 4 features, HDPSO selected 7 features, SADPSO selected 5 features. Similarly, since DA processing mapped features to 20 dimensions (the mapped feature number was

20), DA-BPSO selected 12 features, as shown in Figure 3 [Figure 3: see original paper].

c) CASP dataset: BPSO selected 2 features, HDPSO selected 5 features, SADPSO selected 4 features. Similarly, since DA processing mapped features to 20 dimensions (the mapped feature number was 20), DA-BPSO selected 7 features, as shown in Figure 4 [Figure 4: see original paper].

To further analyze the effectiveness of the improved algorithm, this paper selected SVM as the classifier and used the accuracy of the training set (Train) and test set (Test) for comparison. All three datasets were randomly divided into training and test sets at a 7:3 ratio, i.e., 70% for constructing the learning training set and 30% for testing. Since the feature selection process used the optimal result from 1,000 iterations each time, to prevent disturbance from local optima, each strategy was run 10 times to select the optimal feature combination. The comparison results are shown in Table 2 .

Table 2: Comparison of Experimental Results Between Original Features and Four Strategies

Dataset	Original Features	BPSO	HDPSO	SADPSO	DA-BPSO
Wine Quality	Train: 0.5957, Test: 0.5267	Train: 0.5957, Test: 0.5267	Train: 0.5957, Test: 0.5267	Train: 0.5957, Test: 0.5267	Train: 0.6014, Test: 0.5933
Clinical Diabetes	Train: 0.3600, Test: 0.1056	Train: 0.3600, Test: 0.1056	Train: 0.3600, Test: 0.1056	Train: 0.3600, Test: 0.1056	Train: 0.3733, Test: 0.2978
CASP	Train: 0.7312, Test: 0.6415	Train: 0.7312, Test: 0.6415	Train: 0.7312, Test: 0.6415	Train: 0.7312, Test: 0.6415	Train: 0.7489, Test: 0.6900

According to the table above, using SVM as the classifier, the training set accuracies for original features are 0.5957, 0.3600, and 0.7312 respectively, while the test set accuracies are 0.5267, 0.1056, and 0.6415 respectively. To more intuitively display the experimental results, Figures 5 [Figure 5: see original paper] and 6 [Figure 6: see original paper] were plotted to show the accuracy fluctuations in training and test sets.

The result analysis shows that on the training set, original features perform better than BPSO and HDPSO. However, on the test set, performance is significantly worse, primarily due to the influence of redundancy among features on test set classification accuracy. Meanwhile, SADPSO shows slightly improved

accuracy on both training and test sets compared to HDPSO while reducing the number of features. DA-BPSO, despite having more features after mapping, demonstrates good accuracy performance compared to HDPSO and SADPSO, indicating strong robustness of the improved algorithm. Compared with original features, DA-BPSO shows better results, particularly on the clinical diabetes dataset where accuracy shows an upward trend (as shown in Figure 6 [Figure 6: see original paper]). In summary, the improved algorithm exhibits good adaptability to nonlinear clinical experimental data in Traditional Chinese Medicine, can prevent falling into local optima, and thus obtains optimal feature subsets.

3 Conclusion

This paper addresses the problem that traditional Discrete Binary Particle Swarm Optimization algorithms easily fall into local optimal solutions when dealing with nonlinear clinical data in Traditional Chinese Medicine, preventing them from obtaining optimal feature subsets. We propose a feature combination method that fuses Denoising Autoencoder with BPSO, fully leveraging the advantages of Denoising Autoencoder in adding noise during model construction while performing nonlinear mapping and reconstruction to obtain a super-complete basis. Combined with the BPSO algorithm to search for optimal feature combinations, this approach can effectively prevent falling into local optima and enhance model robustness and generalization. Through experimental comparisons on clinical diabetes data and UCI datasets, the improved algorithm is proven to significantly improve classification accuracy and nonlinear structure expression, making it a suitable data analysis method for the Traditional Chinese Medicine domain. However, the improved algorithm also has limitations: the number of hidden layers affects algorithm speed, requiring more iterations to achieve optimal results. In future work, we will continue to improve the algorithm' s search efficiency, while further research can be conducted on how to ensure reasonable parameter settings during model construction.

References

[1] Zhang Boli, Wang Yongyan. Research and development of modern Chinese medicine by multi-components compatibility: theory and practice: fundamental research on key scientific issues of prescriptions [M]. Shenyang: Liaoning Science and Technology Press, 2010.

[2] Li Haojun, Liu Zhongfeng, Wang Wanliang. Binary particle swarm optimization algorithm with arcshaped transfer function [J]. Journal of Chinese Computer Systems, 2017, 38 (12): 2637-2640.

[3] J. Kennedy, R. C. Eberhart. A discrete binary version of the particle swarm algorithm [J]. 1997, 5: 4104-4108 vol. 5.

[4] Bi Kai, Wang Xiaodan, Xing Yaqiong. Cluster ensemble selection based on improved BPSO [J]. Journal of Systems Engineering and Electronics, 2016, 38

(3): 692-698.

[5] Vincent P. A connection between score matching and denoising autoencoders [J]. Neural Computation, 2011, 23 (7): 1661-74.

[6] Zhang Z, Jiang T, Li S, et al. Automated feature learning for nonlinear process monitoring: an approach using stacked denoising autoencoder and k-nearest neighbor rule [J]. Journal of Process Control, 2018, 64 (April 2018).

[7] Dengli B U. MPRM minimization algorithm based on SADPSO [J]. Journal of Chongqing University of Posts & Telecommunications, 2016.

[8] Lin Jun, Xu Lu, Liu Long. Feature selection method based on SVM-RFE and particle swarm optimization [J]. Journal of Chinese Computer Systems, 2015, 36 (8): 1865-1868.

[9] Liu Yanmin. Hybrid particle swarm optimizer for constrained optimization problems [J]. Journal of Tsinghua University: Science and Technology, 2013 (2): 242-246.

[10] Lore K G, Akintayo A, Sarkar S. LLNet: A deep autoencoder approach to natural low-light image enhancement [J]. Pattern Recognition, 2016, 61.

[11] Wang Xianbao, He Wenxiu, Wang Xingang, et al. Capsule defects detection based on stacked denoising autoencoders [J]. Computer Science, 2016, 43 (2): 64-67.

[12] Li Peng, Ning C. Feature learning for music auto-tagging using denoising autoencoder [J]. Journal of East China University of Science & Technology, 2017, 43 (2): 241-247.

[13] Li Yanghui, Xie Ming, Yi Yang. Sentiment analysis of micro-blogging based on DAE and its improved model [J]. Application Research of Computers, 2017, 34 (2): 373-377.

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