Postprint of Research on Fall Detection Algorithm Based on PSO Pattern Search

Authors: Ren Xiaokui, Li Feng, Cheng Lin

Date: 2019-01-28T00:00:00+00:00

Abstract

To address the shortcomings of the particle swarm optimization algorithm, including severe late-stage homogeneity, slow convergence speed, and susceptibility to local minima, the pattern search algorithm is introduced into the particle swarm optimization algorithm to optimize support vector machine parameters for fall detection applications. First, wearable devices are employed to collect fall detection datasets, and the initial data undergo mean filtering to eliminate noise effects. Then, feature extraction is performed on the filtered data, and the extracted multidimensional data is dimensionally reduced using the singular value decomposition algorithm. Finally, the dimensionally reduced data is used to evaluate the performance of the particle swarm pattern search algorithm. Through comparison with the support vector machine algorithm and the support vector machine algorithm combined with particle swarm optimization, the particle swarm pattern search algorithm demonstrates improved specificity and sensitivity in fall detection.

Full Text

Preamble

Title: Research on Fall Detection Algorithm Based on PSO Pattern Search

Authors: Ren Xiaokui, Li Feng, Cheng Lin

Affiliation: School of Electronics & Information Engineering, Liaoning Technical University, Huludao, Liaoning 125105, China

Abstract: To address the shortcomings of particle swarm optimization (PSO), including severe homogeneity in later stages, slow convergence, and susceptibility to local minima, this paper introduces the pattern search algorithm into PSO to optimize support vector machine (SVM) parameters for fall detection applications. First, wearable devices are used to collect fall detection datasets, and the raw data undergo mean filtering to eliminate noise. Next, features are extracted from the filtered data, and the resulting multidimensional data is reduced in dimensionality using singular value decomposition (SVD). Finally, the dimensionality-reduced data is used to evaluate the performance of the PSO pattern search algorithm. Compared with standalone SVM and SVM combined with standard PSO, the proposed PSO pattern search algorithm demonstrates improved specificity and sensitivity in fall detection.

Keywords: fall detection; particle swarm optimization; pattern search; dimension reduction; support vector machine

0 Introduction

According to World Health Organization statistics, an estimated 646,000 fatal fall accidents occur worldwide annually, making falls a leading cause of death from accidental injuries. When a fall occurs, while some elderly individuals who experience minor falls can self-rescue, those with severe injuries who do not receive timely assistance may face serious consequences, including life-threatening situations. Consequently, fall detection has remained a prominent research focus [**?**].

Current fall detection methods can be broadly categorized into three types: (a) camera-based video detection [**?**, **?**], which analyzes human contours and postures in video imagery; (b) dedicated environmental change detection [**?**, **?**], which employs ultrasonic arrays to analyze ground impact patterns from falls; and (c) wearable sensor-based detection [**?**, **?**], which installs specific sensors to extract motion features and performs fall detection through pattern recognition methods. However, video detection raises privacy concerns, dedicated environments are heavily influenced by specific conditions and involve complex, costly processes with limited detection ranges. In contrast, wearable devices offer significant advantages including compact size, portability, and privacy protection, leading to their widespread development. Hu Lisha et al. [**?**] provided a comprehensive review of wearable fall detection technologies, summarizing recent research achievements in this domain.

Support vector machines exhibit excellent performance in classification tasks, and fall detection can be naturally formulated as a binary classification problem. However, SVM performance heavily depends on parameter settings. Particle swarm optimization is a global parallel optimization algorithm that offers advantages such as short evolution time and high optimization precision compared to other methods, making PSO-SVM particularly effective for fall detection. Sun Xiaowen et al. [**?**] employed threshold discrimination followed by PSO-SVM for secondary decision-making. Pei Liran et al. [**?**] utilized RBF-based SVM with PSO for parameter optimization. Ma Wengang et al. [**?**] applied the conjugate gradient method for data optimization and implemented a two-stage decision process using threshold determination and PSO-SVM.

Noise interference from various sources affects classification performance, necessitating data preprocessing. Ma Wengang et al. [?] used the conjugate gradient method to reduce nonlinear errors. Wang Rong et al. [?] applied Kalman filtering to correct posture angles. He Jian et al. [?] combined Kalman filtering for posture correction with KNN algorithms, achieving satisfactory fall detection results.

During feature extraction, more feature vectors provide more detailed behavior descriptions but risk the curse of dimensionality, requiring dimensionality reduction. Bai Yong et al. [?] improved fall detection sensitivity through SVD methods. Li Lei [?] achieved excellent results by improving the KPCA algorithm.

Although PSO is widely applied, traditional PSO suffers from slow convergence and susceptibility to local minima. To address this, Wang Xibin et al. [?] introduced the pattern search algorithm into PSO, yielding promising results. Following the approach in [?], this paper incorporates PSO pattern search for fall detection. The methodology involves: (1) preprocessing raw data using mean filtering; (2) performing dimensionality reduction via SVD [?]; (3) optimizing SVM parameters using PSO pattern search; and (4) conducting fall detection using the SVM algorithm with the dimensionality-reduced dataset.

---

1 SVM Basic Principles

Support vector machine (SVM), proposed by Vapnik and colleagues in the 1990s, has proven highly effective for practical problems. The fundamental concept involves using kernel functions to map input sample space into a high-dimensional feature space where optimal classification is achieved.

Given a dataset:

$$T = \{(x_1, y_1), (x_2, y_2) \dots, (x_n, y_n)\}$$

where $x_i \in \mathbb{R}^n$ represents the i-th feature vector, (x_i, y_i) denotes sample points, and $y_i \in \{-1, 1\}$ indicates class labels. The classification hyperplane is obtained by solving:

$$\min \frac{1}{2}\|w\|^2$$

subject to:

$$y_i[w \cdot x_i + b] \geq 1, \quad i = 1, 2, \dots, l$$

where w is the normal vector, b is the bias term, x_i represents training samples, and y_i denotes sample class labels. Using Lagrange multipliers and satisfying KKT conditions yields the optimal classification function:

$$f(x) = \operatorname{sign}(w^* \cdot x + b^*) = \operatorname{sign}\left(\sum_{i=1}^{l} \alpha_i^* y_i (x_i \cdot x) + b^*\right)$$

where α^* and b^* are hyperplane parameters, and $(x_i \cdot x)$ represents the vector dot product. For linearly inseparable cases, slack variables are introduced into constraints and penalty functions are added to the objective function, ultimately transforming into minimizing:

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i$$

where C is the penalty coefficient controlling error sample punishment. Kernel functions are introduced into SVM, with the most commonly used being the radial basis function:

$$K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}$$

where parameter γ affects the complexity of the SVM classification algorithm.

2.1 PSO Algorithm

Particle swarm optimization (PSO), proposed by Eberhart and Kennedy in 1995, draws inspiration from simulating bird flock foraging behavior. Bird flocks adjust their search paths through individual experience and population communication to locate areas with maximum food density. Each bird' s position/path represents a combination of independent variables, with food density at each point representing the function value. In PSO, particles update their velocities and positions based on individual and swarm experience to find optimal solutions.

In D-dimensional space with N particles, particle i has position x_i, fitness function $f(x_i)$, and velocity v_i. The best position experienced by particle i is p_{best}, while the best position experienced by the population is g_{best}. The position and velocity variation ranges for dimension d are $[x_{\min,d}, x_{\max,d}]$ and $[-v_{\max,d}, v_{\max,d}]$, respectively, with boundary value constraints.

The iteration process follows:

$$v_{id}^{k} = wv_{id}^{k-1} + c_1 r_1 (p_{\text{best},id} - x_{id}^{k-1}) + c_2 r_2 (g_{\text{best},d} - x_{id}^{k-1})$$

$$x_{id}^{k} = x_{id}^{k-1} + v_{id}^{k}$$

where v_{id}^{k} and x_{id}^{k} represent the d-th dimensional components of particle i' s velocity and position vectors at iteration k; c_1 and c_2 are constants regulating learning steps; r_1 and r_2 are random numbers in $[0, 1]$ adding search randomness; and w adjusts the solution space search range.

A threshold ε and maximum iteration count are set. If the threshold or maximum iterations are satisfied, the process terminates, with g_{best} as the optimal solution and the current particle position representing the optimal SVM parameters.

2.2 PSO Pattern Search Algorithm

Pattern search, also known as the Hooke-Jeeves method, alternates between axial searches and pattern searches starting from an initial base point. Axial searches determine new base points and function descent directions along coordinate axes, while pattern searches along lines between two base points accelerate function descent. Pattern search exhibits strong local search capability but suffers from numerous iterations, heavy computational load, and susceptibility to local minima with poor global search ability. PSO offers shorter evolution time and higher precision. Combining pattern search with particle swarm optimization leverages the advantages of both algorithms, resulting in the PSO pattern search algorithm.

3.1 Data Preprocessing

Sensors placed at the waist collect signals from human activities for processing to determine whether a fall has occurred. Since acceleration and angular velocity changes are continuous and real-time, data stream processing is challenging. Acceleration data collected by the accelerometer are denoted as $a_x(t), a_y(t), a_z(t)$, while gyroscope data are recorded as $\omega_x(t), \omega_y(t), \omega_z(t)$. The raw collected data is:

$$\begin{bmatrix} a_x(1) & a_y(1) & a_z(1) \\ a_x(2) & a_y(2) & a_z(2) \\ \vdots & \vdots & \vdots \\ a_x(n) & a_y(n) & a_z(n) \\ \omega_x(1) & \omega_y(1) & \omega_z(1) \\ \omega_x(2) & \omega_y(2) & \omega_z(2) \\ \vdots & \vdots & \vdots \\ \omega_x(n) & \omega_y(n) & \omega_z(n) \end{bmatrix}$$

As sensor data is subject to noise interference, effective denoising methods must be applied before analysis to improve fall detection accuracy. Mean filtering is employed here to eliminate spikes in sensor data while preserving original information and removing noise interference.

3.2 Feature Extraction and Dimensionality Reduction

During feature extraction, more feature vectors provide more detailed behavior descriptions but risk the curse of dimensionality. To ensure accurate behavior representation while maintaining computational feasibility, feature vectors must be appropriately processed. In wearable fall detection, since wearable devices provide signal sources that cannot directly form effective training datasets, extracting reasonable fall feature vector sets [?] is essential. Common time-domain feature statistics are shown in Table 1 .

Based on singular value decomposition (SVD), useful feature information can be effectively separated, and dimensionality reduction via SVD retains most valuable information while reducing dimensions. For a dataset:

$$A_{m\times n} = U_{m\times m}\Sigma_{m\times n}V_{n\times n}^T$$

where $A_{m\times n}$ represents m samples with n features each; U and V are orthogonal matrices; and Σ is a non-negative diagonal matrix. According to SVD principles, the first l principal components contain the maximum information. Therefore, the first l principal components are extracted as new dimensionality-reduced data:

$$A'_{m\times l} = U_{m\times l}\Sigma_{l\times l}$$

where $A'_{m\times l}$ is the new feature space; $\Sigma_{l\times l}$ is the diagonal matrix corresponding to the first l eigenvalues; and $U_{m\times l}$ is the matrix comprising the first l columns. Since SVD is fundamentally similar to PCA, where larger singular values contain more information, SVD-based dimensionality reduction maximally preserves original information while significantly reducing computational complexity. Thus, the new feature set contains only useful feature information.

3.3 PSO Pattern Search Algorithm for SVM Parameter Optimization

The PSO pattern search algorithm proceeds as follows:

a) Read fall detection data samples and randomly generate initial positions (c, γ).

b) Calculate each particle' s fitness value using classification error as the fitness evaluation function:

$$\text{Error} = \frac{F}{T+F}$$

where T and F represent correctly and incorrectly classified sample counts, respectively.

c) Update particle positions and velocities according to equations (5) and (6).

d) Set the result from equation (3) as the initial value, assign unit vectors e_j, initial step size $\gamma_0 > 0$, acceleration coefficient $\lambda \in (0,1)$, contraction coefficient, precision ε, and set $k = 0$, $y = p_{\text{best}}$.

e) Starting from y, perform axial probing moves parallel to unit vectors e_j:

If $f(y + \gamma e_j) < f(y)$, then $y = y + \gamma e_j$; otherwise, if $f(y - \gamma e_j) < f(y)$, then $y = y - \gamma e_j$.

f) If $y \neq p_{\text{best}}$, perform pattern move: let $p_{\text{best}}^{k+1} = y$; otherwise return to step b).

g) If $\|p_{\text{best}}^{k+1} - p_{\text{best}}^{k}\| < \varepsilon$, terminate iteration and output results; otherwise, when $f(p_{\text{best}}^{k+1}) < f(p_{\text{best}}^{k})$, set $y = p_{\text{best}}^{k+1} + (p_{\text{best}}^{k+1} - p_{\text{best}}^{k})$ and return to step e); when $f(p_{\text{best}}^{k+1}) \geq f(p_{\text{best}}^{k})$, set $y = p_{\text{best}}^{k+1}$ and return to step e). If $\gamma < \varepsilon$, set $\gamma = \lambda\gamma$ and $k = k + 1$; otherwise, if maximum iterations are reached, return to step b).

The final optimization result (c, γ) is output and incorporated into the SVM algorithm. The training dataset produces a trained model, and the test dataset serves as input to the classification model for prediction, distinguishing between falls and daily activities.

3.4 Algorithm Flow

The fall detection process is illustrated in Figure 1 [Figure 1: see original paper].

4.1 Experimental Scheme

Experimental data were collected from six subjects (three male, three female) with devices placed at the waist, totaling 1,000 data groups: 400 fall groups (100 each for forward, backward, left, and right falls) and 600 normal activity groups (100 each for forward walking, backward walking, jogging, squatting then standing, bending, and lying flat). The sampling frequency was 25 Hz. Three hundred fall samples and 500 normal activity samples served as the training set, with the remainder used for SVM testing. Taking normal walking and forward falling as examples, acceleration and posture angles exhibit dramatic changes during falls. To eliminate noise effects, mean filtering is applied, which removes spikes and smoothes the data curves.

Filtered data is used to extract multidimensional features according to the table format, and SVD-based dimensionality reduction extracts principal components for subsequent algorithm processing. To validate the effectiveness and correctness of the PSO pattern search algorithm, SVM classification performance is compared between initial PSO and PSO pattern search algorithms.

The experimental schemes are: - **Scheme 1:** Feature set + SVM algorithm - **Scheme 2:** Feature set + Initial PSO algorithm + SVM - **Scheme 3:** Feature set + PSO pattern search algorithm + SVM

4.2 Experimental Results

Accuracy (Ac), specificity (Sp), and sensitivity (Se) are selected as evaluation metrics for algorithm performance, with parameter descriptions provided in Table 2 .

Table 2 : Parameter Specification - **Accuracy:** Probability of correct classification in samples - **Specificity:** Probability of detecting falls in fall samples - **Sensitivity:** Probability of identifying non-falls in non-fall data - **True Positive:** Fall samples correctly detected as falls - **False Negative:** Fall samples detected as non-falls - **True Negative:** Non-fall data correctly identified as non-falls - **False Positive:** Non-fall samples detected as falls

Table 3 : Comparison of Three Schemes

Scheme	Accuracy	Sensitivity	Specificity	Computation Time (s)
1. Feature set + SVM				
2. Feature set + Initial PSO + SVM				
3. Feature set + PSO pattern search + SVM				

Based on equations (12)-(14), experimental data is calculated and presented in Table 3. Scheme 1, using standalone SVM, shows the lowest performance. Schemes 2 and 3 both incorporate PSO for SVM parameter optimization. Compared to Scheme 1, Scheme 2 improves accuracy by 0.07 (4% improvement), while Scheme 3 improves accuracy by 0.09 (11% improvement) over Scheme 1 and by 0.06 (7% improvement) over Scheme 2. Improvements in sensitivity and specificity are also observed.

Regarding computational time, Scheme 1 requires grid search for SVM parameters, resulting in the longest time. Scheme 2 significantly reduces SVM parameter optimization time through PSO. Scheme 3 further incorporates pattern search to avoid local optima, achieving the highest accuracy while consuming the least time.

5 Conclusion

This paper introduces the PSO pattern search algorithm for fall detection classification. Using wearable devices for data collection, mean filtering for denoising, SVD for dimensionality reduction, and pattern search-enhanced PSO for SVM parameter optimization, the proposed method improves fall detection accuracy and reduces algorithm runtime.

References

[1] Xu Tao, Zhou Yun, Zhu Jing. New advances and challenges of fall detection systems: a survey [J]. Applied Sciences, 2018, 8(3).

[2] Yang Renbing, Chen Wenbo, Qian Qing, et al. Fall detection algorithm based on multi-feature extraction in infrared image [J]. Infrared Technology, 2017, 39(12): 1131-1138.

[3] Qu Chang, Sun Jie, Wang Junze, et al. Automatic fall detection for the elderly using kinect sensor [J]. Chinese Journal of Sensors and Actuators, 2016, 29(3): 378-383.

[4] Nadee C, Chamnongthai K. Octagonal formation of ultrasonic array sensors for fall detection [C]//Proc of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference. 2016: 1-4.

[5] Nadee C, Chamnongthai K. Multi sensor system for automatic fall detection [C]//Proc of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference. 2015: 930-933.

[6] Litvak D, Zigel Y, Gannot I. Fall detection of elderly through floor vibrations and sound [C]//Proc of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2008: 4632-4635.

[7] Tu Biqi. Research on fall detection algorithm for the elderly based on muti-sensor fusion [D]. Hangzhou: Zhejiang Sci-Tech University, 2017.

[8] Song Jiahua. The research on the key technologies of fall detection [D]. Jinan: Shandong University, 2017.

[9] Hu Lisha, Wang Suzhen, Chen Yiqiang, et al. Fall detection algorithms based on wearable device: a review [J//OL]. Journal of Zhejiang University: Engineering Science, 2018(9): 1-11.

[10] Sun Xiaowen, Sun Ziwen, Qin Fang. Research on human fall detection based on threshold and PSO-SVM [J]. Computer Engineering, 2016, 42(5): 317-321.

[11] Pei Liran, Jiang Pingping, Yan Guozheng. Research on fall detection system based on support vector machine [J]. Optics and Precision Engineering, 2017, 25(1): 182-187.

[12] Ma Wengang, Wang Xiaopeng, Wu Zuopeng. PSO-SVM feature vector fall detection algorithm based on human postures [J]. Chinese Journal of Sensors and Actuators, 2017, 30(10): 1504-1511.

[13] Wang Rong, Zhang Yun, Chen Jianxin. Design and implementation of fall detection system using tri-axis accelerometer [J]. Journal of Computer Applications, 2012, 32(5): 1450-1452, 1456.

[14] He Jian, Zhou Mingwo, Wang Xiaoyi. Wearable method for fall detection based on kalman filter and k-NN algorithm [J]. Journal of Electronics & Information Technology, 2017, 39(11): 2627-2634.

[15] Li Lei, Zhang Fan, Shi HuaJi, et al. Research on feature vector extraction and dimension reduction of wearable fall detection [J/OL]. Application Research of Computers, 2019(1): 1-3.

[16] Bai Yong, Sun Xiaowen, Qin Fang, et al. The falling detection algorithm based on svd feature dimension reduction and SVM [J]. Computer Applications and Software, 2017, 34(1): 247-251.

[17] Wang Xibin, Zhang Xiaoping, Wang Hanhu. Parameter optimization of support vector machine and application based on particle swarm optimization mode search [J]. Journal of Computer Applications, 2011, 31(12): 3302-3304, 3326.

[18] Pannurat N, Thiemjarus S, Nantajeewarawat E. Automatic fall monitoring: a review [J]. 2014, 14(7): 12900-12936.

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

Source: ChinaXiv —Machine translation. Verify with original.