Evolutionary Optimization of User Intent Recognition for Transfemoral Amputees

Gholamreza Khademi, Hanieh Mohammadi, Dan Simon
Department of Electrical and Computer Engineering
Cleveland State University
Cleveland, Ohio, USA
Email: g.khademi17@vikes.csuohio.edu

Elizabeth C. Hardin
Motion Study Laboratory
Cleveland VA Medical Center
Cleveland, Ohio, USA
Email: ehardin@fescenter.org

Abstract—Lower-limb prosthetic legs help amputees regain their walking ability. User intent recognition is utilized to infer a human gait mode (fast walk, slow walk, etc.) so the controller can be adjusted depending on the detected gait mode. In this paper, mechanical sensor data is collected from an able-bodied subject and used for user intent recognition. Feature extraction, principal component analysis, correlation analysis, and K-nearest neighbor methods are used, modified, and optimized with an evolutionary algorithm for improved performance. The system successfully classifies four different walking modes with an accuracy of 96%.

Keywords—user intent recognition; lower-limb prosthesis; evolutionary algorithm; K nearest neighbor

I. INTRODUCTION

The design and development of lower-limb prostheses has received considerable attention due to the rapidly growing number of people with limb loss. There are nearly 120,000 lower-limb amputations in the United States every year, mainly due to vascular disease [1]. Powered lower-limb prostheses have great advantages over passive prostheses; amputees who use passive prostheses expend up to 60% more metabolic energy than healthy individuals [2]. Powered prostheses are being developed to alleviate this difficulty [3]. Moreover, powered prostheses are better able to help amputees handle multiple activity modes: standing, level walking, walking up and down a ramp, stepping up and down, etc. For each walking mode, a different control strategy or control gains are used to control the prosthesis [4]. Therefore, it is important to infer the user’s intent automatically while transitioning from one walking mode to another one, and to subsequently activate the suitable controller or control gains. Pattern recognition techniques [5] are used to address such problems in real time.

In previous research, electromyography (EMG) signals were collected to determine the user’s intent [6]: for example, level walking, and walking up and down a ramp with various slopes. EMG-based pattern recognition has been used for both healthy subjects and transfemoral amputees [7]. The fusion of mechanical sensor data with EMG signals can improve user intent classification accuracy [8]. Although EMG-based recognition can yield high accuracy in user intent detection, drawbacks include changes in EMG signals due to electrode shift [9] or muscle volume [10], and the invasive nature of EMG data collection. Therefore, EMG-based user intent recognition needs to be periodically retrained. To overcome this problem, a user independent recognition system was recently proposed [11]. Mechanical sensor data for a transfemoral amputee [12] and an able-bodied subject [13] showed satisfactory user intent recognition accuracy.

In our study, mechanical sensor data are experimentally collected for user intent recognition. We intentionally avoid the use of EMG data because of its invasiveness. The general architecture of the user intent recognition system includes feature extraction, principal component analysis, correlation analysis, and a modified K-nearest neighbor classifier trained with an evolutionary algorithm. This research builds upon and improves previous similar work [13]. Simulation results verify the improved performance of the proposed method.

Section II introduces the architecture of the user intent recognition system. Section III explains the experimental setup, the gait data collection, and the formulation of the training and test data. Section IV presents results and discussion. Section V presents conclusions and future work.

II. USER INTENT RECOGNITION SYSTEM ARCHITECTURE

User intent recognition consists of several steps. There are many tunable parameters in the system; therefore, an evolutionary algorithm is used to optimize system performance. The architecture of the user intent recognition system is illustrated in Fig. 1. The role of each subsystem is explained in more detail in the remainder of this section.

A. Sensor Data

The appropriate selection of input data has a strong influence on the accuracy of user intent detection. Input signals must be chosen to accurately reflect various gait modes. Signals reflecting the state of the prosthesis, user-prosthesis interactions, and prosthetic-environment interactions are commonly used [12].

Fig. 1: Architecture of user intent recognition system. An evolutionary algorithm (not shown) is used to optimize the system components.
B. Data Preparation

In this step, sensor signals are processed and filtered to eliminate noise and to handle missing data points [14]. Input signals are normalized to equalize the relative magnitude of each sensor measurement.

C. Feature Extraction

Different techniques are available to extract suitable features from a signal to be used in the classification step [12]. Feature extraction needs to be computationally fast for real-time implementation. In this paper, a vector of selected mechanical sensors is used as the relevant feature at each time step.

D. Principal Component Analysis

To eliminate the least relevant features and keep the most important features, the dimension of the training data needs to be reduced [15]. Principal component analysis (PCA) is a well-known linear transformation for converting data to a lower dimension [16].

E. Correlation Analysis

The training data set is a matrix, each column corresponding to a given feature (i.e., selected sensor measurement), and each row corresponding to a different sample time and a different gait mode. PCA is used to reduce the number of features (columns), while correlation analysis removes highly correlated samples (rows) from the training set. In the correlation analysis step, a correlation matrix is found and samples (rows) with a correlation greater than a user-defined (or optimized) threshold are removed. Correlation analysis is applied separately to each gait mode. The correlation of two sample points \( X_i \) and \( X_j \) (row vectors of length \( N \)) is their normalized sample covariance, which is defined as

\[
\rho_{ij} = \frac{(X_i - E(X_i))(X_j - E(X_j))^T}{(N - 1)\sigma_i\sigma_j}
\]

where subtraction is element-by-element, \( E(X_i) \) is the average of the \( i \)-th sample point taken over all measurements, and \( \sigma_i \) is the sample standard deviation of \( X_i \). Although correlation analysis facilitates real-time classification because of the subsequent reduction in training data size, it may increase classification error. A tradeoff between training time and classification error must be chosen by a suitable choice for the correlation threshold.

F. K-Nearest Neighbor (K-NN) Classification

Various classification methods have been used for user intent recognition [17]. K-NN can operate with minimal computational effort and training information, which makes it appealing for real-time implementation. K-NN is a simple machine learning algorithm, but it is possible to modify it for enhanced performance. Here a modified version of K-NN is proposed.

K-NN includes two steps: (1) Given a test point, find its \( K \) nearest neighbors in the training set based on some similarity metric; (2) Identify the class of the test point by determining the maximum number of classes among the \( K \) nearest neighbors. The typical similarity metric is the normalized Euclidean distance. This simple version of K-NN has been modified here for user intent recognition. Experimental results shown later in this paper verify that the modified K-NN algorithm classifies data significantly better than the standard K-NN algorithm. The changes to the standard K-NN algorithm are described as follows.

1) Distance-Weighted Nearest Neighbors: In the basic K-NN algorithm, each of the \( K \) nearest neighbors have an equal level of importance. To generalize K-NN, the contribution of each neighbor can be weighted on the basis of its distance to the test point. An inverse distance scheme is suggested:

\[
\alpha_i = \frac{1}{\sum_{i=1}^{K} \frac{1}{c + h_i^p}} \quad i = 1, ..., K
\]

where \( \alpha_i \) is the weight contribution of the \( i \)-th nearest neighbor, \( h_i \) is the Euclidean distance of the \( i \)-th nearest neighbor to the test point, \( w \) is a user-selectable (or optimized) tuning parameter, and \( c \) is a small constant to prevent division by zero.

2) Classification History: Simple K-NN classifies a test point without considering the history of previously classified test points. In user intent recognition, our knowledge of human gait behavior can improve K-NN. We know that human transitions from one gait mode to another are considerably slower than real time classification. That is, there is a high probability that the current walking mode is the same as the walking mode a few time steps previously. This motivates the incorporation of an additional step to K-NN. After K-NN classification, the final identified gait mode is based on the currently classified mode and the \( n - 1 \) most recently classified modes, using majority voting. \( n \) is the user-selected (or optimized) value of the length of the vector of recently-classified modes. The currently classified mode, and the most recently classified modes, have a proportionately greater influence on the final identified mode, so the previously-classified modes are weighted as follows:

\[
\beta_i = \frac{1}{\sum_{i=1}^{n} \frac{1}{(1 - t_i)w}} \quad t_i = 0, -1, -2, ..., n + 1
\]

where \( \beta_i \) is the relative weight of the \( i \)-th most recently-classified mode, and \( \nu \) is a user-selectable (or optimized) tuning parameter. Although this enhancement to K-NN alleviates chattering between identified modes, inappropriate value for \( n \) may cause an undesirable time delay in user intent detection.

K-NN parameters, \( K, n, w \) and \( \nu \), need to be found to minimize classification error. Classification error is defined as the number of incorrectly classified test data points divided by the total number of classified test data points.

G. Biogeography-Based Optimization

There are multiple degrees of freedom in the user intent recognition system, and optimization techniques can be used to tune the parameters and obtain a classification system with the highest possible accuracy. In this research we use Biogeography-Based Optimization (BBO) as the evolutionary optimization algorithm. BBO belongs to the family of gradient-free, population-based optimization algorithms. BBO is based on mathematical models that describe the emigration, immigration, and distribution of species among islands [18].
For the purposes of user intent optimization, we could select any evolutionary algorithm; the particular algorithm is not important. We have chosen BBO because of its demonstrated effectiveness and its rapid increase in popularity for optimizing real-world problems, including biomedical applications [18].

III. EXPERIMENTAL SETUP AND DATA COLLECTION

To test the performance of the proposed user intent recognition method, multiple sets of experimental data were collected for various gait modes: standing (ST), slow walking (SW), normal walking (NW), and fast walking (FW). Data collection was performed at the Motion Studies Laboratory of the Cleveland Department of Veterans Affairs Medical Center (VAMC). Forty-seven reflective markers were placed on volunteer subjects to create a 3D model for computing joint angles and torques. A 16-camera Vicon system was used to collect kinematic data at 100 Hz, while ground reaction force (GRF) data were collected from force platforms in the treadmill at 1000 Hz. Data were filtered by a 6 Hz second-order low pass filter and input to inverse dynamics software for post-processing.

Three 15-second trials were collected for the transition from ST to SW and vice versa, five trials were collected for the transition from ST to NW and vice versa, eight trials were collected for the transition from FW to ST and vice versa, and seven trials were collected for the transition from ST to FW and vice versa. The SW speed was 0.9 m/s, the NW speed was 1.2 m/s, and the FW speed was 1.5 m/s. Figure 2 illustrates the experimental setup for both able-bodied and amputee subjects, although only able-bodied data were used in this paper. Follow-on work will extend the proposed method to amputee gait data.

Unilateral hip and ankle angles were selected as relevant user intent features. GRF and socket force typically represent the prosthesis-environment interaction and the user-prosthesis interaction, respectively. In this paper hip moment is used instead of socket force as a less invasive measurement because these two quantities are directly related to each other.

In summary, hip and ankle angles, GRF along three axes, and hip moment, comprise the six input signals which were used for user intent recognition. Fig. 3 shows an example of test data for a walking trial lasting approximately 18 seconds, which included different walking modes. The subject was asked to transition from ST to SW to NW to FW. The subject was then asked to transition through the walking modes in the reverse order.

IV. RESULTS AND DISCUSSION

This section illustrates the use of training and test data to assess the performance of an optimized user intent recognition system. The code used to generate these results is available at http://embeddelab.csuohio.edu/Prosthetics/UserIntent/.

Training data is collected in an 8530 × 6 matrix where each row represents a different data point (either ST, SW, NW or FW), and each column represents one of the measurements. All measurements were normalized. PCA was applied to reduce the measurement dimension, so the training matrix was reduced to an 8530 × 3 matrix. Then BBO was implemented to find optimum values for \( K, n, w, \text{and} \, v \) to minimize classification error. The user intent recognition system was run on a PC with a 2.60 GHz processor and 24 GB of memory. The search space was constrained to

\[
1 \leq K, n \leq 100, \quad 0 \leq w, \, v \leq 5 \quad (4)
\]

Parallel computing reduced the optimization time from 7.77 days to about 20 hours. Since BBO is a stochastic algorithm, 20 Monte Carlo runs were used to optimize the tuning parameters. The best solution was \( K = 12, \, n = 31, \, w = 0.9995, \, v = 0.3924 \), with a classification error of 3.59%.

Table 1 shows the performance of K-NN for different parameter values. It should be noted that basic K-NN is a special case of the proposed K-NN. The first row of Table 1 shows \( K = 7, n = 1, w = 0, v = 0 \), which means all 7 nearest neighbors have equal weights, and previously-classified gait modes are not used to inform the current classification mode. The second row of Table 1 shows that nearest neighbors are weighted according to (2). It is clear that weighted K-NN has a lower error than the simple K-NN. The third row of Table 1 shows that previously-classified modes are used in addition to weighted K-NN, which further decreases classification error. The fourth row of Table 1 shows that the optimized K-NN provides the minimum classification error.

Table 1 shows that training error is lower than test error. Table 1 also shows that NW classification is worse than other modes. This is because the number of trials collected for NW was initially fewer than the number for the other classes, so the system had less opportunity to be optimized for NW. The effect in Table 1 of using previously-classified modes is interesting. The second row shows that when previously-classified modes are not used, ST error is 0 while overall error is 11.5%.

![Fig. 2: Experimental setup. The figure on the left shows able-bodied data collection, while the figure on the right shows amputee data collection using a Genium prosthesis.](image)
The third row of Table 1 shows that the use of previously-classified modes decreased chattering and overall classification error, but caused a transition delay that resulted in a 1.58% error in ST mode. It can be concluded that a high value for \( n \) results in high transition error which may dominate the overall classification error. The method presented here is a general version of previous research [13]. For fair comparison, the tuning parameters of [13] were optimized by BBO, which resulted in an error of 4.67%, which is about one percent higher than the method proposed here. Fig. 4 shows the performance of the classifier using both simple K-NN and optimized K-NN.

To implement classification in real-time, the classifier must process and assign labels to test data faster than the sampling frequency, which is normally 100 Hz. In this paper, PCA and correlation analysis were investigated to make the user intent algorithm feasible for real-time implementation. PCA did not greatly influence classification time since the number of measurements is not large. But correlation analysis is used to decrease the number of training cases and thus decrease computational effort. Table 2 shows that as the threshold discussed in Section II(E) decreases, the number of test cases and the classification time decreases, while the classification error increases. The number of test cases is very sensitive to the correlation threshold, and all of the thresholds used in Table 2 are very close to 1.

Table 2: Correlation analysis with different threshold values

<table>
<thead>
<tr>
<th>Number of Test Cases</th>
<th>8530</th>
<th>3779</th>
<th>3170</th>
<th>2008</th>
<th>719</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification Error (%)</td>
<td>3.39</td>
<td>7.20</td>
<td>5.52</td>
<td>12.51</td>
<td>25.56</td>
</tr>
<tr>
<td>Classification Time (see)</td>
<td>0.0298</td>
<td>0.0142</td>
<td>0.0122</td>
<td>0.0083</td>
<td>0.0044</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this study, K-NN was modified to enhance the performance of a user intent recognition system. An evolutionary algorithm was applied to obtain optimal values for the classifier parameters. Experimental data was used for training and testing the system. It was shown that the system can classify four different walking modes with an accuracy of 96%. Correlation analysis was used to address real-time requirements. For future work, more advanced feature extraction is suggested; for instance, Kalman filtering [19]. Also, multi-objective optimization can find a tradeoff between classification time and accuracy.

REFERENCES


![Fig. 4: Classifier for simple K-NN (top) and optimized K-NN (bottom)](image-url)