ISBN 9789090311722

Feature Selection for Machine Learning Based Step Length Estimation Algorithms

Stef Vandermeeren, Herwig Bruneel, Heidi Steendam
Department of Telecommunications and Information Processing, Ghent University, Belgium,
e-mail: {firstname}.{lastname}@ugent.be

***Abstract*—An accurate step length estimation can provide valuable information to different applications. It can, for instance, be used to detect various gait impairments caused by e.g. Parkinson's disease or multiple sclerosis. Another application is a people dead reckoning (PDR) system, which is used to track users with the help of an inertial measurement unit (IMU) only. In a PDR system this IMU is used to 1) extract steps out of the movement of the user, 2) estimate the length of a step, and 3) estimate the direction of a step. In this work, we will only consider the estimation of the step length using machine learning techniques. We use feature selection to determine the features from a large collection of features that result in the best performance and compare two different metrics to select these features. This resulted in a step length estimator with a mean absolute error of $3.47\,cm$ for a known test person and $4.25\,cm$ for an unknown test person.**

I. Introduction

In the literature, several approaches can be found to estimate the step length from IMU data. In parametric based approaches [1], the step length is written as a function of different variables. The drawback of this approach is that it relies on parameters that depend on the specific user. Another approach uses a Kalman filter [2] to estimate the step length from a double integration of the acceleration in a fixed reference frame. However, the sensor measurements contain a bias, which will result in erroneous step length estimations. A last approach is to use supervised learning algorithms [3], [4]. In this approach, the algorithm is first trained with examples of steps with known length. In this way, the algorithm can build a model for the step length from the training set, which will be used to predict the length for new data in the inference phase. To build the model, the algorithm is fed with *features*, which are scalar numbers extracted from the measured acceleration signal, e.g. the maximum, the mean and the variance. So far, previous machine learning methods used a predetermined feature set, derived from the accelerometer, to train the supervised learning algorithms. The complexity of both the training and the inference phase of the supervised learning algorithms rises with the number of used features. Therefore, a low-complexity algorithm preferably employs as few as possible features. However, thoughtlessly reducing the number of features can strongly affect the performance of the algorithms. Hence, selecting which features to be used is a crucial step. In the literature, the features for machine learning based step length estimation are selected in an ad hoc way. In this contribution, we evaluate the influence of different features and propose a systematic approach to select the features.

II. Methods

A. Feature extraction from the accelerometer signal

Common to all supervised learning algorithms is that they require as input a set of features. In this work, we restrict our attention to data captured by the accelerometer contained in a handheld smartphone. The supervised learning algorithm predicts the output, i.e. the length of a step, based on the features calculated for that step. To determine which features are most suitable for step length estimation, we calculate for each step in our experimental data a large number of features (i.e. 128), including but not limited to the minimum, maximum, mean, variance, and energy of all acceleration components and the acceleration magnitude.

B. Ranking the features

In a supervised-learning-based step length estimator, the algorithm determines the distance a user moved based on the features. It is clear that the choice of used features will have an impact on the accuracy. For example, a feature of which the value is independent of the step length is useless in step length estimation. Therefore, a good step estimator only uses features that are able to discriminate between large and small steps. To decide which of the 128 features calculated in Section II-A are suitable for step length estimation, we consider two metrics that express how much the step length influences a feature. In the first metric, we calculate the correlation between a feature and the step length, and this correlation is then converted into a p-value. The higher the p-value, the stronger the feature is correlated to the step length, thus the better it is suitable to estimate the step length. The second metric determines the mutual information between the value of a feature and the corresponding step length. As the mutual information is a measure of the dependency between two variables, a large mutual information implies the feature can decrease the uncertainty about the step length significantly, while low mutual information entails the feature is useless for step length estimation. Based on these two metrics, we create two ranked feature sets where the features from Section II-A are sorted from highest to lowest p-value and mutual information, respectively.

C. Feature Selection

In the previous section, we generated two ranked feature sets. In this section, we propose an algorithm that selects a subset of features resulting in optimal performance with a limited number of features, based on the ranking of the features in one of the two generated ranked sets. The algorithm follows a similar approach as in [5], but in contrast to [5], where a classification problem was considered, we handle in this paper a regression problem. The feature selection algorithm starts with an empty feature subset and sequentially updates the subset to obtain the final feature set. In previous research, often the mean error on the step length is used to evaluate the performance of a step length estimator. A problem with the mean error, however, is that an estimator that half of the time largely overestimates the step length and the other half underestimates the step length, results in a mean step length error close to 0. Hence, to compare the performance of the different subsets, we use the mean absolute error (mae) of the step length. The closer the mae gets to zero, the higher the accuracy of our step length estimator.

The feature selection algorithm consists of three phases. In the initialization phase, the algorithm selects the highest ranked feature and computes the performance. In the subsequent addition phase, the next features in the ranking are added, one by one, and after each addition, the performance is determined. If the feature improves the performance, it is selected for the final feature set, otherwise, we exclude the feature. Finally, in the deletion phase, the features selected for the final feature set are evaluated once more. If withdrawing a feature from the set will not significantly reduce the performance, it safely can be removed from the final feature set. If in the deletion phase none of the features was removed, the algorithm stops. In the other case, we repeat the addition phase followed by a deletion phase. This approach yields optimal performance in the sense that adding or removing a feature will not improve the accuracy.

III. Results

In this section, we compare the performance of machine-learning-based step length estimators employing the proposed systematic feature selection algorithm. We apply the selection algorithm to five supervised-learning algorithms with training data of one test person. For each of the considered machine learning algorithms, we rank the features using the two approaches from Section II-B and determine the final feature sets using the selection algorithm from Section II-C. Next, we determine if the optimal feature set derived from one test person also performs well on new data from the same person (test person 1) used to train the algorithm, and on data from a new test person (test person 2). In Table I, we show the results from these tests for approach 1 and 2, where respectively the p-value and mutual information are used to rank the features. From this table, it follows that the RBF SVM and ridge regression algorithms have the lowest mean absolute error and that on average, approach 1 results in a lower mean absolute error than approach 2. All our step length estimators, except the one based on the decision tree, perform better than the methods from [3], [4]. Based on the results from Table I, we can conclude that our feature selection approach combined with the RBF SVM or ridge regression algorithm results in the best performance when compared with current state-of-the-art neural network approaches. Even when our algorithm is tested on data from a new test person our algorithm outperforms current state-of-the-art neural network approaches.

Table I
Accuracy comparison of different machine learning algorithms and approaches (approach 1=ranking with p-value; approach 2=ranking with mutual information) with two neural network step length algorithms, validated on new data of a test person that was used for training (person 1) and new data of a new test person (person 2)

mae (cm)	**validation set person 1**		**validation set person 2**	
	Approach 1	Approach 2	Approach 1	Approach 2
k-Nearest Neighbours	3.74	3.81	4.93	5.28
RBF SVM regression	**3.47**	**3.41**	4.25	4.78
Decision Tree	5.01	5.11	5.28	6.61
Elastic Net	3.87	3.62	4.3	**3.84**
Ridge regression	3.62	3.48	**4.14**	4.25
[4]	4.94		6.27	
[3]	4.26		8.22	

IV. Conclusion

In this work, we used the p-score and mutual information to identify and rank the features that could potentially be used for step length estimation and proposed a method to systematically build a feature set for a machine-learning-based step length estimator. We trained multiple machine learning algorithms with this feature set and compared the different algorithms in terms of mean absolute error. To verify the robustness of our algorithm, we also validated our machine learning approach on data from a different test person than the one used for training our algorithm. The RBF SVM algorithm combined with the first feature ranking approach resulted in a mean absolute error of $3.47\,cm$ when tested on the test person that was used for training the algorithm, and a mean absolute error of $4.25\,cm$ when tested on a new test person. We also compared our method with two neural network based step length estimators and both algorithms achieved a lower performance than the presented algorithm. Hence, we showed that our step length estimator can make an accurate step length estimate even for new test persons without changing anything to our algorithm and that we achieve a similar or better accuracy than current state-of-the-art methods.

References

[1] Qinglin Tian, Zoran Salcic, I Kevin, Kai Wang, and Yun Pan. A multi-mode dead reckoning system for pedestrian tracking using smartphones. *IEEE Sensors Journal*, 16(7):2079–2093, 2016.

[2] Alberto Ferrari, Pieter Ginis, Michael Hardegger, Filippo Casamassima, Laura Rocchi, and Lorenzo Chiari. A mobile Kalman-filter based solution for the real-time estimation of spatio-temporal gait parameters. *IEEE transactions on neural systems and rehabilitation engineering*, 24(7):764–773, 2016.

[3] Haifeng Xing, Jinglong Li, Bo Hou, Yongjian Zhang, and Meifeng Guo. Pedestrian stride length estimation from IMU measurements and ANN based algorithm. *Journal of Sensors*, 2017, 2017.

[4] Stephane Beauregard and Harald Haas. Pedestrian dead reckoning: A basis for personal positioning. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, pages 27–35, 2006.

[5] Stef Vandermeeren, Samuel Van de Velde, Herwig Bruneel, and Heidi Steendam. A feature ranking and selection algorithm for machine learning-based step counters. *IEEE Sensors Journal*, 18(8):3255–3265.