

Optimal model selection for posture recognition in home-based healthcare

Shumei Zhang · Paul McCullagh · Chris Nugent ·
Huiru Zheng · Matthias Baumgarten

Received: 28 April 2010 / Accepted: 19 October 2010
© Springer-Verlag 2010

Abstract This paper investigates optimal model selection for posture recognition. Accuracy and computational time are related to the trained model in a supervised classification. An optimal model selection is important for a reliable activity monitoring system. Conventional guidance on model training uses large instances of randomly selected data in order to characterize the classes. A new approach to the training of a multiclass support vector machine (SVM) model suited to limited training sets such as used in posture recognition is provided. This approach picks a small training set from misclassified data to improve an initial model in an iterative and incremental fashion. In addition, a two step grid-search algorithm is used for the parameters setting. The best parameters were chosen according to the testing accuracy rather than conventional validating accuracy. This new approach for model selection was evaluated against conventional approaches in an activity classification study. Nine everyday postures were classified from a belt-worn smart phone's accelerometer data. The classification derived from the small training set and the conventional randomly selected training set differed in two aspects: classification performance to new data (85.1% Pick-out small training set vs. 70.3% conventional large training set) and computational efficiency (improved 28%).

Keywords Optimal model · Posture recognition · Accelerometer · Multi-class SVM

1 Introduction

The increase in the aging population demographics and related prevalence of chronic diseases are already an economic burden on health care systems [26]. The provision of health, social care and positive lifestyle choices within this group is important for independent living. Keeping healthy postures is useful in daily life for people to maintain fitness and ameliorate the symptoms of chronic diseases such as arthritis. Maintenance of good posture for 'standing', 'lying' and 'sitting' positions where the bones and joints are in the correct alignment can reduce muscle strain. Good posture contributes to enhanced relaxation, reduction of stress, good appearance, and ultimately assists in maintaining health and wellness. With poor posture, the bones and joints are improperly aligned and fatigue and pain can result. For example, sitting upright is better than sitting with an arched back. Robust posture recognition allows more intelligent monitoring between healthy and unhealthy postures. A portable posture measurement system can be utilized in various health-related applications such as fitness training and rehabilitation at home [24]. A posture monitoring system combined with a real-time posture-aware reminder system can empower the user, and assist with self management. Coaching can reduce the risk of hospitalization and supply significant savings in health care costs, especially for the elderly [15].

In this paper a posture recognition prototype is presented. Related work in posture classification is discussed in Sect. 2. Posture data acquisition is described in Sect. 3. Section 4 is focused on obtaining the optimal multiclass

S. Zhang · P. McCullagh (✉) · C. Nugent · H. Zheng ·
M. Baumgarten
Computer Science Research Institute,
School of Computing and Mathematics,
University of Ulster, BT37 0QB Northern Ireland, UK
e-mail: P.J.McCullagh@ulster.ac.uk

S. Zhang
e-mail: zhang-s2@email.ulster.ac.uk

SVM model. In particular, how to learn from a small optimal training set is addressed. In Sect. 5 posture classification experiments using a mobile phone are described and results are presented. The discussion of the results is detailed in Sect. 6. Finally in Sect. 7, a conclusion with opportunity for future work is provided.

2 Related work

Studies have focused on human posture classification for home care or human behavior analysis [10, 17]. Also various postures activity-sensing methods have been tried. Wearable sensing of posture facilitates comparison with surveillance cameras, since the sensors are cheap and lightweight and do not require complex room setups. For example, studies have used multiple accelerometers fixed to specific places on the body, such as wrists, arms, thighs, sternum, waist and lower legs [4, 7, 13]. Using many sensors is likely to get a high accuracy in terms of posture and motion classification. Nevertheless, it is also likely to be inconvenient and too burdensome for long-term real life monitoring. A few studies have investigated using a single accelerometer device attached at the waist, sternum or back [21, 23, 33].

Different postures have been recognized using various classification methods. For example, [25] discriminated four postures (standing, crawling, laying and sitting) using histogram projections based on an indoor surveillance camera. Mattmann et al. [24] classified 27 postures (15 sitting and 12 standing postures) by using 21 strain sensors attached to the back region of tight-fitting clothing. Allen et al. [3] distinguished three postures (sitting, standing and lying) and four posture transitions (sit-to-stand, stand-to-sit, lie-to-stand, and stand-to-lie) from data obtained using a single, waist-mounted, triaxial accelerometer. They investigated two classification methods: a rule-based Heuristic algorithm and a Gaussian mixture model algorithm. Sung et al. [29] introduced a wearable kinesthetic system for upper limb posture and gesture measuring by using a sensing garment. Boulay et al. [5] recognized four general posture categories (standing, sitting, bending and lying) and eight detailed posture sub-categories based on the video sequence. They further classified standing postures into standing with one arm up, standing with arms along the body and T-shape posture. Sitting postures include sitting on chair and sitting on the floor. Lying postures may be distinguished as lying with spread legs and lying with curled-up legs.

Studies also have tried a variety of classification algorithms for daily activity and posture recognition. Ravi et al. [27] compared 18 base-level classifiers and meta-level classifiers by four different data settings in the Weka

toolkit [31]. They found that Plurality Voting provided accuracy ranging from 90 to 99%. However the approach was not applicable when training and testing data came from different subjects. The ‘Boosting’ support vector machine (SVM) outperformed other classifiers when the testing and training data came from different subjects, but the accuracy reduced to 73%. They also found that meta-level classifiers perform better than base level classifiers. Sung et al. [29] developed a real-time classifier from Gaussian Mixture Models using frequency measures derived from the continuous accelerometer sensing data. An interesting study by Martiskainen et al. [22] used the multiclass SVM method to recognize cow behavior patterns based on a neck worn three-dimensional accelerometer. They measured behavior activities that included standing, lying, ruminating, feeding, normal and lame walking, lying down, and standing up. Their SVM classifier model achieved a reasonable recognition for the standing, lying, ruminating, feeding, walking normally, and lame walking. But it was poor for lying down and standing up. Their overall precision of the SVM multi-class model was 78%.

The smart phone provides an opportunity to unobtrusively monitor ambulating daily activities. This research uses a smart phone embedded with an accelerometer for data sensing, and a multiclass SVM classifier to recognize nine human postures in daily life. It aims to train an optimal model which will improve the classification accuracy and reduce the computational consumption for activity classification, which is important for eventual real-time implementation.

3 Data acquisition

3.1 The devices

This study uses a HTC touch phone (HD T8282) to record the movement data of daily activities. This smart phone includes an embedded three-dimensional G-sensor which measures the acceleration for classification of a number of activities, both inside and outside of the home environment. This means that users are not required to wear any further technology and can use their handset as the platform from which their activities can be recorded and processed. It is easy to use without assistance and has been found to be comfortable and convenient. The phone is belt-worn at the left waist in a horizontal orientation as shown in Fig. 1a. Axes are defined as follows in this case: X axis is vertical, Y axis is horizontal and Z axis out from the display of the phone. The phone has two orientations depending on how it is placed in the holster as shown in Fig. 1b, c. The monitoring was designed to adapt to the two

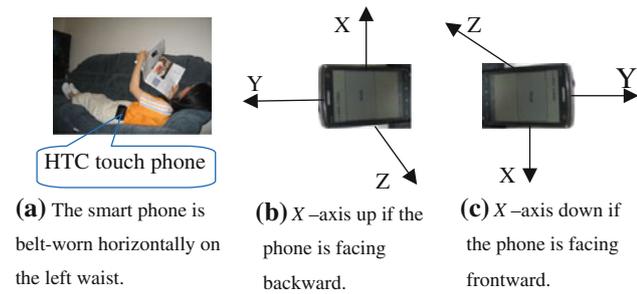


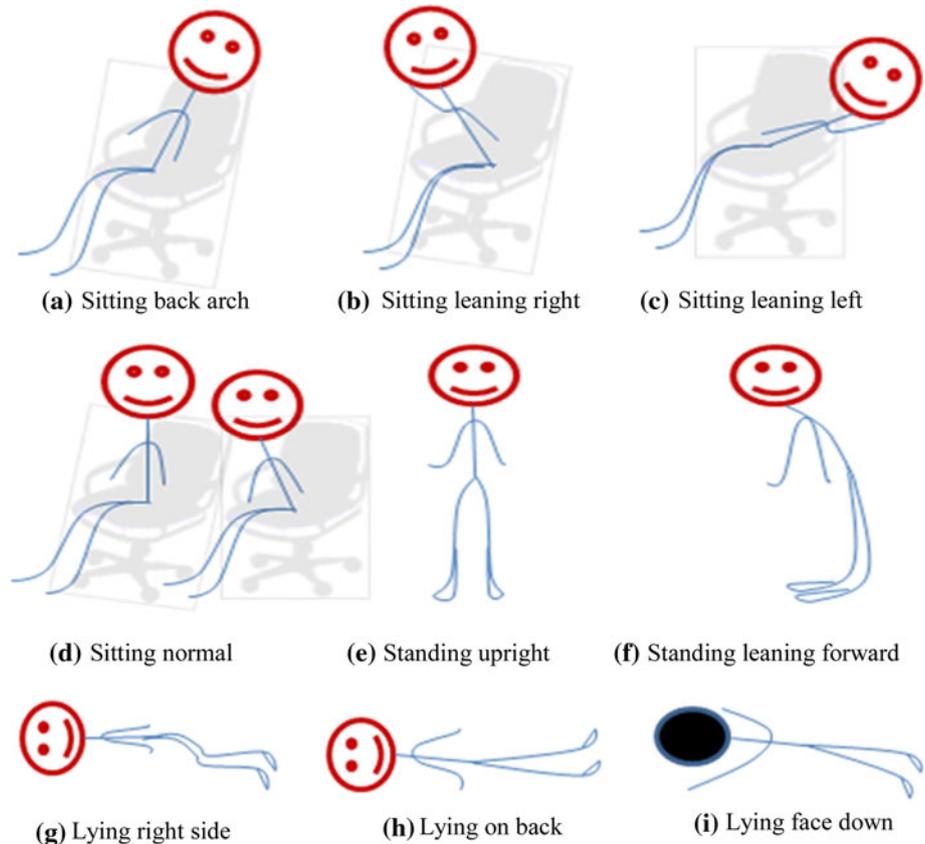
Fig. 1 HTC phone monitoring system configurations

orientations automatically. Based on these configurations, acceleration values in three dimensions with an associated time stamp (t, a_x, a_y, a_z) were collected and subsequently used for the basis of the evaluation of the activity classification algorithms. In addition, a Logitech video camera was fixed in the ceiling of the laboratory for activity recording synchronized with the HTC phone. The video is used to validate the posture classification results.

3.2 Posture selection

Sitting, standing and lying are prevalent activities in human daily life. Subjects may have variability in posture for the same activity. Nevertheless, fatigue or pain could be

Fig. 2 Inactivity in different postures



caused if the subject maintains an unhealthy posture such as “sitting leaning side” for a long time. In order to remind users to keep health postures in their daily life, nine postures (four sitting, two standing and three lying) were selected in this study, as Fig. 2 shown. The nine postures include sitting back arch (Sit-B), sitting leaning right (Sit-R), sitting leaning left (Sit-L), sitting normal (Sit-N, sitting upright or sitting leaning forward), standing upright (Sta-U), standing leaning forward (Sta-F), lying right side (Lyi-R), lying on back (Lyi-B) and lying face down (Lyi-Fd). The nine postures can be used to distinguish healthy and unhealthy behavior in the further reminder systems. For example, Sit-N, Sta-U, Lyi-R and Lyi-B can be defined as healthy postures, and Sit-B, Sit-R, Sit-L, Sta-F and Lyi-Fd are classified as unhealthy postures [18]). In a posture-aware reminder system, an alert reminder will be triggered when the subject keeps an unhealthy posture for a pre-defined period.

3.3 Phone’s position selection

Usually, people put the mobile phone in their pocket, or use it belt-worn on the waist. A position which is both convenient for users and provides high classification accuracy is needed. Two positions (waist and pocket) were initially tested. Figure 3a shows that if the phone is worn in the

waist, most postures have different acceleration values in three dimensions. Figure 3b shows that some postures cannot be recognized when the phone is in the pocket. The two sitting postures (Sit-N with Sit-R), and the two standing postures (Sta-U with Sta-F) have very similar values for the acceleration in three dimensions. This indicates that positioning the phone at the waist is better than in the pocket for distinguishing the above nine posture. So the phone was positioned belt-worn in left waist for the activity monitoring.

Note that the posture label was marked in the vertical acceleration A_x signal only in Fig. 3a, b. Nevertheless, the classification combined three axis's acceleration A_x , A_y and A_z to recognize the nine postures.

4 Method

In this section we investigate the possible classifiers, describe the multi-class SVM classifier as used in this

study, and describe the approach for optimizing the classifier.

4.1 Classifier selection

In order to select a suitable classifier, we evaluated six classifiers selected from different categories, available in the RapidMiner toolkit [25]. The classifiers are the commonly used methods in the data mining literature, and represent comparative approaches:

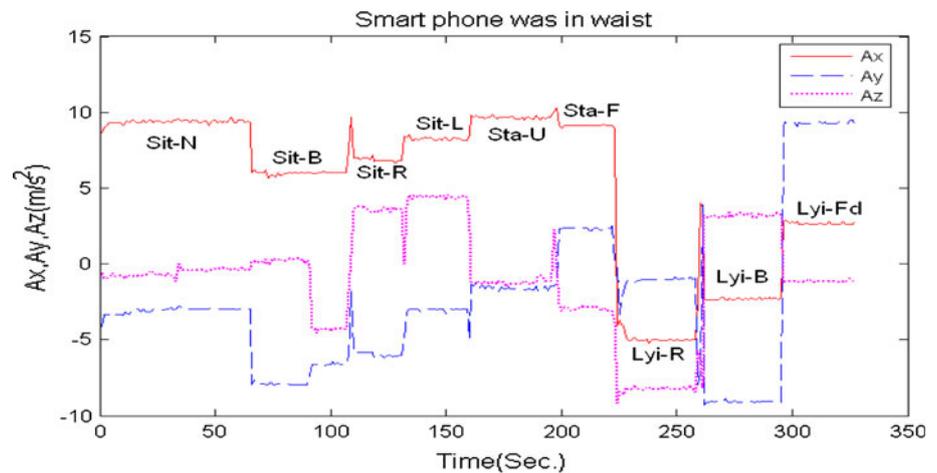
K-NN (k-nearest neighbours): it is a type of instance-based learning algorithm based on an explicit similarity measure [8].

Naïve Bayes: it is probabilistic classifier using estimated normal distributions [11].

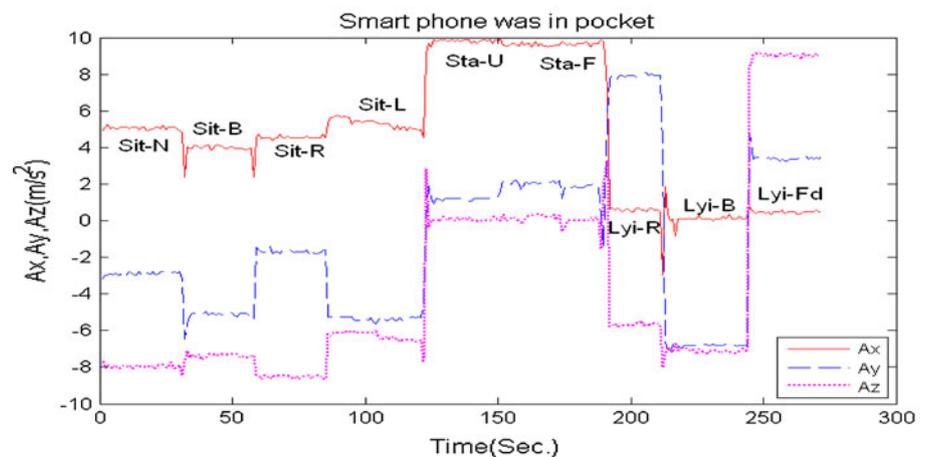
Decision Tree: the decision is determined by the criterion using a tree-like model [32].

W-DTNB (Weka-Decision table Naïve bayes): it performs the Weka learning rules using a decision table/naïve bayes hybrid classifier [14].

Fig. 3 Comparison 3D-acceleration signals for different postures and for different phone positions



(a) 3D-acceleration signals measured by a smart phone in waist.



(b) 3D-acceleration signals measured by a smart phone in pocket.

Neural Net (Artificial Neural Network): it learns a neural net by means of a feed-forward neural network. The learning is done via back propagation [6].

LibSVM (Support Vector Machine): a SVM constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space to map the testing set. It is based on the statistical learning theory. LibSVM is a library for SVMs developed by Chang and Lin [9]. It supports multi-class classification.

All classifiers were run on data sets in two different settings and using the model application process.

- *Setting1*: Data collected from six subjects (each with nine postures) were randomly separated into equally sized (50%) subsets. One of the subsets was used as the training set, and the other as the testing set.
- *Setting2*: The training set is same as *Setting1*, and data collected from a further seven subjects were used as a testing set only.

The experimental results show that all classifiers perform better in *setting1* (95.07–98.98%) than in *setting2* (57.66–70.29%). The reason is that testing data for *setting1* is from the same subjects as the training set, whereas testing data for *setting2* is from different than the training set. Nevertheless, the classification ability for previously unseen data is important in evaluating classifier’s performance. Hence classifier selection was conditioned on three aspects: (a) classification accuracy for *setting2*, (b) the balance of the precision for nine classes, and (c) the execution time. Table 1 shows that two classifiers LibSVM and Neural Net have similar ability for the aspect (a) (70.29 vs. 69.18%) and aspect (b) (46.39–100% vs. 54.7–100%). However, LibSVM is faster than Neural Net (less 1 vs. 3 s). Considering that the classification time may be important for our intended application that is real-time unhealthy posture reminder system. Therefore we selected the LibSVM classifier for the multiple posture recognition.

4.2 Multi-class SVMs

Multi-class SVMs are usually realized by combining several binary SVMs [16]. There are four types of SVMs that

manage multi-class problems: one-vs-all, one-vs-one, error-correcting output code, and all-vs-all [1].

Figure 4 shows an example for binary classification of linearly non-separable data such as sitting and standing. The decision boundary is given by $d(x, w, b) = w^T x + b = 0$. For the overlap, the algorithm allows a *soft margin* $d(x, w, b) = w^T x + b = \pm 1$. The width of a soft margin can be controlled by a corresponding penalty parameter C . In order to find the optimal separating hyperplane with the optimal margin, the linear non-separable SVM classifier can be formulated as shown in Eq. (1).

$$\text{minimize: } \frac{1}{2} w^T w + C \sum_{i=1}^m \xi_i$$

$$\text{constraints: } y_i [w^T x_i + b] \geq 1 - \xi_i, \quad i = 1 \dots m, \xi_i \geq 0 \quad (1)$$

where weights w is the subject of learning; C is a design parameter called *error cost* that can be adjusted by users, and can either decrease or increase the penalty errors; the scalar b is called a bias; ξ denotes a slack variable. The parameter C controls the trade-off between slack variable penalty and the margin.

The SVM machine learning algorithms solve the constrained minimization problems by using a Lagrange function and Karush–Kuhn–Tucker (KKT) conditions [12]. After training, the classifier obtains a SVM model (decision hyperplane), and then, the classifier can predict the class membership for new patterns using an indicator function, such as the class of a pattern x_j is determined by Eq. (2).

$$\text{class}(x_j) = \text{sign}(y_j) = \text{sign} \left(\sum_{i=1}^m \lambda_i y_i x_i^T \cdot x_j + b \right), \quad \lambda_i \geq 0 \quad (2)$$

where x_j denotes the new instance, x_i denotes the training instances, y_i denotes the corresponding class labels, $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ is the set of Lagrange multipliers of the training instance with $\lambda_i \geq 0$, and $\lambda_i > 0$ represent the support vectors (SVs), and $\lambda_i = 0$ signify non-SVs. Patterns that are non-SVs do not influence the classification result in Eq. (2). This means to classify a new instance x_j , significant computational time can be saved when the number of SVs is small.

Table 1 Comparison of the performance for different classifiers based on two data settings

Classifier	Accuracy (%) (apply model)		The range of nine classes precision (%)	Execution time for 2,350 instances (s)
	Setting 1	Setting 2		
K-NN	98.98	66.43	28–100	Less 1
Naïve Bayes	97.58	70.01	31–100	Less 1
Decision Tree	98.28	68.02	30–100	Less 1
W-DTNB	95.32	53.27	33–100	Less 1
Neural Net	96.61	69.18	54.7–100	3
LibSVM	95.07	70.29	46.39–100	Less 1

Bold values indicate best accuracy, best precision, and longest execution time

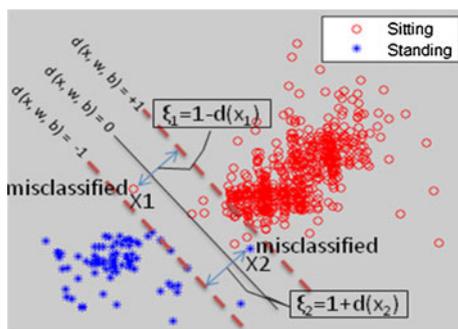


Fig. 4 Binary classification of linearly non-separable data

A *kernel function* is used to transform the input vectors into the higher-dimensional feature space in order to construct the optimal separating hyperplane. For example, linear, polynomial and radial basis function (RBF) are three basic kernels. A usual kernel function is the RBF as Eq. (3) shown. Here γ is kernel parameter, which controls the kernel radius. For more details of SVMs, see e.g. Vapnik and Lerner [30].

$$\text{RBF: } k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \quad \gamma > 0 \quad (3)$$

In this study, a software package LIBSVM [9] was used to implement the classifier algorithms. It uses the one-versus-one method [19]. For the k -class activity classification problem, the basic idea of the method is to construct $k(k-1)/2$ classifiers and each one is trained on data from two classes. After all $k(k-1)/2$ classifiers are constructed, a voting strategy is used for testing. Using this strategy, each binary classification is considered to be a voting, and then the point is predicted in the class with the largest vote. So it is also called the “Max Wins” strategy. In case two classes have identical votes, the one with the smallest index will be selected. This study will recognize 9 postures (9 classes), so 36 binary classifiers will be constructed.

4.3 Obtaining the optimal SVM model

A SVM model is determined by training data, kernel function and parameters. This study used the RBF, since it can handle the case when the relation between classes and features is nonlinear, and it also has less parameters than other nonlinear kernels such as the polynomial kernel. The number of kernel parameters influences the complexity of model selection. After the kernel function was decided, the optimal training data selection and best parameters setting can improve the classification accuracy and computational efficiency. The parameters include penalty parameter C that controls the soft margin and the gamma (γ) for the RBF kernel, which controls the kernel radius.

4.3.1 Optimal training data selection using the pick-out approach

To classify a new pattern using the SVM classifier, the dot product between the new pattern and every support vector (SV) from the SVM model is calculated. Patterns that are non-SVs in the training sets do not influence the classification result. This means the computational time can be reduced if the number of SVs is small in a training model. In addition, the value of SVs should build a proper decision hyperplane to classify new data. The size of the training data and parameters settings determine the number of SVs. The proper value of SVs depends on the training data values. Optimal training set selection is important for the optimal SVM model training.

Generally, model training uses large samples of randomly selected training data in order to characterize the classes, the remaining small samples as testing data [16]. In this way, satisfactory testing results can only be found when testing and training data came from same subjects. However, the approach may produce poor generalization to new subjects [27]. Moreover, the large training set leads to the training model that includes more Support Vectors (SVs). This will increase the computational consumption during the testing stage. The problem to be solved is how to reduce the training dataset and without degrading the final classification result. A number of studies have focused on data selection for SVM training. For example, Sohn and Dagli [28] have described a method that used the K -nearest neighbor algorithm to calculate the fuzzy class membership for each of the samples in the training dataset. Nevertheless, it requires huge computational expense for large training sets. Koggalage and Halgamuge [20] used the k -means clustering technique to find initial clusters that are further altered to identify non-relevant samples, and hence reduce the training set size. Abe and Inoue [2] used Mahalanobis distance method to estimate boundary data. In this study, nine ubiquitous postures in daily life were classified using the SVM algorithm. The classification ability of a SVM classifier relies on building an appropriate hyperplane from the limited training data rather than the number of the training data described in Sect. 4.2. The data shown in Fig. 3a indicates many similar attributes values existing in the nine postures. Within a large number of similar training samples maybe only one of them contributes to the hyperplane building. However the large number training data will raise the computation cost for the model training and data testing.

The Pick-out method (pick the optimal data out) is proposed for the training data selection in this paper. This method is motivated by the SVM decision function that is fully determined by only a small subset (SVs) of the training data. Consequently, it is attractive to remove

samples from the training set that have less influence on the final decision function. In this method, iterative and incremental learning technique is used to select the proper and smallest training set efficiently. The Pick-out approach is a boosting algorithm. The idea is that one subject performs each of nine postures in several different poses such as sitting leaning right in different angles. The typical feature values are picked out and labeled from each of postures as the single training set SingleD to train a single model named SingleM. The interval scaled values of 3D-acceleration (a_x , a_y , a_z) for each posture are defined as typical feature values here. Then the model SingleM is used to classify unseen data from the other five subjects (total six training subjects), and adds the interval scaled misclassified data from the five new subjects into the previous training set SingleD as new training set 6PickD to train a new model called 6PickM. The advantage of the Pick-out method is that the training data can be optimized to be representative and also the smallest set. It saves computational time for the typical data picking using the method of added misclassified data based on a single model. Figure 5 shows the processing of training the model 6PickM using the Pick-out approach. The size of the data set SingleD and 6PickM as well as the instances for each posture are shown in Table 2 below Sect. 4.3.2.

In order to compare with the conventional approach, a model named 6LargeM was trained on a large training set 6LargeD that is 50% of each of the nine postures random selected from the six training subjects. The number of training data in 6LargeD is more than in 6PickD (943 vs. 125 as Table 2 shown). However 6PickD includes all typical feature values from the six subjects, since 6PickD uses the Pick-out method to add the misclassified data from all data of the six subjects based on a weak classifier SingleM as described above. The two SVM models (6PickM and 6LargeM) will be compared from both

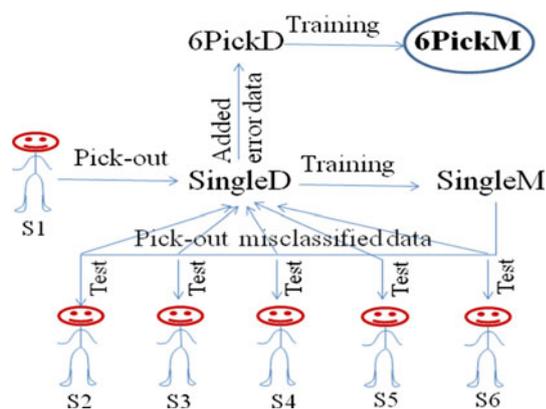


Fig. 5 Pick-out approach processing. A optimal model 6PickM trained on small training set 6PickD drawn from incrementally added typical data from six training subjects (s_1, \dots, s_6)

aspects classification accuracy and computational time in the experiments section.

4.3.2 Grid-search and best parameters setting

There are two parameters (γ , C) chosen by users during the training stage while using RBF kernels as above discussed. An optimal model is trained by optimal training data with the best pair parameters (γ , C). Generally, choosing the best parameters combines the Grid-search algorithm with a k-fold cross validation [16, 22]. However, this study found that it is not rigorous if the best parameters are decided by the cross validation accuracy only. In order to get the best parameters, this study performed two steps: better parameters region searching and best parameters deciding according to its better region. The better parameters region searching combined the grid-search with k-fold cross-validation algorithms as usual. In k-fold cross-validation, the data set is divided into k subsets of equal size. Each of subsets is tested using the classifier trained on the remaining k-1 subsets. Study [16] shows that trying exponentially growing sequences of C and γ is an efficient method for grid searches. This study tried the parameters $\gamma = 2^0, 2^{-1}, \dots, 2^{-12}$ and $C = 2^0, 2^1, \dots, 2^{10}$. Therefore, $13 \times 11 = 143$ pairs of parameters are tried and the region with the better cross-validation accuracy is chosen as the better region. The k-fold number was designed different for the three kinds of data sets as Table 2 shown ($k = 20$ and $k = 10$), since their data number was different. For the small data set, the larger k will be better, because thus leaves more instances in the training set [30]. The instances number for each posture, k numbers and the better parameters region for each training data set are shown in Table 2. Note that the boundary of the better parameters' region is set according to ordered validation accuracy from the above tried 143 pairs of parameters. For example, the training set 6PickD acquired validation accuracy is ordered from 80 to 41%, tried the parameters C is from 1 to 1,024 and γ from 1 to 0.000244 using the grid-search method. The better validation accuracy is from 80 to 71% for the 6PickD as shown in Table 2, consequently the corresponding better parameters region was selected as shown in Table 2.

Table 2 shows that the 6PickD obtained lower validation accuracy among the three training sets. The reason is that the 6PickD data set was picked up from six different subjects using the Pick-out method described above. The data number is small and not balance. The higher numbers of misclassified postures were added. The optimal model is derived according to its predicting accuracy for new data, not the cross-validation accuracy.

After identifying the better region on the grid, a further grid search on that region is carried out for choosing the

Table 2 Instances number and the better (γ , C) region with cross-validation accuracy for three training data sets

Data Sets	Instances for each posture									Total	K -fold	Better γ region	Better C region	Validation accuracy (%)
	Sit-N	Sit-B	Sit-R	Sit-L	Sta-U	Sta-F	Lyi-R	Lyi-B	Lyi-Fd					
SingleD	10	10	10	10	13	10	10	10	10	93	20	0.125–0.5	1–64	97–99
6PickD	14	18	15	15	15	15	11	12	10	125	20	0.039–1	1–128	71–80
6LargeD	77	93	81	87	101	124	113	125	142	943	10	0.001–0.04	1–16	85–89

Table 3 Best parameters (γ , C) choosing for the model SingleM

γ value	C value	Training accuracy (%)	Validating accuracy (%)	Testing accuracy for each subject						
				S1 (%)	S2 (%)	S3 (%)	S4 (%)	S5 (%)	S6 (%)	Average (%)
0.6	1	100	98	100	66.9	77	81.4	78.1	89.4	82.1
0.1	1	100	99	100	67.8	76.7	81.8	83.1	99.6	84.8
0.09	1	100	99	100	68.1	76.8	81.4	82.5	99.2	84.7
0.1	4	100	94.1	100	68.1	76.8	81.8	82.9	99.6	84.8
0.1	7	100	94	100	77.1	82	85.8	83.1	99.6	87.9
0.1	10	100	94	100	66	76.5	81.8	82.9	99.6	84.5

Bold values indicate best parameters for validation accuracy or testing accuracy

best parameters. In this stage, the whole training data with different (γ , C) setting according to the enlarged better region is used to train a model. Meanwhile, the model is used to test on data from the six subjects. The best parameters are chosen according to the average testing accuracy. As an example, Table 3 shows part of results for the best (γ , C) selection based on the model SingleM training and testing. First keep C value as a constant to try γ value from large to small in a small interval according to its better region (such as from 0.6 to 0.09, interval = -0.01). The best γ value is decided by the best average testing accuracy. Second, keep the best γ value as a constant to try C value from small to large in a small interval (such as from 4 to 10, interval = 1). Finally the best pair parameters (γ , C) = (0.1, 7) is obtained according to the higher average testing accuracy (87.9 vs. 84.8%). Note that the best parameters are decided by the average testing accuracy, not their validating accuracy (94 vs. 99%). For the SingleM, the best $\gamma = 0.1$ is out of its better region from 0.125 to 0.5, so the enlarged better region is necessary. The reason is that a big interval was used in the first step (such as 2^n or 2^{-n}) for the roughly better region searching. It is possible to improve the grid search efficiency, but it maybe decrease the region of interest.

In same way, the model 6PickM and 6LargeM were trained with the best parameters (γ , C) = (0.06, 10) and (γ , C) = (0.02, 1), respectively. The optimal model is trained by optimal training data with best parameters simultaneously. If the training data set is changed, the best parameters need to obtain from a grid-search again. This is why the k -fold cross validation accuracy can be used to

search the better region of parameters only, but it is not suitable to decide the best parameters.

5 Experiments

Two models 6PickM and 6LargeM were trained using different approaches (pick-out small training set vs. conventional large training set). The experiments were performed by 13 subjects (ages from 23 to 50) in the laboratory where a video was fixed in the ceiling for the validating the experiments results by synchronized video recording. Thirteen subjects were arranged into two groups. Group1 includes six subjects whose data were divided into two subsets of 50% each, randomly. One such subset is used as the training set and the other as the test set. Group2 includes seven subjects whose data are considered as testing set only. The experiments were designed in two sessions: (1) Compare the testing accuracy between two models by two groups: testing data and training data came from same subjects (Group1) and different subjects (Group2). (2) Compare the computational time between two models. All subjects in the two groups were measured undertaking the nine postures using a belt-worn HTC phone. The data sampling frequency was set at 1 Hz, since it is fast enough for the posture recognition. Four channels of data (t , a_x , a_y , a_z) were stored in the phone as a text file for each experiment. And then, the data set were formed based on two sets of variables such as the input set x and the classes label set y represented as $(x_1, y_1), \dots, (x_m, y_m)$, where each x is a vector of acceleration values in three

dimensions, represented as $x = (a_x, a_y, a_z)$ and $y_i = f(x_i) = (1, 2, 3, 4, 5, 6, 7, 8, 9)$ represents nine classes: Sit-N, Sta-U, Lyi-S, Sit-B, Sta-F, Lyi-B, Sit-R, Sit-L, and Lyi-Fd. As the experiment is simulating real daily life, the test protocol was designed very flexible. For example, subjects were asked to undertake a series of activities in a random order along with their natural unconstrained behavior.

5.1 Classification accuracy comparison

First, the classification accuracy was compared between two models for the six training subjects in group1. The testing data were the remaining 50% data of each of six subjects after training. The experiment results for each subject were shown in Table 4. It includes the instance number of each posture and testing accuracy for two models.

Table 4 compares testing accuracy for each subject based on two models, both random in turn on the higher position. 6LargeM produced higher average accuracy than 6PickM for the training subjects in group1 (95.1 vs. 91.6%), since 6LargeM was trained on the large training set from the six training subjects, it is familiar with the remaining data.

The classification accuracy was compared between two models for the previously unseen data also. The unseen data were collected from seven new subjects in group2. The experiment details for each subject that include the instances number of each posture and testing accuracy are shown in Table 5. Table 5 shows that 6PickM got higher accuracy for each subject than 6LargeM except the subject10.

Figure 6 compared the average accuracy of two models for each of the two groups and the total average of the two groups. The model 6PickM has a little lower average accuracy than 6LargeM for the training subjects in group1 (91.6 vs. 95.1%), but it has higher average accuracy than 6LargeM for the unseen data in group2 (85.1 vs. 70.3%). This means 6PickM has better generalization capability for

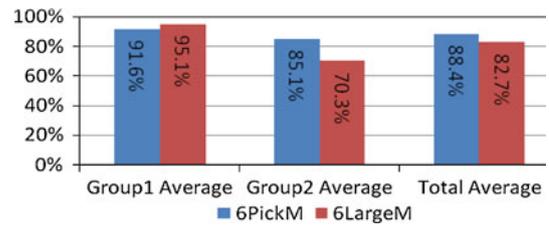


Fig. 6 Comparisons the average accuracy for two groups

Table 4 Comparison testing accuracy between two models for training subjects

ID	Instances for each posture									Accuracy		
	Sit-N	Sit-B	Sit-R	Sit-L	Sta-U	Sta-F	Lyi-R	Lyi-B	Lyi-Fd	Total	6PickM (%)	6LargeM (%)
S1	21	21	19	20	22	26	18	17	23	187	98.4	89
S2	32	22	22	29	48	54	45	34	43	329	83.9	99.4
S3	30	56	41	44	28	36	46	46	40	367	78.5	97.3
S4	26	35	23	19	31	30	40	33	37	274	91.2	93.3
S5	29	27	28	31	48	69	37	70	100	439	98.4	92.2
S6	14	21	26	27	29	25	37	47	37	263	98.9	99.2
Total	152	159	159	170	206	240	223	247	280	1859	91.55	95.07

Bold values indicate the average accuracy for the training subjects

Table 5 Results of classification for previously unseen data using two models

ID	Instances for each posture									Accuracy		
	Sit-N	Sit-B	Sit-R	Sit-L	Sta-U	Sta-F	Lyi-R	Lyi-B	Lyi-Fd	Total	6PickM (%)	6LargeM (%)
S7	37	44	34	34	64	40	48	43	46	390	78.2	76.4
S8	73	50	35	49	123	71	63	49	43	556	97.5	87.2
S9	32	23	33	35	38	34	35	35	32	297	79.5	56.9
S10	88	28	26	21	35	45	29	50	40	362	88.4	89
S11	22	18	20	33	34	19	17	18	24	205	91.2	51
S12	35	29	27	29	38	27	29	31	30	275	89.1	66.2
S13	11	25	14	26	108	21	18	32	10	265	71.7	65.3
Total	298	217	189	227	440	257	239	258	225	2350	85.09	70.29

Bold values indicate the average accuracy for the testing subjects

new subjects than 6LargeM, since it includes more typical feature values. Every subject in group2 is new to both models, so the accuracy for group2 was lower than for group1 classified by the two models.

The confusion matrix with the accuracy of each posture based on the model 6PickM for all subjects in the two groups was shown in Table 6. Consider the accuracy calculation for each posture problem, classification result for each posture is given as (Cor_i, Tot_i) from Table 6. Where $i = 1 \dots 9$ represents the nine postures. Cor_i stands for the correct number in the diagonal line for each posture and Tot_i indicates the total number in the row of Total for each posture. The accuracy for each of the postures was calculated by Eq. (4). The total average accuracy for all postures was calculated by Eq. (5). Table 6 shows that the three kinds of lying postures (Lyi-R, Lyi-B and Lyi-Fd) were mostly correctly to a high degree (in 95.7, 99.8, and 100% of the each case respectively). In addition Sta-U also got a high accuracy in 94.1%. However the three types of sitting postures (Sit-B, Sit-R and Sit-L) were confused with each other most often (76.1, 77.9 and 60.1% of the each cases respectively).

$$A_i = \frac{Cor_i}{Tot_i} \times 100\% \quad (4)$$

$$A = \frac{\sum_{i=1}^9 Cor_i}{\sum_{i=1}^9 Tot_i} \times 100\% \quad (5)$$

5.2 Computational time comparison

The model is trained once and reused for testing. Here the computational time only assesses the testing time, not the

training time. The SVs number is related with the number of training instances and the value of parameters. In theory, testing time will be reduced when the number of SVs is small according to Eq. (2). This verified by the experiments below by testing different numbers of instances using the two models 6PickM and 6LargeM. Details of the two models are shown in Table 7.

Table 7 shows that 6LargeM has more SVs than 6PickM (623 vs. 109) which will cost the computational time when classifying a new pattern.

In order to compare the testing time for different number of instances using the two models, all data from 13 subjects were random organized in several files. The small number of data was combined different number of subjects' data together such as 6 subjects or 13 subjects. The large number of data was copied the 13 subjects' data several times. The details of experiments result were shown in Table 8. The classification was performed offline on a computer, which used the Intel 3 GHz processor with 2 GB of RAM. Table 8 shows that the testing time was improved using 6PickM compared to 6LargeM. For small number of data, the computational efficiency was improved in a range from 22 to 43.9%. Nevertheless, the improvement of testing time was stable in 28% based on 6PickM than 6LargeM when the data size is more than 33,572 as Fig. 7 shown.

5.3 Evaluation accuracy by different classifiers

The Pick-out method reduces the training set size whilst improving the classification accuracy. In order to evaluate this approach, we used six different classifiers to run on

Table 6 Confusion matrix with the accuracy of each posture using 6PickM for all subjects

Postures	Sit-N	Sit-B	Sit-R	Sit-L	Sta-U	Sta-F	Lyi-R	Lyi-B	Lyi-Fd	Average accuracy
Sit-N	398	5	39	71	8	0	0	0	0	
Sit-B	0	306	41	78	0	0	0	0	0	
Sit-R	3	64	293	1	0	0	0	0	0	
Sit-L	12	26	0	226	0	0	0	0	0	
Sta-U	35	1	3	0	608	58	0	0	0	
Sta-F	2	0	0	0	30	439	0	0	0	
Lyi-R	0	0	0	0	0	0	442	1	0	
Lyi-B	0	0	0	0	0	0	20	504	0	
Lyi-Fd	0	0	0	0	0	0	0	0	505	
Total	450	402	376	376	646	497	462	505	505	
Accuracy	88.4%	76.1%	77.9%	60.1%	94.1%	88.3%	95.7%	99.8%	100%	88.2%

Bold values in the diagonal line indicate the correct number for each posture

Table 7 Details about the two SVM models

Model name	Training data	Instances number	Feature number	Class number	Best parameters (γ , C)	SVs
6PickM	6PickD	125	3	9	(0.06, 10)	109
6LargeM	6LargeD	943	3	9	(0.02, 1)	623

Table 8 Comparison testing time for different number of instances based on two models

Number of instances	Testing time (s)		T2-T1 (s)	Percentage for improvement (%)
	6PickM (T1)	6LargeM (T2)		
1,241	0.157	0.28	0.123	43.9
1,858	0.29	0.484	0.194	40
4,219	0.547	0.703	0.156	22.2
8,438	0.969	1.297	0.328	25.3
16,876	1.688	2.454	0.766	31.2
33,572	3.469	4.672	1.203	25.7
595,800	57.6	80.5	22.9	28.4
617,693	59.89	83.5	23.61	28.2
702,104	67.844	94.719	26.875	28.3

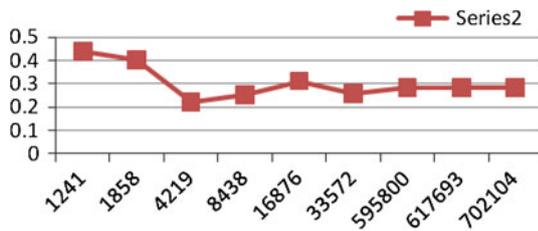


Fig. 7 Percentage of testing time improvement for different number of data based on model 6PickM than 6LargeM. It is stable in 28% for the large number of data

two different training models *6PickM* and *6LargeM*, and each of the two models was tested using the two same data sets named as *Testing1* and *Testing2* described below. The results are shown in Table 9.

- *6PickM*: trained on data *6PickD* which is optimized small set using Pick-out method from six training subjects. Details were described in Sect. 4.3.1.
- *6LargeM*: trained on data *6LargeD* which is 50% of each of the nine postures were randomly selected from the six training subjects.
- *Testing1*: was the subsequent 50% of the subset come from six training subjects relating to *6LargeD*.
- *Testing2*: was collected from a further seven subjects.

The results in Table 9 show that the training model *6LargeM* achieves very different accuracy for the two testing sets. It performs with very high accuracy (95–99%) for the training subjects (*Testing1*) for each of the six classifiers, but with low accuracy (53–70%) for the unseen data (*Testing2*) for each of the classifiers. However, *6PickM* produces similar accuracy for both testing sets and higher accuracy than *6LargeM* for the *Testing2* (unseen data). Running classifiers on unseen data is important for many applications. These results indicate that the *6PickM* trained using Pick-out approach, whilst achieving lower accuracy on training subjects (*Testing1*) is superior for

Table 9 Evaluation of the Pick-out method by six classifiers based on two training models with two testing sets

Classifier	Accuracy (%)			
	6LargeM		6PickM	
	Testing1	Testing2	Testing1	Testing2
K-NN	98.98	66.43	74.7	80.74
Naïve Bayes	97.58	70.01	78.85	78.18
Decision Tree	98.28	68.02	77.23	74.95
W-DTNB	95.32	53.27	67.22	62.29
Neural Net	96.61	69.18	88.21	82.66
LibSVM	95.07	70.29	91.55	85.09

unseen data (*Testing2*). Moreover it is also suitable for most types of classifiers, not only for SVM classifier.

5.4 Evaluation the two steps grid-search algorithm

A two steps grid-search algorithm was compared with the one step cross-validation algorithm for the best parameters selection. We ran the SVM classifier on the two training sets *6PickD* and *6LargeD* with the same testing set (unseen data *Testing2* described in Sect. 5.3) respectively based on the two pairs of the best parameters (γ , C) selected using the two algorithms: cross-validation and two steps grid-search. The classification accuracy based on the best parameters for each of the two training sets and each of the two algorithms is shown in Table 10. The results indicate that the new grid-search algorithm performs better than the cross-validation, especially for the small size training set *6PickD* (85.09 vs. 66.94%).

6 Discussion

The classification accuracy and computational time are related to the SVM model. An optimal model is trained by

Table 10 Evaluate the two algorithms for parameters selection using the two training sets 6PickD and 6LargeD based on corresponding two pairs of the best parameters

Training set	Algorithm	Parameters		Testing set	Accuracy (%)
		γ	C		
6PickD	Cross-validation	0.5	4	Testing2 (unseen data)	66.94
	Two steps grid-search	0.06	10		85.09
6LargeD	Cross-validation	0.00781	1		68.87
	Two steps grid-search	0.02	1		70.29

Bold values indicate the higher accuracy of the two algorithms based on each training set

the optimal training data with best parameters simultaneously. Conventional guidance on the model training is designed to use large samples of randomly selected training set. A novel Pick-out approach which is different to the generally promoted approach for training set selection was discussed in this study. This approach picked a small optimal training set using incrementally added misclassified data from six training subjects during the testing by a single training model. In addition, the best parameters were chosen using two steps grid-search algorithm. First, a better region searching combined the grid-search with k -fold cross-validation algorithms. And then, best parameters are decided by combining the grid-search with model training and testing according to its better region. This new approach for optimal model selection was evaluated against conventional approaches with nine prevalent postures classification from belt-worn smart phone's accelerometer data. Two models (6PickM and 6LargeM) were compared using: classification accuracy and computational efficiency. The main result was that classification derived from 6PickM (trained on small optimal training set) and 6LargeM (trained on conventional large randomly selected training set) did differ in generalization to new data and computational time. 6LargeM produced higher average accuracy for the training subjects (95.1 vs. 91.6%). Nevertheless, 6PickM obtained higher average accuracy for new subjects (85.1 vs. 70.3%). Computational time was improved 28% compared with 6LargeM. The Pick-out method is an incremental modeling approach. It can get more typical feature values by the increment of model learning. Thus the model can be optimized in both aspects: high classification accuracy for unseen data and computational efficiency, since it learned the most characteristic values and includes a small number of support vectors (SVs).

Classification of the human nine postures in daily life for thirteen subjects was achieved with the multiclass SVM classifier. Some postures were classified with high accuracy, but also some misclassification of the nine postures occurred. Each of the misclassified postures became confused with one or two other postures. Most of these postures closely resembled each other (e.g. Sta-U and Sta-Fd).

Specially, some subjects habitually take the posture Sit-B with Sit-L or Sit-R at the same time, so more confusion happened between Sit-L with Sit-B or between Sit-B with Sit-R. Additionally, if the subject only leaning a small angle when he/she was Sit-L or Sit-R, the confusion also happened between Sit-L with Sit-N or between Sit-R with Sit-N. Overall, the model 6PickM exhibits a quite robustness for both training subjects (91.6%) and previously unseen new subjects (85.1%). The nine postures were recognized in different accuracy: High level values for Lyi-Fd (100%), Lyi-B (99.8%), Lyi-R (95.7%), and Sta-U (94.1%); Average level values for Sit-N (88.4%) and Sta-F (88.3%); Lower precision values for Sit-B (76.1%), Sit-R (77.9%) and Sit-L (60.1%). Nevertheless, the limitation on the three types of sittings recognition can be ignored for the further posture-aware reminder systems, since the sitting postures could define in two groups: sitting healthy (Sit-N) and sitting unhealthy (Sit-Uh which includes Sit-B, Sit-L and Sit-R), thus the confusion happened most often between Sit-B, Sit-L and Sit-R is not significant for unhealthy posture alert.

7 Conclusion and future work

This study discussed a novel method which selects typical training data (Pick-out) with a two step grid-search algorithm for optimal model selection. The Pick-out method adopted the iterative and incremental learning technique. The advantages of the Pick-out approach compared with other data reducing methods are: its ability to reduce the training set size with lower computational cost (e.g. a distance calculation is not required); faster selection of new attributes (simply picking the misclassified typical data); improvement of the generalization performance of the resulting classifiers (not only suitable for SVM, as shown in Table 9). The Pick-out method can use incremental learning, accumulating a representative training set in a domain such as Activities of Daily Living. Additionally, Experiments show that the model trained on data selected by Pick-out approach is competitive with the standard approach of use of a large randomly selected training set.

Picking typical data out is a promising method to train an optimal model with high classification capability for unseen data and saving computational time. This is important for real-time activity-aware reminder applications. Accelerometers embedded in a smart phone can be used to easily monitor various postures in human daily life. Multiclass SVMs proved reliable in classification of human daily activities.

An optimal model selection is important for the daily activity classification. Nevertheless, the testing data with training set should keep identical sensing criterion. In this study, the activity data were sensed by one accelerometer embedded in a smart phone, so the phone position also influences the classification result. In this research, the model was trained on data from the position of middle left waist. If a subject puts the phone on right side or front of the waist, even put it in a pocket, the feature values (i.e. acceleration values in three dimensions) will be different with the training set for same posture, thus the classification accuracy will be decreased if the phone was in a different position during testing with training. Hence, a robust monitoring system should detect the phone position first before the data collecting. We have done this work using a retrained SVM classifier to detect the phone position, and run it in real-time before collecting the posture data. If the phone was not in middle left waist, the system will alert a sound in real-time until the phone position was in predefined position such as left middle waist for this study.

The future work will verify the model that was trained using the novel Pick-out approach by more subjects. In addition, a real-time activity recognition embedded with the posture-aware reminder system will be implemented using a smart phone only. This would enable a reasonable reminder to issue when the subject is in an unhealthy posture such as sitting on back arch for a predefined period.

Acknowledgments The authors acknowledge the support of University of Ulster Vice Chancellor Scholarship Programme, and thank all members of the Smart Environments Research Group for their help with collecting the experimental data. Special thanks are due to Mr W Burns for assisting with the code for data collection using the HTC phone.

References

1. Abe S (2005) Support vector machines for pattern classification. Springer-Verlag, New York
2. Abe S, Inoue T (2001) Fast training of support vector machines by extracting boundary data. *Artif Neural Netw ICANN* 2130:308–313
3. Allen FR, Ambikairajah E, Lovell NH, Celler BG (2006) Classification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models. *Physiol Meas* 27(10):935–952
4. Bao L, Intille SS (2004) Activity recognition from user-annotated acceleration data. *Pervasive Comput* 3001:1–17
5. Boulay B, Brémond F, Thonnat M (2006) Applying 3D human model in a posture recognition system. *Pattern Recogn Lett* 27(15):1788–1796
6. Brown EN, Kass RE, Mitra PP (2004) Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nat Neurosci* 7:456–461
7. Bussmann J, Martens W, Tulen J, Schasfoort F, VAN DEN BE (2001) Measuring daily behavior using ambulatory accelerometry: the activity monitor. *Behav Res Methods Instrum Comput* 33(3):349
8. Bremner D, Demaine E, Erickson J, Iacono J, Langerman S, Morin P, Toussaint G (2005) Output-sensitive algorithms for computing nearest-neighbor decision boundaries. *Discrete Comput Geom* 33(4):593–604
9. Chang CC, Lin CJ (2001) LIBSVM: a library for support vector machines, (2001) Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
10. Cucchiara R, Grana C, Prati A, Vezzani R (2005) Probabilistic posture classification for human-behavior analysis. *IEEE Trans Syst Man Cybern Part A* 35(1):42–54
11. Domingos P, Pazzani M (1997) On the the optimality of the simple Bayesian classifier under zero-one loss. *Mach Learn* 29(2–3):103–130
12. Fletcher R (1981) Practical methods of optimization: constrained optimization. vol 2. John Wiley & Sons, Inc., Somerset, NJ, p 224
13. Foerster F, Smeja M, Fahrenberg J (1999) Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Comput Hum Behav* 15(5):571–583
14. Hall M, Frank E (2008) Combining naive Bayes and decision tables, In Proc 21st Florida Artificial Intelligence Research Society Conference, Miami, Florida. AAAI Press
15. Harms H, Amft O, Tröster G, Roggen D (2008) Smash: a distributed sensing and processing garment for the classification of upper body postures. Proceedings of the ICST 3rd international conference on body area networks ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp 22
16. Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. *IEEE Trans Neural Netw* 13(2): 415–425
17. Juang CF, Chang CM (2007) Human body posture classification by a neural fuzzy network and home care system application. *IEEE Trans Syst Man Cybern Part A* 37(6):984–994
18. Karhu O, Kansi P, Kuorinka I (1977) Correcting working postures in industry: a practical method for analysis. *Appl Ergon* 8(4):199–201
19. Knerr S, Personnaz L, Dreyfus G, Fogelman J, Agresti A, Ajiz M, Jennings A, Alizadeh F, Alizadeh F, Haeberly J (1990) Single-layer learning revisited: a stepwise procedure for building and training a neural network. *Optim Methods Softw* 1:23–34
20. Koggalage R, Halgamuge S (2004) Reducing the number of training samples for fast support vector machine classification. *Neural Inf Process Lett Rev* 2(3):57–65
21. Lee S, Park H, Hong S, Lee K, Kim Y (2003) A study on the activity classification using a triaxial accelerometer. Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE
22. Martiskainen P, Järvinen M, Skön JP, Tiirikainen J, Kolehmainen M, Mononen J (2009) Cow behaviour pattern recognition using a three-dimensional accelerometer and support vector machines. *Appl Anim Behav Sci* 119(1–2):32–38
23. Mathie MJ, Celler BG, Lovell NH, Coster ACF (2004) Classification of basic daily movements using a triaxial accelerometer. *Med Biol Eng Comput* 42(5):679–687

24. Mattmann C, Clemens F, Tröster G (2008) Sensor for measuring strain in textile. *Sensors* 8(6):3719
25. Mierswa I, Wurst M, Klinkenberg R, Scholz M, Euler T (2006) YALE: rapid prototyping for complex data mining tasks. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp 935–940
26. Panini L, Cucchiara R (2003) A machine learning approach for human posture detection in domotics applications. *Image Analysis and Processing, 2003. Proceedings. 12th International Conference on*, pp 103
27. Ravi N, Dandekar N, Mysore P, Littman ML (2005) Activity recognition from accelerometer data. *Proceedings of the National Conference on Artificial Intelligence*. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. pp 1541
28. Sohn S, Dagli CH (2002) Advantages of using fuzzy class memberships in self-organizing map and support vector machines. *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on IEEE*, pp 1886
29. Sung M, Marci C, Pentland A (2005) Wearable feedback systems for rehabilitation. *J Neuroeng Rehabil* 2:17
30. Vapnik V, Lerner A (1963) Pattern recognition using generalized portrait method. *Autom Remote Control* 24(6):774–780
31. Witten IH, Frank E (2005) *Data mining: practical machine learning tools and techniques*, 2nd edn. The Morgan Kaufmann series in data management system, Elsevier
32. Yang T (2006) Computational verb decision trees. *Int J Comput Cogn* 4(4):34–46
33. Zhang S, McCullagh P, Nugent C, Zheng H (2009) A theoretic algorithm for fall and motionless detection. *Pervasive computing technologies for healthcare, Proceedings of the 3rd Annual IEEE International Conference*, pp 1–6