2011 15th Annual International Symposium on Wearable Computers

Unsupervised adaptation to on-body sensor displacement in acceleration-based activity recognition

Hamidreza Bayati, José del R. Millán and Ricardo Chavarriaga
EPFL, Chair on Non-Invasive Brain-Computer Interface (CNBI)
CH-1015 Lausanne, Switzerland.
{ricardo.chavarriaga,jose.millan}@epfl.ch

Abstract

A common assumption in activity recognition is that the system remain unchanged between its design and its posterior operation. However, many factors can affect the data distribution between two different experimental sessions including sensor displacement (e.g. due to replacement or slippage), and lead to changes in the classification performance. We propose an unsupervised adaptive classifier that calibrates itself to be robust against changes in the sensor location. It assumes that these changes are mainly reflected in shifts in the feature distributions and uses an online version of expectation-maximisation to estimate those shifts. We tested the method on a synthetic dataset in addition to two activity recognition datasets modeling sensor displacement. Results show that the proposed adaptive algorithm is robust against shift in the feature space due to sensor displacement.

1 Introduction

Activity recognition from wearable sensors is largely being studied in applications like gaming [1], industrial maintenance [2] and health monitoring [3]. In particular, acceleration sensors have been applied for recognising different activities from modes of locomotion [4] to complex daily living activities [5]. Typically, the design of these systems (e.g. feature selection, classification) assumes that the characteristics of the sensor network will not change. However, during system operation body-worn sensor location may slip or rotate. Similarly, is unrealistic to expect users to precisely re-attach the sensors at the same location from day to day. In order to address the issue of sensor location variability, we propose a self-calibrating approach based on probabilistic classifiers. The method tracks changes in the feature distribution in an unsupervised manner using an online implementation of the Expectation-Maximisation algorithm (EM). We tested the method on two scenarios of Human-computer interaction (HCI).

Several methods have been previously proposed to cope with those changes in activity and gesture recognition using body-worn sensors, Kunze et. al. used gyroscope and accelerometers to distinguish between rotation and translation [6]. They show that sensor translation does not significantly affect the acceleration signals while rotations does. Using physical concept behind they proposed a heuristic method to deal with these variations and achieved higher recognition rates for displaced sensors on body segments. Other approaches have focused on the use of displacement-invariant features [7]. Förster et. al. use genetic programing for extracting invariant features [8]. They located six acceleration sensors on lower arm to simulate six sensor placements in a Human Computer Interface (HCI) scenario. They leaved one sensor out from training and used evolving features of other sensors to train a classifier. Using evolving features compared to standard features, they reported higher recognition rate and robustness against sensor displacement. In another work, the same group proposed an online unsupervised self-calibration algorithm [9]. Using online adaptation they adjusted the centres in a nearest centre classifier (NCC). They applied the method on synthetic data in addition to two real life datasets, namely the HCI scenario described above and a fitness scenario dataset.

As state above, changes in sensor placement affect the signal feature distributions amongst different sessions. A particular case, termed *covariate shift*, is when the training and testing feature distributions change but the conditional distribution of the classifier output given input is the same. Based on this assumption, Sugiyama et. al. proposed a modification of cross validation technique called importance weighted cross validation (IWCV) that can be used in model and parameter selection in classification tasks [10]. They used IWCV to select parameters in importance weighted LDA (IWLDA) were the weights are the ratio of the test and train patterns distribution in the calibration session. In experimental studies, this ratio is replaced

1550-4816/11 $26.00 © 2011 IEEE
DOI 10.1109/ISWC.2011.11

by its empirical estimates either Direct Importance Estimation by Kullback-Leibler Importance Estimation Procedure (KLIEP) or Unconstrained Least Square Importance Fitting (uLSIF) [11]. This method has been tested in Brain Computer Interface (BCI) applications [12]. However, it should be noticed that adaptation requires a calibration session to estimate the ratio of distribution between training and test session.

The rest of this paper is structured as follows, in Section 2 we describe the proposed method followed by a toy example using synthetic data (Sec 3.1). Then we validate it using the same applications introduced by Förster and Colleagues [9]; a Human-computer-Interaction (Sec 3.2) and a fitness scenario (Sec 3.3).

2 Unsupervised adaptation

Classification methods for activity recognition typically assume that the feature distribution used for training will remain the same during the system operation. As mentioned before several factors can induce changes in the system leading to a decrease in performance. We propose a method to provide online unsupervised adaptation to changes in the feature distribution resulting from sensor displacement. We assume that sensor displacement results in changes in the overall feature distribution but the conditional distributions of classes given these features remain the same (i.e. *covariate shift*) [10]. Moreover, we assume that the change in the feature distribution can be fully characterised by shift of an unknown magnitude and direction. Given this assumption, the proposed method estimates the distribution shift using an online version of the Expectation-Maximisation algorithm. Once the shift vector has been estimated, incoming samples can be *shifted back* and classified using the original classifier (i.e. the one trained in the original feature distribution).

Specifically, let $\mathcal{C}(\mathbf{x})$ be a classifier trained on data with feature distribution $p(\mathbf{x})$. If during runtime the distribution of incoming samples $\mathbf{y}$ is equal to $p(\mathbf{x})$ shifted by a vector θ, performance will not be affected if samples are shifted back before classification: $\mathcal{C}(\mathbf{y}-\theta)$. Therefore, self-adaptation can be achieved by estimating the shift vector θ in an online, unsupervised manner.

Let $p(\mathbf{x})$ be the training feature distribution,

$$p(\mathbf{x}) = \sum_{i=1}^{I} P(\mathbf{z}=\omega_i) P(\mathbf{x}|\mathbf{z}=\omega_i) \tag{1}$$

where $\mathbf{x}$ represents the features, $P(\mathbf{z}=\omega_i)$ is the prior probability of class i, I is the number of classes, and the class-conditional distribution is a normal distribution with mean $\mu_\mathbf{i}$ and covariance matrix Σ_i: $P(\mathbf{x}|\mathbf{z}=\omega_i) \equiv N(\mathbf{x}|\mu_i, \Sigma_i)$.

Let $\mathbf{y}$ be the samples recorded during system operation. Given the method assumptions, $(\mathbf{y}-\theta)$ should follow the same distribution as the training samples(Eq 1). Given a matrix $\mathbf{Y}$ where the $j-th$ column represents the j-th observation, $\mathbf{y}_j$ and $\mathbf{Z}$ be a matrix of labels, with corresponding $\mathbf{z}_j$ that are latent variables. We can define the likelihood for a specific value of θ,

$$\ln p(\mathbf{Y}|\theta) = \ln \sum_{\mathbf{Z}} p(\mathbf{Y}, \mathbf{Z}|\theta) \tag{2}$$

We use Expectation-Maximisation (EM) algorithm to maximise the likelihood over θ [13]. Given an initial shift estimation θ^{old} the *E-step* corresponds to compute the posterior probabilities given the shift vector $p(\mathbf{Z}|\mathbf{Y}, \theta^{old})$. For $j-th$ observation it is computed as follows,

$$P(\mathbf{z}_j=\omega_s|\mathbf{y}_j, \theta^{old}) = \frac{P(\mathbf{z}_j=\omega_s) P(\mathbf{y}_j-\theta^{old}|\mathbf{z}=\omega_s)}{\sum_{i=1}^{I} P(\mathbf{z}_j=\omega_i) P(\mathbf{y}_j-\theta^{old}|\mathbf{z}=\omega_i)} \tag{3}$$

The *M-step* corresponds then to evaluate θ^{new},

$$\theta^{new} = \arg\max_{\theta} Q(\theta, \theta^{old}) \tag{4}$$

where

$$Q(\theta, \theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{Y}, \theta^{old}) \ln p(\mathbf{Y}, \mathbf{Z}|\theta) \tag{5}$$

$$Q(\theta, \theta^{old}) = \sum_{j=1}^{J} Q_j(\theta, \theta^{old}) \tag{6}$$

where J is the number of patterns and $Q_j(\theta, \theta^{old})$ is defined as follows:

$$\sum_{i=1}^{I} P(\mathbf{z}_j=\omega_i|\mathbf{y}_j, \theta^{old}) \big(\ln P(\mathbf{z}_j=\omega_i) + \ln N(\mathbf{y}_i-\theta|\mu_i, \Sigma_i) \big) \tag{7}$$

In order to have a runtime estimation of the distribution shift we use an online version of Levenberg-Marquardt algorithm [14]. This yields an on-line update rule that maximises Eq. 7 using its gradient ($\mathbf{g}$) and Hessian (H),

$$\theta^{new} = \theta^{old} + \mathbf{\Delta}\theta \tag{8}$$

where,

$$\mathbf{\Delta}\theta = (H + \lambda\mathcal{I})^{-1}\mathbf{g} \tag{9}$$

Algorithm 1 Online shift estimation

for every new sample $\mathbf{y}_j$ **do**
  Compute posterior probability of the *shifted* sample using Eq. 3.
  Classify the pattern based on maximum posterior rule.
  Compute shift update, $\mathbf{\Delta}\theta$
  if $(|\mathbf{\Delta}\theta| > \Theta)$
    Update the shift θ (Eq. 8).
  endif
end for

$$\mathbf{g} = \sum_{i=1}^{I} P(\mathbf{z}_j = \omega_i | \mathbf{y}_j, \theta^{old}) \Sigma_i^{-1} (\mathbf{y} - \theta^{old} - \mu_i) \quad (10)$$

$$H = \sum_{i=1}^{I} P(\mathbf{z}_j = \omega_i | \mathbf{y}_j, \theta^{old}) \Sigma_i^{-1} \quad (11)$$

The λ term in Eq. 9 is a small positive number and $\mathcal{I}$ is identity matrix, this regularisation term prevents from inverting a singular matrix. In the current experiments λ was set to the absolute value of the smallest non-positive eigenvalue of $H + 0.01$. Although in practical applications it can be set to a fixed value to reduce the computational cost.

To sum up, given a trained Gaussian classifier–i.e. Linear or Quadratic Discriminant Analysis, LDA or QDA respectively– shifts in the feature distribution can be estimated online using Algorithm 1. In order to avoid unnecessary changes in the shift estimation when there is no change in the feature distribution, the shift θ is only updated when the magnitude of the estimated change exceeds a threshold (Θ). Note that at the beginning of the operation, an initial value for the shift has to be set. Having no knowledge about how the distribution may have changed since training, we set this value to be zero, thus assuming no change.

3 Results

3.1 Synthetic Dataset

To illustrate our method we present a toy example of a two-class problem in a two-dimensional feature space. A training dataset was generated where both classes correspond to Gaussian distributions (means: $\mathbf{m}_A = [0, 0]^T$ and $\mathbf{m}_B = [4, 4]^T$; random covariance matrices). The testing set was created from a shifted version of the same distributions (both training and testing sets contain 200 patterns per class) where the mean of both classes were shifted by a random vector θ drawn from a normal distribution with zero mean and 100 standard deviation. The update threshold Θ was set to zero.

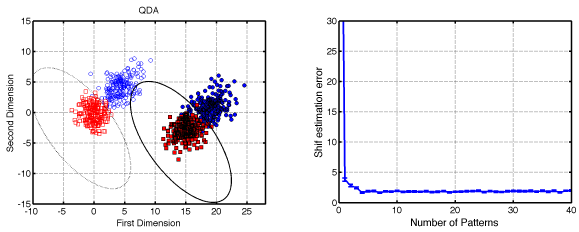

Figure 1: Evaluation of adaptive classifiers on a 2-class synthetic data. See text for details. (a) adaptive QDA. (b) Average error in the shift estimation using the adaptive QDA.

To evaluate the method we performed 100 repetitions of the simulation. For each repetition we compare the classification accuracy (CA) of a fixed Gaussian classifier with respect to the proposed adaptive classifier. In addition, we assessed the level of bias of these classifiers by computing the confusion index (CI) for a two-class problem [15],

$$CI = \left| \frac{a_1}{k_1} - \frac{a_2}{k_2} \right| \times 100\% \quad (12)$$

where a_i and k_i are respectively the number of correctly classified patterns and the total number of patterns for class i. This index, CI, is close to 100% if the classifier is biased towards one of the classes, while it tends to zero for an unbiased classifier.

Figure 1(a) shows the feature distribution for both training and testing datasets (empty and filled symbols respectively) where each class is represented by a different colour. The dotted and solid lines correspond to the decision boundary of the original classifier and adapted classifier respectively. It can be seen that the original classifiers result in a completely biased classification in the shifted feature space (solid line). In contrast, the adaptive process based on the shift estimation yields an unbiased classifier (dotted line). Table 1 shows the performance of the fixed and adaptive LDA and QDA classifiers after changes in the feature distribution. The accuracy (CA) of the fixed classifier is close to chance level and its outputs are highly biased $(CI \rightarrow 1)$. In contrast, the adaptive mechanism is able to prevent this bias resulting in a high classification accuracy for both types of classifiers. It should be noticed that the reported performance corresponds to the online shift estimation.

In order to illustrate the evolution of the shift estimation, we perform another simulation with a random shift vector θ drawn from a normal distribution with mean $[70, -80]^T$ and standard deviation of 10. We performed 100 repetitions of this simulation where the feature distribution shift remain constant for each repetition. Figure 1(b) shows

Table 1: Synthetic data - average classification accuracy (CA) and confusion index (CI) over 100 repetitions.

Classifier	CA(Avg. ± std)	CI (Avg. ± std)
LDA	51.04% ± 6.14	97.73% ± 13.67
QDA	49.84% ± 8.38	95.22% ± 17.68
Adaptive LDA	84.59% ± 2.09	7.80% ± 3.88
Adaptive QDA	84.39% ± 2.11	3.73% ± 2.88

the average error in the shift estimation computed as the Euclidean distance between the actual and the estimated shift. It can be shown that the distance between the actual and estimated shifts quickly decreases and remains stable after a small number of presented samples.

3.2 HCI Gesture Dataset

We tested the proposed method on an acceleration-based gesture based HCI scenario [8, 9]. Five different hand gestures, namely a triangle, an upside-down triangle, a circle, a square, and an infinity symbol should be distinguished. Gestures were recorded using six USB acceleration sensors at different positions to the right lower arm of the subject, c.f. Fig. 2(a). For each action 50 repetitions are available. Data are manually windowed to contain only a single action with duration between five to eight seconds. We performed two set of simulations, on the first we used only the mean and variance of the y-acceleration as done by Förster and colleagues [9]. In the second configuration we used a larger set of features were for three different axes of acceleration we compute mean, standard deviation, min, max, energy in addition to magnitude of acceleration signals and correlation between each pair of three axes. Canonical Variate Analysis (CVA) is used to reduce the feature dimensionality to four (i.e. corresponding to the to number of classes minus one) [16, 17].

We created training and testing sets containing two thirds and one third of the data respectively. As in previous studies, sensor displacement was emulated by testing the classifier using data from a different sensor of the one used for training [9]. We report the classification performance of a static LDA classifier, as well as the proposed adaptive version of LDA (aLDA). The update threshold Θ was set to 1.5 based on the training dataset.

For comparison, we evaluate the *Importance Weighted LDA* (IWLDA) that also relies on the covariance shift assumption but requires a calibration dataset to estimate the distribution shift (c.f. Section 1, [11]). In the reported simulations we used the all test samples as calibration dataset, therefore corresponding to the performance of an off-line recognition system. KLIEP was applied for the importance estimation (for IWLDA we set $\lambda = 1$ and for KLIEP we set

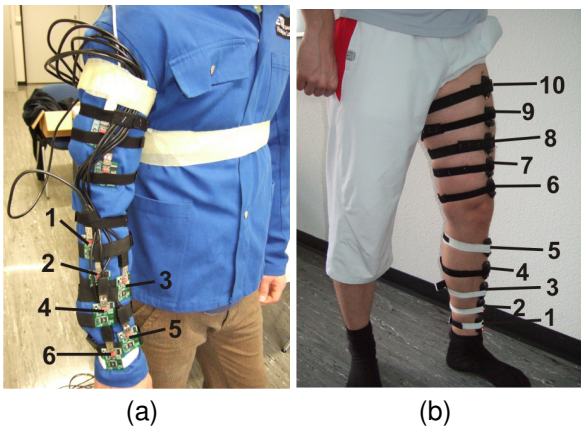

Figure 2: Sensor placement on the two experimental setups. (a) Gesture recognition scenario. (b) Fitness Scenario.

$\delta = 0.01$ and three Newton iterations). We also compare with the reported performance of the adaptive NCC classifier (aNCC) originally used on this dataset [9]. It should be noticed that the reported results for IWLDA and aNCC the feature distribution change is first estimated and then kept fixed for estimating the accuracy on the training set. In contrast, for aLDA we report the accuracy of the classification while the adaptation process takes place, therefore emulating the online performance.

Classification improvement for aLDA and IWLDA for the two set of features are shown in Fig. 3. In these plots the x-axis corresponds to the test accuracy of the fixed classifier (LDA), while the y-axis corresponds to the accuracy of the adaptive classifier. Each point corresponds to one of the tested sensor combinations. Red circles show the performance when there is no change in the sensor location (i.e. the classifier is tested on data from the same sensor it was trained). Points above the diagonal line correspond to an improvement due to the adaptation process with respect to the static classifier. It shows that for both sets of features the adaptive LDA outperforms the static classifier in most of the cases, while the accuracy remains similar when there is no change in the sensor location. In contrast IWLDA performance is less consistent, as it decreases when only y-acceleration is used as features.

Table 2 shows the average performance with respect to the sensor change. The performance of the LDA classifier decreases significantly when tested with data recorded at a different location. In contrast, aLDA consistently outperforms both the LDA and IWLDA classifiers for both sets of features. Surprisingly, IWLDA does not allow any improvement with respect to the LDA classifier when tested on another sensor location.

Since the adaptation process relies on the estimation of changes in the feature distribution, one may expect that it

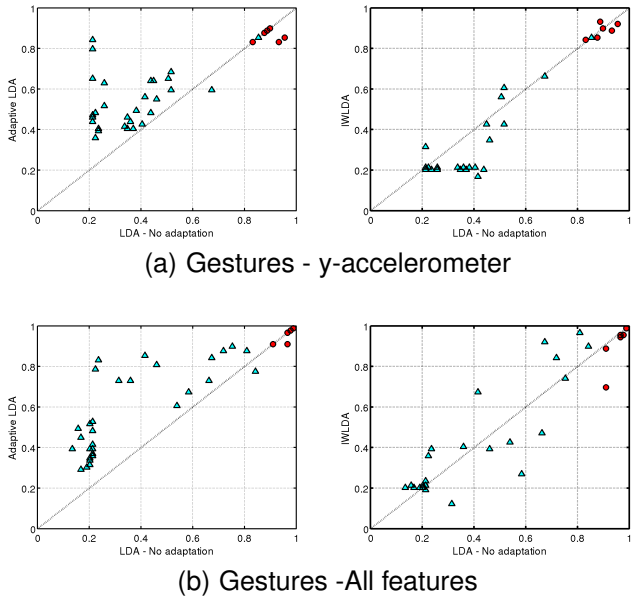

Figure 3: Classification accuracy on the HCI scenario using both sets of features (see text). Each plot shows the accuracy of the adaptive classifier vs. the accuracy of a static classifier. Red circles show the cases when the classifier is tested at the same location it was previously trained. (Left), aLDA. (Right), IWLDA.

performs better when there are small changes in the sensor location. In the case of no sensor location change ($t = s$), the aLDA adaptive mechanism yield a small decrease in performance with respect to the static classifier. In contrast, aLDA average performance is about 20% higher than LDA when tested in sensors located next to the training sensors ($|t - s| = 1$). Similarly, aLDA also improves performance in the other sensor combinations ($|t - s| > 1$). In particular, we observe that the aLDA is quite robust for the location of sensors 3 to 6 (i.e. sensors located closer to the wrist). Indeed the average performance after displacement among of these positions is equal to 75.2% and 86.9% for the two simulated set of features (c.f. Fig 4).

In comparison to the reported results for aNCC using the y-acceleration as feature, the adaptive LDA has a better performance when tested on the same sensor. For small displacements the average performance is similar but aLDA exhibit less variance across sensors. Finally, when the displacement is larger, the average performance for aLDA is lower than for aNCC. This is mainly due to a sharp decrease of performance when any of sensors 1 or 2 is tested on locations 3 to 5.

3.3 Fitness Activity Dataset

The method was also tested on a fitness scenario where five different aerobic movements of the leg were recorded

Table 2: Classification accuracy - HCI scenario

Y-acceleration

Classifier	$t = s$	$\|t - s\| = 1$	$\|t - s\| > 1$
LDA	89.7 ± 4.4	43.6 ± 21.4	32.1 ± 9.7
aLDA	86.3 ± 2.9	62.8 ± 13.2	49.4 ± 11.8
IWLDA	89.0 ± 3.6	41.46 ± 24.3	23.0 ± 6.1
aNCC	82.4 ± 2.0	63.5 ± 19.8	59.4 ± 22.5

All features

Classifier	$t = s$	$\|t - s\| = 1$	$\|t - s\| > 1$
LDA	95.3 ± 3.4	46.1 ± 26.0	30.5 ± 19.4
aLDA	94.4 ± 3.8	68.1 ± 19.7	53.0 ± 20.6
IWLDA	90.5 ± 10.7	48.0 ± 32.3	32.3 ± 20.3

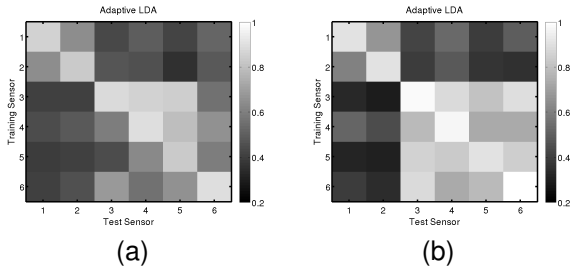

Figure 4: Classification accuracy - HCI gesture scenario. Accuracy is encoded by grey levels. Each row denotes the sensor used for training and each column represents the sensor used for testing the algorithms. (a) y-acceleration. (b) All features.

using 10 bluetooth acceleration sensors located at the subject leg [8, 9]. As depicted in Figure 2(b) five of the sensors were placed on the lower leg and the other five on the thigh. the sensors were located equidistantly and roughly with the same orientation to model only translation. During the experiment, the subject watches five times a video of an aerobic teacher and emulates the depicted movements. The video contains all movement classes equally represented. It should be noticed that in this type of applications sensor displacement due to the fast movements is likely to occur in real applications. For each sensor, the mean and variance of the acceleration magnitude based on a sliding window with two thirds of overlap is used as feature. As in the previous application, the data was divided into a training and a testing set containing two thirds and one third of the data respectively, and simulation parameters for aLDA were the same as before. We tested separately the sensors located on the different leg segment (i.e. thigh or lower leg), as preliminary results show that little adaptation can be achieved for location changes between different limb segments.

Contrasting with the previous scenario, in this case the performance of the aLDA and IWLDA classifiers do not significantly differ from the static LDA (c.f. Fig 5). A per-

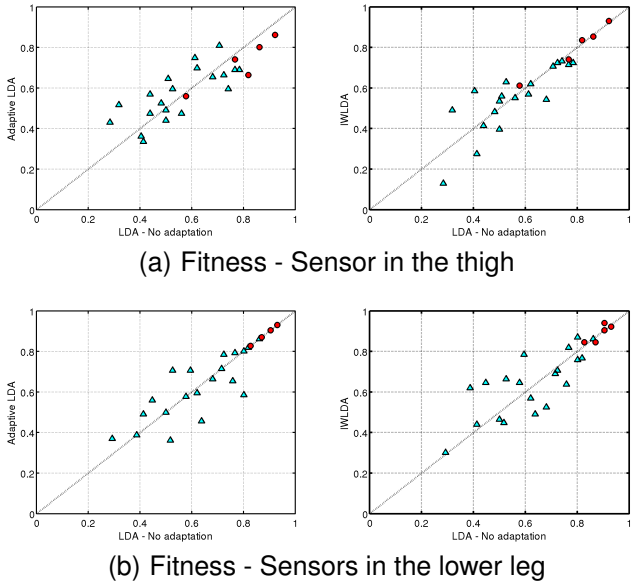

Figure 5: Classification accuracy - Fitness scenario. (Left) adaptive LDA. (Right) IWLDA

Table 3: Classification accuracy - Fitness scenario

Thigh			
Classifier	$t = s$	$\|t - s\| = 1$	$\|t - s\| > 1$
LDA	79.0 ± 13.2	62.4 ± 11.6	50.2 ± 15.1
aLDA	72.6 ± 11.8	62.0 ± 10.2	53.8 ± 14.1
IWLDA	79.5 ± 12.2	62.7 ± 9.5	48.2 ± 16.9

Lower leg			
Classifier	$t = s$	$\|t - s\| = 1$	$\|t - s\| > 1$
LDA	88.8 ± 4.0	76.6 ± 8.1	52.7 ± 12.7
aLDA	88.8 ± 4.0	75.1 ± 9.3	53.2 ± 12.5
IWLDA	89.1 ± 4.4	77.6 ± 7.8	54.2 ± 11.7
aNCC	82.8 ± 5.9	74.4 ± 9.9	49.5 ± 9.4

formance increase is only observed when there is a large change in the sensor location ($|t - s| > 1$) specially for sensors located in the thigh. Indeed, as can be seen in the Table 3 the performance decrease of the static LDA classifier when tested in other locations is not as steep as in the HCI scenario. The average performance of the static LDA when testing in the closest sensor to the training one ($|t - s| = 1$) is about 62% and 76% for sensors in the thigh and lower leg respectively. Actually, the accuracy of the static LDA is already higher than the reported accuracy of the adaptive NCC for the sensors in the lower leg. The aLDA performance for each tested combination is shown in Fig 6, a gradual degradation with respect to the sensor displacement can be observed specially for the sensors on the lower leg.

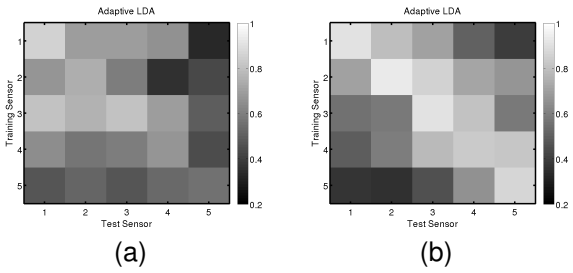

Figure 6: Classification accuracy - Fitness scenario. (a) Sensors on the thigh. (b) Sensor on the lower leg.

4 Discussion

Robustness to sensor displacement is an important aspect for practical applications using wearable sensors. At operation time, the exact placement of these sensors cannot be ensured as they may slip or can be placed at slightly different positions from one day to the next. In this work we proposed an adaptive mechanism that is based on the unsupervised, online tracking of the feature distribution. The proposed method extends probabilistic Gaussian classifiers assuming that changes in the sensor location mainly results in a shift of the overall feature distribution, without affecting the conditional distribution of the classifier outputs (i.e. covariance shift). Given this assumption, unsupervised adaptation is achieved by estimating the features shift by means of an online version of expectation maximisation using the Levenberg-Marquardt algorithm.

Simulations using synthetic data shows that this method is able to quickly estimate the features shift, and adapt the classifier if the underlying assumption holds (c.f. Section 3.1). Although such an assumption is unlikely to fully hold in real applications, experimental results on two applications using body-worn accelerometers show that this method is able to compensate for strong performance decrease, as in the case of the gesture recognition application, without compromising the performance when the original classifier performs well, e.g. fitness scenario. We use an experimental setup using sensors located at different positions of the upper and lower limbs, allowing to emulate sensor displacement by testing the classifier in a sensor located at a different position than the one used for training.

Moreover, we also compare the proposed method with another technique based on the covariance shift assumption (i.e. IWLDA) that uses calibration data to estimate the feature change based on the ratio between the distribution before and after the shift takes place. We also compared with the performance of an adaptive version of the NCC classifier (aNCC) previously reported on the same datasets [9]. In their study, Förster and colleagues use calibration data to update the classifier and then keep it fixed for testing proce-

dures. Moreover, the aNCC requires a free parameter corresponding to the learning rate. In contrast, we apply our method without a re-calibration phase and report the testing performance while the adaptation process is taking place, thus providing an estimation of the online performance of the system.

In the gesture recognition scenario the performance of a LDA classifier quickly drops after a change in the sensor location, while the performance decrease for the adaptive LDA is not as strong. In particular, aLDA performance remain particularly high (above 75% if using only features from y-accelerometer) for sensor located close to the wrist . Compared to the aNCC, the aLDA classifier performs similarly for small sensor displacements while having less variability across sensor combinations. However, the aNCC relies on a calibration process and the classifier remains fixed during the testing period. In contrast, the IWLDA approach fails to adapt to the changes in the feature distribution having accuracies closer to the fixed classifier. For the fitness scenario, the adaptive LDA does not perform significantly better than the static classifier. This may be due to the fact that the LDA classifier already seems robust to small sensor displacements in this application–indeed, LDA outperforms the adaptive NCC–thus leaving less opportunity for adaptation. A similar performance pattern was observed for the IWLDA, showing that our approach converges to the same estimation than the calibration process of this method.

Several methods have been previously proposed to detect changes in a particular sensor [18, 19]. Correspondingly the proposed method, besides the adaptation process, the estimated shift provides a measure of how much the current feature distribution resembles the one used for training, and can be used as an evaluation of the sensor reliability. Fig 7 shows how the estimated shift (average and mean over the testing dataset) correlates with the change in performance with respect to the original sensor location. In general larger estimated shifts corresponds to a decrease in accuracy although, in a few cases a performance decrease is observed even though the estimated shift is small suggesting that in these cases the covariate shift assumption is not satisfied. This was mainly observed when sensors in the lower leg were tested on locations closer to the knee joint.

The current technique can be extended to take into account more realistic assumptions on the feature distribution change (e.g. allowing for scaling and rotations). Nevertheless, this may imply iterative processes relying on a larger amount of data, thus compromising its application on wearable, runtime applications. Reported results show that the simple covariate shift assumption already provides a simple mechanism to increase robustness to sensor displacement while providing a way to assess the reliability of the sensor during its online use. Furthermore, this is achieved in an unsupervised manner without requiring a calibration phase and using only one free parameter (Θ) that can be directly extracted from the available training data. Moreover, although the method has here been tested using accelerometers, it can also be applied to any type of sensors (e.g. textile sensors in smart clothing). Future work will be devoted to test its performance on other setups, as well as to assess whether the same approach can be used to increase robustness to other types of changes such as sensor rotation or changes in the actual motion patterns, e.g. as a result of fatigue or towards adaptation to new users.

Acknowledgment

We would like to thank K. Förster and D. Roggen from the Wearable Computing Lab at ETH Zurich for providing the experimental data and insighful discussions. This work was supported by the EU-FET project ICT-225938 *Opportunity: Activity and Context Recognition with Opportunistic Sensor Configuration*. This paper only reflects the authors' views and funding agencies are not liable for any use that may be made of the information contained herein.

References

[1] H. Kang, C. W. Lee, and K. Jung, "Recognition-based gesture spotting in video games," *Pattern Recognition Letters*, vol. 25, no. 15, pp. 1701–1714, 2004.

[2] T. Stiefmeier, D. Roggen, G. Tröster, G. Ogris, and P. Lukowicz, "Wearable activity tracking in car manufacturing," *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 42 –50, apr. 2008.

[3] M. Tentori and J. Favela, "Activity-aware computing for healthcare," *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 51 –57, 2008.

[4] K. Van Laerhoven and O. Cakmakci, "What shall we teach our pants?" *IEEE Int Symposium on Wearable Computers, 2000*, pp. 77 –83, 2000.

[5] N. Ravi, N. D, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," *Proc 17th conf Innovative applications of artificial intelligence*, vol. 3, 2005.

[6] K. Kunze and P. Lukowicz, "Dealing with sensor displacement in motion-based onbody activity recognition systems," in *UbiComp '08: Proc int conf on Ubiquitous computing*. New York, NY, USA: ACM, 2008, pp. 20–29.

[7] U. Steinhoff and B. Schiele, "Dead reckoning from the pocket–an experimental study," in *IEEE Int Conf*

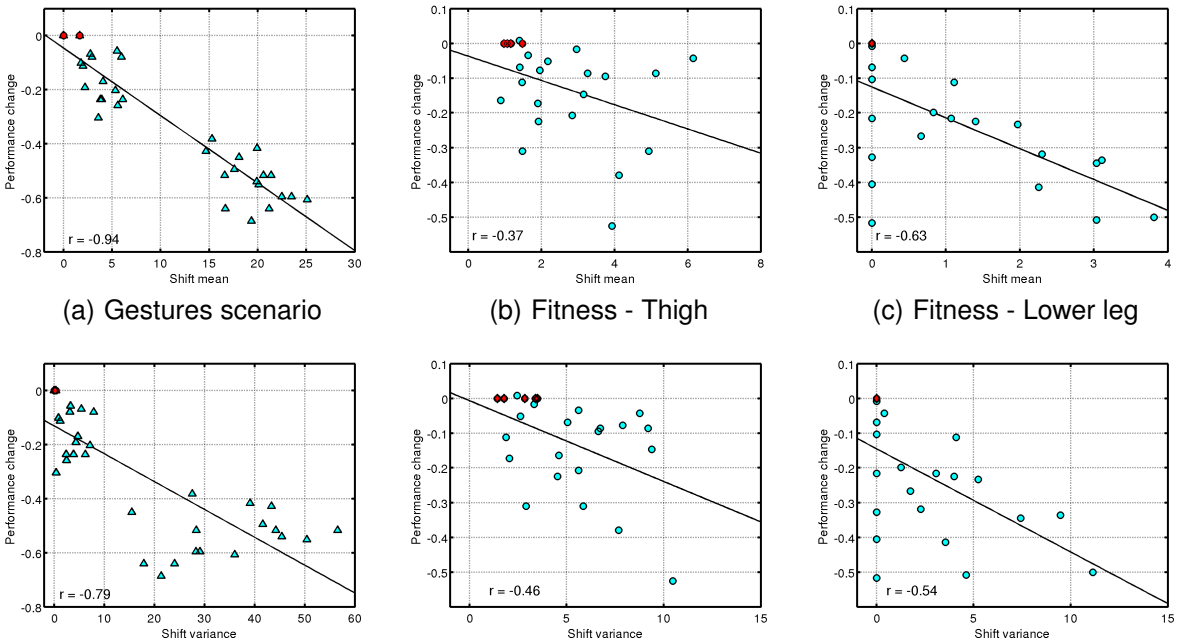

Figure 7: Change in performance (accuracy on testing location - accuracy on training location) with respect to the mean (*Top*) and variance (*Bottom*) of the estimated shift. Mean and variance are normalized with respect to the estimated values on the training set. *Left:* Gesture recognition scenario. *Middle:* Fitness scenario - sensors in the thigh. *Right:* Fitness scenario - sensors in the lower leg.

Pervasive Computing and Communications (PerCom 2010), Mannheim, Germany, 2010.

[8] K. Förster, P. Brem, D. Roggen, and G. Tröster, "Evolving discriminative features robust to sensor displacement for activity recognition in body area sensor networks," *Int Conf Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009*, pp. 43 –48, dec. 2009.

[9] K. Förster, D. Roggen, and G. Tröster, "Unsupervised classifier self-calibration through repeated context occurences: Is there robustness against sensor displacement to gain?" in *IEEE Int Symposium Wearable Computers*, 2009.

[10] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, 2000.

[11] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. von Bnau, and M. Kawanabe, "Direct importance estimation for covariate shift adaptation," *Annals of the Institute of Statistical Mathematics*, vol. 60, no. 4, pp. 699–746, 2008-12-01.

[12] Y. Li, H. Kambara, Y. Koike, and M. Sugiyama, "Application of covariate shift adaptation techniques in Brain Computer Interface," *IEEE Trans Biomedical Engineering*, vol. 57, no. 6, pp. 1318–1324, 2010.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*, J. K. M. Jordan and B. Schoelkopf, Eds. Springer, 2007.

[14] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. pp. 431–441, 1963.

[15] A. Satti, C. Guan, D. Coyle, and G. Prasad, "A covariate shift minimization method to alleviate non-stationarity effects for an adaptive Brain Computer Interface," in *Int Conf on Pattern Recognition*, 2010.

[16] W. J. Krzanowski, *Principles of multivariate analysis*. Oxford: Oxford University Press, 1998.

[17] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*, 2nd ed. Wiley-Interscience, November 2000.

[18] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.

[19] H. Sagha, J. d. R. Millán, and R. Chavarriaga, "Detecting anomalies to improve classification performance in an opportunistic sensor network," in *IEEE Workshop on Sensor Networks and Systems for Pervasive Computing, PerSens 2011*, Seattle, March 2011.