Automatic Transfer of Activity Recognition Capabilities between Body-Worn Motion Sensors: Training Newcomers to Recognize Locomotion

Alberto Calatroni, Daniel Roggen, Gerhard Tröster
Wearable Computing Laboratory
ETH Zürich, Switzerland
Email: alberto.calatroni@ife.ee.ethz.ch

***Abstract*—Wearable human activity recognition systems rely on one or more body-worn motion sensors and usually machine learning approaches are used to interpret the data. These approaches require a time-consuming training process involving explicit user intervention. In a pervasive computing scenario, users can buy new "general purpose", untrained, sensor-enabled devices to upgrade or replace their on-body sensing infrastructure. Yet they want to keep the recognition capability of the previous system. In this paper we address the problem of transferring, without user intervention, the activity recognition capability of an existing sensor node to the newly deployed and untrained node. We present a method which is independent of sensor modality and location, assuming the co-presence of trained sensor nodes and untrained ones, to operate an autonomous transfer of the activity recognition capabilities. Labels of recognized activities are transferred from trained to untrained nodes, which incrementally associate their sensor signals to the received labels. We compare this method with a naive approach involving a direct activity model transfer and we use as a baseline the conventional manual training. We assess the performance on a realistic dataset with eight body-worn sensors for the recognition of posture and modes of locomotion. With the novel method, a newly deployed sensor reaches recognition accuracy in average within $9.3\,\%$ of the baseline and $17.3\,\%$ higher than the naive approach, with a bandwidth requirement of $12\,\mathrm{bytes/s}$.**

I. Introduction

Human activity recognition is useful for a different range of applications, like healthcare, human-computer interaction (HCI) or assistance of industry workers. Activity recognition relies on sensor nodes embedded in the infrastructure and worn on the body, which interpret sensed signals in terms of human activities through machine learning techniques running on the nodes [1], [2]. Networked body-worn sensor nodes are appealing, as they can operate wherever the user is. The presence of body-worn motion sensors in daily life is now increasing, as most mobile phones contain accelerometers, gyroscopes and magnetic field sensors. Other accessories may be instrumented, such as watches or sports shoes. Recent advances in the integration of electronics into textiles [3] pave the way to smart garments. In our work, we assume upcoming ambient intelligence environments with a rich availability of interconnected sensors provided by a wide range of suppliers, including commercial gagdgets, accessories and clothing that the user may decide to buy.

Imagine a pervasive computing scenario where users can buy new "general purpose", untrained, sensor-enabled devices, such as a watch or smart shirt, to upgrade or replace their pre-existing on-body sensing infrastructure. Yet they want to keep the recognition capability of the previous system. One of the open problems which then arise is: assuming a machine learning approach, how can we make use of the new sensor nodes that a user might wear for activity recognition, without having the user perform an extensive training phase? In case of a commercial gadget, training could be provided by the manufacturer, but this would only cover the activity classes foreseen by the manufacturer. In this paper we propose two ways to exploit the coexistence of systems on the user's body to transfer capabilities to recognize activities from one system to another. In a concrete example, a person might wear smart shoes which are able to recognize modes of locomotion and produce labels thereof. If the user subsequently buys a watch and wears both systems for a day, the capabilities of the shoe can be transferred to the watch. In this paper we investigate two ways to perform this transfer. In both approaches, we assume that the systems coexist for enough time (for example, a typical day of the user's routine) and that the two systems are networked and able to exchange data.

We contribute with two approaches to transfer the ability to recognize modes of locomotion from a body-worn sensor node to a newly deployed one. The first approach consists in a naive transfer of the classifier models from one trained node to the new one, while the second approach involves sharing of the recognized labels to perform a "system-supervised learning" on the newcomer. By system-supervised learning, we mean that the system (existing sensor nodes) provide the labels for the training on the newcomer. The second approach is more general and allows transfer across sensing modalities of different kinds. The two approaches are particular cases of Transfer Learning. We evaluate the performance and limits of the two approaches in a task of recognition of posture and modes of locomotion from body worn accelerometers. We present the results using Nearest Class Center (NCC), k-Nearest Neighbors and SVM classifiers.

II. State of the Art

The problem of training a previously unknown and untrained, newly deployed, body-worn sensor can be tackled in different ways. One approach is to first try to locate the new sensor node and then upload a suitable classifier model to the node. Kunze showed that it is possible to determine the orientation of an accelerometer in a pocket [4], the position of an acceleration sensor on the body [5] or even, to some extent, in the environment [6]. This is done by training a machine learning algorithm with signals coming from different body positions. The limitation of this approach is that the set of possible locations where a sensor node can be placed is limited by the training set used. Furthermore, this approach requires a connection between the new device and a repository which has to be rich enough to contain many different sets of feature and classifier combinations.

A complementary approach is to perform an online calibration, starting from a classifier model that was pretrained in certain positions. This has been shown with an NCC classifier in [7], but it works only if the starting model and the target model are similar enough.

Two works show that position independent features [8] and classifiers [9] can be devised. The second work focused on the recognition of locomotion activities and in both cases an offline training phase is required. One limitation of these two approaches is that the system designer needs to know the possible positions where sensors could be placed. Furthermore, the approaches are suitable only if the newly deployed sensors and the ones used for training share the sensing modality.

A different set of approaches involve the identification of a label source that can be used to train the new sensor while deployed. The user herself can provide the labels: this has been proposed in [1], where the user is asked to provide annotations. The limitation is the need for a substantial user participation, which limits the usability.

Transferring cababilities from one sensor node to another is an instance of *transfer learning*. Transfer learning is the process of using the "knowledge" acquired in one domain to gain knowledge in another domain [10]. Various approaches exist to enable this transfer, also within machine learning. For example, TrAdaBoost [11], derived from AdaBoost [12], consists in using examples from the old domain to build up the "difficult" examples for the new domain. The assumptions are that the two expert systems are operating in the same domain (or very similar) and with the same types of features. This is unfortunately not always realistic, particularly in opportunistic, heterogeneous systems, where sensor nodes can be measuring very different kinds of physical quantities.

Van Kasteren [13] showed that a transfer learning approach can exploit the knowledge (classifier models) related to activity recognition in one smart home to train similar activity recognition systems in other homes, thus avoiding many costly training data collections. Again, the need for a common feature space induced the authors to create "meta-features", which are the common ground on which the transfer of knowledge occurs, and the method was applied only to binary sensors.

We have already shown in previous work [14] that a transfer learning is possible at the classifier level, with presegmented instances, when different systems interact and share labels. The previous results were though limited to a very constrained setup and did not consider continuous recognition.

In this work we go beyond the state of the art, presenting an approach which is complementary to what has been proposed in [5] and [4] and which extends [14] to more realistic settings. Our approach circumvents the need for common feature spaces and is applicable also to sensor nodes using different sensing modalities.

III. System Architecture and Transfer Methods

In order to overcome the limitations of the state of the art, we present two methods to transfer activity recognition capabilities to a newly deployed sensor. We will henceforth call Teacher and Learner the pretrained and new sensor respectively. These methods can be directly implemented in an ecology of networked sensor nodes which run pattern recognition algorithms, that we call Context Cells. The target platform deployed on the nodes is the Jennic JN5139 microcontroller and the used radio link is ZigBee. The choice of ZigBee is due to many advantages that it brings to the application at hand, the most prominent of which is the ease with which a new sensor can self-advertise and join a preexisting network. For more details on Context Cells, refer to [14]. We use these nodes as a reference for the implementation of our algorithms.

A. Sensor Node Operation

In the following, we assume the sensor nodes to be implementing a typical activity recognition chain, which consists of:

- a sensor (S), in our case a $3D$ accelerometer;
- a feature extraction block (FX), which segments sensor data into windows and calculates a feature vector of length F for each window;
- a machine learning block (ML), which compares the feature vectors with stored classifier models ($M_1 \dots M_C$, where C is the number of activity classes) and outputs a decision (label).

B. Capability transfer by naive classifier model transfer

The first approach for transferring the capabilities from Teacher to Learner is to copy the classifier model and send it over a wireless link from one node to the other. This can be done for any classifier used. In case of an NCC classifier, it translates in transferring one centroid corresponding to each of the learnt activity classes; for a kNN, all the instances of the different activity classes must be transferred and for an SVM we need to send the support vectors and the coefficients for the decision functions. For this transfer, only a short amount of time is needed, when the two sensor systems are worn together on the user's body. After the transfer, the Learner can start classifying the datastream and provide class outputs. This approach is a naive form of transfer that is suited if the position and modality of the new sensor node roughly correspond to the ones of the existing one. A graphical representation of how the method works in shown in the upper part of Figure 1.

C. Capability transfer by system-supervised learning

In order to decouple the feature spaces of the two involved systems, we propose to operate a *system-supervised learning* on the new sensor. By this we mean a supervised learning, where the supervision (i.e. the labels) are provided by the Teacher sensor node. The two nodes have to be synchronized and must be present on the user's body for at least one normal day, so that a sufficient number of instances of the different activities is performed. Every time that the Teacher recognizes an activity class, the label is broadcast via the wireless link, so that the Learner(s) can associate the correct part of their signal buffers to the right label and it can store the instance with the label. The Learner can use the stored instances to perform a normal batch training at the end of the day, or perform the training incrementally as the labels are received if the type of classifier used allows it. A graphical representation of how the method works in shown in the lower part of Figure 1.

IV. Dataset

In order to validate the proposed approach, we conducted simulations using a dataset [15] that was recorded in a naturalistic environment. The activity recognition environment and scenario were designed to generate many activity primitives, yet in a realistic manner, and the dataset is very sensor-rich. Subjects operated in a room (see Figure 2) simulating a studio flat with a lazy chair, a kitchen, doors giving access to the outside, a coffee machine, a table and a chair.

The subjects performed 5 times a "free run" (10-20 minutes), where a set of activities were gathered, that normally occur in a real-life situation, when a person is preparing breakfast or lunch. The run consists of temporally unfolding situations, and in each situation (e.g. *preparing sandwich*), composite activities (e.g. *cutting bread*) occur, as well as atomic activities (e.g. *reach* for bread, *move* to bread cutter, *operate* bread cutter).

In the simulations in the present work we extracted streams from eight accelerometers (see Figure 3) coming from five "free runs" of one of the subjects and we focus on the postures and locomotion. Specifically, the dataset contains instances of standing, walking, sitting and lying, giving us a four class problem, for which a random guess would yield an accuracy of 25%.

These these activities occur in an indoor environment in a natural way and were not scripted. This means that the dataset does not contain clean, long, uninterrupted activity instances, e.g. of "walking", but rather short and noisy ones. Those usually involve walking between close-by locations, such as when displacing objects from a shelf to a table. Furthermore, there are instances which are borderline between "standing" and "walking" (e.g. the user is making one or two steps while standing), which are generally labeled as "standing". Figure 4 shows an example of an instance of each activity collected with the eight different sensors.

V. Simulations and Results

A. Simulation procedure

Each of the eight body-worn sensors selected from the dataset is trained from ground truth labels to build eight possible Teacher systems. The signal instances are obtained by a sliding window approach (width 1 second), yielding N_i instances for each class i and a total of $N = 4357$ instances for the five runs. The window width is a trafeoff between information content and latency. In fact, 1s is enough to contain a representative part of the signal for the examined postures and modes of locomotion, while it corresponds to a delay for the classification which is acceptable for most applications. For each instance, a vector containing four features is extracted ($F = 4$). These are the mean values of the x, y and z accelerometer axes and the standard deviation of the magnitude. The choice of these generic features is aligned with our vision where a sensor system is worn by the user on a non predefined part of the body, hence we opted for general purpose features.

The two methods proposed in section III are executed on real data obtained from the described dataset and evaluated in terms of the accuracy which is reached on the Learner systems compared to the baseline. The accuracy is defined as the percentage of the test instances in which the output of the sensor system agrees with the manually labelled ground truth. Another evaluation is done by means of a confusion matrix. Each entry $(\mathbf{i}, \mathbf{j})$ of such a matrix represents which fraction of the instances belonging to class $\mathbf{i}$ are classified by the system as belonging to class $\mathbf{j}$. This matrix shows which classes are mostly confused by a classifier. The closer it is to a diagonal matrix, the better.

The capabilities transfer is performed for all possible Teacher-Learner combinations and with the following classifiers:

1) Nearest-Centroid Classifier (NCC)
2) k-Nearest Neighbors (kNN), with $k = 11$
3) Support Vector Machines (SVM)

The classifier parameters were chosen empirically. The classifications were performed with an ad-hoc toolbox, the ones with SVM relied on LIBSVM [16].

For each classifier, the five runs are used as follows (each run contains on average 871 instances):

- two runs used to train the Teacher systems from ground truth;
- two runs simulating the co-presence of the Teacher and Learner systems on the body: here we transfer the capabilities from one system to the other and we assess the accuracy of the Teacher against ground truth;
- the last run is used to assess the Learner node accuracy against ground truth.

The process is subject to a five-folds cross validation and for each fold the accuracy is computed. We report the average and standard deviation of the accuracy across the five folds.

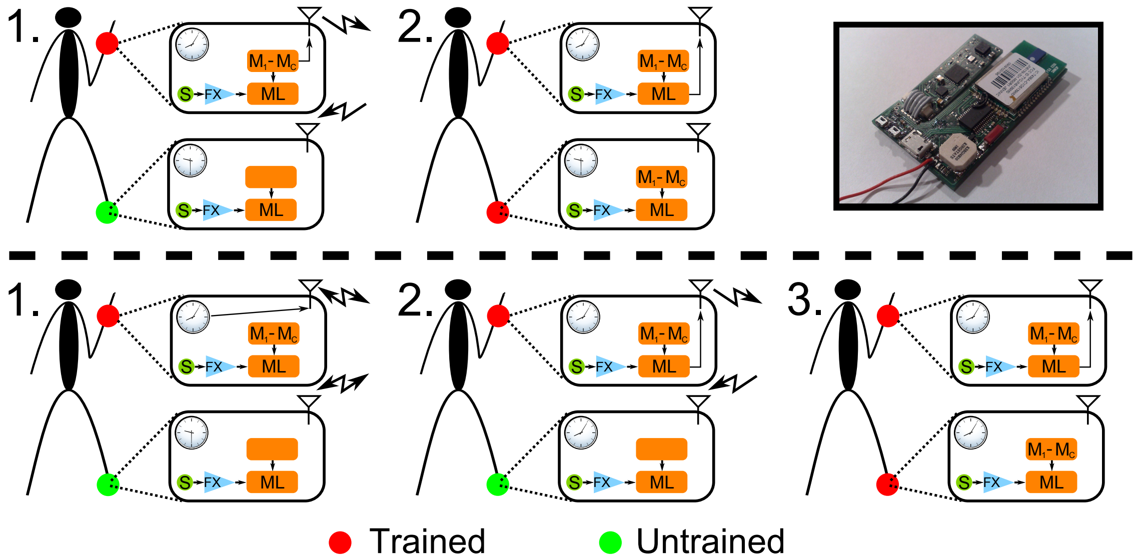

Fig. 1. Schematic description of the naive classifier model transfer (top) and system-supervised learning approach (bottom) on an example setup. In the naive method, in phase 1 the Learner joins the network and the Teacher sends to it the stored classifier models $M_1 \ldots M_C$. In phase 2, the Learner is ready to recognize the user activities. In the system-supervised learning method, in phase 1 the newly deployed sensor on the (Shoe) interchanges timestamps with the Teacher node (RUA) so that their clocks are synchronized. In phase 2, the Teacher node sends labels every time that an activity has been detected and the Learner stores the labels and the corresponding measured signals. Finally, the Learner can perform a batch training using the stored data and in phase 3 it is now ready to recognize the user activities.

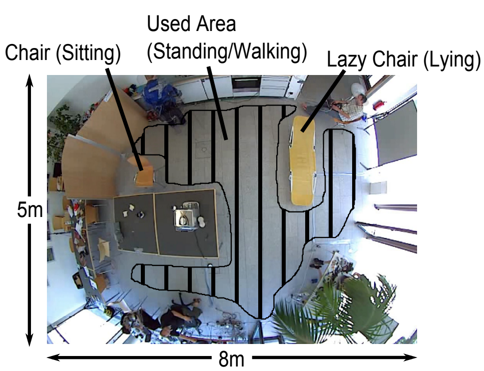

Fig. 2. Top view of the room where the activities were recorded. The postures that we used in the simulations are reported in brackets along with the locations where they were taking place.

Fig. 3. Set of wearable sensor nodes, comprising accelerometers, used for the simulations.

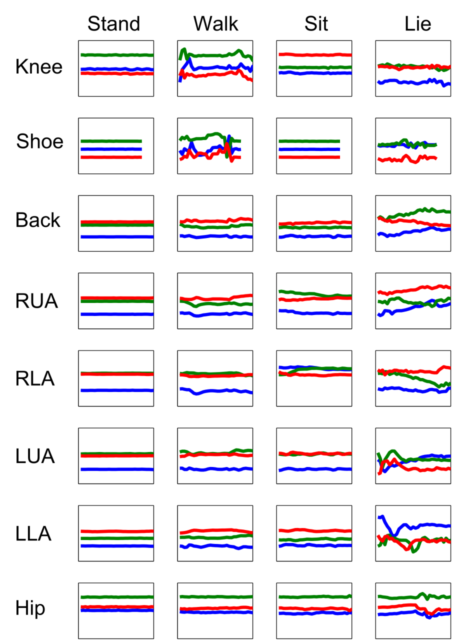

Fig. 4. Example of the sensor signals. Each plot shows the signals from the three accelerometer axes.

B. Results

1) Baseline: Table I shows the baseline accuracies in percentage (standard deviation in brackets) for the eight sensor nodes, obtained by training from ground truth (which can be considered an ideal Teacher). The obtained values represent how well these systems perform when used as Teachers and also represent the performance upper bound when the sensor

	Knee	Shoe	Back	RUA	RLA	LUA	LLA	Hip
NCC	92(0)	76(9)	72(6)	78(2)	73(1)	83(3)	75(2)	73(4)
11-NN	93(1)	83(3)	78(5)	83(1)	79(1)	86(2)	83(1)	73(4)
SVM	93(0)	77(7)	73(5)	81(2)	76(1)	86(2)	82(1)	73(4)

TABLE I
TEACHER ACCURACY, OBTAINED BY TRAINING WITH GROUND TRUTH LABELS, FOR NCC, 11-NN AND SVM CLASSIFIERS (STANDARD DEVIATION ACROSS FOLDS IN BRACKETS). THE SENSOR POSITIONS ARE THOSE REPORTED IN FIGURE 3. FOR COMPARISON, RANDOM GUESSING WOULD YIELD 25% ACCURACY.

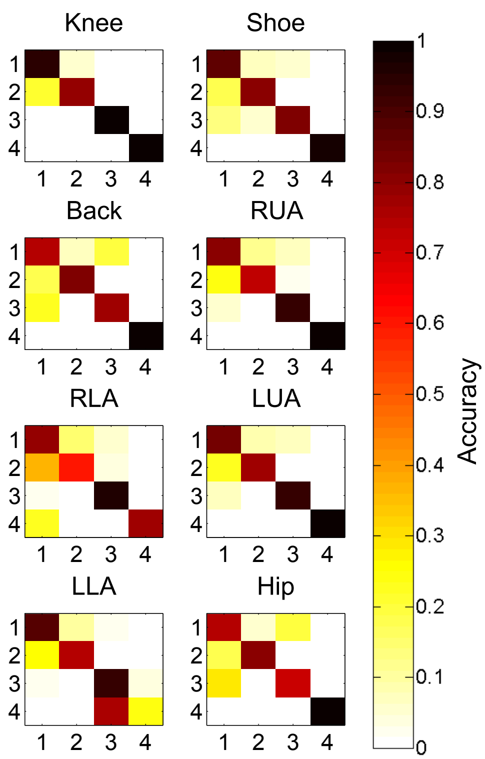

Fig. 5. Confusion matrices for the eight sensor nodes using 11-NN classifier. Legend of the activities: 1 = Standing; 2 = Walking; 3 = Sitting; 4 = Lying

nodes become Learners (so-called "baseline" in the following). The sensor node which is best suited to the recognition task is the accelerometer mounted on the upper part of the knee, reaching 93 % accuracy. The nodes on the upper parts of the arms follow with 86 % and 83 %. The normalized confusion matrices for the different sensor nodes are shown in Figure 5 for the 11-NN classifier and are very similar for the other two classifiers.

2) Naive classifier model transfer: The accuracies achieved by the three classifiers with the naive classifier model transfer are summarized in table II, for each Teacher-Learner combination. The values obtained by the Learners are in average (considering all classifiers) 26.6 % lower than the baseline (I). NCC performs slightly better, achieving on average an accuracy 23.7 % lower than baseline, compared to 29.4 % and 26.8 % for 11-NN and SVM respectively. In the specific case of the model transfer from the RLA to the LLA sensor node, we reach an accuracy 5 % lower than the baseline with NCC and SVM.

In terms of network load, the naive model transfer requires

	From\To	Knee	Shoe	Back	RUA	RLA	LUA	LLA	Hip
NCC		-	61(11)	44(7)	44(5)	52(5)	67(6)	**73(3)**	56(4)
11-NN	**Knee**	-	58(7)	43(9)	44(5)	49(6)	65(9)	68(5)	58(5)
SVM		-	60(10)	43(9)	44(6)	50(4)	63(8)	**72(3)**	57(5)
NCC		69(10)	-	43(6)	**70(2)**	54(3)	40(3)	47(8)	67(8)
11-NN	**Shoe**	51(6)	-	57(6)	61(3)	43(6)	43(6)	38(3)	60(8)
SVM		43(16)	-	51(4)	69(4)	49(3)	40(2)	46(6)	**65(8)**
NCC		31(5)	34(10)	-	36(1)	23(1)	51(7)	24(2)	41(8)
11-NN	**Back**	38(5)	53(12)	-	46(3)	35(2)	58(8)	38(5)	53(6)
SVM		44(7)	58(11)	-	44(3)	31(3)	56(8)	36(8)	49(5)
NCC		29(7)	**68**(12)	51(4)	-	58(1)	40(3)	44(3)	70(5)
11-NN	**RUA**	14(9)	56(12)	57(4)	-	54(4)	31(6)	33(2)	60(9)
SVM		29(14)	60(15)	55(3)	-	44(2)	30(5)	29(2)	54(3)
NCC		58(7)	64(9)	47(6)	63(2)	-	52(5)	**80(2)**	69(8)
11-NN	**RLA**	41(6)	53(7)	49(5)	64(3)	-	50(8)	**74(3)**	54(9)
SVM		53(6)	60(11)	50(7)	67(4)	-	51(6)	**77(3)**	**63(10)**
NCC		**85(2)**	37(12)	56(6)	23(2)	40(2)	-	**66(4)**	51(6)
11-NN	**LUA**	**84(2)**	40(10)	53(8)	28(3)	42(3)	-	69(3)	58(5)
SVM		**84(2)**	40(8)	54(9)	27(3)	42(3)	-	68(4)	51(5)
NCC		80(14)	60(2)	46(6)	50(4)	**67(3)**	62(5)	-	66(6)
11-NN	**LLA**	70(11)	58(8)	48(6)	60(4)	68(2)	61(7)	-	**64(11)**
SVM		57(3)	60(6)	50(7)	63(4)	**71(4)**	59(7)	-	62(9)
NCC		45(14)	**66(10)**	46(6)	*68(2)*	**67(1)**	49(5)	**78(3)**	-
11-NN	**Hip**	52(8)	65(8)	57(9)	66(6)	58(8)	54(5)	55(8)	-
SVM		53(17)	**67(9)**	54(10)	70(6)	57(9)	48(5)	57(11)	-

TABLE II
ACCURACY TABLE FOR THE NAIVE APPROACH. EACH ROW REPRESENTS A TEACHER AND EACH COLUMN A LEARNER. EACH ELEMENT CONTAINS THE ACCURACY OBTAINED ON THE LEARNER WHEN ITS CLASSIFIER MODEL IS OBTAINED BY STREAMING THE TEACHER'S CLASSIFIER MODEL AND USING IT WITHOUT ANY MODIFICATION ON THE LEARNER. THE ACCURACIES THAT ARE WITHIN 10 % FROM THE CORRESPONDING BASELINE ARE IN BOLD. RANDOM GUESSING WOULD YIELD 25% ACCURACY.

80 bytes for NCC (4 classes, each represented by a 4-elements feature vector of 32 bit floating point type and a list of the 4 class labels, each encoded as a 32 bit integer). kNN requires the transfer of one feature vector for each instance, so we need to transfer an amount of data of the order of the number of instances (in our simulations, about 30 kilobytes). For the SVM, the amount of data depends on the number of stored support vectors, which in turn varies according to the specific case. In our simulations, the number of support vectors was on average 93 % of the number of instances, leading to a data amount similar to kNN. In all cases, though, the transfer happens only once, when the new sensor node (Learner) joins the network.

3) System-supervised learning: Table III reports the accuracies obtained with the system-supervised learning approach, for each Teacher-Learner combination and the three classifiers. The Learners reach on average an accuracy which is 9.3 % lower than the baseline. Again, NCC obtains the best results, achieving on average an accuracy 5.1 % below the baseline, whereas 11-NN is 9.7 % below and SVM only performs 13.1 % lower than baseline.

The transmission capacity which is needed in this case does not depend on the classifier. The sensor nodes need to stream one label and a start and stop time for each instance. The streaming in this case has to be realtime. In our simulations we have one instance every 1s and by encoding the label and time stamps with integers, this amounts to 12 bytes/s.

VI. DISCUSSION

A. Naive classifier model transfer

The naive approach transfers one classifier model from a Teacher to a Learner. Classifier models are tightly related to the feature spaces where they operate. This means that if the Learner feature space does not match the one of the Teacher,

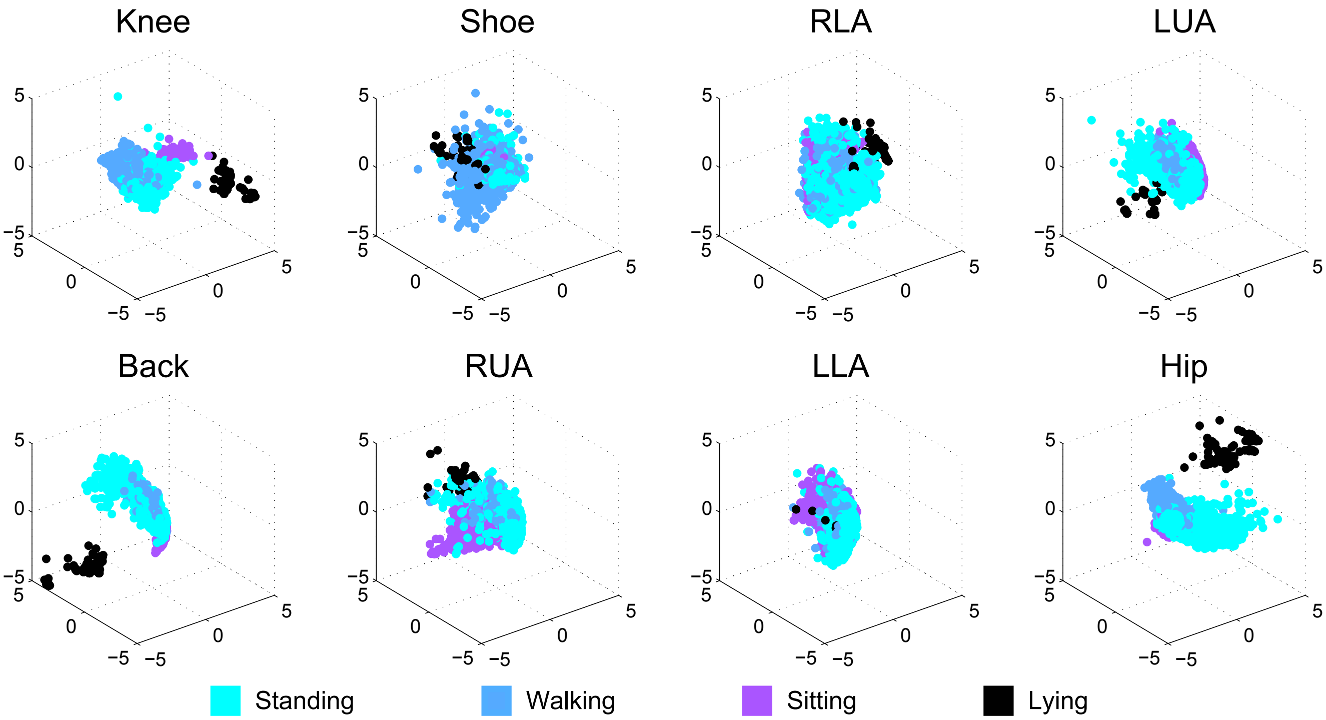

Fig. 6. Feature spaces for the eight sensor nodes, projected into 3-D space through a Principal Components Analysis. The different locations of the instances in the feature spaces explain the sensitivity of the naive transfer to the chosen Teacher and Learner.

	From\To	Knee	Shoe	Back	RUA	RLA	LUA	LLA	Hip
NCC		-	58(10)	**67(6)**	**75(1)**	**70(3)**	**76(8)**	**70(2)**	**66(5)**
11-NN	**Knee**	-	67(8)	61(2)	72(5)	**69(4)**	71(5)	70(6)	59(4)
SVM		-	60(13)	55(8)	68(4)	**68(5)**	69(7)	71(3)	55(9)
NCC		**84(7)**	-	**71(8)**	**76(3)**	**72(2)**	**78(8)**	**68(12)**	**70(4)**
11-NN	**Shoe**	74(8)	-	**73(7)**	**78(7)**	**76(4)**	**81(7)**	**79(3)**	61(2)
SVM		62(15)	-	**66(5)**	65(8)	60(8)	66(5)	66(13)	52(7)
NCC		**82(3)**	**68(10)**	-	**79(2)**	**70(3)**	71(6)	61(4)	**71(6)**
11-NN	**Back**	70(10)	**78(2)**	-	**75(3)**	**73(2)**	**79(5)**	**76(2)**	**63(2)**
SVM		57(13)	56(7)	-	65(3)	60(4)	70(8)	66(12)	52(6)
NCC		**88(1)**	**78(12)**	**72(8)**	-	**73(1)**	**82(4)**	**73(3)**	**70(6)**
11-NN	**RUA**	72(6)	**81(4)**	**76(6)**	-	**78(0)**	**85(3)**	**82(1)**	**63(4)**
SVM		79(3)	**72(11)**	**72(6)**	-	**75(2)**	**84(2)**	**80(2)**	58(7)
NCC		**88(2)**	**78(12)**	**71(8)**	**77(2)**	-	**80(5)**	**71(4)**	**67(8)**
11-NN	**RLA**	72(5)	**76(3)**	**72(5)**	**80(1)**	-	**82(2)**	**79(2)**	62(4)
SVM		77(5)	**68(13)**	**70(4)**	**79(3)**	-	**80(2)**	**77(4)**	56(10)
NCC		**88(2)**	**69(11)**	**72(8)**	**78(1)**	**72(1)**	-	**73(2)**	**71(6)**
11-NN	**LUA**	73(6)	**80(3)**	**78(6)**	**83(1)**	**79(2)**	-	**82(2)**	**64(1)**
SVM		79(3)	**72(10)**	**75(7)**	**80(2)**	**75(2)**	-	**81(2)**	54(7)
NCC		**83(5)**	**67(10)**	**71(11)**	**73(2)**	**67(5)**	72(4)	-	61(11)
11-NN	**LLA**	72(7)	**76(4)**	**73(5)**	**79(4)**	**76(3)**	**80(3)**	-	60(2)
SVM		74(3)	66(12)	**63(9)**	**74(4)**	**70(4)**	**80(2)**	-	50(9)
NCC		**83(3)**	58(11)	**70(7)**	**78(2)**	**68(5)**	71(5)	53(3)	-
11-NN	**Hip**	66(4)	64(8)	60(6)	62(2)	62(3)	65(8)	65(6)	-
SVM		74(14)	56(16)	57(7)	60(9)	56(11)	63(9)	59(14)	-

TABLE III

ACCURACY TABLE FOR THE SYSTEM-SUPERVISED LEARNING APPROACH. EACH ROW REPRESENTS A TEACHER AND EACH COLUMN A LEARNER. EACH ELEMENT CONTAINS THE ACCURACY OBTAINED ON A LEARNER WHEN ITS CLASSIFIER MODEL IS OBTAINED BY TRAINING THE LEARNER'S CLASSIFIER MODEL USING ITS OWN SENSOR DATA, LABELED WITH THE OUTPUT OF THE TEACHER CLASSIFIER. THE ACCURACIES THAT ARE WITHIN 10% FROM THE CORRESPONDING BASELINE ARE IN BOLD. RANDOM GUESSING WOULD YIELD 25% ACCURACY.

also the model is not suitable to provide classification. Thus, in the general case of transfer from any body position to any other body position, this approach does not perform well. This can be seen in Figure 6, where we show a three-dimensional projection (obtained with Principal Component Analysis) of the 4-dimensional feature spaces of the different sensor nodes. The mismatch in feature spaces arises because of different reasons. Firstly, it is inherent to the on-body sensor placement. While performing the same activity (e.g. Walking), different parts of the body move in a different way. This is reflected in the signals seen by the corresponding sensor nodes (see Figure 4) and ultimately in the feature spaces. A second reason is the orientation of the sensor nodes. Two sensor nodes placed very close to each other but with a different orientation will see similar signals but on different axes, which induces a rotation of the feature spaces.

Despite the generally not satisfactory performance of this method, it is very appealing because of its simplicity and reduced need for data exchange, both in terms of amount of data and time for which the sensor nodes need to run together. Furthermore, there are cases where the method reaches good accuracies (see Table II). This suggests that a hybrid approach could be used. In a preliminary phase, the sensor nodes could use the algorithms presented in [5] to detect their on-body placement and in case of correspondence, the naive classifier transfer can be operated. This would be an alternative approach to downloading a pre-defined model retrieved from a database, which would only need a local connection between the sensor nodes in contrast to a connection to a remote repository. The location discovery phase can also be performed by a correlation analysis of the sensor node signals or features, if we allow the sensor nodes to be deployed on the body together for some time, to assess whether they are mounted in close vicinity. Once the transfer has been performed, the Learner node can perform a self-calibration (as proposed in [7]), to try to adjust the feature space to the needs.

B. System-supervised approach

In contrast to the naive approach, the system-supervised method performs well (on average $17.3\,\%$ better than the naive approach), reaching accuracies which are in some cases at the baseline level. This is mostly due to the "masking" effect that we obtain by working at a different level of abstraction. In fact, by broadcasting labels, we remove the need of having an identical distribution of the activity classes in the feature spaces and the Teacher's feature space is then "hidden" behind the class labels, which are the only entities which are shared by the Teacher. The only factors influencing the performance reached with this method compared to the maximum achievable are the number of wrong labels (the Teacher is not perfect) and the distribution of these errors among the different activity classes.

Furthermore, with this method the Teacher and Learner do not need to implement either the same feature extraction algorithm or classifier and can be of different sensor modality. This provides freedom to the sensor node designers and differs significantly from the state of the art, where there are constraints in the relationships between feature spaces (e.g. in the transfer learning approach [11]). By masking the feature spaces, the system-supervised method is also suitable in case the labels are generated by simple sensors like reed switches (see [17]), case in which we do not have any feature space on the Teacher side.

The disadvantage of this approach is the need for the Teacher and Learner sensor nodes to be worn at the same time for a time span that allows the activities of interest to occurr at least a few times (typically one day in a practical implementation). In addition to that, the sensor nodes need to synchronize their clocks, so that the correspondance between each label and the Learner signal can be established properly. This can be done by using any of the existing protocols (for example [18]), which allow microsecond synchronization accuracies, which is sufficient for activity recognition purposes.

C. Comparison of different classifiers

When comparing the performance of the different classifiers, we observe that NCC is achieving lower baseline results compared to k-NN or SVM. The reason for this is that our feature spaces present overlap, particularly between the instances of Walking and Standing, and NCC stores only centroids (average of the training instances), which are then close to each other inducing classification mistakes. The overlap is due to the presence of many instances where the subject is performing only a step or is moving slightly. These instances have been labeled as Standing or Walking according to the activities in which they were embedded. When using an external Teacher, the labels are not anymore correct with a high probability. In this case, NCC is the classifier reaching the performance which is closest to the corresponding baseline. This again can be explained by the use of centroids, which are average examples for each activity. This averaging reduces the variation of the centroid positions in the presence of wrong labels. If the Teacher sensor node broadcasts a wrong label, a Learner operating with NCC experiences a shift of the corresponding activity centroid in the wrong direction, but only by an amount which is inversely proportional to the number of total instances. On the contrary, with an instance-based classifier like k-NN, if we insert a wrongly labeled instance, this remains in the Learner classifier model. Mitigation can be obtained by increasing the parameter k, so that the presence of a few mislabeled instances do not affect too much the classification output. The sensitivity of SVM to errors in the training labels appears to be higher than with the other classifiers. This is due to the fact that wrongly assigned training instances can induce changes in the set of support vectors. KNN and SVM can model much more complex feature spaces than NCC.

In terms of computational resources needed by the sensor nodes, the NCC classifier is the cheapest, with a number of operations of the order of $C{\cdot}F$. KNN are SVM instead grow in complexity with the number of instances N, but kNN can be implemented on sensor nodes by posing a limit on the number of stored instances and SVM can be efficiently implemented on nodes in a distributed way [19]. The system-supervised learning method, applied with the NCC classifier, has been already implemented and tested in the Context Cells.

Despite NCC being suitable for the present problem, its limitations should not be overlooked. NCC would in fact fail for complex feature spaces, where classes are for example represented by multiple clusters.

D. Discussion on selected examples

Table II shows that there are cases when also the naive model transfer gives satisfactory results. For example, when the RLA sensor node acts as a Teacher and the LLA sensor node as a Learner, this latter reaches an accuracy which is at the level of the baseline. This reflects the similarity between the signals collected by the accelerometers mounted on the arms, for most of the activities. When using the LUA sensor node as a Teacher, the accuracy drops a little bit, because of different orientations of the sensors, since there is often an angle between the arm and the forearm. When using the system-supervised learning, we can see that the accuracies reached by the LLA Learner are much less sensitive to the used Teacher. They are very close to the baseline for all Teachers, except for Hip, Shoe and Knee. The Hip and Shoe Teachers make more mistakes than the others when generating labels (see Tables I) and the Knee Teacher, though providing the least mistakes, is the one having the most bias in the label errors (see the confusion matrices in Figure 5). This generates bigger changes in the classifier decision boundaries compared to having evenly spread errors.

VII. Conclusion

In this paper we presented two methods to transfer activity recognition capabilities from a body-worn, trained sensor node (Teacher) to a previously unknown and untrained one (Learner). The first step for both methods is to establish a communication channel between the nodes. With the first method, it is sufficient that the nodes are connected for a short amount of time, during which the classifier models are sent over the radio link from one node to the other (a few seconds with ZigBee). The Learner achieves recognition accuracies which are in average $26.6\,\%$ lower than what would be obtained by training the Learner from ground truth. The second method requires the nodes to be synchronized and to stay connected while deployed both on the user's body for the duration of a typical usage day. Within this time span, only activity labels are sent by the Teacher through the radio link requiring a modest bandwidth (as low as $12\,\mathrm{bytes/s}$ for modes of locomotion recognition) and the Learner performs a batch learning at the end of the time span. After this transfer, Learners perform in average only $9.3\,\%$ lower than their maximum capabilities and the process is carried out without manual labeling or user intervention.

These methods are directly usable in context recognition applications which use networked sensor nodes, thanks to their low requirement on resources and communication, opening the way to open ended activity recognition scenarios, where new sensors can join a preexisting network and learn how to recognize activities. Thus, the on-body sensor infrastucture can be upgraded, for instance when the user buys new smart garments or sensorized gadgets, and the activity recognition capability can be preserved and translated to the new system after the upgrade. This is especially important in applications where the user may have personalized the recognition capabilities of its body-worn system. We envision such personalized applications to become more widespread in pervasisve computing [20].

We here showed the feasibility of the approaches for the case of modes of locomotion, but they are general and can be extended to gesture recognition or other context recognition tasks, which will be object of future investigation. In future work we will also compare the present results to a scenario where an ambient intelligence environment, as opposed to wearable nodes, trains the wearable sensor nodes. This extends the work presented in [17], where labels are gathered from reed switches automatically. We see the two approaches as complementary and very promising when combined, enabling wearable sensors to opportunistically exploit all the labels available in a given situation, trying to make best use of the open endedness and richness of the sensor networks.

ACKNOWLEDGMENT

The project OPPORTUNITY acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 225938.

REFERENCES

[1] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing: Proc. of the 2nd Int'l Conference*, Apr. 2004, pp. 1–17.

[2] P. Lukowicz, J. A. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, and T. Starner, "Recognizing workshop activity using body worn microphones and accelerometers," in *Pervasive Computing: Proc. of the 2nd Int'l Conference*, Apr. 2004, pp. 18–22.

[3] C. Zysset, K. Cherenack, T. Kinkeldei, and G. Tröster, "Weaving integrated circuits into textiles," in *Proceedings of the 14th IEEE International Symposium on Wearable Computers (ISWC 2010)*. IEEE Computer Society, Oct. 2010.

[4] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole, "Which way am i facing: Inferring horizontal device orientation from an accelerometer signal," in *Wearable Computers, 2009. ISWC '09. International Symposium on*, 4-7 2009, pp. 149 –150.

[5] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster, "Where am i: Recognizing on-body positions of wearable sensors," in *In: LOCA04: International Workshop on Locationand Context-Awareness*. Springer-Verlag, 2005, pp. 264–275.

[6] K. Kunze and P. Lukowicz, "Symbolic object localization through active sampling of acceleration and sound signatures," in *Proc. 9th Int. Conf. on Ubiquitous Computing - Ubicomp 2007*, Innsbruck, Austria, 2007.

[7] K. Förster, D. Roggen, and G. Tröster, "Unsupervised classier self-calibration through repeated context occurences: is there robustness against sensor displacement to gain?" in *Proceedings of the 13th International Symposium on Wearable Computing*. IEEE Computer Society, 2009, pp. 77–84.

[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," in *Proceedings of the fifth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2009)*. IEEE press, 2009.

[9] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, K. Fishkin, B. Schiele, P. Nixon, and A. Quigley, Eds. Springer Berlin / Heidelberg, 2006, vol. 3968, pp. 1–16.

[10] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345 – 1359, 2010.

[11] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *ICML '07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM, 2007, pp. 193–200.

[12] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[13] T. van Kasteren, G. Englebienne, and B. J. A. Kröse, "Transferring knowledge of activity recognition across sensor networks," in *Pervasive*, ser. Lecture Notes in Computer Science, P. Floréen, A. Krüger, and M. Spasojevic, Eds., vol. 6030. Springer, 2010, pp. 283–300.

[14] A. Calatroni, C. Villalonga, D. Roggen, and G. Tröster, "Context cells: Towards lifelong learning in activity recognition systems," in *Proceedings of the 4th European Conference on Smart Sensing and Context (EuroSSC)*, ser. LNCS. Springer, 2009.

[15] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. R. Millán, "Collecting complex activity data sets in highly rich networked sensor environments," in *Seventh International Conference on Networked Sensing Systems*, 2010.

[16] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[17] A. Calatroni, D. Roggen, and G. Tröster, "A methodology to use unknown new sensors for activity recognition by leveraging sporadic interactions with primitive sensors and behavioral assumptions," in *Proc. of the Opportunistic Ubiquitous Systems Workshop, part of 12th ACM Int. Conf. on Ubiquitous Computing*, 2010. [Online]. Available: http://www.wearable.ethz.ch/resources/UbicompWorkshop_OpportunisticUbiquitousSystems

[18] D. Cox, E. Jovanov, and A. Milenkovic, "Time synchronization for zigbee networks," in *System Theory, 2005. SSST '05. Proceedings of the Thirty-Seventh Southeastern Symposium on*, March 2005, pp. 135–138.

[19] K. Flouri, B. Beferull-Lozano, and P. Tsakalides, "Training a svm-based classifier in distributed sensor networks," in *Proc. of 14nd European Signal Processing Conference 2006*, pp. 1–5.

[20] D. Roggen, C. Lombriser, M. Rossi, and G. Tröster, "Titan: An enabling framework for activity-aware "pervasiveapps" in opportunistic personal area networks," *EURASIP Journal on Wireless Communications and Networking*, 2011.