Partially Supervised Gesture Recognition

Narges S. Milani[1], Denijel Sakic[2], Arne Grumpe[1], Christian Wöhler[1], Gernot Fink[2]

[1]Image Analysis Group
TU Dortmund University
Otto-Hahn-Str. 4
D-44227 Dortmund

[2]Lehrstuhl Informatik XII
TU Dortmund University
Otto-Hahn-Str. 16
D-44227 Dortmund

1 Introduction

Gestures provide an intuitive framework for interaction between humans and cognitive systems. This kind of interaction requires a reliable recognition of user-performed gestures by the system. Because of the variable nature of gestures performed by different individuals, it is desirable that the system is able to learn from a limited initial set of manually labeled samples performed by a small number of persons and to extend its knowledge based on gestures performed by additional persons previously unknown to the system, where it is preferable to minimize the manual labeling effort.

Recognizing gestures includes the capture of the temporal trajectories of human body-parts, extraction of the gesture-specific characteristics and the distinction of various gestures using these heuristic characteristics. The novelty of the work presented in this study is in the development of a learning framework by means of diverse ensembles of classifiers [1] which addresses the problem of gesture recognition by classifying fixed-length trajectory segments, as opposed to the variable-length trajectory problems usually solved by applying hidden Markov models (HMMs) [2] or dynamic Bayesian networks [3]. Similar ensemble systems have been developed in recent works such as [4] with the distinction of using multi-camera framework for 3D data capture, while in the current work the gesture motions are captured using the Microsoft Kinect sensor to directly generate 3D trajectory information. This sensor is an active 3D vision system capable of determining and tracking objects and human body joints in its field of view. The low price and relatively high measurement accuracy of this sensor has made it one of the most researched-on sensors in the field of computer vision recently.

The use of ensemble of multiple classifiers has been shown to improve recognition performance in a wide range of pattern recognition problems

[5, 1]. Some popular ensemble creation methods like boosting and bagging combine the same type of classifiers with different subsets or representations of the training data, resulting in a high performance ensemble [6]. In the present work the diversity of the ensemble is achieved by using different normalization and dimensionality reduction methods for the training data. For the final decision of the ensemble two different methods are used and their performances are compared.

In the following section, an overview of the basics of a learning algorithm and the ensemble forming and the decision making criteria is presented. In Section 3 a brief introduction to the Kinect sensor and the employed database is discussed. Section 4 explains the experimental setup and finally in Section 5 the results are presented and explained.

2 Classification Basics

The basic principle of a classification system is to classify each input vector into predefined classes. In this section an overview of the classifiers employed in Section 4 of this study is provided.

2.1 Polynomial Classifier

A polynomial classifier (PC) determines the probability of a given feature vector to belong to a certain class. The classifier decision function is a polynomial of degree d consisting of the weighted sum of all monomial terms with total degrees in the range $0, \dots, d$ of the elements of the original data vector $\vec{v}$, and is given by:

$$\vec{d}_{PC}(\vec{v}) = W^T . \vec{x}(\vec{v})$$

where $x(\vec{v}) = \{v_1, \dots, v_m, \dots, v_1 v_m, \dots, v_1^d, \dots, v_m^d\}$ is the polynomial structure list and W^T is a coefficient matrix. The sum of the elements of d_{PC} always corresponds to 1. A detailed description of the polynomial classifier can be found in [7].

2.2 Multilayer Perceptron

Neural networks provide a (generally nonlinear) mapping from the space of input features into the decision space. They are composed of layers of neurons, and the mapping function is learned from training data. The multilayer perceptron is a neural network with feed-forward connections consisting of an input layer, in the general case several hidden layers, and one layer encoding the network output. The number of input neurons corresponds to the number of features used for classification. The number of neurons in the output layer coincides with the number of classes in the classification problem, thus providing a decision value for each class. The connection between two consecutive neurons is weighted, and for each

neuron of a specific layer connections exist with all neurons from the previous layer. The activation of a neuron is given by its (mostly nonlinear) activation function, whose argument corresponds to the sum of its inputs multiplied by the weights of the corresponding connections minus a scalar offset (threshold) value [6].

The MLP in this work has a single hidden layer and the activation function for the neurons in this layer is the hyperbolic tangent function while the output layer neurons are activated by a linear function. Training an MLP for classification means calculating all the weights and thresholds of the neurons such that the desired neuron in the output layer displays the maximum activation value for each input feature vector of the corresponding class. For a more detailed description on such algorithms refer to [6, 8].

2.3 Partially-supervised Learning

Partially-supervised learning or self-learning (as mentioned in [9]) is an iterative process of learning from a small set of labeled data and a much larger set of unlabeled data. The objective is to use or determine a minimum number of human-labeled data in order to train a reliable algorithm. Accordingly, in semi-supervised learning (SSL) the system is trained with a small set of labeled data (here called the “training set”), and going through the unlabeled data pool (called the “learning set”) in an iterative manner, the samples that can be classified reliably by the system are selected. These samples are evaluated by testing on another set of labeled data (called the “test set”) and then added to the training samples with their estimated class labels, enabling the system to re-train on a larger dataset [10].

In contrast, an active learning (AL) system, after being trained on the same labeled dataset, selects the samples containing the highest amount of information about the classification problem from the set of unlabeled samples, which, once labeled by the human expert, would increase the classification performance. The labeled samples are similarly added to the training set, which is then used to re-train the AL system [11].

2.4 Ensemble

A combination of multiple diverse classifiers provides independent and possibly more accurate views about the classification problem at hand. The advantage of using ensembles of classifiers has been presented in the literature [1]. The possibility to combine several independent class assignments for the same input results in often more reliable classification systems compared to single classifiers. This is especially important for

semi-supervised learning systems, since adding a misclassified sample to the training set would affect the success rate of the system.

The ensemble consists of base classifiers including neural networks and polynomial classifiers, each of which determines an individual class assignment. The overall decision is then aggregated by a majority vote or by a weighted sum based on the confidence band values of the classifiers [9, 4].

2.5 Confidence-based Inference

The most important step in partially supervised learning is the sample selection for re-training the model. One of the approaches for active selection of the training data is the use of confidence bands as an evaluation measure [9].

In definition, a confidence interval encompassing a model or function estimating a value by regression is the area in which the probability for the true model to reside is 1- α. The value 0.05 is commonly used for α, meaning that the true model is enclosed in the interval with a probability of 95% [12]. For an evaluation process the extent of the bands in the data space can be used.

There are several approaches to calculate confidences. We depict the algorithm by Martos et al. [13] which requires the covariance matrix of the parameters. In theory it is possible to calculate the confidence band for every classifier which allows for a calculation of the Jacobian matrix containing the partial derivatives of the residuals of the model function with respect to the model parameters. For a PC we have $\mathrm{J}_{PC} = [\mathrm{x}(\vec{v}_1)\ \mathrm{x}(\vec{v}_2)\ \ldots \mathrm{x}(\vec{v}_M)]^T$, where M is the number of elements in the polynomial structure list $\mathrm{x}(\vec{v})$. From this we can calculate the matrix K corresponding to the Jacobian matrix weighted by the uncertainty of the mislabeled data (either by the human expert or the learning system). This matrix is utilized to compute the covariance matrix according to

$$K = \frac{\mathrm{J}_{PC}}{\sigma_i}$$

$$C = (K^T K)^{-1}$$

To compute confidence band values, the Jacobian vector $\vec{g}$ of the model function $\vec{g} = \partial \vec{d}_{PC} / \partial \omega$ is also required, where ω denotes the elements of the model parameter matrix W. For an arbitrary sample $w = x(\vec{v}_i)$, the confidence interval is proportional to $\sigma_i \sqrt{c(w)}$ with $c(w) = \vec{g}^T C \vec{g}$.

Since the uncertainty of the specific label σ_i is not known, identical uncertainties for all labels are assumed in [9], which simplifies the

confidence band calculation in the sense of independence on an estimate of σ. The so-called "normalized confidence" [9] value then becomes

$$\sigma_{C,norm}(w) = \sqrt{c(w)}$$

The confidence values are computed for all samples at each iteration of the learning process and can be used as a selection criterion.

3 Data Acquisition and Database formation

The developed of a partially supervised learning system is used for gesture recognition in an intuitive framework for human-machine-interaction. One of the challenges of the learning algorithm described above is the need for 3D trajectory information collection. This is solved by the use of the Microsoft Kinect sensor which is capable of tracking the body joints in real-time. In this study, a total of 10 distinctive gesture classes were introduced to a group of 18 individuals, who have been asked to perform the gestures in front of a Kinect system. These gestures include the classes proposed in [14]: (1) up, (2) down, (3) come here, (4) go away, (5) stop, (6) point, (7) circle, (8) wave horizontally, and (9) wave vertically plus (10) idle as an additional class. Based on the Microsoft software development kit [2], a software has been implemented to record the 3D trajectories of the head and both hands of the person. Specific feature extraction techniques similar to those proposed in [14] are used to form a database for the subsequent classification application. In the following sections a brief introduction of the Kinect sensor together with the steps regarding database creation is presented.

3.1 Kinect Sensor

The Microsoft Kinect sensor comprises an RGB camera, an near-infrared laser projection unit, and a near-infrared camera. The sensor is able to generate 3D data by a triangulation process based on the projection of structured light, as described by the inventors [15]. Hence, three kinds of online outputs can be accessed using the sensor: the RGB image of the scene in front of the sensor, the depth map consisting of the 3D coordinates of each image point, and the 3D coordinates of the joints of the human body in case a human body is detected in the field of view. To access these data in an online manner, we have developed a software based on the Microsoft Kinect for Windows SDK, which is available at http://www.microsoft.com/en-us/kinectforwindows.

3.2 Gesture Database

A database of 3D trajectories of various gestures has been created prior to the learning phase. After the initial data recording, the gestures are concatenated and labeled. The "idle" gesture (standing still with respect to

the camera with both hands relaxed) is used as a reference for detecting the beginning and ending of gestures.

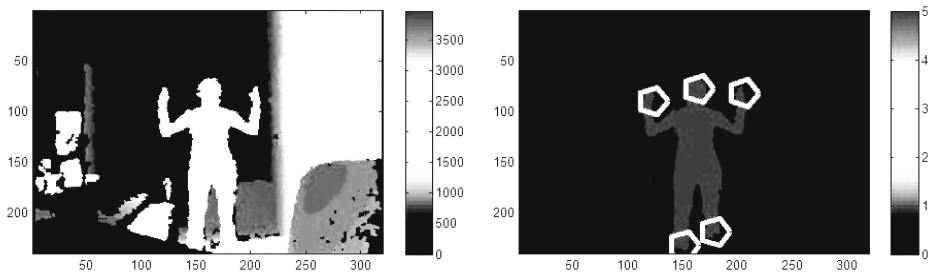

Figure 1: Left: Depth map of the Microsoft Kinect field of view. Right: Hands, head, and feet of the detected human in the field of view, marked with pentagons.

The features used in this work have been adopted from [14]. These features require information about the position of the person's head and both hands as well as the height for trajectory normalization. It is desired that the gestures are independent of the orientation and the position of the person relative to the sensor. To address this problem, "delta features" are introduced denoting the relative change from the previous window. From these, a total of 90 features can be calculated for each window of frame sequences, including hand trajectory, normalized trajectory, velocity in different directions, curvature, orientation and vicinity (which is described as the general shape of a feature window). Both 3D features and 2D features obtained by projection of the 3D data into a so-called "action plane" are extracted. For more detailed explanation of the features refer to [14]. Different from [14], the action plane is not inferred from the hand position information as that would require completion of the gesture before computation of the features. Instead, the plane used to generate the 2D features corresponds to a vertical plane inclined by 45° with respect to the direction into which the person is oriented. The orientation of the person is assumed to be known (in our scenario all persons are directed towards the sensor). Our trajectory database is available online at http://www.bv.e-technik.tu-dortmund.de.

4 Experimental Setup

As mentioned earlier, a total of 18 individuals participated in performing the gestures and a total of 2119 gestures have been recorded, as can be seen in Table 1 along with the number of samples in each gesture class.

Feature extraction is carried out by a sliding window approach (window size = 5, shift = 1) similar to [4]. For the number of subsequent resampled points (frames) comprehensive tests have been carried out. The initial results suggested that the choice is closely related to the dimension chosen

for the dimensionality reduction technique. Results of a grid search on the problem concluded on the selection of resampling window size 4 with shift 1 and dimension 25.

To ensure the heterogeneity of the ensemble, different representations of the features (i.e. normalization methods) as well as various dimensionality reduction techniques are combined with each classifier type. For normalization we tried three methods including min-max rescaling, softmax scaling and Z-score normalization. Softmax scaling is a nonlinear method which compresses the data exponentially in the [0,1] interval.

As for dimensionality reduction, PCA [6, 7, 8], LDA [6], and ICA [16] followed by LDA methods were subject to testing. The degree of the polynomial classifier was set to 1, 2 and 3 since the higher degrees cannot be implemented on a normal desktop computer because of the large size of the weight matrix. The polynomial classifier is additionally favorable in this work because of its short training time and robust behavior regarding partially mislabeled data.

The MLP classifier used here has one single hidden layer, where the number of hidden neurons is automatically determined ensuring that the number of training samples and the dimensionality of the data are at least five times and two times larger than the number of network parameters, respectively.

Gesture name	Total gestures	Total samples
Up	201	4393
Down	204	3971
Come here	222	4179
Go away	225	3985
Stop	202	3733
Point	201	3544
Circle	202	5028
Wave	202	5149
Vertical wave	204	5566
Idle	256	3830
Sum	2119	43378

Table 1 : Total number of gestures and samples per class in the database

4.1 Ensemble Creation Setup

In the context of ensemble learning, a common assumption is that combining heterogeneous classifiers within the ensemble which do not mislabel the same individual samples leads to an increase in recognition performance [1]. Comprehensive tests have been carried out in this regard on a total of 36 system architectures, as defined by the different approaches to normalization and dimensionality reduction described above. The systems were trained on a fraction of the training set and were tested on an independent test set. The experiment was repeated for 5% to 30% of the

training set (with 5% increments) (cf. Tables 2 and 3). The success rate of the classifiers trained on only a fraction of the data is equally important as the diversity of the ensemble.

There are several approaches to aggregating the outputs of the individual classifiers. Assume that each of the L base classifiers generates a class assignment y_L and a decision value vector $d_L = (d_{L1}, d_{L2}, \dots, d_{LN})$ for a given sample, where $\sum_{i=1}^{N} d_{Li} = 1$. For majority voting, the class assignment is considered confident enough to be adopted if the sum of the decision values of all base classifiers is higher than a threshold. The second approach is to compute a sum of the decision values weighted by the renormalized confidence values. The ensemble decision value for the i-th class is determined based on the decision value $d_{i,k}$ and the renormalized confidence values $\sigma_{i,k}$ of the i-th class and the k-th base classifier such that $d_{ens,i} = \sum_k (d_{i,k}/\sigma_{i,k})$.

5 Results

The problem of gesture recognition is tackled by starting with a two-class problem and adding one class at a time. The classes are chosen based on the initial confusion matrix, such that one has a better than average and the other has a worse than average recognition rate. For the consecutive tests the classes are added based on the same concept while at each problem it has been tried that the system trained with the initial training data can achieve a recognition rate of 70%. Base classifiers are chosen based on the conditions described earlier and the results of Tables 2 and 3. Finally, classifiers number 11, 18 and 36 are chosen to form the ensemble.

In the semi-supervised learning scenario using a 6-fold cross-validation scheme, 3 persons are randomly chosen for the training and 3 for the testing set, leaving 12 randomly chosen persons for the learning set. The system is trained with 7% of the training data (for the 2 class problem) and 15% of the training data (for more than 2 classes), and the rest of the samples of the training persons are added to the learning set. At each iteration of the algorithm a label is assigned to each of the samples in the unlabeled learning set, and a sample is be added to the training set if all the following criteria are fulfilled:

1. All three base classifiers agree on the assigned label.
2. The maximum decision value of all three classifiers is higher than a threshold θ_1.
3. The difference between the two highest decision values of the three classifiers is larger than a threshold θ_2.
4. The normalized confidence values of all three classifiers do not exceed a threshold θ_3.

#	Type	Degree	Normalization	Dim Red	5% (≈ 1655 samples)	10% (≈ 3278 samples)	15% (≈ 4673 samples)	20% (≈ 6464 samples)	25% (≈ 8255 samples)	30% (≈ 9993 samples)
1	PC	1	Z-score	PCAglobal	46.11±3.19	44.97±3.18	46.92±2.48	47.19±2.84	45.06±3.21	45.70±3.79
2	PC	1	Softmax scaling	PCAglobal	47.15±3.45	45.50±3.23	47.94±2.66	48.14±2.59	45.41±3.10	45.86±3.79
3	PC	1	Min-max rescaling	PCAglobal	47.21±4.18	45.23±3.51	48.26±3.58	49.06±2.87	45.84±3.42	46.65±4.10
4	PC	1	Z-score	LDA	48.51±2.61	49.94±2.55	54.02±2.96	53.55±3.13	51.21±3.19	51.35±4.28
5	PC	1	Softmax scaling	LDA	51.86±3.42	50.61±3.51	53.40±2.94	53.93±2.32	51.21±3.83	51.49±3.60
6	PC	1	Min-max rescaling	LDA	48.39±4.07	48.21±3.96	52.20±3.35	52.45±3.33	49.55±3.59	49.55±4.25
7	PC	1	Z-score	ICAUNDLDA	55.42±3.65	57.71±3.51	61.63±3.37	62.63±2.96	60.00±3.49	60.82±4.35
8	PC	1	Softmax scaling	ICAUNDLDA	53.92±3.28	56.97±3.30	61.79±3.37	61.88±2.35	59.64±3.95	60.56±4.21
9	PC	1	Min-max rescaling	ICAUNDLDA	55.92±2.90	57.95±3.90	62.81±3.55	63.56±2.21	62.13±3.97	62.76±4.61
10	PC	2	Z-score	PCAglobal	63.98±2.99	64.81±3.80	67.22±2.64	69.95±2.35	66.80±3.21	68.46±3.15
11	**PC**	**2**	**Softmax scaling**	**PCAglobal**	**65.14±3.17**	**65.49±4.23**	**68.03±3.15**	**70.33±2.61**	**67.33±3.65**	**68.99±3.47**
12	PC	2	Min-max rescaling	PCAglobal	61.65±3.17	62.74±3.83	64.64±3.04	67.12±2.12	64.34±3.34	66.43±3.18
13	PC	2	Z-score	LDA	60.93±3.03	63.97±3.65	66.65±2.71	67.82±2.37	65.08±3.10	66.99±4.72
14	PC	2	Softmax scaling	LDA	63.13±2.82	64.00±3.66	66.38±3.10	68.27±2.86	65.48±3.70	66.64±3.89
15	PC	2	Min-max rescaling	LDA	57.23±2.98	59.57±4.32	63.83±2.81	65.07±2.85	62.74±3.23	63.43±4.47
16	PC	2	Z-score	ICAUNDLDA	59.16±2.70	64.67±3.60	68.65±2.46	69.75±2.79	69.52±2.89	70.53±2.83
17	PC	2	Softmax scaling	ICAUNDLDA	56.67±2.00	63.18±2.70	67.85±2.24	69.73±1.89	68.59±3.29	69.76±2.21
18	**PC**	**2**	**Min-max rescaling**	**ICAUNDLDA**	**57.58±2.60**	**65.51±3.06**	**68.85±2.60**	**70.43±2.63**	**70.14±2.87**	**71.51±2.69**
19	PC	3	Z-score	PCAglobal	8.85±1.87	14.58±5.45	49.80±3.61	64.63±3.77	67.29±4.02	71.34±3.33
20	PC	3	Softmax scaling	PCAglobal	9.40±1.61	16.38±6.16	53.48±3.56	67.60±3.30	68.72±3.99	72.62±3.46
21	PC	3	Min-max rescaling	PCAglobal	10.06±1.02	14.21±4.89	47.11±3.90	62.20±3.12	64.76±4.12	68.82±3.09
22	PC	3	Z-score	LDA	9.95±1.06	14.70±5.37	47.87±4.57	61.28±3.44	64.88±4.20	69.65±3.39
23	PC	3	Softmax scaling	LDA	9.53±1.94	15.63±6.21	50.63±3.53	63.30±2.89	67.03±4.07	70.07±3.03
24	PC	3	Min-max rescaling	LDA	10.00±1.67	14.05±3.83	45.52±4.05	58.43±4.01	62.37±3.69	65.49±3.67
25	PC	3	Z-score	ICAUNDLDA	10.68±1.70	13.26±4.34	44.47±3.79	56.28±4.06	62.44±4.35	67.63±2.23
26	PC	3	Softmax scaling	ICAUNDLDA	7.33±1.01	11.70±4.39	43.36±3.72	56.95±2.80	61.72±4.03	66.57±2.10
27	PC	3	Min-max rescaling	ICAUNDLDA	9.17±1.84	13.24±4.57	45.34±3.41	57.20±3.70	63.30±3.09	68.05±2.03

Table 2 : Recognition rates of supervised learning for the polynomial classifiers (PC) with different configurations. The experiment is repeated 10 times on a 6-fold cross validation scheme for each configuration. The finally selected classifiers for the ensemble are denoted by bold characters (number 11 and 18).

#	Type	Normalization	Dim Red	5% (≈ 1608 samples)	10% (≈ 3345 samples)	15% (≈ 4813 samples)	20% (≈ 6479 samples)	25% (≈ 8046 samples)	30% (≈ 9900 samples)
28	MLP	Z-score	PCAglobal	45.70±3.28	58.46±3.71	65.31±3.15	67.01±3.48	68.41±2.98	71.53±4.55
29	MLP	Softmax scaling	PCAglobal	44.67±1.99	58.81±2.85	65.56±3.34	67.15±3.28	68.98±2.82	72.76±4.56
30	MLP	Min-max rescaling	PCAglobal	43.44±2.54	59.03±3.05	64.06±3.10	66.25±3.59	66.81±3.07	69.92±5.24
31	MLP	Z-score	LDA	46.48±4.21	61.23±2.44	66.88±2.52	68.13±2.64	69.15±2.76	71.91±2.99
32	MLP	Softmax scaling	LDA	52.34±3.48	57.44±4.31	58.11±4.04	59.39±3.47	57.16±3.81	59.12±3.67
33	MLP	Min-max rescaling	LDA	44.43±3.30	59.49±2.37	63.00±2.90	65.30±3.49	65.88±3.57	68.95±4.07
34	MLP	Z-score	ICAUNDLDA	43.51±4.44	60.60±3.34	70.58±2.79	71.71±1.99	69.90±3.73	72.89±3.17
35	MLP	Softmax scaling	ICAUNDLDA	41.90±2.94	61.36±2.11	69.87±2.87	71.13±1.96	69.79±3.23	72.01±3.05
36	**MLP**	**Min-max rescaling**	**ICAUNDLDA**	**42.00±3.88**	**63.88±3.02**	**71.11±2.46**	**72.52±1.99**	**70.60±3.67**	**72.91±3.86**

Table 3 : Recognition rates of supervised learning for the Multilayer Perceptron (MLP) with different configurations. The experiment is repeated 10 times on a 6-fold cross validation scheme for each configuration. The finally selected classifiers for the ensemble are denoted by bold characters (number 36).

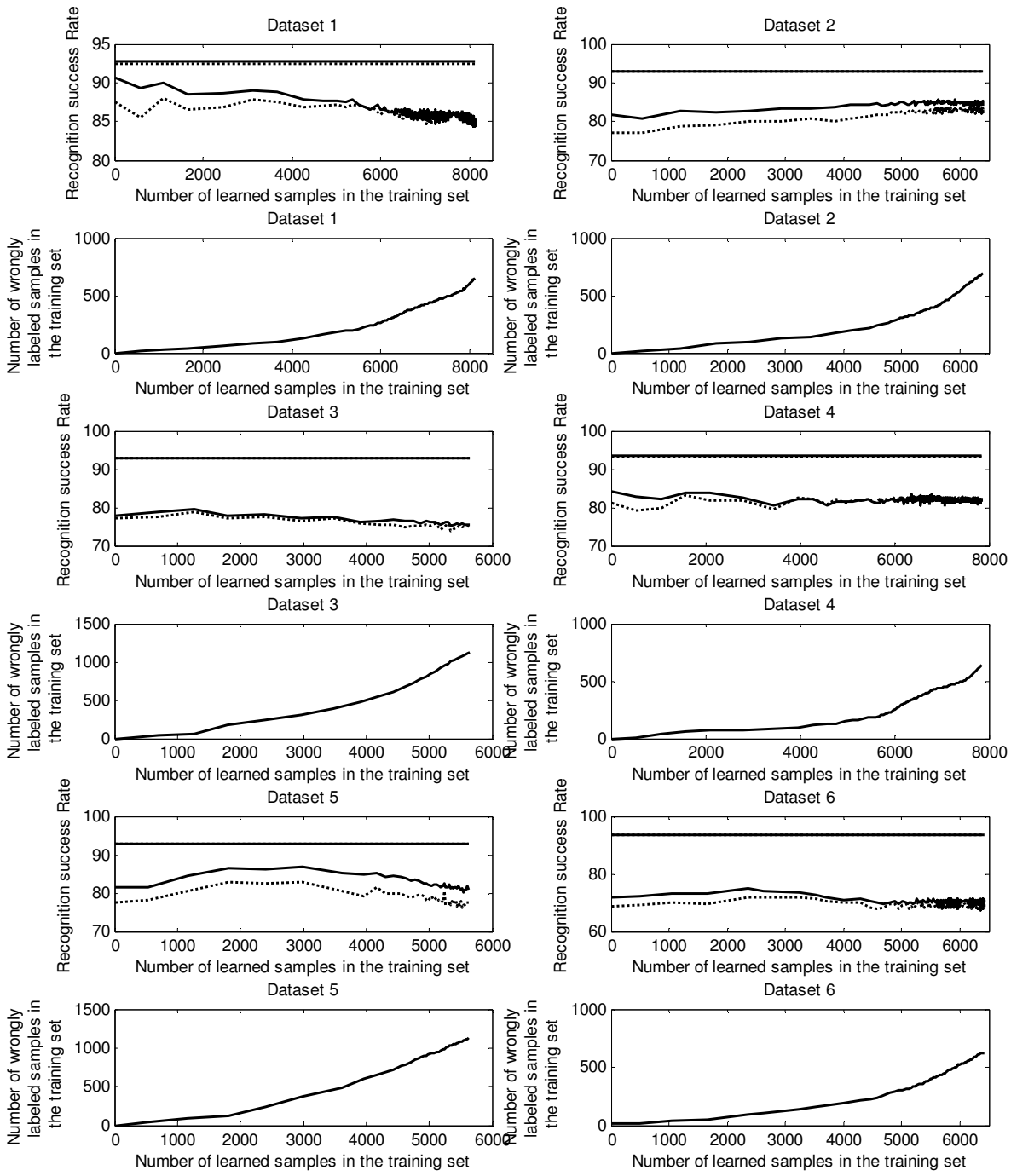

Figure 2 : Results of the two-class gesture classification problem (classes “come here” and “circle”). Each data set represents one of the cross-validation folds. In the recognition rate diagrams, the full and the dotted curves represent the ensemble results using majority voting and weighting by renormalized confidence, respectively. The horizontal line denotes the recognition rate achieved by supervised learning on the same test set.

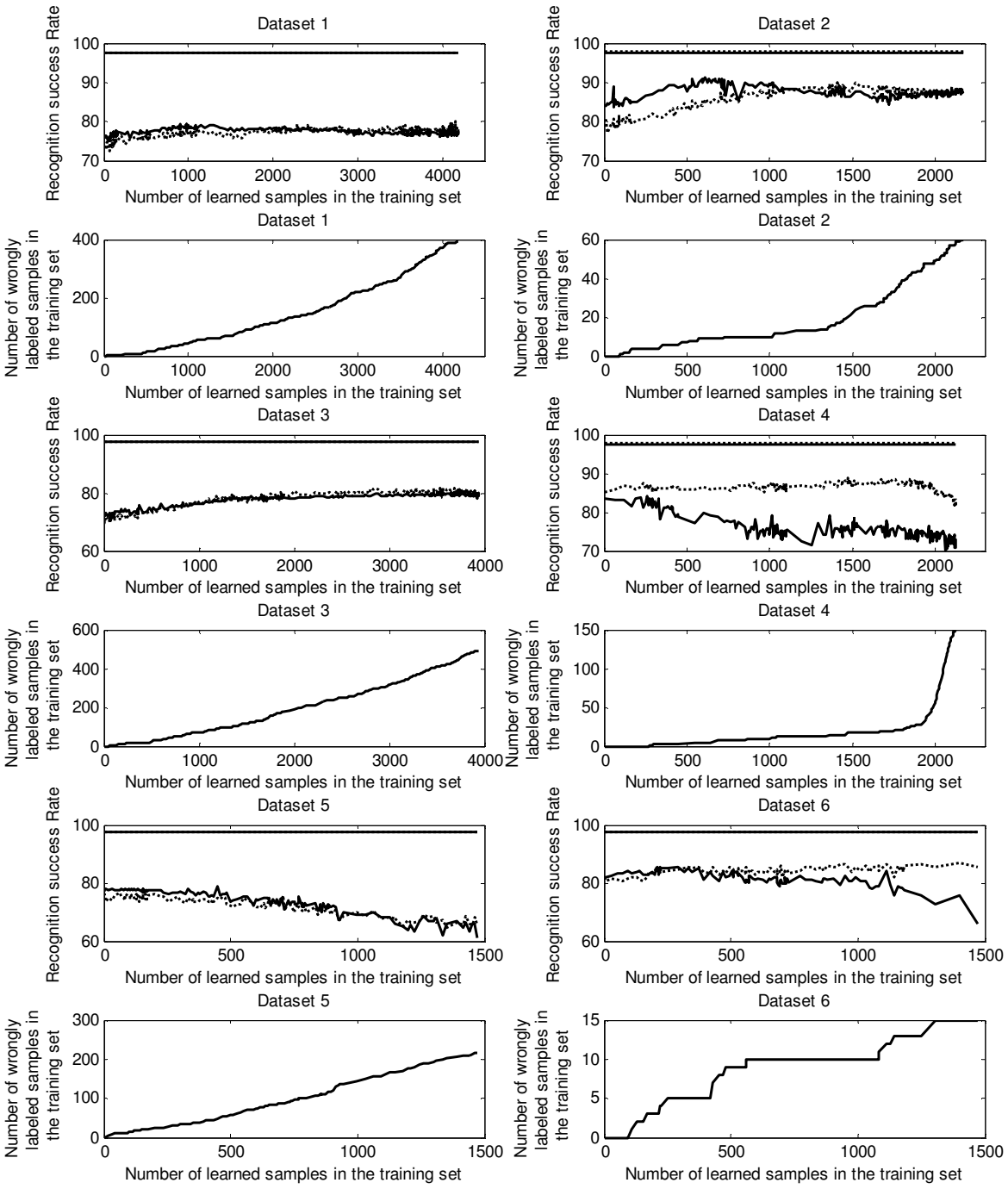

Figure 3 : Results of the three-class gesture classification problem (classes “come here”, “circle”, and “wave”). Each data set represents one of the cross-validation folds. In the recognition rate diagrams, the full and the dotted curves represent the ensemble results using majority voting and weighting by renormalized confidence, respectively. The horizontal line denotes the recognition rate achieved by supervised learning on the same test set.

For testing, dynamic thresholding has been applied: The initial values for the thresholds are selected as $\theta_1 = 0.7, \theta_2 = 0.1$ and $\theta_3 = 0.9$, and the ensemble starts to learn as long as no more unlabeled samples are selected. At this point the thresholds θ_1 and θ_3 are increased by 2%, respectively.

The increase of θ_1 aims at reducing the number of wrongly labeled data in the training set, requiring a higher decision value for the respective samples which have a higher confidence band value with respect to the previously learned samples. The value of θ_2 remains unchanged.

The results (Figures 2 and 3) show that the recognition rate of the semi-supervised learning algorithm is highly dependent on the persons performing the gestures in the initial training set. In some of the cross-validation data sets, the recognition rate increases with the number of additional learned samples in the training set, while in other folds the recognition rate is affected by the wrongly labeled samples. The other factor determining the fate of semi-supervised learning is the initial recognition rate. In our experiments it has proven hard to improve an initial recognition rate of over 80%. Through dynamic thresholding, after learning all possible unlabeled samples with higher confidences, the system is able to continue the learning process by including samples with a relatively longer distance to the true classification model. These samples are not necessarily the most informative samples to learn and therefore may or may not have a significant effect on the increase of the recognition rate. In other words, the effects of the wrongly labeled samples learned during the first phases of the semi-supervised process may be compensated by the increase of correctly labeled samples in the consecutive phases, leading to an overall increase of the recognition rate during semi-supervised learning.

6 Conclusion

In this study we have examined the problem of semi-supervised ensemble learning of gestures acquired with a Microsoft Kinect sensor. Starting with a small initial number of gestures performed by only three persons, the gesture recognition system adapts itself to new, previously unknown persons by applying a semi-supervised learning approach. Here, the achievable recognition rate on the test set depends on the individual persons who performed the gestures in the initial training set.

Future work will involve an investigation of the suitability of using an active learning algorithm together with semi-supervised learning or the development of a measure to identify the wrongly labeled samples in the training set. Additionally, developing a method to stop the learning process in order to prevent over-training will be beneficial.

7 Literatur

[1] L. Rokach, Pattern Classification using Ensemble Methods, World Scientific, 2010.

[2] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE transactions on Systems, Man and Cybernetics, Part C,* vol. 37, no. 3, pp. 311-324, 2007.

[3] H. Suk, B. Sin and S. Lee, "Hand gesture recognition based on dynamic bayesian network framework," *Pattern Recognition,* vol. 43, no. 9, p. 3059–3072, 2010.

[4] J. Schumacher, D. Sakic, A. Grumpe, G. A. Fink and C. Wöhler, "Active Learning of Ensemble Classifiers for Gesture Recognition," in *Proceedings of DAGM-OAGM Pattern Recognition Symposium*, Graz, Austria, 2012.

[5] L. Kuncheva, Combining Pattern Classifiers: methods and algorithms, Wiley, 2004.

[6] S. Marsland, Machine Learning: An Algorithmic Perspective, Chapman & Hall/CRC Machine Learning & Pattern Recognition Series. CRC Press, 2009.

[7] J. Schürmann, Pattern Classification: A Unified View of Statistical and Neural Approaches, John Wiley & Sons, Inc., 1996.

[8] C. M. Bishop, Pattern Recognition and Machine Learning, Cambridge, UK: Springer, 2006.

[9] M. Hillebrand, C. Wöhler, L. Krüger, U. Kreßel and F. Kummert, "Self-learning with Confidence Bands," in *Proceedings 20 Workshop Computational Intelligence*, Dortmund, Germany, 2010.

[10] X. Zhu and A. B. Goldberg, Introduction to Semi-Supervised Learning, Morgan & Claypool, 2009.

[11] Y. Freund, H. Seung, E. Shamir and N. Tishby, "Selective sampling using the query," *Machine Learning,* vol. 28, pp. 133-168, 1997.

[12] O. J. Kardaun, Classical Methods of Statistics, Springer, 2005.

[13] A. Martos, L. Krüger and C. Wöhler, "Towards Real Time Camera Self Calibration: Significance and Active Selection," in *Proceedings of the 4th International Symposium on 3D Data Processing, Visualization and Transmission*, Paris, France, 2010.

[14] J. Richarz and G. A. Fink, "Visual Recognition of 3D Emblematic Gestures in an HMM Framework," *Journal of Ambient Intelligence and Smart Environments,* vol. 3, no. 3, pp. 193-211, 2011.

[15] B. Freedman, A. Shpunt, M. Machline and Y. Arieli, "Depth mapping using projected patterns". US Patent 2010/0118123 A1, 13 May 2010.

[16] M. Lennon, G. Mercier, M. Mouchot and L. Hubert-Moy, "Independent Component Analysis as a tool for the dimensionality reduction," in *IEEE International Geoscience and Remote Sensing Symp.*, Sydney, Australia, 2001.