Computers and Electronics in Agriculture 93 (2013) 17–26

Contents lists available at SciVerse ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Sow-activity classification from acceleration patterns: A machine learning approach

Hugo Jair Escalante [a,*], Sara V. Rodriguez [b], Jorge Cordero [b], Anders Ringgaard Kristensen [c], Cécile Cornou [c]

[a] *Computational Sciences Department, Instituto Nacional de Astrofísica, Óptica y Electrónica, Tonantzintla, 72840 Puebla, Mexico*

[b] *Graduate Program in Systems Engineering, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, 66450 Nuevo Leon, Mexico*

[c] *HERD – Centre for Herd-oriented Education, Research and Development, Department of Large Animal Sciences, University of Copenhagen, Groennegaardsvej 2, DK-1870 Frederiksberg C, Denmark*

A R T I C L E   I N F O

Article history:
Received 21 June 2012
Received in revised form 17 December 2012
Accepted 13 January 2013

Keywords:
Accelerometer measurements
Logitboost with trees
Pattern classification
CLOP
Sow-activity classification

A B S T R A C T

This paper describes a supervised learning approach to sow-activity classification from accelerometer measurements. In the proposed methodology, pairs of accelerometer measurements and activity types are considered as labeled instances of a usual supervised classification task. Under this scenario sow-activity classification can be approached with standard machine learning methods for pattern classification. Individual predictions for elements of times series of arbitrary length are combined to classify it as a whole. An extensive comparison of representative learning algorithms, including neural networks, support vector machines, and ensemble methods, is presented. Experimental results are reported using a data set for sow-activity classification collected in a real production herd. The data set, which has been widely used in related works, includes measurements from active (*Feeding*, *Rooting*, *Walking*) and passive (*Lying Laterally, Lying Sternally*) activities. When classifying 1-s length observations, the best method achieved an average recognition rate of 74.64%, for the five activities. When classifying 2-min length time series, the performance of the best model increased to 80%. This is an important improvement from the 64% average recognition rate for the same five activities obtained in previous work. The pattern classification approach was also evaluated in alternative scenarios, including distinguishing between active and passive categories, and a multiclass setting. In general, better results were obtained when using a tree-based logitboost classifier. This method proved to be very robust to noise in observations. Besides its higher performance, the suggested method is more flexible than previous approaches, since time series of any length can be analyzed.

1. Introduction

Automated monitoring of animal behavior enables oestrus, health disorders, and animal-welfare in general to be supervised on a large scale. It is therefore an important research area within livestock production. Recent research and development have targeted animal activity recognition, since the recognition of basic animal activities can help to detect and monitor important events such as oestrus, pregnancy or parturition. Data collected from sensors physically-attached to animals have been successfully used to classify the activities of individual animals when housed in groups (Cornou and Lundbye-Christensen, 2010; Firk et al., 2002; Umstatter et al., 2008). The main motivation behind physically-attached sensors is to gather real-time (first hand) information of the animals' behavior. In addition, sensors such as infrared and accelerometers are affordable and accurate, which make them suitable tools for commercial research. The activities of dairy cows, sows and other species have been monitored and classified using data collected from these types of sensor.

The present work focused on the classification of individual sows' activity using accelerometers measurements. Previous studies (Cornou and Lundbye-Christensen, 2008; Cornou and Lundbye-Christensen, 2010) used dynamic linear models to classify different sow activities. In particular, Cornou and Lundbye-Christensen (2010) used a Multi-Process Kalman Filter (MPKF) which achieved excellent classification results for passive (lying laterally, LL, and lying sternally, LS) and active (feeding, FE, rooting, RO, and walking, WA) activities. The authors reported a 64.4% average recognition rate. The current study aimed at improving the recognition performance obtained by the MPKF for active (FE RO WA) and passive (LL and LS) sow activities by classifying accelerometer data using a supervised machine learning process that neglects time dependencies between sample-measurements. Specifically, the classification of time series is approached as a standard

* Corresponding author.
E-mail address: hugojair@inaoep.mx (H.J. Escalante).

0168-1699/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.compag.2013.01.003

(atemporal) pattern classification task that can be solved with a variety of techniques (Duda et al., 2000; Hastie et al., 2009). In this way, time series of arbitrary length (duration) can be analyzed by combining the predictions provided by the model for the elements (acceleration measurements) of the series. This formulation offers a wide flexibility for the on-line monitoring of animals. Furthermore, it was hypothesized that four accelerometer measurements (axes x,y,z, and the norm of the acceleration vector) recorded at an instant (1 Hz in this study) are informative enough to discover and recognize sow activities.

Using the data set from Cornou and Lundbye-Christensen (2010), classification results are generated by applying six of the most representative classifiers from the fields of machine learning and pattern recognition (Duda et al., 2000; Hastie et al., 2009; Saffari and Guyon, 2006): neural networks (`neural`), support vector machines (`SVM`), Naïve Bayes (`naive`), a linear classifier (`zarbi`), random forest (`RF`) and logitboost-with-trees (`logitboost`). The performance of these classifiers is evaluated under different scenarios.

The main contributions of the study presented in this paper are as follows:

- A highly-effective supervised-learning approach to sow-activity classification where time dependencies between measurements are ignored.
- The proposed approach is able to classify measurements recorded at an instance (a second) of time, facilitating the real-time monitoring of animal-behavior in practice.
- In addition, a method combining predictions made at the observation (second) level for classifying time series of varying length is proposed.

The remainder of this paper is organized as follows: Section 2 describes the method used to obtain the accelerometer data of five types of sow activities. Section 3 follows with background information on pattern classification and leads into Section 4 where the supervised learning approach is presented. Section 5 reports the classification results obtained using the six classification methods for the five activity types. The classifiers are tested on an individual basis, per activity, and as a multiclass problem, across all activities. The classifiers are also tested under different data-input scenarios, using data samples at 1 Hz (an observation) and for time series of accelerometer data of 2 min (a series of observations). Section 6 concludes with the findings of this study and outlines future work directions.

2. Acceleration measurements

2.1. Time series recording

Time series of acceleration measurements were collected for 11 group-housed sows, in a Danish production herd. These experimental sows were housed in a dynamic group of approximately 100 sows, where the pen was 22.45 m long by 12.45 m wide. Resting areas were straw-bedded and activity areas had solid or slatted floors.

The accelerometers and a battery package was placed on a box fitted on each experimental sow using a neck collar, so that the box was placed on the lowest part, i.e. at the bottom of the neck, for each of the 11 sows. Acceleration data were measured in three dimensions using a digital accelerometer (LIS3L02DS from STMicroelectronics) four times per second, 24 h a day, during 20 days. Furthermore, the sows were video recorded 24 h a day. Video recordings were used to identify the types of activity that the experimental sows (individually marked on their back) were performing.

2.2. Data set construction

This study used the two data sets from Cornou and Lundbye-Christensen (2008), Cornou and Lundbye-Christensen (2010). Five types of activity were included: feeding (FE), rooting (RO), walking (WA), lying sternally (LS) and lying laterally (LL).

The data sets contain extracts (observations) of time series corresponding to each of the five activities. Each extract is a 4D vector of measurements, with values for the three-dimensional axes x, y and z and the length of the acceleration vector $a = \sqrt{x^2 + y^2 + z^2}$. A learning (or training) data set was used to train discriminative models and a test data set was used to evaluate the classification methods.

- The learning data set includes 46 series of 10 min: 6, 7, 11, 11 and 11 series, respectively for FE, RO, WA, LS and LL.
- The test data set includes 490 series of 2 min: 84, 79, 107, 110 and 110, respectively for FE, RO, WA, LS and LL.

Video recordings were used to select the series' extracts. The procedure was carried out by a single person, who simultaneously analyzed the video and noted the start and end of activities on the printed time series. Since a change of activity can be visualized on the series (for one or more axes), this ensured a good concordance between the activity and the series' extract. Any overlapping of activity (especially between RO and FE) was reduced to a minimum. Moreover, missing data and the fact that sows perform more rarely RO and FE activities resulted in a smaller number of series for these activities.

The two data sets differ in terms of time series' length. For the learning data set, a length of 10 min was chosen in order to have sufficient training time for the development of the classification method, and considered as a maximum length (especially with respect to FE), since the extract should contain an activity performed continuously. For the test data set, a length of 2 min was considered as sufficient to recognize a given activity, and short enough to reduce overlapping of different activities. The data set used in this study is described in more details in Cornou and Lundbye-Christensen (2010).

3. Pattern classification

A wide range of pattern classification methods have been developed (Duda et al., 2000; Hastie et al., 2009), as pattern classifiers are important core components within machine learning and pattern recognition systems. The supervised pattern-classification process involves finding a map between observations (inputs) and labels (outputs), given a set of data for which the correspondence between inputs (observations) and outputs (labeled data) are known. In this study, during the process of classifying sow activities, the four accelerometer measurements are considered to be the observations and the labels correspond to the different activities to be recognized. Classic pattern classification problems include: handwritten digit recognition, spam filtering and face recognition. In this work, observations are 4D accelerometer measurements and the labels correspond to the different activities to recognize (Section 4).

Let us consider a data set $\mathcal{D}$ formed by N pairs in the form $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbb{R}^d$ is an observation, d is the dimensionality of the observations, and $y_i \in C$ indicates the corresponding class label for $\mathbf{x}_i$, where $C = \{1, \ldots, K\}$ for a problem of K classes or labels associated to the problem at hand.

The classification problem for $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{1,\ldots,N}$ consists of finding a function f of the form $f : \mathbf{x}_i \in \mathbb{R}^d \rightarrow y_i \in C$, from the paired samples in $\mathcal{D}$. The learned function f must be able to classify

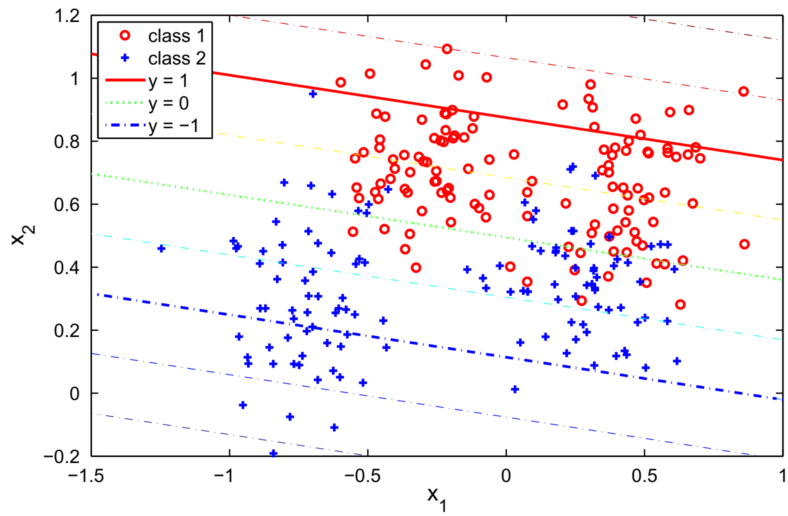

Fig. 1. A linear classifier for a two class problem (red circles versus blue crosses). The green (dotted) line, given by $(\mathbf{x}) = \mathbf{wx} + b = 0$, separates examples from both classes. Example generated with the CLOP toolbox (Saffari and Guyon, 2006). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

unseen observations $\mathbf{x}$ that are generated from the same distribution as $\mathbf{x}_{1,\ldots,N}$, with unknown associated labels.

For convenience, when performing binary classification (where $K = 2$) labels are usually represented by $y_i \in \{-1, 1\}$ instead of $y_i \in \{1, 2\}$. Under this setting the label associated to an unseen example $\mathbf{x}$ can be obtained by $y = sign(f(\mathbf{x}))$. Fig. 1 shows a synthetic binary classification problem for observations in $\mathbf{x}_i \in \mathbb{R}^2$ and the solution found using a linear classifier.

The form of f is defined by a learning algorithm. For example, f can take the form $f(\mathbf{x}) = \mathbf{wx} + b$, as in Fig. 1, where $\mathbf{w}$ is a vector of parameters of the model learned from $\mathcal{D}$. Classifiers using a function of this form are called linear classifiers (Duda et al., 2000). There are many other classification methods which use alternative function forms and learning algorithms, as for instance, neural networks, probabilistic classifiers (e.g., Naïve Bayes) and similarity based methods (e.g., $1-NN$). In this study, we consider the set of classification methods available in a machine learning toolbox called CLOP[1] (Section 5.1). A detailed description of each of these methods being out of the scope of this paper, we refer the reader to Duda et al. (2000) and Hastie et al. (2009) for a comprehensive review.

4. Supervised learning of sow activities

This section describes the supervised learning approach to sow-activity classification. It presents the process by which time series are preprocessed in order to obtain the training examples for building a pattern classification model. Then, the proposed approach to sow-activity classification is described, providing details on how time series of arbitrary length can be classified.

4.1. Representation of observations

The data considered in this study consist of 4D time series labeled with one out of the five activity types (introduced in Section 2). In order to develop a supervised learning method for sow-activity classification, data must be preprocessed, transforming labeled time series into fixed length patterns. A time series j of length M_j can be seen as matrix of dimensions $4 \times M_j$:

$$\mathbf{S}_j = \begin{pmatrix} x^j_{1,1} & x^j_{1,2} & \cdots & x^j_{1,M_j} \\ x^j_{2,1} & x^j_{2,2} & \cdots & x^j_{2,M_j} \\ x^j_{3,1} & x^j_{3,2} & \cdots & x^j_{3,M_j} \\ x^j_{4,1} & x^j_{4,2} & \cdots & x^j_{4,M_j} \end{pmatrix} \quad (1)$$

where $x_{a,b}$ is the bth measurement in dimension a for time series j. Each column of $\mathbf{S}_j$ is the 4D measurement at a given time. For each activity type k, all corresponding time series are combined, resulting in a large series of the form $\mathcal{S}_k = [\mathbf{S}_1, \ldots, \mathbf{S}_{N_k}]$, where N_k is the number of series associated to activity k. Since each $\mathbf{S}_j$ is a matrix, $\mathcal{S}_k$ becomes:

$$\mathcal{S}_k = \begin{pmatrix} x^1_{1,1} & x^1_{1,2} & \cdots & x^1_{1,M_1} & \cdots & x^{N_k}_{1,1} & x^{N_k}_{1,2} & \cdots & x^{N_k}_{1,M_{N_k}} \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ x^1_{4,1} & x^1_{4,2} & \cdots & x^1_{4,M_1} & \cdots & x^{N_k}_{4,1} & x^{N_k}_{4,2} & \cdots & x^{N_k}_{4,M_{N_k}} \end{pmatrix} \quad (2)$$

with 4 rows and $\sum_{n=1}^{N_k} M_n$ columns. Fig. 2 shows the combined time series ($\mathcal{S}_k$) for the five activities considered in this work, corresponding to the training set. As expected, passive activities (lying laterally and sternally) show small variations in the four dimensions. However, noisy measurements are noticeable for both activities. Most of this noise reflects where a new series starts, approximately each 10 min. This is due to the position of the sensor on the neck collar, which can slightly differ from sow to sow, or to the angle of the lying position. These noisy measurements complicate the construction of classification models at the observation level. It should be noticed, however, that some of the classification methods considered in this study (`logitboost`, `RF`, `neural`) are specifically designed to work with noisy and/or mislabeled data. The sows' active activities such as walking, feeding and rooting, show larger acceleration variations in the four dimensions, which makes the differences from series to series less noticeable.

Since all of the measurements in $\mathcal{S}_k$ belong to activity k, a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1,\ldots,N}$ of pairs of accelerometer-measurements ($\mathbf{x}_i \in \mathbb{R}^4$) and activity-type labels $y_i \in \{FE, RO, LL, WA, LS\}$ is generated as follows. For each activity type k, each column of $\mathcal{S}_k$ is an

[1] http://clopinet.com/CLOP/.

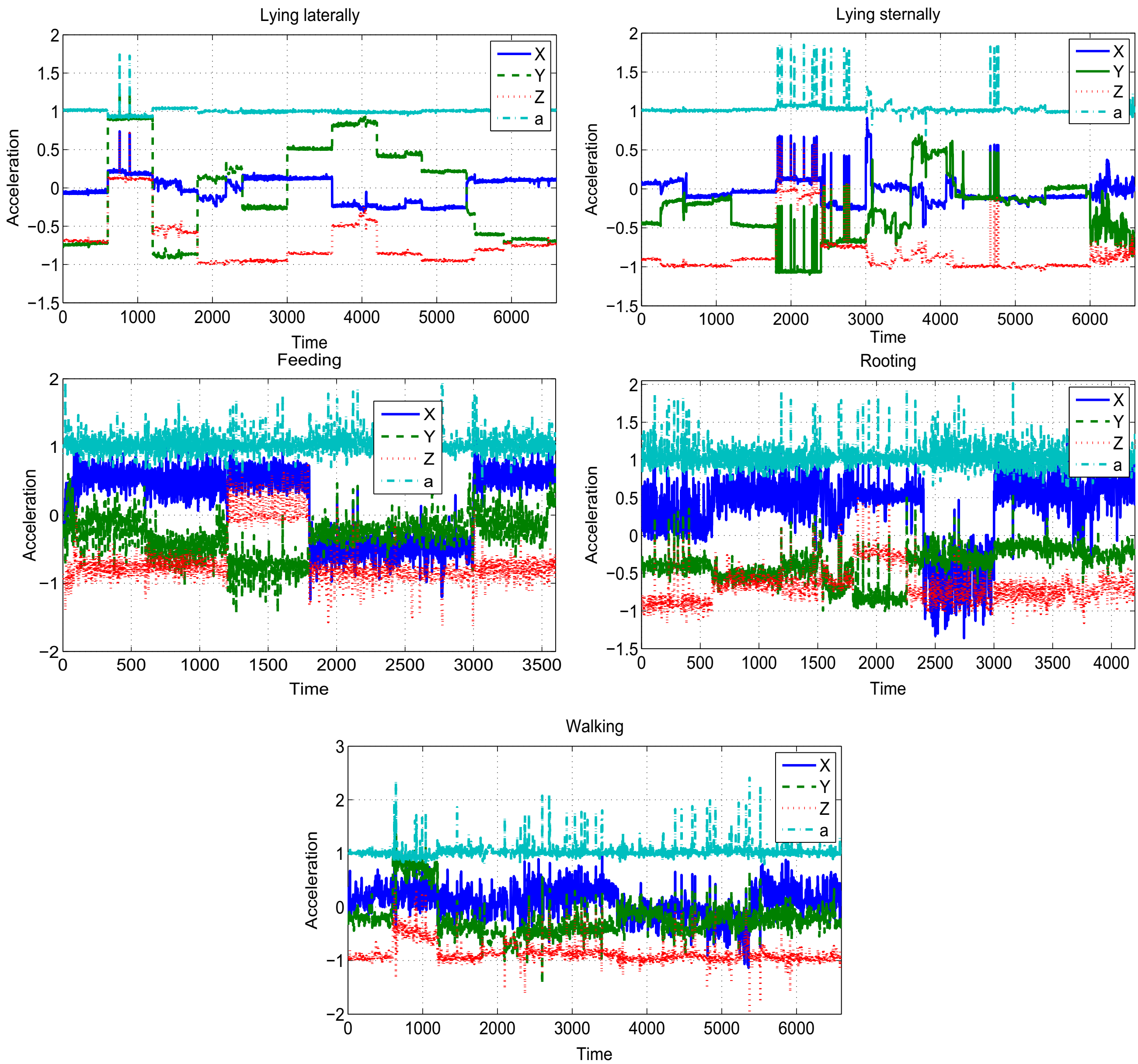

Fig. 2. Combined (training) time series ($\mathcal{S}_k$) of each of the five activities, k = {*FE*, *RO*, *LL*, *WA*, *LS*}, considered in this study. An observation is the vector values of acceleration measurements from the four considered axes measured at a given time t.

observation $\mathbf{x}_i$, with the corresponding label y_i, set manually by observing the activity of which S_k belongs to. Hence, $N = \sum_{k=1}^{K} M_{S_k}$, where M_{S_k} is the number of columns in the matrix $\mathcal{S}_k$. In machine learning, it is a convention that input data are represented by a matrix where each row is an observation and the columns are the different attributes associated with the instances. We obtain this standard representation by taking the transpose of each $\mathcal{S}_k$ and using its rows as the observations.

Learning and test data sets are represented as described above, with $\mathcal{D}$ denoting the generated learning/training and $\mathcal{T}$ the test data sets. Data sets $\mathcal{D}$ and $\mathcal{T}$ can be used with any learning algorithm in a standard supervised classification setting (Section 3). Data set $\mathcal{D}$ is used to train the classification model, that is, learning the parameters of the pattern classification function (e.g., $\mathbf{w}$ and b for the linear classifier from Fig. 1). The trained model is then used to make predictions for observations in $\mathcal{T}$ into each of the sow-activity classes. The performance of the classification model is evaluated by comparing the predictions made by the model with the true labels for observations in $\mathcal{T}$.

This representation can be used to develop classification methods at the observation level. A function f can be learned to map instantaneous measurements (i.e., of duration 1 s) into the considered sow-activity types. Since each observation can be labeled, series of arbitrary length can also be labeled by combining the predictions made for the series' observations (see Section 4.2).

For learning and evaluating the proposed approach a methodology similar to previous works on sow-activity classification (Cornou and Lundbye-Christensen, 2008; Cornou and Lundbye-Christensen, 2010) was adopted. For each activity type a function f_k is learned, where f is able to discriminate between observations of activity type k and the other activities $j{:}j \neq k$. Thus, there are k binary classification problems, one for each activity. In this setting, for each activity type and given an observation $\mathbf{x}$, the goal is to determine whether $\mathbf{x}$ corresponds to activity k or not. The performance of each classifier is therefore evaluated independently.

Additionally, the multiclass performance of the proposed approach, that is, the performance obtained when all of the k classifiers are run simultaneously is evaluated. Under this scenario,

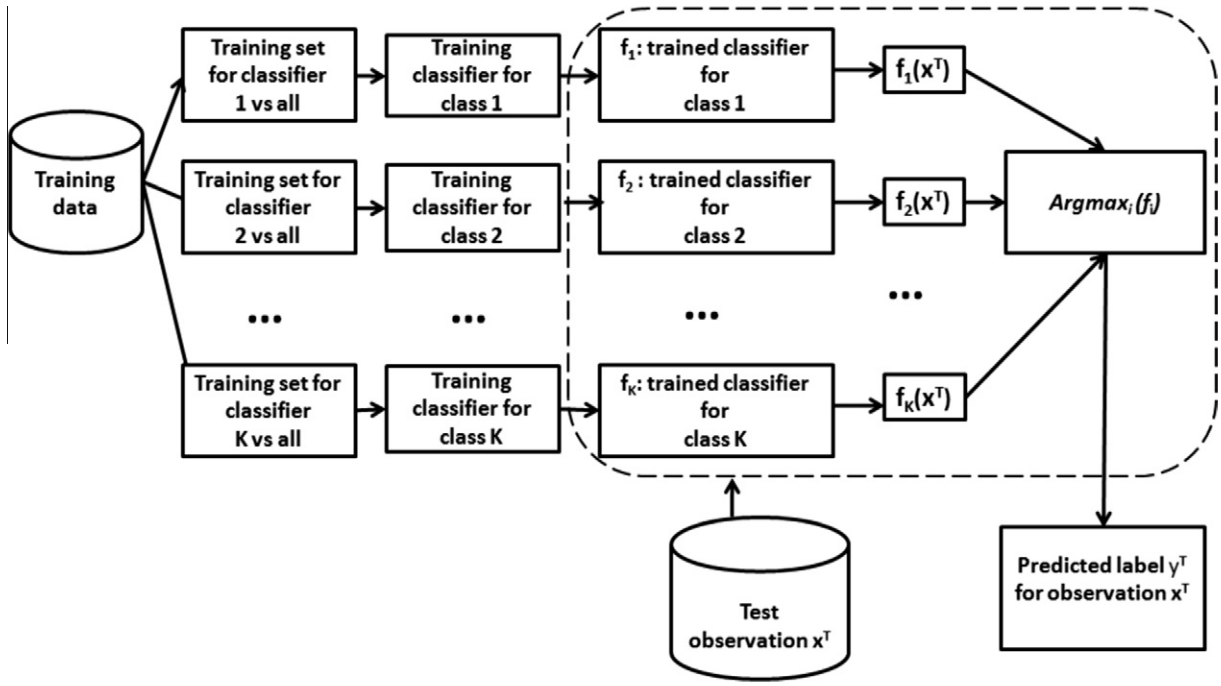

Fig. 3. The considered one-vs-all multiclass classification approach.

given an observation $\mathbf{x}$, the goal is to assign it to one of the k activity types, by using the learned classifiers $f_{1,\ldots,K}$. There are several ways to build multiclass classifiers. Here, one of the most successful and most used technique, the one-vs-all approach (Rifkin and Klautau, 2004), was considered. Under this approach, the outputs of the k binary models are combined to generate multiclass predictions, as illustrated in Fig. 3.

Let $\mathbf{x}^T \in \mathbb{R}^4$ denote an observation whose class is unknown. The vector is passed through the K classifiers and each binary classifier provides a confidence value $f_k(\mathbf{x}^T) \in [-1, 1]$ that indicates how confident classifier f_k is in recognizing the activity associated to observation $\mathbf{x}^T$ as the kth. The label assigned to $\mathbf{x}^T$ corresponds to the classifier that obtains the highest confidence: $y^T = \mathrm{argmax}_k(f_k(\mathbf{x}^T))$, see Rifkin and Klautau (2004) for more details.

4.2. Labeling time series

Trained classifiers f_k can make activity predictions for observations with duration of 1 s. This is advantageous as time series of arbitrary length can then be classified, whereas in previous works, methods were only evaluated at the series-level using series of fixed length (Cornou and Lundbye-Christensen, 2008; Cornou and Lundbye-Christensen, 2010). Besides this important advantage, the proposed approach can still make predictions for time series of any length greater than one observation, by combining the predictions made for the individual acceleration measurements of a given series.

When time series of arbitrary length (i.e., composed of more than a single observation) need to be classified, the outputs of the learned model have to be postprocessed. The following approach, based on ideas from set classification (Ning and Karypis, 2008), was adopted. Let $T = \{\mathbf{x}_1, \ldots, \mathbf{x}_{M_T}\}$ be a time series of length M_T and let $\{f_k(\mathbf{x}_1), \ldots, f_k(\mathbf{x}_{M_T})\}$ be the confidence values for class k as returned by classifier f_k for each observation of T. For each activity type k:

$$p_k(T) = \frac{1}{M_T}\sum_{i=1}^{M_T} f_k(\mathbf{x}_i) \tag{3}$$

is calculated. To discriminate between series of type k and any other series, $Y^T = sign(p_k(T))$ is used. In that case, series T belongs to activity k if and only if $Y^T \geqslant 0$. For the multiclass setting, (Section 4.1), series T is associated with the label that achieves the highest value of $p_k(T)$, i.e.:

$$\arg\max_k(p_k(T)). \tag{4}$$

The activity corresponding to the maximum $p_k(T)$ across the time series is determined and the corresponding label is used to label the time series.

5. Experimental results and discussion

This section describes the experiments designed to evaluate the performance of the proposed approach and their results. After describing the experimental setup, the performance of different classifiers for making predictions at observation level is reported. Then, the performance of the proposed approach for classifying time series of varying time length is presented.

5.1. Experimental setup

Some additional characteristics of the data set used in the study (from Cornou and Lundbye-Christensen (2010); see also Section 2) are summarized in Table 1. The data set is almost balanced in terms of observation numbers available for each activity type. However, the challenge is the low dimensionality of data (4D) and the noise in the data (Fig. 2).

The following classifiers available in the CLOP toolbox (Saffari and Guyon, 2006) were considered in the experimental study: neural network (`neural`), support vector machine classifier (`SVM`), Naïve Bayes classifier (`naive`), linear classifier (`zarbi`), random forest (`RF`) and logitboost with trees (`logitboost`).

5.2. Sow-activity classification of observations

This section presents the results obtained in the classification of acceleration measurements at the observation level (1 Hz). Table 2 shows the percentage of correct classifications at the observation

Table 1
Characteristics of the learning and test data sets. For each activity, the number of time series, their duration (in min) and the number of observations are indicated.

Activity	Learning			Test		
	# Series	Duration (min)	# Observ.	# Series	Duration (min)	# Observ.
Feeding (FE)	6	10	3600	84	2	10,080
Rooting (RO)	7	10	4200	79	2	9480
Walking (WA)	11	10	6600	110	2	13,200
Lying Laterally (LL)	11	10	6600	107	2	12,840
Lying Sternally (LS)	11	10	6600	110	2	13,200
Totals	46	460	27,600	490	980	58,800

level for each of the considered activities and for each of the considered methods. The results are obtained from the test data set.

The classification performance obtained from the different classifiers unveils the difficulty of the problem: most results shown in Table 2 are below 70% of correct classifications in the test data set. Although the performance of the considered classifiers is low, it should be noted that the classification methods were evaluated on an observation basis (≈60,000 observations). Moreover, the results obtained with methods `zarbi` and `logitboost` are comparable with those obtained (at the series-level) in previous work (Cornou and Lundbye-Christensen, 2010) (see also Table 4), even though the reported methods exploited time dependencies.

From Table 2, it can be seen that the best performing method is the `logitboost` classifier, followed by a linear method (`zarbi`). On average, the difference in performance between `logitboost` and the best alternative, `zarbi`, is of more than 9%, and the average difference in performance of `logitboost` over the other methods is significantly larger. The `naive` method outperformed `logitboost` in the LL activity, although it is clear from its performance in the other activities (close to zero) that this classifier always predicted the LL activity regardless of the observation. The `zarbi` classifier outperformed `logitboost` in WA and LS activities, making it a regular classification method. The performances of the widely used classifiers `SVM` and `neural` are rather poor regardless of the activity type.

Results indicate furthermore that, in general, the passive activities (LL and LS) are more difficult to classify for the considered methods, as compared to the active ones (FE, RO, and WA). This was also reported in previous works (Cornou and Lundbye-Christensen, 2008; Cornou and Lundbye-Christensen, 2010) and may be due to the fact that time series for passive activities contain noisy measurements due to the eventual movements of sows. Besides, as previously mentioned, the concatenation of time series of different sows (and time) may be the main source of noise into the combined series, due to slightly different angles of lying positions and slightly different positions of the sensors around the neck collars (Fig. 2).

The area under the ROC curve (AUC) obtained by the considered classifiers on test set observations was also calculated. The ROC curve is a plot of the true positives rate versus the true negative rate, by varying the values of the classification threshold (Fawcett, 2006) (Fig. 4). The AUC summarizes with a real number in $[0,1]$ the performance of a classifier: the closer to 1, the better is the performance of the method. AUC is one of the most used evaluation measures in machine learning and pattern recognition and its main advantage is that it is independent of the classification thresholds of classifiers (Fawcett, 2006). Therefore, the integrity of the confidence values provided by the binary classifiers $f_{1,\dots,K}$ for each activity type can be evaluated. This is particularly important for the current study as the outputs of classifiers are combined to make predictions at the series-level.

Table 3 shows the AUC performance obtained by the considered classifiers on the observations from the test data set. It can be seen that `logitboost` outperforms the other methods in all of the activities. Here, `zarbi` obtains the worst average performance, while both `RF` and `neural` obtained competitive performances. Fig. 4 shows the ROC curves for the two best performing models: `logitboost` and `RF`. ROC curves for `logitboost` largely outperform those from `RF` across the five considered activities.

Results from Tables 2 and 3, and from Fig. 4 suggest that the best option for building a series-level sow-activity predictor is the `logitboost` classifier.

5.3. Sow-activity classification of time series

This section reports the performance of the considered methods in the classification of sow activity in time series. The 490 time series of the test set (average duration of 2 min, i.e. 120 observations) were used. For classifying a time series, the predictions made by classifiers for the individual observations of a given time series are combined (Section 4.2).

The classification was done using $sign(p_k(T))$ (Eq. (3)). Table 4 shows the percentage of correct classifications at the series-level for each activity type obtained by each of the six classification methods. For comparison, the results obtained by Cornou and Lundbye-Christensen (2010) are provided under the column heading reference in Table 4.

From Table 4, it can be seen that `logitboost` obtains the best average classification performance at the series-level, along with `zarbi` which is also highly effective. Both classifiers, `logitboost` and `zarbi`, outperform significantly the average performance obtained by the multi-process Kalman filter (MPKF) used in Cornou and Lundbye-Christensen (2010). The MPKF method obtained better classification performance than both `logitboost` and `zarbi` for the LL activity. However, for activities RO and WA, the classifiers `zarbi` and `logitboost` significantly outperform the performance of the MPKF approach. These results suggest that the supervised learning approach to sow-activity classification is an effective solution for this task, even though this approach does no consider time dependencies between observations.

Results obtained at the observational level (see Table 2) and series level (see Table 4) indicate that the `logitboost` method is the preferable approach to adopt in a sow-activity classification system. `logitboost` clearly outperforms the MPKF approach and most of the other classification methods. `zarbi` may also be

Table 2
Percentage of correct classifications at the observation level obtained with the considered classifiers in the test data set. The best result for each activity is shown in bold.

Activity	Neural	SVM	Naive	RF	Zarbi	Logitboost
Feeding	32.49	0.33	0.00	44.70	62.27	**90.79**
Rooting	27.62	0.92	0.06	31.96	60.97	**66.54**
Walking	26.92	0.07	0.00	70.62	**90.24**	84.63
Lying laterally	46.29	8.67	**99.96**	8.49	36.44	64.50
Lying sternally	34.52	0.01	0.00	28.78	**78.30**	66.75
Average	33.57	2.00	20.00	36.91	65.64	**74.64**

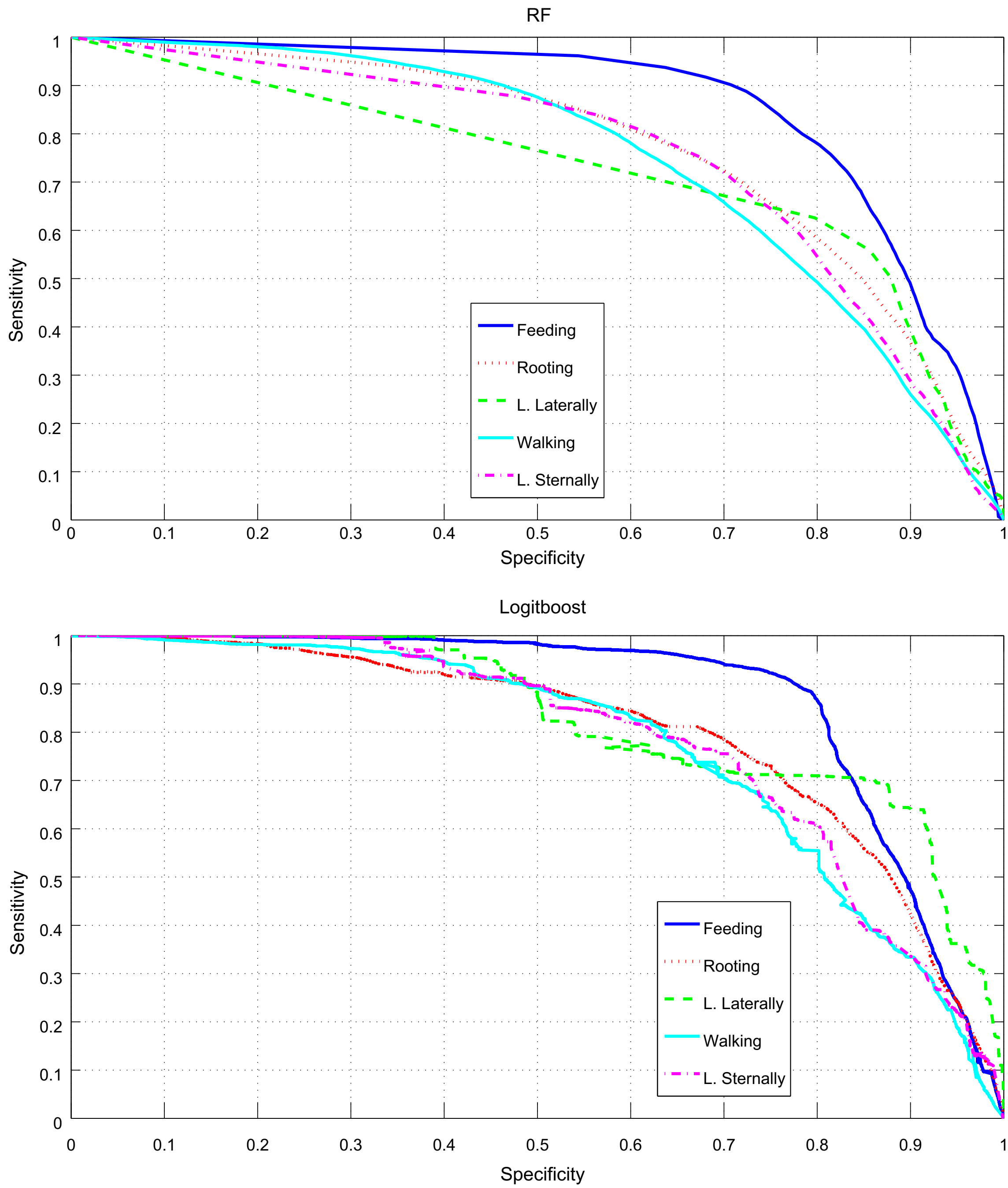

Fig. 4. ROC curves for the `RF` (top) and `logitboost` (bottom) classifiers for the five considered activities.

Table 3
AUC performance at the observation level (every second) obtained with the considered classifiers in the test set. The best result for each activity is shown in bold.

Activity	Neural	SVM	Naive	RF	Zarbi	Logitboost
Feeding	0.8469	0.6156	0.6096	0.8457	0.6010	**0.8719**
Rooting	0.7655	0.6978	0.6728	0.7589	0.6633	**0.7986**
Walking	0.7494	0.7339	0.7342	0.7480	0.7278	**0.7693**
Lying laterally	0.7136	0.7131	0.5324	0.7311	0.4773	**0.8261**
Lying sternally	0.7271	0.6518	0.6712	0.7845	0.6547	**0.7866**
Average	0.7605	0.6824	0.6440	0.7737	0.6248	**0.8105**

Table 4
Percentage of correct classifications at the series-level (2 min) obtained with the considered classifiers in the test set. The best result for each activity is shown in bold.

Activity	Neural	SVM	Naive	RF	Zarbi	Logitboost	Reference
Feeding	42	0	0	47	65	**100**	79
Rooting	22	0	0	22	65	**78**	56
Walking	17	0	0	93	**97**	91	74
Lying laterally	45	9	**100**	7	36	65	83
Lying Sternally	35	0	0	25	**81**	65	30
Average	32	2	20	39	69	**80**	64

considered another option as it performed well at both the observational and series level. However, `logitboost` showed a more regular performance across activities than `zarbi` at both observation and series-level; besides `logitboost` outperforms significantly `zarbi` in terms of AUC performance (see Table 3).

5.4. Classifying between passive and active categories

In Cornou and Lundbye-Christensen (2010), classification results for the five activity types were combined into passive (LL, LS) and active (FE, RO, WA) categories. This approach was further elaborated in this study, using all of the considered classification methods. In the supervised learning context, this is a standard binary classification task. Hence, observations labeled with activities LS and LL were assigned the passive label (i.e., 1, the positive class) and activities FE, RO and WA, were labeled as active (i.e., −1, the negative class). Then the considered classifiers were used for building *f*. The predictions made for the observations were combined to label the times series as described in Section 4.2. Table 5 shows the results obtained with the considered methods for active versus passive activity classification at the series-level.

Results indicate that, on average, `logitboost` outperforms the other methods, although the difference between `logitboost` and the `neural` method is small. The majority of the classifiers had difficulty in correctly labeling the passive category at a times series level, as compared to the active category. This is likely due to the fact that, in general, little variation was observed in time series from passive activities in the four dimensional measurements (Fig. 2), which is an expected pattern since the sow is almost not moving. However, the collected data for passive activities contains few observations with high variations as well, mainly for the LS activity. It is hypothesized that such variations are incorrectly classified as active behaviors at the observation level, and that these errors are reflected when classifying the series.

Fig. 5 shows the predictions made by the `logitboost` classifier for each of the 490 test-set time series in the active versus passive activity classification problem. The outputs were normalized to the interval $[0, 1]$. Hence, the classification threshold is set as $y = 0.5$. Errors made by the model are shown as red squares. It can be seen that most errors are made for observations belonging to activities LS, LL and WA. The errors obtained for WA activity can be a result of the small variation observed both for the length of the vector and for the *z*-axis for this activity. This pattern may be confused by the classifiers with the corresponding small variations of all axes for activities LL and LS. Nevertheless, most FE and RO time series were correctly classified.

Columns 7 and 8 of Table 5 shows that the performance obtained with `logitboost` (94.46% and 85.00% for active and passive activities, respectively) is lower than what was reported in previous work (96.00% and 94.00%, correspondingly). However, it should be noticed that in the previous study (Cornou and Lundbye-Christensen, 2010), one model (MPKF) was trained and applied for each of the five activities, of which results were later on combined, and not for each category (active versus passive). In this study, a single binary classifier was trained for each category, with each of the considered methods. Hence, the results are not directly comparable.

5.5. Multiclass performance of the classifiers

Previous tables reported results of per-activity classification performance. The reported evaluation measures indicated the performance of each model in discriminating between each activity and the rest. These evaluation measures assess the performance of the per-activity models and not the performance when all models are evaluated simultaneously. The latter scenario being more realistic, the multiclass performance of the proposed approach was evaluated. Eq. (4) was used to obtain the multiclass predictions from the output of the binary classifiers.

Table 5
Percentage of correct classifications for the active versus passive activity classification task at the series-level (2 min).

Activity	Neural	SVM	Naive	RF	Zarbi	Logitboost	Reference
Active	90.03	74.91	0.00	100.00	68.26	94.46	96.00
Passive	80.45	63.18	100.00	35.00	61.36	85.00	94.00
Average	85.24	69.05	50.00	67.50	64.81	89.73	95.00

Table 6 shows the percentage of correct classifications in a multiclass setting (i.e., the percentage of series that were labeled correctly by the whole multiclass classifier) for the considered methods. The best classifier (`logitboost`) classified around 51% of the ≈60,000 observations and above 60% of the times series. This indicates that one out of two observations was correctly labeled and associated with one of the five activity classes. While this result may seem low, this should be set in perspective with random predictions, which would result in an accuracy of 20%. Since the results reported herein are, for the considered data set, the best reported so far, results from Table 6 indicate that the sow-activity classification is an open problem with large room for improvement.

Table 7 shows the confusion matrix for the `logitboost` classifier for the time series experiment. It can be seen that WA is the less misclassified activity, whereas LS is the most confused. FE and RO are frequently confused. As expected, both passive activity types (LS and LL) are frequently confused with each other.

5.6. Classifying series of variable lengths

In a final experiment the series-level performance of the supervised learning approach to sow-activity classification was assessed using time series of variable lengths, applying the `logitboost` classifier only. Fig. 6 shows the performances of both the passive versus active categories and the multiclass classification for times series of different observation numbers. For the passive versus active task, the accuracy is reported as the percentage of time series correctly labeled, for both passive and active categories.

This figure shows the flexility and robustness of the suggested method. `logitboost` achieves a similar performance for any given series' length. This indicates stable predictions regardless of the number of observations in the series. Therefore, the proposed approach can be used in a non-specific time setting to make predictions for the activities performed by each sow. This is an advantage when the goal is to analyze individual sow's behavior in (near) real time.

5.7. Logitboost for sow-activity classification

Experimental results indicate a good performance of the proposed approach for sow-activity classification. This approach appeared more effective than previous works and is flexible in terms of the length of time series. It can be argued that the `logitboost` classifier is the best of the considered methods for a sow-activity classification system, although some of the other methods showed acceptable performance as well.

The `logitboost` classification technique combines outputs from multiple individual methods, called weak learners (i.e. an ensemble method) – the usual weak learners being decision trees (Lutz, 2006). These methods have proved to reduce overfitting and are very robust to noisy training data (Hastie et al., 2009) and `logitboost` is a method from the boosting family. These methods build many individual classifiers iteratively and each of the classifiers is associated to a weight proportional to its ability to classify training instances. In each iteration *l*, a single classifier $h_l(\mathbf{x})$ is built, taking into account the weights associated with the instances. The weights are assigned to classifiers, and instances are updated in each iteration of the algorithm. When an observation $\mathbf{x}$ needs to be classified, it is passed through all of the

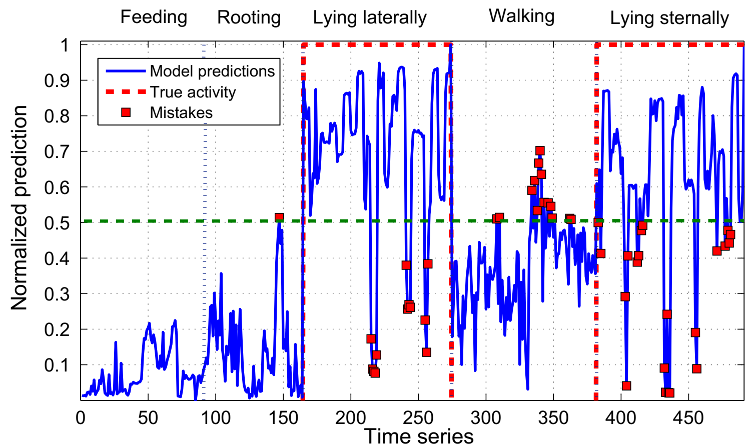

Fig. 5. Predictions made by the `logitboost` classifier for the active versus passive classification problem. The predictions of the model are scaled to the interval [0,1]. The horizontal line at $y = 0.5$ is the classification threshold (i.e., time series with a confidence value above 0.5 were labeled as passive and active otherwise). Mistakes made by the model are shown as red squares. The labels above the plot show the particular activity (FE, RO, LL, WA or LS) activity being performed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
Percentage of correct classifications for sow-activity classification under the multi-class setting.

Level	Neural	SVM	Naive	RF	Zarbi	Logitboost
Observations	45.14	37.38	22.38	40.44	35.15	51.86
Series	52.14	19.35	22.40	48.88	37.27	61.10

Table 7
Confusion matrix for the predictions made with the `logitboost` classifier. Diagonal (bold) values indicate the percentage of correctly recognized time series for each activity type.

True predicted	Feeding	Rooting	Walking	L. laterally	L. sternally
Feeding	**75.29**	24.71	0.00	0.00	0.00
Rooting	24.05	**46.84**	29.11	0.00	0.00
Walking	0.93	0.00	**92.52**	0.93	5.62
L. laterally	6.36	3.64	9.09	**60.00**	20.91
L. sternally	6.36	0.00	46.36	16.36	**30.92**

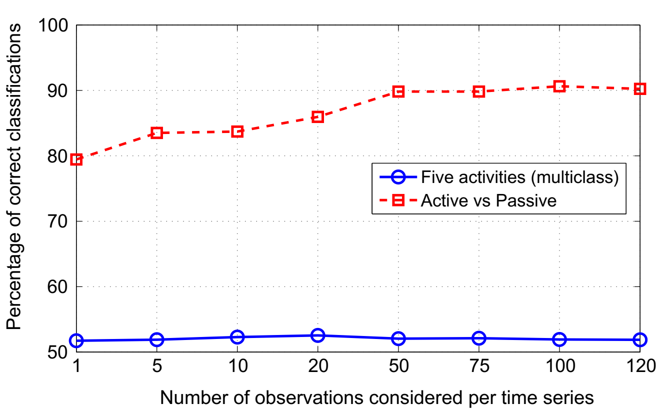

Fig. 6. Performance of the `logitboost` classifier when varying the number of observations used for labeling time series.

individual methods and a weighted voting strategy is used to assign a label to the unseen instance. Hence, $f(\mathbf{x}) = \sum_{l=1}^{L} z_l \times h_l(\mathbf{x})$, where z_l is the weight associated with classifier l, and $y = sign(f(\mathbf{x}))$. The particularity of `logitboost` lies in that each individual model is a (regression) decision tree and the logit transform is used internally by the algorithm to transform the outputs of individual models into probabilities. Details of the particular implementation of the `logitboost` considered in this work are found in Lutz (2006) and Saffari and Guyon (2006).

The nature of `logitboost` and the characteristics of the considered data set have clear synergetic properties. On the one hand, the data set contains potentially mislabeled data due to the fact that observations were generated by labeling time series of considerable duration (see Figs. 2 and 5). It is here suggested that `logitboost` is able to ignore, to some extent, the contribution of such mislabeled instances when building the decision function f. This is achieved automatically by assigning low weights to classifiers that misclassify noisy instances. Besides, models that are unable to classify, accurately enough (i.e. better than random guessing), a subset of training instances are disregarded for the model. Hence, only regular classification models are considered by `logitboost`. On the other hand, even when noisy observations are used to build individual classifiers, the fact that the `logitboost` classifier is built by many individual models (approximately 10,000 in the considered setting) reduces the probability that bad performing models dominate the predictions of the ensemble method. Thus, noisy instances and the impact of bad classifiers into `logitboost` are diminished automatically by the model. Training a `logitboost` binary classifier using ≈60,000 observations takes approximately 1 h on a Pentium Core i3 processor and 4 GB in RAM. While this time is not negligible, once the model is trained it is able to make predictions efficiently.

6. Conclusions

A supervised learning approach to the problem of sow-activity classification was suggested. Under the proposed formulation, pairs of accelerometer measurements and activity types are considered as labeled instances of a usual supervised classification task. In this setting, sow-activity classification can be approached with standard machine learning methods for pattern classification. Individual predictions for elements of a series are combined to classify it as a whole, on time series that can have arbitrary length. An extensive comparison of different learning algorithms for facing the classification task was performed.

Experimental results are reported for a sow-activity classification data set used in previous studies. The results show that some classifiers can achieve competitive performance, even when no temporal information is incorporated in the classification methods.

When classifying observations (1 s) the best method obtained performance as high as 75% on average for the five types of activities considered. When classifying time series of 2 min length, the classification performance increased to 80% (on average) for the five activities. This was a marked improvement from the 64% classification performance for the five activities reported in Cornou and Lundbye-Christensen (2010). A very good performance was also obtained when distinguishing between active (FE, RO, WA) and passive (LS, LL) categories. The highest average performance under this setting approached 90%, which is lower than the best previous-work's results (95%). However, the proposed approach builds a single classification model, whereas the reference method required five models (one per activity type). The multiclass performance obtained by the suggested approach was also evaluated and the best result in this configuration was 61%. Besides being very effective, the proposed formulation is highly flexible, as time series of arbitrary length can be classified. The `logitboost` classifier obtained the best results across the different settings. This can be due to the fact that this method can overcome the influence of noisy observations which are present in the considered data set.

Future work includes the incorporation of time dependencies between observations into a boosting classifier in order to further improve the performance of the proposed formulation. Finally, a hierarchical classification approach where, for example, observations are first labeled as active or passive, and then a second level classifier can determine the particular class for an observation, is also a promising research direction.

References

Cornou, C., Lundbye-Christensen, S., 2008. Classifying sows' activity types from acceleration patterns: an application of the multi-process Kalman filter. Applied Animal Behavior Science 111, 262–273.

Cornou, C., Lundbye-Christensen, S., 2010. Classification of sows' activity types from acceleration patterns using univariate and multivariate models. Computers and Electronics in Agriculture 72, 53–60.

Duda, R.O., Hart, P., Stork, D., 2000. Pattern Classification. Wiley-Interscience.

Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874.

Firk, R., Stamer, E., Junge, W., Krieter, J., 2002. Automation of oestrus detection in dairy cows: a review. Livestock Production Science 75, 219–232.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Lutz, R.W., 2006. Logitboost with trees applied to the WCCI 2006 performance prediction challenge datasets. In: Proceedings of the 19th International Joint Conference on Neural Networks. IEEE, Vancouver, BC, Canada, pp. 1657–1660.

Ning, X., Karypis, G., 2008. The set classification problem and solution methods. In: Proceedings of the 2008 IEEE International Conference on Data Mining Workshops. IEEE Computer Society, Washington, DC, USA, pp. 720–729.

Rifkin, R., Klautau, A., 2004. In defense of one-vs-all classification. Journal of Machine Learning Research 5, 101–141.

Saffari, A., Guyon, I., 2006. Quick Start Guide for CLOP. Technical Report. Institute for Computer Graphics and Vision. Graz University of Technology. Graz, Austria.

Umstatter, C., Waterhouse, A., Holland, J., 2008. An automated sensor-based method of simple behavioural classification of sheep in extensive systems. Computers and Electronics in Agriculture 64, 19–24.