Exploring the applicability of Reservoir methods for Classifying Punctual Sports Activities Using On-body Sensors

Doug P. Hunt$^1$  Dave Parry$^1$  Stefan Schliebs$^1$

$^1$ School of Computing and Mathematical Sciences, AUT University, Private Bag 92006, Auckland, New Zealand 1142, Email: dphunt, dparry, sschlieb@aut.ac.nz

Abstract

This paper explores the use of a reservoir computing (RC) method, Echo State Networks (ESN) to classify inertial sensor motion data collected from sensors worn by horse riders into punctual activities of interest within the context of an adapted movement environment. RC methods incorporate both temporal and spatial aspects within the model and therefore may have applicability classifying signals with the varying temporal signatures often seen across activity instances even when performed by the same subject. ESN’s, as one of a number of RC methods has a potential advantage in this case of being able to directly incorporate the inertial data into the reservoir without the need to segment this data into sliding windows. This is part of a wider set of work to build a wearable coach for technique feedback for Equestrian sport. Our use of RC methods on inertial data to classify punctual human activities is novel.

Keywords: Echo State Networks, Reservoir Computing, Spatio-temporal data processing, Punctual Activity Classification, Activity Classification, Machine Learning, Equestrian sport

1 Introduction

Human activity recognition using on-body sensors presents a number of challenges (Avci et al. 2010). In many situations this includes a lack of knowledge of overall context such that while it may be possible to distinguish a gesture such as pronating the wrist, it is difficult, without some idea of overall context, to conclude reliably that the gesture is associated with, for example, opening a door by twisting the door handle or turning a car key in a vehicle ignition. One method used by researchers to resolve this dilemma is to embed the sensor within some other object such as a car key, pen or baseball bat (Verplaetse 1996) that has a particular use that constrains the context. Another method, using a more generalised sensor, is to assume or predicate (Lukowicz et al. 2004, Ward, Lukowicz, Troster & Starner 2006) a particular context or domain so that the choice of meaning of the gesture is constrained by the predefined domain. This is our own approach and in our case our predicated domain is that of Equestrian sport. Sporting domains have some additional benefits as a result of often being strongly defined by rules and traditions.

Another challenge in human activity recognition is the variability in both the spacial and temporal aspects of a particular action both across subjects and even within a single subject who performs an activity (or action) more than once (Ward, Lukowicz & Troster 2006). Some obvious examples of spatial variability includes differences between lefthanded and right-handed activities and differences in technique such as between the drive shot of a professional golfer and an amateur. Examples of obvious temporal differences include taking 75 seconds to mount your horse because the horse is moving away from you or mounting in 7 seconds (or less) as the horse is standing still.

Human activities may be broadly distinguished into durative or punctual activities. Durative activities usually occur over a longer period of time and have some sort of repeating rhythmic component. Some simple examples of durative activities include running, walking, rowing, grooming a horse and cycling. Punctual activities tend to be shorter and happen once rather than multiple times and so often do not have a repeating rhythmic component to the signal. Some simple examples of punctual activities include bowling a ball in cricket, hitting a ball in baseball, a backhand shot in tennis and mounting a horse. Most of the activity recognition literature looks at durative activities. This work looks at punctual activities using current state-of-the-art temporal pattern recognition methods.

This paper is divided into 5 main parts. In section 2 we describe the application domain including our goal so that other practitioners can understand our motivation and assess the relevancy of this work. In addition we define the activities to be classified, with an illustration and identify the entities involved with our goal scenario. In section 3 we describe the experimental set up including our approach, the equipment used, the participant, the activity setting/scenario, the data obtained and its main characteristics/meta-data such as sample rate and precision, the preprocessing applied to the data prior to classification and a description of the classification engine. In section 4 we present the results obtained and in section 5 we discuss our conclusions and future work.

In reporting this work we have followed the recommendations in “How to do good research in activity recognition” (Plötz 2010).

2 Application Domain

The goal of our research is to construct an Activity Classification System (ACS) to classify activities of interest within Equestrian sport. Data input into the
system is collected from horse riders using wearable inertial sensors. The ACS is envisaged to be part of a wider, wearable coaching system that would be designed to provide horse riders with postural and riding feedback as they train. We envisage such a system as having a small number of generalised sensors on the rider’s body at strategic positions that are used for multiple purposes.

In this paper and planned follow on work we have set out to classify the activities Mounting and voluntary Dismounting. Mounting is when the rider gets on the horse and dismounting is when they get off the horse. At this stage we are only seeking to classify voluntary dismounts and we exclude involuntary dismounts or falls.

As with all activities, identifying the precise beginning and ending of a mount/dismount is not trivial and so a definition of the activity with a recognisable start and end is required. For this work, our activity definitions are:

**Mount** A stirrup mount from the time when a rider with one leg in the stirrup, lifts the second leg off the ground or mounting block in order to mount until the time when they are seated in the saddle.

**Dismount** The time from when a rider leans forward (prior to dismounting) until they are standing on the ground.

Neither definition is inclusive of all possible classes of either mounting or dismounting but we consider them sufficient to cover a reasonable percentage of the mounts and dismounts that are likely to be encountered in equestrian sport. As far as we are aware, there are no other generally accepted definitions for these activities, for the purposes of activity classification.

The entities involved are the horse rider and the horse. In this case the horse is a built for purpose wooden model that allows multiple mounts and dismounts in a laboratory setting without having to deal with horse welfare or safety issues.

### 3 Experimental Set-up

#### 3.1 Approach

Mounts and dismounts are punctual rather than durative activities. Punctual activities are short, simple activities such as opening a drawer, sitting down, picking up a cup, bowling a ball in cricket, hitting a ball in baseball, a backhand shot in tennis or shooting a hoop in basketball. Durative activities are of longer duration and usually have a repeated element such that there is a rhythmic or cyclic nature to the activity such as walking, running, rowing, cycling or grooming a horse. Most of the ACS research for wearable sensors that has successfully moved out of a scripted activity, laboratory environment into the real world has either been based on classifying durative activities or has made use of additional non-inertial sensors such as cameras and sound in addition to inertial sensors to reliably classify activity.

Much of the successful work classifying durative activities has used sliding window techniques to break the raw input into fixed blocks or windows and then calculates a number of signal derivatives for each window. Those window derivatives are then used as inputs to the ACS. Example activities include walking, running and being still. These being successfully classified based on windowed properties of the sensor signal such as signal standard deviation (Lee & Mase 2002, Lester et al. 2005), spectral frequency (Lukowicz et al. 2004, Lester et al. 2005), spectral entropy (Bao & Intille 2004, Ermes et al. 2008, Lester et al. 2005), integrals, means and variances (Lester et al. 2005). Sliding window techniques have had some success with scripted punctual activities but have been markedly less successful in real world situations.
In this work we use a Reservoir Computing (RC) technique, called Echo State Networks (ESN) to directly classify the pre-processed input data without the need to window this data.

3.2 Data Collection

Data was recorded from two laboratory and 55 real life riding sessions from 20 participants (and their horses) over a three month period using a commercial available, six degrees of freedom inertial sensor from SparkFun (SparkFun Electronics Inc 2008). Data collection was done as part of a Masters project (Hunt 2009) and all data was collected within Sweden. Each session was videoed so that activities could be manually classified by participants, riding domain experts and the research team.

The data for this set of experiments was taken from one of the two laboratory sessions and during this session the participant, who was an experienced rider, wore the sensor on her right wrist using a simple stretchable Velcro bandage for attachment. The participant self-described herself as right-handed. The “horse” used during the laboratory sessions was a built-for-purpose wooden framed horse (Diana) of approximately 16 hands in height (163cm at the “shoulder”), draped with a standard European riding saddle and stirrups. During this particular laboratory session the participant mounted and dismounted 17 times following a proscribed script. During the laboratory sessions the video camera was fixed into position using clamps so that the researcher was free to move around if needed.

3.2.1 Sensor

The SparkFun 6DoF inertial sensor contains a Freescale MMA7260Q triple-axis accelerometer, two InvenSense IDG300 500 o per second gyroscopes and both a Honeywell HMC1052L and a HMC1051Z magnetic sensor.

The sensor outputs readings from a 12 bit analogue to digital converter that gives a reading range between 0 and 1023.

The sensor outputs are:
1. a sample start character “A”
2. an unsigned 15 bit serialised sample number
3. three axis of magnetic readings
4. three axis of accelerometer readings
5. pitch, roll and yaw readings
6. a sample stop character “F”

Giving a total of 12 data fields per sample. For example:

A,0,569,498,504,577,344,576,467,446,462,F
A,1,569,493,503,567,342,571,466,458,464,F
A,2,569,496,504,0,340,1023,465,456,464,F

Sensor readings were sampled at 10Hz and broadcast via Bluetooth to an on-body receiver for logging and later analysis. The accelerometer in this sensor has a settable scale and for this session it was set to record ±2G’s.

3.2.2 Session Script

The script asked the participant to start and finish each mount/dismount pair at the same spot in the laboratory, within three metres of the wooden horse but clear of all obstacles. Prior to each mount/dismount pair the participant clapped her hands over her head two times as a synchronisation signal (to enable the inertial data to be synchronised with the video). After each set of claps and upon mounting the participant was asked to pause for approximately five seconds by standing or sitting still.

3.3 Data Description

The participant was asked to mount and dismount as closely as possible to her normal technique and apart from the requested pauses was not asked to keep to any particular time schedule. The duration of each mount and dismount is, never-the-less, reasonably consistent with a significant but gradual shortening of duration as the participant gets used to and more proficient at the script. The first mount takes 11 seconds while the last mount takes 6.5 seconds with progressive shortening in between. The first dismount takes 9 seconds and the last dismount takes 6.7 seconds again with progressive shortening. There is also a small shortening of the interval between each mount/dismount pair with the first interval being 11.3 seconds and the last being 7.2 seconds. However there is also a longer interval after the first two pairs.

The interval between each mount and dismount (when the participant is sitting on the horse) is much more consistent (probably as a result of the script). The first and second mount to dismount intervals are both 4 seconds and the last mount to dismount interval is 3.4 seconds.

The periods prior to the mount/dismount series records the sensor on a bench after being turned on, being fitted to the participant’s wrist and then the participant waiting around while the researcher checked and adjusted equipment such as the video camera. The period after the series is essentially the participant taking off the sensor and it being placed back on a bench.

Figure 2 depicts the recorded time series after pre-processing and manual labelling. The three panels each show the 3-dimensional recordings from the accelerometer, gyroscope and magnetometer respectively. The alternating background stripes show the activity undertaken during recording (mounting, dismounting and null class). The figure depicts all 17 mounts and dismounts.

From figure 2, careful observation shows that there is a slight drift upwards over time with the Gyroscope data. This drift is relatively common with lower priced Gyroscopes and this tendency has been ignored in our data analysis. In addition, the y axis of the Magnetometer data also drifts upwards over time, while the x and z axis do not show the same level of drift. The presence of the drift on only a single axis is possibly due to the sensor moving slightly on the participant’s wrist as they do things. In addition, some level of drift is common with consumer level Magnetometers such as this one but normally the drift would be present on all axis. Both instances of signal drift have been ignored in this work.

3.4 Preprocessing and Labelling

3.4.1 Editing & Error Checking

The raw data files are hand edited to remove erroneous set up data and commands that were logged before and after the main data file. Column labels are inserted as the first line of the cleaned file. An error checking routine is then run against the file to check for missing and out of range data. This routine
also strips off the start and end characters (A & F) and changes the end-of-line character sequence. The raw files have an unusual LF-CR end-of-line (eol) sequence and this is replaced with CR-LF. Once the 15 bit sample number has been checked for missing or duplicate samples it is replaced by an unsigned integer that does not overflow. No out of range, invalid or missing data was found in this file.

3.4.2 Synchronisation & Labelling

The overhead hand clap leaves a distinctive signal in the sensor data and provides an obvious counterpart in the video. The sensor data is searched visually to identify the first two sets of overhead hand claps and the time between the sets of claps is measured. The video is then run and the first two sets of hand claps are identified. The time between clap sets is checked to ensure the same claps are being used. The sensor data and video are then synchronised based on a detailed visual evaluation of the first set of hand claps using both the sensor signal and a frame by frame view of the video. Once synchronisation is established it is checked against later hand claps to ensure correctness and negligible temporal drift.

Once synchronisation is complete the video is used to find the frame where the occurrence of each activity starts and finishes. The video frame numbers are then translated into an applicable sample numbers from the sensor stream, based on the synchronisation point, and class labels are added to the sensor data based on our activity definitions.

3.4.3 Data Cleaning

The sensor readings are normalised to a range between -1 and 1. The Echo State Network (ESN) prefers input in the range \([-1, 1]\) and so normalisation gives an opportunity to extract maximum information from the signal. Prior to normalisation we remove the 0.1% upper and lower quantiles. Removing the outliers and replacing them with the signal mean value allows us to maximise the signal range at the cost of a very small number of changes to the signal. Outlier removal is somewhat controversial and we discuss this in section 5.

3.5 ESN Model Description

In this study, we have chosen an Echo State Network (ESN) as our classifier. ESN were first introduced in (Jaeger 2001). ESN have been employed for a wide range of spatio-temporal real-world problems such as speech recognition (Jaeger et al. 2007, Verstraeten et al. 2006), financial forecasting (Ilies et al. 2007) and the prediction of chaotic dynamics (Jaeger & Haas 2004). Here we briefly outline the concept of the method and refer to the excellent review on ESN and related reservoir techniques presented in (Lukoševičius & Jaeger 2009) for further details. For the sake of consistency with previous descriptions of ESN, we adopt the nomenclature defined in the review article mentioned above.

The ESN attempts to learn a functional mapping from a (possibly multi-dimensional) input time series \(u(t) \in \mathbb{R}^{N_u}\) to a target time series \(y_{\text{target}}(t) \in \mathbb{R}^{N_y}\) based on a training data set \(\{u(t), y_{\text{target}}(t)\}\) with \(t = 1, \ldots, T\) where \(T\) is the size of the training set. The ESN is a neural network consisting of \(N_u\) input neurons, \(N_r\) reservoir (hidden) neurons and \(N_y\) output neurons. The reservoir neurons are either fully or sparsely interconnected with connection weights specified by a weight matrix \(W \in \mathbb{R}^{N_r \times N_r}\). Matrix \(W\) is initialized with random (uniform) weights and then scaled by the spectral radius \(\rho(W)\) which is the largest eigen value of \(W\). Generally, all \(N_u\) input neurons are connected to all reservoir neurons via

Figure 2: Input Data Set
connection weights defined by a separate input matrix $W_{\text{in}} \in \mathbb{R}^{N_x \times N_x}$. The weights of the input matrix are initialized as either -1 or 1 and then scaled by a scaling factor.

At time step $t$, the input $u(t)$ is fed into a neural network. The output of all reservoir neurons in the network is computed:

\begin{align}
   x(t) & = f(W_{\text{in}}u(t) + Wx(t-1)) \quad (1) \\
   x(t) & = (1 - a)x(t-1) + ax'(t) \quad (2)
\end{align}

function $f$ being a neuron activation function usually defined as the (element-wise) hyperbolic tangent $\tanh(\cdot)$, and factor $a \in \mathbb{R}$ being a leaking rate that controls the contribution of the previous neural output to its current state.

Since recurrent neural networks are difficult to train through gradient-descent based learning methods, ESN propose an elegant and efficient solution for imposing a desired input-output behaviour onto the network. Input and output vectors at time step $t$ are concatenated and then linearly transformed into the final output of the network:

\begin{equation}
   y(t) = f_{\text{out}}(W_{\text{out}}[u(t)|x(t)])
\end{equation}

where $W_{\text{out}} \in \mathbb{R}^{N_y \times (N_y + N_x)}$ is a weight matrix connecting all reservoir and all input neurons with $N_y$ output neurons, $\cdot | \cdot$ represents the vertical concatenation of vectors and $f_{\text{out}}$ is the activation function of the output neurons which is usually chosen as the identity.

The learning task is defined as an optimization problem in which the difference between $y(t)$ and $y_{\text{target}}(t)$ is minimized. Arguably the most popular method of computing matrix $W_{\text{out}}$ is linear regression or its regularized extension called ridge regression:

\begin{equation}
   W_{\text{out}} = Y_{\text{target}}X^{T}(XX^{T} + \alpha^2I)^{-1}
\end{equation}

where $I \in \mathbb{R}^{N_y \times N_y}$ is the identity matrix and $\alpha \in \mathbb{R}$ is a regularization factor that has to be carefully tuned for optimal results. Matrix $Y_{\text{target}} \in \mathbb{R}^{N_y \times T}$ contains all vectors $y_{\text{target}}$ concatenated into a matrix. We note that the connection weights $W$ are not modified by the learning rule and only the output weights $W_{\text{out}}$ are updated during training.

The role of the reservoir is the transformation of the input signal into a high-dimensional intermediate feature space represented by the neural network output at any time $t$. Although linear methods are then used to transform the feature vector into a desired target output, the mapping of the input $u(t)$ to output $y(t)$ is of non-linear nature.

The concept of generating intermediate feature vectors is also exploited in other kernel-based machine learning algorithms, most prominently the Support Vector Machine (SVM). By choosing a straightforward linear learning rule, the training process becomes highly efficient. ESN allow the exploitation of the interesting characteristics of recurrent neural networks without the need of mathematically and computationally complex training algorithms.

### 4 Results

We used an evolutionary algorithm, i.e. a Particle Swarm Optimiser (PSO) to do a search of the parameter space for the ESN, to find a suitable configuration to use for our experiment. We optimize the regularization parameter $\alpha$, the number of reservoir neurons $N_x$, the scaling factor of the input weights, the leaking rate $a$ and the spectral radius $\rho(W)$. The ranges for these ESN parameters are generally accepted sensible ranges and were taken from (Lukoševičius 2012) and are shown in table 1. We ran the PSO with 14 particles over 50 generations and within each generation the ESN was initialized and trained five times to take some account of the stochastic nature of the initialization process. Each ESN was trained on the first 50% of the time series and the entire series was used for testing. The root mean square error between target and actual network output on the test set was used as a fitness measure for the PSO.

The results of this search process can be seen in figure 3. As is visible from the diagram, four of the five parameters (Number of neurons, Input Scaling, Leak Rate and Spectral Radius) had settled into a small range within 20 generations. Regularization, the fifth parameter, appears less critical for this problem and a range of configurations reported satisfying test errors.

![Figure 3: Optimising the ESN parameters](image)

Table 1: PSO Parameter Ranges

<table>
<thead>
<tr>
<th>ESN parameters &amp; ranges</th>
<th>value</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regularisation $\alpha$</td>
<td>$0$ to $5$</td>
<td></td>
</tr>
<tr>
<td>Number of Neurons $N_x$</td>
<td>$100$ to $1,000$</td>
<td></td>
</tr>
<tr>
<td>Input Scaling $\alpha$</td>
<td>$0$ to $1$</td>
<td></td>
</tr>
<tr>
<td>Spectral Radius $\rho(W)$</td>
<td>$0$ to $1$</td>
<td></td>
</tr>
</tbody>
</table>

The selected parameters from the PSO that were used to run the ESN were Regularisation (2.7), Neurons (939), Input Scaling (0.9147), Leaking Rate (0.05937) and Spectral Radius (1.0).
Fifty percent of the data file was used for training and the full file was used for testing. The file was divided in two parts with no account for where that separation point fell in terms of the classes. In this case, as a result of where the mounts and dismounts occur, the first half of the file includes the first 8 mounts and the first 7 dismounts. The output from the test run is shown in figure 4.

Using a cut off point set at 0.5 of the continuous output $y(t)$ has all mounts and dismounts successfully classified with one false positive mount classified during the sequence when the participant takes off the sensor. No false positive dismounts were classified. Within each mount and dismount there is a slight lag between the class label and the classified label in most cases with an associated drop off towards the end of each class. This is an expected characteristic of the reservoir property as the reservoir needs some time to establish a recognised pattern.

<table>
<thead>
<tr>
<th>Class</th>
<th>Predicted Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3298</td>
</tr>
<tr>
<td>1</td>
<td>103</td>
</tr>
<tr>
<td>2</td>
<td>58</td>
</tr>
</tbody>
</table>

Table 2: Confusion Matrix of Results

5 Conclusions and future work

5.1 Conclusion

The ESN classifier has worked well in this particular situation, using data from a scripted activity in a laboratory setting collected from a single participant. The false positive mount possibly indicates that the ESN classifier would benefit from additional training that provides a wider variety of signals that are outside of the desired classes. This is supported by some follow on work (not shown here) where we used the last eight mounts and dismounts as training data with the full set of mounts for testing and in this case we provided the “taking the sensor off” segment as part of the training and no false positives were classified.

Our use of overhead claps for synchronisation and scripted pauses in activity introduce significant signal artefacts into our data. While it is not possible with ESN to tell what properties of the signal are used by the classifier it is probable that our artefacts make a significant contribution to classification. For example, the initial spike in the classifier response in the wider gap between the second dismount and the third mount may be indicative of the ESN starting to respond to the synchronisation signal. In addition, the clear pause in activity after mounting consistently occurs just prior to dismounting and so this artefact is undoubtedly contributing to dismount classification success. We would be unwise to conclude that our current classifier is suitable for use on data that does not contain artificial artefacts.

The current pre-processing methods that we use to normalise the signal between -1 and +1 will have created additional signal artefacts including magnifying the drift in the Magnetometer and Gyroscope data. Our removal of outliers has enabled us to amplify the central portion of the signal but the outliers are real data and so by removing them we have changed the underlying activity signatures. These issues, without resolutions, will make it difficult to generalise the results across participants and across other sensors.

This data was collected from a single individual using a single (wooden) horse on the same day using the same equipment. This provides unrealistic consistency. In the real world not only do riders not follow a script when mounting and dismounting they also come in all sizes, temperaments and skill levels; their horses come in all sizes, temperaments and training levels; additional equipment may be involved such as a rider holding a crop while mounting and differing techniques may be used while mounting. All of this cautions us against simplistically concluding that we are close to having a simple, reliable method of classifying this (or any other punctual activity) that works across riders and situations.

This reported experiment is a idealised situation, designed to provide the best chance of successful classification and so while we are pleased that these re-
results are positive they need considerable further development before we can safely conclude that RC methods are well suited to classifying punctual human activities based on inertial data. Despite this caution, these results are positive enough that we and perhaps other researchers are willing to follow this path further.

5.2 Future work

This work is part of a larger set of work designed to resolve some of the issues mentioned in our conclusions as well as additional issues. Some of our future plans include:

- Comparing the ESN classifier with more traditional kernel based classifiers such as Support Vector Machine.
- Testing to see if the classifier will generalise across participants.
- Testing to see if the ESN classifier can generalise across activities in different places and time for the same individual.
- Training and testing the ESN on data collected during real world situations.
- Designing methods of translating our sensor data from the form in which it was captured (sensor dependent) into a standardised format more useful for input into the ESN in a way that allows us to draw the maximum information out of the data while still allowing it to be comparable between data collection sessions. This includes alternate methods of pre-processing the data to filter out noise, to account for drift and methods for including outliers.

References


