Activity and Gait Recognition with Time-Delay Embeddings

Jordan Frank
Department of Computer Science
McGill University, Montreal, Canada
jordan.frank@cs.mcgill.ca

Shie Mannor
Faculty of Electrical Engineering
The Technion, Haifa, Israel
shie@ee.technion.ac.il

Doina Precup
Department of Computer Science
McGill University, Montreal, Canada
dprecup@cs.mcgill.ca

Abstract

Activity recognition based on data from mobile wearable devices is becoming an important application area for machine learning. We propose a novel approach based on a combination of feature extraction using time-delay embedding and supervised learning. The computational requirements are considerably lower than existing approaches, so the processing can be done in real time on a low-powered portable device such as a mobile phone. We evaluate the performance of our algorithm on a large, noisy data set comprising over 50 hours of data from six different subjects, including activities such as running and walking up or down stairs. We also demonstrate the ability of the system to accurately classify an individual from a set of 25 people, based only on the characteristics of their walking gait. The system requires very little parameter tuning, and can be trained with small amounts of data.

Introduction

Mobile devices are ubiquitous in modern society. Many of these devices come equipped with sensors such as accelerometers and Global Positioning Systems (GPS) that can collect data about the movements of a user. This creates the potential of intelligent applications that recognize the user's activity and take appropriate action. For example, a device might notice that a user is in distress and alert a care giver, or it might make decisions regarding whether incoming phone calls should be put through or saved in a message box.

Applications for recognizing activities from sensor data have been developed in a wide range of fields such as health care (Kang et al. 2006), fitness (Lester, Choudhury, and Borriello 2006), and security (Kale, Cuntoor, and Chellappa 2002). Different types of sensors have been used, e.g., accelerometer data for recognizing physical activities (Lester, Choudhury, and Borriello 2006) and both mobile phone usage data (Farrahi and Gatica-Perez 2008) and GPS data (Krumm and Horvitz 2006; Liao, Fox, and Kautz 2007) for analyzing human mobility and other higher-level, temporally extended activities (Eagle and Pentland 2006).

Our focus is on detecting primitive physical activities (e.g., walking, running, biking, etc.) from data collected using accelerometers built into wearable sensing devices. Previous work on this problem has shown recognition accuracy above 85% on large, complex data sets (Subramanya et al. 2006; Lester, Choudhury, and Borriello 2006). These existing approaches use standard (mostly linear) signal processing methods, through which hundreds of signal statistics are computed. Heavy-duty feature extraction tools are then used to determine which of these features are useful for classification. While these approaches work well, they require a lot of computing power, so they are not well suited for real-time detection of activities on low-powered devices.

In this work, we propose an alternative method for representing time series data that greatly reduces the memory and computational complexity of performing activity recognition. We adopt techniques from nonlinear time series analysis (Kantz and Schreiber 2004) to extract features from the time series, and use these features as inputs to an off-the-shelf classifier. Additionally, we present a novel approach for classifying time series data using nonparametric nonlinear models that can be built from small data sets. This algorithm has been implemented and is able to perform classification of activities in real time on an HTC G1© (also known as Dream in some countries) mobile phone.

We will focus in particular on accelerometer data; intuitively, the acceleration recorded by the device is the result of a measurement performed on a non-linear dynamical system, consisting of the hips and legs, with their joints, actuations and interaction with the ground. Time-delay embedding methods, e.g., (Sauer, Yorke, and Casdagli 1991), provide approaches for reconstructing the essential dynamics of the underlying system using a short sequence of measurements, evenly spaced in time. Assuming that the dynamics of the underlying system are noticeably different when a person is performing different activities, the parameters of the different time-delay embedding models can be used to classify new data. Our approach achieves very good classification performance, comparable to the state-of-art approaches, but extracts fewer than a dozen features from the time series data, instead of hundreds. The memory and computational savings are crucial, given that for many applications, activity recognition would have to run in real time as a small component of larger system on a low-powered mobile device.

The main contributions of this paper are twofold. First, we propose the use of the reconstructed coordinates in a time-delay embedding of a time series as input to a supervised learning algorithm for time series classification. While time-delay values have been used as input to neural networks (Waibel et al. 1989), the novelty of our approach is in the incorporation of noise-reduction and the explicit treatment of the dynamics. Second, we propose a lightweight classification algorithm based on time-delay embeddings that is efficient enough to run in real time on a low-powered device. Both methods are empirically validated on noisy real-world data.

Time-Delay Embeddings

In this section we present a brief overview of time-delay embeddings. We refer the reader to the standard text by Kantz and Schreiber (Kantz and Schreiber 2004) for further details. The purpose of time-delay embeddings is to reconstruct the state and dynamics of an unknown dynamical system from measurements or observations of that system taken over time. The state of the dynamical system at time t is a point $x_t \in \mathbb{R}^k$ containing all the information necessary to compute the future of the system at all times following t. The set of all states is called the phase space. Of course, the state of the system cannot be measured directly, and even the true dimension of the phase space k is typically unknown. However, we assume access to a time series $\langle o_t = \omega(x_t)\rangle$ generated by a measurement function $\omega : \mathbb{R}^k \to \mathbb{R}$, which is a smooth map from the phase space to a scalar value. Throughout this paper, we use "smooth" to mean continuously differentiable. In general, it is hard to reconstruct the state just by looking at the current observation. However, by considering several past observations taken at times $t, t-\delta_1, \ldots t-\delta_{m-1}$, the reconstruction should be easier to perform. These m measurements can be considered as points in $\mathbb{R}^m$, which is called the *reconstruction space*. The key problem is to determine how many measurements m have to be considered (i.e., what is the dimension of the reconstruction space), and at what times they should be taken (i.e., what are the values of δ_i), in order to capture well the system dynamics. In particular, if we consider the map from $\mathbb{R}^k$ to $\mathbb{R}^m$ (corresponding to taking the measurements and then projecting the time series data into the reconstruction space) we would like this map to be one-to-one and to preserve local structure. If this were the case, by looking at the time series in $\mathbb{R}^m$, one would essentially have all the information from the true system state and dynamics.

We consider the *time-delay reconstruction* in m dimensions with time delay τ, which is formed by the vectors $\mathbf{o}_t \in \mathbb{R}^m$, defined as: $\mathbf{o}_t = (o_t, o_{t+\tau}, o_{t+2\tau}, \ldots, o_{t+(m-1)\tau})$. We assume that the true state of the system x_t lies on an attractor $A \subset \mathbb{R}^k$ in an unknown phase space of dimension k. We use the term "attractor", as in dynamical system theory, to mean a closed subset of states towards which the system will evolve from many different initial conditions. Once the state of the system reaches an attractor, it will typically remain inside it indefinitely (in the absence of external perturbations). An *embedding* is a map from the attractor A into reconstruction space $\mathbb{R}^m$ that is one-to-one and preserves local differential structure. Takens (1981) showed that if A is a d-dimensional smooth compact manifold, then provided that $m > 2d$, $\tau > 0$, and the attractor contains no periodic or-

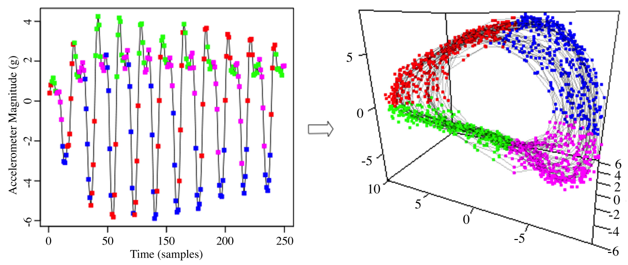

Figure 1: Example of time-delay embedding for accelerometer data from biking activity. Left: raw accelerometer data. Right: time-delay embedding with $m = 6, \tau = 11, p = 3$. The coloured points in the left diagram are mapped to the same coloured points in the embedding.

bits of length τ or 2τ and only finitely many periodic orbits of length $3\tau, 4\tau, \ldots, m\tau$, then for almost every smooth function ω, the map from $\mathbb{R}^k$ to the time-delay reconstruction is an embedding. In other words, any time-delay reconstruction will accurately preserve the dynamics of the underlying dynamical system, provided that m is large enough and τ does not conflict with any periodic orbits in the system. However, from the point of view of activity recognition, we would like m to be as small as possible, as it determines the minimum time interval after which an activity can be recognized. More precisely, if the data is sampled at a rate of r measurements per second, then the time window needed for the time-delay reconstruction is $((m-1)\tau + 1)/r$ seconds.

Takens' theorem only holds when the measurement function ω is deterministic, but in practice all measurements are corrupted by noise from a variety of sources, ranging from the measurement apparatus to rounding errors due to finite precision. In the presence of noise, the choice of m and τ becomes very important. There are a number of techniques for choosing good values for m and τ based on geometrical and spectral properties of the data (Buzug and Pfister 1992). However, in our experiments, we found that grid search over a small parameter space, using a small validation data set and optimizing classification accuracy on it, suffices to find good parameters.

Methods developed for nonlinear time series analysis have not been widely used in the machine learning community[1], but are quite popular in other areas of research. In the following section, we explain how we use time-delay embeddings for the purpose of classifying time series data.

Proposed Approach

The main idea of our approach is to extract features using time-delay embeddings, followed by a noise-reduction step. Then we use these features as input to a classifier. Define the time-delay embedding function $T(o, m, \tau)$ which takes as parameters a time series $o = \langle o_i \rangle_{i=1}^{N}$, and the time-delay embedding parameters m and τ. Let $M = N - (m-1)\tau$. The function T returns an $m \times M$, matrix $\mathbf{O}$ whose ith column is the ith time-delay vector $(o_i, o_{i+\tau}, o_{i+2\tau}, \ldots, o_{i+(m-1)\tau})$.

[1]The obvious exception is the time-delay neural network research. However these methods do not consider noise reduction, or other transformations on these models, they simply use lagged time sequences as inputs to a network.

For the noise reduction step, principal component analysis (PCA) is performed on the embedding of a short segment of the training data (Jolliffe 2002). PCA computes a projection matrix $\mathbf{P}$ from the time-delay embedding space to a space of lower dimension p, which we will call *model space*. Once the projection matrix $\mathbf{P}$ has been computed, any sequence of observations can be projected into the *model space* by computing the time delay embedding , centering the data by subtracting its mean, then multiplying by $\mathbf{P}$.

Figure 1 illustrates the concept of time-delay embedding using accelerometer data collected from a subject riding a stationary bicycle. On the top is a short segment that was used to find the projection $\mathbf{P}$. The parameters used, $m = 6$, $\tau = 11$, and $p = 3$, were chosen based on previous experiments. On the bottom is the result of embedding the entire data set and then projecting onto the basis specified by $\mathbf{P}$. Observe that the system evolves along a set of tightly clustered trajectories, in a periodic fashion. This is quite intuitive, considering that cycling involves a periodic, roughly two-dimensional leg movement. However, it is remarkable that this structure can be reconstructed automatically from the time series on the top, especially since the data is non-stationary (i.e., the peaks do not have the same magnitude, and the period varies over time). Note that the colours are only used as a visual aid, to show where different regions from the time series get mapped in the final model space.

Given a sequence of observations, the computation described above will produce a matrix whose columns form a sequence of points in the p-dimensional Euclidean model space, representing the states of the dynamical system. We call this sequence of points a *model*. The coordinates of these points can be used as input features to a learning algorithm. However, we can also treat subsequent pairs of points as Euclidean vectors, and look for richer information on the dynamics of the system. In the next section, we describe a novel learning algorithm that exploits this aspect to our advantage.

Geometric Template Matching Algorithm

The geometric template matching (GTM) algorithm is an efficient algorithm for assessing how well a given time series matches a particular model. The time series is projected into model space, then short segments of the projected data are compared geometrically against their nearest neighbours in the model. We treat subsequent pairs of points in model space as vectors. Segments of the projected data set that are geometrically similar to nearby vectors in the model are given higher scores, while segments that differ from the model are given low scores.

The algorithm, presented in Algorithm 1, takes as input a time series and a model, and computes nearest-neighbours for the time series based on Euclidean distance. A set of scores for each segment of the time series is returned as a result. In the algorithm, we denote the Euclidean norm by $|\mathbf{u}|$, and the dot product of two vectors $\mathbf{u} \cdot \mathbf{v}$. For notational convenience we write the vector going from point a to point b as $[a,b]$.

For classification purposes, one can simply assign to the sequence the class label associated with the model that obtains the highest average score. In order to perform classification in real-time, as opposed to using batch input, a buffer of the most recent $(m-1)\tau + s$ samples is maintained, and the GTM algorithm is run on the buffered data (which will result in one iteration of the outer loop, and thus one score) as new data become available. In our experiments, we are able to compute a score for 8 models twice per second using the embedding parameters $m = 12, \tau = 2, p = 6, s = 32$, and $n = 4$, with no noticeable impact on the performance of the phone.

Algorithm 1 Geometric Template Matching

```
Input: Model $u = \langle u_i \rangle_{i=1}^N$, parameters $(m, \tau, p, \mathbf{P})$, sequence length $s$, neighbourhood size $n$, and the sequence to score, $o = \langle o_i \rangle_{i=1}^T$.
Output: A set of scores, $r_1, \ldots, r_{T-(m-1)\tau-s} \in [-s, s]$.
1. Initialize the score, $r_i = 0$ for $i = 1, \ldots, (M-s)$.
2. Compute matrix $\mathbf{V}$ that represents the model for sequence $o$, whose columns form the sequence $\langle v_i \rangle_{i=1}^M$, where $M = T - (m-1)\tau$.
3. Repeat for $i = 1, \ldots, (M-s)$:
     Repeat for $j = i, \ldots, (i+s-1)$:
       a. Let $w_1, \ldots, w_n$ be the indices of the $n$ nearest neighbours of $v_j$ in $u$.
       b. Let $\mathbf{u}$ be the mean of the vectors $[u_{w_1}, u_{w_1+1}], \ldots, [u_{w_n}, u_{w_n+1}]$.
       c. Let $\mathbf{v}$ be the vector $[v_j, v_{j+1}]$.
       d. Update the score, $r_i = r_i + (\mathbf{u} \cdot \mathbf{v}) / \max(|\mathbf{u}|, |\mathbf{v}|)$.
4. Return the scores, $r_1, \ldots, r_{T-(m-1)\tau-s}$.
```

Experiments

We demonstrate our approach in two experiments focusing on identifying activities and users, respectively.

Activity Recognition

The data set used for the activity recognition experiment was collected using the Intel Mobile Sensing Platform (MSP) (Choudhury et al. 2008) that contains a number of sensors including a 3-axis accelerometer and a barometric pressure sensor; these are the only sensors we will use. Six participants were outfitted with the MSP units, which clip onto a belt at the side of the hip, and data was collected from a series of 36 two-hour excursions (6 per user) which took place over a period of three weeks. The participants were accompanied by an observer who recorded the labels as the activities were being performed. The labels were: walking, lingering, running, up stairs, downstairs, and riding in a vehicle. We omitted the vehicle activity for now (because we are not processing GPS signal). Taking into account equipment failures and data with obvious errors in the labelling, the working data set consists of 50 hours of labeled data, roughly equally distributed among the six participants.

The accelerometer data is sampled at 512Hz, which we decimate to 32Hz. The accelerometer data consists of three measurements at each time step, corresponding to the acceleration along each of the three axes, x, y, and z. We combine these three measurements to form a single measure of the

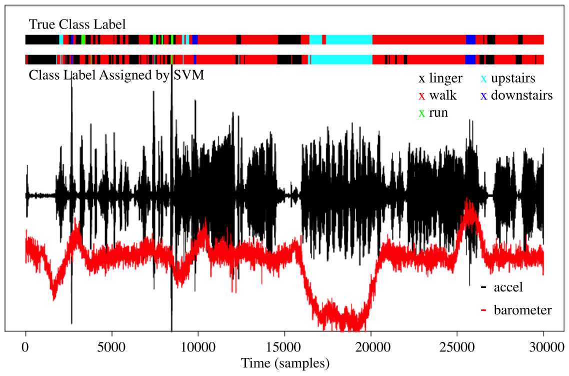

Figure 2: Classification results on a segment of the data from user 6 using a classifier trained on user 4. The top coloured bar above the data shows the labels assigned by a human labeller and the bottom coloured bar shows labels assigned by the classifier. The black line shows the magnitude of the accelerometer data and the red line is the gradient of the barometric pressure.

magnitude of the acceleration vector $a = \sqrt{x^2+y^2+z^2} - g$, where $g = 9.8m/s^2$ is the Earth's gravity. Subtracting g causes the acceleration when the device is at rest or moving at a constant velocity to be 0. The barometric pressure is sampled at 7.1Hz, smoothed, and then upsampled to 32Hz. The gradient of the signal is used as an additional feature.

For the experiment, we split the data set up into six parts, each containing the data from a specific participant. We began by projecting all of the accelerometer data into a time-delay reconstruction space with parameters $\tau = 3$ and $m = 16$. For each user, we constructed a training set by selecting randomly 25% of these embedded data points. This corresponds to approximately 2 hours of data, or 230,000 samples for each participant. Next, we performed PCA on the data points in the training set. We then projected all the reconstructed accelerometer data onto the principal components corresponding to the 9 largest eigenvalues from the training data. This produces 9 features for each data point in the original time series, and we combined these features with the barometric pressure to form the input features for a Support Vector Machine (SVM) classifier (Cristianini and Shawe-Taylor 2000). As a basis for comparison, we also trained an SVM just using the raw accelerometer value and gradient of the barometric pressure as inputs at each time step.

We tested each classifier on the entire data set for each user. The results are shown in Table 1. The first row contains the results when training using only the single raw accelerometer value and the gradient of the barometric pressure, trained on data from User 1, and testing on all of the users. It is clear that using features obtained by time-delay embeddings significantly improves classifier performance. Figure 2 shows the result of using the SVM trained on User 3 to label a segment of the data for User 6.

The classifier performs well across all users, regardless of the user on which it was trained. The average accuracy for the experiments using the time-delay embedding on the entire data set is 85.48±0.30%. This result demonstrates that these features are useful for activity recognition devices, because the system can be calibrated on one user, then deployed to other users, and the performance is very similar (the accuracy loss, as can be seen from the table, is 1-5%).

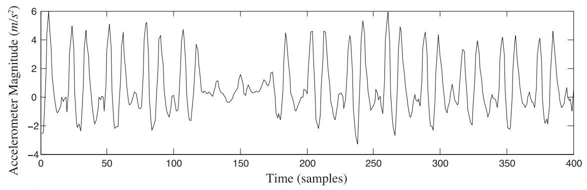

Figure 3: An example segment of the accelerometer data.

Gait Recognition

Gait recognition is the problem of identifying a person by their manner of walking, or carriage. The problem of gait recognition has been studied in depth in the computer vision and biomechanics communities, where the goal is to identify an individual based on a sequence of images or silhouettes captured while the individual is walking (Nixon, Tan, and Chellappa 2006). Recent work has considered gait recognition using wearable sensors as a means for biometric identification (Gafurov, Snekkenes, and Bours 2007). For our experiment, we implemented a data collection and analysis tool for the Google Android operating system, which runs on the HTC G1© mobile phone. In our experiments, the phone was placed in the front trouser pocket of an individual, and collected data from the 3-axis accelerometer in the device at a rate of 25Hz[2] as the subject walked. We collected a set of 25 traces, each containing between 12 and 120 seconds of continuous walking data from one of 25 individuals.

The data was trimmed to remove the time before and after the walk, and the first and last few steps, but no other post-processing was performed. Figure 3 contains a short sequence from one of the data sets. As we can see, the data is noisy, and there is a substantial amount of variability in the signature for each cycle of the walk. In addition, the subjects were asked to walk to one end of a hall and back, which required them to turn 180°. This leads to a few cycles of the walk that are substantially different than the others (as seen in samples 120–160 in Figure 3).

Each data set was split into a training and test set. Since the data is sequential, a single block of data of a fixed length was chosen for the test set, and the remaining data was used as the training set. While this adds a single point of discontinuity in the training set when the test set is not at the very beginning or end of the data set, this does not have a noticeable effect on our results. Time-delay embedding models were constructed for each of the training sets with dimension $m = 12$, delay time $\tau = 2$ and projected dimension $p = 6$. The number of nearest neighbours considered in each model was 4. All of the parameters were chosen based on previous experiments, before this data set was collected.

Each test set was then compared to each model. For each test set S and model T, we project S into the model-space for T, and then every segment of length 32 was compared to the model T as described in the previous section. The scores

[2]The sampling rate on the phone is inconsistent, and ranges from 17Hz to 30Hz depending on the processor load, which is an additional source of noise.

		Testing						
		User 1	User 2	User 3	User 4	User 5	User 6	Total
Training	User 1, simple	77.26%	75.61%	76.94%	77.86%	78.55%	75.90%	77.11%
	User 1	87.89%	82.6%	84.29%	86.73%	85.76%	84.67%	85.27%
	User 2	85.95%	84.35%	84.04%	86.59%	86.04%	85.50%	85.49%
	User 3	86.28%	82.98%	85.08%	87.03%	85.99%	85.07%	85.44%
	User 4	86.00%	82.69%	84.11%	89.41%	85.56%	86.12%	85.79%
	User 5	86.15%	82.89%	84.06%	86.24%	86.61%	83.83%	85.04%
	User 6	86.29%	83.48%	83.85%	87.56%	85.56%	87.75%	85.83%
							Average:	85.48±0.30%

Table 1: Results for activity recognition task. Each row corresponds to the results of training on the participant given in the first column, and the values indicate the classification accuracy on the data set for the user specified by the column header. The last column gives the results on the entire data set. The first row corresponds to using only the raw accelerometer and barometric pressure gradient data, without using time-delay embedding.

for each segment were then averaged to give a score for how well model T matched the data in S.

We used 5-fold cross-validation, and so the above procedure was repeated 5 times with 5 different test and training sets. The scores were then averaged across the 5 runs. Remarkably, our algorithms achieved 100% accuracy on the test set: every test set was associated with the correct user model.

Discussion

The results we obtain for both experiments have high accuracy, despite the fact that we use significantly fewer features than are generally used for time series analysis based on signal processing techniques. For the gait recognition task, our method requires a time window of 22 samples. The average walking cycle has a period of approximately one second, and so for a 1Hz signal to be detected confidently using any method that computes the spectrum of the data, hundreds of samples must be considered.

Our activity recognition results use significantly fewer features than current approaches, but achieve similar results. To provide some perspective, we note that a similar data set was considered by Subramanya et al. (2006), who applied the methods proposed by Lester, Choudhury, and Borriello (2006) and achieved an accuracy of 82.1% on the same set of activities we consider. Their system used 650 features of the time series, composed of cepstral coefficients, FFT frequency coefficients, spectral entropy, band-pass filter coefficients, correlations, and a number of other features that require a nontrivial amount of computation. A modified version of AdaBoost was then used to select the top 50 features for classification. Comparatively, we used 16 samples of the raw signal, corresponding to a window of 48 samples, or one and a half seconds, and then computed a linear projection to get the 9 features that are used to train the classifier. We emphasize that existing results are obtained on different data sets, and we only report their results because their work is the closest in nature to our own. We make no claim that our method obtains better results, only that the features that we use are considerably easier to compute, while the classification accuracy is similar.

The accuracy figures could be improved with further post-processing, because as can be seen in Figure 2, most of the mislabeled segments are short. State of the art activity recognition systems routinely perform temporal smoothing, which reduces such errors. As a simple test, we aggregated sequences of 8 predictions and took a majority vote at 1/4 second intervals (instead of making 32 separate predictions every second), matching the rate of existing activity recognition systems (Lester, Choudhury, and Borriello 2006). With this approach, the error rate decreased by 2-4%, depending on the experiment. However, in this paper we want to focus on the power of the representation, and not try to use other methods to mitigate errors. Better performance could also be achieved using other classification methods, such as the decision stumps classifiers used by Lester, Choudhury, and Borriello (2006) or the semi-supervised CRFs used by Mahdaviani and Choudhury (2007). We chose SVMs simply because of the ease of using off-the-shelf code.

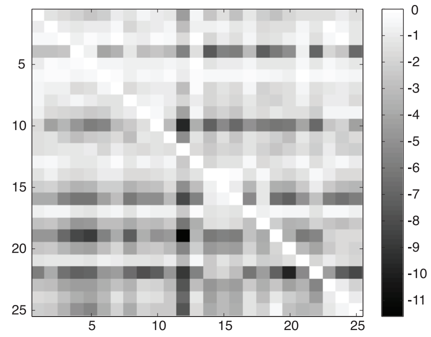

Figure 4: Scores for the gait recognition task. Rows represent test sets and columns represent models. The shading represents the difference between the score and the maximum score for the row in terms of empirical standard deviations. White represents no difference (i.e., the maximum score for the row), and darker shading represents larger differences.

For the gait recognition task, we were able to perfectly classify the 25 individuals. While this result is very encouraging, the difference in scores between the top two models is usually small. In terms of the empirical standard deviation over the 5 cross validation runs, the average difference between the top two scores is only 0.76, which is not significant. The average difference between the top score and the fifth score is 1.34 empirical standard deviations, still not very significant.

Figure 4 shows scores for each test set when compared to each model. The shading in the ith row and jth column depicts the difference between the average score for test set i with model j and the maximum average score for any of the test sets under model j, in terms of empirical standard deviations. The white squares represent the maximum average scores. Although the maximum scores lie on the diagonal, for many of the test sets a large number of the models obtain scores that are close to the maximum score. For example, the difference between the highest and lowest average score for test set 6 is only 1.05 empirical standard deviations. On the other hand, for some test sets, such as test sets 10 and 22, the classifier clearly identifies the correct model.

Other authors have considered using wearable sensors to perform gait recognition. Due to the personal nature of these data sets, they are not publicly available, so we can only offer a qualitative comparison. Gafurov, Snekkenes, and Bours (2007) analyzed a data set consisting of data collected from 50 individuals. They reported a recognition rate of 86.3% using features that were hand-crafted for the gait recognition task. We also note that the sensors that they used were special-purpose motion sensors, and have both a higher sampling rate and less noise than the sensors that we employed. Our method appears to offer better performance, despite not being designed for this particular task.

Conclusion and Future Work

In this paper, we demonstrated that time-delay embeddings provide a powerful substitute for the representations that are typically used in the activity recognition literature, especially given the need for small memory and processing requirements. Our results show that with few, easy-to-compute features, we can obtain classification accuracy comparable to state-of-the-art activity recognizers, without any tuning of the classifier, or any post-processing of the labels assigned by the classifier. Other existing approaches use much more complicated features and specially tuned classifiers. In the gait recognition task, our proposed classification algorithm achieves 100% test set accuracy on a noisy data set consisting of 25 individuals. Not only is the accuracy high, the algorithm is efficient enough that it can classify in real time, on a mobile phone.

While we focus on the problem of activity recognition, we would like to emphasize the usefulness of these methods for reconstructing a state representation for any data generated by a nonlinear dynamical system. This includes modelling of state spaces and dynamics for partially observable systems, often considered in reinforcement learning.

There are some obvious steps to produce an industrial-strength activity recognition system. Using additional methods such as hidden Markov models to perform temporal smoothing of the classifier output, and testing other types of discriminative classifiers are obvious next steps in this process. We are also working on incorporating other types of sensors, such as GPS, audio, and video, to provide further context for detecting more interesting activities. Ultimately, the goal of this work is to move beyond recognizing primitive activities, in order to build a system that will recognize higher-level activities such as going for a walk to the store, or driving to work.

Acknowledgments This work was partially supported by NSERC, FQRNT, and the Israel Science Foundation under contract 890015. The authors would like to thank Dieter Fox for the data used in the activity recognition experiment.

References

Buzug, T., and Pfister, G. 1992. Optimal delay time and embedding dimension for delay-time coordinates by analysis of the global static and local dynamical behavior of strange attractors. *Phys. Rev. A* 45(10):7073–7084.

Choudhury, T.; Borriello, G.; Consolvo, S.; Haehnel, D.; Harrison, B.; Hemingway, B.; Hightower, J.; et al. 2008. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing* 32–41.

Cristianini, N., and Shawe-Taylor, J. 2000. *An introduction to Support Vector Machines: and other kernel-based learning methods*. Cambridge University Press.

Eagle, N., and Pentland, A. 2006. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing* 10(4):255–268.

Farrahi, K., and Gatica-Perez, D. 2008. What did you do today?: Discovering daily routines from large-scale mobile data. In *Proceedings of ACM International Conference on Multimedia*, 849–852.

Gafurov, D.; Snekkenes, E.; and Bours, P. 2007. Gait authentication and identification using wearable accelerometer sensor. In *2007 IEEE Workshop on Automatic Identification Advanced Technologies*, 220–225.

Jolliffe, I. 2002. *Principal component analysis*. Springer, New York.

Kale, A.; Cuntoor, N.; and Chellappa, R. 2002. A framework for activity-specific human recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, volume 706.

Kang, D.; Lee, H.; Ko, E.; Kang, K.; and Lee, J. 2006. A wearable context aware system for ubiquitous healthcare. In *IEEE Engineering in Medicine and Biology Society*, 5192–5195.

Kantz, H., and Schreiber, T. 2004. *Nonlinear time series analysis*. Cambridge University Press.

Krumm, J., and Horvitz, E. 2006. Predestination: Inferring destinations from partial trajectories. In *Proceedings of Ubicomp*, 243–260.

Lester, J.; Choudhury, T.; and Borriello, G. 2006. A practical approach to recognizing physical activities. *Lecture Notes in Computer Science* 3968:1–16.

Liao, L.; Fox, D.; and Kautz, H. 2007. Extracting places and activities from gps traces using hierarchical conditional random fields. *The International Journal of Robotics Research* 26(1):119–134.

Mahdaviani, M., and Choudhury, T. 2007. Fast and scalable training of semi-supervised CRFs with application to activity recognition. In *Proceedings of Neural Information Processing*.

Nixon, M.; Tan, T.; and Chellappa, R. 2006. *Human identification based on gait*. Springer.

Sauer, T.; Yorke, J.; and Casdagli, M. 1991. Embedology. *Journal of Statistical Physics* 65(3):579–616.

Subramanya, A.; Raj, A.; Bilmes, J.; and Fox, D. 2006. Recognizing activities and spatial context using wearable sensors. In *Proceedings of Uncertainty in Artificial Intelligence*.

Takens, F. 1981. Detecting strange attractors in turbulence. *Dynamical systems and turbulence* 898(1):365–381.

Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; and Lang, K. 1989. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing* 37(3):328–339.