Activity Recognition for Users of Rolling Walker Mobility Aids

Mathieu Sinn and Pascal Poupart
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada N2L 3G1
{msinn,ppoupart}@cs.uwaterloo.ca

Abstract

We present Smart Walkers, a comprehensive approach to enhancing independent and safe mobility of elderly people. The idea of the Smart Walkers project is to equip rolling walker mobility aids with sensors and actuators. The goal is to assist users, caregivers and clinicians, e.g., by monitoring the physical and mental conditions of the user, detecting risks of falling, assessing the effectiveness of therapeutic interventions, and providing active navigation assistance.

The key problem in building the Smart Walkers technology is the ability to recognize the user activity from the stream of sensor measurements. In this paper we present supervised and unsupervised machine learning algorithms for this purpose and discuss their performance on real user data. We find that the best results are obtained for Conditional Random Fields with feature functions based on thresholding, achieving an accuracy of 85-90%.

1 Introduction

Safe and independent mobility is a key factor in the quality of life of elderly people. Independent mobility sustains social interaction and thereby physical and mental activity; on the other hand, injuries caused by falls constitute a major health risk. Mobility aids, such as canes, rolling walkers and wheel chairs, encourage independent mobility, however, improper use can induce additional risks of falling, particularly as the individual motoric capabilities deteriorate.

To improve the utility of mobility aids, we are developing a mixed-initiative system, called Smart Walker, which is a customary four-wheel rolling walker equipped with a set of sensors. A prototype is shown in Fig. 1. The sensors measure the load on the four wheels, the walked distance and the acceleration in the three spatial dimensions. In our future work, we plan to use the sensor measurements to monitor the user's physical and mental conditions, and to detect moments of instability. Furthermore, we plan to instrument the Smart Walker with actuators which can immobilize the walker, e.g., while the user is standing, or provide navigation assistance.

The key problem in building the Smart Walker technology is to extract high-level information about the user's activities from the stream of sensor measurements. In [1], we compared supervised and unsupervised machine learning algorithms for this purpose. We found that Conditional Random Fields (CRFs) yielded better results than Hidden Markov Models (HMMs), presumably due to the less restrictive assumptions on the distribution of the observation sequence. The CRFs that we used were based on feature functions which, in order to discriminate between activities, compared the sensor measurements to fixed threshold values. While in [1] we suggested to choose the threshold values manually, in this paper we consider an automatic approach based on Linear Regression models. We also generalize some of the ideas by considering "smooth" thresholding based on sigmoid functions, and using the raw observations to discriminate between activities.

The paper is structured as follows: Sec. 2 provides some background on the Smart Walker project and our previous research. In Sec. 3, we describe the problem of activity recognition and present algorithms based on Conditional Random Fields. In Sec. 4, we evaluate the algorithms in two experimental studies. An outlook on our future work in Sec. 5 concludes the paper.

Figure 1: A prototype of the Smart Walker.

2 The Smart Walker Project

The Smart Walker prototype has been developed at the Toronto Rehabilitation Institute (see [2]). It is a four-wheel rolling walker equipped with four load sensors (one in each leg) to measure the ground reaction forces, a wheel encoder to measure the walked distance, a 3D-accelerometer to measure the instantaneous acceleration, and two video cameras (facing forward and backward, respectively) to record the environment and the position of the lower limbs relative to the walker. As a standard feature of rolling walkers, the Smart Walker prototype is instrumented with brakes that can slow down the walker and lock the wheels, and a seat on which the user can rest when desired.

2.1 Project Goals

The goal of the Smart Walker project is to integrate knowledge from system design engineering, computer science and kinesiology to build an intelligent mobility aid which assists users, caregivers and clinicians in the following ways:

As a *user companion*, the Smart Walker can continuously monitor the user's physical and mental conditions. The key step in assessing these high-level states is the ability to recognize the activity of the user from the stream of sensor measurements. Based on this context information, quantitive measures such as the variation of the Center of Pressure (COP, see Sec. 3.1) or Walker User Risk Index curves (see [3]) can be considered in order to assess the condition of the user.

As a *caregivers' support*, the Smart Walker can take charge of monitoring the user's motoric capabilities and supervising the execution of daily excercises. Typically, it is impossible for caregivers to continuously attend their patients. On the other hand, self-assessments of patients are often unreliable, either due to poor memory or in order to avoid therapeutic interventions. Thus, the Smart Walker can help caregivers to obtain a complete and valid assessment of the user's condition. Again, a key step in providing this functionality is the agent's ability to recognize the user activity from the sensor data.

As a *clinical assistant*, the Smart Walker can provide clinicians and kinesiologists with longitudinal data of the physical and mental conditions of walker users. In contrast to tests performed in a clinical

setting, this data is collected in the users' everyday-life environment and over continuous periods of time. The data can be used for diagnosis, e.g., whether a non-walker user should be presribed a walker or a walker user should be prescribed a wheel chair; further applications are the evaluation of therapeutic interventions, and the development of future rolling walker designs.

The key problem in implementing these roles is to develop algorithms for the recognition of user activities from the stream of sensor measurements. We discuss different approaches to this problem in Sec. 3. In our future work, we plan to equip the Smart Walker with actuators and an audio system to provide active assistance. For example, actuating the brakes can be used to immobilize the walker when the user is sitting on the walker; furthermore, it allows the Smart Walker to influence the steering by inducing more resistance on one side, which eases turns in that direction. The audio system can be used for giving acoustic prompts and alerts, e.g., in case of emergency. Of course, a precondition for implementing such active roles is that the algorithms for activity recognition are as accurate and robust as possible.

2.2 Related Work

The idea to instrument rolling walkers with intelligent computer technology has attracted considerable attention in the past decade. Various research groups have instrumented walkers with sensors and actuators to support users and caregivers. The walker in [4] provides navigation assistance via a display for providing directions to the user, and an (optional) active drive mechanism. The walker uses pre-defined environment maps and learns a hierarchical model of the user's daily walking routines, hence it is unable to provide assistance in an unknown environment or when the user's walking routines change with time. The walker in [5] is instrumented with laser range finders to measure the distance between the walker and the user, and with wheel encoders to measure the velocity. It offers navigation assistance via servo brakes mounted at the rear wheels, however, it can distinguish only three different user states (walking, stopping, state of emergency). The walker in [6] is equipped with load sensors at the handles which allows to detect heel strikes, toe-off events and left/right single support phases. In contrast to our work, this walker is unable to recognize higher-level activities, such as turning left/right or walking up/down a ramp.

3 Activity Recognition

As we have seen in the previous section, a key step in building the Smart Walker is to develop algorithms for recognizing user activities from the sensor measurements. In this section, we first give a description of the sensor measurements and certain statistics derived therefrom, and then review algorithms for activity recognition based on Hidden Markov Models. In Sec. 3.3 we explain algorithms based on Conditional Random Fields; in particular, we present our idea to discriminate between activities using feature functions based on thresholding. For examples of activities that we wish to recognize, see the description of our experiments in Sec. 4.

3.1 Sensor Measurements and Derived Statistics

The raw sensor data of the Smart Walker consists of 8 measurements at each time point: $x_t^{\text{dist.}}$, the *walked distance* measured by a wheel encoder; $x_t^{\text{fr.le.}}$, $x_t^{\text{fr.ri.}}$, $x_t^{\text{re.le.}}$, $x_t^{\text{re.ri.}}$, the *load* on the four wheels (*front/rear* and *left/right*); $x_t^{\text{x-acc.}}$, $x_t^{\text{y-acc.}}$, $x_t^{\text{z-acc.}}$, the *acceleration* in *x*-, *y*-, *z*-direction. The measurements are digitalized with 16-bit resolution and 50-Hz sampling, so we obtain 50 data points per second and sensor, where the measurements range between 0 and $2^{16}-1$. In particular, $x_t^{\text{dist.}}$ is the walked distance modulo 2^{16}, where walking backwards results in a decrease of $x_t^{\text{dist.}}$.

From the raw sensor data we compute the following measures: the *speed* of the walker $x_t^{\text{speed}} = x_t^{\text{dist.}} - x_{t-1}^{\text{dist.}}$ where we add (substract) 2^{16} if an overflow (underflow) of $x_t^{\text{dist.}}$ is detected; the *total load* on the four wheels $x_t^{\text{tot. load}} = x_t^{\text{fr.le.}} + x_t^{\text{fr.ri.}} + x_t^{\text{re.le.}} + x_t^{\text{re.ri.}}$; the *frontal plane center of pressure* (FCOP), measuring the relative difference between the load on the left and the right wheels:

$$x_t^{\text{FCOP}} \quad = \quad \frac{d_f(x_t^{\text{fr.le.}} - x_t^{\text{fr.ri.}}) + d_r(x_t^{\text{re.le.}} - x_t^{\text{re.ri.}})}{x_t^{\text{fr.le.}} + x_t^{\text{fr.ri.}} + x_t^{\text{re.le.}} + x_t^{\text{re.ri.}}}$$

where the constants $d_f = 22.25$ and $d_r = 26.6$ are the distances of the front/rear load cells to the midline of the walker (in centimeters); the *sagittal plane center of pressure* (SCOP), measuring the

relative difference between the load on the rear and the front wheels:

$$x_t^{\text{SCOP}} \quad = \quad \frac{(x_t^{\text{fr.le.}} - x_t^{\text{re.le.}}) + (x_t^{\text{fr.ri.}} - x_t^{\text{re.ri.}})}{x_t^{\text{fr.le.}} + x_t^{\text{fr.ri.}} + x_t^{\text{re.le.}} + x_t^{\text{re.ri.}}}.$$

For the recognition of the user activity we use the measurements x_t^{speed}, $x_t^{\text{tot.load}}$, x_t^{FCOP}, x_t^{SCOP}, $x_t^{\text{x-acc.}}$, $x_t^{\text{y-acc.}}$, $x_t^{\text{z-acc.}}$. The advantage of using the measures FCOP and SCOP instead of the raw load sensor recordings is that they take into account the *relative* difference between the load on the left/right and rear/front wheels, so they do not depend on the body weight of the user. In order to include information on past measurements, we compute the mean and the variance over the previous w time points, for instance,

$$\mu_t^{\text{speed}}(w) \; = \; \frac{1}{w} \sum_{k=0}^{w-1} x_{t-k}^{\text{speed}} \quad \text{and} \quad \sigma_t^{\text{speed}}(w) \; = \; \frac{1}{w} \sum_{k=0}^{w-1} \big(x_{t-k}^{\text{speed}} - \mu_t^{\text{speed}}(w)\big)^2.$$

In our experiments, we chose the time horizons $w = 1, 5, 25$, that is, besides the actual measurements we consider past information up to half a second. Of course, the variances for $w = 1$ are equal to 0. After computing the means and variances, we obtain a 35-dimensional vector of observations at each time point t, namely, the means and variances of x_t^{speed}, $x_t^{\text{tot.load}}$, x_t^{FCOP}, x_t^{SCOP}, $x_t^{\text{x-acc.}}$, $x_t^{\text{y-acc.}}$, $x_t^{\text{z-acc.}}$ for the time horizons $w = 1, 5, 25$ (excluding the zero variances obtained for $w = 1$).

3.2 Hidden Markov Models

In [1] we compared the performance of various probabilistic models for activity recognition. While we found that Hidden Markov Models (HMMs) were outperformed by Conditional Random Fields, HMMs are still interesting because they allow for unsupervised learning. In an HMM, the sensor measurements are represented by a sequence of observations $\boldsymbol{X} = (X_1, \ldots, X_n)$ with values in some set $\mathcal{X}$, and the user activities by a sequence of hidden states $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ with values in some finite set $\mathcal{Y}$. In [1] we discretized the sensor measurements so that $\mathcal{X}$ was finite. The HMM is parameterized by the *initial distribution* $\pi(y)$, the *transition probabilities* $\theta(y', y)$, and the *emission probabilities* $\phi(x, y)$. The joint probability of the observations $\boldsymbol{x} = (x_1, \ldots, x_n)$ and the hidden states $\boldsymbol{y} = (x_1, \ldots, x_n)$ in this model is given by

$$P(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Y} = \boldsymbol{y}) \quad = \quad \pi(y_1)\Big(\prod_{t=2}^{n} \theta(y_{t-1}, y_t)\Big)\Big(\prod_{t=1}^{n} \phi(x_t, y_t)\Big).$$

If labeled training data $(\boldsymbol{x}, \boldsymbol{y})$ with $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$ is available, then the HMM parameters can be learned using Maximum Likelihood estimation. If the training data is unlabeled, i.e., $\boldsymbol{y}$ is unavailable, then a common approach is to use the EM algorithm. In [1] we also considered an alternative method based on Bayesian learning, where we put Dirichlet priors on the multinomial distributions $\pi(y)$, $\theta(y', y)$ and $\phi(x, y)$ and used Gibbs sampling to estimate the posterior over the hidden states and the parameters.

3.3 Conditional Random Fields

In contrast to HMMs, Conditional Random Fields (CRFs) are discriminative and model the *conditional* distribution of $\boldsymbol{Y}$ given $\boldsymbol{X}$ (see [7]). An advantage of CRFs is that they do not make conditional independence assumptions on the observations. CRFs are parameterized by vectors of real-valued *feature functions* $\boldsymbol{f}$ and *model weights* $\boldsymbol{\lambda}$. For any $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$, the probability of $\boldsymbol{Y} = \boldsymbol{y}$ conditional on $\boldsymbol{X} = \boldsymbol{x}$ is given by

$$P(\boldsymbol{Y} = \boldsymbol{y} \,|\, \boldsymbol{X} = \boldsymbol{x}) \quad \propto \quad \exp\big(\boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})\big). \tag{1}$$

A particularly important class of CRFs are *linear-chain* models where the feature functions are allowed to depend on at most two consecutive labels. In that case, the inner product of model weights and feature functions in (1) can be written as the sum of weighted state and transition feature functions, namely,

$$\boldsymbol{\lambda}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}) \quad = \quad \sum_{t=1}^{n} \boldsymbol{\mu}^T \boldsymbol{f}^{\text{state}}(x_t, y_t) \; + \; \sum_{t=2}^{n} \boldsymbol{\nu}^T \boldsymbol{f}^{\text{trans}}(x_t, y_{t-1}, y_t), \tag{2}$$

where we suppose that the transition feature functions are non-constant both in y_{t-1} and y_t, so there is no overlap with the state features. The assumption that the feature functions only depend on the current observation x_t is without loss of generality: if they depended on, say, $x_{t-l}, \ldots, x_{t+l}$, then simply consider the modified observations $\tilde{x}_t = (x_{t-l}, \ldots, x_{t+l})$ instead of x_t.

The feature functions can be used to incorporate domain-specific knowledge. A typical choice are binary feature functions; depending on whether the associated model weight is positive or negative, the co-occurrence of certain observations and labels then increases or decreases the conditional probability of a particular label sequence. Given a fixed set of feature functions, the model weights are usually learned from labeled training data by maximizing the conditional log-likelihood function. An efficient dynamic programming algorithm is available to compute the marginal distributions of $\boldsymbol{Y}$ given $\boldsymbol{X}$. See [8] for more details.

3.4 Feature Functions Based on Discriminant Rules

A crucial step in applying CRFs is the selection of the state and transition feature functions in (2). In [1] we used feature functions based on thresholding in order to discriminate between activities, where we suggested to choose the threshold values manually by a visual inspection of the data. In the present paper we consider more generally *discriminant rules* which, besides thresholds, may also take into account the raw observations, or "smooth" thresholds based on sigmoid functions. In the following, suppose that the set of labels is given by $\mathcal{Y} = \{1, \ldots, k\}$.

Let us explain the idea of discriminant rules for the state feature functions in (2). Suppose we want to compute $\boldsymbol{\mu}^T \boldsymbol{f}^{\text{state}}(x_t, y_t)$ for $y_t = i$. Intuitively, the larger this quantity, the higher is the compatibility of the events $X_t = x_t$ and $Y_t = i$. The idea of discriminant rules is to consider any potential alternative, $Y_t = j$ with $j \in \mathcal{Y} \setminus \{i\}$, and to evaluate whether $X_t = x_t$ is more compatible with $Y_t = i$ or $Y_t = j$. Mathematically, this procedure can be written as

$$\boldsymbol{\mu}^T \boldsymbol{f}^{\text{state}}(x_t, i) \quad = \quad \sum_{j \in \mathcal{Y} \setminus \{i\}} \boldsymbol{\mu}_{ij}^T \boldsymbol{d}_{ij}(x_t) \tag{3}$$

where $\boldsymbol{d}_{ij}(\cdot)$ is a vector of functions discriminating between i and j, which is associated with the model weights $\boldsymbol{\mu}_{ij}$. Examples will be considered in the sections that follow. Similarly, discriminant rules can be introduced for the transition feature functions in (2). In our experiments in Sec. 4, however, we consider a very simple transition model where

$$\boldsymbol{\nu}^T \boldsymbol{f}^{\text{trans}}(x_t, h, i) \quad = \quad \nu \, \mathbf{1}(h = i) \tag{4}$$

for $h, i \in \mathcal{Y}$. Note that ν is a scalar weight and the transition feature function simply reflects whether or not an activity persists. The reason for using this simplistic model is that we want to avoid a bias towards certain transitions due to the design of the experimental courses.

3.4.1 Binary Threshold Functions

A simple type of discriminant rules, which we have first considered in [1], is obtained by comparing the observations to fixed threshold values. Suppose that the observations are one-dimensional and real-valued; in the case of multi-dimensional observations, we consider thresholds separately for each component of x_t. Let $\mathbf{1}(\cdot)$ denote the function which evaluates to 1 if the statement in the brackets is true, and to 0, otherwise. For each pair of labels $i, j \in \mathcal{Y}$ with $i \neq j$, introduce a threshold τ_{ij} and consider the weighted discriminant rules

$$\boldsymbol{\mu}_{ij}^T \boldsymbol{d}_{ij}(x_t) \quad = \quad \mu_{ij}^{(g)} \mathbf{1}(x_t \geq \tau_{ij}) + \mu_{ij}^{(l)} \mathbf{1}(x_t < \tau_{ij}). \tag{5}$$

In [1] we suggested to choose the thresholds manually by a visual inspection of the observations. Although the results that we obtained were encouraging, this approach has the disadvantages of being time-consuming (wherefore our observations in [1] did not include the means and variances over past measurements) and subjective. In this paper, we propose to select the tresholds using Linear Regression (LR) models.

For the rest of this section, suppose we are given training data $(\boldsymbol{x}, \boldsymbol{y})$ where $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$. Using a LR model, a standard way for selecting τ_{ij} is to regress the values x_t for which $y_t = i$ on 1, and the values x_t for which $y_t = j$ on -1. Then the resulting LR coefficients

α_{ij} and β_{ij} are used to compute the threshold $\tau_{ij} = -\alpha_{ij}/\beta_{ij}$ (see, e.g., Chapter 4 in [9]). Note that this threshold implicitly models the a priori probabilities of $Y_t = i$ and $Y_t = j$. In the present paper, we consider thresholds based on normalized numbers to avoid a bias towards labels with a high frequency in the training data set. Based on the normalized numbers, the threshold is simply given by

$$\tau_{ij} \quad = \quad \frac{1}{2}\left(\frac{1}{n_i}\sum_{t=1}^{n}\mathbf{1}(y_t = i)\, x_t \;+\; \frac{1}{n_j}\sum_{t=1}^{n}\mathbf{1}(y_t = j)\, x_t\right)$$

where n_i, n_j denote the number of points in the training data set for which $y_t = i$ and $y_t = j$, respectively.

3.4.2 Sigmoid Threshold Functions

While a major advantage of binary threshold functions is their computational simplicity, they do not reflect whether x_t exceeds τ_{ij} only slightly or by a wide margin. A natural way to incorporate this information is using the sigmoid function $\text{sigm}(x) = 1/(1 + e^{-x})$ which can be regarded as a continuous analogue of the binary-valued function $\mathbf{1}(x \geq 0)$. The weighted discriminant rules in (3) are then given by

$$\boldsymbol{\mu}_{ij}^T \boldsymbol{d}_{ij}(x_t) \;=\; \mu_{ij}^{(g)}\text{sigm}(\gamma_{ij}(x_t - \tau_{ij})) + \mu_{ij}^{(l)}\text{sigm}(\gamma_{ij}(\tau_{ij} - x_t))$$

where γ_{ij} is an additional scaling parameter. We propose to determine γ_{ij} by maximizing

$$\mathcal{L}(\gamma_{ij}) \quad = \quad \sum_{t=1}^{n}\mathbf{1}(y_t = i)\,\gamma_{ij}(x_t - \tau_{ij}) - \sum_{t=1}^{n}\mathbf{1}(y_t = i \text{ or } y_t = j)\,\log\big(1 + \exp(\gamma_{ij}(x_t - \tau_{ij})\big).$$

Note that $\mathcal{L}(\gamma_{ij})$ is concave, so it has a unique maximum which can be found using Newton's method. In a logistic regression model, $\mathcal{L}(\gamma_{ij})$ is the log-likelihood of the intercept $-\gamma_{ij}\tau_{ij}$ and the slope γ_{ij} (see, e.g., Chapter 4 in [9]). Thus, maximizing $\mathcal{L}(\gamma_{ij})$ is equivalent to finding the maximum likelihood estimates of a logistic regression model where not the whole parameter space is explored but only a linear subspace.

3.4.3 Using Raw Observations

Finally, let us consider discriminant rules which take the raw observations into account. By μ and σ we denote the sample mean and standard deviation of x_t. Again, we first suppose that $\mathcal{X}$ is one-dimensional and consider discriminant rules separately for each component of x_t in the multi-dimensional case. Using raw observations, the weighted discriminant rules in (3) become

$$\boldsymbol{\mu}_{ij}^T \boldsymbol{d}_{ij}(x_t) \quad = \quad \mu_{ij}^{(ic)} + \mu_{ij}^{(sl)}\big(\sigma^{-1}(x_t - \mu)\big).$$

Note that the standardization of x_t is necessary to avoid large values of $\mu_{ij}^{(ic)}$ and $\mu_{ij}^{(sl)}$ to be penalized during the training of the CRF.

3.5 Feature Functions Using Data Binning

As an alternative to feature functions based on discriminant rules, we consider *data binning*. Suppose the domain of observations $\mathcal{X}$ is one-dimensional. Data binning means to partition $\mathcal{X}$ into a finite number of subsets $A_1, \ldots, A_m$. We use the indicators of these subsets as feature functions, so that

$$\boldsymbol{\mu}^T \boldsymbol{f}^{\text{state}}(x_t, i) \quad = \quad \sum_{j=1}^{m}\mu_{ij}\,\mathbf{1}(x_t \in A_j)$$

for $i \in \mathcal{Y}$. When $\mathcal{X}$ is multi-dimensional, we partition each subdomain separately and sum up the corresponding weighted features. For our experiments in Sec. 4, we chose m proportional to k, the number of different labels. We consider two strategies for selecting the sets $A_1, \ldots, A_m$: Writing x_{min} and x_{max} for the minimum and the maximum value of x_t in the training data set, the first strategy is to divide $[x_{\text{min}}, x_{\text{max}}]$ into k intervals of equal size; the second strategy is to divide $[x_{\text{min}}, x_{\text{max}}]$ into k intervals such that each interval contains the same number of points in the training data set.

Table 1: Accuracy for Experiment 1 (in %)

	NT	ST	FW	LE	RI	BW	TF	Total	S.E.
CRF+ManTh	63.9	32.7	91.5	55.9	52.9	88.5	18.3	77.1	8.6
CRF+BinTh	80.9	70.2	95.2	74.4	64.7	91.2	60.9	87.0	3.4
CRF+SigmTh	88.3	71.0	95.9	77.0	70.5	92.3	56.2	88.7	3.5
CRF+Raw	91.4	56.9	96.0	70.5	60.3	88.2	42.4	86.2	3.1
CRF+EqBin	75.4	72.7	94.9	74.2	66.7	92.3	53.0	86.3	3.9
CRF+PrBin	89.0	67.4	92.4	71.8	66.4	87.3	52.8	85.2	6.0

Table 2: Accuracy for Experiment 2 (in %)

	ST	FW	LE	RI	SI	UR	DR	UC	DC	Total	S.E.
CRF+ManTh	90.5	82.7	53.3	30.8	99.1	35.1	30.1	43.9	25.7	77.5	5.5
CRF+BinTh	89.2	82.0	56.2	51.8	98.0	65.4	54.1	60.4	55.1	82.5	3.9
CRF+SigmTh	89.9	85.1	63.0	51.1	99.0	79.2	57.9	60.6	54.1	84.9	4.1
CRF+Raw	89.4	84.7	58.1	46.0	99.1	67.4	62.8	55.2	47.3	83.4	4.7
CRF+EqBin	88.7	84.8	57.5	53.4	98.7	72.3	52.0	55.9	57.8	83.9	3.6
CRF+PrBin	85.3	76.7	52.0	52.1	88.4	67.1	56.7	48.7	55.1	77.3	8.3

4 Experimental Results

In this section, we evaluate the proposed algorithms on real user data. The data was collected in two different experimental settings which we describe in Sec. 4.1. The results are discussed in Sec. 4.2.

4.1 Experimental settings

In the first experiment, 12 healthy young subjects (19-53 years old) were asked to walk twice through a predefined course. At the beginning, the participants were sitting on a chair and not touching the walker (**NT**); then they were standing and holding the walker (**ST**); they had to walk forward/backwards (**FW**/**BW**) and execute left/right turns (**LE**/**RI**) to navigate through the course; finally, they were asked to release the walker and sit back on the chair. The behavior of getting up/down and taking/releasing the walker is subsumed under Transferring (**TF**). In total, the data set consists of 98,259 time points, corresponding to a duration of approx. 33 minutes.

The participants of the second experiment were 15 older adults (80-97 years old), 8 of which were residents of a long term health care facility and regularly using a walker. The participants were asked to walk through two different courses. In the first course, the participants had to walk forward (**FW**) and perform left/right turns (**LE**/**RI**); meanwhile, they were asked to execute real-life tasks like picking up objects from the ground, turning in a confined space, or walking fast as if they were trying to catch a bus. In the second course, the participants had to go up/down a ramp (**UR**/**DR**) and a curb (**UC**/**DC**). We also recorded some spontaneous activity between the two courses; for example, the participants sat on the walker (**SI**) after performing the first course and then navigated to the second course. In total, the data set consists of 130,195 time points, corresponding to a duration of approx. 44 minutes.

The training data collected in the two experiments was labeled manually using the recordings of a video camera mounted on the walker. We compare six different methods for activity recognition: **CRF+ManTh** uses binary threshold functions as described in Sec. 3.4.1. Same as in [1], we chose the threshold values manually by a visual inspection of the data. As this procedure is time-consuming, the model does not include the means and variances over past measurements. **CRF+BinTh** uses binary threshold functions with the thresholds obtained by LR models. **CRF+SigmTh** uses the sigmoid threshold functions. **CRF+Raw** uses discriminant rules based on raw observations. **CRF+EqBin** uses feature functions based on data binning with bins of equal size. **CRF+PrBin** uses data binning with the bins all containing the same number of observations.

We use leave-one-out cross-validation to evaluate the performance of the methods. The labels are predicted by maximizing the conditional marginal distributions. Table 1 and 2 show the percentage

of time points for which an activity is predicted correctly; the last two columns display the total accuracy and the standard error.

4.2 Discussion

As can be seen, CRF+SigmTh achieves the best performance among all six methods with an overall accuracy of 88.7% in Experiment 1, and 84.9% in Experiment 2. Except for the CRF+EqBin method in Experiment 2, these differences in the performance are all statistically significant (one-tailed Wilcoxon signed-rank test, significance level $\alpha = 0.05$). The second-best methods are CRF+BinTh in Experiment 1, and CRF+EqBin in Experiment 2. In both experiments, the CRFs with automatically chosen thresholds perform significantly better than the CRFs with manually chosen ones.

In Experiment 1, all methods have problems to correctly identify transfers (TF), which is not surprising as this is an intermediate activity between not touching the walker (NT) and standing (ST). Turning left and right (LE/RI) is sometimes confused with walking forward (WF), however, even for a human observer it is often hard to tell when a turn exactly starts or ends. Overall, the results for Experiment 1 are better than for Experiment 2. A possible explanation is that the participants in Experiment 1 performed the course twice, so the training data set always includes one recording of the person for which the activity is currently predicted. Furthermore, some of the activities exhibited during Experiment 2 are highly individual; for example, the participants used very different strategies to move the walker up and down the curb. Even for simple activities there is a higher variability in Experiment 2, as the participants were instructed to walk at different speeds or pick up objects from the ground.

5 Conclusions

In this paper we presented the Smart Walker, a four-wheel rolling walker equipped with sensors to provide assistance to users, caregivers and clinicians. A key problem in implementing assistive functionalities is the recognition of user activities from the sensor measurements. For this purpose, we presented and evaluated different methods based on Conditional Random Fields. Experiments with real user data showed that these methods can achieve a good accuracy; the best results were obtained by "smooth" thresholding based on sigmoid functions.

Fundamental questions for our future research are how the accuracy achieved by the algorithms compares to the agreement among different human labelers. In cooperation with kinesiologists we will evaluate how much accuracy is actually needed for providing robust measures, e.g., of the user's stability, and develop strategies to cope with the basic uncertainty about the "true" user activity. Finally, we will consider individualized user models which can incorporate prior information, e.g., about the user's body weight, his physical and mental conditions, or whether he is left- or right-handed.

Acknowledgements

The research of Mathieu Sinn was supported by a Postdoctoral Research Fellowship (PDRF) of the Canadian Bureau for International Education (CBIE).

References

[1] Omar, F., Sinn, M., Truszkowski, J., Poupart, P., Tung, J. & Caine, A. (2010). Comparative analysis of probabilistic models for activity recognition with an instrumented walker. *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*.

[2] Tung, J., Gage, W.H., Zabjek, K., Brooks, D., Maki, B.E., Mihailidis, A., Fernie, G. & McIlroy, W.E. (2007). iWalker: a 'real-world' mobility assessment tool. *Proceedings of the 30th Canadian Medical & Biological Engineering Society*.

[3] Pardo, R.D., Deathe, A.B. & Winter, D.A. (1993). Walker user risk index, a method for quantifying stability in walker users. *American Journal of Physical Medicine and Rehabilitation* 72(2):301-305.

[4] Glover, J., Thrun, S. & Matthews, J. (2004). Learning user models of mobility-related activities through instrumented walking aids. *Proceedings of the IEEE International Conference on Robotics and Automation*.

[5] Hirata, Y., Muraki, A. & Kosuge, K. (2006). Motion control of intelligent walkers based on renew of estimation parameters for user state. *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*.

[6] Alwan, M., Wasson, G., Sheth, P., Ledoux, A. & Huang, C. (2004). Passive derivation of basic walker-assisted gait characteristics from measured forces and moments. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*.

[7] Lafferty, J., McCallum, A. & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings ICML01*.

[8] Sutton, C. & McCallum, A. (2006). An introduction to conditional random fields for relational learning. *In:* L. Getoor and B. Taskar (eds.), *Introduction to Statistical Relational Learning*. Cambridge, MA: MIT Press.

[9] Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction. Second Edition.* New York, NY: Springer.