Improved Activity Recognition via Kalman Smoothing and Multiclass Linear Discriminant Analysis

Neil Dhir[1,†] and Frank Wood[1]

Abstract— **Improving activity recognition, with special focus on fall-detection, is the subject of this study. We show that Kalman smoothed in-painting of missing pose information and task-specific dimensionality reduction of activity feature vectors leads to significantly improved activity classification performance. We illustrate our findings by applying common classification algorithms to dimensionally reduced feature vectors, and compare our accuracy to previous work. In part two we investigate our methods on a small subset of the data, in order to ascertain what accuracy performance is achievable with the smallest amount of information available.**

I. Introduction

We investigate task-specific activity recognition, with the goal of improving classification performance. We identify two areas where improvements can be exploited to improve activity classification accuracy. First, we propose a smoothing model which can accurately in-paint missing pose data. Second, we suggest transforming high-dimensional activity feature vectors into a low-dimensional eigenspace using supervised dimensionality reduction. While our ultimate aim is a real-time system that infers pose continuously from low-rank observations; there is immediate practical utility associated with the smoothing methods developed in this paper, particularly when used in conjunction with Zhang et al.'s [1] technique for detecting when a serious fall may have occurred. In this system, high confidence fall activity recognition, preceding a period of constant negative gravitational acceleration is necessary. The literature provides many techniques for activity detection see e.g. [2]–[4], but few have focused on smoothing and dimensionality reduction for pose models as we do here. In the literature, feature vectors used for classification are typically very large; Luštrek and Kaluža [5], use feature vectors ranging in dimensionality from 240 to 2,700. With careful use of cross-validation and regularization, high-dimensional feature vectors can be used, with high accuracy, for activity recognition. Higher accuracy can be achieved still by using task-specific regularization.

II. Previous Work

Zhang et al. [1] present a system with real-time classification of human movements based on data collected from a smartphone mounted on the subject's waist. Using their algorithm, body motion is labeled with five categories. Their system measures acceleration, based on a tri-axial accelerometer, where the system is continuously monitoring changes to the gravitational acceleration parameter. If the smartphone acceleration is near absolute gravitational acceleration, for the duration of a second or more, the subject is considered motionless. Once this is detected the data is backdated for 1.5s [1], and that section of the data is used as the sequence input for two classification algorithms which are employed to determine if there actually was a fall event (one of the five activity labels). But the feature vector used in the backdated period has a size of 192, which is large and could lead to problems with overfitting, especially as the number of training samples used was low at $N \approx 730$.

A very different approach is taken by Olivieri et al. [6]. They demonstrate a low-cost, home-based health care system based on automatic imaging recognition from video sequences. They propose a software package based upon a spatio-temporal motion representation, called Motion Vector Flow Instance (MVFI) templates, that capture relevant velocity information by extracting dense optical flow from video sequences of human actions. Automatic recognition is achieved by first projecting each human action video sequence, consisting of approximately 100 images, into a canonical eigenspace (i.e. dimensionality reduction), and then performing supervised learning to train multiple actions from a large video database. The MVFI is approximately 100% accurate in binary classification between fall activities and other actions [6], where they show that their methods is robust and can perform in real-time.

Bourke and Lyons [7] describe a threshold-based algorithm, to distinguish between activities of daily living (ADL) and falls. A gyroscope based fall-detection sensor array is used. Data analysis was performed to determine the angular accelerations, angular velocities and changes in trunk angle recorded, during eight different fall and ADL types. They summarize their approach as thus; fall detection is achieved by applying a threshold to the peak values from the resultant angular velocity signals, recorded from fall and ADL data. Consequently by setting the threshold values just below the lowest recorded fall peak values of the studied parameters, this ensures that any value which exceeds these limits will be recorded as a fall. They achieved 100% specificity. That said, their method relies heavily on test data (240 recorded simulated falls) to obtain these thresholds, highlighting a drawback of this supervised learning system.

For more recent studies, consider the work done by Albert et al. [8]. In their study, 15 subjects were asked to simulate four different types of falls; left and right lateral, forward trips, and backward slips, while wearing smart phones and dedicated accelerometers. Nine subjects also wore the devices for ten days, to provide data for comparison with the

[1]Department of Engineering Science, University of Oxford, Parks Road, OX1 3PJ Oxford, United Kingdom. [†]Corresponding author: neild@robots.ox.ac.uk

simulated falls. Five classification schemes were applied to a large time-series feature set to detect falls. Their results are robust, with both the Support Vector Machine (SVM) and the Sparse Multinomial Logistic Regression classifier, achieving accuracies close to 98% for pooled subject data when using 10-fold cross-validation, while that accuracy decreased to 97% when subject-wise cross-validation was used [8].

We will build upon the work by Luštrek and Kaluža [5] by using the same body-centered coordinate system and classification schemes. Their work is relevant to this study because their dataset features significant sections of missing information, such that 7.5% of their dataset is irretrievable due to sensor failure. But of main importance is the dimensionality reduction that we shall investigate in order to improve classification performance, with particular attention to their experiments where a feature vector with dimensionality of 720 was used. Smoothed data is found from Kalman Smoothing (KS) which we describe in section III-C. On this data we implement several data-complete canonical transformations, with special focus on linear discriminant analysis (LDA), which are described in section III-D. Our experiments are outlined in section IV and finally a discussion and conclusion follow in section V.

III. Methods

Bold lowercase Roman letters denote vectors and scalar variables are denoted by simple Roman letters. Formally the problem can be stated as follows; assume there is a labeled training set $\mathcal{S} = \{(\boldsymbol{\phi}_1, y_1), \ldots, (\boldsymbol{\phi}_N, y_N)\}$, where $|\mathcal{S}| = N$, $j \in \mathcal{J} = \{1, \ldots, N\}$ and $\boldsymbol{\phi}_j \in \mathcal{X} = \{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N\}$ are the task-specific feature vectors, with the activity (classes) given by $y_j \in \mathcal{Y} = \{1, \ldots, K\}$. The training set is such that $\mathcal{X} \subseteq \mathbb{R}^D$. A classifier is then a function $h : \mathcal{X} \to \mathcal{Y}$, that maps an instance $\boldsymbol{\phi}_j$ to a label $\hat{y}_j = h(\boldsymbol{\phi}_j)$. The accuracy of a classifier is evaluated using a loss function $l(h(\boldsymbol{\phi}_j), y)$, which measures the disparity between the predicted actual label set [9]. The following sections describe how the feature vectors $\boldsymbol{\phi}_j$ are chosen and evaluated.

A. Dataset

The dataset was collected with the Ubisense real time infrared motion capture system [5], consisting of six infrared cameras and infrared light sources. Three volunteers were equipped with 12 infrared reflectors. The markers were attached to the ankles, knees, hips, shoulders, elbows and wrists (see Figure 2). They were tracked with the cameras, and their three dimensional coordinates were measured. Artificial Gaussian noise was added according to the specifications of the system's manufacturer. The standard deviation of the noise was 43.6mm horizontally and 54.4mm vertically [5]. Data was collected at 60Hz which was downsampled to 10Hz (to simulate typical smart phone sample frequency). The recording coordinate system was right-handed with the y-axis as the vertical axis and the other two axes aligned with the square walls of the room. A coordinate transformation was used to map the exogenous reference frame to an endogenous frame, where the y-axis passes through the two hip tags, the z-axis becomes the vertical axis, with the origin located between the two hip tags and finally the x-axis is normal to the yz-plane.

Eight different short movement scenarios were repeated ten times by each subject: walking in a straight line, walking in a straight line whilst limping on the right leg, walking with a heavy burden in the right hand, walking in a circle, walking then stopping and resuming walking, falling in various fashions, lying down (which could be mistaken for falling) and sitting down (which also could be mistaken for falling). Each scenario was labeled with one or more activities: falling, the process of lying down, the process of sitting down, walking, sitting (stationary) and lying (stationary).

B. Attribute Set

Let the collection of body tags be in the set $i \in \mathcal{I} = \{1, \ldots, 12\}$ and the time-frame $t \in \mathcal{T} = \{1, \ldots, 10\}$, where the attribute vector, from which the classifier infers the subject's activity, consists of ten consecutive snapshots of the subject's posture, describing one second of activity. Luštrek and Kaluža [5] used several attribute sets, clean and noisy, exogenous and endogenous. We focus our work on noisy observations of joint positions in endogenous coordinates because we envision, ultimately, inferring pose from sensors attached to the body.

Let $\mathbf{u}_i^t$ denote the coordinates of the arbitrary tag i at time-frame t. The feature vector $\boldsymbol{\phi}_j$ is then designed by letting

$$\boldsymbol{\psi}_{i,j}^t = \left[\mathbf{u}_i^t \in \mathbb{R}^3, \|\dot{\mathbf{u}}_i^t\| \in \mathbb{R}^1, \alpha_i^t \in \mathbb{R}^1, \beta_i^t \in \mathbb{R}^1\right]_j^{\mathsf{T}} \ \forall i, j, t \quad (1)$$

$$\boldsymbol{\phi}_j = \underbrace{\left[\boldsymbol{\psi}_{1,j}^1, \ldots, \boldsymbol{\psi}_{12,j}^1, \ldots \boldsymbol{\psi}_{i,j}^t \ldots, \boldsymbol{\psi}_{1,j}^{10}, \ldots, \boldsymbol{\psi}_{12,j}^{10}\right]}_{\text{1s of activity}}, \quad (2)$$

where α and β are the angles of movement between the tag and the z-axis, and the tag and the xz-plane respectively. We solve the problem of missing data, by smoothing the original dataset, from which feature vectors are generated.

C. Kalman Smoothing

Kalman filters are typically used for online inference problems. But in an offline setting, such as in our problem domain, we can go one step further and condition on past *and* future observations (i.e. the tag coordinates), leading to our uncertainty being significantly reduced and our posterior state beliefs (i.e. the missing tag coordinates due to sensor failure) improved [10]. Because linear Gaussian state-space models (also known as linear dynamical systems) can be represented by a tree-structured directed graph, inference problems are solved efficiently using the sum-product algorithm [11], the forwards and backwards recursions of which are known as Kalman Smoothing.

Because the model has linear-Gaussian conditional distributions, the transition and emission distributions (which define a first order Markov model), of the state and observations (recall that $\mathbf{z}$ are the inferred tag-coordinates, conditioned on the available data $\mathbf{x}$), can be written [11] in the general linear

form

$$\begin{aligned} \mathbf{z}_t &= \mathbf{A}\mathbf{z}_{t-1} + \mathbf{w}_t \quad & \mathbf{w} &\sim \mathcal{N}(\mathbf{w}|\mathbf{0},\boldsymbol{\Gamma}) \\ \mathbf{x}_t &= \mathbf{C}\mathbf{z}_t + \mathbf{v}_t & \mathbf{v} &\sim \mathcal{N}(\mathbf{v}|\mathbf{0},\boldsymbol{\Sigma}) \\ \mathbf{z}_1 &= \boldsymbol{\mu}_0 + \mathbf{u} & \mathbf{u} &\sim \mathcal{N}(\mathbf{u}|\mathbf{0},\mathbf{P}_0), \end{aligned}$$

where we determine the parameters of the model $\boldsymbol{\theta} = \{\mathbf{A},\boldsymbol{\Gamma},\mathbf{C},\boldsymbol{\Sigma},\boldsymbol{\mu}_0,\mathbf{P}_0\}$, using maximum likelihood through the expectation-maximisation algorithm.

D. Dimensionality Reduction

For each scenario iteration (ten for each subject) $\boldsymbol{\theta}$ was found through likelihood maximisation. Missing information, i.e. tag coordinates due to sensor-failure, were inpainted from the KS model and feature vectors were created from this smoothed dataset. As noted in the previous section, we focus our attention on a specific feature set used by Luštrek and Kaluža [5], where $\boldsymbol{\phi}_j \in \mathbb{R}^{720}\ \forall j$, and $N = 1,302$. The authors avoid overfitting by using cross-validation and regularization. We will investigate the latter further by investigating task-specific regularization by way of dimensionality reduction (DR) through canonical transformations. Six methods were investigated: Multiclass LDA, Principal Component Analysis (PCA), Factor Analysis (FA), Truncated Singular Value Decomposition (TSVD), Gaussian Random Projection (GRP) and Partial Least Squares Regression (PLSR) [12]. We provide a synopsis of LDA, as it was found to increase classification accuracy the most; we seek projection vectors $\mathbf{w}_k$, $k \in \{1,\ldots,|\mathcal{Y}|-1\}$, arranged by columns in a projection matrix $\mathbf{W}$. We are looking for a projection that maximizes the ratio of between-class to within-class scatter. It can be shown [13] that the optimal projection matrix $\mathbf{W}^*$ is the one whose columns are the eigenvectors corresponding to the largest eigenvalues λ_k, of the following generalized eigenvalue problem

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmax}} \frac{|\mathbf{W}^\mathsf{T}\mathbf{S}_B\mathbf{W}|}{|\mathbf{W}^\mathsf{T}\mathbf{S}_W\mathbf{W}|},$$

where $\mathbf{S}_W$ is the within-class scatter and $\mathbf{S}_B$ is the between-class scatter. The projections with maximum class separability information are the eigenvectors corresponding to the largest eigenvalues of $(\mathbf{S}_W^{-1}\mathbf{S}_B - \lambda_k)\mathbf{w}_k^* = 0$, where $\mathbf{w}_k^* \subset \mathbf{W}^*$ are the columns on the optimal projection matrix.

IV. Experiments

Classification experiments were carried out in the Waikato Environment for Knowledge Analysis (WEKA), an open-source suite of machine learning software written in Java.

Using our smoothed data and dimensionally reduced feature vectors, we compare performance with Luštrek and Kaluža [5]. In their study the authors used eight different classification schemes: Pruned C4.5 Decision Tree (C4.5), Propositional Rule Learner (PRL), Naive Bayes Classifier using Estimator Classes (NB), 3-Nearest Neighbors (3-NN), multiclass Support Vector Machine (SVM), Random Forest (RF), Bagging of the fast decision tree learner (Bag.) and boosting of a nominal class classifier (the fast decision tree learner) using the Adaboost M1 method (M1).

A. Single Tag Classification

In the second part of our experiments, information redundancy was investigated and physical dimensionality reduction studied. Now $\boldsymbol{\phi}_j \in \mathbb{R}^{60}$, instead of using the full set of tags, each tag was classified individually in order to ascertain which tags were most informative in terms of activity recognition, upon which LDA was implemented, see Figure 2. For all experiments accuracy was computed using ten-fold cross-validation, regularization and each classification scheme was repeated ten times, yielding 100 folds for each algorithm. Where N remained the same for all experiments in this paper.

V. Results

The results are summarized in Figure 1, where the full distribution of the classification accuracies have been summarized in a box plots for LDA, because it showed that the best performance compared to the others schemes (of which only the max accuracy is shown in Figure 1).

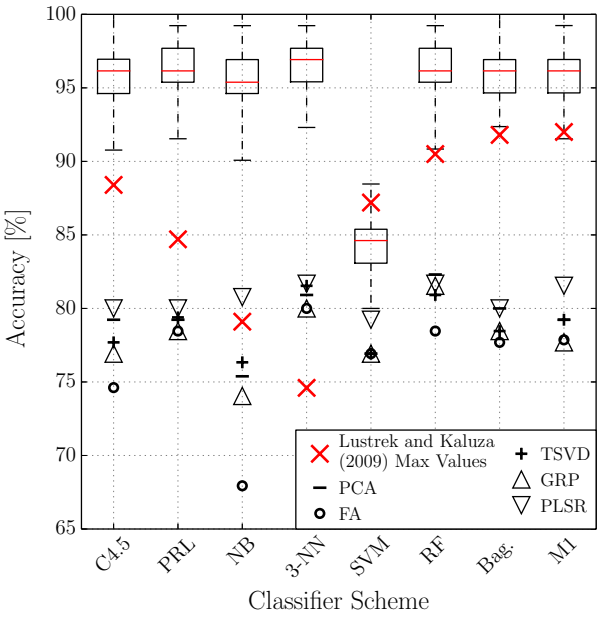

Fig. 1. Classification performance shown for all eight schemes, for all six dimensionality reduction methods. LDA results, being the best, are shown as box-plots over all folds.

A like-for-like comparison is only possible by considering the best accuracies in our experiments. These results can be seen at the maximum whiskers of each LDA box plot in Figure 1, they are summarized in Table I:

TABLE I
Classification scheme (max) accuracy [%] comparison

Study	C4.5	PRL	NB	3-NN	SVM	RF	Bag.	M1
Luštrek and Kaluža [5]	88.4	84.7	79.1	74.6	87.2	90.5	91.8	92.0
Dhir and Wood	100.0	99.2	99.2	99.2	88.5	99.2	100.0	99.2

Luštrek and Kaluža [5] only reported the best accuracies for their experiments, and not the distribution over all their folds, why only one point per scheme is shown in Figure 1. As is seen in Table I, the KS model, which also doubles as a

generative model, used for inferring missing data, produces data of high accuracy which validates its use as a generative motion model, the outputs of which function well as viable substitutes for classification. An accurate generative model which can be sampled accordingly to infer pose continuously from low-rank observations, has immediate practical utility since data collection becomes easier, faster and negates the use of complex feature selection to facilitate high classification accuracy. This means that equipping the user with the simplest of collection devices (e.g. smart phone), could be enough to infer complex motion and pose.

That being said, the increased classification accuracy is more likely to have been derived from dimensionality reduction methods. As can be seen multiclass LDA performs particularly well where LDA: $\boldsymbol{\phi}_j \in \mathbb{R}^{720} \rightarrow \boldsymbol{\phi}_j \in \mathbb{R}^5 \ \forall j$. LDA preserves as much of the class discriminating information as possible, by explicitly modeling the difference between them, and thus finding a linear combination of features which separates the activities. By using a new basis we project the dataset onto a dimensional space with more powerful data representation. We are performing offline inference, hence the means and covariances are known, making this method particularly suitable for our chosen application domain.

In the second part of our experiments, our feature vectors are still high dimensional ($\boldsymbol{\phi}_j \in \mathbb{R}^{60} \ \forall j$). But we investigate what can be considered physical DR, by treating each tag as independent and running the smoothing model and the classifiers on each independently. Where no information was passed between tags. The original and dimensionally reduced classification results are shown in Figure 2. The results are not as good as in part one. First, the KS model does not perform as well, owing to the lack of information passed to the model from the other tags, resulting in less exact inferred pose predictions. Moreover, the amount of

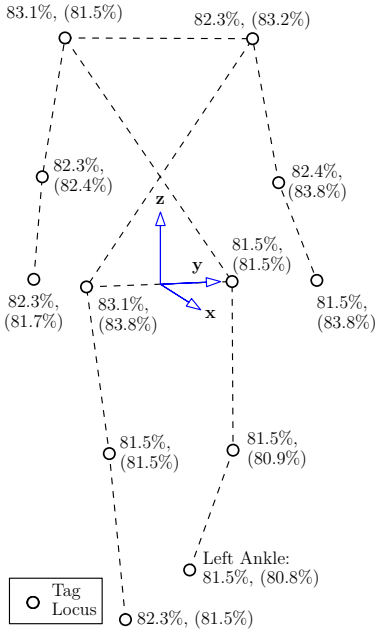

Fig. 2. Time frame illustration from a walking activity, with best individual tag classification accuracy quoted with each tag for feature vector of size $\boldsymbol{\phi}_j \in \mathbb{R}^{60} \ \forall j$, and of size $\boldsymbol{\phi}_j \in \mathbb{R}^5 \ \forall j$, within parentheses.

information contained in one second of activity, or ten sequential body poses, is not enough to produce classifiers which are discriminative enough to accurately categorize the activities. Having an average generative model coupled with an average discriminative classification performance, even with LDA (the minor difference between classification accuracy between LDA and the original feature vector size, would suggest that dimensionality reduction is not the foremost problem), suggests that other features need to be considered for single tags, or more tags used for these features in order to maximize the utility of the information used for classifying human motion models.

ACKNOWLEDGMENT

The authors acknowledge the support of the RCUK Digital Economy Programme grant number EP/G036861/1 (Oxford Centre for Doctoral Training in Healthcare Innovation).

REFERENCES

[1] T. Zhang, J. Wang, P. Liu, and J. Hou, "Fall detection by embedding an accelerometer in cellphone and using kfd algorithm," *International Journal of Computer Science and Network Security*, vol. 6, no. 10, pp. 277–284, 2006.

[2] A. Bourke, J. Obrien, and G. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait & Posture*, vol. 26, no. 2, pp. 194–199, 2007.

[3] J. Hwang, J. Kang, Y. Jang, and H. Kim, "Development of novel algorithm and real-time monitoring ambulatory system using bluetooth module for fall detection in the elderly," in *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, vol. 1. IEEE, 2004, pp. 2204–2207.

[4] N. Noury, P. Barralon, G. Virone, P. Boissy, M. Hamel, and P. Rumeau, "A smart sensor based on rules and its evaluation in daily routines," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, vol. 4. IEEE, 2003, pp. 3286–3289.

[5] M. Luštrek and B. Kaluža, "Fall detection and activity recognition with machine learning," *Informatica*, vol. 33, no. 2, pp. 197–204, 2009.

[6] D. N. Olivieri, I. Gómez Conde, and X. A. Vila Sobrino, "Eigenspace-based fall detection and activity recognition from motion templates and machine learning," *Expert Syst. Appl.*, vol. 39, no. 5, pp. 5935–5945, Apr. 2012.

[7] A. Bourke and G. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Medical Engineering & Physics*, vol. 30, no. 1, pp. 84–90, 2008.

[8] M. V. Albert, K. Kording, M. Herrmann, and A. Jayaraman, "Fall classification by machine learning using mobile phones," *PloS one*, vol. 7, no. 5, p. e36556, 2012.

[9] O. Dekel and O. Shamir, "Multiclass-multilabel classification with more classes than examples," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 137–144.

[10] K. P. Murphy, *Machine learning: a probabilistic perspective*. The MIT Press, 2012.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] A. A. Farag and S. Y. Elhabian, "A tutorial on data reduction: Linear discriminant analysis (LDA)," University of Louisville, Tech. Rep., October 2008.