Intelligent Gait Analysis using Marker Based Motion Capturing System

Danny Möbius[a], Marika Kaden[b], Daniel Staps[b], and Thomas Villmann[b]

[a] Institut für Mechatronik, Chemnitz, Germany

[b] SICIM, Hochschule Mittweida, Mittweida, Germany

Marker-based systems can digitally record human movements in detail. Using the digital biomechanical human model Dynamicus, which was developed by the Institut für Mechatronik, it is possible to model joint angles and their velocities such accurately that it can be used to improve motion analysis in competitive sports or for ergonomic evaluation of motion sequences. In this paper, we use interpretable machine learning techniques to analyze the gait. Here, the focus is on the classification between foot touchdown and drop-off during normal walking. The motion data for training the model is labeled using force plates. We analyze how we could apply our machine learning models directly on new motion data recorded in a different scenario compared to the initial training, more precise on a treadmill. We use the properties of the interpretable model to detect drift and to transfer our model if necessary.

1. Introduction

In the *Institut für Mechatronik* (IfM) a digital biomechanical human model names *Alaska/Dynamicus* was developed [1,2]. It uses the data of a marker-based system, which record the human movements. By means of the digital model it is possible to obtain the joint angels and their velocities very precise. Marker-based systems have a broad application not only on competition sports, also in ergonomic evaluation and movie industry. Even marker-less systems get better and better in the recent years, especially under the use of machine learning techniques, it cannot achieve such precise results in predicting the joint angles, yet [3]. The model *Dynamicus* is successfully applied in several areas like detection/simulation of the movement of the subject and the environment when entering a passenger car or truck, automated evaluation of work processes with the EAWS method or the acquisition/simulation of a jump on the force plate.

In our current gait study, the goal is to precisely detect the time when a foot touchdown and drop-off during normal walking using only the data of the motion capturing system. In a first step we divide the normal walking into two phases: the swinging and the standing phase of each foot. Using artificial intelligence, more precise machine learning (ML) we could train a model to predict these phases for the left and right food very precise. The training of the model is taken place with labeled data, i.e., beside the different joint angels of the body provided by *Dynamicus* the time of the phases is given. These labels were obtained by using additionally force plates. This measuring system provides the exact timing for the phases. As we can see later, the model show good practical performance.

Yet, the force plates cannot easily apply in every scenario to train a scenario specific model, e. g. during walking on a treadmill. The idea is to train a ML model on the data, where the labels are given, i.e., in the scenario, where the force plates can be used, and then apply this model to other scenarios. But as we can imaging, standard walking on the floor and walking on the treadmill have different influences on the movement. The questions are: Can we apply our model to other scenario or is a transformation necessary? If the model should be transferred, how we could adapt our model? For easier communication we will name the data recorded on the floor with available labels *train data* and the data of the treadmill without knowledge of the ground truth label *test data*.

Detection of data-drift is still an ongoing research topic [5]. We want to use prototype-based machine learning models to tackle the problem. These models have several advantages: Due to their competitive strategy they are quite intuitive and interpretable [7]. During minimizing the corresponding cost function the margin is maximized, which results in a robust decision making \cite{margin}. Further the complexity of the model can be directly chosen by the applicant, i.e., the complexity of the model is not given, but can be adapted to boundary conditions. Another advantage is that we can use our model to detect drift.

2. Prototype Based Models

We do not want to overload the reader with formulas; thus, we describe the ideas only in rough illustrating manner. In the Generalized Learning Vector Quantization model (GLVQ, [11]) we have, beside the data with respective class assignments called labels, prototypes also equipped with labels. During learning the prototypes are adapted following geometrically an attraction and repulsing scheme, i.e., if the label of a training data sample and the corresponding nearest prototype agrees the prototype is attracted by the data point and it is pushed away otherwise. This very intuitive learning scheme is the result of applying stochastic gradient descent on the GLVQ-cost function. This cost function approximates the classification error to be optimized during training and, additionally, describes the (hypothesis) margin \cite{margin}. In the recall phase, a new data point gets the label of the nearest prototype, also known as Winner-Takes-All rule.

The dissimilarity measure used in the competition is in general the squared Euclidean distance, but any other measure can be applied like kernel distances or divergences [9,10] under mild mathematical conditions. A powerful extension is the Generalized Matrix LVQ (GMLVQ, [12]). Here, the distance is replaced by a parameterized version:

$$d_{\Omega}(\boldsymbol{x},\boldsymbol{w}) = \left(\Omega(\boldsymbol{x}-\boldsymbol{w})\right)^2$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is a given data sample and $\boldsymbol{w} \in \mathbb{R}^n$ represents a prototype and $\boldsymbol{\Omega} \in \mathbb{R}^{mxn}$ is called the mapping matrix, which maps the data from an $\boldsymbol{n}$ –dimensional to a $\boldsymbol{m}$ –dimensional vector space. This means that the above matrix $\boldsymbol{\Omega}$ has the goal to map the data in such a way that they become better separable. For $\boldsymbol{m}$ we obtain a mapping into the two-dimensional visualization space while maximizing the classification performance.

In the specific gait analysis task to be investigated here, pure classification accuracy tells nothing about the applicability of the model. Yet, *IfM* provides a tool, which shows the single phases of the left and right foot. An illustration of the output of the learned GLVQ model is depicted in Figure 4. By means of this tool it is possible to evaluate the classification result visually. The obtained results are promising and show that it is possible to obtain a model to detect these phases only using the joint angles provided by *Dynamicus*.

Another possibility to evaluate the performance is the use of the time-difference between ground truth and the prediction for detecting the starting point of each phase and the detection of ending a phase. The mean time-difference for starting is 0.2 seconds with a standardization of $\pm$0.23 and for ending 0.31 $\pm$ 0.42 seconds. Thus, our ML-model works in praxis.

3. Drift Detection

Another challenge in our scenario is that we only have labeled data for the normal working mode on the floor. We do not have any label information for the data obtained from the treadmill due to technical difficulties. One possibility to overcome the lack of information is to visualize the high dimensional data using a maybe non-linear visualization method like t-distributed Stochastic Neighbor Embedding (t-SNE) [6]. In Figure 1 we illustrate the result of t-SNE on a subset of the data. On a first glance the data of the treadmill seems to be similarly distributed like the normal labeled data. But if we take a closer look, we recognize a slightly different shape of the manifold. However, we cannot decide with certainty out of this visualization whether the treadmill data are drifted compared to normal mode.

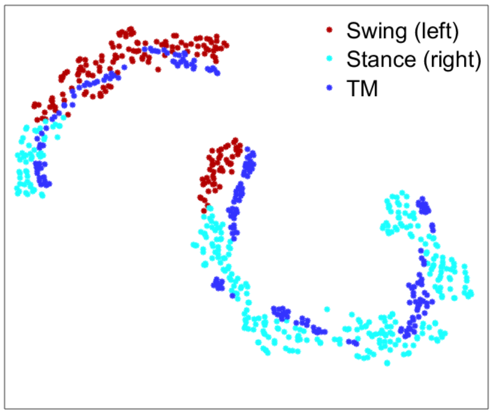

Figure 1: t-SNE visualization of the data (red - data with the label swing left, cyan- data with the label stance left and blue - data recorded on the treadmill)

Therefore, we use the GMLVQ with the interpretable recall phase to detect drift. In particular, we analyze the distribution of the distances of the data points to their nearest prototypes. The assumption for this is that if the new data are similar distributed in the vector space like the training data, the distance distributions should be similar, too. In contrast, if drift has taken place, the distribution of distance values should differ. The big advantage on looking on the distribution of the distances is that these distributions are only one-dimensional and thus easier to handle than comparing distributions in a high-dimensional data space. In Figure 2 the distance distributions are illustrated by respective histogram plots. We can detect a slightly different distribution of the distances referring to a drift. Especially for distances in the range of [10,40], i.e., data samples with larger distances to the winning prototype, the distribution is changed.

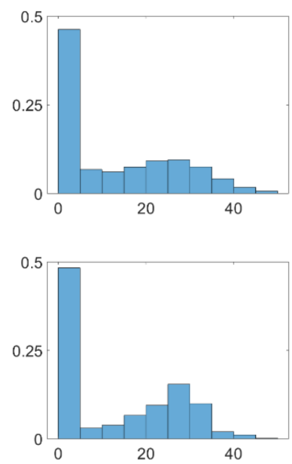

Figure 2: Histogram plots of the distances between data and closest prototype (top: data recorded by normal walking, bottom: data recorded on the treadmill)

In the 2D data visualization by means of the t-SNE the label information is not considered. Hence, if a drift in the data would be detected inspecting the visualization, the consequences of this drift for class discrimination is neither clear nor obvious. Here, the GMLVQ provides a possibility for visualization by taking the perspective of the later discrimination model. Setting the mapping dimension m in GMLVQ, we can visualize the data and optimize the model to be as best class discriminating as possible. In Figure 3 we depict the training and test data mapped using the trained Ω matrix of the GMLVQ classifier. It can be observed that the test data seems to be slightly drifted orthogonal to the decision hyperplane. The difference is not huge, but directly influences the decision. So, it might result in a higher misclassification rate for the *stance* class.

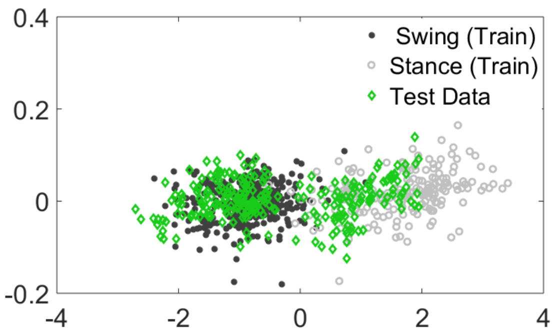

Figure 3: Visualization of the mapped data (training data - grey, test data - green)

In consequence of this observation, a direct application of the learned model on the data recorded during working on the floor to data captured in another scenario (treadmill) is not recommended. Thereby, it should be pointed out again that we do not have any labels for the second data set, such a direct evaluation of the classification results is not possible

4. Conclusion and Future Work

We apply matrix variant of the Generalized Learning Vector Quantization for a classification task in the field of motion detection using a digital bio-mechanical human model. Moreover, we figured out possibilities for drift detection in the data using the interpretability of this GMLVQ approach. To apply a model on novel data, we must check in advance whether the data have drifted, especially if the data are measured using new or slightly different positioned sensors, or the environment of the measurement process has been changed. The application of GMLVQ enables to detect those drift types, which have a high impact to the class separation. This detection is based on the utilization of the learned GMLVQ mapping matrix for class discriminating data visualization.

So far, the analyze of the GMLVQ-based visualization is done manually to detect class drift. A future work is to automatize these detections. Moreover, the mapping matrix could be also learned in a way that the model decision is not influenced by the drift, which is related to robust adversarial learning [13, 14]. Yet, for this task data labels are required also for the new (drifted) data. Hence, we must modify these ideas accordingly, to apply it successfully for this task.

Acknowledgements

This work is co-financed by tax funds on the basis of the budget adopted by the Saxon Landtag's budget.

Europäische Union

Europa fördert Sachsen.
ESF
Europäischer Sozialfonds

Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des vom Sächsischen Landtag beschlossenen Haushaltes.

References

[1] www.ifm-chemnitz.de

[2] H. Hermsdorf, N. Hofmann, and A. Keil: Chapter 16 - Alaska/dynamicus – human movements in interplay with the environment. Academic Press, DHM and Posturography, p. 187-198, 2019

[3] E. Ceseracciu, Z. Sawacha, C. Cobelli. Comparison of marker less and marker-based motion capture technologies through simultaneous data collection during gait: proof of concept. PLoS One.2014

[4] N. Hofmann, J. Alhakeem, H. Hermsdorf, D. Möbius , S. Öhmichen, H. Schulz, M. Kaden, and T. Villmann: Analysemethoden zur Abschätzung von Belastungsintensitäten bei Kniebeugen unter Verwendung interpretierbare Modelle der Künstlichen Intelligenz: Machine Learning Report 01, MLR-01-2020, 2020

[5] A. S. Iwashita and J. P. Papa: An Overview on Concept Drift Learning. IEEE Access, vol. 7, pp. 1532-1547, 2019 L. J. P. van der Maaten, G. E. Hinton: Visualizing Data Using t-SNE. Journal of Machine Learning Research. 9: 2579- 2605, 2008

[6] C.Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nature Machine Intelligence 1 (5), 206-21, 2019

[7] K. Crammer, R. Gilad-Bachrach, A. Navot and N. Tishby. Margin Analysis of the LVQ Algorithm. Advances in Neural Information Processing Systems, 2003

[8] T. Villmann, S. Haase, and M. Kaden. Kernelized vector quantization in gradient-descent learning. Neurocomputing 147, 83-95, 2015.

[9] T. Villmann and S. Haase. Divergence based vector quantization. Neural Computation, 2011

[10] A. Sato and K. Yamada. Generalized learning vector quantization. Advances in neural information processing systems, 1996.

[11] P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. Neural Computation, vol. 21, pp. 3532-3561, 2009

[12] S. Saralajew, and T. Villmann. Transfer learning in classification based on manifold models and its relation to tangent metric learning. International Joint Conference on Neural Networks (IJCNN). IEEE, 2017

[13] B. Paassen, A. Schulz and B. Hammer. Supervised Linear Transfer Learning for GMLVQ. Proceeding on Workshop New Challenges in Neural Computtion, MLR-03-2016, 14-19, 2016

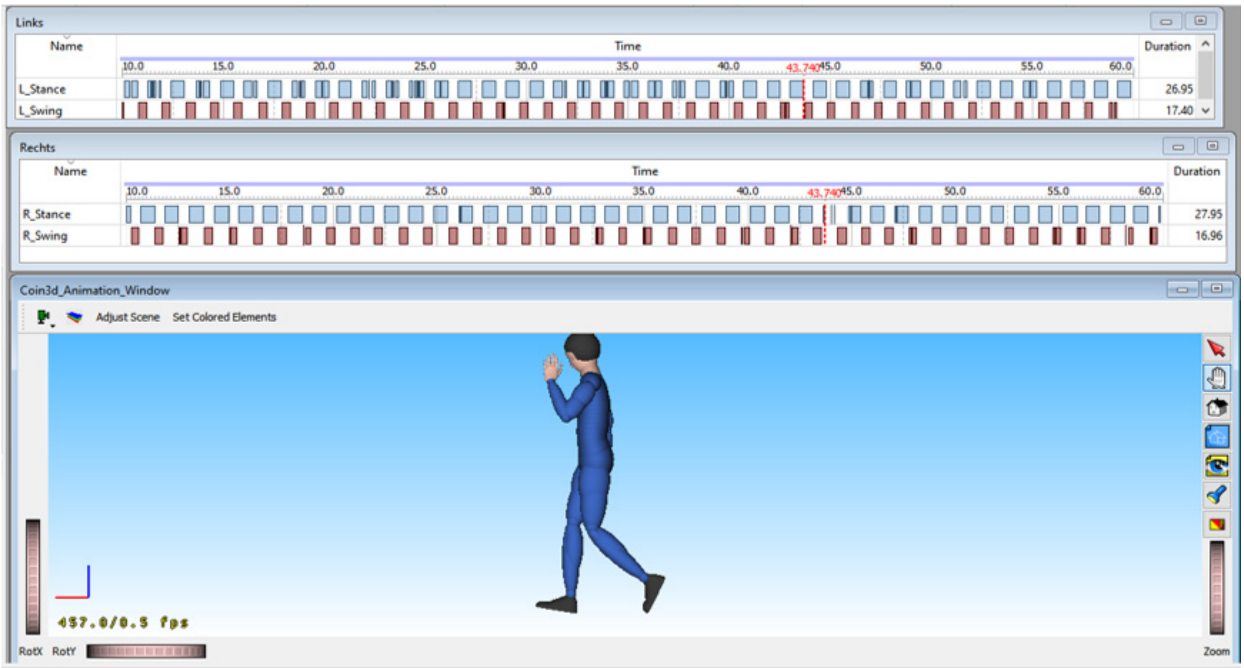

Figure 4: Classification result integrated in a tool provided by IfM