Time Series Analysis for Long Term Prediction of Human Movement Trajectories

Conference Paper · November 2008

DOI: 10.1007/978-3-642-03040-6_69 · Source: DBLP

4 authors:

Sven Hellbach
Westsächsische Hochschule Zwickau

Julian Eggert
Honda Research Institute Europe GmbH

Edgar Körner
Honda Research Institute Europe GmbH

Horst-Michael Gross
Technische Universität Ilmenau

in: Proc. International Conference on Neural Information Processing (ICONIP 2008), Auckland (NZ), LNCS 5507, pp. 567-574, Springer 2009

Time Series Analysis for Long Term Prediction of Human Movement Trajectories

Sven Hellbach[1], Julian P. Eggert[2], Edgar Körner[2], and Horst-Michael Gross[1]

[1] Ilmenau University of Technology, Neuroinformatics and Cognitive Robotics Labs, POB 10 05 65, 98684 Ilmenau, Germany
sven.hellbach@tu-ilmenau.de
[2] Honda Research Institute Europe GmbH, Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany
julian.eggert@honda-ri.de

Abstract. This paper's intention is to adapt prediction algorithms well known in the field of time series analysis to problems being faced in the field of mobile robotics and Human-Robot-Interaction (HRI). The idea is to predict movement data by understanding it as time series. The prediction takes place with a black box model, which means that no further knowledge on motion dynamics is used then the past of the trajectory itself. This means, the suggested approaches are able to adapt to different situations. Several state-of-the-art algorithms such as Local Modeling, Cluster Weighted Modeling, Echo State Networks and Autoregressive Models are evaluated and compared. For experiments, real movement trajectories of a human are used. Since mobile robots highly depend on real-time application, computing time is also considered. Experiments show that Echo State Networks and Local Model show impressive results for long term motion prediction.

1 Introduction

For autonomous robots, like SCITOS [1], it is important to predict the motion of people and other robots in their environment, for example to avoid collisions. Hence, further actions can be planned more efficiently. Most approaches in this field focus on optimal navigation strategies [2, 3]. This paper suggests to spend more effort into prediction of the motion of the dynamic objects (i. e. in most cases the motion of humans in the scene) instead. Often, only linear approximations or linear combinations are used to solve this problem.

Plenty of algorithms exist for time series analysis and prediction. Their fields of application reach from prediction of economic data to climate and biologic data, such as neural activities [4]. The new approach is the interpretation of movement data as time series to perform a long-term prediction. For this prediction, an assortment of time series analysis algorithms has been implemented and comparatively tested.

For this, it is necessary to know the motion trajectories of the surrounding dynamic objects. For simplification, a tracker is assumed, which is able to provide

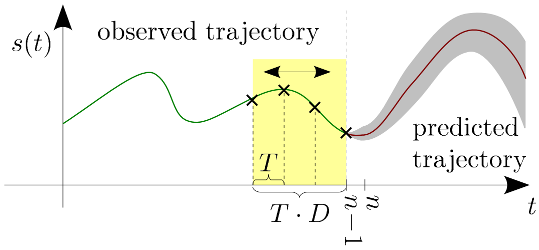


Fig. 1. The observed trajectory (green) is to be predicted (red) for up to 500 time steps (about 8.3 sec. at 60 Hz). This is achieved only by exploiting the past trajectory's characteristics using a window (yellow) of D points equally spaced with interval T.

such trajectories in real-time. A possible tracker to be used is presented in [5]. In this case, the given trajectory of the motion can be interpreted as a time series $\mathcal{T}$ with values $\mathbf{s}_i$ for time steps $i = 0, 1, \ldots, n-1$: $\mathcal{T} = (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1})$.

The next section introduces our time series analysis approach to mobile robotics and techniques chosen to be tested. In section 3 the comparing experiments with their conditions and results are presented, while the paper concludes in section 4.

2 Time Series Prediction

The algorithms presented in this paper are intended to be used for motion prediction to enable a more anticipative mobile robot navigation in dynamic environments. Basically, for all presented algorithms the prediction for each future point on the trajectory is done iteratively for up to 500 time steps (this corresponds to about 8.3 sec. of motion with a sampling frequency of 60 Hz) (Fig. 1).

The prediction in general takes place with the so-called black box model which means that no further background information or knowledge about the motion dynamics is used than the past trajectory itself. The aspired prediction is to follow the trajectory's characteristics, only, which can be found in their past. Furthermore, no explicit trajectory models are given, to be able to freely adapt to yet unknown situations.

2.1 Echo State Networks

For prediction of time series, Echo State Networks are often used [6]. They have some specific features which differ from "standard" neural networks: The hidden layer consists of neurons which are randomly connected. When the connectivity is low, this layer provides independent output trajectories. For this reason, the hidden layer is also called "reservoir". Furthermore, there are neurons which are connected to loops in the reservoir, so that past states "echo" in the reservoir. That is the reason, why only the actual time series value is needed as input.

Another characteristic of Echo State Networks is that only the output weights are adapted and learned. All other weights (input, reservoir, feedback) are chosen randomly and stay statically. For training, the net is randomly initialized, and the training time series is used as net input step by step.

This paper suggests to use multiple instances of the network, as a kind of simple stochastic search in the parameter space. The fixed weights are initialized differrently in a random manner. All instances are trained using the same input data. During the training process, the output of each network is compared with the corresponding values of the training trajectory. The network showing the best prediction results for the yet unknown training data is then selected for further application.

2.2 Autoregressive Models

The next type of time series analysis algorithms introduced here are Autoregressive Models (AR). These models assume a linear relation in the time series which means that any time series value can be determined by using a linear combination of p previous values. The coefficients of the linear combination – the AR coefficients – have to be calculated to predict future values. Several Algorithms exist to determine these coefficients, e. g. Wiener Filter [7], Durbin-Levinson [8], and Yule-Walker [8].

2.3 Embedding Space

For applying the approaches in sections 2.4 and 2.5, an embedding in a higher dimensional space is necessary. This embedding can be regarded as a kind of the well known sliding window approach. An observation window with size $T \cdot D$ is put on the trajectory (Fig. 1). Each T-th time step from this window is used to generate this *regular embedding*. So the time series is transformed into a D-dimensional space - the *embedding space*. To each embedding $\mathbf{e}_t = \left(\mathbf{s}_t, \mathbf{s}_{t-T}, \mathbf{s}_{t-2T}, \ldots, \mathbf{s}_{t-(D-1)T}\right)^T$ belongs an output $\mathbf{o}_t$, which stands for the successor $\mathbf{s}_{t+1}$ of the selected window.

The two introduced parameters T and D don't need to be defined by hand. Time series analysis offers techniques to automatically determine these parameters [4]. In our work, we used genetic algorithms to find the best suited embedding dimension.

2.4 Local Modeling

Local Modeling, which is described in [9], is based on the aforementioned regular embedding. The principle idea is a simple nearest neighbor search in the embedding space of the last point in the time series $\mathbf{e}_{n-1}$ for which the prediction needs to be calculated.

In the general case, a polynomial is estimated for prediction describing the relationship between embedding $\mathbf{e}_i$ and output $\mathbf{o}_i$. The nearest neighbors are used to decide the polynomial's coefficients $\mathbf{v}$ applying linear regression. In practice, the polynomial degree g is usually low. Often it is enough to use $g = 0$ (*Local Averaging Model*) or $g = 1$ (*Local Linear Model*).

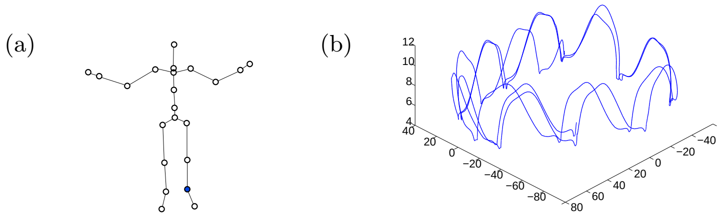


Fig. 2. Example of movement data from the University of Glasgow. Data is available for the body limbs shown in (a) and an exemplary trajectory of the movement of the left ankle while walking in circles (b).

After determining the coefficients, the prediction is calculated using the same polynomial interpolation. To get good prediction results, it is crucial to choose proper parameters, such as the embedding parameters T and D and the number of the nearest neighbors N. Especially with higher polynomial degrees, the algorithm is extremely sensitive to the choice of these parameters. Therefore, an evolutionary algorithm was implemented which often leads to good results as recommended in [9].

2.5 Cluster Weighted Modeling

The Cluster Weighted Modeling, which is described in [9], is also operating in the embedding space. The viewpoint lies not on single embedding points like in the Local Modeling. Now the embedding space is clustered and covered with Gaussians.

An Expectation-Maximization-algorithm (EM-algorithm) can be used to optimize most of the algorithm's parameters. The whole algorithm can be found in detail in [9]. Only the number of clusters and the cluster function remain to be chosen manually. All other parameters are initialized randomly and adapted using the optimization. As cluster function, similar functions like the Local Modeling polynomials, can be used. Since, calculation time strongly depends on the number of clusters, the values of these parameters should not be too high for an online application.

3 Motion Prediction

The algorithms presented in this paper are intended to be used for motion prediction to enable a mobile robot navigating in dynamic environments. To be comparable and reproducible, movement data taken from the University of Glasgow [10] is used. This benchmark data is available as 3D coordinate representation for each limb of a human performing a certain action, e. g. walking (see Fig. 2). Using this data is even more challenging, because several basic motions are combined (i. e. intrinsic movement, e. g of the foot combined with the walking direction). The data set consists of 25 trajectories containing 1,500 up to 2,500 sampled points in Cartesian space.

3.1 Test Conditions

The movement data has a resolution of 60 time steps per second, so that an average prediction horizon of about 500 steps corresponds a prediction of 8.3 seconds into the future at 60 Hz. Present movement prediction techniques are designed to predict an objects position for the next time frame or at least to gap a loss of the object during a only a few frames.

Quality Measures For comparing the prediction results, some kind of quality measures for comparison are necessary. The used quality measures are based on the normalized mean square error $NMSE$. Hence, the standard mean square error is normalized using the variance σ^2 of the time series.

$$NMSE = \frac{1}{N \cdot \sigma^2} \sum_{i=1}^{N} (\mathbf{s}_i^{pred} - \mathbf{s}_i^{orig})^2 = \frac{MSE}{\sigma^2} \tag{1}$$

Since the trajectories are three-dimensional and dimensions with greater difference suppose to be more important, the highest variance of all dimensions is used as normalization.

Two different kinds of the defined measure are used. The first one, the short term error STE, is responsible for evaluating a short period of the prediction. It uses the first $N = 75$ prediction steps (which means 1.25 sec) with a weighting of $\frac{1}{f}$ for the f-th prediction step. On the other hand, the performance is evaluated using the long term error LTE, which uses all prediction steps with a weighting of $1/\sqrt{f}$, since some of the algorithm show the tendency to drift away

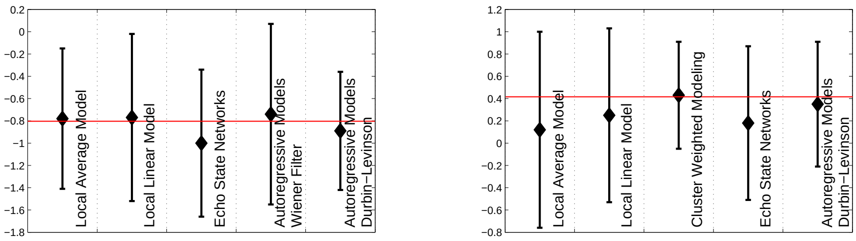
Reference Algorithms Additional simple reference algorithms were used that should be outperformed clearly to get a useful prediction. The first algorithm is a simple repetition of the last time series value and is called constant algorithm in the following. Also a linear algorithm is used as reference. This algorithm simply does a linear approximation in the last two points in the time series. The result of the better one is used as reference.

3.2 Experimental Results and Comparison

The following tests show the advantages and disadvantages of the different algorithms presented here. For the application, a number of parameters had to be decided to apply the algorithms. The values presented in the following are chosen after extensive test, which are not discussed here.

Especially for Echo State Networks the choice of the parameters is important. It has shown that the scaling of the weights is essential. The feedback weights $\mathbf{w}_{back}$ must be scaled very low (ca. 10^{-20}) to guarantee stable networks dynamics.

As input for the Wiener Filter the embedding presented in section 2.3 is used instead of only using the last p values. Experiments show that using the



(a) (b)

Fig. 3. The graphs shows the STE (a) and LTE (b) plotted for each of the investigated algorithms. The ordinate is scale logarithmically. Hence, lower values mean a better prediction. The error bars represent the standard deviation from the mean. For the STE, all results lie relatively close together while the reference algorithm (red line) can only beaten clearly by the Echo State Networks. Longer predictions show more differences in the results of the algorithms. Also the mean errors are higher than STE, as being expected in longer predictions. The reference is beaten more clearly in general. Local Average Models (LAM) and Echo State Networks show the best results.

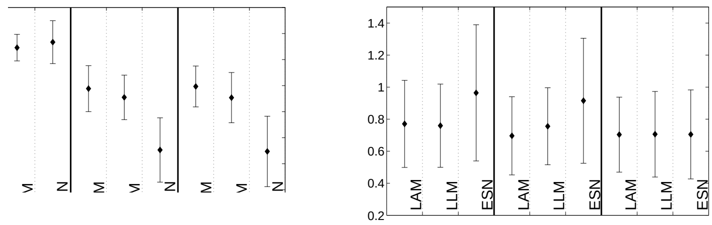
embedding leads to better results. For the other Autoregressive Models values around $p = 100$ for AR depth often lead to the best results.

For generating the embedding, the number of histogram bins for calculating the mutual information has to be specified. Proper values are between 15 and 30. In most cases, the smaller value is used to keep the calculation time low. To fasten the whole embedding procedure, not every embedding point is used for the classification in true and false neighbors, but a random selection of around 5-10% of the time series embedding points.

In the prediction of movement data, the Echo State Networks lead to the best results for the STE as it is shown in Fig. 3(a), while for long term prediction Local Models have slightly better results (Fig. 3(b)). The Autoregressive Models perform barely better than the reference. Here the Durbin-Levinson algorithm achieves the best prediction quality. Cluster Weighted Models show the worst performance, and their mean errors stay even behind the simple reference algorithms. The best algorithms still beat the simple references clearly and are able to predict movements for several seconds (about 100 prediction steps) very well.

The choice of the number of neurons in the Echo State Network reservoir, for example, has only a minor effect. In tests the difference in the prediction results of movement data between 25 and 250 neurons were insignificant. It can be presumed that the structure of the movement data does not allow a higher accuracy in the prediction unlike other chaotic time series [6].

The evaluation discussed in the previous paragraphs used a time horizon of 1000 time steps for training. Towards, online application such a long training phase would mean to observe the person for several seconds. Since, this is not possible in most cases, the tests depicted in Fig. 4 are tested with less data. Only 300 time steps of the trajectory are used now. Those 300 points in time are



(a) (b)

Fig. 4. The graphs show the STE (a) and LTE (b) plotted for the most promising algorithms of the previous tests (Local Average Model (LAM), Local Linear Model (LLM), and Echo State Networks ESN) in a similar fashion as in Fig. 3. Each plot is separated into 3 sections. From left to right, these sections show the results for the test with the subsampled trajectory, the interpolated trajectory, and the comparison with the normal trajectory.

subsampled for the three left most results in Fig. 4(a) and Fig. 4(b), as it would be the case when using a slow tracker. As it can be expected, the prediction quality significantly decreases (compared to the three right most results in 4(a) and Fig. 4(b)). A logical step at this point is to use interpolation to fill the missing gaps. A spline interpolation is used for the test in Fig. 4 to gain 300 time steps of training data again. The results can be compared to the ones using the original trajectory (see the three midway results in 4(a) and Fig. 4(b)).

Calculation Time For any online application, the calculation time plays a big role, since the movement is supposed to be predicted before it continues. Since, only MatLab implementations were tested on time series with lengths around 1,000 till 2,500 time steps, only a first hint can be given here.

Autoregressive Models and Echo State Networks with lower number of neurons show a calculation time of about 3-10 ms per prediction step. This is absolutely complying with online requirements.

Local Models and Cluster Weighted Models need longer calculation times between 50 and 250 ms. In the first case (Local Models), most calculation time is spend on the search for the nearest neigbors in the high number of training data. The Cluster Weighted Models are slow because of a long optimization time (the EM-algorithm).

4 Conclusions and Future Works

The intention of this paper was to connect the well-known field of time series prediction and movement data handling from robotics in a consistent way. Different behaviors from the tested time series analysis algorithms were observed. Generally, it can be said that movement data behaves different than periodical and chaotic time series.

The tested algorithms show very good results in predicting several seconds of the movement data. Echo State Networks and Local Models pointed out to ba a suitable algorithm for movement prediction

Autoregressive Models and again Echo State Networks are able to predict fast enough for an online application without any further adaptation. From the current point of view, Echo State Networks are the "winning" approach which is able to solve the problem best. Hence, further analysis should have the focus on this approach and on additional improvements.

The other algorithms can be upgraded as well. Local Models can be a good alternative to Echo State Networks if they could be accelerated without loss of quality. Besides this, enhanced versions of the Autoregressive Models such as ARMA or ARIMA Models could be tested. Furthermore, the usage of an irregular embedding is imaginable.

As a next step, an adequate navigation strategy exploiting the prediction results needs to be investigated. One drawback for predicting motion data is the fact that human beings may perform unexpected motion. Since the discussed algorithms rely on the known characteristics, it is possible to use them for detection of such unexpected behavior.

Acknowledgment

Thanks to our students Sören Strauß and Sandra Helsper for doing the evaluation work and contributing good ideas.

References

1. Gross, H.M., Böhme, H.J., Schröter, C., Müller, S., König, A., Martin, C., Merten, M., Bley, A.: Shopbot: Progress in developing an interactive mobile shopping assistant for everyday use. In: to appear: Proc. IEEE Internat. Conf. on Systems, Man and Cybernetics (IEEE-SMC). (2008)
2. Pett, S., Fraichard, T.: Safe navigation of a car-like robot within a dynamic environment. In: Proc. of European Conf. on Mobile Robots. (2005) 116 – 121
3. Owen, E., Montano, L.: Motion planning in dynamic environments using the velocity space. In: Proc. of RJS/IEEE IROS. (2005) 997 – 1002
4. Abarbanel, H., Parlitz, U.: Nonlinear Analysis of Time Series Data. In: Handbook of Time Series Analysis. WILEY-VCH (2006) 1 – 37
5. Scheidig, A., Müller, S., Martin, C., Gross, H.M.: Generating person's movement trajectories on a mobile robot. In: Proc. of International Symposium on Robots and Human Interactive Communications (RO-MAN). (2006) 747 – 752
6. Jaeger, H., Haas, H.: Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. Science **April** (2004) 78 – 80
7. Wiener:, N.: Extrapolation, Interpolation, and Smoothing of Stationary Time Series. Wiley (1949)
8. Shumway, R.H., Stoffer, D.S.: Time Series Analysis and Its Applications. Springer Texts in Statistics (2000)
9. Engster, D., Parlitz, U.: Local and Cluster Weighted Modeling for Time Serie Prediction. In: Handbook of Time Series Analysis. WILEY-VCH (2006) 38 – 65
10. http://paco.psy.gla.ac.uk/data_ptd.php