Efficient SVR model update approaches for respiratory motion prediction for the 11. CURAC 2012

R. Dürichen, T. Wissel, A. Schweikard

University of Luebeck, Institute for Robotics and Cognitive Systems, Luebeck, Germany

Contact: duerichen@rob.uni-luebeck.de

Abstract:

In order to successfully ablate moving tumours in robotic radiosurgery, respiratory motion prediction is needed to compensate time delays. In this context, recent studies revealed a high potential of support vector regression (SVR). However, high computational cost is one major drawback, particularly caused by batch mode training. We evaluate two approaches to reduce the update rate as well as computation time, while keeping a low prediction error. The update rules are either based on information about the respiratory phase or based on the current prediction error. An evaluation on patient data sets revealed that the second approach on average decreases computation time by 88.53% compared to a batch mode implementation. The prediction error increased by 0.3%, hence indicating enhanced efficiency.

Keywords: radiosurgery, respiratory motion prediction, support vector regression, model update

1 Problem

In modern robot-based radiation surgery, precisely radiating moving tumours has become more and more feasible, while sparing surrounding critical structures. One state-of-the-art system is the CyberKnife® (Accuray Systems, Sunnyvale, CA). Apart from radiating the tumour from multiple locations, the system can compensate internal tumour motion that is caused by respiration. For compensation optical markers, attached to the patient's chest, are continuously tracked. Their position is correlated with internal landmarks by stereoscopic X-ray images [1]. However, due to kinematic limitations, data acquisition and processing, time latencies have to be taken into account. In case of CyberKnife® Synchrony this latency is 115ms. Thus, to reduce this systematic error, the position of the optical markers has to be predicted.

Several studies have shown that support vector regression (SVR) can precisely predict respiratory motion and is capable of competing with state-of-the-art motion prediction algorithms [2–5]. Implemented in usual batch mode (BM) the SVR model is recomputed at every incoming sample point. This leads to high computational cost and is therefore unpractical for real-time applications. To resolve this, Ernst et al. [2] suggested a global update factor τglobal, that reduces the update rate, but also increases the prediction error.

In this study, we investigate two SVR update methods to decrease the computation time while maintaining a low prediction error. First, we analyzed the occurrence of resulting Support Vectors (SV) over time to obtain an update factor depending on the respiratory phase. Second, the current prediction error at time t is used as an update criterion. If the prediction error exceeds a certain threshold, the SVR is recomputed.

2 Methods

For simplicity, all further explanations are reduced to a one dimensional signal. The algorithms can easily be expanded to three dimensions. It is assumed that the respiratory signal is equidistantly sampled with a sampling rate fs and that the resulting sequence has length N. Let yt be the signal amplitude at time point t and yt+δ the amplitude of the predicted point. Here, δ represents the prediction horizon. The predicted value is denoted by $\hat{y}_{t+\delta}$ and depends on $u_t = y_1, \dots, y_t$.

2.1 Support Vector Regression (SVR)

Given a training set

$$T = \{(u_n, y_{n+\delta}), \dots, (u_{n-L+1}, y_{n-L+1+\delta})\} \text{ with } u_i = \{y_i, \dots, y_{i-M+1}\} \quad (1)$$

where again $u_i \in \mathbb{R}^M$ represents the training data and $y_{i+\delta} \in \mathbb{R}$ the training labels, the SVR fits a functional relationship $y = f(u)$ to this data. Here, L is the amount of training samples and M the number of past observations used to predict a new label. The ε-SVR does not penalize a fitting error up to ε per sample and optimizes $f(u)$ to be as smooth as possible. In the linear case with slope vector w and offset b, $f(u)$ can then be used to predict a point $\hat{y}_{n+\delta} = f(u_n) = \langle w, u_n \rangle + b$, where $\langle .,. \rangle$ denotes the dot product. As smoothness is equivalent to minimizing the l2-norm of w, a convex optimization problem, that is regularized by violations of the mentioned ε constraint, can be formulated and solved to finally derive $f(u)$. A full account of the underlying principles and mathematics is given in [6]. Training samples exceeding an absolute error of ε after performing the optimization are called Support Vectors (SV). Essentially, only these samples will contribute to the final definition of $f(u)$ and are hence most important for the training or updating process.

The SV machine can be extended to nonlinear cases by using kernel functions used to implicitly transform the linear case above to higher dimensions. The SVR was implemented with a Gaussian radial basis kernel (width parameter γ = 2) using LIBSVM Toolbox [7]. Further, we set the regularization constant C for prediction errors on the training data to 30, while L = 1000 and M = 5. The tube size ε was set to the standard deviation of the first á trous wavelet scale [2].

2.2 Updating of SVR model

Phase factor update rule (PFUR)

In contrast to general BM training, Ernst et al. [2] introduced a global update factor τ_{global}, updating the model if $mod(t, 1/\tau_{global}) = 0$ is true, which happens at frequent intervals. Thus, an update factor τ_{global} = 1 is equivalent to the BM implementation. We will refer to this method as constant factor update rule (CFUR). Here, this idea is extended to a phase specific update factor. Therefore, we assume that the occurrence of the SVs is not homogenously distributed over time and that the sets of SVs used by the SVR model at time t and $t-1$ are similar. Note again, that only SVs will contribute information to the SVR model after training. Consequently, the effective update rate of SVR adapts to the phase of the respiration period, e.g. maximum in- and expiration. We define two phases: maximum in- and expiration with τ_{max} = 1 and in- as well as exhalation phase with $\tau_{slope} < 1$. The length of the first phase is determined by a proportion parameter θ given in percent of the mean period duration per maximum. These phases are centered on the maxima.

Error update rule (EUR)

The second update algorithm depends on the current prediction error. If at time t a value y_t is measured, the algorithm computes $\hat{y}_t$based on the current SVR model and input $u_{t-\delta}$. If $|\hat{y}_t - y_t|$ exceeds a threshold e_{th}, the SVR model is updated. This aims at updating the model once it becomes too inaccurate. In order to estimate a general threshold independent of the signal, e_{th} is set to a multiple f of the chosen tube size ε.

3 Results

To investigate the PFUR and the EUR, we carried out three experiments on real patient data. First, we experimentally verified the assumptions being the basis of PFUR. Second, by varying τ_{slope}, θ and f, the potential of these two update algorithms is evaluated and compared to CFUR for the same signal**.** Third, EUR is further explored on 33 patient files.

According to the latencies of the CyberKnife system, the prediction horizon δ was set to 3 samples (≈115ms). All signals have been globally scaled to a range of [0, 1] and reduced to the first principle component for simplicity. Investigations were done on an i7-2600 CPU@3.40 GHz with 16GB RAM and all algorithms were implemented in Matlab.

The performance was evaluated using computation time t_{calc} and number of SVR model updates n_{update}. Whereas t_{calc} depends on the specific hardware of the computer and software implementation of the algorithm, n_{update} only depends on the algorithm and is approximately governed by a linear relationship to t_{calc}. Prediction accuracy was measured in terms of relative root mean square error RMS_{rel} [2]:

$$RMS_{rel} = \frac{\sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2/N}}{\sqrt{\sum_{i=1}^{N}(y_i - y_{i-\delta})^2/N}} \quad (2)$$

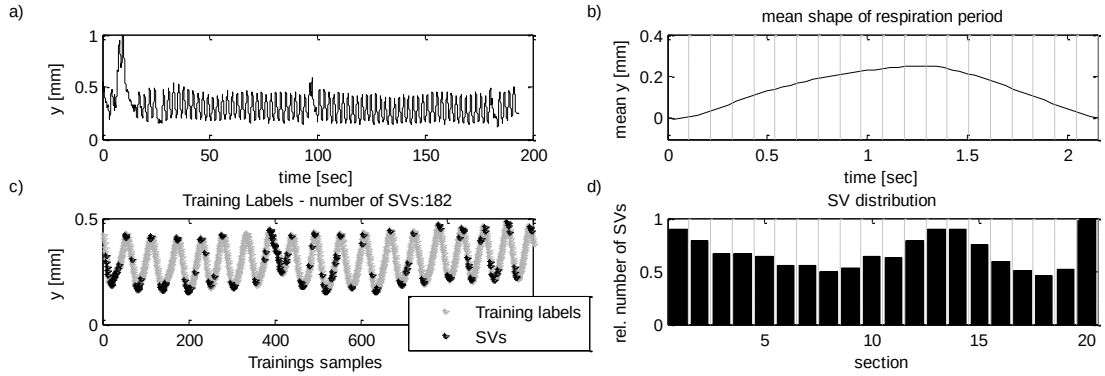

Figure 1: Analysis of the SV location; a) part of the analysed real signal; b) mean shape of the respiration period; c) number and location of SVs within a training batch for one time step *t*; d) number of SVs sorted into 20 bins relative to the maximum number of SVs among all bins

3.1 Support Vector distribution over time

In the first experiment, 20000 data points (≈770s) of a respiratory signal were processed by ε-SVR using a batch mode implementation (part of the signal is shown in fig. 1.a). At each time point, the number and location of the SVs were extracted from the SVR model. Fig. 1.c shows training labels and SVs for a certain time point *t*. The SVs are mainly distributed around the maxima in- and expiration. Over time the SVR model used an average amount of 446 from 1000 possible training samples as SVs. Per model update 3.3 elements from the SV set changed on anverage. Finally, a mean respiratory period was calculated from the signal (fig. 1.b) and divided into 20 bins. Depending on their location within the whole signal, each SV was assigned to one of the bins. Fig. 1.d shows a histogram of the accumulated number of SVs per bin normalized by the maximum number of training samples among all bins.

3.2 Comparing performance and efficiency of PFUR and EUR

We further analyzed the signal used in experiment one with PFUR and EUR. For PFUR, τ_{slope} was varied between 0 and 1 in steps of 0.1 and θ was incremented in steps of 5% between 5-20%. The impact of the multiplication factor f for EUR was tested by grid search using f = {0.05, 0.1, 0.25, 0.5, 0.667, 0.8, 1, 1.25, 1.5, 2, 4, 10, 20}. The results were compared with CFUR using a global update factor τ_{global} varied *between* 0 and 1 in steps of 0.1 and assessed using RMS_{rel}, t_{calc} and n_{update} (Fig.2.a-f).

3.3 Analysis of multiple patients with error update rule (EUR)

Based on the promising results of EUR, 33 patient files were evaluated only with this method. These signals were ran domly selected from a pool of 304 signals (31 patients) using 20000 data points for each analysis. As in section 3.2, the multiplication factor f was varied to find the optimal f_{opt}. To compare the results for different patients i, the average RMS_{rel}, t_{calc} and n_{update} were calculated relative to CFUR τ_{global} =1:

$$\overline{RMS_{rel}} = mean\left(\frac{RMS^i_{rel,EUR}}{RMS^i_{rel,CFUR}}\right); \overline{t_{calc}} = mean\left(\frac{t^i_{calc,EUR}}{t^i_{calc,CFUR}}\right); \overline{n_{update}} = mean\left(\frac{n^i_{update,EUR}}{n^i_{update,CFUR}}\right) \quad (3)$$

Figures 3.a - c show the results averaged across 33 patients. In addition the results of CFUR with τ_{global} = 0.1, 0.5 and 1

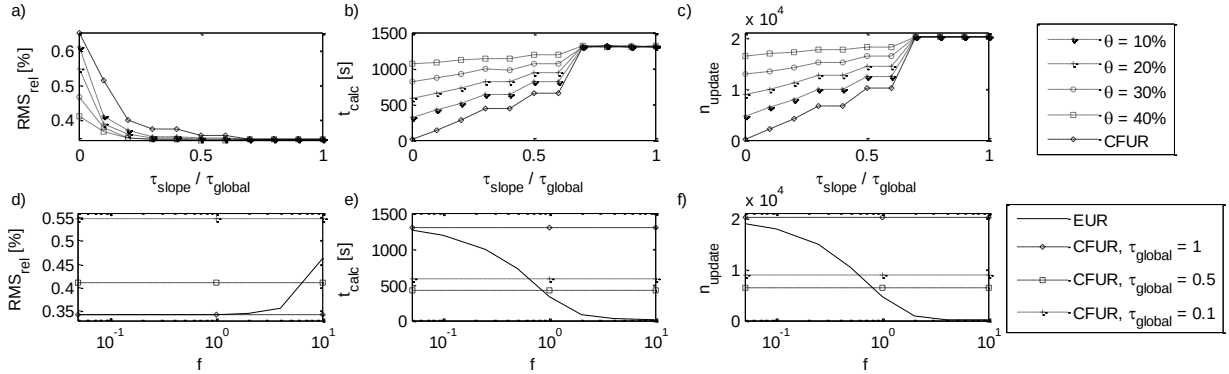

Figure 2: Evaluation of the relative RMS error RMS_{rel}, computation time t_{calc} and number of model updates n_{update} for one patient; (a-c) PFUR for different update factors τ_{slope} and size parameters θ; (d-f) EUR for different multiplication factor f compared to a CFUR with different τ_{global}

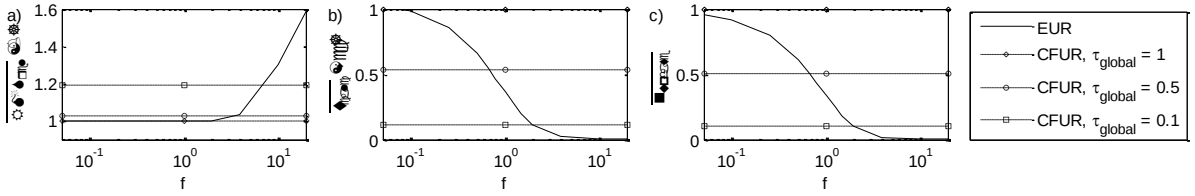

Figure 3: Evaluation of 33 patients (≈770s per patient) with EUR, a-c) shows the avg. relative RMS error, avg. computation time and avg. number of model updates with respect to the results for a batch mode implementation for different multiplication factor *f*

are illustrated. On average, EUR (f = 2) reduces t_{calc} by 88.53 %, and increases RMS_{rel} by 0.3%, compared to batch mode training. The mean computation time to predict approximately 770s of respiratory signal was 534.4s.

4 Discussion

In this work, we presented two updating methods to decrease computation time of an ε-SVR for respiratory motion prediction – one based on the current respiratory phase (PFUR) and one on the prediction error (EUR). In an initial analysis, we showed that the location of the SVs strongly depends on the phase in the respiratory period. We found that training samples at the maximum in- or expiration are likely to comprise more SVs than other periods. This result supports the initial assumption of PFUR, i.e. that the update rate can be reduced during in- and exhalation. The concept of this approach was studied for various τ_{slope} and θ on one data set. For τ_{slope} = 0.3 and θ = 15%, t_{calc} was reduced by 317. 4s compared to BM ($n_{update,PFUR}$ = 15159 instead of 20000 model updates), while the relative RMS error only increased by 0.34%. However, one drawback of this method is that the respiration phases, especially maximum in- and expiration, have to be known in advance or have to be identified online. This restricts the use of the PFUR for practical applications and makes it dependent on the accuracy with which the location of such extrema can be estimated. In contrast, EUR is independent of prior information and reduced t_{calc} by 1278.7s compared to an equivalent BM implementation, while yielding the same RMS_{rel} increase as for the PFUR ($n_{update,EUR}$ = 111 for f = 4). The result was confirmed in an evaluation of 33 patients. The average computation time was reduced by 88.53 % from $t_{calc,BM}$ = 4292.2s (BM) to $t_{calc,EUR}$ = 534.4s, while the average RMS_{rel} was increased by 0.3% only. With this method, respiratory motion prediction with SVR is performed more efficiently compared to the BM implementation. Even though it is still not possible to perform online motion prediction with SVR on an ordinary office computer, because the computation time for updating the model is larger than the sample time. But knowing when to update the model can be useful for implementing efficient multi-thread, online SVR predictors in the future. An example could be a dual-thread predictor, where one thread performs online predictions, while the second thread updates the SVR model. To reduce the calculation time even further, the algorithms could be implemented in C/C++ for instance or on dedicated hardware. Finally, it should be pointed out that EUR is not restricted to respiratory motion signals and can be applied to other applications.

5 References

[1] A. Schweikard, G. Glosser, M. Bodduluri, M. J. Murphy, J. R. Adler, „Robotic Motion Compensation for Respiratory Movement during Radiosurgery“, *Computer Aided Surgery*, Nr. 5, 2000.

[2] F. Ernst, A. Schweikard, „Forecasting respiratory motion with accurate online support vector regression (SVRpred)“, *Int J CARS*, Nr. 5, 2009.

[3] W. D. D'Souza, K. Malinowski, H. H. Zhang, „Machine Learning for Intra-Fraction Tumor Motion Modeling with Respiratory Surrogates“, *International Conference on Machine Learning and Applications*, 2009.

[4] N. Riaz, P. Shanker, R. Wiersma, O. Gudmundsson, W. Mao, B. Widrow, L. Xing, „Predicting respiratory tumor motion with multi-dimensional adaptive filters and support vector regression“, *Physics in Medicine and Biology*, Nr. 19, 2009.

[5] A. Krauss, S. Nill, U. Oelfke, „The comparative performance of four respiratory motion predictors for real-time tumour tracking“, *Physics in Medicine and Biology*, Nr. 16, 2011.

[6] A. J. Smola, B. Schölkopf, „A tutorial on support vector regression“, *Statistics and Computing*, Nr. 3, 2004.

[7] C.-C. Chang, C.-J. Lin, „LIBSVM: A library for support vector machines“, *ACM Trans. Intell. Syst. Technol.*, Nr. 3, 2011.