Support Vector Machines and Learning about Time*

Stefan Rüping and Katharina Morik
University of Dortmund
CS Department, AI Unit
Dortmund, Germany

Abstract

The analysis of temporal data is an important issue in current research, because most real-world data either explicitly or implicitly contains some information about time. The key to successfully solving temporal learning tasks is to analyze the assumptions that can be made and prior knowledge one has about the temporal process of the learning problem and find a representation of the data and a learning algorithm that makes effective use of this knowledge. This paper will present a concise overview of the application of Support Vector Machines to different temporal learning tasks and the corresponding temporal representations.

1 Introduction

There is a multitude of learning tasks related to temporal phenomena and, correspondingly, there are many possible representations for temporal data. Learning tasks and representations are closely related: the No Free Lunch Theorem [25, 26] implies that finding an adequately biased representation can make a hard learing problem easy (and, vice versa, that finding this representation itself is hard).

*A short version of this paper appears in the Proceedings of the IEEE International Conference on Accoustics, Speech and Signal Processing (ICASSP 2003)

Statistical time series analysis has developed two big classes of representations, namely those in the time domain and those in the frequency domain [4]. Analysis in the time domain is based on the correlation between the current and previous observations, while the frequency domain tries to decompose the time series into cyclic components at different frequencies. For learning tasks, time series analysis has the following objectives : description (to describe a time series by certain statistics), explanation (to understand the process behind a time series), prediction (to predict future values of the time series) and control (control the process behind the time series to generate certain future values).

For Machine Learning, an overall theory of temporal analysis is much less developed. Learning tasks are usually taken from specific real-world problems and representations are often constructed ad hoc. Morik [15] differentiates between two different aspects of time, the linear precedence of events and immediate dominance of temporal categories. These terms originate from natural language theory [9]. Immediate dominance refers to the construction of higher-level categories of the time-dependent elements, exemplary learning tasks based on the concept of immediate dominance are the discovery of frequent episodes [14] or first order logic learning based on Allen's interval relations [1]. The aspect of linear precedence refers to the linear temporal order of the single events and is most prominent in the framework of time series analysis.

The focus of this paper lies on the time series representation and the types of learning tasks that can be solved with support vector machines (SVMs, [24]). Support vector machines have been applied to very different kinds of learning problems, for example to time series prediction by Mukherjee et al. in [17] and by Müller et al. in [18]. A regression problem for time series has been solved with SVMs in [20], where certain coefficients of chemical components have been predicted from chromatography time series. Chang et al. [3] have presented an approach for time series segmentation with SVMs, which consists of simultaneously learning multiple SVMs models for one time series.

All of these applications are based on a single time series representation, the phase space representation, i. e. creating d-dimensional examples by moving a window of length d over the time series. The theoretical foundation for this is the theorem of Takens [23, 19] which states that for dynamical systems of a certain type, the phase-space reconstruction and the unobserved internal structure of the system are topologically identical, given the embedding dimension is large enough. At first glance, this seems like the perfect tool: We know that we can find out all we need to know about a time series simply by looking at large enough time windows and in the SVM we have a learning algorithm that is well known to perform well on high-dimensional data. But this conclusion is fallacious. First of all, Taken's theorem does only hold for dynamical systems which can be described by differential equations of a certain form. Generally, one cannot decide whether this is the case for a given real-word data set. Second, the theory of structural risk minimization [24], on which the SVM is based, is only formulated for independent, identically distributed data. Clearly, the independence assumption is violated for time series data. Although versions of the central theorems of structural risk minimization do also hold for dependent data of weak dependence structure [8] and in practice, SVMs have been shown to perform quite well on time series data, one should be careful to transfer results of the SVM to this type of data. Finally, even if the premises of Taken's theorem and the structural risk minimization principle do hold, a different representation of the data may lead to a much easier generalization (it is the "large enough dimension"-part of Taken's theorem that can cause much trouble).

The next section will give a short introduction to Support Vector Machines. In section 3 we will investigate the relation between the SVM and statistical time series modeling, in particular autoregressive models and the Fourier transform. After that, in section 4, we will discuss alternative representations that were used in different applications with real-world data. Finally, section 5 will present novel temporal learning tasks which can be solved using SVMs.

2 Support Vector Machines

Support Vector Machines are based on the work of Vladimir Vapnik in statistical learning theory [24]. Statistical learning theory deals with the question, how a function f from a class of functions $(f_\alpha)_{\alpha \in \Lambda}$ can be found, that minimizes the expected risk

$$R[f] = \int \int L(y, f(x)) dP(y|x) dP(x) \tag{1}$$

with respect to a loss function L, when the distributions of the examples $P(x)$ and their classifications $P(y|x)$ are unknown and have to be estimated from finitely many examples $(x_i, y_i)_{i \in I}$.

The SVM algorithm solves this problem by minimizing the regularized risk $R_{\text{reg}}[f]$, which is the weighted sum of the empirical risk $R_{\text{emp}}[f]$ with respect to the data $(x_i, y_i)_{i=1\ldots n}$ and a complexity term $||w||^2$

$$R_{\text{reg}}[f] = R_{\text{emp}}[f] + \lambda ||w||^2.$$

In their basic formulation, SVMs find a linear decision function $y = f(x) = \text{sign}(w \cdot x + b)$ that both minimizes the prediction error on the training set and promises the best generalization performance. One of the major tricks of SVM learning is the use of kernel functions to extend the class of decision functions to the non-linear case. This is done by mapping the data from the input space X into a high-dimensional feature space $\mathcal{X}$ by a function

$$\Phi : X \to \mathcal{X}$$

and solving the linear learning problem in $\mathcal{X}$. The actual function Φ does not need to be known, it suffices to have a kernel function k which calculates the inner product in the feature space.

$$K(x, y) = \Phi(x) \cdot \Phi(y)$$

3 Time Series Models

Using the phase space model to represent the time series data together with a linear prediction function leads to the class of autoregressive (AR) models [4]. Obviously, AR models can be learned by a SVM with linear kernel, so it does not surprise that the SVM does not perform very different on data generated from an AR model than other methods for AR model estimation, like the Yule-Walker equations. However, it can be seen that the SVM is more robust against outliers in the data. The following table compares the mean absolute error of a AR model learned using the Yule-Walker equations against a SVM model. In the first case, the data was generated from an AR model, in the second case 10% of outliers were added (validation set results averaged over 4 runs with different models).

For time series analysis in the frequency domain, the Fourier transformation can be used to transform the examples for the SVM. There also exist kernel function, which make this transformation explicitly, e. g. the Fourier kernels proposed by Vapnik in [24], Ch. 11. or the time-frequency kernel of

Data	Tool	d=2	d=3	d=4	d=5
AR model	AR	0.640	0.624	0.624	0.633
	SVM	0.648	0.634	0.632	0.639
AR+outliers	AR	0.942	0.922	0.919	0.911
	SVM	0.885	0.826	0.832	0.827

Figure 1: Performance of AR and SVM model.

Davy et al. [7]. Vapnik's kernel is based on the typical kernel trick: While the constructing the Fourier series expansion of a time series is hard, calculating the inner product in the feature space given by the Fourier expansion of order N is easy:

$$K(x_i, x_j) = \frac{sin(\frac{2N+1}{2}(x_i - x_j))}{sin\frac{(x_i - x_j)}{2}}$$

Vapnik also suggests different kernels for regularized Fourier expansions for improved approximation properties. The kernel of Davy et al. is based on Cohen's time-frequency distributions [5], which generalize the Fourier transformation for the analysis of non-stationary time series.

4 Time Series Representations for Real-World Data

Time series models are a well understood research area. However, in practice we see over and over again that much data manipulation has to be done in order to get good results. In this section we will show some examples of representation tricks that can be used when analysing time series with Support Vector Machines. The main advantage of SVMs is that the relevant equations that describe the generelization errors of SVMs [24] do not depend on the dimensionality of the data but only on the margin of the separating hyperplane, which makes the SVM especially suited for high-dimensional data. While this reasoning is not strictly valid - the margin depends on the geometry of the data and hence also on the dimension - empirical evidence show that this property of SVMs does hold in practice (see e. g. [10]). This allows us to enhance the represenation with certain attributes which improve the temporal analysis of the data.

Often, time series can be decomposed into a long-term linear trend, a cyclical component and a rest, where the trend and the cyclical component are fitted before the actual analysis of the rest is being performed. For the trend, when analysing a time series on the basis of the phase space representation and a linear kernel (i. e. using an AR model), we can simply add the time t as another attribute to the data. By this, the SVM will learn a function $w_1 x_1 + \ldots + w_d x_d + w_{d+1} t$, i. e. it will automatically decide, which effects in the data to attribute to the trend $w_{d+1} t$ and which to the rest model $w_1 x_1 + \ldots + w_d x_d$.

The case of cyclical components is more complicated, because they cannot be as easily identified and filtered from the data as a simple trend. But often the most difficult problem in practice is that lots of statistical procedures for modeling periodic functions cannot be applied, because what looks like a periodic component actually is not one. In Figure 2 you can see the weekly sales of a certain item in a retail store. One can easily see the effects of christmas sales in the 51st and preceeding weeks and also the lower effects of easter sales in the 12th week. One can easily imagine that these effects will occur periodic every year at the same time. But they don't! In some years, christmas is in the 51st week of the year, but in some it is already in the 50th week. On a daily time scale, christmas will repeat itself every 365 days in most years, but in 366 days in a leap year. The data of the eastern can vary about five weeks. Therefore, in [21] 20 additional binary attributes were used to mark the presence of holidays, special sale promotions, and other significant events in that particular week. In the example of Figure 2, this reduced mean absolute error by more than 20% from 116.15 to 91.063 (test error over one year, averaged over 4 different stores).

Another case of sales time series is shown in Figure 3. This time, the subject are newspapers and the sales are reported on a daily basis. We can clearly see that there is a cycle of 6 days in the data (the newspaper does not appear on sundays) and that sales are especially high for the weekend edition on saturdays. Hence, instead of using the phase space representation on the original time series, we can also reason that we have indeed 6 different time series of sales (one for each weekday) and use the phase space representation there. And third, we can also use a mixed representation where we use the last couple of days plus the last couple of weekdays to predict the time series. Testing these three representations, we can see that modeling only one time series achieves a mean absolute test error of 8.102, the approach with 6 time series has an error of 5.942 and the mixed approach reaches an error of 5.654.

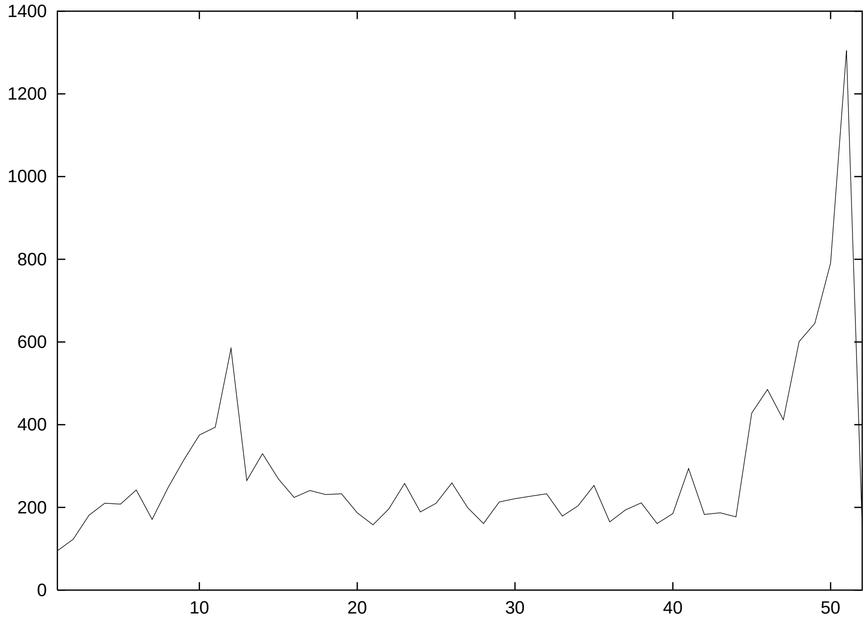

Figure 2: Retail Store Sales per Week.

Sometimes, best results are achieved if one drops the idea of a time series at all. For example, for the task of recommending drug administration from recorded vital signs of intensive care patients - a high dimensional, noisy classification problem on multivariate time series - it was found in [16] that the best representation was to ignore time dependencies completely and make a non-temporal classification based on the last obeservation only. In the field of chromatography, Ritthoff et al. [20] solved the problem of predicting certain chemical coefficients based on the chromatographical analysis of a substance (which is a time series of intensities of chemical components) by describing the time series by chosen analytical properties, e. g. the location of its maximum. That is, they did not use the time series observations at all, but only new, aggregated features. This technique alone reduced the error by 50%.

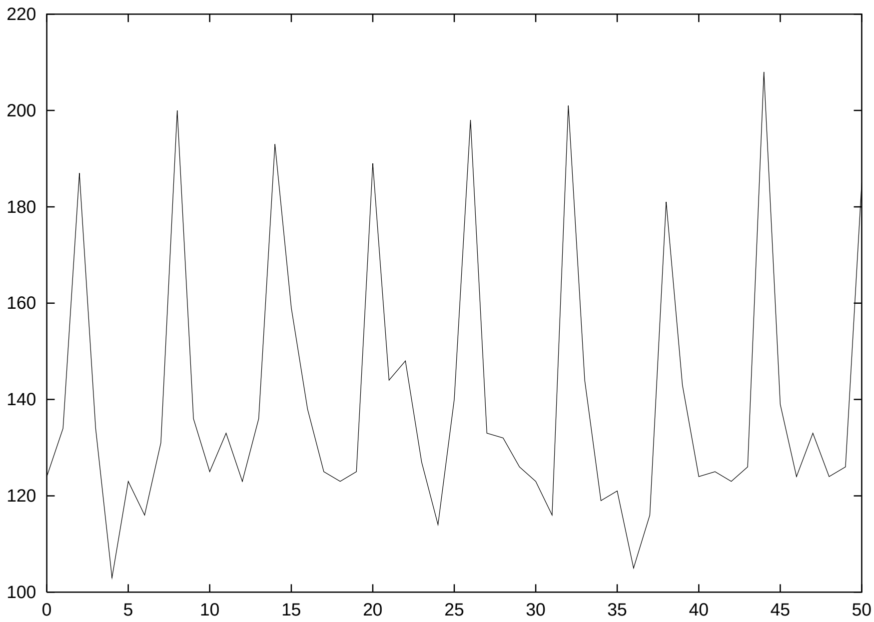

Figure 3: Newspaper Sales per Day.

5 Advanced Temporal Learning Tasks

The phase space represenation bases on the asumption that the temporal dependence structure of the time series can be sufficiently captured in a short finite window of observations. This allows the examples generated from each window to be treated as if they were generated independently and thus, the order in which the examples are presented to the learning algorithm is not important. This assumption fails, if the process that generates the time series changes over time. This scenario is called concept drift. Usually, concept drift is treated by using only a certain number of the newest examples, where the actual number of examples used is chosen heuristically. For SVMs, Klinkenberg and Joachims [12] proposed an approach where this number is chosen based on efficient performance estimators for SVMs [11]: for each possible number b of batches, a SVM is learned on the newest b batches and its leave-one-out performance on the last batch of data is estimated. The final classifier is the SVM with the best estimated performance.

This approach was generalized in two ways in [13]. By assigning a specific weight to each example, one can limit the influence that this examples has on the final decision function. For temporal data, the weight can be set according to the age of the example, giving less influence to older, possibly outdated examples. As an alternative, one can learn a temporary decision function on only the newest batch of data and use this function to identify all examples that correspond to the newest model. All batches, where the temporary classifier performs significantly worse than on the newest set, are discarded from the final training set, from which the final classifier is learned. This last approach is of advantage, if the change in the process behind the model is very sharp instead of a slow drift. The scenario is called concept shift.

Another interesting problem is the detection of outliers in time series. The procedure of Bauer [2] constructs a certain ellipse (derived from an assumed autoregessive model) around the phase space representation of the time series all points outside of this ellipse are delcared as outliers. This procedure can be generalized by using a SV estimation of the support of the points [22] in the phase space. This directly applies the definition of Davies and Gather [6] of the α-outlier-region as the region of points with the lowest probability, so that the probabilty of the whole region is α.

6 Summary and Conclusions

This paper presented a concise overview of time series analysis using Support Vector Machines. Of course, due to space constraints we could not cover all existing approaches in as much detail as they deserved.

In conclusion, we find that the key to successful time series analysis with SVMs lies in finding the right represenation. The excellent generalization properties of the SVM, especially its good performance on high-dimensional data, make it easy to improve results by adding additional temporal features or constructing specialised kernel functions.

Acknowledgments

The financial support of the Deutsche Forschungsgemeinschaft (SFB 475, "Reduction of Complexity for Multivariate Data Structures") is gratefully acknowledged.

References

[1] J. F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154, 1984.

[2] M. Bauer, U. Gather, and M. Imhoff. Analysis of high dimensional data from intensive care medicine. In R. Payne, editor, *Proceedings in Computational Statistics*, Berlin, 1999. Springer Verlag.

[3] Ming-Wei Chang, Chih-Jen Lin, and Ruby C. Weng. Analysis of nonstationary time series using support vector machines. In Seong-Whan Lee and Alessandro Verri, editors, *Pattern Recognition with Support Vector Machines — First International Workshop, SVM 2002*, LNCS 2388, pages 160–170. Springer, Aug 2002.

[4] Christopher Chatfield. *The Analysis of Time Series: An Introduction.* Chapman and Hall, 3rd edition, 1984.

[5] Leon Cohen. Time-frequency distributions - a review. *Proceedings of the IEEE*, 77(7):941–981, July 1989.

[6] Laurie Davies and Ursula Gather. The identification of multiple outliers. *J. Am. Statist. Ass.*, 88:782–792, 1993.

[7] M. Davy, A. Gretton, A. Doucet, and P.W.J. Rayner. Optimised support vector machines for nonstationary signal classification. *IEEE transactions on Signal Processing*, 9(12), Dec. 2002.

[8] Thomas Fender. *Empirische Risikominimierung für dynamische Datenstrukturen.* PhD thesis, Department of Statistics, University of Dortmund, Germany, work in progress.

[9] Gerald Gazdar and Chris Mellish. *Natural Language Processing in PROLOG.* Addison Wesley, Workingham u.a., 1989.

[10] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of the European Conference on Machine Learning*, pages 137 – 142, Berlin, 1998. Springer.

[11] Thorsten Joachims. Estimating the generalization performance of a SVM efficiently. In Pat Langley, editor, *Proceedings of the International Conference on Machine Learning*, pages 431–438, San Francisco, 2000. Morgan Kaufman.

[12] Ralf Klinkenberg and Thorsten Joachims. Detecting concept drift with support vector machines. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 487–494, San Francisco, CA, USA, 2000. Morgan Kaufmann.

[13] Ralf Klinkenberg and Stefan Rüping. Concept drift and the importance of examples. In Jürgen Franke, Gholamreza Nakhaeizadeh, and Ingrid Renz, editors, *Text Mining – Theoretical Aspects and Applications*. Springer, Berlin, Germany, 2002. To appear.

[14] H. Mannila, H. Toivonen, and A. Verkamo. Discovering frequent episode in sequences. In *Procs. of the 1st Int. Conf. on Knowledge Discovery in Databases and Data Mining*. AAAI Press, 1995.

[15] Katharina Morik. The representation race - preprocessing for handling time phenomena. In Ramon López de Mántaras and Enric Plaza, editors, *Proceedings of the European Conference on Machine Learning 2000 (ECML 2000)*, volume 1810 of *Lecture Notes in Artificial Intelligence*, Berlin, Heidelberg, New York, 2000. Springer Verlag Berlin.

[16] Katharina Morik, Michael Imhoff, Peter Brockhausen, Thorsten Joachims, and Ursula Gather. Knowledge discovery and knowledge validation in intensive care. *Artificial Intelligence in Medicine*, 19(3):225–249, 2000.

[17] Sayan Mukherjee, Edgar Osuna, and Frederico Girosi. Nonlinear prediction of chaotic time series using support vector machines. In *Proc. of IEEE NNSP 97*, Amelia Island, FL, Sep 1997.

[18] K. Müller, A. Smola, G. Ratsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In *Proceedings of the International Conference on Artificial Neural Networks*, Springer Lecture Notes in Computer Science. Springer, 1997.

[19] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Physical Review Letters*, 45:712–716, 1980.

[20] Oliver Ritthoff, Ralf Klinkenberg, Simon Fischer, and Ingo Mierswa. A hybrid approach to feature selection and generation using an evolutionary algorithm. In John A. Bullinaria, editor, *Proceedings of the 2002 U.K. Workshop on Computational Intelligence (UKCI-02)*, pages 147–154, Birmingham, UK, september 2002. University of Birmingham.

[21] Stefan Rüping. Zeitreihenprognose für Warenwirtschaftssysteme unter Berücksichtigung asymmetrischer Kostenfunktionen. Master's thesis, Universität Dortmund, 1999. in German.

[22] Bernhard Schölkopf, Robert C. Williamson, Alex J. Smola, and John Shawe-Taylor. SV estimation of a distribution's support. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Neural Information Processing Systems 12*. MIT Press, 2000.

[23] F. Takens. Detecting strange attractors in turbulence. In D. A. Rand and L. S. Young, editors, *Dynamical systems and turbulence*, volume 898 of *Lecture Notes in Mathematics*, pages 366 – 381. Springer, Berlin, 1980.

[24] V. Vapnik. *Statistical Learning Theory*. Wiley, Chichester, GB, 1998.

[25] D.H. Wolpert and W.G. Macready. No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fé Institute, Santa Fé, CA., 1995.

[26] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimisation. *IEEE Trans. on Evolutionary Computation*, 1:67 – 82, 1997.