Incorporating background knowledge for better prediction of cycle phases

Ursula Sondhauss*and Claus Weihs
Department of Statistics, University of Dortmund
Vogelpothsweg 87, 44221 Dortmund, Germany
Email: weihs,sondhaus@statistik.uni-dortmund.de
Tel.: ++49 (0)231/755-5853,Fax:-4387

May 18, 2001

Abstract

When predicting the state of a system, we sometimes know that the succession of states is cyclic. This is for example true for the prediction of business cycle phases, where an upswing is always followed by upper turning points, and the subsequent downswing passes via lower turning points over to the next upswing and so on. We present several ideas how to implement this background knowledge in popular static classification methods. Additionally, we present a full dynamic model. The usefulness for the prediction of business cycles is investigated, revealing pitfalls and potential benefits of ideas.

1 Introduction

In the literature, business cycles are typically either treated as a univariate phenomenon and tackled by univariate time series methods, or they are modelled as a *multivariate* phenomenon and tackled by static multivariate classification methods [Meyer and Weinberg, 1975; Heilemann and Münch, 1996]. As a consequence, either the time-dependency or the interplay of different economic variables is ignored.

*This work has been supported by the Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 475. It will be presented at the workshop 'Learning from Temporal and Spatial Data' on the 'Seventeenth International Joint Conference on Artificial Intelligence' August 4th - 10th, 2001 Seattle, Washington, USA

In a preliminary comparative study we showed that multivariate classification methods (ignoring knowledge about time-dependencies) and a dynamic Bayesian network that generalizes the Naive Bayes classifier for time-dependencies (ignoring dependencies between predictors) obtained about the same, unsatisfying, average prediction accuracy.

Thus, in that study, some multivariate classification methods generated the same error rates as the dynamic Bayesian network without using background knowledge about time dependencies in business cycles. Therefore, there was hope that in order to improve prediction accuracy for the multivariate classification methods advantage could be taken of the cyclical structure of business cycle phases for which the following pattern is true: lower turning points $\hookrightarrow$ upswing $\hookrightarrow$ upper turning points $\hookrightarrow$ downswing $\hookrightarrow$ lower turning points $\hookrightarrow$ and so on.

In this paper, we introduce and analyze several ideas on the incorporation of this background knowledge in different types of classification rules. The general problem of predicting cycles is formulated in Section 2. In Section 3 ideas on adapting static classification rules to the above cyclical structure are described. The data used for learning and testing the prediction models for business cycle phases and the design of our comparative study are presented in Section 4 and all the considered classification methods are briefly outlined. In Section 5 we compare the performance of the implemented ideas. And finally, consequences are summarized in Section 6.

2 Basic Notations

We consider classification problems that are based on some K-dimensional real-valued vector $\vec{x} \in \mathbf{M} \subset I\!R^K$ of observations of predictor variables $X_1, ..., X_K$ on some object and we want to decide about the class $s \in \mathbf{S} := \{1, ..., J\}$ the object belongs to. Any considered object with $\vec{x} \in \mathbf{X}$ has to belong to one and only one out of these J classes.

In case of prediction of cycle phases, we classify not really various objects, but rather one object - called *system* - at different times $t = 1, ..., T$. And at each time-point the system is situated in one out of J possible *states* $s \in \mathbf{J} := \{1, ..., J\}$. The chronological order of how the system passes through states is fixed: Given the system is is state s_{t-1} at time $t-1$, it either stays there or moves on to a certain other state $s^{\oplus}$ so that $S_t \in \{s, s^{\oplus}\} \subset \mathbf{S}$, $t = 1, ..., T$. In the following, we assume a corresponding numbering of states where $s^{\oplus} = s+1$ for $s = 1, 2, ..., J-1$ and $s^{\oplus} = 1$ for $s = J$.

Most classification methods base their assignment of objects into classes on certain transformations of the respective observations for each of the considered classes:

$$m(s, \circ) : \mathbf{X} \rightarrow I\!R, \; s \in \mathbf{S}.$$

The size of these transformations gives information on the strength of membership of the object in the classes. Without loss of generality, we assume higher values to indicate stronger membership. That is, these $m(s, \vec{x})$, $s \in \mathbf{S}$, $\vec{x} \in \mathbf{X}$ are interpreted as *membership values*.

There are many ways and intuitions for the construction of membership values using examples of observations and classes for some objects in a learning set $\mathbf{L}$: In discriminant analysis (Linear: LDA, or quadratic: QDA) membership values are based on some notion of distances to estimated centers of classes, whereas in Support Vector Machines (SVM) we use distances to learnt class boundaries.

For all Bayes rules, membership values are estimated conditional class probabilities:

$$\begin{aligned} m(s, \vec{x}) &= p_{\mathbf{L}}(s \mid \vec{x}) \\ &= \frac{p_{\mathbf{L}}(\vec{x} \mid s) p_{\mathbf{L}}(s)}{p_{\mathbf{L}}(\vec{x})} \end{aligned} \tag{1}$$

for each class $s \in \mathbf{S}$.

Irrespective of the various derivations of membership values, the manner of assignment is always the same: The rule assigns to the class with highest membership value. Therefore, we call this type of rules *argmax rules*.

For multi-class problems, there are two distinct basic structures to decide on a certain elementary state $s, s \in \mathbf{S}$ where the cyclical structure can easily be implemented: multi-class argmax rules or composition of binary argmax rules.

3 Adaption of static classification rules for prediction of cycle phases

3.1 Adapting multi-class argmax rules

Multi-class argmax rules use membership values for each elementary state $m(s, \circ) : \mathbf{X} \to I\!R, s \in \mathbf{S}$:

$$\hat{s} = \mathbf{pred}(\vec{x}) = \arg \max_{s=1,\dots,s} m(s, \vec{x}).$$

For these rules, we can take advantage of the cyclical structure by restricting the comparison of membership values to admissible transitions. That is, we start in the last known state of the system s_0 and predict the next state by

$$\mathbf{pred}(\vec{x}_1 \mid s_0) \quad = \quad \arg \max_{s=s_0, s_0^{\oplus}} m(s, \vec{x}_1).$$

For the consequent times $t = 2, ..., T$ the predicted state $\hat{s}_{t-1}$ from the preceding time is used as if it was the true one:

$$\begin{aligned}
\mathbf{pred}(\vec{x}_t \mid s_0, \hat{s}_1, ..., \hat{s}_{t-1}) &= \mathbf{pred}(\vec{x}_t \mid \hat{s}_{t-1}) \\
&= \arg \max_{s=\hat{s}_{t-1}, \hat{s}_{t-1}^{\oplus}} m(s, \vec{x}_t).
\end{aligned}$$

This adaption was proposed by Weihs *et al* [1999] for the prediction of business cycle phases and is called *classification with exact transitions*. When classifying with exact transitions in a first step the information of $\hat{s}_{t-1}$ is used to decide on admissible states $\hat{s}_t \in \left\{\hat{s}_{t-1}, \hat{s}_{t-1}^{\oplus}\right\} \subset \{1, 2, ..., J\}$. In a second step we choose between those two, using the information in $\vec{x}_t$. In the following, we will drop the time-index t and denote variables from time-slice $t-1$ with a minus: $v_- := v_{t-1}$, $t = 1, ..., T$, if statements are valid for all $t = 1, ..., T$, and where indexing is not needed for understanding.

We may gain further improvement of the rules, if we can exploit the information in $\hat{s}_{t-1}$ also for the second decision. For membership values on a ratio scale, we can do this by weighting membership values with transition weights that tell us something about the willingness of the system to pass over to state s given state $\hat{s}_-$:

$$m(s, \vec{x} \mid \hat{s}_-) = w(\hat{s}_-, s) m(s, \vec{x}).$$

In cyclic systems, $w(s_-, s) = 0$ is true for all inadmissible transitions $s_- \hookrightarrow s$, $s_-, s \in \mathbf{S}$.

How weighting works, and why membership values have to be on a ratio scale, can be understood best by looking at the following representation of the weighted rule:

$$\begin{aligned}
\mathbf{pred}(\vec{x} \mid \hat{s}_-) &= \arg \max_{s=\hat{s}_-, \hat{s}_-^{\oplus}} w(\hat{s}_-, s) m(s, \vec{x}) \\
\leftrightarrow \mathbf{pred}(\vec{x} \mid \hat{s}_-) &=
\end{aligned}$$

$$\left\{ \begin{array}{c} \hat{s}_- \\ \hat{s}_-^{\oplus} \end{array} \right\} \text{ if } \frac{m(\hat{s}_-, \vec{x})}{m(\hat{s}_-^{\oplus}, \vec{x})} \frac{w(\hat{s}_-, \hat{s}_-)}{w(\hat{s}_-, \hat{s}_-^{\oplus})} \left\{ \begin{array}{c} \geq 1 \\ < 1 \end{array} \right\}.$$

We are coding the evidence of $\vec{x}$ for S being either $\hat{s}_-$ or $\hat{s}_-^{\oplus}$ with the ratio of the corresponding membership values, as well as the (assumed) evidence of $S_- = \hat{s}_-$ with the ratio of the transition weights. And we combine these evidences by multiplying and thus giving both evidences same importance.

An intuitive choice of weights are estimated transition probabilities from the training set, e.g. the observed frequencies:

$$\begin{aligned}
w_{\mathbf{L}}(\hat{s}_-, s) &:= p_{\mathbf{L}}(s \mid \hat{s}_-) \\
&:= \frac{N_{\mathbf{L}}(\hat{s}_- \hookrightarrow s)}{N_{\mathbf{L}}(\hat{s}_-)}.
\end{aligned}$$

In case of membership values that use estimates of a-priori class probabilities, like bayes-rules with unequal class probabilities $p_{\mathbf{L}}(s)$ we use the ratio

$$\frac{p_{\mathbf{L}}(s \mid \hat{s}_{-})}{p_{\mathbf{L}}(s)}$$

is chosen as weights. We simply *replace* $p_{\mathbf{L}}(s)$ by $p_{\mathbf{L}}(s|\hat{s}_{-})$ in the calculation of membership values for bayesrules in equation (1):

$$\begin{aligned} m(s, \vec{x} \mid \hat{s}_{-}) &= m(s, \vec{x}) \frac{p_{\mathbf{L}}(s \mid \hat{s}_{-})}{p_{\mathbf{L}}(s)} \\ &= \frac{p_{\mathbf{L}}(\vec{x} \mid s) p_{\mathbf{L}}(s|\hat{s}_{-})}{p_{\mathbf{L}}(\vec{x})}. \end{aligned} \tag{2}$$

The resulting membership values can be interpreted as estimated conditional class probability given $\vec{x}$ and $\hat{s}_{-}$ under the additional assumption of conditional independence of $\vec{X}$ and S_{-} given $S = s$. This is the well-known assumption in Hidden Markov Models (HMMs) of order one: all relevant past information $s_0, \vec{x}_0, ..., s_{t-1}, \vec{x}_{t-1}$ is summarized in the last state s_{t-1} and is propagated solely through the transition probabilities $p(s_t|s_{t-1}) \equiv p(s|s_{-})$, $s_t = 1, ..., J$, $t = 1, ..., T$.

3.2 Composition of binary elementary argmax rules

Other multi-class rules use membership values not for elementary states but for various sets out of the product set over $\{1, ..., J\}$, $m : \mathbf{X} \rightarrow I\!R^{2^J - 1}$. This is true, for example, if the final decision consists of a path of binary argmax decisions.

In the so-called *one-against-rest* strategy each class is trained against the other $J - 1$ classes [Schölkopf *et al* (1995)]. The class with the highest value in the decision function is selected. So J argmax rules have to be trained.

An example is the max win strategy of [Friedmann (1996)]. Each class is trained against every other class with a binary SVM. Thus we get a collection of J(J-1)/2 membership functions

$$m(s, s', \dot{}) : \mathbf{X} \rightarrow I\!R, s' = 1, ..., s - 1, s = 2, ..., J.$$

The class that obtains the most votes is selected. If this is not unique (i.e. two or more classes get the most votes) between these classes the one with the highest value in the membership function gets assigned.

Another strategy uses decision directed acyclic graphs (DDAG) [Platt *et al.*(2000)]. Classes are listed and the first decision is made between the first and the last element on the list. The one which is not voted for is eliminated from the list. This is repeated until only one class is left and the observation gets assigned to it. The same argmax rules as in the max win strategy have to be learnt, but to make a decision only $J-1$ decision nodes in the DDAG have to be evaluated, and each is constructed only on the two classes which are examined.

For the appropriate strategy in a cyclic structure only J membership functions for binary argmax rules have to be learnt, namely $m(1,2,\circ)$, $m(2,3,\circ)$,..., and $m(J,1,\circ)$. Dependent on the state s_0 or respectively the predicted state $\hat{s}_-$ from the preceding time slice, we decide on the current state $\hat{s}$ based on $m(\hat{s}_-,\hat{s}_-^{\oplus},\circ)$.

4 Design of Comparison

4.1 Data

The data set consists of 13 "stylized facts" [Lucas (1987)] for the German business cycle and 157 quarterly observations from 1955/4 to 1994/4 (price index base is 1991). The stylized facts are given in table 4.1.

The experts' classification of the data into business cycle phases (abbreviated as PH) was done by Heilemann and Münch [1996] using a 4-phase scheme. Phases are called *lower turning points* (abbreviated "ltp"), *upswing* ("up"), *upper turning points* ("utp"), and *downswing* ("down").

IE	real investment in equipment-gr
C	real private consumption-gr
Y	real gross national product-gr
PC	consumer price index-gr
PY	real gross national product price deflator-gr
IC	real investment in construction-gr
LC	unit labor cost-gr
L	wage and salary earners-gr
M1	money supply M1
RL	real long term interest rate
RS	nominal short term interest rate
GD	government deficit
X	net exports (X)

Table 1: **Our predictors** of business cycle phases are based on economic aggregates that cover all important economic fields: real activity (labor market, supply/demand), prices, and monetary sphere. The abbreviation 'gr' stands for growth rates with respect to last year's corresponding quarter.

4.2 Design

There are six full cycles in the considered quarters. All methods (have to) rely on some assumption of structural stability over this period, though this is not really valid. Thus, we decided to perform a leave-one-cycle out analysis for the comparison of methods.

For a fair comparison, all optimization in order to gain a rule has to be done on each of the six training sets alone. Rules are then compared with respect to their prediction accuracy measured as the average prediction error on the validation sets:

$$\mathbf{APE} := \frac{1}{6}\sum_{i=1}^{6}\left(\frac{1}{T_i}\sum_{t=1}^{T_i}\mathbb{I}_{s_t}(\hat{s}_t)\right),$$

where T_i is the number of time-points in the i-th cycle, $i = 1, ..., 6$, and $\mathbb{I}_s$ is the indicator function for state $s \in \mathbf{S}$.

This gives an average error on a new cycle, which seems to be more appropriate as performance criterion than the average error on a single new observation. Cycles form a natural entity in the given task, and the structural instability across cycles together with a performance criterion based on single observations would lead to an unwanted preference of methods that predict well on long cycles.

4.3 Description of static procedures

In the past, mainly static classification methods were used for the multivariate prediction of business cycles. One reason is the fact that typically the last true phase is not known (for sure) to do the prediction for the next one. It is only by observing the continuing evolution of the economy for some more quarters that it becomes apparent what phase the business cycle was in. Another reason for using static methods is that we are not only reaching for a good prediction, but also for a description of phases in terms of the stylized facts. Thus methods were applied that use as predictors known entities, and for which we want to understand the connection they have with business cycle phases.

The ideas of modifying static methods outlined in Section 3 now allow for both, description and prediction: we describe phases using their membership functions $m(s, \circ) : \mathbf{X} \to I\!R$, $s \in \mathbf{S}$ and we hope to get better predictions when combining their evidence with the knowledge on the cyclical structure. We do not base our classification rule on unknown entities, as our predictions are dependent on the true state of the system at some point in the past and not necessarily on the last one.

In the following, we give a short description of those static methods that have already been applied to the problem, and of which we had membership values for their prediction on the test cycles, so that the ideas could be easily implemented.

- **Name**: Linear Discriminant Analysis
- **Short**: LDA

Bayes rule with uniform class prior and equal costs. Equivalent to Fisher Discriminant Analysis. No model selection involved.

- **Name**: LDA with variable selection
- **Short**: LDA-VS

Policy as for LDA. Optimization of subset of predictors in terms of lowest leave-one-observation-out error.

- **Name**: Quadratic Discriminant Analysis
- **Short**: QDA

Policy as for LDA. No model selection involved.

- **Name**: QDA with variable selection
- **Short**: QDA-VS

Policy as for LDA. Optimization of subset of predictors in terms of lowest leave-one-observation-out error.

- **Name**: Minimal Error Classifier of type 1 based on QDA
- **Short**: Mec1-Q

Bootstrap estimation of errors in projected 2-dimensional spaces. Optimization of projected space in terms of lowest estimated error with Simulated Annealing and the Nelder/Mead algorithm.

- **Name**: Minimal Error Classifier of type 2 based on QDA
- **Short**: Mec2-Q

Estimation of errors in projected 2-dimensional spaces by the uniformly minimal variance unbiased estimator on the original space followed by numerical integration. Optimization of projected space in terms of lowest estimated error with Simulated Annealing and the Nelder/Mead algorithm.

All these algorithms were programmed in R and C with Numerical Recipes [Press *et al.* (1993)]. Details can be found in [Röhl and Weihs (1999)].

- **Name**: Neural Networks
- **Short**: NN

Multi-Layer Perceptron with one hidden layer and sigmoid activation function. Conjugate gradient method for change of weights. Neural Connection© 2.0 [1997] was used. Optimization of weights and number of nodes in hidden layer in terms of errors on 10% test set.

- **Name**: Binary linear Support Vector Machines with one-against-rest strategy
- **Short**: SVM-OR

Quadratic optimization problem solved with active set algorithm [Fletcher (1981)] in SAS/IML. Leave-one-observation-out optimization of error-penalty parameter

- **Name**: Binary linear Support Vector Machines with max-win strategy based on one-against-one comparisons
- **Short**: SVM-OO

Quadratic optimization as for SVM-OR. Leave-one-observation-out optimization of error-penalty parameter

- **Name**: Binary linear Support Vector Machines using a Decision Directed Acyclic Graph
- **Short**: SVM-DD

Quadratic optimization as for SVM-OR. Leave-one-observation out optimization of error-penalty parameter.

4.4 Description of dynamic procedures

In our study, there is one classification method that is based on multivariate time-series model: the so-called rake-model [Sondhauss and Weihs (1999)] . This is a dynamic Bayesian network with two time-slices, where the multivariate distribution of predictors and state in a time-slice is dependent on their realization in the preceding time-slice in a certain way. The assumed stochastic independencies within a time-slice reflect those of the Naive-Bayes classifier. The independence assumptions between time-slices broaden those of HMMs to allow for time-dependencies between predictor variables. The rake model is a multivariate version of so-called Markov regime switching models introduced by Hamilton [1989] that goes beyond their typical application for predicting switches between two regimes based on one observational variable modeled as conditional Gaussian variable [Diebold and Rudebusch (1996)].

The distribution of each predictor variable $X_{t,k}$ is modeled to be dependent not only on the current state s_t (like in HMMs), but also on its predecessor $X_{t-1,k}$, $k = 1, ..., K$, $t = 1, ..., T$. But it is assumed to be conditionally independent of all other past and current variables (like in the Naive Bayes classifier), given the current state and the predecessor:

$$P\left(X_{k,t} | s_t, \{x_{1,t}, ..., x_{K,t}\} \backslash x_{k,t}, s_{t-1}, \vec{x}_{t-1}, ..., s_0\right)$$

$$= P\left(X_{k,t}|s_t, x_{k,t-1},\right), t = 1, ..., T, k = 1, ...K.$$

The current state S_t is conditionally independent of the past $S_0, \vec{X}_0, S_1, \vec{X}_1, ..., \vec{X}_{t-1}$ given the preceding state $S_{t-1}=s_{t-1}$:

$$P\left(S_t|s_{t-1}, \vec{x}_{t-1}, ..., s_0\right) \ = \ P\left(S_t|s_{t-1}\right), t = 1, ..., T.$$

This is different from the Naive Bayes classifier, where (non-conditional) independence of S_t and the past is assumed, $t = 1, ..., T$.

The conditional independence assumptions in the rake-model lead to

- a decomposition of the joint probabilities of state variable and predictor variables in time-slice t given s_{t-1} and $\vec{x}_{t-1}$, so that the conditional class probabilities can be calculated as follows:

$$\begin{aligned} & p\left(s_t \mid \vec{x}_t|s_{t-1}, \vec{x}_{t-1}\right) \\ &= \ p\left(s_t|s_{t-1}\right) \frac{p(\vec{x}_t \mid \vec{x}_{t-1}, s_t)}{p(\vec{x}_t \mid \vec{x}_{t-1}, s_{t-1})}, t = 1, ..., T. \end{aligned}$$

- a decomposition of the conditional probability of predictor variables given current state and predecessor, so that it resolves into:

$$p(\vec{x}_t|\vec{x}_{t-1}, s_t) \ = \ \prod_{k=1}^{K} p(x_{k,t}|x_{k,t-1}, s_t), t = 1, ..., T.$$

For the given problem, we discretized the observed values of stylized facts: $x_k \to q_k, q_k \in \{1, ..., Q_k\}, k = 1, ..., K$. There were either two or three intervals defining the discretization , that is $Q_k \in \{2, 3\}$, $k = 1, ..., K$.

For the estimation of transition probabilities we used observed frequencies on the training sets. This maximum likelihood estimation would not have been a good choice for the estimation of the probabilities $p(q_k|q_k^-, s^-)$, because for some specifications of $s \in \{1, ..., J\}$ and $q_k^- \in \{1, ..., Q_k\}$ there are only very few observations or even none in the training sets. Thus we used bayesian parameter learning with uniform dirichlet priors. For further details, see Sondhauss and Weihs [1999]. The equivalent sample sizes for the dirichlet priors were optimized using a leave-one-observation-out analysis.

Exact forward propagation of evidence in dynamic Bayesian networks was used to predict the phase of the cycle at time-points t given the evidence of the last known state s_0 and observations $\vec{x}_1, ..., \vec{x}_t$, $t = 1, ..., T$:

$$\hat{s}_t \ = \ \arg\max_{s \in \mathbf{S}} p_{\mathbf{L}}(s \mid \vec{x}_t, ..., \vec{x}_1, s_0)$$

5 Results

In general, the performance of classification rules for the prediction of business cycle phases is pretty low, as it can be seen in the first column of Table 2: at best we get an error rate of 37%. This is not surprising, given the difficulties of the problem, namely the complex and changing dependencies. Quite a surprise, though, is the even poorer performance, when classifying with exact transitions.

To see how this can happen, we looked at the classifiers predictions for cycles. Typical for the pitfall we ran into is the following course of predictions of the modified NN^+ classifier for the third cycle, compared to the static classifier NN and the true phases in Table 3.

	Average Prediction Error		
	Static	Exact	
Method		equal	estimated
LDA	0.52	0.60	0.55
QDA	0.53	0.60	0.61
LDA-VS	0.52	0.63	0.52
QDA-VS	0.51	0.52	0.53
Mec1-Q	0.46	0.55	0.52
Mec2-Q	0.37	0.44	0.44
NN	0.37	0.53	–
SVM-OR	0.55	0.56	–
SVM-OO	0.50	0.55	–

Table 2: Average Prediction Errors Using Exact Transitions

NN	3 1 1 1 2 1 1 2 1 2 3 2 3 3 3 3
NN^+	3 3 4 1 2 2 2 2 2 2 3 3 3 3 3 3
True	4 1 1 1 1 1 1 2 2 2 3 3 3 3 3 3

Table 3: Predictions of the NN classifier with and without exact transitions on the third cycle compared to the true phases coded as LTP=4, Up=1, UTP=2 and Down=3

Once the classifier has mispredicted, it had big difficulty to predict the phase for the consequent quarters, because it is only allowed to compare for example upper turning points (2) with downswing (3), where the evidence in the predictor variables potentially indicates the true upswing (1), and might be no real help. After an error, either the classifier 'waits' in the mispredicted state for the cycle to pass that state (like in the example), or it passes through all states, until prediction and true state meet again.

The importance of this risky behaviour of the classification with exact transition is emphasized by the average local error rates given in Table 4 calculated for the various methods for the four phases: the turning point phases are particularly hard to identify, thus the probability that we get trapped is very high.

	Local Error Rate			
Method	LTP	Up	UTP	Down
LDA	0.72	0.32	0.81	0.51
QDA	0.97	0.26	0.92	0.33
LDA-VS	0.74	0.36	0.69	0.46
QDA-VS	0.90	0.40	0.86	0.16
Mec1-Q	0.56	0.38	0.84	0.46
Mec2-Q	0.38	0.31	0.82	0.30
NN	0.76	0.24	0.41	0.35
SVM-OR	0.82	0.39	0.67	0.49
SVM-OO	0.66	0.40	0.73	0.47

Table 4: Local Errors Rates

For example, when we "help" the classifiers to identify the beginning of a new cycle, by correctly starting in a lower turning point instead of a downswing, the difficulty to identify this transition is circumvented, and all of a sudden the results change quite a lot, as you can see in Table 5.

	Average Prediction Error		
	Static	Exact	
Method		equal	estimated
LDA	0.52	0.41	0.48
QDA	0.53	0.70	0.70
LDA-VS	0.52	0.55	0.49
QDA-VS	0.51	0.41	0.40
Mec1-Q	0.46	0.44	0.41
Mec2-Q	0.37	0.35	0.29
NN	0.37	0.34	–
SVM-OR	0.55	0.56	–
SVM-OO	0.50	0.xy	–

Table 5: Average Prediction Errors Using Exact Transitions Given True First Phase

For the "good" methods Mec2-Q and NN we now observe an improvement in the APE. But of course, changing the starting value in a leave-one-cycle-out analysis so that a certain phase transition no longer needs to be identified, is cheating: in real life we are highly interested in identifying phase transitions correctly.

So we have to find another way to help classifiers out of the trap: we no longer propagate the predicted state as the true one, but we propagate the probability that a certain state is true, given the state s_0 at time-point $t_0 := 0$ and the past observations of predictor variables. Of course, we can only hope this strategy to be useful for probabilistic rules, where membership values have some interpretation as conditional class probabilities. The first step is the same as before. We predict $\hat{s}_1$ using $\vec{x}_1$ and s_0:

$$\mathbf{pred}(\vec{x}_1 \mid s_0) \quad = \quad \arg \max_{s=s_0, s_0^{\oplus}} p_{\mathbf{L}}(s \mid \vec{x}_1, s_0).$$

The prediction of $\hat{s}_2$ is different. Instead of assuming $\hat{s}_1 = \mathbf{pred}(\vec{x}_1 \mid s_0)$ to be the true state, we propagate to be in state s_0 with probability $p_{\mathbf{L}}(s_0 \mid \vec{x}_1, s_0)$ and in state $s_0^{\oplus}$ with probability $(1 - p_{\mathbf{L}}(s_0 \mid \vec{x}_1, s_0))$.

Thus, the probability to be in state $s_0^{\oplus}$ now is the sum of the probabilities of the two paths that can lead from s_0 to $s_0^{\oplus}$: $s_0 \to s_0 \to s_0^{\oplus}$ and $s_0 \to s_0^{\oplus} \to s_0^{\oplus}$:

$$\begin{aligned} & p_{\mathbf{L}}(s_2 \mid \vec{x}_2, \vec{x}_1, s_0) \\ & \quad = \sum_{s=s_0, s_0^{\oplus}} p_{\mathbf{L}}(s_2 \mid \vec{x}_2, s) p_{\mathbf{L}}(s \mid \vec{x}_1, s_0). \end{aligned}$$

Later, more than two states are possible and the general rule for prediction use conditional state probabilities recursively calculated by:

$$\begin{aligned} & p_{\mathbf{L}}(s_t \mid \vec{x}_t, ..., \vec{x}_1, s_0) \\ & \quad = \sum_{s \in \mathbf{S}} p_{\mathbf{L}}(s_t | s, \vec{x}_t, ..., \vec{x}_1, s_0) p_{\mathbf{L}}(s | \vec{x}_{t-1}, ..., \vec{x}_1, s_0) \end{aligned}$$

Actually, we add-on the structure of a HMM on classification rules (lets denote this by HMM-CR) and predict phases using the forward procedure for finding the next state. The parameters of the distribution of the HMM-CR are defined separately for the transition probabilities and the so-called *emission probabilities* of HMMs, the $p(\vec{x}|s)$, $\vec{x} \in \mathbf{X}$, $s \in \mathbf{S}$. The transition probabilities are either set to be equal for admissible states (non-weighted comparison) or estimated as observed frequencies on the training set (weighted comparison). The emission probabilities comply with the estimated conditional probabilities on the training set that were used to build the rule (see equation 1).

The average prediction errors in Table 6 show that the *classification with forward propagation* leads to an improvement of results for most classifiers, though the size of improvement is disappointing:

	Average Prediction Error		
	Static	Propagated	
Method		equal	estimated
LDA	0.52	0.54	0.50
QDA	0.53	0.50	0.50
LDA-VS	0.52	0.54	0.52
QDA-VS	0.51	0.51	0.52
Mec1-Q	0.46	0.45	0.45
Mec2-Q	0.37	0.34	0.38
NN	0.37	0.37	–
SVM-OR	0.55	0.56	–
Rake	–	–	0.36

Table 6: Average Prediction Errors Using Forward Propagation

Whether or not a weighting of membership values with estimated transition probabilities leads to better predictions, can not finally be decided upon by our results: for LDA, QDA and QDA-VS we observed a superiority of predicting with weighting, but for QDA-VS and Mec2-Q a superiority of predicting without weighting. Theoretically we would assume a superiority for the weighted strategy, but these considerations depend on the additional assumptions

- that transition probabilities are stable over time, and
- that they are only dependent on the last state, and
- that they are as important for the prediction as the observation vector.

Thus, it is not very surprising that the results do not confirm the theoretical considerations.

6 Conclusions

Summarizing, from the analysis of the results one might deduce the following general conclusions on the incorporation of background knowledge about a cyclical class structure into classification rules:

- Incorporation of cyclic structure by weighting membership values is only useful for membership values on a ratio scale.
- Prediction based on classification with exact transitions is risky, because one false prediction might entail many succeeding errors. In domains, where one phase is particularly difficult to detect it is very likely that this might cause bad classification results.

- A promising method for state prediction is forward propagation of state-probabilities as in hidden Markov models.

For the prediction of business cycle phases none of the implemented ideas has lead to a major improvement of average prediction accuracy, though. This might have been caused by two reasons:

- The minimum average prediction error that can be obtained when predicting the four phases of business cycles based on the given data and the design of comparison is about 33%. This high error rate might be caused by the known structural instability of all dependencies - those between past, current and future economic entities as well as those between economic entities at the same time. The resemblance of the average prediction errors of the best methods (NN, Mec2-Q, and Rake) - taking into account their totally different model assumptions - might suggest this explanation. Moreover, in Weihs, Röhl, and Theis [1999] it was found that a-priori restricting oneself to a certain group of only two predictors leads to best forecasts on the sixth cycle based on exact transitions. This might indicate that the other so-called 'stylized facts' of the German economy are unstable in their relationship to business cycle phases. Therefore, our next step will be to analyze the adapted methods for this group of two predictors.

- All compared multivariate classification methods are based on resampling methods to identify the best classification rule that ignore the cycle structure of the data. Therefore, these methods might be improved by using the leave-one-cycle-out idea together with our ideas for predicting the next state also for model identification. This might possibly lead to better classification results, too.

References

[Diebold and Rudebusch (1996)] Francis X. Diebold, and G. D. Rudebusch. Measuring business cycles: a modern perspective. *The Review of Economics and Statistics*, 78:67–77, 1996.

[Fletcher (1981)] Roger Fletcher. *Practical Methods of Optimization/2.* Wiley, Chichester, 1981.

[Friedmann (1996)] . Jerome H. Friedman. Another approach to polychotomous classification. Technical Report, Stanford Department of Statistics, October 1996.

[Hamilton (1989)] . James D. Hamilton. A new approach to the anylysis of non-stationarity time series and the business cycle. *Econometrica*, 57(2):357–384, 1989.

[Heilemann and Münch (1996)] Ullrich Heilemann and Heinz J. Münch. West German Business Cycles 1963-1994: A Multivariate Discriminant Analysis. *CIRET–Conference in Singapore, CIRET–Studien 50*, June 1996.

[Lucas (1987)] Robert E. Lucas. Models of business cycles. Basil Blackwell, 1987.

[Meyer and Weinberg (1975)] J. R. Meyer and D. H. Weinberg. On the classification of economic fluctuations. *Explorations in Economic Research*, 2:167–202, 1975.

[Neural Connection© 2.0 (1997)] *Neural Connection© 2.0 Users Guide.* SPSS Inc. and Recognition Systems Inc., Chicago, 1997.

[Platt *et al.*(2000)] John C. Platt, Nello Cristianini, and John Shawe-Taylor. Large Margin DAGs for Multiclass Classification. *Advances in Neural Information Processing Systems*, 12:547–553, MIT Press, 2000.

[Press *et al.* (1993)] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C.* 2nd edition, Cambridge University Press, February, 1993.

[Röhl and Weihs (1999)] Michael C. Röhl and Claus Weihs. Direct Minimization of Error Rates in Multivariate Classification. Technical Report 43/99 SFB 475, University of Dortmund, 1999.

[Schölkopf *et al* (1995)] Bernhard Schölkopf, Christopher J. C. Burges, and Vladimir N. Vapnik. Extracting Support Data for a given Task. In: Fayyad, U.M., Uthurusamy, R. (eds.) *Procceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 252–257, AAAI Press, Menlo Park, CA, 1995.

[Sondhauss and Weihs (1999)] Ursula M. Sondhauss and Claus Weihs. Dynamic Bayesian Networks for Classification of Business Cycles. Technical Report 17/99 SFB 475, University of Dortmund, 1999.

[Weihs *et al* (1999)] Claus Weihs, Michael C. Röhl and Winfried Theis. Multivariate Classification of Business Phases. Technical Report 26/99 SFB 475, University of Dortmund, 1999.