Dynamic Bayesian Networks for Classification of Business Cycles

Ursula Sondhauss and Claus Weihs*
Department of Statistics
University of Dortmund
Germany

March 23, 1999
revised June, 1999

Abstract

We use Dynamic Bayesian networks to classify business cycle phases. We compare classifiers generated by learning the Dynamic Bayesian network structure on different sets of admissible network structures. Included are sets of network structures of the Tree Augmented Naive Bayes (TAN) classifiers of Friedman, Geiger, and Goldszmidt (1997) adapted for dynamic domains. The performance of the developed classifiers on the given data was modest.

1 Introduction

Business cycle research attained new attention in the last two decades. Causes were on the one hand that cyclical changes became more important relative to growth in developed economies, and on the other hand new theories corresponding to the course of and the reasons for cyclical developments as well as improved facilities for the analysis of empirical developments and for testing complex systems of hypotheses. The new actuality of business cycle research can also be recognized by means of numerous publications (see e.g. Filardo 1994, Symposium on Developments in Business Cycle Research 1996) and also by the immediate discussion of new ideas in empirical business cycle analysis (see e.g. Council of Economic Advisers 1994).

*This work has been supported by the Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 475.

In the literature, business cycles are mainly treated as a univariate phenomenon. Often business cycles are analyzed by univariate time series models ignoring the interplay of different economic variables thus missing the possible advantages of multivariate analysis. Diagnoses and predictions based on multistage linked models and not only on unlinked individual predictions can lead to better predictive quality due to profit from aggregation because of mutual compensation of errors. Furthermore a multivariate analysis of business cycles offers the possibility of analyzing changes in form, structure, and duration of business cycles, as well as in their causes. Of special interest is the multivariate characterization of the different phases of a business cycle by more economic variables than e.g. the Gross National Product (GNP) alone and an analysis of the influence of these variables.

Probabilistic inference in Bayesian networks can be used to answer diagnostic questions of influence and characterization of that kind (Pearl, 1988). It has also been shown that Bayesian networks can be used successfully for classification (Friedman, Geiger, and Goldszmidt, FGG, 1997). Dynamic Bayesian networks (Dean and Kanazawa 1988) can model the behaviour of business cycles as a multivariate time series. So in this paper we learn structures of Dynamic Bayesian networks on restricted sets of network structures (inspired by the TAN-classifier of FGG for non-dynamic domains) using the methods in Friedman, Murphy, and Russell (FMR, 1998) for learning in dynamic domains. The data was delivered by the RWI (Rheinisch-Westfaelisches Institut fuer Wirtschaftsforschung) in Essen, Germany.

The paper is organized as follows: After this introduction, Section 2 reviews the domain and the research questions we are interested in. A short description of our data set is given here as well. In section 3 the basic notations and definitions for Bayesian networks and Dynamic Bayesian networks used in this study are introduced. Our main focus lies on Section 4 that comprises the specific task of learning Bayesian network classifiers and how it was transformed in this study. Section 5 contains a subsumption of results and their ratings by the domain experts. Finally in Section 6 we sketch our future research interests.

2 Domain and Data

Researchers from the RWI developed an econometric model for predicting important aggregates of the national accounts and also for simulating important economic policies (Rau et al. 1977).

Motivated by the increasing interest in the cyclical dimension of short term economic development, the RWI started looking at comparable cyclical phases in the past, when analyzing the current stage and dynamics of the German economy.

Despite the conviction of the domain experts that business cycles have to be understood as a multivariate phenomenon, they are mainly treated univariately.

A first study by Heilemann and Muench (1996) obtained promising results with respect to the advantages of a multivariate approach.

Within the Collaborative Research Centre (SFB475) at the University of Dortmund a cooperation started of the researchers of the statistical department and the researchers of the RWI. The long term goal is to characterize business cycles by the state of stylizing economic variables. The aim is to find a multivariate and more comprehensive definition of business cycles as hitherto that can be widely accepted. The weights of individual variables or groups of variables for the forecasts of business cycles shall be identified as well as their stability over time and in international comparison.

The potentials of Bayesian networks seemed promising here. Among other desirable properties

- their representation of uncertainty in the domain matches the representation in the econometric model of the domain experts and
- they provide the researchers with a joint probability for the economic variables along with algorithms for probabilistic inference that can be used to answer diverse questions of characterization and diagnosis.

In this study Bayesian networks are used to classify phases of business cycles. In future work these results will be compared with the results of other methods applied to the problem.

2.1 Variables

The selection of predictor variables was carried out by our domain experts by means of both theoretical deliberations and the "stylized facts" (Lucas 1983) of the West German economy. A list of these variables is given in Table 1.

In the following sections a vector of these variables or the set of them will be called stylized facts or $Styfacts$.

For the definition of the class variable $Phase$ a four phase scheme is used: upswing (1), upper turning point (2), downswing (3), and lower turning point (4).

2.2 Training Data and Test Data

Our data set consists of quarterly measurements obtained over the period 1955/4 to 1994/4 without missing values. The data is not seasonally adjusted, but this is balanced by the frequent use of growth rates with regard to the corresponding quarter of the previous year (Heilemann and Barabas 1996, p.404). These 157 observations were classified into business cycle phases by the domain experts based on economic and heuristic considerations.

Table 1: Predictor variables

Abbrevation	Variable
GNP	GNP, real (y)[a]
Consumption	Private consumption, real (y)
Gov Deficit	Government deficit, percent of GNP
Earners	Wage and salary earners (y)
Net exports	Net exports as percent of GNP
Money M1	Money supply M1 (y)
Equipment	Investment in equipment, real (y)
Construction	Investment in construction, real (y)
Unit Labour Cost	Unit labour cost (y)
GNP PD	GNP price deflator (y)
Consumer PI	Consumer price index (y)
Short-I-Rate	Short term interest rate, nominal
Long-I-Rate	Long term interest rate, real

[a]y=yearly growth rate

We constructed the classification rule without the last complete cycle and classified the remaining cases afterwards. This reduced the learning sample to 112 quarters that include 30 examples for phase 1, 18 for phase 2 , 38 for phase 3, and 26 for phase 4. The test data begins with examples for an upswing in 1983/4 and ends with an uncompleted upswing in 1994/4. There are 29 cases of upswing, the upper turning point lasted 6 quarters, the downswing 9 quarters and the lower turning point 1 quarter only.

Splitting the data in this specific manner into test and training set was motivated by the standard prognostic/forecast method for cyclical domains. It is going to be the general setting for the evaluation of all methods that are going to be compared in our future research.

Note, though, that there are antipodal interpretations of the results from such a splitting by leaving-one-cycle out: On the one side it can be used to analyze the specific properties of this business cycle compared to the others. On the other side the predictive quality of the classifiers is compared. The selected last cycle is an extra challenge for any classifiers: it includes the German reunification in 1989 with its changes of the political and economical situation of Germany. Note that nevertheless all data is based on the West German economy only.

2.3 Discretizing

Being growth rates, stylized facts are naturally represented by continuous variables. However, in the analysis of business cycles discretization is of special interest. It strongly influences the meaning of characterizations of different phases in rules like "when there is a medium rate of consumption and a low short term interest rate then the current phase of the business cycle is with high probability an upswing". Apart from theoretic exceptions[1], classifiers discretize the predictor variables by separating their multidimensional space in subsets corresponding to the different classes. Thus discretization is either a part of the learning procedure of the classifier or it can be done previously.

We decided to prediscretize the data and not to use the advantages of methods where the discretization is learned with the structure of the Bayesian network. Such procedures could have been the multivariate method of Monti and Cooper (1998) or the dual representation of Friedman, Goldszmidt, and Lee (1998). Particularly the latter would have been attractive for our purposes, because it considers the classification task. But with only 112 training cases it seemed unreasonable to use it.

A univariate discretization was performed with the FUSINTER method of Zighed, Rakotomalala, and Rabaseda (1996). In their paper the authors showed that the results of their method are very close to those of the standard method of Fayyad and Irani (1993). It is a bottom-up strategy that uses an information-gain criterion for the decision on a new split. To avoid over-splitting, the criterion penalizes low numbers of examples in the new intervals. As these numbers are used to estimate the probabilities of belonging to a class related to the defined intervals, this is considered to be a desirable property when the discretized data is going to be used for learning a Bayesian network. For our calculations, we used the free software SIPINA_W© v2.0 (Zighed and Rakotomamala 1997).

3 Notation and Basic Definitions

3.1 Bayesian Networks

Let $\vec{X} = (X_1, ..., X_n)$ be a vector of real random variables from a domain $\mathbf{U} = \{X_1, ..., X_n\}$. A very general probability space for $\vec{X}$ is denoted by $(\mathbb{R}^n, \mathcal{B}^n, P_B)$, where $\mathbb{R}^n$ is the set of n-dimensional vectors of real numbers, $\mathcal{B}^n$ is the Borel set of $\mathbb{R}^n$, and P_B is a probability function with domain $\mathcal{B}^n$.

A pair $B = (G, P_B)$ is a Bayesian network of this model, if

1. $G := (\mathbf{U}, E)$ is a directed acyclic graph (DAG) over $\mathbf{U}$ and the set E of directed edges and

[1] e.g. the classification of the reell numbers into the set of real numbers without the rational numbers $\mathbb{R}\backslash\mathbb{Q}$ and the rational numbers $\mathbb{Q}$

2. the structure of the graph G reflects the independence assumptions of the distribution P_B. That is for all $i, i = 1, ..., n$:
   (a) X_i is independent of all its non-descendants given its parents Π_i in the Graph G.
   (b) There is no real subset $\Pi_i' \in \Pi_i$ for which (a) is true.

By the so-called directed Markov property 2.(b) the joint distribution P_B can be factorized as follows:

$$P_B(X_1, ..., X_n) = \prod_{i=1}^{n} P_B(X_i|\Pi_i). \tag{1}$$

Another independence relation on $\mathbf{U}$ follows directly from the definition of a Bayesian network:

- For all $X_i \in \mathbf{U} : X_i$ is - conditioned on a set of variables called the Markov blanket of X_i - independent of all other variables in $\mathbf{U}$.
- The Markov blanket $\mathbf{\Delta}(X_i)$ consists of the parents of X_i, its children, and its children's parents.

For classification tasks the domain is $\mathbf{U} = \{C, A_1, ..., A_n\}$, where C is the class variable and $\mathbf{A} = \{A_1, ..., A_n\}$ is the set of attributes that are used as predictor variables. For the task at hand $C = Phase$ and $\mathbf{A} = Styfacts$. Of special interest is the conditional distribution of C given $\mathbf{A}$, $P_B(C|A_1, ..., A_n) = P_B(C|A_{i_1}, ..., A_{i_k})$ with $\{A_{i_1}, ..., A_{i_k}\} = \mathbf{\Delta}(C) \subset \mathbf{U}$.

In the following, we restrict our interest on domains, where $\vec{X}$ is a n-dimensional discrete vector. Without loss of generality, we can assume that $\vec{X} \in I\!N^n$, where $I\!N^n$ is the set of n-dimensional vectors of natural numbers. Let $M_i = \{1, ..., r_i\}$ represent the finite set of possible values of X_i, and $M_{\Pi_i} = \{1, ..., q_i\}$ represent the finite set of possible combinations of values that the parent set Π_i of X_i can take (given a DAG G).

Then the corresponding probability space of a Bayesian network (G, P_B) on $\mathbf{U}$ is given by

$$(\mathbf{M}, \mathcal{A}(\mathbf{M}), P_B) \tag{2}$$

with $\mathbf{M} \subseteq \mathbf{M}_1 \times \mathbf{M}_2 \times ... \times \mathbf{M}_n \subseteq I\!N^n$, the product of the sets of possible values for $X_i, i = 1, ..., n$, $\mathcal{A}(\mathbf{M})$ is the collection of all possible subsets of $\mathbf{M}$, and P_B is a probability function with domain $\mathcal{A}(\mathbf{M})$. Actually, because of the factorization in (1), P_B represents a collection of products of local distributions:

For each instantiation of $\Pi_i = j \in \{1, ..., q_i\}$ the distribution $P_B(X_i|\Pi_i = j)$ is a multinomial distribution with parameter vector $\vec{\theta_{ij}} = \{\theta_{ij1}, ..., \theta_{ijr_i}\}$. Let $\theta_{\mathbf{i}} = \vec{\theta_{i1}}, ..., \vec{\theta_{iq_i}}$ denote the parameters for the set of multinomial distributions for node $X_i, i = 1, ..., n$ and let $\mathbf{\Theta}_B = \{\theta_{\mathbf{i}}, i = 1, ..., n\}$ denote the parameters that are needed to encode the probability P_B of the Bayesian network (G, P_B).

3.2 Learning Bayesian Networks

Expressed in the language of statistical decision theory, learning a Bayesian network $B \in \mathbf{B}$ from observed data $\mathbf{d} = \{\vec{x}_1, ..., \vec{x}_N\} \in \mathbf{M}^N$ means to find the "best" decision function $f \in \mathcal{F} : \mathbf{M}^N \to \mathbf{B}$ in an admissible set of decision functions $\mathcal{F}$. A decision function f assigns to any possible $\mathbf{d} \in \mathbf{M}^N$ a Bayesian network B out of a (possibly restricted) set $\mathbf{B} = \{(G, P_B), G \,\text{is DAG over}\, \mathbf{U}\}$.

When learning a Bayesian network from discrete data $\mathbf{d}$, it is commonly assumed that $\mathbf{d}$ is the realization of a multinomial sample from a population with distribution $P_B \in \mathbf{P_B}$, where $\mathbf{P_B}$ is the set of all distributions on $\mathcal{A}(\mathbf{M}^N)$ that can be described by a Bayesian network on $\mathbf{U}$. That is, $\vec{x}_1, ..., \vec{x}_N$ are assumed to be realizations of random variables $\vec{X}_1, ..., \vec{X}_N$ that are independent and identically distributed (i.i.d) according to some $P_B \in \mathbf{P_B}$.

There are a lot of ways to define what can be considered to be the "best" decision function. In learning Bayesian networks the common approaches are based on scoring functions $s : \mathbf{M}^N \times \mathbf{B} \to I\!R$ that give an evaluation on how well a given network $B \in \mathbf{B}$ matches the data $\mathbf{d} \in \mathbf{M}^N$. Given a scoring function, the best decision function is the one that assigns to each $d \in \mathbf{M}^N$ the network $B \in \mathbf{B}$ that maximizes this scoring function: $f_{opt}(\mathbf{d}) := \arg\max_{B \in \mathbf{B}}(s(B, \mathbf{d}))$.

An ad-hoc scoring function is based on the maximum-likelihood principle: the log-likelihood $s(B, \mathbf{d}) := LL(B|\mathbf{d}) = logP_B(\mathbf{d})$. But the number of parameters (here and in the following used to measure the complexity) that are necessary to encode $P_B \in \mathbf{P_B}$ can be very large depending on G. Thus an application of the maximum-likelihood principle might lead to an overfitting and thus to Bayesian networks that match the given data well, but have low predictive quality for new cases. Therefore, a penalty in the scoring function for the complexity of $B \in \mathbf{B}$ is needed.

The scoring functions that are most frequently used to learn Bayesian networks fulfill this condition: the MDL scoring function (Lam and Bacchus 1994) and the BDe scoring function (Heckerman, Geiger, and Chickering 1995) combine the likelihood with some penalty relating to the complexity of the model. The MDL scoring metric is based on the principle of Minimum description length. The BDe scoring function evolves from the search for a network with largest posterior probability given priors over network structures G and parameters $\mathbf{\Theta_B}$.

The derivations of these scoring functions as well as the log-likelihood rely on the assumption that $\vec{x}_1, ..., \vec{x}_N$ are realizations of random variables $\vec{X}_1, ..., \vec{X}_N$ that are i.i.d. according to P_B. The following factorization is used:

$$P(\vec{X}_1, ..., \vec{X}_N) = \prod_{t=1}^{N} P_B(\vec{X}_t) = \prod_{t=1}^{N} \prod_{i=1}^{n} P_B(X_{it}|\Pi_{it}). \tag{3}$$

But of course the required independence assumption for this factorization does not match the assumption that $\vec{x}_1, ..., \vec{x}_N$ is the realization of a multivariate time-series $\vec{X}_t$ at the discrete time-points $t = 1, ..., N$. But as this is the appropriate assumption for the data at hand: $\vec{x}_t := (phase, styfacts)'_t, t = 1, ..., 112$ we need to use dynamic Bayesian networks.

3.3 Learning Dynamic Bayesian Networks

In their paper FMR show how the independence assumption of $\vec{X}_1, ..., \vec{X}_N$ can be replaced by the assumption of an underlying stationary Markovian process for $\vec{X}_t, t = 1, 2, 3, ..$ in the derivation of scoring functions.

The assumption of a stationary Markovian process allows for the following factorization of the joint distribution for a finite sequence $\vec{X}_1, ..., \vec{X}_N$ of the process:

$$P(\vec{X}_1, ..., \vec{X}_N) = P(\vec{X}_1) \prod_{t=2}^{N} P(\vec{X}_t | \vec{X}_{t-1}). \tag{4}$$

A Dynamic Bayesian network of a discrete stationary Markovian process $\vec{X}_t \in \mathbf{M}, t = 1, 2, 3, ..$ is defined by:

- a prior network B_1 with a corresponding probability space $(\mathbf{M}, \mathcal{A}(\mathbf{M}), P_1)$ for initial states $\vec{X}_1$, where P_1 is the initial distribution of the Markovian process and
- a transition network B with a corresponding probability space $(M^2, \mathcal{A}(\mathbf{M}^2), P_{\vec{X}|\vec{X}_{-1}})$ for a random vector of transitions $(\vec{X}_{-1}, \vec{X})'$, where $P_{\vec{X}|\vec{X}_{-1}}$ is the transition probability of the Markovian process.

The set of transition networks is denoted by $\mathbf{B}_{\rightarrow}$. Note that in a transition network variables of the time-slice "-1" have no parents: $\Pi_{i,-1} = \emptyset, i = 1, ..., n$.

The factorization in equation (4) can be used to derive analogs of the MDL, BDe, and log-likelihood scoring functions for learning Dynamic Bayesian networks. FMR show that the task of learning a Dynamic Bayesian network can be divided in two separate steps. That is, we learn a Dynamic Bayesian network by learning two (standard) Bayesian networks:

1. The prior network B_1 has to be learned with a sample $\mathbf{d} = \{x_{11}, ..., x_{1M}\}$. This learning needs multiple observations of time series sequences $x_t, t = 1, ..., N_m$ with length $N_m, m = 1, ..., M$, respectively.
2. In another step, a transition network B on $\mathbf{U} = \{X_{1,-1}, ..., X_{n,-1}, X_1, ..., X_n\}$ has to be learned under the restriction that $B \in \mathbf{B}_{\rightarrow}$. The learning set is built from an observation of the time series

$$\mathbf{d} = \{\vec{x}_{11}, ..., \vec{x}_{1N_1}, ..., \vec{x}_{bfM1}, ..., \vec{x}_{MN_M}\}$$

by concatenating all pairs of consecutive observations $(\vec{x}_{m(t-1)}, \vec{x}_{mt})'$, $t = 2, ..., N_m$,$m = 1, ..., M$. That is, we generate a new datavector $\tilde{\mathbf{d}}$ by

$$\tilde{\mathbf{d}} = \{(\vec{x}_{-1}, \vec{x})'_l, l = 1, ..., L, L = \textstyle\sum_{m=1}^{M} N_m - M\}.$$

3.4 Classification of Business Cycles

Classification of the phases of business cycles (apart from time point t=1) will be based on the estimated conditional distribution

$$\hat{P}_B(Phase|Phase_{-1}, Styfacts_{-1}, Styfacts)$$

of the learned transition network $B \in \mathbf{B}_{\rightarrow}$. Time-point $t = 1$ is of no special interest, thus it suffices to learn a transition network that is useful for the classification of $Phase$. The training data $\mathbf{d} = \{(phase, styfacts)'_t, t = 1, ..., 112\}$ is therefore converted into

$$\tilde{\mathbf{d}} = \{(phase_{-1}, styfacts_{-1}, phase, styfacts)'_t, t = 2, ..., 112\}.$$

Assuming for the economic variables an underlying stationary Markovian process is closer to the demand of the domain than the independence assumption in (3). It should be kept in mind though that particularly the assumption of stationarity is also questionable.

4 Learning Bayesian Network Classifiers

In this Section, we start by giving an outline of the approach (4.1). We motivate and describe the details in the subsections that follow: In 4.2 we briefly introduce the TAN classifiers of FGG and in 4.3 we present the restricted sets of network structures based on those of TAN classifiers adapted for dynamic domains. Subsequently, we discuss the application of the outlined method for learning classifiers for the phases of German business cycles in subsection 4.4. Finally in subsection 4.5 we describe the different search strategies we pursued to learn classifiers on the given training set.

4.1 The Basic Approach

To learn the structure of transition networks we used the well-known K2-Algorithm of Cooper and Herskovits (1992) on different sets $\mathbf{B}$ of admissible network structures. As in its original form the fit of a network B to the data was evaluated by the log-likelihood $LL(B|\mathbf{d})$ as scoring function. For the calculations, we used the free software INES (Borgelt, Kruse and Lindner 1995).

To estimate the parameters $\mathbf{\Theta}_B$ once the structure is learned, we did not use the maximum-likelihood estimator, though this would maximize our scoring function. The maximum-likelihood estimators $\tilde{\theta}_{ijl}$ for $l = 1, ..., r_i, j = 1, ..., q_i, i = 1, ..., n$, are given by

$$\tilde{\theta}_{ijl} = \begin{cases} 0, & N_{ij} = 0 \\ \frac{N_{ijl}}{N_{ij}} & N_{ij} > 0 \end{cases} \tag{5}$$

where N_{ijl} is the number of cases in the sample with $X_{it} = l$ and $\Pi_{it} = j, t = 1, ..., N$ and $N_{ij} := \sum_{l=1}^{r_i} N_{ijl}, j = 1, ..., q_i, i = 1, ..., n$.

Instead, following FGG, we used a smoothed variant

$$\hat{\theta}_{ijl} = \frac{N_{ijl} + 5\frac{N_{il}}{N}}{N_{ij} + 5} \tag{6}$$

with $N_{il} := \sum_{j=1}^{q_i} N_{ijl}, l = 1, ..., r_i, i = 1, ..., n$.

Only when $Phase_{-1}$ is parent of $Phase$ no smoothing was performed: zero estimates $\hat{\theta}(Phase|Phase_{-1}) = 0$ capture a true feature of the domain. The phases of a business cycles have to follow the pattern $...4 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow ....$ Thus, if $\theta(Phase|Phase_{-1}) \in \mathbf{\Theta}_B$, this parameter was estimated according to (5).

The smoothing can be interpreted as Dirichlet priors on $\{\theta_{ij1}, ..., \theta_{ijr_i-1}\}$ with parameters $p_{ijl} = \sum_{j=1}^{q_i} \frac{N_{ij}}{N}$ for all $l = 1, ..., r_i - 1$ and equivalent sample size 5 for $j \in 1, .., q_i$ and $i = 1, ..., n$ (see Heckerman, Geiger, and Chickering 1995). The equivalent sample size N'_e is a measure for the uncertainty in the prior: It is the same as the posterior Dirichlet distribution that results from an uninformative prior over [0,1] and N'_e cases with frequencies p_{ijl}, $l = 1, ..., r_i - 1$, $j \in 1, .., q_i$ and $i = 1, ..., n$.

Because of our small training set, smoothing was crucial: Not only we would otherwise encounter unreliable estimates but also many of them would be zero. Any combination of values of the predictors not included in the training set could not be classified by an unsmoothed classifier.

4.2 TAN Classifiers

The log-likelihood is in general unwarranted as a scoring function to guide the learning process of Bayesian network structures. It favors complete structures because of having no feature to avoid overfitting. But the two commonly used scoring functions MDL and BDe that include penalty terms for the complexity of the Bayesian network are not appropriate for the classification task (FGG). These scoring functions measure the likelihood that the data was generated from a Bayesian network B considering the joint distribution $P_B(C, A_1, ..., A_m)$ of that Bayesian network. They penalize the complexity of the whole structure. The target fit in a classification task though should be that of the data with the conditional distribution of $P_B(C|A_1, ..., A_m)$. The complexity of the classifier is evaluated more appropriately by the number of parameters of this conditional distribution.

To our knowledge, a computationally feasible scoring function that learns an optimal Bayesian network classifier in the set of all possible network structures over $\mathbf{U}$ has not been developed yet. The idea of the TAN method of FGG is to restrict the search on networks with a certain structure: The edges of a naive

Bayes classifier are prescribed, that is the class variable is forcibly a parent of every attribute. Additionally there are possible edges between the attributes but maximally one more parent for each. Within this type of Bayesian networks - denoted in the following by $\mathbf{B^T}$ - they show that the optimal classifier with respect to the log-likelihood as scoring function can be efficiently found.

4.3 (S)TAN classifiers for Dynamic Domains

The TAN method for transition networks on a domain of variables

$$\mathbf{U} = \{C_{-1}, A_{1,-1}, ..., A_{n,-1}, C, A_1, ..., A_n\}$$

could be extended in the following way:

On the subset $\{C, A_1, ..., A_n\}$ of variables in the actual time-slice the original TAN Models are learned. Any information from the preceding time-slice passes through C. That is, only C can have (maximally two) parents from $\{C_{-1}, A_{1,-1}, ..., A_{n,-1}\}$ (see Figure 1). A set of Bayesian networks, satisfying these restrictions will be represented by $\mathbf{B}^T_{2,\to}$.

Transition (S)TAN:

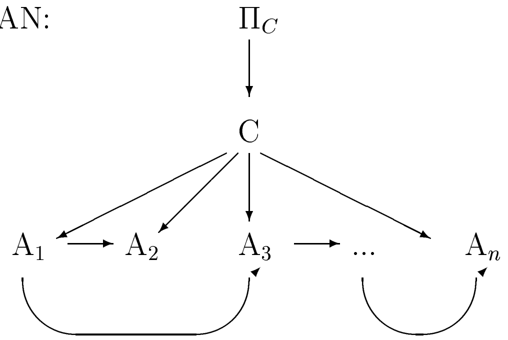


Figure 1: Example for the structure of a transition (S)TAN classifier.

Particularly when $\Pi_C = C_{-1}$ these "transition TAN classifiers" have a structure that corresponds to the structure of the transition network of a Hidden Markov model (Rabiner and Juang 1993), when this is represented as a Dynamic Bayesian network with observation variable $O := (A_1, ..., A_n)'$. This might be suitable for many domains. In our domain this gives phases of business cycles a convenient interpretation as states of some underlying process that influences the behaviour of economic variables.

As we used the K2-algorithm for learning network structures and not the optimal one for the TAN classifiers, we name those classifiers learned on $\mathbf{B}^T_{2,\to}$ "sub-optimal" TAN classifiers, abbreviated as "STAN".

We did not consider another TAN-inspired classifier for dynamic domains that includes no search for the structure. The corresponding networks would be given some kind of "Rake structure" (see Figure 2), where each attribute $A_i, i = 1, ..., n$

has $\Pi_i = \{C, A_{i,-1}\}$ as parents. It might be interesting to investigate their behaviour though, as this structure ensures that all variables of the transition network are in the Markov blanket of C, and it might be a suitable way to model the (main) dependencies on $\{A_{1,-1}, ..., A_{n,-1}, A_1, ..., A_n\}$.

RAKE:

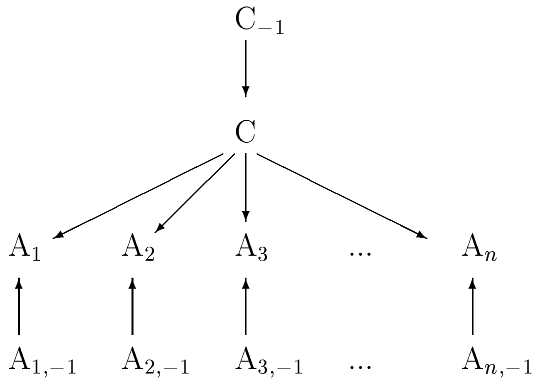


Figure 2: Structure of a RAKE classifier.

4.4 STAN Models for the Specific Task

The restricted set $\mathbf{B}^T_{2,\rightarrow}$ of transition STAN classifiers can result in a raise of the complexity of the learned classifier, when it is compared with the complexity of networks learned with the MDL- or BDe-score on the set $\mathbf{B}$ of all networks on $\mathbf{U}$. The reason is that for all $B \in \mathbf{B}^T_{2,\rightarrow}$ the set $\{A_1, ..., A_n\} \subseteq \mathbf{\Delta}_B(C)$, where as the Markov blanket $\mathbf{\Delta}_{\tilde{B}}(C)$ of a network $\tilde{B}$ that got learned with the MDL or BDe score might be smaller. This reduces the number of parameters needed to encode $P_{\tilde{B}}(C|\mathbf{\Delta}_{\tilde{B}}(C))$.

By forcing the Bayesian network to include all arcs of the naive Bayesian classifier we make sure that all stylized facts will be used for classifying the phase of the business cycle. A possible drawback is that no implicit feature selection takes place. This is not so relevant in our analysis because heavy feature selection has already been carried out by our experts. An analysis on the importance of certain stylized facts for the classification can be performed with the estimated conditional probability of the Bayesian network.

But as the training data is very sparse, the forced complexity of $P_B(C|\mathbf{\Delta}_B(C))$ for $B \in \mathbf{B}^T_{2,\rightarrow}$ might lead to an overfitting. Thus we decided to compare the STAN classifiers with classifiers based on Bayesian networks that were learned on the set $\mathbf{B}_2$ of networks, where each node can have a maximum number of two parents. The log-likelihood was used as scoring function. Note that $\mathbf{B}^T_{2,\rightarrow} \subset \mathbf{B}_2$.

4.5 The Search Strategies

Three basic strategies were pursued: In the first one the classification rule is built on all variables of the preceding time slice and all variables of the actual one. In the second strategy $Phase_{-1}$ is not included, because when we want to classify the phase of an actual business cycle, $Phase_{-1}$ is not necessarily known. In what phase the business cycle was in a certain quarter is classified best by the experts when the whole cycle is completed. The last strategy does not use any information from the preceding time slice. That is we ignore the time dependence in the domain to learn a Bayesian network as if we had an identically and independently distributed sample from some joint distribution $P_B(Phase, Styfacts)$.

The main classification approaches and their abbreviations are:

1. Strategy:
   $\mathbf{U} = \{Phase_{-1}, Styfacts_{-1}, Phase, Styfacts\}$

   – TRANS: $B \in \mathbf{B}_{2,\rightarrow}$, the set transition networks with a maximum number of two parents, and $s = LL(B|\mathbf{d})$. Smoothed parameter estimates apart from $\hat{\theta}(Phase|Phase_{-1})$.

   – STAN1: $B \in \mathbf{B}^T_{2,\rightarrow}$, $s = LL(B|\mathbf{d})$, $\Pi(Phase) \subset \{Phase_{-1}, Styfacts_{-1}\}$. Smoothed parameter estimates apart from $\hat{\theta}(Phase|Phase_{-1})$.

2. Strategy:
   $\mathbf{U} = \{Styfacts_{-1}, Phase, Styfacts\}$

   – PRED: $B \in \mathbf{B}_{2,\rightarrow}$, $s = LL(B|\mathbf{d})$. Smoothed parameter estimates.

   – STAN2: $B \in \mathbf{B}^T_{2,\rightarrow}$, $s = LL(B|\mathbf{d})$, $\Pi(Phase) \subset \{Styfacts_{-1}\}$. Smoothed parameter estimates.

3. Strategy:
   $\mathbf{U} = \{Phase, Styfacts\}$

   – BN: $B \in \mathbf{B}_2$, $s = LL(B|\mathbf{d})$. Smoothed parameter estimates.

   – STAN3: $B \in \mathbf{B}^T$, $s = LL(B|\mathbf{d})$. Smoothed parameter estimates.

   – NBAY: The naive Bayes classifier. No scoring function is needed. Smoothed parameter estimates.

4.6 Finding an ordering

The K2-algorithm needs an ordering on the variables. Searching in the set of transition networks $\mathbf{B}_{\rightarrow}$ induces a partial ordering where $\{Phase_{-1},\ Styfacts_{-1}\}$ comes before $\{Phase, Styfacts\}$. Also, no ordering has to be found on $\{Phase_{-1},\ Styfacts_{-1}\}$. Thus an ordering on $\{Phase, Styfacts\}$ is needed that is useful for the classification task and helps to find a good representation of the correlation between the stylized facts.

In all STAN classifiers *Phase* precedes all attributes in the ordering. For the learning of networks within $\mathbf{B}_2$ this is a reasonable position, as it maximizes the number of possible directed links of the class variable *Phase* with its predictors.

To find an ordering within the group of stylized facts we performed an analysis on their rank correlations. The more significant correlations were found between an attribute and the others the earlier it was placed in the ordering. Analogous to the argumentation to set *Phase* on the first place this procedure ensures that an attribute that correlates with many of the other attributes can have many directed links with the other variables.

4.7 Interpretation as Bayesian Learning

The domain experts did not give priors on the network structures nor on certain parameters. When our procedure nevertheless is interpreted as a Bayesian approach to learn, the following assumptions (see Heckerman, Geiger, and Chickering 1995) were made: parameter independence and parameter modularity, an uninformative prior over different sets of admissible network structures, and an informative prior over the parameters for the smoothed estimation.

The assumption of a multinomial sample on $\mathbf{U} = \{Phase,\ Styfacts\}$ was replaced by the assumption of an stationary Markovian process underlying $\{(Phase, Styfacts)'_t, t = 1, ..., 112\}$.

5 Results

In Figures 3, 4, and 5 one can see the learned structures from the different classification approaches. Only the subgraph of G on the Markov blanket $\mathbf{\Delta}(Phase)$ is presented there. In the complete graph of TRANS and PRED all attributes in the present time-slice selected its predecessor as parent. This indicates that the "Rake classifiers" (Figure 2) might be interesting.

In the STAN1 classifier and the TRANS classifier one can see that once $Phase_{-1}$ is in the set of possible parents of *Phase* the algorithm will not include any of the other variables additionally.

Comparing STAN1 and STAN2 the results show that the transition parents of *Phase* that replace $Phase_{-1}$ are *Short-I-Rate*$_{-1}$ and $Earners_{-1}$. This matched the prior assessment of the importance of these two variables by the domain experts.

Phase proved to be quite a good summarizing variable within a time-slice: In the BN classifier only the four variables *Construction*, *Long-I-Rate*, *Gov-Deficit*, *Net exports* do not include *Phase* in their set of two parents and are not members of the Markov blanket of *Phase*. These variables are commonly considered to be of minor influence on the classification of *Phase*.

Table 2: Error Rates

Model	Training set	Test set
Strategy 1 with $Phase_{-1}$		
SIMPLE	19.8 %	11.1 %
STAN1	4.5 %	35.6 %
TRANS	18.9 %	44.4 %
Strategy 2 with $Styfacts_{-1}$		
STAN2	7.2%	68.9 %
PRED	30.9 %	66.7 %
Strategy 3		
STAN3	9.9 %	71.1 %
BN	9.9 %	60 %
NBAY	38.7 %	60 %

Only the classifiers of the same strategy should be compared with respect to their error rates as they can be used on different levels of information. For strategy 1 we added in Table 2 the classifier "SIMPLE" that says the business cycle is in the same phase now as in the last quarter. This is considered to be a proper benchmark for any classifier using $Phase_{-1}$. One can see that STAN1 outperforms the SIMPLE classifiers clearly on the training set whereas the TRANS classifier is only slightly better. But on the test set the misclassification rate is more than three times higher for both of them.

When only the misclassification rate on the test set is considered, the behaviour of the classifiers of strategy 2 or 3 is disastrous. On the training set the STAN classifiers are the best but the antipode is true on the test set. This could be a matter of overfitting, but a deeper look at the progression of the predictions over time (Figures 6, 7, and 8) gives license for another interpretation: the STAN classifiers of Strategy 2 and 3, and the BN classifier show a reasonable behaviour by signalizing two cycles in the quarters from 1983/4 to 1994/4. The domain experts were not really taken aback from that. There are discussions among economists whether the quarters 1984/1 to 1986/1 are to be considered as the valley in the plateau of an upswing or whether two cycles should be classified (see e.g. Tichy 1994).

In the quarters around the German reunification 1989 all our classifiers ran into problems. The following downswing from 1992/1 to 1994/1 is not captured by the STAN classifiers and the BN classifier. They all indicate an earlier change into phase 4 (the lower turning point) than the experts. Only the predictions of TRANS classifier matched the experts classification there. The PRED and the

NBAY classifier show a comparable behaviour on the test set by omitting phase 2 completely and overemphasizing phase 3.

Summarizing, the STAN classifiers and the BN classifier do not mirror the classification of the experts on the test set. But differently from the other classifiers, their classification corresponds to a reasonable cyclical behaviour apart from the quarters around the german reunification.

We performed a first analysis of the influence of attributes on the classification. Pie charts of the estimated distribution $\hat{P}_B(PredictedClass|A_i = j), j = 1, ..., q_i, i = 1, ..., n$ were a helpful tool to discuss the importance of different economic variables. A difficulty arose with the interpretation of the division of the values of the variables low, (medium, and) high by the discretization: There is still a trend in the stylized facts over time, so that this division is questionable. The trend and the resulting problem for the discretization can also be one of the reasons for the bad performance on the test set.

But nevertheless, our results made domain experts very much interested in future applications of Bayesian networks for the analysis of economic data.

6 Further research

The findings and the facilities for interpretation of this analysis will be compared with those of other methods (Linear and Quadratic Discriminant analysis, CART, Support Vector Machines, univariate time-series analysis, Neural Networks) on the same data.

As the discretization is important, improvements will be hoped for by performing it on de-trended data. The sparsity of the data limited the methods for analysis. To solve this problem, a new data set will be installed with monthly data of variables that are comparable to the set at hand.

Finally, we envisage an analysis with Hidden Markov models to verify the four phase scheme of German business cycles and an analysis with "Rake classifiers" (Figure 2).

References

Borgelt C., Kruse, R, and Lindner, G. (1998). Lernen probabilistischer und possibilistischer Netze aus Daten: Theorie und Anwendung. *Kuenstliche Intelligenz, Themenheft Data Mining 1*, 11-17. Gesellschaft fuer Informatik.

Cooper, G. F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning 9*, 309-347.

Council of Economic Advisors (1994). The Annual Report of the Council of Economic Advisors. In *Economic Report of the President.*

Dean, T. and Kanazawa, K. (1988). Probabilistic temporal reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence.* St.

Paul, Minnesota. American Association for Artificial Intelligence.

Fayyad, U. M. and Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceeding of the 13th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann.

Filardo, A. J. (1994). Business-cycle phases and their transitional dynamics. *Journal of Business & Economic Statistics 12*, 299-308.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning 29*, 131-163.

Friedman, N., Goldszmidt, M., and Lee, T. (1998). Bayesian Network Classification with Continous Attributes: Getting the Best of Both Discretization and Parametric Fitting. In *Proceedings of the 15th International Conference on Machine Learning.* Morgan Kaufmann.

Friedman, N., Murphy, K., and Russell, S. (1998). Learning the structure of dynamic probabilistic networks. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence.* Morgan Kaufmann.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning 20*, 197-243.

Heilemann, U. and Barabas, G. (1996). Saisonbereinigung und makro oekonomische Modelle: Befunde und Erfahrungen mit dem RWI-Konjunkturmodell. *ifo-Studien 41*, Berlin.

Heilemann, U. and Muench, H. J. (1996). West German Business Cycles 1963-1994: A Multivariate Discriminant Analysis, *CIRET–Conference in Singapore, CIRET–Studien 50.*

Lam, W. and Bacchus, F. (1994). Learning Bayesian belief networks. An approach based on the MDL principle. *Computational Intelligence 10*, 269-293.

Lucas, R. E. (1983). Understanding business cycles. In *Studies in Business-Cycle Theory.* Cambridge, Mass. and London.

Monti, S. and Cooper, G. F. (1998). A multivariate discretization method for learning Bayesian networks from mixed data. In *Proceedings of 14th Conference of Uncertainty in Artificial Intelligence.* Morgan Kaufmann.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems.* San Francisco CA, Morgan Kaufmann.

Rau, R., Heilemann, U., Korthaus, E., and Muench, H. J. (1977). Das RWI-Konjunkturmodell - Hypothesen, Struktur und Ergebnisse. *RWI-Papiere 6.* Essen.

Rabiner, L. R. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition.* Prentice Hall.

Symposium on Developments in Business Cycle Research (1996). *Review of Economics and Statistics 78*, 1-110.

Tichy, G. (1994). *Konjunktur - Stilisierte Fakten, Theorie, Prognose.* Springer Lehrbuch, Berlin

Zighed, D. A., Rakotomalala, R. (1997). SIPINA_W© v2.0 for Windows. A new tool for Knowledge Discovery in Databases.
`http://eric.univ-lyon2.fr/~ ricco/sipina.html.`

Zighed, D. A., Rakotomalala, R., and Rabaseda, S. (1996). A discretization method of continuous attributes in induction graphs. In *Proceedings of the 13th European Meeting on Cybernetics and System Research*. ASCS, Vienna.

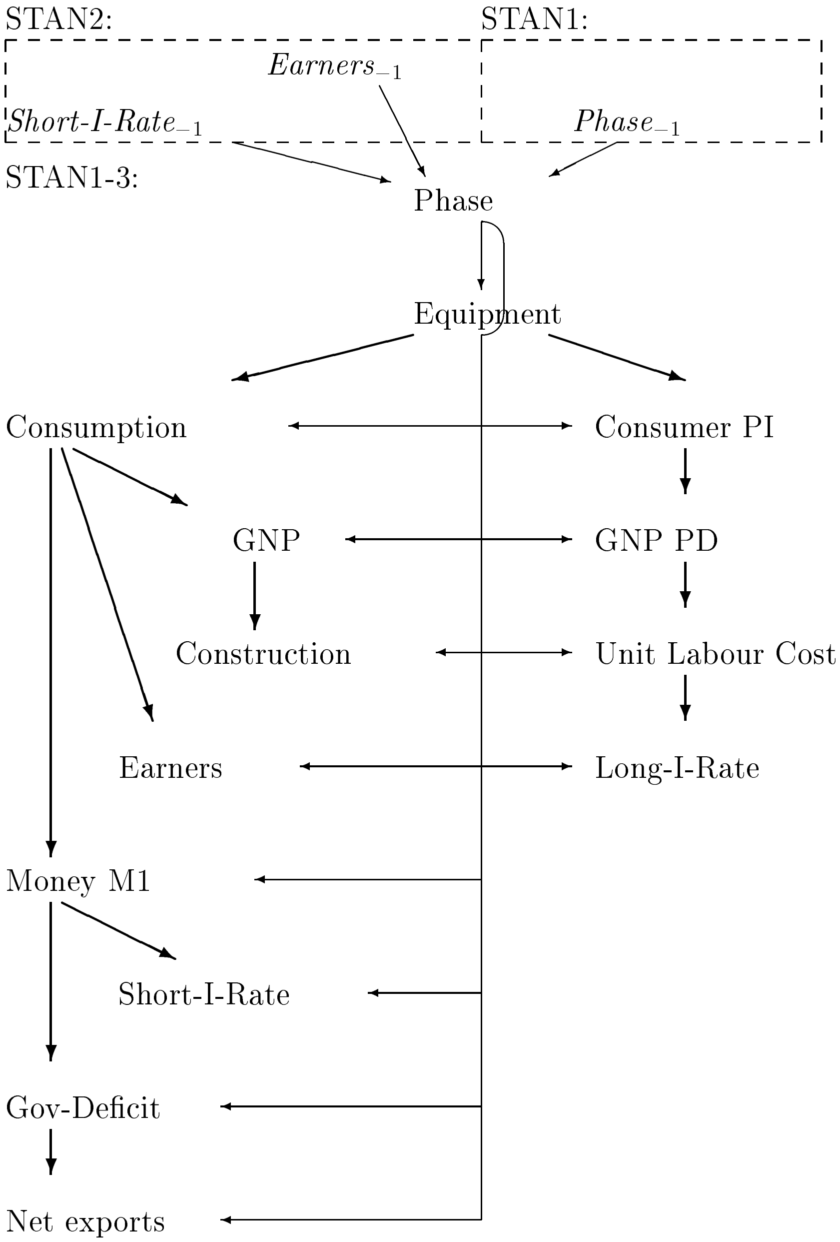


Figure 3: Structures of the learned STAN classifiers. *The inner edges form the required skeleton from the naive Bayes classifier, the edges on the left and right side are correlation edges between stylized facts, and edges at the top are transition edges learned in strategy 1 and 2 respectively.*

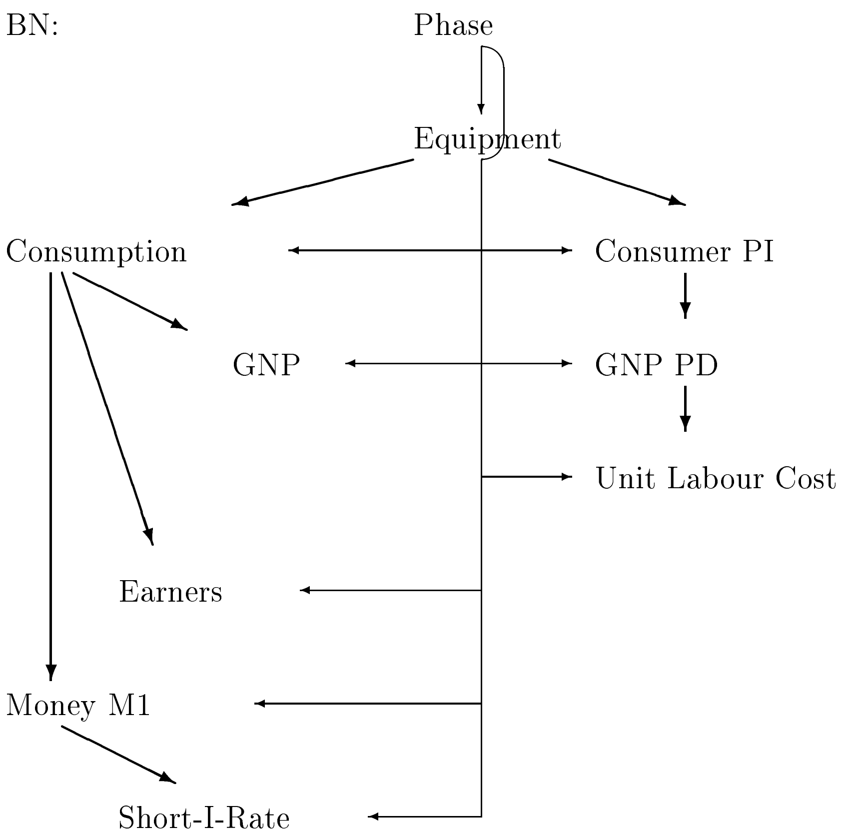


Figure 4: Structure of the learned BN classifier.

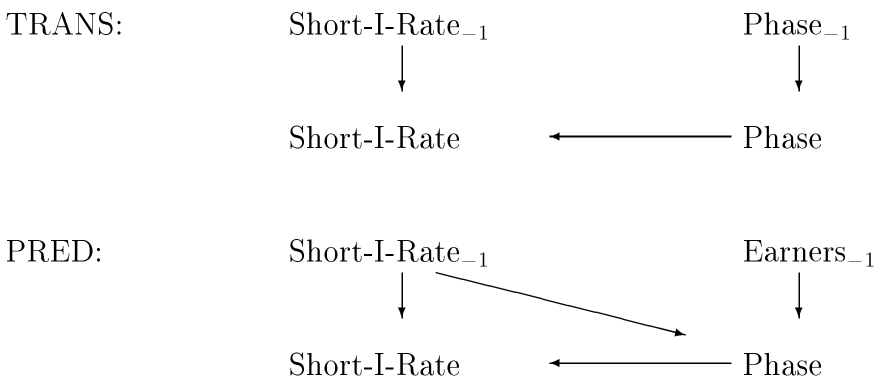


Figure 5: Structures of the learned TRANS and PRED classifier.

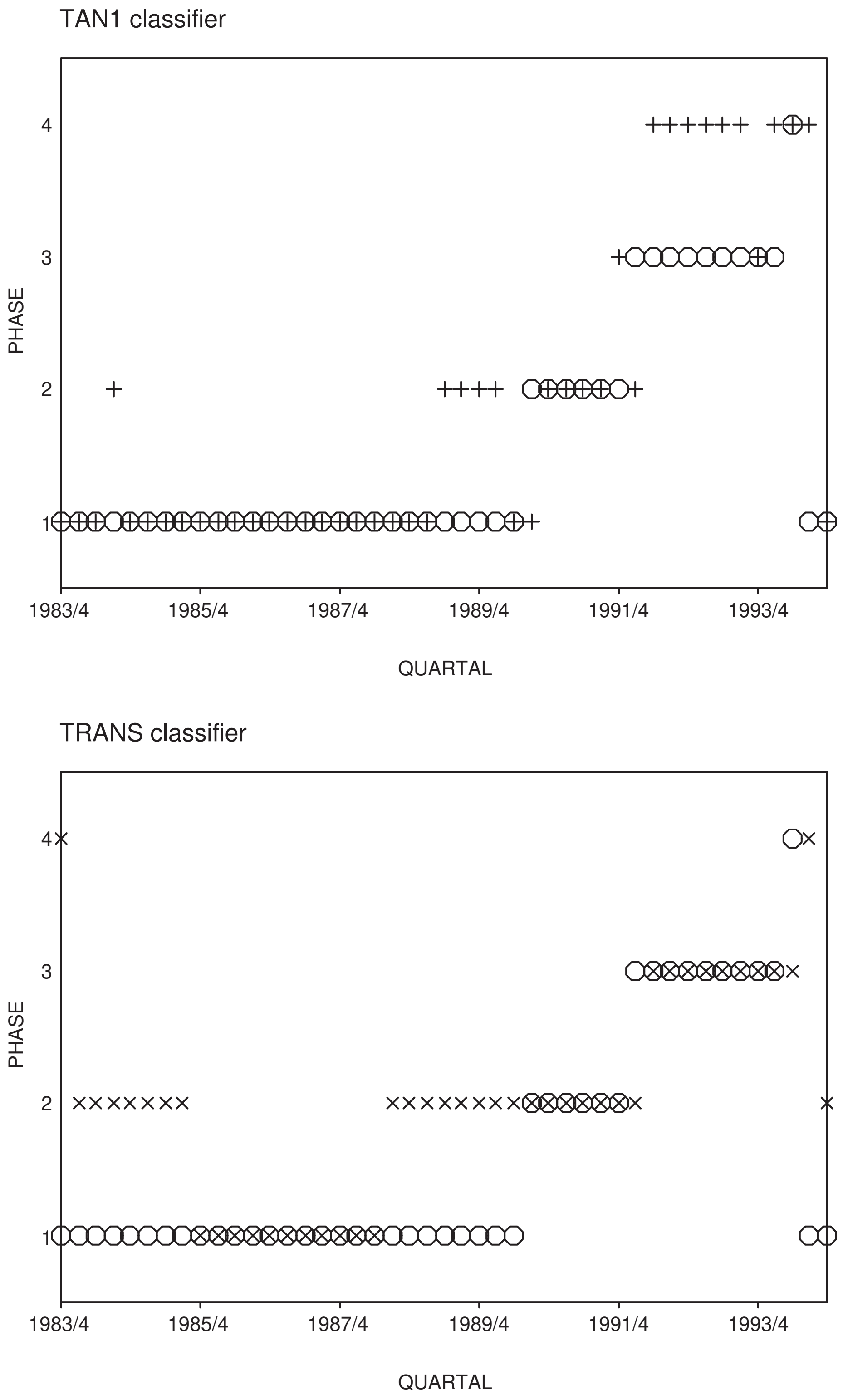
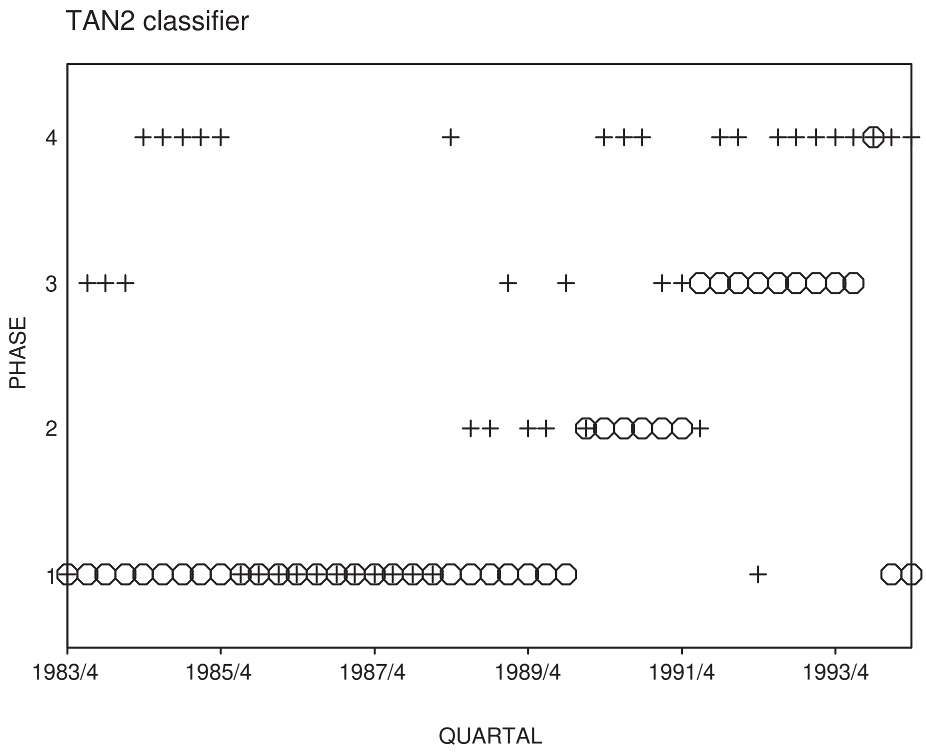


Figure 6: Performance of classifiers of Strategy 1 on the test set. The classification was carried out: by experts (○), with STAN1 (+), and with TRANS (×).



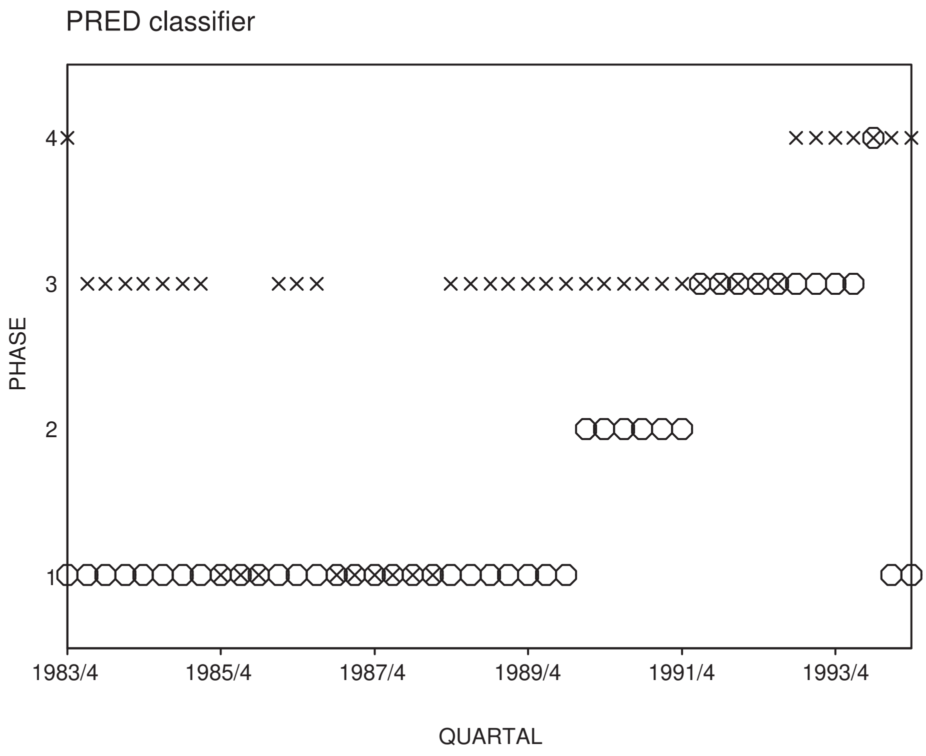


Figure 7: Performance of classifiers of Strategy 2 on the test set. The classification was carried out: by experts ($\circ$), with STAN2 (+), and with PRED ($\times$).

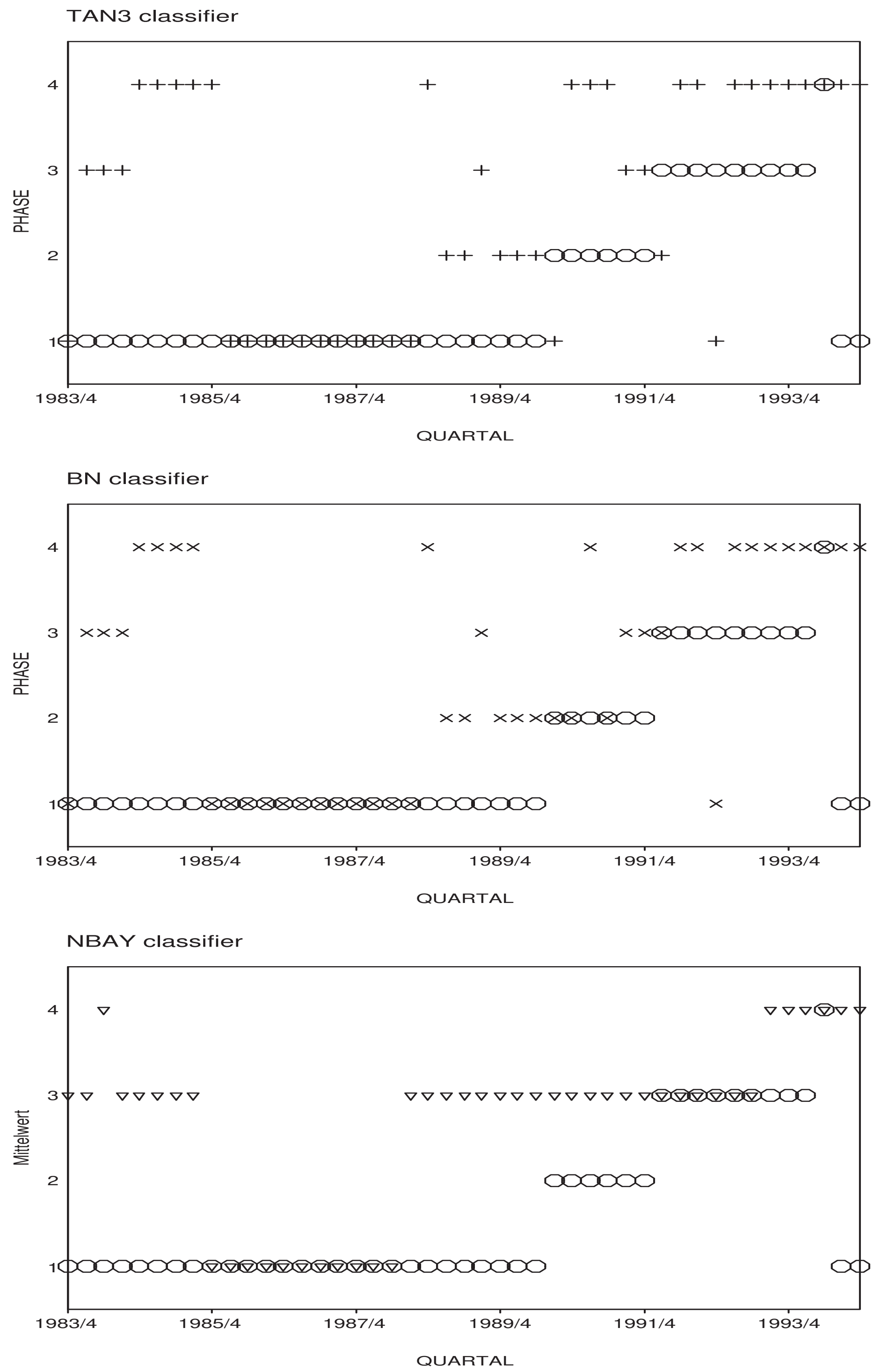


Figure 8: Performance of classifiers of Strategy 3 on the test set. The classification was carried out: by experts ($\circ$), with STAN3 ($+$), with BN ($\times$), and with NBAY (∇).