Adam, Timo; Ötting, Marius; Michels, Rouven

Markov-switching decision trees

AStA Advances in Statistical Analysis

Suggested Citation: Adam, Timo; Ötting, Marius; Michels, Rouven (2024) : Markov-switching decision trees, AStA Advances in Statistical Analysis, ISSN 1863-818X, Springer Berlin Heidelberg, Berlin/Heidelberg, Vol. 108, Iss. 2, pp. 461-476, https://doi.org/10.1007/s10182-024-00501-6

AStA Advances in Statistical Analysis (2024) 108:461–476
https://doi.org/10.1007/s10182-024-00501-6

ORIGINAL PAPER

Markov-switching decision trees

Timo Adam[1,2] · **Marius Ötting**[2] · **Rouven Michels**[2]

Received: 30 November 2022 / Accepted: 31 October 2023 / Published online: 29 May 2024

Abstract

Decision trees constitute a simple yet powerful and interpretable machine learning tool. While tree-based methods are designed only for cross-sectional data, we propose an approach that combines decision trees with time series modeling and thereby bridges the gap between machine learning and statistics. In particular, we combine decision trees with hidden Markov models where, for any time point, an underlying (hidden) Markov chain selects the tree that generates the corresponding observation. We propose an estimation approach that is based on the expectation-maximisation algorithm and assess its feasibility in simulation experiments. In our real-data application, we use eight seasons of National Football League (NFL) data to predict play calls conditional on covariates, such as the current quarter and the score, where the model's states can be linked to the teams' strategies. R code that implements the proposed method is available on GitHub.

Keywords Decision trees · EM algorithm · Hidden Markov models · Time series modeling

Timo Adam and Marius Ötting have contributed equally to this work.

✉ Timo Adam
timo.adam@uni-bielefeld.de

✉ Marius Ötting
marius.oetting@uni-bielefeld.de

Rouven Michels
r.michels@uni-bielefeld.de

[1] Department of Mathematical Sciences, University of Copenhagen, Universitetsparken 5, 2100 Copenhagen, Denmark

[2] Faculty of Business Administration and Economics, Bielefeld University, Universitätsstraße 25, 33615 Bielefeld, Germany

1 Introduction

Driven by an ever-increasing amount of data, machine learning has revolutionised empirical research in various fields. In many of these fields, machine learning tools have been applied to time series, e.g., in ecology (Wang 2019; Wijeyakulasuriya et al. 2020; Nathan et al. 2022), finance (Choudhry and Garg 2008; Das and Padhy 2012), and sports (Power et al. 2017; Decroos et al. 2019), to name but a few examples. However, standard machine learning tools are designed only for cross-sectional data, as they assume the observations of the response variable to be independent of each other. Moreover, as machine learning tools lack a time series component, they are not able to account for common characteristics typically found in time series, such as trends or cyclical fluctuations around these trends.

In this paper, we demonstrate how to overcome such limitations by proposing a combination of decision trees and time series modeling. In particular, we combine decision trees with a versatile class of time series models, namely hidden Markov models (HMMs), where the observations are assumed to be driven by underlying (hidden) states. In practical applications, such states can serve as proxies for the state of the economy (Goodwin 1993; McCulloch and Tsay 1994; Oelschläger and Adam 2021; Adam et al. 2022), the behavioural mode of an animal (DeRuiter et al. 2017; Leos-Barajas et al. 2017, 2017b; Adam et al. 2019), or, in the context of sports, the momentum or tactics of teams (Sandholtz and Bornn 2020; Sandri et al. 2020; Ötting et al. 2021; Ötting and Karlis 2022). The state process is modeled by a Markov chain, which induces serial correlation in the observations. Adding such a time series component to machine learning tools, as we will demonstrate here for the specific case of decision trees, can improve the model's fit, its prediction accuracy, and its interpretability.

To fit such Markov-switching decision trees to time series, we consider the expectation-maximisation (EM) algorithm, which is routinely used for model fitting in HMMs (Zucchini et al. 2016). The EM algorithm alternates between an expectation (E) step, in which the states are guessed based on the current parameter estimates, and a maximisation (M) step, in which the model's likelihood is maximised with respect to the parameters using the state guesses obtained in the previous E step.

To demonstrate the usefulness of the proposed method, we present a simulation experiment and a case study from sports analytics, namely American football, a sport which has seen a steady rise in statistical analyses in recent years (see, e.g., Yam and Lopez 2019; Yurko et al. 2019, 2020; Chu et al. 2020; Dutta et al. 2020; Lopez 2020; Reyers and Swartz 2021). In particular, we model play calls, which can either be a run or a pass. Due to their intuitive interpretation, decision trees have previously been used to predict such play calls in American football (Joash Fernandes et al. 2020). In our case study, the hidden states can serve as proxies for the level of risky playing style, and covariates, such as the current score or the number of yards teams need for a new first down, are considered to build the state-dependent trees.

The paper is structured as follows: in Sect. 2, we introduce HMMs as well as decision trees and outline the EM algorithm that is used for model fitting. In Sect. 3, we simulate data from Markov-switching decision trees and demonstrate the feasibility of our approach by comparing misclassification rates between fitted Markov-switching decision trees and standard decision trees. In Sect. 4, we present our case study of play call predictions, where we compare the predictive power of the proposed method to standard decision trees. R code that implements the proposed method is available on GitHub.[1]

2 Methods

In this section, we provide a brief introduction to HMMs and decision trees (Sect. 2.1) and introduce the EM algorithm that is used for model fitting (Sect. 2.2).

2.1 Model formulation and dependence structure

HMMs comprise two stochastic processes; the observation process, $\{Y_t\}_{t=1,\dots,T}$, and the underlying (hidden) state process, $\{S_t\}_{t=1,\dots,T}$. The latter is modeled by an N-state, first-order Markov chain. The $N \times N$ transition probability matrix (t.p.m.), $\mathbf{\Gamma}$, summarises the state transition probabilities, $\gamma_{ij} = \Pr(S_t = j \mid S_{t-1} = i), i, j = 1, \dots, N$. For the start of the time series, one can assume the Markov chain to be in its stationary distribution, such that the initial distribution $\boldsymbol{\delta}$ is given by the solution to $\boldsymbol{\delta}\mathbf{\Gamma} = \boldsymbol{\delta}$ subject to $\sum_{i=1}^{N} \delta_i = 1$. If this assumption is not being made, then the $N - 1$ (free) parameters in $\boldsymbol{\delta}$ need to be estimated.

In basic HMMs, the state that is active at time t, S_t, selects one of N possible distributions that generates the corresponding observation, Y_t. For instance, for binary variables, a standard choice for the state-dependent distributions would be different Bernoulli distributions, where the success probabilities vary across the states. In this paper, we do not make any such parametric distributional assumption, and instead let the Markov chain select one of N possible decision trees that generates the corresponding observation. The dependence structure of Markov-switching decision trees is illustrated in Fig. 1.

For fitting the state-dependent trees, we use the CART algorithm proposed by Breiman et al. (1984), where we focus on classification trees. In particular, we use the Gini index as impurity measure to select the splitting variables and the split points. For each state i, we thus obtain a tree consisting of M_i regions, $R_{m_i}, i = 1, \dots, N, m_i = 1, \dots, M_i$, which is built using p covariates, $\mathbf{x}_t = (x_{t1}, \dots, x_{tp})$. To select the tree size, we consider standard procedures, such as stopping the

[1] See https://github.com/timoadam/MarkovSwitchingDecisionTrees.

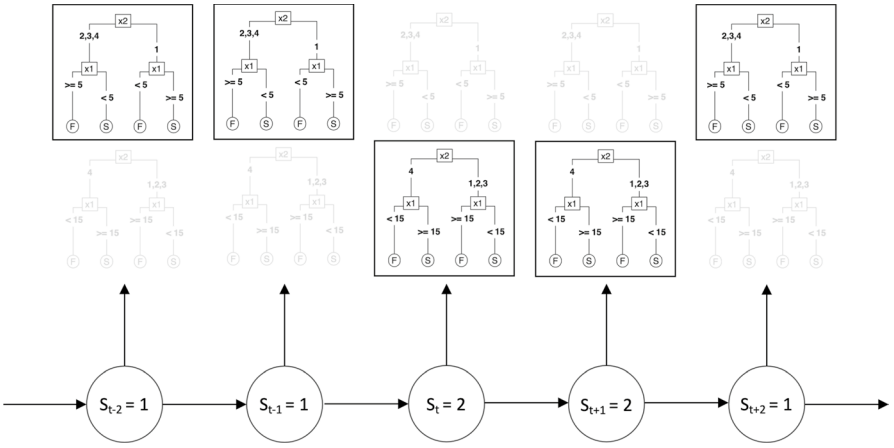

Fig. 1 Dependence structure of Markov-switching decision trees. The (hidden) state that is active at time t, S_t, selects one of N (in this illustration, $N = 2$) possible decision trees that generates the corresponding observation, Y_t (in this illustration, $Y_t \in \{\mathrm{F}, \mathrm{S}\}$ (i.e., "success" or "failure") is a binary outcome)

splitting only when a minimum node size is reached or using cost-complexity pruning as proposed by Breiman et al. (1984).

2.2 Model fitting using the EM algorithm

2.2.1 The complete-data log-likelihood

We start by representing the state sequence $\{S_t\}_{t=1,\ldots,T}$ by the indicator variables $u_i(t) = I(S_t = i)$ and $v_{i,j}(t) = I(S_{t-1} = i, S_t = j)$, $i, j = 1, \ldots N$, $t = 1, \ldots, T$. The joint log-likelihood of the observations and the states (i.e., the complete-data log-likelihood; CDLL) can then be written as

$$
\begin{aligned}
l(\boldsymbol{\theta}) &= \log\left(\delta_{s_1} \prod_{t=2}^{T} \gamma_{s_{t-1},s_t} \prod_{t=1}^{T} \Pr(Y_t = y_t \mid S_t = s_t)\right) \\
&= \log(\delta_{s_1}) + \sum_{t=2}^{T} \log(\gamma_{s_{t-1},s_t}) + \sum_{t=1}^{T} \log\left(p_{s_t}(y_t)\right) \\
&= \sum_{i=1}^{N} u_i(1) \log(\delta_i) + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T} v_{i,j}(t) \log(\gamma_{i,j}) + \sum_{i=1}^{N} \sum_{t=1}^{T} u_i(t) \log\left(p_i(y_t)\right)
\end{aligned}
$$

where $p_i(y_t) = \Pr(Y_t = y_t \mid S_t = i)$ and

$$\Pr(Y_t = k \mid S_t = i) = \frac{1}{n_{\tilde{m}_i}} \sum_{\substack{j = 1, \dots, T : \\ \mathbf{x}_j \in R_{\tilde{m}_i}}} I(y_j = k), \tag{1}$$

with $\tilde{m}_i \in 1, \dots, M_i$ being the node for which $\mathbf{x}_t \in R_{\tilde{m}_i}$ and $n_{\tilde{m}_i}$ denoting the number of observations in region $R_{\tilde{m}_i}$ for the tree of state i. In other words, Eq. (1) gives the probability that Y_t equals k in state i by calculating the proportion of class k observations for the node $\tilde{m}_i$ that is uniquely determined by $\mathbf{x}_t$. Note that the CDLL consists of three separate summands, each of which only depends on (i) the initial state probabilities, (ii) the transition probabilities, and (iii) the probabilities of the observations under the state-dependent trees, which considerably simplifies the maximisation within the M-step.

2.2.2 The E-step

The E-step consists of computing the conditional expectations of the indicator variables that represent the state sequence. To compute these, we require the forward and backward probabilities. The forward probabilities, which are denoted by $\alpha_t(i) = \Pr(Y_1 = y_1, \dots, Y_t = y_t, S_t = i)$, are summarised in the row vectors $\boldsymbol{\alpha}_t = (\alpha_t(1), \dots, \alpha_t(N))$, which can be evaluated via the forward algorithm by applying the recursion

$$\begin{aligned} \boldsymbol{\alpha}_1 &= \boldsymbol{\delta}\mathbf{P}(y_1); \\ \boldsymbol{\alpha}_t &= \boldsymbol{\alpha}_{t-1}\boldsymbol{\Gamma}\mathbf{P}(y_t), \end{aligned}$$

$t = 2, \dots, T$, with $N \times N$ diagonal matrix $\mathbf{P}(y_t) = \text{diag}(p_1(y_t), \dots, p_N(y_t))$.

The backward probabilities, which are denoted by $\beta_t(j) = f(y_{t+1}, \dots, y_T \mid S_t = j)$, are summarised in the row vectors $\boldsymbol{\beta}_t = (\beta_t(1), \dots, \beta_t(N))$, which can be evaluated via the backward algorithm by applying the recursion

$$\begin{aligned} \boldsymbol{\beta}_T &= \mathbf{1}; \\ \boldsymbol{\beta}_t^\top &= \boldsymbol{\Gamma}\mathbf{P}(y_{t+1})\boldsymbol{\beta}_{t+1}^\top, \end{aligned}$$

$t = T - 1, \dots, 1$, with $\mathbf{P}(y_{t+1})$ as defined above. We let $\hat{\alpha}_t^{[m]}(i)$ and $\hat{\beta}_t^{[m]}(j)$ denote the forward and backward probabilities obtained in the m-th iteration, which are computed using the estimates obtained in the $(m-1)$-th iteration (or initial values in the first iteration).

The m-th E-step involves the computation of the conditional expectations of the indicator variables given the current parameter estimates and fitted, state-dependent trees.

- Since $\hat{u}_i(t) = \Pr(S_t = i \mid y_1, \dots, y_T) = f(y_1, \dots, y_t, S_t = i) f(y_{t+1}, \dots, y_T \mid S_t = i) / f(y_1, \dots, y_T)$ and $f(y_1, \dots, y_T) = \sum_{i=1}^N f(y_1, \dots, y_T, S_t = i)$, it follows from the definition of the forward and backward probabilities that

$$\hat{u}_i^{[m]}(t) = \frac{\hat{\alpha}_t^{[m]}(i)\hat{\beta}_t^{[m]}(i)}{\sum_{k=1}^{N} \hat{\alpha}_T^{[m]}(k)}, \tag{2}$$

$i = 1, \ldots, N, t = 1, \ldots, T$.

- Since $\hat{v}_{i,j}(t) = \Pr(S_{t-1} = i, S_t = j \mid y_1, \ldots, y_T) = f(y_1, \ldots, y_{t-1}, S_{t-1} = i) \Pr(S_t = j \mid S_{t-1} = i) \Pr(y_t, \ldots, y_T \mid S_t = j) / \Pr(y_1, \ldots, y_T)$, it follows from the definition of the forward, backward, and transition probabilities that

$$\hat{v}_{i,j}^{[m]}(t) = \frac{\hat{\alpha}_{t-1}^{[m]}(i)\hat{\gamma}_{i,j}^{[m-1]}\hat{p}_j(y_t)\hat{\beta}_t^{[m]}(j)}{\sum_{j=1}^{N} \hat{\alpha}_T^{[m]}(j)}, \tag{3}$$

$i, j = 1, \ldots, N, t = 1, \ldots, T$.

Note that, while the indicator variables are deterministic, the above conditional expectations are probabilities: Eq. (2) denotes the probability of state i being active at time t, while Eq. (3) denotes the probability of switching from state i to state j at time t.

2.2.3 The M-step

The m-th M-step involves the maximisation of the CDLL with the indicator variables replaced by their current conditional expectations [see Eq. (2)] with respect to the model parameters:

- As only the first term in the CDLL depends on δ_i, using a Lagrange multiplier to ensure that $\sum_{i=1}^{N} \hat{\delta}_i^{[m]} = 1$ results in

$$\hat{\delta}_i^{[m]} = \frac{\hat{u}_i^{[m]}(1)}{\sum_{i=1}^{N} \hat{u}_i^{[m]}(1)} = \hat{u}_i^{[m]}(1),$$

$i = 1, \ldots, N$.

- Similarly, as only the second term in the CDLL depends on $\gamma_{i,j}$, using a Lagrange multiplier to ensure that $\sum_{j=1}^{N} \hat{\gamma}_{i,j}^{[m]} = 1, i = 1, \ldots, N$, results in

$$\hat{\gamma}_{i,j}^{[m]} = \frac{\sum_{t=2}^{T} \hat{v}_{i,j}^{[m]}(t)}{\sum_{k=1}^{N} \sum_{t=2}^{T} \hat{v}_{i,k}^{[m]}(t)},$$

$i, j = 1, \ldots, N$.

- As only the third term in the CDLL depends on Eq. (1), the optimisation problem effectively reduces to maximising the joint probability of the observations under the state-dependent trees, where the t-th observation is weighted by the $\hat{u}_i^{[m]}(t)$'s. Thus, we can exploit existing algorithms, namely the CART algorithm (Breiman

et al. 1984), to fit the state-dependent trees, where the observations are weighted according to Eq. (2) (i.e., the state-dependent trees are re-fitted in each M-step using the weights that were obtained in the previous M-step). Such weighting of the observations within the CART algorithm is implemented in the R package rpart (Therneau and Atkinson 2019).

The EM algorithm alternates between the E- and the M-step until some convergence criterion is satisfied. Here, we consider the absolute difference between the CDLLs obtained in two consecutive iterations and stop the algorithm if it falls below 10^{-3}.

3 Simulation experiment

As decision trees are a non-parametric machine learning tool, it is not feasible to compare estimated parameters with true parameters, as is typically done in a regression context. However, we can explore the viability of Markov-switching decision trees by simulating data from a Markov-switching decision tree and comparing the performance of fitted trees with and without a Markovian structure. Specifically, we simulate 100 time series, each consisting of 2000 observations. For each time series, we generate binary observations, denoted as Y_t, where Y_t can take values from the set $\{\mathrm{F}, \mathrm{S}\}$ (representing "success" or "failure") based on a Markov-switching decision tree with initial state distribution $\boldsymbol{\delta} = (0.5, 0.5)$, t.p.m.

$$\boldsymbol{\Gamma} = \begin{pmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{pmatrix},$$

a uniformly distributed covariate $x_1 \in [0, 20]$, and a categorical covariate $x_2 \in \{1, \ldots, 4\}$.

Figure 2 displays the true data-generating trees along with the results obtained from fitted standard decision trees and Markov-switching decision trees for a single simulation run. In the bottom panel, it is evident that the Markov-switching decision trees effectively capture the splits of the data-generating trees. Furthermore, the success probabilities estimated in the leaf nodes closely align with those of the data-generating process. The means of the estimated diagonal values of the t.p.m. are calculated as 0.949 and 0.948 for γ_{11} and γ_{22}, respectively. These estimates are remarkably close to the corresponding true values of 0.95. In contrast, as depicted in the middle panel of Fig. 2, standard decision tree fail to identify the correct thresholds for the splits and do not accurately estimate the "success" and "failure" probabilities.

This visual assessment is substantiated by comparing the predictive performance of both approaches using out-of-sample observations. For forecasting observations in the Markov-switching context, we employ the standard hidden Markov model (HMM) framework to generate one-step-ahead forecasts (Zucchini et al. 2016). The resulting misclassification rates across 100 simulation runs are presented in Fig. 3. It is evident that the predictive performance of the Markov-switching decision trees (median misclassification rate: 0.115) surpasses that of the standard decision trees (median misclassification rate: 0.275) by a substantial margin. In addition, due to

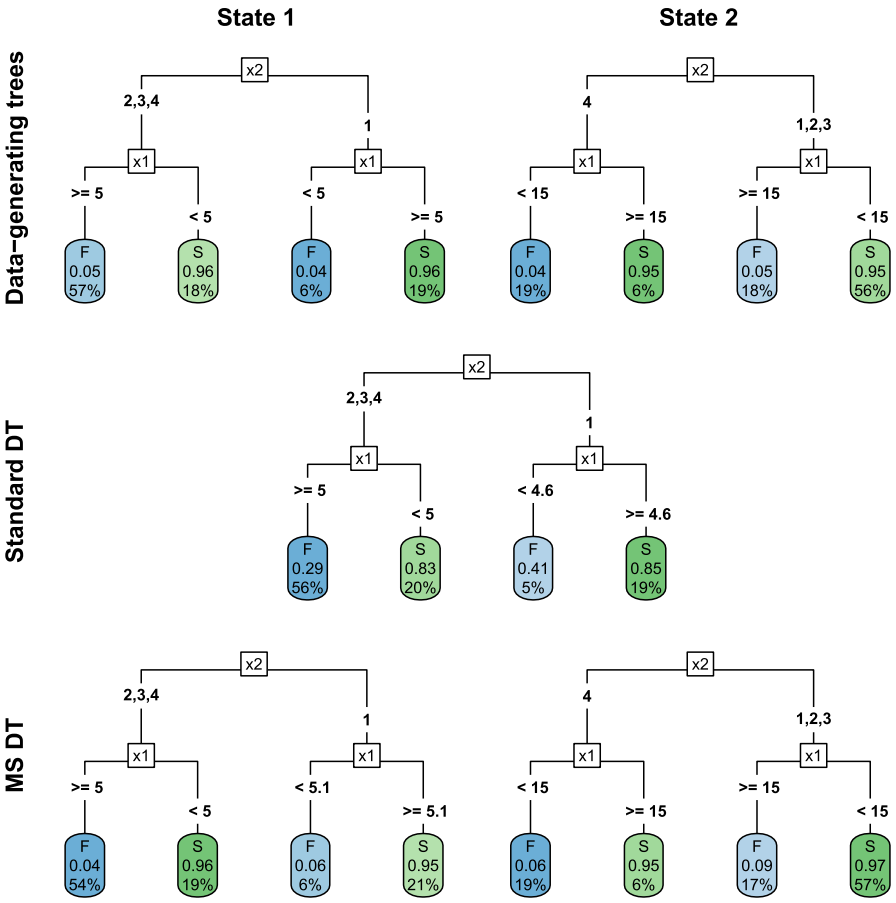

Fig. 2 True data-generating trees (top), example fitted standard decision trees (middle), and example fitted Markov-switching decision tree (bottom). When the state process is in state 1, a "success" outcome is generated with a probability of 95% for $x_1 < 5$ if $x_2 = 1$ and for $x_1 \geq 5$ if $x_2 = 1$. When the state process is in state 2, this pattern is reversed, i.e., the above combination of higher and lower values most likely leads to a "failure" outcome

the lack of flexibility caused by the missing time-series component, the standard decision trees are, on average, deeper than the Markov-switching trees. While the average tree depth obtained for the former is 2.8, the average tree depth obtained for the latter is 2.1. Hence, the average depth of the Markov-switching trees is much closer to the true depth (i.e., 2), which is due to the fact that the standard decision trees are less flexible and therefore require more splits.

In this simple simulation experiment, we demonstrated potential pitfalls associated with the application of decision trees to time series that exhibit serial correlation and state-switching over time. Without incorporating this time series structure, standard decision trees fail to deliver precise forecasts of future observations. On the contrary, Markov-switching decision trees appropriately account for state-switching over time,

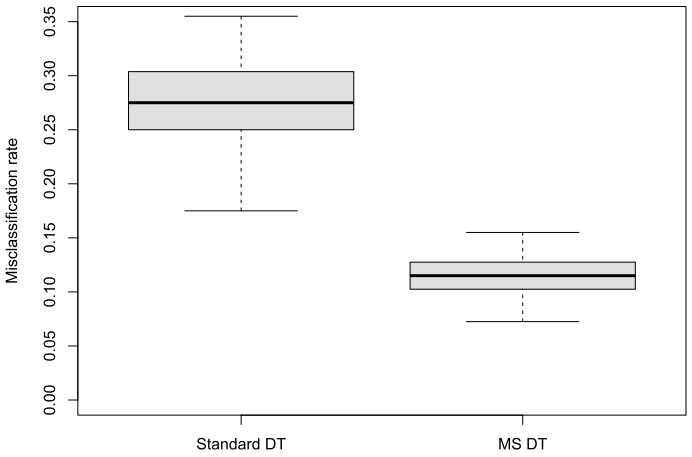

Fig. 3 Boxplots of misclassification rates for standard decision trees (left) and Markov-switching decision trees (right) obtained across 100 simulation runs

enabling more accurate predictions. Additionally, standard decision trees can lead to misleading interpretations of the relationship between covariates and observations, further emphasising the importance of incorporating the time series structure.

4 Application to American football data

In American football, the possession team (i.e., the offense) attempts to reach the opposing team's (i.e., the defense) end zone by either running or passing the ball. For the defense, it is thus of interest to accurately predict the opponent's play. For that purpose, and driven by the availability of play-by-play NFL data, the use of machine learning approaches for play call predictions has been investigated in multiple studies (see, e.g., Heiny and Blevins 2011; Joash Fernandes et al. 2020; Wu et al. 2021). In particular, Joash Fernandes et al. (2020) argue that decision trees are most likely to be adopted in practice, as they are intuitive to interpret (as opposed to alternative, black-box approaches).

In this section, we fit Markov-switching decision trees to NFL play-by-play data, where the underlying states serve as a proxy for the current level of a team's risk-taking—more risky styles of play are usually aligned with a higher propensity to throw a pass (as opposed to performing a run).

4.1 Data

We consider play-by-play data for 8 NFL seasons from 2012–13 until 2018–19 retrieved from www.kaggle.com,[2] covering 2526 regular-season games and 319,369

[2] https://www.kaggle.com/datasets/maxhorowitz/nflplaybyplay2009to2016.

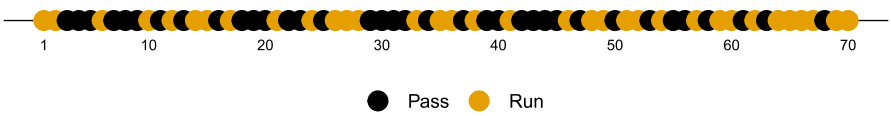

Fig. 4 Example time series found in the data: play calls of the New England Patriots observed for the game against the Pittsburgh Steelers played on November 03, 2013

plays (i.e., run or pass) in total. Our covariates are the quarter (*qtr*), the difference in the current score (*score_differential*), the current *down* (e.g., one corresponds to the first of four possible attempts to reach the new first down), the yards to go for a new first down (*ydstogo*), whether the quarterback is in shotgun formation (*shotgun*), i.e., the quarterback stands five yards behind the center at the beginning of a play, and a dummy indicating whether the match was played at home. These covariates have also been considered in the previous studies on NFL play-call prediction briefly introduced above.

One example time series found in our data, corresponding to the 70 play calls observed for the New England Patriots in the game against the Pittsburgh Steelers, is shown in Fig. 4. The play calls underline that there are periods with a high number of passing plays (e.g., at the beginning of the game and around play call 30) as well as those where more runs are called (e.g., after the 60-th play), which motivates the need for a time series modeling approach to account for the serial correlation present in the data.

To evaluate the predictive performance, the season 2018–19 serves as test data, and the Markov-switching decision trees are fitted to data from seasons 2012–13 until 2017–18. We compare the predictive performance of our approach to a standard decision tree.

4.2 Results

To address heterogeneity across teams, we fitted Markov-switching decision tree with $N = 2$ states to the data for each team separately instead of pooling the data of all teams. To avoid local maxima, for all teams we fitted the model 100 times, each time considering random starting values for the numerical maximisation. On average, it took approximately one minute to fit a single model.

Here, we present the fitted state-dependent trees only for one team, namely the New England Patriots, which is one of the most successful teams in the period considered.[3] To facilitate interpretation, we fitted a Markov-switching decision tree with a tree depth of 3 to analyze the play-calls of the New England Patriots. In Fig. 5, the fitted trees for both states are displayed, indicating that a run is less likely in state 1 compared to state 2. Considering that run plays involve less risk (as evidenced by the greater variance in yards gained during passing plays), state 2 can be interpreted

[3] Readers interested in the results for the remaining teams can use the R code provided in the GitHub repository: https://github.com/timoadam/MarkovSwitchingDecisionTrees.

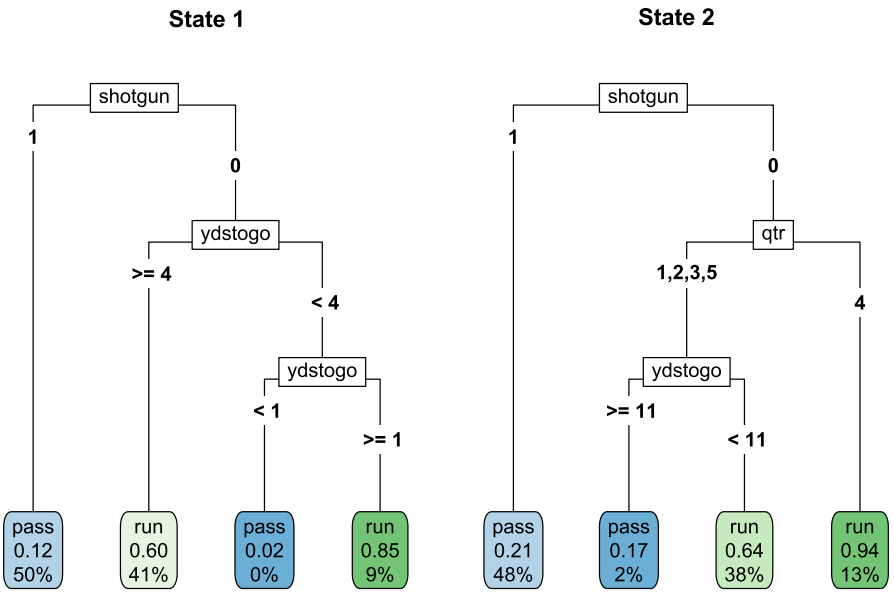

Fig. 5 Markov-switching decision trees fitted to data of the New England Patriots with a tree depth of 3

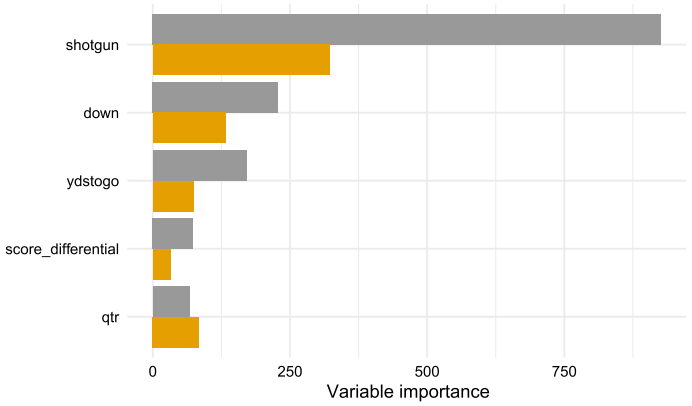

Fig. 6 Variable importance plots for the simple fitted Markov-switching decision trees shown in Fig. 5, i.e., with a tree depth of 3

as representing a less risky style of play, while state 1 is more aligned with riskier plays. However, it should be noted that, in general, the underlying states in HMMs can only be seen as crude approximations of the actual underlying behavior.

The understanding of the two states can be enhanced through variable importance plots, presented in Fig. 6. In both states, the most crucial predictors are being in shotgun formation and the down. However, the importance of the remaining predictors varies between the two states. In state 1, the yards to go emerges as the

Table 1 The first two columns display the misclassification rates for each team's test data using the standard decision tree and the Markov-switching decision tree, and the last two columns display the tree depth

Team	Misclassification rates		Tree depth	
	Standard DT	MS DT	Standard DT	MS DT
All	0.288	**0.286**	6.250	3.172
PIT	0.225	**0.223**	8	1.5
TEN	0.300	**0.286**	6	1.5
CLE	**0.268**	0.276	8	5
MIN	0.253	**0.239**	8	2.5
NO	**0.253**	0.255	9	2.5
DET	0.243	**0.241**	3	2
DAL	0.257	**0.255**	1	3
TB	0.265	**0.262**	9	2
HOU	0.305	**0.298**	7	5
NYJ	0.259	**0.254**	7	4.5
IND	0.285	0.285	7	3
JAX	**0.286**	0.292	5	5
DEN	**0.249**	0.252	7	5
CIN	**0.284**	0.298	4	1
CAR	0.337	**0.324**	6	3
PHI	**0.323**	0.345	6	3.5
KC	**0.324**	0.327	5	2
BAL	0.391	**0.379**	6	4.5
ATL	0.288	**0.285**	8	4
MIA	**0.334**	0.349	8	4
ARI	**0.266**	0.270	8	3.5
SF	**0.303**	0.310	10	5
SEA	0.416	**0.374**	7	4
LA	**0.251**	0.265	1	1.5
NYG	0.228	**0.227**	7	1
WAS	**0.272**	0.284	7	4.5
GB	0.315	**0.303**	1	1
CHI	0.363	**0.356**	7	3
NE	**0.209**	0.212	6	2
BUF	0.319	**0.300**	8	3
OAK	**0.281**	0.288	9	3
LAC	0.273	0.273	1	5.5

For the Markov-switching decision trees, the average tree depth of the two states is shown

Lower misclassification rates are indicated in bold

third most influential predictor, whereas in state 2, the actual quarter holds greater importance.

To further assess the usefulness of our proposed method, we compare the predictive performance of the fitted Markov-switching decision trees to standard decision

trees which do not account for the time series structure of the data. For the 32 teams, we predict all play calls of the 2018–19 season. When fitting the Markov-switching decision trees for each team individually, we do not impose a restriction on the tree depth, as demonstrated in Fig. 5, i.e., the tree depth can be larger than three. To predict the play calls, we use the standard HMM machinery to obtain one-step-ahead forecasts analogously to the simulation experiment (Zucchini et al. 2016). Table 1 presents the misclassification rates for all teams, comparing the forecasts obtained using the Markov-switching tree approach with those obtained using the standard decision tree approach. On average (i.e., across all teams), the misclassification rate is slightly lower when using Markov-switching decision trees (0.286 vs. 0.288). Furthermore, for more than half of the teams, the Markov-switching approach yields a lower or equal misclassification rate compared to the standard approach. To compare the complexity of the trees, Table 1 displays the tree depth obtained for both approaches. For the Markov-switching trees, we report the average tree depth of the two states. On average, the tree depth obtained for the standard decision trees is almost twice as large as the tree depth obtained for the Markov-switching trees—as in the simulation experiment, it may be the case that the standard decision trees compensate for the missing time series structure by growing more complex trees.

Although the predictive performance here is quite promising, the difference between the two approaches is fairly small—it should be noted that the application of Markov-switching decision trees to the NFL data serves only as a case study. In practice, it may very well be the case that the coaching staff has more insights into the next play call. Nevertheless, given that the computational cost of obtaining a single prediction is less than a second and the results are easily interpretable, our approach can be considered a valuable supplementary tool for the coaching staff.

5 Discussion

We have presented a versatile framework for fitting Markov-switching decision trees to time series data. Our proposed methodology has demonstrated superiority over standard decision trees when applied to data that displays both serial correlation and state-switching dynamics over time. Specifically, the enhanced flexibility of Markov-switching decision trees, with the inclusion of one tree per state, can significantly improve the accuracy of time series forecasts while maintaining interpretability of the individual trees. The simplicity and non-parametric nature of Markov-switching decision trees is to be regarded advantageous compared to existing parametric state-switching models for categorical outcomes, such as Markov-switching logistic or multinomial regression models and, more generally, Markov-switching generalised additive models (Langrock et al. 2017).

A well-known challenge with Hidden Markov Models (HMMs) revolves around determining the appropriate number of states. In our simulation experiment and case study, we focus solely on $N = 2$ states. In practical applications, the choice of the number of states can be guided by expert knowledge or evaluated using information criteria like AIC and BIC. However, applying such criteria to Markov-switching

decision trees is not straightforward, as it is necessary to derive the number of parameters used to fit the model—while the parameters associated with the state process are estimated, the trees that determine the state-dependent process are non-parametric.

While our simulation experiment and case study focused on binary outcomes, it is important to note that Markov-switching decision trees can be applied to time series with categorical outcomes in general. Our approach can also be extended to incorporate other machine learning techniques. For example, one could explore the possibilities of Markov-switching regression trees or more advanced ensemble methods like Markov-switching random forests. Thus, the methodology introduced in this paper serves as a foundation for future research, aiming to integrate machine learning techniques with time series modeling.

Supplementary information

The R code that was used for the case study presented in Sect. 4 is available at https://github.com/timoadam/MarkovSwitchingDecisionTrees.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Adam, T., Griffiths, C.A., Leos-Barajas, V., et al.: Joint modelling of multi-scale animal movement data using hierarchical hidden Markov models. Methods Ecol. Evol. **10**(9), 1536–1550 (2019). https://doi.org/10.1111/2041-210X.13241

Adam, T., Mayr, A., Kneib, T.: Gradient boosting in Markov-switching generalized additive models for location, scale, and shape. Econom. Stat. **22**, 3–16 (2022). https://doi.org/10.1016/j.ecosta.2021.04.002

Breiman, L., Friedman, J., Olshen, R., et al.: Classification and Regression Trees. Wadsworth, New York (1984). https://doi.org/10.1201/9781315139470

Choudhry, R., Garg, K.: A hybrid machine learning system for stock market forecasting. Int. J. Comput. Inf. Eng. **2**(3), 689–692 (2008). https://doi.org/10.5281/zenodo.1071852

Chu, D., Reyers, M., Thomson, J., et al.: Route identification in the National Football League. J. Quant. Anal. Sports **16**(2), 121–132 (2020). https://doi.org/10.1515/jqas-2019-0047

Das, S.P., Padhy, S.: Support vector machines for prediction of futures prices in Indian stock market. Int. J. Comput. Appl. (2012). https://doi.org/10.5120/5522-7555
Decroos, T., Bransen, L., Van Haaren, J., et al.: Actions speak louder than goals: valuing player actions in soccer. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1851–1861. https://doi.org/10.1145/3292500.3330758 (2019)
DeRuiter, S.L., Langrock, R., Skirbutas, T., et al.: A multivariate mixed hidden Markov model for blue whale behaviour and responses to sound exposure. Ann. Appl. Stat. **11**(1), 362–392 (2017). https://doi.org/10.1214/16-AOAS1008
Dutta, R., Yurko, R., Ventura, S.L.: Unsupervised methods for identifying pass coverage among defensive backs with NFL player tracking data. J. Quant. Anal. Sports **16**(2), 143–161 (2020). https://doi.org/10.1515/jqas-2020-0017
Goodwin, T.H.: Business-cycle analysis with a Markov-switching model. J. Bus. Econ. Stat. **11**(3), 331–339 (1993). https://doi.org/10.2307/1391958
Heiny, E.L., Blevins, D.: Predicting the Atlanta Falcons play-calling using discriminant analysis. J. Quant. Anal. Sports **7**(3), 2 (2011). https://doi.org/10.2202/1559-0410.1230
Joash Fernandes, C., Yakubov, R., Li, Y., et al.: Predicting plays in the National Football League. J. Sports Anal. **6**(1), 35–43 (2020). https://doi.org/10.3233/JSA-190348
Langrock, R., Kneib, T., Glennie, R., et al.: Markov-switching generalized additive models. Stat. Comput. **27**, 259–270 (2017). https://doi.org/10.1007/s11222-015-9620-3
Leos-Barajas, V., Gangloff, E.J., Adam, T., et al.: Multi-scale modeling of animal movement and general behavior data using hidden Markov models with hierarchical structures. J. Agric. Biol. Environ. Stat. **22**(3), 232–248 (2017). https://doi.org/10.1007/s13253-017-0282-9
Leos-Barajas, V., Photopoulou, T., Langrock, R., et al.: Analysis of animal accelerometer data using hidden Markov models. Methods Ecol. Evol. **8**(2), 161–173 (2017). https://doi.org/10.1111/2041-210X.12657
Lopez, M.J.: Bigger data, better questions, and a return to fourth down behavior: an introduction to a special issue on tracking datain the national football league. J. Quant. Anal. Sports **16**(2), 73–79 (2020). https://doi.org/10.1515/jqas-2020-0057
McCulloch, R.E., Tsay, R.S.: Statistical analysis of economic time series via Markov switching models. J. Time Ser. Anal. **15**(5), 523–539 (1994). https://doi.org/10.1111/j.1467-9892.1994.tb00208.x
Nathan, R., Monk, C.T., Arlinghaus, R., et al.: Big-data approaches lead to an increased understanding of the ecology of animal movement. Science **375**(6582), abg780 (2022). https://doi.org/10.1126/science.abg1780
Oelschläger, L., Adam, T.: Detecting bearish and bullish markets in financial time series using hierarchical hidden Markov models. Stat. Modell. (2021). https://doi.org/10.1177/1471082X211034048
Ötting, M., Karlis, D.: Football tracking data: a copula-based hidden Markov model for classification of tactics in football. Ann. Oper. Res. (2022). https://doi.org/10.1007/s10479-022-04660-0
Ötting, M., Langrock, R., Maruotti, A.: A copula-based multivariate hidden Markov model for modelling momentum in football. AStA Adv. Stat. Anal. (2021). https://doi.org/10.1007/s10182-021-00395-8
Power, P., Ruiz, H., Wei, X., et al.: Not all passes are created equal: objectively measuring the risk and reward of passes in soccer from tracking data. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 1605–1613, (2017) https://doi.org/10.1145/3097983.3098051
Reyers, M., Swartz, T.B.: Quarterback evaluation in the national football league using tracking data. AStA Adv. Stat. Anal. (2021). https://doi.org/10.1007/s10182-021-00406-8
Sandholtz, N., Bornn, L.: Markov decision processes with dynamic transition probabilities: an analysis of shooting strategies in basketball. Ann. Appl. Stat. **14**(3), 1122–1145 (2020). https://doi.org/10.1214/20-AOAS1348
Sandri, M., Zuccolotto, P., Manisera, M., et al.: Markov switching modelling of shooting performance variability and teammate interactions in basketball. J. R. Stat. Soc. Ser. C **69**(5), 1337–1356 (2020). https://doi.org/10.1111/rssc.12442
Therneau, T., Atkinson, B.: Rpart: recursive partitioning and regression trees. https://CRAN.R-project.org/package=rpart, R package, version 4.1–15 (2019)
Wang, G.: Machine learning for inferring animal behavior from location and movement data. Ecol. Inform. **49**, 69–76 (2019). https://doi.org/10.1016/j.ecoinf.2018.12.002
Wijeyakulasuriya, D.A., Eisenhauer, E.W., Shaby, B.A., et al.: Machine learning for modeling animal movement. PLoS ONE **15**(7), 0235750 (2020). https://doi.org/10.1371/journal.pone.0235750

Wu, J., Gunnell, E., Sun, Y.: PlayGuessr: commercial application of machine learning in football play prediction. In: CS & IT Conference Proceedings, CS & IT Conference Proceedings. (2021) https://doi.org/10.5121/csit.2021.111714

Yam, D.R., Lopez, M.J.: What was lost? A causal estimate of fourth down behavior in the national football league. J. Sports Anal. **5**(3), 153–167 (2019). https://doi.org/10.3233/JSA-190294

Yurko, R., Ventura, S., Horowitz, M.: nflWAR: a reproducible method for offensive player evaluation in football. J. Quant. Anal. Sports **15**(3), 163–183 (2019). https://doi.org/10.1515/jqas-2018-0010

Yurko, R., Matano, F., Richardson, L.F., et al.: Going deep: models for continuous-time within-play valuation of game outcomes in American football with tracking data. J. Quant. Anal. Sports **16**(2), 163–182 (2020). https://doi.org/10.1515/jqas-2019-0056

Zucchini, W., MacDonald, I.L., Langrock, R.: Hidden Markov Models for Time Series: an Introduction Using R. Chapman & Hall/CRC, Boca Raton (2016). https://doi.org/10.1201/b20790