Aalborg Universitet

Segmentation of Nonstationary Time Series with Geometric Clustering

Bocharov, Alexei; Thiesson, Bo

Published in:
Pattern Recognition - Applications and Methods

DOI (link to publication from Publisher):
10.1007/978-3-642-36530-0_8

Publication date:
2013

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Bocharov, A., & Thiesson, B. (2013). Segmentation of Nonstationary Time Series with Geometric Clustering. In P. L. Carmona, J. S. Sanchez, & A. L. N. Fred (Eds.), *Pattern Recognition - Applications and Methods* (Vol. 204, pp. 93-107). Springer Publishing Company. https://doi.org/10.1007/978-3-642-36530-0_8

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: December 15, 2025

Segmentation of Nonstationary Time Series with Geometric Clustering

Alexei Bocharov and Bo Thiesson

Microsoft Research, One Microsoft Way, Redmond, WA 98052, U.S.A.
{alexeib,thiesson}@microsoft.com

Abstract. We introduce a non-parametric method for segmentation in regime-switching time-series models. The approach is based on spectral clustering of target-regressor tuples and derives a switching regression tree, where regime switches are modeled by oblique splits. Such models can be learned efficiently from data, where clustering is used to propose one single split candidate at each split level. We use the class of ART time series models to serve as illustration, but because of the non-parametric nature of our segmentation approach, it readily generalizes to a wide range of time-series models that go beyond the Gaussian error assumption in ART models. Experimental results on S&P 1500 financial trading data demonstrates dramatically improved predictive accuracy for the exemplifying ART models.

Keywords: Regime-switching time series, Spectral clustering, Regression tree, Oblique split, Financial markets.

1 Introduction

The analysis of time-series data is an important area of research with applications in areas such as natural sciences, economics, and finance to mention a few.

Many time series exhibit nonstationarity due to regime switching. Proper detection and modeling of this switching is a major challenge in time-series analysis. In regime-switching models, different time series regimes are described by submodels with different sets of parameters. A particular submodel may apply to multiple time ranges when the underlying time series repeatedly falls into a certain regime. For example, volatility of equity returns may change when affected by events such as earnings releases or analysts' reports, and we may see similar volatility patterns around similar events.

The intuition in this paper is to match proposed regimes with modes of the joint distribution of target-regressor tuples, which is a particular kind of mixture modeling. Prior research offers quite a variety of mixture modeling approaches to the analysis of nonstationary time series. In Markov-switching models (see, e.g., [12,13]) a Markov evolving hidden state indirectly partitions the time-series data to fit local auto-regressive models in the mixture components. Another large body of work (see, e.g., [27,28]) have adapted the hierarchical mixtures of experts in [15] to time series. In these models–also denoted as gated experts–the hierarchical gates explicitly operate on the data in order to define a partition into local regimes. In both the Markov-switching and the gated expert

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 93–107.
DOI: 10.1007/978-3-642-36530-0_8 © Springer-Verlag Berlin Heidelberg 2013

models, the determination of the partition and the local regimes are tightly integrated in the learning algorithm and demands an iterative approach, such as the EM algorithm.

We focus on a conceptually simple direction that lends itself easier to explanatory analysis. The resulting design differs from the above work in at least three aspects: 1) we propose a modular separation of the regime partitioning and the regime learning, which makes it easy to experiment independently with different types of regime models and different separation methods, 2) in particular, this modularity allows for non-parametric as well as parametric regime models, or a mixture thereof, 3) the regime-switching conditions depend deterministically on data and are easy to interpret.

We model the actual switching conditions in a regime-switching model in the form of a regression tree and call it the *switching tree*. Typically, the construction of a regression tree is a stagewise process that involves three ingredients: 1) a *split proposer* that creates split candidates to consider for a given (leaf) node in the tree, 2) one or more *scoring criteria* for evaluating the benefit of a split candidate, and 3) a *search strategy* that decides which nodes to consider and which scoring criterion to apply at any state during the construction of the tree. Since the seminal paper [2] popularized the classic classification and regression tree (CART) algorithm, the research community has given a lot of attention to both types of decision trees. Many different algorithms have been proposed in the literature by varying specifics for the three ingredients in the construction process mentioned above.

Although there has been much research on learning regression trees, we know of only one setting, where these models have been used as switching trees in regime-switching time series models–namely the class of auto-regressive tree (ART) models in [18]. The ART models generalize classical auto-regressive (AR) models (e.g., [11]) by having a regression tree define the switching between the different AR models in the leafs. As such, the well-known threshold auto-regressive (TAR) models [25,24] can also be considered as a specialization of an ART model with the regression tree limited to a single split variable. The layout of our algorithms is strongly influenced by [18] (which we repeatedly refer to for comparison), but our premises and approach is very different.

In particular, we propose a different way to create the candidate splits during the switching tree construction. A split defines a predicate, which, given the values of regressor variables, decides on which side of the split a data case should belong.[1] A predicate may be as simple as checking if the value of a particular single regressor is below some threshold or not. We will refer to this kind of split as an *axial split*, and it is in fact the only type of splits allowed in the ART models. We make use of general multi-regressor split predicates, which in this paper we approximate with linear predicates called *oblique splits*. Importantly, we show evidence that for a broad class of time series, the best split is not likely to be axial.

It may sometimes be possible to consider and evaluate the efficacy of all feasible axial splits for the data associated with a node in the tree, but for combinatorial reasons, oblique splitting rarely enjoys this luxury. We therefore need a split proposer, which is more careful about the candidate splits it proposes. In fact, our approach is extreme in that respect by only proposing a *single* oblique split to be considered for any given node

[1] For clarity of presentation, we will focus on binary splits only. It is a trivial exercise to extend our proposed method to allow for n-ary splits.

during the construction of the tree. Our oblique split proposer involves a simple two step procedure. In the first step, we use a spectral clustering method to separate the data in a node into two classes. Having separated the data, the second step now proceeds as a simple classification problem, by using a linear discriminant method to create the best separating hyperplane for the two data classes. Any discriminant method can be used, and there is in principle no restriction on it being linear, if more complicated splits are sought.

Oblique splitting has enjoyed significant attention for the classification tree setting. See, e.g., [2,20,3,10,14]. Less attention has been given to the regression tree setting, but still a number of methods has come out of the statistics and machine learning communities. See, e.g., [7,16,4] to mention a few. Setting aside the time-series context for our switching trees, the work in [7] is in style the most similar to the oblique splitting approach that we propose in this paper. In [7], the EM algorithm for Gaussian mixtures is used to cluster the data. Having committed to Gaussian clusters it now makes sense to determine a separating hyperplane via a quadratic discriminant analysis for a projection of the data onto a vector that ensures maximum separation of the Gaussians. This vector is found by minimizing Fisher's separability criterion.

Our approach to proposing oblique split candidates is agnostic to any specific parametric assumptions on the noise distribution and therefore accommodates without change non-Gaussian or even correlated errors (thus our method is more general than ART, which relies on univariate Gaussian quantiles as split candidates). This approach allows us to use spectral clustering - a non-parametric segmentation tool, which has been shown to often outperform parametric clustering tools (see, e.g., [26]).

Spectral clustering dates back to the work in [8,9] that suggest to use the method for graph partitionings. Variations of spectral clustering have later been popularized in the machine learning community [23,19,21], and, importantly, very good progress has been made in improving an otherwise computationally expensive eigenvector computation for these methods [29]. We use a simple variation of the method in [21] to create a spectral clustering for the time series data in a node. Given this clustering, we then use a simple perceptron learning algorithm (see, e.g., [1]) to find a hyperplane that defines a good oblique split predicate for the autoregressors in the model.

Let us now turn to the possibility of splitting on the time feature in a time series. Due to the special nature of time, it does not make sense to involve this feature as an extra dimension in the spectral clustering; it would not add any discriminating power to the method. Instead, we propose a procedure for time splits, which uses the clustering in another way. The procedure identifies specific points in time, where succeeding data elements in the series cross the cluster boundary, and proposes time splits at those points. Our split proposer will in this way use the spectral clustering to produce both the oblique split candidate for the regressors, and a few very targeted (axial) split candidates for the time dimension.

The rest of the paper is organized as follows. In Section 2, we briefly review the ART models that we use as a baseline, and we define and motivate the extension that allows for oblique splits. Section 3 reviews the general learning framework for ART models. Section 4 contains the details for both aspects of our proposed spectral splitting method– the oblique splitting and the time splitting. In Sections 5 and 6 we describe experiments and provide experimental evidence demonstrating that our proposed spectral splitting method dramatically improves the quality of the learned ART models over the current approach. We will conclude in Section 7.

2 Standard and Oblique ART Models

We begin by introducing some notation. We denote a temporal sequence of variables by $X = (X_1, X_2, \ldots, X_T)$, and we denote a sub-sequence consisting of the i'th through the j'th element by $X_i^j = (X_i, X_{i+1}, \ldots, X_j)$, $i < j$. Time-series data is a sequence of values for these variables denoted by $x = (x_1, x_2, \ldots, x_T)$. We assume continuous values, obtained at discrete, equispaced intervals of time.

An autoregressive (AR) model of length p, is simply a p-order Markov model that imposes a linear regression for the current value of the time series given the immediate past of p previous values. That is,

$$p(x_t|x_1^{t-1}) = p(x_t|x_{t-p}^{t-1}) \sim \mathcal{N}(m + \sum_{j=1}^{p} b_j x_{t-j}, \sigma^2)$$

where $\mathcal{N}(\mu, \sigma^2)$ is a conditional normal distribution with mean μ and variance σ^2, and $\theta = (m, b_1, \ldots, b_p, \sigma^2)$ are the model parameters (e.g., [6, page 55]).

The ART models is a regime-switching generalization of the AR models, where a switching regression tree determines which AR model to apply at each time step. The autoregressors therefore have two purposes: as input for a classification that determines a particular regime, and as predictor variables in the linear regression for the specific AR model in that regime.

As a second generalization[2], ART models may allow exogenous variables, such as past observations from related time series, as regressors in the model. Time (or time-step) is a special exogenous variable, only allowed in a split condition, and is therefore only used for modeling change points in the series.

2.1 Axial and Oblique Splits

Different types of switching regression trees can be characterized by the kind of predicates they allow for splits in the tree. The ART models allow only a simple form of binary splits, where a predicate tests the value of a single regressor. The models handle continuous variables, and a split predicate is therefore of the form

$$X_i \leq c$$

where c is a constant value and X_i is any one of the regressors in the model or a variable representing time. A simple split of this type is also called *axial*, because the predicate that splits the data at a node can be considered as a hyperplane that is orthogonal to the axis for one of the regressor variables or the time variable.

[2] The class of ART models with exogenous variables has not been documented in any paper. We have learned about this generalization from communications with the authors of [18].

The best split for a node in the tree can be learned by considering all possible partitionings of the data according to each of the individual regressors in the model, and then picking the highest scoring split for these candidates according to some criterion. It can, however, be computationally demanding to evaluate scores for that many split candidates, and for that reason, [5] investigated a Gaussian quantile approach that proposes only 15 split points for each regressor. They found that this approach is competitive to the more exhaustive approach. A commercial implementation for ART models uses the Gaussian quantile approach and we will compare our alternative to this approach.

We propose a solution, which will only produce a single split candidate to be considered for the entire set of regressors. In this solution we extend the class of ART models to allow for a more general split predicate of the form

$$\sum_i a_i X_i \leq c \tag{1}$$

where the sum is over all the regressors in the model and a_i are corresponding coefficients. Splits of this type are in [20] called *oblique* due to the fact that a hyperplane that splits data according to the linear predicate is oblique with respect to the regressor axes. We will in Section 4 describe the details behind the method that we use to produce an oblique split candidate.

2.2 Motivation for Oblique Splits

There are general statistical reasons why, in many situations, oblique splits are preferable over axial splits. In fact, for a broad class of time series, the best splitting hyperplane turns out to be approximately orthogonal to the principal diagonal $d = (\frac{1}{\sqrt{p}}, \ldots, \frac{1}{\sqrt{p}})$. To qualify this fact, consider two pre-defined classes of segments $x^{(c)}, c = 1, 2$ for the time-series data x. Let $\mu^{(c)}$ and $\Sigma^{(c)}$ denote the mean vector and covariance matrix for the sample joint distribution of X_{t-p}^{t-1}, computed for observations on p regressors for targets $x_t \in x^{(c)}$.

Let us define the moving average $A_t = \frac{1}{p} \sum_{i=1}^{p} X_{t-i}$. We show in the Appendix that in the context where $X_{t-i} - A_t$ is weakly correlated with A_t, while its variance is comparable with that of A_t, the angle between the principal diagonal and one of the principal axes of $\Sigma^{(c)}, c = 1, 2$ is small. This would certainly be the case with a broad range of financial data, where increments in price curves have notoriously low correlations with price values [22,17], while seldom overwhelming the averages in magnitude. With one of the principal axes being approximately aligned with the principal diagonal d for both $\Sigma^{(1)}$ and $\Sigma^{(2)}$ it is unlikely that a cut orthogonal to either of the coordinate axes $X_{t-1}, \ldots, X_{t-p}$ can provide optimal separation of the two classes.

3 The Learning Procedure

An ART model is typically learned in a stagewise fashion. The learning process starts from the trivial model without any regressors and then greedily evaluates regressors one at a time and adds the ones that improve a chosen scoring criterion to model, while scoring criterion keeps improving.

The task of learning a specific autoregressive model considered at any stage in this process can be cast into a standard task of learning a linear regression tree. It is done by a trivial transformation of the time-series data into multivariate data cases for the regressor and target variables in the model. For example, when learning an ART model of length p with an exogenous regressor, say z_{t-q}, from a related time series, the transformation creates the set of $T - \max(p, q)$ cases of the type (x_{t-p}^{t}, z_{t-q}), where $\max(p, q) + 1 < t \leq T$. We will in the following denote this transformation as the *phase view*, due to a vague analogy to the phase trajectory in the theory of dynamical systems.

Most regression tree learning algorithms construct a tree in two stages (see, e.g., [2]): First, in a growing stage, the learning algorithm will maximize a scoring criterion by recursively trying to replace leaf nodes by better scoring splits. A least-squares deviation criterion is often used for scoring splits in a regression tree. Typically the chosen criterion will cause the selection of an overly large tree with poor generalization. In a pruning stage, the tree is therefore pruned back by greedily eliminating leaves using a second criterion–such as the holdout score on a validation data set–with the goal of minimizing the error on unseen data.

In contrast, [18] suggests a learning algorithm that uses a Bayesian scoring criterion, described in detail in that paper. This criterion avoids over-fitting by penalizing for the complexity of the model, and consequently, the pruning stage is not needed. We use this Bayesian criterion in our experimental section.

In the next section, we describe the details of the algorithm we propose for producing the candidate splits that are considered during the recursive construction of a regression tree. Going from axial to oblique splits adds complexity to the proposal of candidate splits. However, our split proposer dramatically reduces the number of proposed split candidates for the nodes evaluated during the construction of the tree, and by virtue of that fact spends much less time evaluating scores of the candidates.

4 Spectral Splitting

This section will start with a brief description of spectral clustering, followed by details about how we apply this method to produce candidate splits for an ART time-series model. A good tutorial treatment and an extensive list of references for spectral clustering can be found in [26].

The spectral splitting method that we propose constructs two types of split candidates–oblique and time–both relying on spectral clustering. Based on this clustering, the method applies two different views on the data–phase and trace–according to the type of splits we want to identify. The algorithm will only propose a *single* oblique split candidate and possibly a few time split candidates for any node evaluated during the construction of the regression tree.

4.1 Spectral Clustering

Given a set of n multi-dimensional data points $(x_1, \ldots, x_n)$, we let $a_{ij} = a(x_i, x_j)$ denote the affinity between the i'th and j'th data point, according to some symmetric and non-negative measure. The corresponding affinity matrix is denoted by$A =$

$(a_{ij})_{i,j=1,\ldots,n}$, and we let D denote the diagonal matrix with values $\sum_{j=1}^{n} a_{ij}$, $i = 1,\ldots,n$ on the diagonal.

Spectral clustering is a non-parametric clustering method that uses the pairwise proximity between data points as a basis of the criterion that the clustering must optimize. The trick in spectral clustering is to enhance the cluster properties in the data by changing the representation of the multi-dimensional data into a (possibly one-dimensional) representation based on eigenvalues for the so-called Laplacian.

$$L = D - A$$

Two different normalizations for the Laplacian have been proposed in [23] and [21], leading to two slightly different spectral clustering algorithms. We will follow a simplified version of the latter. Let I denote the identity matrix. We will cluster the data according to the second smallest eigenvector–the so-called Fiedler vector [9]–of the normalized Laplacian

$$L_{norm} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}$$

The algorithm is illustrated in Figure 1. Notice that we replace L_{norm} with

$$L'_{norm} = I - L_{norm}$$

which changes eigenvalues from λ_i to $1 - \lambda_i$ and leaves eigenvectors unchanged. We therefore find the eigenvector for the second-largest and not the second-smallest eigenvector. We prefer this interpretation of the algorithm for reasons that become clear when we discuss iterative methods for finding eigenvalues in Section 4.2.

1. Construct the matrix L'_{norm}.
2. Find the second-largest eigenvector $e = (e_1, \ldots, e_n)$ of L'_{norm}.
3. Cluster the elements in the eigenvector (e.g. by the largest gap in values).
4. Assign the original data point x_i to the cluster assigned to e_i.

Fig. 1. Simple normalized spectral clustering algorithm

Readers familiar with the original algorithm in [21] may notice the following simplifications: First, we only consider a binary clustering problem, and second, we only use the two largest eigenvectors for the clustering, and not the k largest eigenvectors in their algorithm. (The elements in the first eigenvector always have the same value and will therefore not contribute to the clustering.) Due to the second simplification, the step in their algorithm that normalizes rows of stacked eigenvectors can be avoided, because the constant nature of the first eigenvector leaves the transformation of the second eigenvector monotone.

4.2 Oblique Splits

Oblique splits are based on a particular view of the time series data that we call the *phase view*, as defined in Section 3. Importantly, a data case in the phase view involves

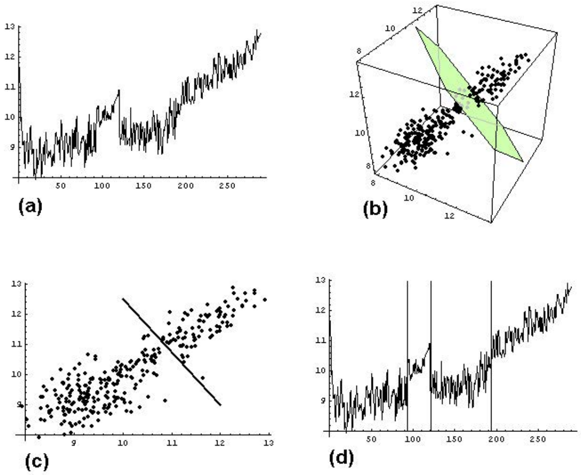

Fig. 2. Oblique split candidate for ART model with two autoregressors. (a) The original time series. (b) The spectral clustering of phase-view data. The polygon separating the upper and lower parts is a segment of a separating hyperplane for the spectral clusters (c) The phase view projection to regressor plane and the separating hyperplane learned by the perceptron algorithm. (d) The effect of the oblique split on the original time series: a regime consisting of the slightly less upward trending and more volatile first and third data segments is separated from the regime with more upward trending and less volatile second and fourth segments.

values for both the target and regressors, which imply that our oblique split proposals may capture regression structures that show up in the data–as opposed to many standard methods for axial splits that are ignorant to the target when determining split candidates for the regressors.

It should also be noted that because the phase view has no notion of time, similar patterns from entirely different segments of time may end up on the same side of an oblique split. This property can at times result in a great advantage over splitting the time series into chronological segments. First of all, splitting on time imposes a severe constraint on predictions, because splits in time restrict the prediction model to information from the segment latest in time. Information from similar segments earlier in the time series are not integrated into the prediction model in this case. Second, we may need multiple time splits to mimic the segments of one oblique split, which may not be obtainable due to the degradation of the statistical power from the smaller segments of data. Figure 2(d) shows an example, where a single oblique split separates the regime with the less upward trending and slightly more volatile first and third data segments of the time series from the regime consisting of the less volatile and more upward trending second and fourth segments. In contrast, we would have needed three time splits to properly divide the segments and these splits would therefore have resulted in four different regimes.

Our split proposer produces a single oblique split candidate in a two step procedure. In the first step, we strive to separate two modes that relates the target and regressors for the model in the best possible way. To accomplish this task, we apply the affinity based spectral clustering algorithm, described in Section 4.1, to the phase view of the time series data. For the experiments reported later in this paper, we use an affinity measure proportional to

$$\frac{1}{1 + ||p_1 - p_2||^2}$$

where $||p_1 - p_2||^2$ is the L2-norm between two phases. We do not consider exogenous regressors from related time series in these experiments. All variables in the phase view are therefore on the same scale, making the inverse distance a good measure of proximity. With exogenous regressors, more care should be taken with respect to the scaling of variables in the proximity measure, or the time series should be standardized. Figure 2(b) demonstrates the spectral clustering for the phase view of the time-series data in Figure 2(a), where this phase view has been constructed for an ART model with two autoregressors.

The oblique split predicate in (1) defines an inequality that only involves the regressors in the model. The second step of the oblique split proposer therefore projects the clustering of the phase view data to the space of the regressors, where the hyperplane separating the clusters is now constructed. While this can be done with a variety of linear discrimination methods, we decided to use a simple single-layer perceptron optimizing the total misclassification count. Such perceptron will be relatively insensitive to outliers, compared to, for example, Fisher's linear discriminant.

The computational complexity of an oblique split proposal is dominated by the cost of computing the full affinity matrix, the second largest eigenvector for the normalized Laplacian, and finding the separating hyperplane for the spectral clusters. Recall that n denotes the number of cases in the phase view of the data. The cost of computing the full affinity matrix is therefore $O(n^2)$ affinity computations. Direct methods for computing the second largest eigenvector is $O(n^3)$. A complexity of $O(n^3)$ may be prohibitive for series of substantial length. Fortunately, there are approximate iterative methods, which in practice are much faster with tolerant error. For example, the Implicitly Restarted Lanczos Method (IRLM) has complexity $O(mh + nh)$, where m is the number of nonzero affinities in the affinity matrix and h is the number of iterations required until convergence [29]. With a full affinity matrix $m = n^2$, but a significant speedup can be accomplished by only recording affinities above a certain threshold in the affinity matrix. Finally, the perceptron algorithm has complexity $O(nh)$.

4.3 Time Splits

A simple but computationally expensive way of determining a good time split is to let the split proposer nominate all possible splits in time for the further evaluation. The commercial implementation of the ART models relies on an approximation to this approach that proposes a smaller set of equispaced points on the time axis.

We suggest a data driven approximation, which will more precisely target the change points in the time series. Our approach is based on another view of the time series data that we call the *trace view*. In the trace view we use the additional time information to label the phase view data in the spectral clustering. The trace view, now traces the clustered data through time and proposes a split point each time the trace jumps across clusters. The rationale behind our approach is that data in the same cluster will behave in a similar way, and we can therefore significantly reduce the number of time-split proposals by only proposing the cluster jumps. As an example, the thin lines orthogonal to the time axis in Figure 2(d) shows the few time splits proposed by our approach. Getting close to a good approximation for the equispaced approach would have demanded far more proposed split points.

Turning now to the computational complexity. Assuming that spectral clustering has already been performed for the oblique split proposal, the additional overhead for the trace through data is $O(n)$.

5 Evaluation

In this section, we provide an empirical evaluation for our spectral splitting methods. We use a large collection of financial trading data. The collection contains the daily closing prices for 1495 stocks from Standard & Poor's 1500 index[3] as of January 1, 2008. Each time series spans across approximately 150 trading days ending on February 1, 2008. (Rotation of stocks in the S&P 1500 lead to the exclusion of 5 stocks with insuffient data.) The historic price data is available from Yahoo!, and can be downloaded with queries of format http://finance.yahoo.com/q/hp?s=SYMBOL, where SYMBOL is the symbol for the stock in the index. We divide each data set into a training set, used as input to the learning method, and a holdout set, used to evaluate the models. We use the last five observations as the holdout set, knowing that the data are daily with trading weeks of five days.

In our experiments, we learn ART models with an arbitrary number of autoregressors and we allow time as an exogenous split variable. We do not complicate the experiments with the use of exogenous regressors from related time series, as this complication is irrelevant to the objective for this paper. For all the models that we learn, we use the same Bayesian scoring criterion, the same greedy search strategy for finding the number of autoregressors, and the same method for constructing a regression tree – except that different alternative split candidates are considered for the different splitting algorithms that we consider.

We evaluate two different types of splitting with respect to the autoregressors in the model: *AxialGaussian* and *ObliqueSpectral*. The AxialGaussian method is the standard method used to propose multiple axial candidates for each split in an ART model, as described in Section 2.1. The ObliqueSpectral method is our proposed method, which for a split considers only a single oblique candidate involving all regressors. In combination with the two split proposer methods for autoregressors, we also evaluate three types of time splitting: *NoSplit*, *Fixed*, and *TimeSpectral*. The NoSplit method does not allow any time splits. The Fixed method is the simple standard method for learning splits on time in an ART model, as described in Section 4.3. The TimeSpectral method is our spectral clustering-based alternative. In order to provide context for the numbers in the evaluation of these methods, we will also evaluate a very weak baseline, namely the method not allowing any splits. We call this method the *Baseline* method.

We evaluate the quality of a learned model by computing the *sequential predictive score* for the holdout data set corresponding to the training data from which the model

[3] `standardandpoors.com`

was learned. The sequential predictive score for a model is simply the average log-likelihood obtained by a one-step forecast for each of the observations in the holdout set. To evaluate the quality of a learning method, we compute the average of the sequential predictive scores obtained for each of the time series in the collection. Note that the use of the log-likelihood to measure performance simultaneously evaluates both the accuracy of the estimate and the accuracy of the uncertainty of the estimate. Finally, we use a (one-sided) sign test to evaluate if one method is significantly better than another. To form the sign test, we count the number of times one method improves the predictive score over the other for each individual time series in the collection. Excluding ties, we seek to reject the hypothesis of equality, where the test statistic for the sign test follows a binomial distribution with probability parameter 0.5.

6 Results

To make sure that the results reported here are not an artifact of sub-optimal axial splitting for the AxialGaussian method, we first verified the claim from [5] that the Gaussian quantiles is a sufficient substitute for the exhaustive set of possible axial splits. We compared the sequential predictive scores on 10% of the time series in our collection and did not find a significant difference.

Table 1 shows the average sequential predictive scores across the series in our collection for each combination of autoregressor and time-split proposer methods. First of all, for splits on autoregressors, we see a large improvement in score with our ObliqueSpectral method over the standard AxialGaussian method. Even with the weak baseline–namely the method not allowing any splits–the relative improvement from AxialGaussian to ObliqueSpectral over the improvement from the baseline to AxialGaussian is still above 20%, which is quite impressive.

The fractions in Table 2 report the number of times one method has higher score than another method for all the time series in our collection. Notice that the numbers in a fraction do not necessarily sum to 1495, because we are not counting ties. We particularly

Table 1. Average sequential predictive scores for each combination of autoregressor and time split proposer methods

Regressor splits	Time splits	Ave. score
Baseline	Baseline	-3.07
AxialGaussian	NoSplit	-1.73
AxialGaussian	Fixed	-1.72
AxialGaussian	TimeSpectral	-1.74
ObliqueSpectral	NoSplit	-1.45
ObliqueSpectral	Fixed	-1.46
ObliqueSpectral	TimeSpectral	-1.44

Table 2. Pairwise comparisons of sequential predictive scores. The fractions show the number of time series, where one method has higher score than the other. The column labels denote the autoregressor split proposers being compared.

	Baseline / AxialGaussian	Baseline / ObliqueSpectral	AxialGaussian / ObliqueSpectral
NoSplit	118 / 959	74 / 1168	462 / 615
Fixed	114 / 990	79 / 1182	226 / 418
SpectralTime	122 / 955	71 / 1171	473 / 604

notice that the ObliqueSpectral method is significantly better than the standard AxialGaussian method for all three combinations with time-split proposer methods. In fact, the sign test rejects the hypothesis of equality at a significance level $< 10^{-5}$ in all cases. Combining the results from Tables 1 and 2, we can conclude that the large improvement in the sequential predictive scores for our ObliqueSpectral method over the standard AxialGaussian method is due to a general trend in scores across individual time series, and not just a few outliers.

We now turn to the surprising observation that adding time-split proposals to either of the AxialGaussian and the ObliqueSpectral autoregressor proposals does not improve the quality over models learned without time splits–neither for the Fixed nor the TimeSpectral method. Apparently, the axial and oblique splitting on autoregressors are flexible enough to cover the time splits in our analysis. We do not necessarily expect this finding to generalize beyond series that behave like stock data, due to the fact that it is a relatively easy exercise to construct an artificial example that will challenge this finding.

Finally, the oblique splits proposed by our method involve *all* regressors in a model, and therefore rely on our spectral splitting method to be smart enough to ignore noise that might be introduced by irrelevant regressors. Although efficient, such parsimonious split proposal may appear overly restrictive compared to the possibility of proposing split candidates for all possible subsets of regressors. However, an additional set of experiments have shown that the exhaustive approach in general only leads to insignificant improvements in predictive scores. We conjecture that the stagewise inclusion of regressors in the overall learning procedure for an ART model (see Section 3) is a main reason for irrelevant regressors to not pose much of a problem for our approach.

7 Conclusions and Future Work

We have presented a method for building regime-switching trees for nonstationary time series. The method is based on geometric clustering. More specifically, spectral clustering has been used in this paper. As such, our method does not rely on any parametric assumptions with regards to the distributions that best describe individual regimes. The clustering-based split proposer is used to propose a single oblique split candidate at each node level in the switching tree, which makes the method computationally efficient.

In the evaluation part of the paper we limited ourselves to an extension of ART models that are built under the assumption of uncorrelated Gaussian error. The joint target-regressor distribution for a regime-switching time series can be modeled as a mixture of Gaussians in this case, and we were able to motivate and then prove empirically that oblique splits are better at learning the mixtures than combinations of axial splits. In fact, the experimental evidence we have collected shows that our approach when used to extend the ART models, dramatically improves predictive accuracy over the current approach. We still experimented under the assumption of Gaussianity. An important future experiment should allow non-Gaussian models in the oblique switching trees.

The focus in this paper has been on learning regime-switching time-series models that will easily lend themselves to explanatory analysis and interpretation. In future experiments we also plan to evaluate the potential tradeoff in modularity, interpretability, and computational efficiency with forecast precision for our simple learning approach compared to more complicated approaches that integrates learning of soft regime switching and the local regimes in the models, such as the learning of Markov-switching (e.g., [12,13]) and gated experts (e.g., [27,28]) models.

References

1. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group, Belmont, California (1984)
3. Brodley, C.E., Utgoff, P.E.: Multivariate decision trees. Machine Learning 19(1), 45–77 (1995)
4. Chaudhuri, P., Huang, M., Loh, W.Y., Yao, R.: Piecewise polynomial regression trees. Statistica Sinica 4, 143–167 (1994)
5. Chickering, D., Meek, C., Rounthwaite, R.: Efficient determination of dynamic split points in a decision tree. In: Proc. of the 2001 IEEE International Conference on Data Mining, pp. 91–98. IEEE Computer Society (November 2001)
6. DeGroot, M.: Optimal Statistical Decisions. McGraw-Hill, New York (1970)
7. Dobra, A., Gehrke, J.: Secret: A scalable linear regression tree algorithm. In: Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 481–487. ACM Press (2002)
8. Donath, W.E., Hoffman, A.J.: Lower bounds for the partitioning of graphs. IBM Journal of Research and Development 17, 420–425 (1973)
9. Fiedler, M.: Algebraic connectivity of graphs. Czechoslovak Mathematical Journal 23, 298–305 (1973)
10. Gama, J.: Oblique linear tree. In: Proc. of the Second International Symposium on Intelligent Data Analysis, pp. 187–198 (1997)
11. Hamilton, J.D.: Time Series Analysis. Princeton University Press, Princeton (1994)
12. Hamilton, J.D.: A new approach to the economic analysis of nonstationary time series and the business cycle. Econometrica 57(2), 357–384 (1989)
13. Hamilton, J.D.: Analysis of time series subject to changes in regime. Journal of Econometrics 45, 39–70 (1990)
14. Iyengar, V.S.: Hot: Heuristics for oblique trees. In: Proc. of the 11th IEEE International Conference on Tools with Artificial Intelligence, pp. 91–98. IEEE Computer Society, Washington, DC (1999)

15. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Neural Computation 6, 181–214 (1994)
16. Li, K.C., Lue, H.H., Chen, C.H.: Interactive tree-structured regression via principal Hessian directions. Journal of the American Statistical Association 95, 547–560 (2000)
17. Mandelbrot, B.: Forecasts of future prices, unbiased markets, and martingale models. Journal of Business 39, 242–255 (1966)
18. Meek, C., Chickering, D.M., Heckerman, D.: Autoregressive tree models for time-series analysis. In: Proc. of the Second International SIAM Conference on Data Mining, pp. 229–244. SIAM (April 2002)
19. Meilă, M., Shi, J.: Learning segmentation by random walks. In: Advances in Neural Information Processing Systems 13, pp. 873–879. MIT Press (2001)
20. Murthy, S.K., Kasif, S., Salzberg, S.: A system for induction of oblique decision trees. Journal of Artificial Intelligence Research 2, 1–32 (1994)
21. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14, pp. 849–856. MIT Press (2002)
22. Samuelson, P.: Proof that properly anticipated prices fluctuate randomly. Industrial Management Review 6, 41–49 (1965)
23. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
24. Tong, H.: Threshold models in non-linear time series analysis. Lecture Notes in Statistics, vol. 21. Springer (1983)
25. Tong, H., Lim, K.S.: Threshold autoregression, limit cycles and cyclical data- with discussion. Journal of the Royal Statistical Society, Series B 42(3), 245–292 (1980)
26. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
27. Waterhouse, S., Robinson, A.: Non-linear prediction of acoustic vectors using hierarchical mixtures of experts. In: Advances in Neural Information Processing Systems 7, pp. 835–842. MIT Press (1995)
28. Weigend, A.S., Mangeas, M., Srivastava, A.N.: Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. International Journal of Neural Systems 6(4), 373–399 (1995)
29. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: Proc. of the 5th SIAM International Conference on Data Mining. SIAM (2005)

Appendix

Lemma 1. *Let Σ be a non-singular sample auto-covariance matrix for X_{t-p}^{t-1} defined on the p-dimensional space with principal diagonal direction $d = (\frac{1}{\sqrt{p}}, \ldots, \frac{1}{\sqrt{p}})$, and let $A_t = \frac{1}{p}\sum_{i=1}^{p} X_{t-i}$. Then*

$$\sin^2(\Sigma d, d) = \frac{\sum_{i=1}^{p} cov(X_{t-i} - A_t, A_t)^2}{\sum_{i=1}^{p} cov(X_{t-i}, A_t)^2}. \tag{2}$$

Proof. Introduce $S_t = \sum_{i=1}^{p} X_{t-i}$. As per bi-linear property of covariance, $(\Sigma d)_i = \frac{1}{\sqrt{p}} cov(X_{t-i}, S_t), i = 1, \ldots, p$ and $(\Sigma d)d = \frac{1}{p}\sum_{i=1}^{p} cov(X_{t-i}, S_t) = cov(A_t, S_t)$. Non-singularity of Σ implies that the vector $\Sigma d \neq 0$. Hence, $|\Sigma d|^2 \neq 0$ and

$$\cos^2(\varSigma d, d) = \frac{((\varSigma d)d)^2}{|(\varSigma d)|^2} = \frac{p\, cov(A_t, S_t)^2}{\sum_{i=1}^{p} cov(X_{t-i}, S_t)^2}.$$

It follows that

$$\begin{aligned}
&\sin^2(\varSigma d, d)\\
&= 1 - \cos^2(\varSigma d, d)\\
&= \frac{p\left(\frac{1}{p}\sum_{i=1}^{p} cov(X_{t-i}, S_t)^2 - cov(A_t, S_t)^2\right)}{\sum_{i=1}^{p} cov(X_{t-i}, S_t)^2}\\
&= \frac{\sum_{i=1}^{p} cov(X_{t-i} - A_t, S_t)^2}{\sum_{i=1}^{p} cov(X_{t-i}, S_t)^2}
\end{aligned}$$

Dividing the numerator and denominator of the last fraction by p^2 amounts to replacing S_t by A_t, which concludes the proof. □

Corollary 1. *When $X_{t-i} - A_t$ and A_t are weakly correlated, and the variance of $X_{t-i} - A_t$ is comparable to that of A_t, $i = 1, \dots, p$, then $\sin^2(\varSigma d, d)$ is small.*

Specifically, let σ and ρ denote respectively standard deviation and correlation, and introduce $\varDelta_i = \frac{cov(X_{t-i} - A_t, A_t)}{\sigma(A_t)} = \rho(X_{t-i} - A_t, A_t)\sigma(X_{t-i} - A_t)$. We quantify both assumptions in Corollary 1 by positing that $|\varDelta_i| < \epsilon\sigma(A_t), i = 1, \dots, p$, where $0 < \epsilon \ll 1$. Easy algebra on Equation (2) yields

$$\begin{aligned}
\sin^2(\varSigma d, d) &= \frac{\varSigma \varDelta_i^2}{\varSigma(\sigma(A_t) + \varDelta_i)^2}\\
&< \frac{p\epsilon^2\sigma(A_t)^2}{p(1-\epsilon)^2\sigma(A_t)^2}\\
&= \frac{\epsilon^2}{(1-\epsilon)^2} \qquad (3)
\end{aligned}$$

Under the assumptions of Corollary 1, we can now show that d is geometrically close to an eigenvector of $\varSigma$. Indeed, by inserting (3) into the Pythagorean identity we derive that $|\cos(\varSigma d, d)| > \frac{\sqrt{1-2\epsilon}}{1-\epsilon}$ and close to 1. Now, given a vector v for which $|v| = 1$, $|\cos(\varSigma v, v)|$ reaches the maximum of 1 iff v is an eigenvector of $\varSigma$. When the eigenvalues of $\varSigma$ are distinct, d must therefore be at a small angle with one of the p principal axes for $\varSigma$.