Long-memory Time Series Ensembles for Concept Shift Detection

Marcelo Mendoza
Department of Informatics
Universidad Técnica Federico
Santa María
Santiago, Chile
marcelo.mendoza@usm.cl

Felipe Bravo-Marquez
PRISMA Research Group
Department of Computer
Science, University of Chile
Yahoo! Labs Santiago
Santiago, Chile
fbravo@dcc.uchile.cl

Barbara Poblete
PRISMA Research Group
Department of Computer
Science, University of Chile
Yahoo! Labs Santiago
Santiago, Chile
bpoblete@dcc.uchile.cl

Daniel Gayo-Avello
Department of Informatics
University of Oviedo
Oviedo, Spain
dgayo@uniovi.es

ABSTRACT

Usually time series are controlled by generative processes which display changes over time. On many occasions, two or more generative processes may switch forcing the abrupt replacement of a fitted time series model by another one. We claim that the incorporation of past data can be useful in the presence of concept shift. We believe that history tends to repeat itself and from time to time, it is desirable to discard recent data reusing old past data to perform model fitting and forecasting. We address this challenge by introducing an ensemble method that deals with long-memory time series. Our method starts by segmenting historical time series data to identify data segments which present model consistency. Then, we project the time series by using data segments which are close to current data. By using a dynamic time warping alignment function, we try to anticipate concept shifts, looking for similarities between current data and the prequel of a past shift. We evaluate our proposal on non-stationary and non-linear time series. To achieve this we perform forecasting accuracy testing against well known state-of-the-art methods such as neural networks and threshold auto regressive models. Our results show that the proposed method anticipates many concept shifts.

Categories and Subject Descriptors: H.2.8 [Information Systems]: Database Application - *Data Mining*

General Terms: Algorithms, Theory, Measurement.

Keywords: Long-term forecasting, streaming.

BigMine'13, August 11-14 2013, Chicago, IL, USA
Copyright 2013 ACM 978-1-4503-2324-6/13/08 $15.00.
http://dx.doi.org/10.1145/2501221.2501225.

1. INTRODUCTION

Learning is the ability to modify current behavior by using information about the past. Time series models allow for an artificial system to analyze new sequential data to automatically project it into the future. This can help people make decisions for possible future scenarios.

Time series models are inductive, therefore they require that detected data structures are consistent with current data and also with future unseen data records. Thus, a main challenge for learning is the generalization ability of the proposed model.

Usually, time series are controlled by dynamic generative processes. Indeed, in many situations, time series are governed by two or more overlapping switching generative processes. In these scenarios, in order to fit new incoming data, an abrupt replacement of one model by a new model is needed. Progressive data changes (or concept drifts) may be incrementally incorporated into time series models in a natural fashion. In fact, time series models can be recomputed using sliding windows, replacing old data records with new ones. However, regular data changes can lead to stability problems. On the other hand, abrupt data changes (or concept shifts) may require the discarding of recent data, forcing to fit a new model. In general, we refer this problem as the stability-plasticity dilemma [13], which is the ability to learn new information without discarding previously acquired knowledge.

Time series models can cope with several data aspects. For stationary data, ARMA models [4] are able to perform accurate forecasts. Sometimes, non-stationary data displays stationarity when the original time series is differenced. For this kind of data ARIMA models are appropriate [4]. However, previous models are limited in their effectiveness if the data shows non-linearity. A number of strategies address this problem by using neural networks [21], which are able to accurately fit non-linear data. Finally, sometimes data can be fitted by modeling two or more data regimes which switch from one to another according to a set of estimated data thresholds. These models are known as threshold auto regressive models [14]. In any case, the stability-plasticity dilemma is targeted using an incremental learning strategy that aim to learn from the data sequence by taking snapshots of the evolving data. This approach is focused on learning the latest snapshots and considers that a single evolving model can explain the data generation process. The learning goal of this method is to keep an updated version of the model.

In the machine learning domain, the stability-plasticity dilemma is addressed by using ensemble-based methods [17]. These methods work on labeled data and attempt to identify past data folds which are consistent with current data. In this direction, splitting historical data into windows of fixed length is a common practice. Then, a weak learner is created from each data chunk. Here the model selected for the sequel is the one that has the lowest error for the current data. We follow this idea to design our proposal.

In time series modeling ensemble-based approaches have to deal with an important challenge: a uniform data segmentation strategy can be orthogonal to how multiple generative processes interact to produce the observed data. Thus, the direct application of ensemble techniques to time series can be unsuccessful. In order to deal with this problem, we propose the construction of time series ensembles by segmenting past data into segments which offer model consistency, i.e. segments that allow the detection of a time series model with good fitting properties. We explore the use of a top-down strategy for the time series segmentation step, creating a time series tree representation. Then, a set of data segments are recovered from the time series tree by following a bottom-up merge procedure. Finally, we conduct model fitting on current data and its closest past data segments. The distance between time series segments is implemented by a dynamic time warping distance function that detects the best alignment between segment-pairs. At this point we study two segment matching strategies: a) *Closest segment*, which merges current data with the closest past data segment, and b) *One segment ahead*, which selects the following model consistent data segment, trying to anticipate the change in the underlying data process.

The remainder of the paper is organized as follows. In Section 2 we summarize related work. In Section 3 we introduce our method. Experimental results are shown in Section 4. Finally we conclude in Section 5.

2. RELATED WORK

The idea of considering multiple generative processes in time series has been studied before for classification tasks [10]. To achieve an autonomous system able to distinguish between sober vs. drunk drivers, a memory based times series procedure was developed. The authors considered that time series are generated from different underlying mechanisms. Therefore, classification algorithms were used to recognize the class of the target time series. The recognition procedure consists of collecting time series samples from the different classes and fitting the α and β parameters of the corresponding ARMA model. The ARMA coefficients are used to build a knowledge representation able to recognize the class of unlabeled time series. To classify unseen time series, their ARMA coefficients are estimated and compared to past series using a distance measure. Finally, time series are classified according to the class of the nearest member of the knowledge memory.

The combination of different forecasting models was reviewed in [9]. According to this work, the combination of multiple individual prediction models leads to increased forecasts accuracy. Basically, when different forecasts are combined, the risk of forecast errors is diversified.

Ensemble models for forecasting purposes were proposed, among others, in [16, 24, 8]. In [16], Kim et al. proposed a genetic fuzzy predictor ensemble for predicting non-stationary time series. The method extracts several fuzzy IF-THEN rules from input-out pairs of the time series together with memberships functions aimed at reducing the prediction error. Rule bases are evolved using genetic algorithms and afterwards the membership functions are evolved in the same manner. In order to enhance the predictability of each rule, an ensemble of rules that combines genetically the resulting fuzzy predictors is created by an equal prediction-error weighting method. Experimental results showed that the predictability of the ensemble of rules outperforms the predictability of one single fuzzy predictor. In [24] the Aggregated Forecast Through Exponential Re-weighting algorithm (AFTER) was proposed. The authors argue that the uncertainty in finding an accurate forecasting model can be reduced by combining different models. In the algorithm, different ARIMA models were combined using a weighting scheme. The weights were sequentially updated according to the past performances of their respective models. Empirical results showed that the AFTER algorithm reduces the error when there is difficulty in identifying the best model. A nonlinear neural network ensemble model was proposed in [18] for financial time series forecasting. Different neural networks models were generated and selected using principal component analysis. The ensemble is then constructed using a support vector machine regression. In [3] a number of recurrent neural networks were trained on different data examples and combined using a boosting based algorithm. In [8] a two level ensemble learning approach for time series prediction was developed using radial basis function networks, k-nearest neighbor and self organizing maps. The ensemble model is able to detect regularities in non-stationary time series and achieves a better performance than the individual models.

Besides the ensemble approach discussed below, the combination of models can also be formed over a set of training sets. Due to the fact than in forecasting problems the training procedure is applied over one single training set, the dataset can be replicated using bootstrap. The bootstrap aggregating or bagging approach refers to the idea of combining different forecasts trained over the bootstrap-replicated training set [6]. Kilian and Inoue addressed the problem of having a number of relevant predictors that individually have weak explanatory power [15]. They used a bagging approach to perform forecasts from multiple regression models. The main argument of the authors is that bagging is a suitable approach for situations in which predictors are moderately large relative to the sample size. In [19] the bagging approach was extended to time series using asymmetric cost functions for predicting signs and quantiles. Unlike financial returns which may not be predicted, their variance, sign, and quantiles may be predictable. Results obtained showed that bagging may improve the binary prediction in small samples.

3. TIME SERIES ENSEMBLE METHOD

In this section we introduce our forecasting framework, which is divided into three parts: 1) A top-down segmentation strategy which identifies past data segments with good model consistency properties, 2) A bottom-up merge segment strategy which address the parsimony/representativeness tradeoff, and 3) An alignment strategy that looks for the closest past segments.

3.1 The Top-down segmentation strategy

Our method, which has the goal of identifying meaningful past data sequences for forecasting purposes, starts by creating a tree representation of the historical time series sequences. To do this we follow a top-down segmentation strategy which looks for the best partition of the sequence according to a given model selection criteria. The partition algorithm evaluates each sequence split by measuring the likelihood of the model fitting procedure to both segments. The best split is the one which maximize the log likelihood of the fitted models. Then, the two time series sub sequences generated by the partition algorithm are segmented by recursive calls to the top-down algorithm. The following procedures implement our strategy.

```
$\mathtt{TOP} - \mathtt{DOWN}(\mathtt{S}, \mathtt{i}, \mathtt{j}, L)$
if $\mathtt{j} > \mathtt{i}$ then
    $\mathtt{p} \leftarrow \mathtt{PARTITION}(\mathtt{S}[\mathtt{i}, \mathtt{j}])$
    $\mathtt{pivot} \leftarrow \mathtt{i} + \mathtt{p} - 1$
    if $\mathtt{length}(\mathtt{S}[\mathtt{i}, \mathtt{pivot}]) \geq L$ then
        $\mathtt{TOP} - \mathtt{DOWN}(\mathtt{S}[\mathtt{i}, \mathtt{pivot}], \mathtt{i}, \mathtt{pivot}, L)$
    end if
    if $\mathtt{length}(\mathtt{S}[\mathtt{pivot} + 1, \mathtt{j}]) \geq L$ then
        $\mathtt{TOP} - \mathtt{DOWN}(\mathtt{S}[\mathtt{pivot} + 1, \mathtt{j}], \mathtt{pivot} + 1, \mathtt{j}, L)$
    end if
end if

$\mathtt{PARTITION}(\mathtt{S})$
$\mathtt{ML} \leftarrow -10^6$
for $i = 2$ to $\mathtt{length}(\mathtt{S}) - 1$ do
    $\mathtt{model1} \leftarrow \mathtt{ModelFitting}(\mathtt{S}[1, \mathtt{i}])$
    $\mathtt{model2} \leftarrow \mathtt{ModelFitting}(\mathtt{S}[\mathtt{i} + 1, \mathtt{length}(\mathtt{S})])$
    $\mathtt{L}_1 \leftarrow \mathtt{length}(\mathtt{S}[1, \mathtt{i}])$
    $\mathtt{L}_2 \leftarrow \mathtt{length}(\mathtt{S}[\mathtt{i} + 1, \mathtt{length}(\mathtt{S})])$
    $\mathtt{w}_1 \leftarrow \frac{\mathtt{L}_1}{\mathtt{L}_1 + \mathtt{L}_2}$
    $\mathtt{w}_2 \leftarrow \frac{\mathtt{L}_2}{\mathtt{L}_1 + \mathtt{L}_2}$
    $\mathtt{Lik} \leftarrow \mathtt{w}_1 \cdot \mathtt{model1.Lik} + \mathtt{w}_2 \cdot \mathtt{model2.Lik}$
    if $\mathtt{Lik} > \mathtt{ML}$ then
        $\mathtt{pivot} \leftarrow \mathtt{i}$
        $\mathtt{ML} \leftarrow \mathtt{Lik}$
    end if
end for
return $\mathtt{pivot}$
```

To segment the entire time series $\mathtt{S}$ the initial call is as follows: $\mathtt{TOP} - \mathtt{DOWN}(\mathtt{S}, 1, \mathtt{length}(\mathtt{S}), L)$. We introduce a minimum length parameter L to avoid the split of extremely short time series segments.

The key to the algorithm is the $\mathtt{PARTITION}$ procedure, which splits each segment according to a model consistency criteria. We use the maximum likelihood identification strategy proposed by Akaike [1] as a model consistency criteria. The maximum likelihood estimation is a well known time series model detection method used to select ARIMA models, which allows the identification of data segments which display good model fitting properties.

A family of ARIMA models may be evaluated for each segment by using step-wise algorithms, a strategy which helps in the search of model coefficients which was proposed by Hyndman and Khandakar [20]. Then, $\mathtt{PARTITION}$ evaluates the quality of each split $\mathtt{S}[1, \mathtt{i}]$ and $\mathtt{S}[\mathtt{i} + 1, \mathtt{length}(\mathtt{S})]$ for each value of the index $\mathtt{i}$ in $\{2, \mathtt{length}(\mathtt{S}) - 1\}$. After that, the best ARIMA model is fitted to each time series segment and the likelihood of each model is retrieved. A global likelihood measure $\mathtt{Lik}$ for the split is calculated by weighting each model likelihood by the time series segment length, penalizing by the length the model fitting of extremely short segments. Finally, the position of the data point around which the maximum global likelihood is reached is returned to the $\mathtt{TOP} - \mathtt{DOWN}$ algorithm, which recursively split the segments around this element. We call this element the $\mathtt{pivot}$ of the sequence. $\mathtt{TOP} - \mathtt{DOWN}$ loops while the end and start sequence index elements $\mathtt{i}$ and $\mathtt{j}$ satisfy the condition $\mathtt{j} > \mathtt{i}$.

We use a concrete example to explain how our segmentation strategy works. A tree structure produced by our strategy on a real-world time series is shown in Figure 1. The lower figure shows the political perception about a given 2008 U.S. presidential election candidate. Each time series point represents daily polarity perceptions. A total of 120 points are represented in the time series which corresponds to the first four months of the campaign. The upper figure shows the tree produced by our top-down segmentation procedure, where the vertical axis represents the likelihood of each time series split.

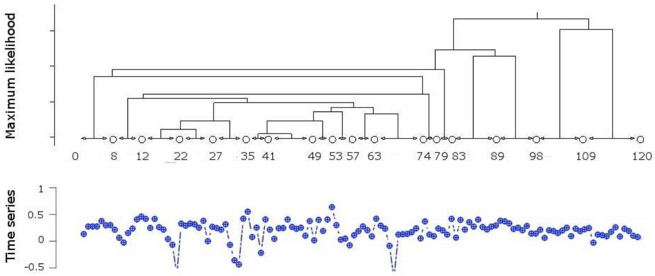

Figure 1: Tree structure produced by our top-down segmentation strategy over a 2008 U.S. presidential election opinion time series.

The tree produced by our procedure provides a way to visualize how different subsequent time series segments are. The closer the split is to the tree leaves, the more consistent the segments are for a model fitting procedure. For example, the segments between data points 35-41 and 41-49 are very close each other, suggesting that a unique model can be fitted in the interval 35-49. On the other hand, the segments between 98-109 and 109-120 were splitted almost at the top of the tree, indicating that two models are possibly needed to get a better description of the 98-120 segment.

We use a data model consistency criteria which is based on a collection of ARIMA modeling properties. However, a common limitation of using ARIMA modeling is that model selection can imply a significant number of potential candidate models. In this scenario, an exhaustive evaluation of every candidate model is unpractical. The state-of-the-art registers several efforts towards addressing the ARIMA model selection task. These methods attempt to automate model selection procedures. We consider a general $\mathtt{ARIMA}(\mathtt{p}, \mathtt{d}, \mathtt{q})$ model for stationary series of the $\mathtt{d}$-*th* order difference, given by the following expression:

$$Y_t^d = \phi_1 Y_{t-1}^d + ... + \phi_p Y_{t-p}^d + e_t - \theta_1 e_{t-1} - ... - \theta_q e_{t-q},$$

where the variables $\{e_t\}$ are a sequence of unobserved and independent zero mean white noise random variables and Y_t is the target variable. In the case of seasonal data, a generalized seasonal ARIMA model $\mathtt{ARIMA}(\mathtt{p}, \mathtt{d}, \mathtt{q}) \times (\mathtt{P}, \mathtt{D}, \mathtt{Q})_{\mathtt{m}}$ may be selected, where $\mathtt{m}$ is the seasonal frequency. The model selection task corresponds to the estimation of the values $\mathtt{p}, \mathtt{q}, \mathtt{P}, \mathtt{Q}, \mathtt{d}, \mathtt{D}$, and $\mathtt{m}$. We do this by following the method proposed by Hyndman and Khandakar [20], which is based on the diffuse prior approach introduced by Durbin and Koopman [12]. The diffuse prior approach starts by choosing proper difference values $\mathtt{d}$ and $\mathtt{D}$, allowing a posterior proper likelihood comparison between models of the same order. To select the difference orders of the model, the model selection method conducts a set of unit-root tests [11] for a null hypothesis of no unit-root. A set of successive unit-root tests are conducted for an increasing sequence of $\mathtt{d}$ values until a first insignificant result is achieved. For seasonal data, a Canova-Hansen test [7] is conducted to find a proper value for the seasonal frequency. Then, if $\mathtt{m} > 1$ (i.e. the displays a relevant seasonal component), the difference order of the seasonal component is also determined by following a set of unit-root tests. Finally, an exhaustive grid search is conducted to select the values of $\mathtt{p}$, $\mathtt{q}$ and possibly $\mathtt{P}$ and $\mathtt{Q}$, if $\mathtt{D} \neq 0$, by minimizing the Akaike information criteria [1] (AIC), defined by the following expression:

$$\mathtt{AIC} = -2\lg(\mathtt{L}) + 2\left(\mathtt{p} + \mathtt{q} + \mathtt{P} + \mathtt{Q} + \mathtt{k}\right), \tag{1}$$

where $\mathtt{k} = 1$ if $\mathtt{d}$ or $\mathtt{D}$ are non zero values, and $\mathtt{L}$ is the maximum likelihood of the fitted model. Notice that the first part of the AIC criteria represents the maximum likelihood model selection criteria, and the second part is a penalizing factor which helps select parsimonious models, i.e. to avoid to choose over parameterized models.

Globally, our partition procedure selects the data point around which the likelihood of both data segments is maximized. Nevertheless, our strategy attempts to also address the tradeoff of splitting extremely short data sequences, where very simple models may be fitted, decreasing the second factor of Eq. (1). To achieve a good balance between data representativeness and model parsimony, we calculate a global fitness function which is a weighted convex combination of the likelihood of both segments and their lengths, described as follows:

$$\mathtt{Lik} = \mathtt{w}_1 \cdot \mathtt{L}(\mathtt{M}_1) + \mathtt{w}_2 \cdot \mathtt{L}(\mathtt{M}_2), \tag{2}$$

where $\mathtt{w}_1$ and $\mathtt{w}_2$ are given by $\frac{L_1}{L_1+L_2}$ and $\frac{L_2}{L_1+L_2}$, respectively. Notice that $\mathtt{M}_1$ and $\mathtt{M}_2$ are the models which optimize the AIC criteria for a given pair of segments S_1 and S_2, where the length of each segments is $\mathtt{L}_1$ and $\mathtt{L}_2$, respectively. Thus, our split criteria fits the Akaike optimum model for each segment (local optimization of likelihood and parsimony) but the split is oriented by the maximum likelihood criteria, which is weighted by each segment length. Thus, Eq. (2) can be expressed as a minimum-maximum split criterion function, given by the following expression:

$$\begin{aligned}\underset{t}{\mathrm{Max}} \quad & \Big\{\mathtt{w}_1 \cdot \mathtt{L}\big(\underset{\mathtt{i}}{\mathrm{Min}}\,\mathtt{AIC}(\mathtt{M}_\mathtt{i} \models \mathtt{S}_{1,\ldots,t})\big) \\ + \quad & \mathtt{w}_2 \cdot \mathtt{L}\big(\underset{\mathtt{j}}{\mathrm{Min}}\,\mathtt{AIC}(\mathtt{M}_\mathtt{j} \models \mathtt{S}_{t+1,\ldots,\mathtt{N}})\big)\Big\}. \end{aligned} \tag{3}$$

3.2 The Bottom-up merge strategy

Despite the fact that the criterion function we have just described is designed to achieve a good balance between data representativeness (i.e. segment length) and model fitting (i.e. likelihood and parsimony), we cannot ensure that the tree structure produced by our top-down strategy may lead to extremely simple short segments. To some extent this situation is controlled by the split criterion function, which in practice, tends to avoid over segmentation by the inclusion of the segment length in the split criterion function defined in Eq. 3. We also use a parameter which defines a minimum segment length (denoted by L) to be considered by the partition procedure (see lines 4 and 7 in the $\mathtt{TOP}-\mathtt{DOWN}$ procedure). But our procedure cannot avoid the partition of a given segment of length L into two very dissimilar parts. Indeed, the use of the Akaike criterion function may lead to very simplistic models because it includes a parsimony component.

We address this situation by defining an automatic bottom-up strategy which verifies the results produced by the top-down segmentation procedure, correcting and avoiding anomalous splits. We fix the case where a naive model with one coefficient was selected (i.e. the model considers only the bias component) by merging this segment with its closest segment, fitting a new model to the composition of both segments. If the new model has at least two coefficients (i.e. is at least an AR(1) or MA(1)) the merge procedure stops, otherwise the merge procedure is repeated. We illustrate our bottom-up merge strategy in Figure 2.

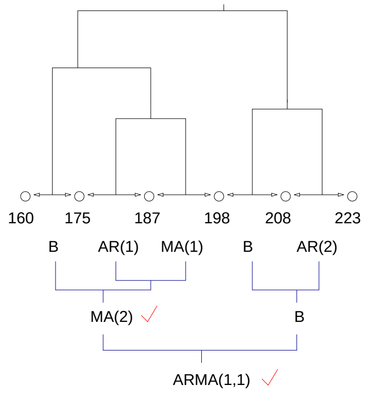

Figure 2: The bottom-up merge strategy.

The segments 160-175 and 198-208 are fitted by a couple of biased one component models (denoted by B). Then, the bottom-up strategy merges the segment 160-175 with 175-198 (note that this segment is as well composed of a couple of segments fitted with models of two coefficients) producing a $\mathtt{MA}(2)$ model. As this new model has three coefficients (the bias coefficient and the two coefficients defined by the second order moving average model) the procedure stops. To fix the 198-208 case, the bottom-up strategy merges this segment with the one included between the values 208-223. However, the new model generated is also a biased one, forcing the merge of the segment 198-223 with the one included between the values 160-198. As the new model fitted is an $\mathtt{ARMA}(1,1)$, the procedure stops.

The bottom-up strategy ensures that every time series segment will be explained by a model, discarding over segmented data. In the previous example, the bottom-up strategy withdraw the initial five segments by replacing them with only one model.

3.3 Forecasting: Looking for similar data sequences in the past

We index each time series segment identified by our top-down / bottom-up segmentation strategy in an ensemble. Based on the available history of the time series up to a given time point t_0, we would like to forecast the value of $Y_{t_{0+h}}$, where h is the number of steps ahead, also known as hops. In practice, we are interested in the projection of m hops ahead, where m is a seasonal regular frequency of interest.

We consider two baseline forecasting cases, which explore forecasting accuracy under two dissimilar scenarios: A short-memory scenario, where the historical data is discarded for forecasting purposes, and a long-memory scenario, where all the available data is considered for forecasting.

Short-memory forecasting.

The data that needs to be projected is composed of the last m observed data points. A short memory strategy will forecast the series by using only the current data, fitting a model on the last m data points and projecting them m steps ahead, discarding past data for forecasting purposes.

Long-memory forecasting.

We consider the complete set of past data as the current data to be forecasted. Then, we fit a model over the entire observed time series, projecting it m steps ahead.

We explore the feasibility of two new strategies, first *closest segment*, which adds to current data the closest historical data segment. Second, we consider the strategy *one-segment ahead*, which adds current data to the segment whose previous segment is closest to the current data.

Closest segment forecasting.

The data to be projected is the current data and the historical segment which is closest to the last past m observed data points. The composition of both segments is forecasted m steps ahead.

One-segment ahead forecasting.

The data to be projected is the current data and the historical segment whose previous segment is closest to the last m observed data points. The composition of both segments is forecasted m steps ahead.

We describe each forecasting strategy in Figure 3.

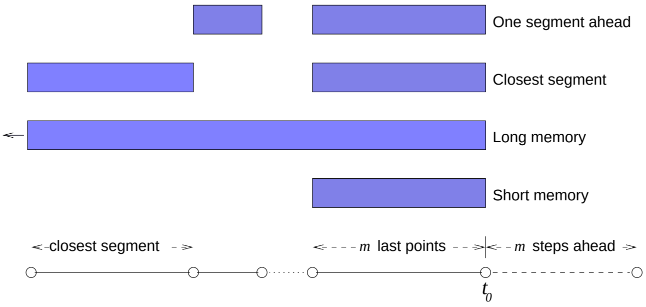

Figure 3: The forecasting strategies.

Closest segment and *one segment ahead* require a distance function evaluation between the current data and each indexed historical segment. We use a flexible way to determine a distance measure between two sequences. As both sequences may be out-of-phase, we use an alignment procedure to find the optimal alignment between both sequences according to a given distance function. This procedure known as *Dynamic Time Warping (DTW)* was first used to match out-of-phase audio signals. Despite the fact that the *DTW* performance may be deteriorated by outliers, we use it due to its alignment flexibility. As a distance function we consider the Euclidean distance.

Closest segment looks for the most similar historical sequence to expanding the current data for which the model will be fitted and forecasted. The rationale is that the current data generative process can be better described if we add current data with similar past data. As data generative processes may change frequently, creating time series segments of short time span, data in a single segment represents a single occurrence of an underlying data generative process. Thus, merging current data and its closest segment can help the reconstruction of a better model.

Closest segment works well only if the underlying data generative process does not change in the next m steps. One way to anticipate a change is to aggregate current data with the historical data segment which is contiguous to the closest segment. Since our time series segmentation is oriented towards model consistency, the historical segment ahead can be an occurrence of a different data generative process. Thus, merging current data with the segment ahead of the closest segment can facilitate the detection of a change point.

4. EXPERIMENTAL RESULTS

We conduct experiments on different types of time series to test the abilities of the proposed forecasting strategies under different scenarios. The forecasting performance of the proposed strategies is studied using both synthetic and real data time series.

4.1 Time series

We use the following time series in our experiments:

Synthetic control chart time series.

This dataset contains a number of time series synthetically generated by changing data processes described by Alcock and Manopoulos [2]. These series corresponds to different classes of control charts. We recover the series classified in the upward shift and downward shift classes. Each series has 60 samples [1]. We use the series numbered as 401 and 501, that will be denoted in this section as `Upward` and `Downward`.

Prey-predator time series.

These experimental time series represent the number of protozoan per ml measured every twelve hours over a period of 35 days. The series were registered in the experiments of Veilleux [22] about the population fluctuation of a prey-predator system, whereas the prey is *Paramecium aurelia* and whereas the predator species is *Didinium natsutum* [2]. The initial part of the data is non-stationary. We use both series of 70 samples for each one, denoted in this section as `Paramecium` and `Didinium`.

River flow.

This series shows the monthly river flow for the Iowa River measured at Wapello, Iowa, for the period September 1958 through August 2006. It has 576 data samples. The flow was measured in cubic feet per second. Some parts of the data show non-linearity. In terms of history, this is the longest time series considered in our experiments [3]. We use the complete series, denoted in this section as `flow`.

US elections 2008.

These series show daily perceptions of candidates of the 2008 US presidential election, Barack Obama and John McCain. The series were generated by processing public Twitter messages and by inferring opinion polarity of users regarding each entity according to a sentiment analysis method based on a lexicon [23]. These series were introduced in [5] and show the polarity measured in the United States during the period of June 2008 through November 2008. We denote each series as `Obama` and `McCain`.

[1] See: `http://archive.ics.uci.edu/ml/databases/`
[2] See: `http://www.buseco.monash.edu.au/~hyndman`
[3] See: `http://waterdata.usgs.gov/ia/nwis/sw`

4.2 Methods

We study the use of different models for the forecasting step. As a base model we consider the ones achieved by the ARIMA model selection procedure proposed by Hyndman and Khandakar [20]. In addition, we investigate the use of neural networks for forecasting [21], which have shown advantages to fit non linear data. Finally, we also investigate the use of threshold autoregressive models [14] for forecasting purposes, which have shown good properties for switching data processes.

Neural Networks.

We explore the performance of the "Neural network nonlinear autoregressive model" for forecasting, which was proposed by Tong [21]. This model considers a single hidden layer, possibly with skip-layer connections and a linear output. The model is estimated using a BFGS optimization method. We denote this method in the results section as NNET.

Self exciting threshold autoregressive model.

We explore the performance of the "Self exciting threshold autoregressive model (SETAR)", which was proposed by Geweke and Terui [14]. This model allow more flexibility by considering a regime switching behavior, where the switch from one regime to another is triggered by a threshold which is conditioned to past values. We consider two regimes and a threshold which is searched over a grid of feasible values. The AR order of the lower and upper regimes is equal to 1, considering a delay parameter equals 3.

We use the neural networks implementation provided in the NNET R package, version 7.3-1 [4]. The SETAR method is included in the TSDYN R package, version 0.8.1 [5]. Finally, the ARIMA model selection procedure is provided in the FORECAST R Package, version 3.21 [6].

4.3 Results on synthetic data

In a first experiment we explore and test the forecasting capabilities of the proposed methods with the Upward and Downward synthetic time series introduced in subsection 4.1. We denote the methods by LM (long memory), SM (short memory), CS (closest segment), and AHEAD (one segment ahead).

We start the evaluation with the Upward and Downward series. We do this by considering the first 40 points as the historical data, and the subsequent 20 data points as the future unseen points. We consider forecast windows of 5 steps ahead, that is to say, $t_0 = 40$, and four forecasts windows S[41 : 45], S[46 : 50], S[45 : 55], and S[56 : 60]. For example, the forecasts for the window S[41 : 45] consider that the current data is the sequence S[36 : 40], for the window S[46 : 50] the current data is S[41 : 45] and so on. The initial segmentation over the sequence S[0 : 40] is considered for every forecasted window. Finally, we compare the real data $S[41 : 60]$ with the forecasted data obtained by using each of the discussed methods. Tables 1 and 2 show the accuracy results for the Upward and Downward series, respectively.

Table 1: Accuracy results for the Upward time series

	ME	RMSE	MAE	MPE	MAPE
LM + ARIMA	-1.33	5.43	4.79	-4	12
LM + NNET	3.91	5.21	4.11	9	9
LM + SETAR	0.18	5.67	4.74	0	12
SM + ARIMA	-0.19	4.17	3.61	-1	9
SM + NNET	-0.73	6.19	5.31	-3	13
SM + SETAR	6.44	7.09	6.44	15	15
CS + ARIMA	-0.83	5.59	4.84	-3	12
CS + NNET	7.74	8.73	7.74	18	18
CS + SETAR	8.01	8.54	8.01	19	19
AHEAD + ARIMA	-0.61	4.18	3.51	-2	9
AHEAD + NNET	-0.72	4.10	3.52	-2	9
AHEAD + SETAR	-2.38	4.81	4.26	-7	11

Table 2: Accuracy results for the Downward time series

	ME	RMSE	MAE	MPE	MAPE
LM + ARIMA	-1.1	3.57	3.14	-14	26
LM + NNET	-8.36	10.21	9.01	-68	72
LM + SETAR	-0.93	3.71	3.51	-17	33
SM + ARIMA	-0.36	4.21	3.76	-10	30
SM + NNET	-1.26	3.41	2.81	-14	23
SM + SETAR	-2.04	3.9	3.48	-22	30
CS + ARIMA	-0.45	3.79	3.42	-11	28
CS + NNET	-0.73	3.41	2.68	-12	22
CS + SETAR	0.66	3.93	3.43	-2	25
AHEAD + ARIMA	-1.17	3.07	3.02	-12	25
AHEAD + NNET	-1.57	3.7	3.2	-18	27
AHEAD + SETAR	3.63	5.24	4.23	19	24

Table 1 shows that the short memory method is very competitive with the one segment ahead strategy. We can observe that the neural networks and the setar methods based on the one segment ahead are better than the long memory and the short memory, and in particular, these outcomes significantly outperform the results achieved by the closest segment strategy. These results illustrate the fact that Upward is a short memory time series but shows changes that are properly detected by the one segment ahead strategy. Something similar occurs with the Downward time series. As Table 2 shows, the best results are achieved by the closest segment and one ahead segment strategies, with a couple of very interesting results achieved by neural networks and setar, showing that Downward presents non linearities.

4.4 Results on real world data

We start the analysis by evaluating the performance of the methods in the Paramecium and Didinium series. We fix t_0 at 50, reserving the last 20 samples for forecasting purposes. We consider forecasted windows of 5 samples, that is to say, we fix the number of hops $m = 5$. As these series have 70 samples, we obtain 4 forecasted windows in the interval between 51-70. We show the accuracy results for these series in Tables 3 and 4.

Table 3 shows that the best results are achieved by the long memory strategy with a setar based forecasting model. These results are very close to the ones achieved by the one segment ahead, showing in this case that the use of ARIMA models may be improved by replacing the long memory with the one segment ahead strategy. In the case of Didinium, in Table 4, we note that the results achieved by the long memory setar and the one segment ahead ARIMA strategies are very close. These results illustrate that as both series have a short history, the impact of the use of past data is limited. However, one segment ahead shows again, good results.

We evaluate the accuracy performance of flow by fixing $t_0 = 504$. Then, the remainder 72 data samples were reserved for forecasting purposes, considering projections of 12 hops ahead due the

[4] See: http://www.stats.ox.ac.uk/pub/MASS4

[5] See: http://tsdyn.googlecode.com

[6] See: http://robjhyndman.com/software/forecast

Table 3: Accuracy results for the Paramecium time series

	ME	RMSE	MAE	MPE	MAPE
LM + ARIMA	21.01	35.82	29.81	19	46
LM + NNET	8.98	26.79	21.36	-3	37
LM + SETAR	11.95	20.96	15.21	10	21
SM + ARIMA	16.47	50.84	45.52	-18	83
SM + NNET	6.48	49.59	47.02	-32	96
SM + SETAR	5.97	34.69	26.75	-12	49
CS + ARIMA	26.17	40.05	34.98	24	51
CS + NNET	13.21	37.64	32.66	-3	60
CS + SETAR	3.11	24.44	18.01	-1	27
AHEAD + ARIMA	10.48	19.78	15.79	6	25
AHEAD + NNET	15.49	33.63	28.14	6	51
AHEAD + SETAR	-4.06	37.32	24.16	-10	37

Table 4: Accuracy results for the Didinium time series

	ME	RMSE	MAE	MPE	MAPE
LM + ARIMA	-2.14	72.71	45.21	-32	47
LM + NNET	2.61	90.38	71.05	-58	87
LM + SETAR	5.31	65.61	40.08	-18	35
SM + ARIMA	-63.91	182.53	143.9	-155	203
SM + NNET	-4.19	121.61	84.67	-72	112
SM + SETAR	-5.38	109.35	77.21	-46	84
CS + ARIMA	-45.69	148.78	101.8	-130	152
CS + NNET	13.58	98.97	74.36	-49	86
CS + SETAR	-62.61	171.15	111.7	-156	176
AHEAD + ARIMA	5.59	68.91	38.81	-16	35
AHEAD + NNET	-7.08	90.30	73.83	-66	93
AHEAD + SETAR	-20.19	94.79	69.78	-62	81

fact that the data presents an annual seasonal frequency. As this series has 576 samples, we obtain six forecasted windows, in the interval between 505 through 576. We show the results in Table 5.

Table 5: Accuracy results for the flow time series

	ME	RMSE	MAE	MPE	MAPE
LM + ARIMA	-1817	6489	5302	-102	120
LM + NNET	-1441	6784	5421	-103	125
LM + SETAR	-1661	7437	6097	-123	144
SM + ARIMA	-1604	6498	5262	-101	122
SM + NNET	-2443	7326	6134	-139	154
SM + SETAR	-2006	8018	6748	-131	151
CS + ARIMA	2120	6987	4833	-20	74
CS + NNET	-742	6480	5062	-83	106
CS + SETAR	-1536	6670	5336	-101	119
AHEAD + ARIMA	-914	6232	4879	-84	105
AHEAD + NNET	-2005	7275	5995	-126	144
AHEAD + SETAR	-1255	7118	5706	-106	127

As Table 5 shows, the best results are achieved by the closest segment strategy combined with the ARIMA model. In fact, the model selection procedure detects seasonality, which explains why in this case the use of a seasonal ARIMA offers good performance results. We can observe that by replacing the long memory with the closest segment strategy, the ARIMA accuracy achieves significant improvements (around 40 MAPE points and 80 MPE points) which in fact is a very remarkable result. This fact indicates to us that flow is a long memory series, but the discarding of some past data periods favors the identification of better models for forecasting purposes.

Finally, for the Obama and McCain series we fixed $t_0 = 120$. that is to say, we consider the first four months as the historical data, projecting the series until the 156-th day. We consider as a frequency of interest 7 hops. As the series have 156 samples, we obtain 5 forecasted windows in the interval between 121 through 156. Accuracy results are shown in Tables 6 and 7.

Table 6: Accuracy results for the Obama time series

	ME	RMSE	MAE	MPE	MAPE
LM + ARIMA	0.005	0.069	0.051	-4	18
LM + NNET	0.008	0.078	0.061	-4	20
LM + SETAR	-0.009	0.077	0.054	-9	20
SM + ARIMA	0.013	0.068	0.048	-5	17
SM + NNET	-0.011	0.071	0.051	-7	19
SM + SETAR	-0.016	0.081	0.058	-12	22
CS + ARIMA	-0.012	0.072	0.048	-10	18
CS + NNET	0.007	0.071	0.051	-4	18
CS + SETAR	0.002	0.072	0.055	-5	19
AHEAD + ARIMA	-0.015	0.051	0.039	-5	12
AHEAD + NNET	-0.014	0.052	0.041	-5	12
AHEAD + SETAR	-0.011	0.059	0.045	-4	12

Table 7: Accuracy results for the McCain time series

	ME	RMSE	MAE	MPE	MAPE
LM + ARIMA	-0.044	0.127	0.101	188	304
LM + NNET	0.009	0.227	0.142	196	299
LM + SETAR	-0.025	0.126	0.095	190	279
SM + ARIMA	-0.017	0.129	0.121	216	299
SM + NNET	-0.009	0.114	0.083	190	252
SM + SETAR	-0.006	0.233	0.142	216	232
CS + ARIMA	0.012	0.125	0.098	188	233
CS + NNET	-0.055	0.181	0.136	231	337
CS + SETAR	-0.033	0.231	0.141	160	222
AHEAD + ARIMA	-0.008	0.097	0.046	122	142
AHEAD + NNET	-0.009	0.091	0.115	130	153
AHEAD + SETAR	0.024	0.253	0.122	177	216

Table 6 shows that the results achieved for the Obama series are very precise for every method evaluated. In this case the best results are achieved by the one segment ahead strategy, but these results in fact are very close to the ones achieved by the other methods, illustrating that this series shows low variability. On the other hand, the McCain series shows a very significant noise level, which explain that the quality of the results is more limited. The results show that the use of a one ahead segment strategy also offer good results, illustrating that a short memory strategy may discard relevant past data and a long memory strategy may include irrelevant past data achieving the one segment ahead strategy a good balance between both scenarios.

5. CONCLUSION

We propose two new forecasting strategies which consider past data to conduct a model selection procedure. We evaluate the proposed strategies against two baselines, a long memory strategy which considers the whole history for model selection, and a short memory strategy, which discards past data. We evaluated our methods by combining them with three different models: ARIMA, neural networks and SETAR. We evaluate our methods considering non linear/linear data, non stationary/stationary data, short/long time data, and seasonal/non seasonal data. In every case, the use of selective memory offers benefits against a short or long memory strategy. In particular our results show that the use of the one segment ahead strategy outperforms many of the evaluated methods, showing its abilities for change detection and forecasting.

Currently, we are extending our proposal to properly work with high frequency data, allowing to us the evaluation of the proposed methods in financial time series. Preliminar results show that this framework may offers improvements for forecasting purposes.

6. ACKNOWLEDGMENT

This work has been partially supported by FONDEF DO9I1185. Marcelo Mendoza was supported by project FONDECYT grant 11121435. Felipe Bravo-Marquez was supported by a CONICYT Master Scholarship. Barbara Poblete was supported by FONDECYT grant 11121511 and U-INICIA VID 2012, grant 3/0612, University of Chile.

7. REFERENCES

[1] H. Akaike. Maximum likelihood identification of Gaussian autoregressive moving average models. *Biometrika*, 60:255–265, 1973.

[2] R. J. Alcock, Y. Manolopoulos, D. E. Laboratory, and D. O. Informatics. Time-series similarity queries employing a feature-based approach. In *In 7 th Hellenic Conference on Informatics, Ioannina*, pages 27–29, 1999.

[3] M. Assaad, R. BonÃl', and H. Cardot. A new boosting algorithm for improved time-series forecasting with recurrent neural networks. *Information Fusion*, 9(1):41 – 55, 2008. Special Issue on Applications of Ensemble Methods.

[4] G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 3rd edition, 1994.

[5] F. Bravo-Marquez, D. Gayo-Avello, M. Mendoza, and B. Poblete. Opinion dynamics of elections in twitter. In *LA-WEB*, pages 32–39. IEEE Computer Society, 2012.

[6] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996. 10.1023/A:1018054314350.

[7] F. Canova and B. E. Hansen. Are seasonal patterns constant over time? a test for seasonal stability. *Journal of Business & Economic Statistics*, 13(3):237–52, July 1995.

[8] A. Chitra and S. Uma. An ensemble model of multiple classifiers for time series prediction. *International Journal of Computer Theory and Engineering*, 2(3):1793–8201, June 2010.

[9] R. T. Clemen. Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, 5(4):559–583, 1989.

[10] K. Deng, A. W. Moore, and M. C. Nechyba. Learning to recognize time series: Combining arma models with memory-based learning. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, CIRA '97, pages 246–, Washington, DC, USA, 1997. IEEE Computer Society.

[11] D. A. Dickey and W. A. Fuller. Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica*, 49(4):1057–72, 1981.

[12] J. Durbin and S. J. Koopman. *Time series analysis by state space methods*, volume 24. Oxford University Press, 2001.

[13] S. Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1(1):17–61, 1988.

[14] N. T. J. Geweke. Bayesian threshold autoregressive models for nonlinear time series. *Journal of Time Series Analysis*, 14:441–454, 1993.

[15] L. Kilian and A. Inoue. Bagging time series models. Econometric Society 2004 North American Summer Meetings 110, Econometric Society, Aug. 2004.

[16] D. Kim and C. Kim. Forecasting time series with genetic fuzzy predictor ensemble. *IEEE Trans. Fuzzy Syst*, 5:523–535, 1997.

[17] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[18] K. Lai, L. Yu, S. Wang, and H. Wei. A novel nonlinear neural network ensemble model for financial time series forecasting. In V. Alexandrov, G. van Albada, P. Sloot, and J. Dongarra, editors, *Computational Science âĂŞ ICCS 2006*, volume 3991 of *Lecture Notes in Computer Science*, pages 790–793. Springer Berlin / Heidelberg, 2006.

[19] T.-H. Lee and Y. Yang. Bagging binary and quantile predictors for time series. *Journal of Econometrics*, 135(1-2):465–497, 00 2006.

[20] C.-S. Ong, J.-J. Huang, and G.-H. Tzeng. Model identification of arima family using genetic algorithms. *Applied Mathematics and Computation*, 164(3):885 – 912, 2005.

[21] H. Tong. *Non-Linear Time Series: A Dynamical System Approach (Oxford Statistical Science Series, 6)*. Oxford University Press (UK), July 1993.

[22] B. G. Veilleux. An analysis of the predatory interaction between paramecium and didinium. *Journal of Animal Ecology*, 48(3):787–803, 1979.

[23] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[24] H. Zou and Y. Yang. Combining time series models for forecasting. *International Journal of Forecasting*, 20(1):69–84, 2004.