# Developing a Bayesian method for locating breakpoints in time series data.

**Kathryn Haglich [1], Jeffrey Liebner [1], Sarah Neitzel [2] and Amy Pitts [3]**

[1] Department of Mathematics, Lafayette College, Easton, Pennsylvania, USA

[2] School of Biodiversity Conservation, Unity College, Unity, Maine, USA

[3] Department of Mathematics, Marist College, Poughkeepsie New York, USA

---

**Address for correspondence:** Jeffrey Liebner, Department of Mathematics, Lafayette College, Easton, Pennsylvania, USA.

**E-mail:** `liebnerj@lafayette.edu`.

**Phone:** (+420) 221 913 282.

**Fax:** (+1) 999 888 666.

---

**Abstract:** Our paper proposes a new approach to finding the quantity and location of breakpoints, or change points, in time series data. This allows for more appropriate data modeling by accounting for structural changes. Bayesian Adaptive Auto-Regression (BAAR) is a Bayesian technique that samples from the distribution of number and locations of possible breakpoints. It proposes new sets of breakpoints as determined by a reversible-jump Markov Chain Monte Carlo and evaluates the proposals using the Metropolis-Hastings algorithm. Simulation results have shown

that our method is able to detect changes in models, and we have provided a demonstration of BAAR as applied to the population of Pacific brown pelicans.

---

**Key words:** Breakpoints; Time Series; Bayesian; AR; BARS

# 1 Introduction

When modeling time series data, it is helpful to identify significant places of change in the model, which we will refer to as breakpoints. By identifying breakpoints, different parts of the data can be fitted with different, more appropriate models, allowing for noteworthy changes to be reflected and identified in the model as a whole. Therefore, it is imperative to find the amount and location of where these breakpoints occur. Our paper focuses on this problem and proposes an accurate and efficient method of finding the quantity and location of breakpoints in time series data.

Since breakpoints are found in numerous types of time series data sets, there has been ample interest in recent years to develop techniques to find the number and placement of breakpoints. The techniques being applicable to many fields. Applications of existing techniques include the analysis of United States Treasury bill rates (Peseran et. al. 2006) and climate records (Ruggieri 2013). The simplest technique relies on expert opinion: the breakpoint locations are approximated by experts in the specific field of the data set based on historical knowledge. This method was used by Seidel and Lanzante (2004) to find breakpoints in global atmospheric temperature data as well as Gamber, Liebner, and Smith (2016) who applied it to CPI data sets. To reduce human-error, more formulaic computational methods have been developed,

one such being the Reverse Order Cusum (ROC) (Pesaran and Timmermann, 2002). This method reverses the data set and uses historical knowledge to group data points together with the boundaries being the locations of the breakpoints. A more notable technique is the Bai-Perron test (Bai and Perron 1998, 2003). To handle the issues that structural changes in data pose for running regression, Bai and Perron developed a general algorithm to find an optimal breakpoint set (1998, 2003). From this algorithm, Zeileis et. al. developed the R package "breakpoints" in strucchange to implement the Bai-Perron test (2007).

In our paper, the Bayesian Adaptive Auto-Regression (BAAR) method develops a Bayesian procedure to find the distribution of the number and location of breakpoints in time series data. BAAR is inspired by Bayesian Adaptive Regression Splines (BARS), a Bayesian curve fitting with free knot splines developed by DiMatteo et al. (2001) and implemented by Walstrom, Liebner, and Kass (2008). This method has also been adaptive for linear regressions known as Bayesian adaptive Linear Regression (BALR). Section two address how we approach the problem of finding the number and location of breakpoints focusing on our Metropolis-Hastings and Markov chain Monte Carlo (MCMC) algorithms to obtain the distributions. In the third section, discusses the simulation results and show that our method works. In the fourth section, we take a look at applications and how our method finds significant breakpoints in data such as Brown Pelican Population. Finally, we will discuss the importance of BAAR on finding the number and location of time series data sets as well as future applications.

# 2 Method

The Bayesian Adaptive Auto-Regression (BAAR) technique is a Bayesian method to find the location and number of breakpoint in time series. The foundation of this method is inspired on the BARS method which consists of the Metropolis Hastings algorithm containing a Markov Chain Monte Carlo (MCMC) (DiMatteo et al., 2001). The Metropolis Hastings algorithm is used to sample from a distribution when direct sampling is difficult. For this project we want to sample the distribution of $\theta = \{K, \tau_1, \ldots, \tau_K\}$ given $g(\theta|x_i, \ldots, x_n)$ where $K$, and $\tau$ are the number and location of breakpoints, given our data $x$ with $n$ observations. The repeated stochastic process of state changes in our MCMC include the birth, the death, and the moving of a breakpoints. A new breakpoint set is proposed at each step of the MCMC, and the Metropolis Hastings ratio determines the set's acceptance. From this, a distribution of possible breakpoint locations can be obtained. For this process to work the $x$ values in the data need to be placed at equal intervals. This method was written and tested repeatedly in R and RStudio (R Core Team, 2017).

## 2.1 Initial Breakpoint

The BAAR function needs to have an input starting breakpoint place(s). In the BARS paper we see that having a more intelligent start location, like with the logsplines starting condition, can significantly reduce burn it periods and run times (DiMatteo et al., 2001). This is opposed to starting with a single middle breakpoint or evenly placed breakpoints. With this knowledge we are taking the Bai-Perron method and breakpoint package to help obtain relatively good initial breakpoints (Bai, Perron, 2003) (Zeileis et al 2007). The algorithm described by Bai and Perron is a frequentist

approach that checks almost every single location for a breakpoint and returns the optimal set (Bai, Perron, 2003). The breakpoint package requires a user to specify an maximum number of breakpoint (Zeileis et al 2007). The larger the maximum number the longer the run time for the function. Based off simulations run with different initial conditions, we recommend using Bai-Perron constrained to finding a maximum of 2 breakpoints combined with a generous burn-in period of 2 times the number of data points (see **Section 3.2**).

## 2.2 Step Type

The MCMC for the BAAR method has three different possible steps: birth, death, and move. The birth step randomly proposes a breakpoint at an available location. An available location is where a breakpoint could be placed given the following constraints. First, the location cannot have a breakpoint or an endpoint currently assigned to it. Second, for the AR(1) or linear models, the location must be at least two data points away from the next adjacent breakpoints. For AR(p) models, the minimum distance away a location must be from its closest breakpoints is $2p$. If a location is in accordance with these constrains, then it is an available location.

The death step randomly chooses an existing breakpoint and proposes a set without that chosen breakpoint. Other birth and death algorithms based on distances between breakpoints at a given iteration of the MCMC were considered. However, numerous simulations have shown that the above mentioned birth and death steps are significantly superior than the experimental functions. We intend to explore other algorithms for the birth and death steps in future research.

The general move step is a subtraction step followed immediately by an addition

step and can be broken down to two specific types of move: jump and jiggle. Jump allows the movement of a breakpoint to any available location. Jiggle restricts the distance a breakpoint can move to a jiggle neighborhood, an interval surrounding the breakpoint's original location. To calculate the jiggle neighborhood, $J_n$,

$$J_n = (x_b - \rho n, x_b + \rho n)$$

where $x_b$ is the original location of the chosen breakpoint, $n$ is the size of the data set, and $\rho$ is the user-inputed percent in decimal form, to determine what percent of the total data should be in the jiggle neighborhood. When a move step is chosen, there is a $\zeta$ probability that a jiggle will be performed, which is determined by the user such that $0 < \zeta < 1$ and $\zeta \in \mathbb{Q}$. The probability of a jump occurring is $1 - \zeta$. Based off of data obtained by simulations on different probabilities the default probabilities are 75% jiggle and 25% jump. This combination increases overall speed and a combination of jiggle and jump more thoroughly explores the distribution.

## 2.3 Probabilities of the Steps

The combined probabilities of performing a birth step, $b_k$, and a death step, $d_k$, are equal to the user imputed value, $c$ such that $c \in \mathbb{Q}$ and $0 < c < 1$. The ratio of birth steps to death steps is determined by $c$, the initial conditions of the starting number of breakpoints, $K_{start}$, and the starting number of available spaces, $A_{start}$. From this, the following equations can be derived for $b_k$ and $d_k$:

$$b_k = c\,\frac{A_{start}}{A_{start} + K_{start} + 1} \qquad d_k = c\,\frac{K_{start} + 1}{A_{start} + K_{start} + 1}$$

Then, the probability of a specific birth step $b$, given $A$ available locations, is the equation

$$b = \ b_k \times \frac{1}{A}.$$

Thus, the probability of a specific death step $d$, given $K$ breakpoints, is the equation

$$d = \ d_k \times \frac{1}{K}.$$

The probability of a move step is $1 - c$. The probability of jiggle, $jg_k$, and the probability of jump, $ju_k$, are calculated by the following equations:

$$ju_k = \zeta(1 - (d_k + b_k)) \quad jg_k = (1 - \zeta)(1 - (d_k + b_k))$$

Then, the probability of a specific jump step, $ju$, given the number of breakpoints from the old breakpoint set $K_o$ and total available spaces $A$, is

$$ju = ju_k \frac{1}{K_o A}.$$

For the jiggle step, $jg$, the probability of a specific step occurring, given the number of breakpoints from the old breakpoint set $K_o$ and the spot available in the jiggle neighborhood $J_n$, is

$$jg = jg_k \frac{1}{K_o J_n},$$

These probabilities were chosen in such a way were detailed balance holds. This proof can be found in detail in appendix 1 and was inspired by DiMatteo and colleges (2001).

### *2.3.1 Metropolis Hastings Ratio and BIC Approximation*

After a specific step is selected, the Metropolis Hastings ratio, as derived below, is used to determine the acceptance of the proposed breakpoint set. To determine

the thresh hold of acceptance, $r_{unif}$ is generated from a uniform distribution from a sample space on the interval (0,1). If the ratio is greater than $r_{unif}$, then the proposed breakpoint set is accepted and kept. Otherwise, the old breakpoint set is retained.

The general Metropolis Hastings ratio is the product of the Bayes factor, determined by the ratio of the posteriors, $g$, and the ratio of the Markov Chain Monte Carlo (MCMC) proposal densities, $q$, whose values depend on the current MCMC step.

$$ratio = \frac{g(\tau_n K_n | x_1, \ldots, x_t)}{g(\tau_o K_o | x_1, \ldots, x_t)} \times \frac{q(\tau_o K_o | \tau_n K_n)}{q(\tau_n K_n | \tau_o K_o)}$$

When the log likelihood of the equation is taken,

$$\begin{aligned} log(ratio) = &\Big[log\big[g(\tau_n K_n | x_1, \ldots, x_t)\big] - log\big[g(\tau_o K_o | x_1, \ldots, x_t)\big]\Big] \\ &+ \Big[log\big[q(\tau_o K_o | \tau_n K_n)\big] - log\big[q(\tau_n K_n | \tau_o K_o)\big]\Big] \end{aligned}$$

As shown by Kass and Wasserman (1995), the log of the Bayes Factor can be approximated with BIC with an error on the order of $O(n^{-1/2})$ when the data size is greater than 25 and the prior follows a normal distribution. Therefore,

$$log\big[g(\tau_n K_n | x_1, \ldots, x_t)\big] - log\big[g(\tau_o K_o | x_1, \ldots, x_t)\big] \approx \frac{-\Delta BIC}{2}$$

which means that

$$log(ratio) \approx \frac{-\Delta BIC}{2} + \Big[log\big[q(\tau_o K_o | \tau_n K_n)\big] - log\big[q(\tau_n K_n | \tau_o K_o)\big]\Big].$$

Then we have,

$$log(ratio) \approx \Big(\frac{-\Delta BIC}{2}\Big) \frac{\pi(\tau_n, K_n)}{\pi(\tau_o, K_o)} \frac{q(\tau_o K_o | \tau_n K_n)}{q(\tau_n K_n | \tau_o K_o)}$$

In the case of a birth step,

$$q(\tau_o K_o | \tau_n K_n) = d \cdot Poisson(K_{old}, \lambda), \quad q(\tau_n K_n | \tau_o K_o) = b \cdot Poisson(K_{old}, \lambda).$$

In the case of a death step,

$$q(\tau_o K_o|\tau_n K_n) = b \cdot Poisson(K_{new}, \lambda), \quad q(\tau_n K_n|\tau_o K_o) = d \cdot Poisson(K_{old}, \lambda).$$

Where $\lambda$ is has default value of one. This means that we are assuming there is one breakpoint, the user can change this value based on preferences.

In the case of a move step, irrelevant of whether it is specifically jiggle or jump,

$$log\big[q(\tau_o K_o|\tau_n K_n)\big] - log\big[q(\tau_n K_n|\tau_o K_o)\big] = 0$$

Henceforth, for a move,

$$log(ratio) \approx \frac{-\Delta BIC}{2}$$

## 2.4 AR model

Once a step has been completed and a new breakpoint set is proposed then the data is fit using an auto-regressive model. A general format given,

$$AR(p) = \beta_0 + \beta_1 Y_{t-1} + \ldots \beta_p Y_{t-p} + \epsilon_t$$

where $p$ is a user specified degree of the AR model, time $t$, and $\epsilon$ the error term. An AR model is fit for both the old breakpoint set as well as the new proposed breakpoint set. Using breakpoint set the data is sliced into chunks and an autoregressive models is used to fit the data. The degree of the AR model, $p$, is specified by the user, the default setting is one. When AR model is fit the data that is returned is the log likelihood. The log likelihood information, $w$, is then used to find the $\Delta BIC$ for both the new and old breakpoint sets. Then we have,

$$\Delta BIC = \frac{-2w_n + log(n) \cdot (K_n + 1) \cdot (3 + p)}{-2w_o + log(n) \cdot (K_o + 1) \cdot (3 + p)}$$

With $K+1$, for both old and new, representing the number of subsections the breakpoint set creates. The $(K+1)(3+p)$ penalizes for the dimensionality per breakpoint section.

For the Bayesian adaptive linear regression (BALR) method instead of using linear regression to obtain log-likelihood information simple linear regression are used. The $\beta$ and $\sigma$ draws information would also be obtained from a linear model rather than an autoregressive model.

After the log likelihood, $w$, and ratio are approximated and either the new or old breakpoint set is chosen then $\beta$ coefficients and $\sigma$ values are obtained by drawing for their distributions.

## 2.5 Derivations of $\beta$ and $\sigma$ draws

In a similar manner done by Peseran et. al. (2006), the posterior draws for both $\beta$ and $\sigma$ are derived as followed. The posterior for the $\beta$ coefficients is

$$\beta|\sigma^2, b_0, B_0, v_0, d_0, S_t, Y_t \sim N(\overline{\beta_j}, \overline{V_j})$$

where

$$\overline{V}_j = (\sigma^{-2}X^TX + B_0^{-1})^{-1}, \quad \overline{\beta}_j = \overline{V}_j(\sigma^{-2}X^TY_t + B_0^{-1}b_0).$$

The conditions are the following: $v_0$ and $d_0$ are the parameters of the inverse gamma prior, which is the inverse gamma squared, (one being the shape the other rate), $S_t$ is the current breakpoint set, and $Y_t$ is the actual data values. For this project the prior for $\beta$ is obtained by $\pi(\beta) \sim N(MLE_\beta, \Phi_\beta)$ such that $\Phi = n \cdot I^{-1}$ and $I$ = observed fisher information matrix (Kass and Wasserman, 1995). The unit information prior

is the combination of both $B_0$ and $b_0$. Thus we get that $b_0$ is the mean of the $\beta$ coefficients, $B_0$ is the variance co-variance matrix of the $\beta$ coefficients for the prior, $\pi(\beta)$.

Pesaran and colleges (2006) also derives the $\sigma$ posterior such that

$$\sigma_j^{-2} \sim \Gamma(v_0, d_0) \longrightarrow \sigma_j^{-2}|\beta, b_0, B_0, v_0, d_0, S_t, Y_t \sim \Gamma(\overline{v}_0, \overline{d}_0)$$

where

$$\overline{v}_0 = v_0 + \frac{n_j}{2}, \quad \overline{d}_0 = d_0 + \frac{1}{2}(Y_t - X\beta)^T(Y_t - X\beta).$$

For this project the prior of $\sigma$ is simply $\pi(\sigma) = \frac{1}{\sigma^2}$ and the likelihood function is just a multivariant normal distribution. Both of these were inspired by the the paper written by Kass and Wasserman (1995).

# 3 Results

## 3.1 Simulated Data Run

To show that the BAAR method does work simulated data was created with two clear breaks (Figure 1). Using simulated data will help ensure that the method created places breakpoints in correct locations. The breakpoint locations are $t = 100$ and $t = 200$ for the simulated data used. The method was run in R with 10,000 iterations, a burn in period of 1,500, along with a jump jiggle probabilities of 25% jump and 75% jiggle and initial conditions obtain from Bai-Perron test constrained to 2.From this data was obtained and figure 2 and figure 3 were created. Figure 2 depicts the distribution of number of breakpoints $K$, showing that 2 breakpoints is the most

probable number of breakpoints. From what we know about the test data the high probability of a 2 break set is correct. Figure 3 depicts the distribution of locations of breakpoints showing that $t = 100$ and $t = 200$ are the most probable locations for breakpoints. These two locations are in fact the true breakpoints signaling that the BAAR method accurately described the number and location of breakpoints for this simulated time series set. Once the breakpoint are found and used to model the simulated data the fits obtained from the $\beta$ and $\sigma$ draws accurately represent the data (Figure 4). The fitted values accurately describe the data because the fitted values lay so close to the true signaling that the breakpoints adequately split up the data.

## 3.2 Move Simulations

To determine the correct probabilities of doing a jump step over a jiggle step, a series of simulations was run on a training data set with one break (Figure 1). 11 different combinations of jump-jiggle probabilities, ranging from all jump to all jiggle, were tested using 5000 iteration runs. Acceptance rates, both overall and for each move type, were used as the metric to gauge each combination's efficiency at exploring the distribution (Table 1).

Generally, as the likelihood of doing jiggle over jump increased, the acceptance rate also increased, peaking at 10% jump 90% jiggle with an overall acceptance rate of 8.74% and jiggle acceptance rate of 19.27%. This shows that jiggle more accurately explores the distribution at lower numbers of iterations than jump by more frequently proposing favorable breakpoints. However, some amount of jump is necessary to effectively explore the space. The acceptance rate of jump steps alone shows a slight parabolic trend that peaks at 40% jump/60% jiggle. Based on these two peaks,

we recommend a jump/jiggle combination that is somewhere in the 10% to 40% jump/90% to 60% jiggle range with 25% jump/75% jiggle being the default setting for our algorithm.

While the overall acceptance rates were relatively low in these simulations, this is not surprising given the nature of the training data, which had only one relatively clear break. Running a similar series of simulations on the real data from our case study on brown pelicans (detailed below) shows that the generally optimal acceptance rate of 23.4% (Robert et al., 1997) is reached at between 10% and 20% jump/90% and 80% jiggle (Table 2).

Table 1: Move Simulation Results

| **Probabilities** | | **Acceptance Rate Results** | | |
|---|---|---|---|---|
| Jump | Jiggle | Total Rate | Jump Rate | Jiggle Rate |
| 0 | 1 | 0.0830 | N/A | 0.1679 |
| 0.1 | 0.9 | 0.0874 | 0.0089 | 0.1927 |
| 0.2 | 0.8 | 0.0678 | 0.0098 | 0.1699 |
| 0.3 | 0.7 | 0.0552 | 0.0097 | 0.1566 |
| 0.4 | 0.6 | 0.0560 | 0.0116 | 0.1810 |
| 0.5 | 0.5 | 0.0454 | 0.0105 | 0.1733 |
| 0.6 | 0.4 | 0.0328 | 0.0059 | 0.1518 |
| 0.7 | 0.3 | 0.0264 | 0.0075 | 0.1678 |
| 0.8 | 0.2 | 0.0176 | 0.0091 | 0.1420 |
| 0.9 | 0.1 | 0.0122 | 0.0088 | 0.1830 |
| 1 | 0 | 0.0040 | 0.0082 | N/A |

Table 2: Pelican Data Acceptance Rate Move Simulations

| **Probabilities** | | **Acceptance Rate Results** | | | | |
|---|---|---|---|---|---|---|
| Jump | Jiggle | Total Rate | Jump Rate | Jiggle Rate | Addition Rate | Subtraction Rate |
| 0 | 1 | 0.2780 | N/A | 0.5489 | 0.0050 | 0.1066 |
| 0.1 | 0.9 | 0.2478 | 0.0885 | 0.5262 | 0.0046 | 0.0930 |
| 0.2 | 0.8 | 0.2154 | 0.0700 | 0.5239 | 0.0045 | 0.1392 |
| 0.3 | 0.7 | 0.2098 | 0.0598 | 0.5584 | 0.0025 | 0.0800 |
| 0.4 | 0.6 | 0.1822 | 0.0598 | 0.5342 | 0.0025 | 0.1692 |
| 0.5 | 0.5 | 0.1522 | 0.0711 | 0.5541 | 0.0012 | 0.0454 |
| 0.6 | 0.4 | 0.1494 | 0.1197 | 0.5278 | 0.0045 | 0.1774 |
| 0.7 | 0.3 | 0.1180 | 0.0863 | 0.5722 | 0.0049 | 0.1594 |
| 0.8 | 0.2 | 0.0756 | 0.0459 | 0.5142 | 0.0029 | 0.0909 |
| 0.9 | 0.1 | 0.0662 | 0.0702 | 0.5496 | 0.0081 | 0.1221 |
| 1 | 0 | 0.0546 | 0.1024 | N/A | 0.0048 | 0.1690 |

## 3.3 Starting Condition Simulations

Starting conditions were analyzed using a longer training data set with eight breaks (Figure 2). Four different starting conditions were tested: Bai-Perron allowed to run until it found all 8 breaks, Bai-Perron constrained to finding 2 breakpoints, Bai-Perron constrained to finding 1 breakpoint, and an arbitrary middle placement. Each starting condition was run through BAAR 10 times with 10,000 iterations per run. The half-life of the mean squared error was used to assess the mean burn-in period for each starting condition.

Unconstrained Bai-Perron performed the best by far, requiring no appreciable burn-in in any of the runs. However, the mean half-life for Bai-Perron constrained to 2 breakpoint was not significantly different from that of the unconstrained Bai-Perron, despite starting with 6 fewer breakpoints, and can be run faster ( 3 seconds faster for the compiled linear Bai-Perron test from 'strucchange'; uncompiled AR Bai-Perron can barely run data of this length up to 2 breakpoints, let alone 8). The upper 95% CI bound for Bai-Perron constrained to 2 breakpoints was 978 iterations, roughly equal to the number of data points.

As a result, we recommend users use Bai-Perron constrained to 2 breakpoints for their starting conditions combined with a generous burn-in period of 2 times the number of observations in the data set. This should result in high-quality sampling of the distribution for a wide range of data types and structures.

Table 3: MSE Half-Lives from Starting Condition Simulations

| **Summary Statistics** | | | **95% Confidence Interval** | |
|---|---|---|---|---|
| Starting Condition | Mean | Standard Deviation | Upper | Lower |
| Unconstrained Bai-Perron | 0 | 0 | 0 | 0 |
| Bai-Perron - Max. 2 Bkpt. | 475.7 | 250.9 | 977.6 | -26.15 |
| Bai-Perron - Max. 1 Bkpt. | 865.3 | 413.4 | 1692 | 38.51 |
| Middle Placement | 730.7 | 266.3 | 1263 | 198.1 |

# 4 Case Study

In the mid-20th Century, brown pelicans (*Pelecanus occidentalis*), one of only two pelican species found in the United States, underwent a dramatic decline (Jehl, 1973; King et al., 1977). This decline was likely caused by the introduction of the pesticide DDT (dichloro-diphenyl-trichloroethane) for public use in the mid-1940s. In addition to being linked to the reproductive failure of numerous other bird species (Porter and Wiemeyer, 1969; Weseloh et al., 1983; Wiemeyer, 1984), DDT was linked specifically to the decline of brown pelicans in both the eastern (Blus, 1982) and western United States (Anderson et al., 1975; Lamont et al., 1970). The link between DDT and the decline of brown pelicans is well-established making it an excellent case study for testing BAAR's efficacy with real data.

The Pacific brown pelican population data generated by the Christmas Bird count from 1938 to 2016 (Figure 2) was run through BAAR using a generous burn-in period of 1,500 iterations and a sampling period of 10,000 iterations. Due to brown pelicans reaching sexual maturity in 3 to 5 years, an AR(3) model was used. In addition to the acceptance rate information explored above, BAAR produces three useful objects for our examination of pelican populations:

A) a list of the number of breakpoints at the end of each iteration, which can be graphed as a histogram (Figure 3);

B) a matrix with the breakpoint set at the end of the each iteration, which can also be graphed as a histogram (Figure 4);

C) and coefficient ($\beta$), sigma ($\sigma$), and fitted value draws that can be used to assess

the posterior mean fit of a given breakpoint set, which can be graphed alongside the fitted values from a single AR(3) model of the population (Figure 5).

BAAR tells us that there is an 86% chance there is 1 breakpoint in the pelican population data and an 14% chance there are 2 breakpoints. If there is 1 breakpoint, there is a 99% chance it is between 1949 and 1952. Given a lag of 3 to 5 years, that breakpoint corresponds very well to the start of public sale of DDT in the mid-1940s and thus fits with existing knowledge about reproductive success and population dynamics in brown pelicans.

On the off (14%) chance there is a second breakpoint, it is 98% likely to exist between 2008 and 2010. This less likely breakpoint corresponds to the removal of brown pelicans from the Endangered Species List and a recent population decline due to breeding failure in Baja California, most likely due to the interplay of climate change and overfishing in the region (Jacqes, 2016).

The one period of note in brown pelican conservation that the BAAR did not pick up was the United States ban on DDT (1972) and introduction of the Endangered Species Act (1973). However, BAAR looks at models of the population, not necessarily trends. A population is much easier to decimate than rebuild, so despite encouraging recent trends, the Pacific brown pelican population can be modeled well by splitting the data at 1949 and utilizing two AR(3) models (Figure 5). The $\Delta$BIC between a single AR(3) model (54.75419) fit across the entire data set and the mean posterior BIC (7.138885) for data split at 1949 is 47.6153, showing that the piecewise model favored by BAAR is significantly better.

BALR, which fits linear rather than autoregressive models, can be used to identify

major changes in trend in the Pacific brown pelican population data and does place a likely breakpoint in the mid- to late 1970s.

# 5 Discussion

Bayesian Adaptive Auto-Regression (BAAR) is a Bayesian technique that samples from the distribution of the quantity and locations of possible breakpoints in time series data. Once the breakpoints are found, the structural change can be accounted for and more accurate and appropriate modeling can be used on the data.

The key feature of BAAR is that it can be used to find breakpoints in a broad variety of time series data. The degree of the AR model is user inputed, so it can be adapted based on the type of desired regression. As a result of the Bai-Perron initial placement step, the jiggle option in the MCMC, and the BIC approximation of the Metropolis-Hastings ratio, BAAR is a computationally efficient. The fast run-time allows it to be able to handle particularly large data sets in a reasonable amount of time. An evident advantage of our method that distinguishes it from Bai-Perron is that BAAR has no user-inputted value for a fixed maximum number of breakpoints that occur in a given model. Using the Bayesian approach with $K$ being an unknown parameter, the procedure is able to find breakpoints in data that people may not be able detect.

This paper uses the population of Pacific brown pelicans as a case study to demonstrate the effectiveness of our technique. BAAR was able to detect the change in the model caused by the introduction of DDT using only the data set as a basis. This indicates the capabilities of Bayesian Adaptive Auto-Regression as an effective technique at finding breakpoints in numerous types of time series data, especially when

outside influences cause structural changes in the models.

# 6 Appendix

## 6.1 Appendix 1:

*Detailed Balance for Addition*

It is necessary to prove that detailed balance holds such that

$$\pi(M_k)p(M_{k+1}|M_k) = \pi(M_{k+1})p(M_k|M_{k+1})$$

where $M_k$ expresses the parameters of the model with $k$ breakpoints. Thus, $M_k = \{k, \tau_1, \ldots, \tau_k\}$ for $k = 1, 2, \ldots, k_{max} - 1$ where $k_{max}$ is the maximum amount of breakpoints that can be placed. Therefore, $\pi(M_{k+1})$ has a density of

$$\pi(M_{k+1}) = \frac{p(y|\tau_1, \ldots, \tau_{k+1})p(\tau_1, \ldots \tau_{k+1})p(k+1)}{p(y)}.$$

When going from $M_k$ to $M_{k+1}$ by proposing an addition step in the MCMC, let

$$M_k = \{k-1, \tau_1, \tau_2, \ldots, \tau_{j*-1}, \tau_{j*+1}, \ldots \tau_k\}$$

$$M_{k+1} = \{k, \tau_1, \tau_2, \ldots, \tau_{j*-1}, \tau_{j*}, \tau_{j*+1}, \ldots \tau_k\}$$

where the sets differ in the $j*$ element. The transition probabilities now follows:

$$\begin{aligned} p(M_{k+1}|M_k) &= p(k+1|k) \times p(add\ \tau_{j*}|k) \times (acceptance\ probability) \\ &= b_k \times \frac{1}{n_{free}} \times min(1, A). \end{aligned}$$

$$p(M_k|M_{k+1}) = p(k|k+1) \times p(delete\ \tau_{j*}|k+1) \times (acceptance\ probability)$$
$$= d_{k+1} \times \frac{1}{k+1} \times min(1, B)$$

such that $A = \frac{\pi(M_{k+1})}{\pi(M_k)} \frac{d_{k+1} \times \frac{1}{k+1}}{b_k \times \frac{1}{n_{free}}}$ and $B = \frac{\pi(M_k)}{\pi(M_{k+1})} \frac{b_k \times \frac{1}{n_{free}}}{d_{k+1} \times \frac{1}{k+1}} = \frac{1}{A}$. Let $A < 1$, then

$$\begin{aligned}
\pi(M_k)p(M_{k+1}|M_k) &= \pi(M_k)b_k \frac{1}{n_{free}} A \\
&= \pi(M_k)b_k \frac{1}{n_{free}} \frac{\pi(M_{k+1})}{\pi(M_k)} \frac{d_{k+1}\frac{1}{k+1}}{b_k \frac{1}{n_{free}}} \\
&= \pi(M_{k+1})d_{k+1} \frac{1}{k+1} \\
&= \pi(M_{k+1})p(M_k|M_{k+1}).
\end{aligned}$$

The case of subtraction where $A > 1$ the proof is similar to the structure above. Likewise, in a move step the proof is comparable.

## Acknowledgements

This research was funded by the National Science Foundation, grant number #1650222. The research was supported by the Lafayette College Research Experience for Undergraduates (REU) Summer 2018.

## References

Anderson, D.W., Jehl, J.R., Risebrough, R.W., Woods, L.A., Deweese, L.R. and Edgecomb, W.G., (1975). *Brown pelicans: improved reproduction off the southern California coast.* Science, 190(4216), pp.806-808.

Bai, J. and Perron, P., (1998). *Estimating and testing linear models with multiple structural changes*. Econometrica, pp.47-78.

Bai, J. and Perron, P., (2003). *Computation and analysis of multiple structural change models*. Journal of applied econometrics, 18(1), pp.1-22.

Blus, L.J., (1982). *Further interpretation of the relation of organochlorine residues in brown pelican eggs to reproductive success*. Environmental Pollution Series A, Ecological and Biological, 28(1), pp.15-33.

Denison, D.G.T., Mallick, B.K. and Smith, A.F.M., (1998). *Automatic Bayesian curve fitting*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 60(2), pp.333-350.

DiMatteo, I., Genovese, C.R. and Kass, R.E., 2001. *Bayesian curvefitting with freeknot splines*. Biometrika, 88(4), pp.1055-1071.

Gamber, E.N., Liebner, J.P. and Smith, J.K., (2016). I*nflation persistence: revisited*. International Journal of Monetary Economics and Finance, 9(1), pp.25-44.

Jacqes, D.L., (2016). *California Brown Pelican Monitoring Summary 2014: The Year of the Blob*. U.S. Fish & Wildlife Service.

Jehl, J.R., (1973). *Studies of a declining population of Brown Pelicans in northwestern Baja California*. The Condor, 75(1), pp.69-79.

Kass, R.E. and Wasserman, L., (1995). *A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion*. Journal of the american statistical association, 90(431), pp.928-934.

King, K.A., Flickinger, E.L. and Hildebrand, H.H., (1977). *The decline of brown pelicans on the Louisiana and Texas Gulf Coast.* The Southwestern Naturalist, pp.417-431.

Lamont, T.G., Bagley, G.E. and Reichel, W.L., (1970). *Residues of O, P-DDD and O, P-DDT in brown pelican eggs and mallard ducks.* Bulletin of environmental contamination and toxicology, 5(3), pp.231-236.

McLeod, A.I. and Zhang, Y., (2008). *Improved subset autoregression: With R package.* Journal of Statistical Software, 28(2), pp.1-28.

Pesaran, M.H., Pettenuzzo, D. and Timmermann, A., (2006). *Forecasting time series subject to multiple structural breaks.* The Review of Economic Studies, 73(4), pp.1057-1084.

Pesaran, M.H. and Timmermann, A., (2002). *Market timing and return prediction under model instability.* Journal of Empirical Finance, 9(5), pp.495-510.

Porter, R.D. and Wiemeyer, S.N., (1969). *Dieldrin and DDT: effects on sparrow hawk eggshells and reproduction.* Science, 165(3889), pp.199-200.

R Core Team( 2017). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing

Roberts, G.O., Gelman, A. and Gilks, W.R., (1997). *Weak convergence and optimal scaling of random walk Metropolis algorithms.* The annals of applied probability, 7(1), pp.110-120.

Ruggieri, E.,( 2013). *A Bayesian approach to detecting change points in climatic records.* International Journal of Climatology, 33(2), pp.520-528.

Seidel, D.J. and Lanzante, J.R., (2004). *An assessment of three alternatives to linear trends for characterizing global atmospheric temperature changes.* Journal of Geophysical Research: Atmospheres, 109(D14).

Wallstrom, G., Liebner, J. and Kass, R.E., (2008). *An implementation of Bayesian adaptive regression splines (BARS) in C with S and R wrappers.* Journal of Statistical Software, 26(1), p.1.

Weseloh, D.V., Teeple, S.M. and Gilbertson, M., (1983). *Double-crested cormorants of the Great Lakes: egg-laying parameters, reproductive failure, and contaminant residues in eggs, Lake Huron 19721973.* Canadian Journal of Zoology, 61(2), pp.427-436.

Wiemeyer, S.N., Lamont, T.G., Bunck, C.M., Sindelar, C.R., Gramlich, F.J., Fraser, J.D. and Byrd, M.A., (1984). *Organochlorine pesticide, polychlorobiphenyl, and mercury residues in bald eagle eggs196979and their relationships to shell thinning and reproduction.* Archives of Environmental Contamination and Toxicology, 13(5), pp.529-549.

Zeileis, A., Leisch, F., Hansen, B., Hornik, K., Kleiber, C. and Zeileis, M.A., (2007). *The strucchange Package.* R manual.

Zhou, S. and Shen, X., (2001). *Spatially adaptive regression splines and accurate knot selection schemes.* Journal of the American Statistical Association, 96(453), pp.247-259.