CIMAT

Centro de Investigación en Matemáticas, A.C.

Improvements in Chronology Building from ^{14}C Measurements Using Bayesian Inference on Autoregressive Gamma Processes

TESIS

para obtener el grado de

Maestro en Ciencias

Con especialidad en

Probabilidad y Estadística

Presenta

Nicolás Elio Kuschinski Kathmann

Director de Tesis:

Dr. J. Andrés Christen Gracia

Guanajuato, Gto. Octubre de 2014

Para Camay

Agradecimientos

Agradezco a CONACYT por el soporte económico brindado durante mis estudios en la maestría, y al CIMAT, institución en la que realicé dichos estudios

Agradezco también a mi asesor de tesis el Dr. J. Andrés Christen Gracia, y a quien fuera mi tutor antes de comenzarla, el Dr. Joaquín Ortega Sánchez.

También quisiera agradecer al Dr. Maarten Blaauw, quien me brindó generosamente su tiempo, tanto para comprender la naturaleza de los problemas presentes como para sugerir numerosas mejoras en la tesis.

Igualmente agradezco a Lucina Kathmann y al Dr. Nicholas Patricca por su apoyo moral y sus consejos para mejorar la redacción.

Extiendo mi agradecimiento a Marco Aquino, quien se tomó el tiempo para explicarme la manera de usar las tablas INTCAL.

Finalmente agradezco a Selene González, ya que sin su apoyo probablemente nunca lograría nada, mucho menos la realización de esta tesis.

Resumen

En altas latitudes se encuentran depósitos de turba con estratos claramente visibles. Estos estratos contienen materia orgánica, la cual se puede fechar por ^{14}C, también conocido como carbono 14.

Es de interés construir una cronología que asocie a cada profundidad su edad correspondiente. Un programa para hacer esto se desarrolló en 2011 por Cristen y Blaauw.

El objetivo de esta tesis es continuar desarrollando el trabajo de Christen y Blaauw, reescribiendo el algoritmo en Python y extendiendo su funcionalidad al permitir la inclusión de información adicional (no obtenida por ^{14}C).

Abstract

In high lattitudes thee are peat deposits with clearly visible starta. These strata contain organic material which can be dated by ^{14}C, also known as carbon 14.

It is of interest to construct a chronology that associates each depth to its corresponding age. A program to do this was developed in 2011 by Christen and Blaauw.

The objective of this thesis is to further develop the work of Christen and Blaauw, rewriting the algorithm in Python and extending its functionality to permit the inclusion of additional information (not obtained by ^{14}C).

Table of contents

Improvements in Chronology Building from ^{14}C Measurements Using Bayesian Inference on Autoregressive Gamma Processes

Nicolás Elio Kuschinski Kathmann

Chapter 1

Introduction

1.1 Chronology building

In paleoecology and archaeology, radiocarbon dating has become the standard dating technique used for organic samples. While extremely useful, Carbon 14 dating is also very expensive. In order to build a complete timeline for a process, an enormous number of measurements may be required, thus significantly complicating any effort to study it.

To provide an alternative to simply taking a very large number of measurements, statistical considerations can be used for chronology building: Carbon 14 measurements are not independent, and using certain assumptions about their dependence structure, the data from some measurements may be used to draw inference about others, thus reducing the number of samples which must be dated directly.

One such form of dependence structure is found in vertically tiered layers of earth. Layers of sediment build atop each other over time, creating deposits of earth in which the depth increases with age. These structures occur frequently in peat bogs and are known as peat cores. Strata are often plainly visible, and hence if any disruption in the process exists, it is usually easy to detect. With this in mind, we may embark on chronology building by using a regression model which associates the depth of a sample to its age.

1.2 Radiocarbon dating

All organic matter has molecules which contain atoms of carbon. Most of this carbon is ^{12}C, which is a stable isotope. Of the remaining carbon, the great majority is ^{13}C, which is also stable. There is a very small remaining portion of ^{14}C, or carbon 14. While ^{14}C is present in very small quantities, it is radioactive, and decays over time at a predictable rate, having a half-life of slightly less than 6000 years.

When an organism dies, it ceases to interact with the environment, and it no longer acquires ^{14}C. As ^{14}C decays it becomes ^{14}N (nitrogen), and by measuring the proportion of ^{14}C in a sample, this information can be used along with the initial proportion of ^{14}C to get a very good estimate of the time of death of the organism. The difficulty in this technique lies in obtaining the proportion of ^{14}C in the organism at the time of death. While it was once thought to be constant, it is now known that this proportion varies over time somewhat unpredictably.

Various techniques have been employed which have generated an accurate calibration curve which adjusts for the variation of ^{14}C in the biosphere. We now have a good understanding of the relationship between ^{14}C measurements and age, but because of the variations in environmental ^{14}C, the function is not injective. The issue is further complicated because measurements are not 100% accurate, and even small uncertainties in the measurements of ^{14}C can propagate into very large uncertainties in dating.

Christen and Pérez (2009) developed a bayesian statistical model in which the posterior distribution of the actual date of death of a sample, given its ^{14}C measurement, is found to be a t distribution.

1.3 BACON

The problem of chronology building in vertical cores of data was partially solved by Christen and Blaauw (2011) using the model of Christen and Pérez (2009) and using an autoregressive gamma process which models the function which relates depth and time as piecewise linear between a sequence of equally spaced nodes.

A prior distribution is fixed on the set of slopes of the function, on the intercept, and also on a memory parameter, and then a computer program uses MCMC techniques to generate a large sample of the posterior distribution of these parameters. The computer program, called BACON, uses a self-adjusting Metropolis-Hastings MCMC algorithm called t-walk (Christen and Fox, 2010) to keep the amount of calibration required to a minimum, thus enabling its use by non-experts.

BACON has had a positive reception by paleoecologists and is now widely used to build chronologies in peat cores.

1.4 The BACON model

In the BACON program, the function G which links age to depth is modeled as piecewise linear. Although the function is stochastic, the a priori distributions for each slope x_j are always such that $P(x_j \leq 0) = 0$, $\forall j$. Let $x = (x_j)$ be the (random) vector of slopes and θ be the (random) intercept. Also, denote with Δc the length of each linear piece and with $c = (c_j)$ the vector of endpoints of the pieces. Then for each depth d the corresponding age a is given by

$$a = G(d, \theta, x) = \theta + \sum_{j=1}^{i} x_j \Delta c + x_{i+1}(d - c_i),$$

with i such that $c_i < d < c_{i+1}$.

The a priori distributions of the slopes are generated by an autoregressive gamma process, in which $x_j = w x_{j+1} + (1 - w)\alpha_j$ where w is a memory parameter in $[0, 1]$ and the α_j are iid $\sim Gamma(a_\alpha, b_\alpha)$ (these parameters being set a priori, and representing prior information about the rate of sediment accumulation in the core.)

1.5 Beyond BACON

The present work intends to build on BACON by extending its portability, its usefulness, and its scope.

- BACON is written in the C++ programming language using the GNU Scientific Library (GSL), and has an interface written in the R scripting language. While R is a very portable language, the same cannot be said about the GSL library, which is easy to install on GNU operating systems like Linux, but which is more problematic for several popular systems such as Microsoft Windows. The BACON program has been re-written using the cross-platform Python programming language, which enables simultaneous maintenance on all operating systems (at the cost of some speed: Python is a scripting language and programs written in it are not as fast as programs written in C++).

- Chronology building is used to understand the change over time of arbitrary environmental indicators. BACON only associates age and depth, while the age of these indicators is tied only loosely to age by measuring them at various depths. While this does produce useful data to understand environmental change, a technique called ghost mapping is proposed to extend BACON so that it can be used to study environmental change more directly.

- The scope of BACON has been extended, allowing it to improve estimations by using information from sources other than the ^{14}C measurements within the core. Most of this information comes from other cores, and it is desireable to allow BACON to work with numerous cores at once, using parallel computing. Rather than simply running BACON separately on each core, any of four different kinds of information can be specified (three of which are dependence structures between the cores), using this information to improve our estimation of G in each core. This new, extended form of BACON has been dubbed BLT (BACON Linking Timescales).

1.6 Contents

- Chapter 2 discusses the BACON model in detail and the results of implementing this model using a t-walk algorithm in Python.

- Chapter 3 discusses the meaning and value of environmental indicators, or "proxies" and goes on to explain the development and implementation of a technique to use BACON to build a chronology of proxies.

- Chapter 4 revises the BACON model to allow for a certain form of external information to improve estimations. The new model, called BLT0, is detailed along with the algorithm, implementation, and results.

- Chapter 5 covers all other forms of BLT, detailing three different forms of correlation structure between cores. These models are implemented and their results are assessed and compared to the results which come from the initial form of BACON.

Chapter 2

BACON in Python

2.1 Details of the model

2.1.1 Assumptions about the chronology

The first assumption that is made is that the function relating depth and age is strictly increasing. This is sensible because strata accumulate over time by layering over previous strata. Of course, it is not possible to always expect natural phenomena to behave so regularly, but shifts or swaps in the progression can often be identified by experts.

Another assumption is the continuity of the function. This is also reasonable to expect, so long as strata accumulate over time without stopping. While certainly there may be periods of time when there is no accumulation of strata because of variability of wheather conditions (dry seasons, etc.), we can suppose that these periods are very short in relation to the time periods we are considering, and hence do not affect any estimations based on this assumption. It is always possible, of course, for dramatic geological events to break the continuity in more important ways, but as in the previous case, these discontinuities can often be identified by experts.

2.1.2 Building the model

2.1.2.1 The age function

Any absolutely continuous and increasing function can be approximated to an arbitrary degree by a sequence of straight line segments: If ever an approximation is insufficiently precise, more segments can be added.

Given the assumptions, one way to approximate the chronology is to model it as piecewise linear, with an intercept θ representing the age of the top stratum and a series of slopes $x = \{x_j\}$ representing the accumulation rates over k total segments, which will be taken by dividing the range of interest into segments of equal length

Let $c = \{c_j\}$ be the vector of endpoints of our segments. For any depth d, our model will propose a corresponding age a by a function G as follows:

$$a = G(d, \theta, x) = \theta + \sum_{j=1}^{i} x_j \Delta c + x_{i+1}(d - c_i),$$

with i such that $c_i < d < c_{i+1}$.

2.1.2.2 Distribution of the ^{14}C measurements

A single ^{14}C measurement provides a *carbon age* (m) and a *standard error* (s). The carbon age is not directly understood to be a measurement of the age, but a measurement of carbon activity which can be associated with an age. This association is found in the INTCAL tables, which are regularly updated with the most current estimations. All results shown in this document were obtained using the 2013 estimations (Reimer et al. 2013). The INTCAL tables are produced by measuring the carbon activity in samples of known historical age and performing a statistical analysis on these measurements.

The standard error reported by the ^{14}C measurements is also not exactly a measurement of the standard error, since this error is also dependent on the age of the sample. The updated estimations of the effects of ages on samples are also found in the INTCAL tables, and these are added to the reported measurement error when calculating the likelihood.

Traditionally, the measurements were expected to behave as if distributed normally with mean and standard deviation as described above, but this model has been shown to predict far fewer outliers than have been observed. Christen and Pérez (2009) proposed that the reported standard deviation was itself a measurement that was subject to error. Assigning a gamma distribution to this error, a t distribution was derived, which has heavier tails than the normal distribution and is more robust to the presence of outliers. This is the model that has been used throughout this document.

2.1.2.3 Additional details about the model

Some further assumptions have been made regarding the structure of the accumulation rates x, involving an additional "memory" parameter w, however a Bayesian model is used, and these assumptions need not be explicitly included in either the function G or in the likelihood, but rather encoded implicitly in the prior distributions for x. This structure is discussed in the corresponding section.

2.2 The log-likelihood function

To compute the log-likelihood function of a set of parameters (θ, x, w) we can separate the contributions of the measurements at each depth d_i. Computing $\hat{a}_i = G(d_i, \theta, x)$ and the ${}^{14}C$ measurement (m_i, s_i), the INTCAL tables are used to obtain μ and λ as functions of m_i, s_i and $\hat{a}_i$. The contribution to the log-likelihodd is then calculated using the t model as

$$log\mathcal{L}(\hat{a}_i) \quad = \quad \frac{log(\lambda/2)}{2} - \frac{3}{2} log(1 + \lambda/2)(m - \mu)^2$$

In order to calculate the full log-likelihood function, this calculation is repeated for each depth d_i at which there is a ${}^{14}C$ measurement available and we obtain $log\mathcal{L}(\hat{a}) = \sum_i log\mathcal{L}(\hat{a}_i)$

2.3 Bayesian statistics and priors

2.3.1 The concept of prior distributions

Bayesian techniques were used to complete the modeling process and perform calculations. In classical statistics, parameters are assumed to be fixed, unknown points, but in the Bayesian approach these parameters are modeled as random variables themselves. Whatever information is known about the parameters before data collection is encoded in a "prior distribution", which is simply a probability distribution over the parameter space. Any restrictions on the values of parameters which we wish to include in the model can be encoded directly into the prior distribution simply by truncating the support of the parameters to a certain subset of the space. The parameters are not necessarily assumed to be independent, and may have any sort of joint distribution. Very complex restrictions are possible by restricting certain joint behaviors of the parameters.

2.3.2 The prior distribution of (θ, x, w)

The first necessary restriction will be to bound the entire function G. For each core, prior knowledge will be used to determine the maximum age ($\overline{a}$) and minimum age ($\underline{a}$) possible for any sediment in the core. The prior will then be truncated so that $\pi(\theta, x, w)$ is 0 if (θ, x, w) would put any sediment outside of the interval $[\underline{a}, \overline{a}]$

It is unusual for accumulation rates of sediment to change drastically with time. They depend on climate, and climate change is, in general, a gradual process. We will model this change by describing x with an autoregressive gamma process:

$$x_j = wx_{j+1} + (1 - w)\alpha_j$$

where theα_j are iid $\sim$ $Gamma(a_\alpha, b_\alpha)$. w is a memory parameter, which represents how much the process depends on the past, and it will have its own prior distribution. To calculate the contribution to the log-prior of x we will evaluate the prior of the α since the prior of x is defined implicitly. Note that x_k, the last segment, remains free so the prior distribution will be $x_k \sim Gamma(a_\alpha, b_\alpha)$. To choose a_α and b_α we will use any prior knowledge available about the accumulation rate in the core in question.

It is worth noting that given w and x_{j+1}, the support of x_j is bounded below by wx_{j+1} and that so long as w is positive (it is: see below) then the positive support of x_k assures us that x is a vector of positive slopes only, and hence G will always be increasing.

At first it may seem strange to have x_j depend on x_{j+1} rather than x_{j-1} but we must remember that x_{j+1} is deeper than x_j and hence older: The future depends on the past.

For θ, prior knowledge can be used if it is available, but for the examples in this document, we used the uninformative prior $\theta \sim U(\underline{a}, \overline{a})$.

Finally, the memory parameter w is in the interval $[0, 1]$. As is customary for parameters restricted to this interval, the prior used will be a $w \sim Beta(a_w, b_w)$ with a_w and $b_w$chosen to reflect any prior knowledge available.

2.4 The posterior distribution and the energy function

2.4.1 Bayesian inference

In Bayesian statistics, the objective is to improve the prior distribution by using the data. The prior distribution is, in general, quite uninformative, containing only what is known about the parameters beforehand.

The prior distribution is then combined with the data using Bayes's theorem:

$$\pi_{a|b}(a|b) = \frac{\pi_{b|a}(b|a)\pi_a(a)}{\int \pi_{a|b}(a|x)\pi_a(x)dx}$$

We interpret this theorem using a as the parameters, and b as the data. Hence, π_a is the prior information available about the parameters and $\pi_{b|a}$ is the likelihood function. Hence, what we want to find is $\pi_{a|b}$: The distribution of the parameters once the data is known. This distribution is called the *posterior distribution*, and its calculation is one of the primary objectives of Bayesian inference.

2.4.2 The difficulties involving the posterior distribution

Using Bayes's theorem, once we have the likelihood function and the prior distribution, the posterior distribution can simply be written as

$$\pi_{(\theta,x,w)|Y}(\theta, x, w|Y) = \frac{\mathcal{L}(Y|\theta, x, w)\pi(\theta, x, w)}{\int_S \mathcal{L}(Y|u, v, b)\pi(u, v, b)d(u, v, b)}$$

with $\mathcal{L}$ the likelihood function written above and $\pi(\theta, x, w)$ the prior distribution. Unfortunately this is not useful because we do not know any of the properties of this function: In particular, we are not even able to evaluate it at a single point because we do not know how to solve the integral $\int_S \mathcal{L}(Y|u, v, b)\pi(u, v, b)d(u, v, b)$.

What we can do is observe that the value of the troublesome integral does not depend on the value of (θ, x, w), so we can instead write

$$\pi_{(\theta,x,w)|Y}(\theta, x, w|Y) \propto \mathcal{L}(Y|\theta, x, w)\pi(\theta, x, w)$$

$$= I_{supp}(\theta, x, w)\mathcal{L}(Y|\theta, x, w)\pi(\theta), \pi(w)\pi(x|w)$$

which can be calculated at any point. If we restrict ourselves to the support we will note that the function

$$U(\theta, x, w) = -log\mathcal{L}(Y|\theta, x, w) - log\pi(\theta) - log\pi(w) - log\pi(x|w)$$

differs from the negative of the log posterior by a constant. This function U is called the *energy function.*

2.5 Sampling from the posterior distribution

2.5.1 The basic idea

While it is unreasonable to attempt to analyse the properties of the posterior distribution directly, there is an indirect approach which can give us quite a lot of information. Rather than thinking of the posterior distribution as a function, we can take advantage of the fact that it is, after all, a probability distribution, and can hence be analyzed using statistical techniques so long as there is some way to obtain a sample.

If there is a large sample from the posterior distribution, the Glivenko-Cantelli theorem will allow us to study the distribution by studying the empirical distribution as derived from the sample. If the distribution has finite moments, for instance, these moments can be estimated by using the empirical moments from the sample. The objective, therefore, is to obtain a large sample of the posterior distribution: The bigger the better.

The posterior distribution is a very abstract entity, and there is nothing in the data already collected that in any way resembles a sample from the posterior distribution. It is impossible to collect a sample from a phenomenon that is distributed in this way (such a phenomenon does not exist). Instead, the sample is created using a computer.

2.5.2 MCMC

To generate a sample of the posterior distribution MCMC techniques will be used. In MCMC a Markov chain is designed over the parameter space which has the posterior as an asymptotic distribution. From an arbitrary starting point, the Markov chain is simulated using a computer for a large number of iterations. Eventually the Markov chain will converge to the posterior distribution, and its state can be considered an element of the sample.

The power of MCMC lies in the fact that there are known algorithms, such Metropolis-Hastings algorithms (described in detail in the chapter about BLT) which are able to produce the required Markov chain even with the ability to calculate nothing more than the indicator function of the support and the energy function.

2.5.3 The t-walk

In general, MCMC techniques are very complex, requiring careful calibration by an expert, both to optimize (insofar as this is possible) the convergence speed and also to identify how long the chain must be simulated before the sample is drawn.

BACON, as a piece of software, is not intended to be used only by professional Bayesian statisticians, but also by non-experts such as paleoecologists who are not versed in Bayesian statistics at all, let alone calibration of MCMC.

A solution to this problem is the use of a Python library written by Cristen in 2010, called the t-walk (http://www.cimat.mx/˜jac/twalk/). The t-walk uses a special kind of general purpose Metropolis-Hastings algorithm which is self-adjusting to sample from any distribution when given the support and energy functions.

2.6 Using Python

2.6.1 Why use Python?

One of the disadvantages of the current implementation of BACON is portability. The program is written in the C++ programming language, using the GSL library. For an operating system with the necessary tools, the process of installing the necessary libraries and compiling the program is trivial, however not all targeted operating systems have the necessary tools in place. In particular, the most popular system in the world, Microsoft Windows, does not have the required development tools off the shelf. Hence, in order to make any changes or updates to the software, the entire GSL library must be compiled manually under Windows, which is a lengthy (and occasionally difficult) process. Another issue is that many users lack the technical expertise to compile software themselves (and in Windows it is too complex of a process to describe completely in the manual) so users need to be notified of the existence of new executables, which must be distributed to them directly.

The solution which was undertaken here was to rewrite BACON from scratch in the Python 2 (hereafter simply "Python") programming language, which is frequently used for scientific and mathematical computing. Python is a cross-platform scripting language, meaning the program does not have to be compiled in order to run. While tools to run Python programs are not included by default in Windows either, they can be quickly and easily installed.

Numerous Python libraries were used, most of which are considered standard libraries that are easily installed using any platform and included by default in many python development packages. The only exception is the *pytwalk* library, which can be found at http://www.cimat.mx/~jac/twalk/

Python is easily installed and maintained accross platforms, and allows maintenance and updates on BACON to be done on all platforms simultaneously.

2.6.2 Disadvantages

The biggest disadvantage to using Python (or any other scripting language) is speed: Unlike C++, programs written in Python must be read, parsed, and executed by a Python interpreter rather than directly by the operating system. This extra step slows down the operation significantly. While the difference in speed is hardly noticeable for simple tasks, computer-intensive processes, like MCMC, require the computer to run algorithms through millions of iterations, and the difference in speed is a real issue.

For BACON, the difference in computer time between C++ and Python is so large that it is not recommended to use the Python implementation directly. An alternative approach (using a C++/Python hybrid) has been proposed - but not yet implemented - which may solve this issue (see next section).

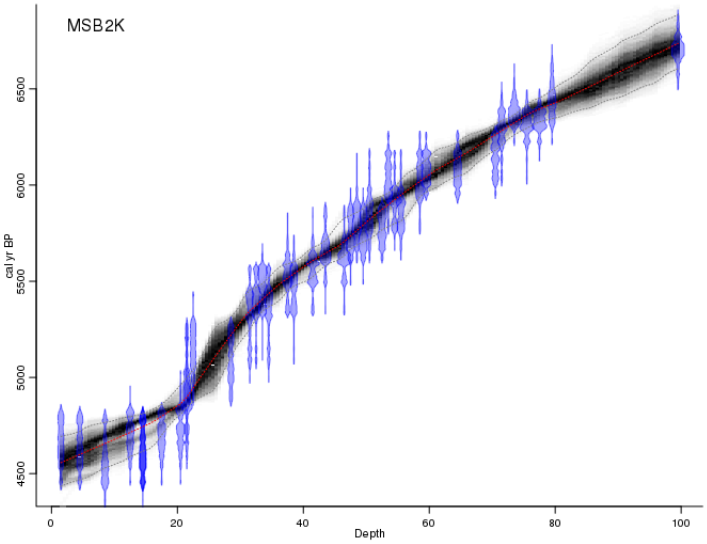

Figure 2.1: The output of the original BACON program

2.7 Results and Discussion

BACON in Python is about 50 times slower than the original version of BACON written in C++. BACON usually performs around 8 million iterations in about 3 minutes, while in the Python version, this is an operation that takes several hours. Depending on what kind of information is required, however, a much smaller run may be used to get reasonable estimations.

A comparison of results between the C++ and Python versions of BACON confirms that both programs are in fact estimating the same posterior distribution. Figure 2.1 is the graph generated by the R interface of the output of the posterior distribution of the MSB2K peat core using the C++ version of BACON at the program's default settings. This is the same core and settings as were used and published in the original Blaauw ande Christen (2011) paper.

For figures 2.2 and 2.3, the Python version of BACON was used for a run of 50,000 iterations over 20 sections, using the same prior distribution for the parameters. Figure 2.2 shows the history of the log-posterior function evaluated at the state of the Markov chain, and it is clear that the number of iterations is not sufficient to get a large iid sample. While the burn-in time (time until convergence) of the Markov chain is possibly within the 50,000

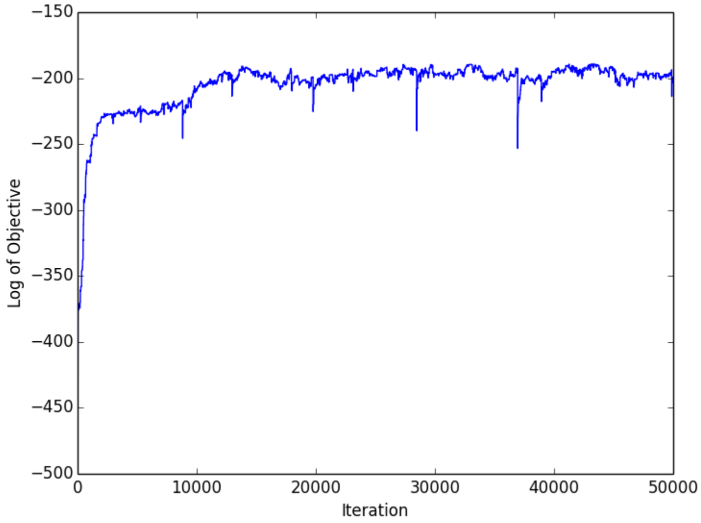

Figure 2.2: The energy of the Python version over50,000 iterations. In order to get a good posterior sample, 50,000 iterations is not enough.

iterations used, it is fairly clear is that there is not enough time given for many of the elements of the sample to be considered uncorrelated.

Figure 2.3 shows the resulting estimation of the log-posterior: The first 20000 iterations are unused (since we consider this prior to convergence) and then every 100 iterations thereafter the state of the chain is taken, and the parameters (θ, x, w) are used to plot the function G with 2% opacity. The result is that at each vertical cross-section, the graph is equivalent to a greyscale kernel estimate of the posterior density of the depth at a given age (using a uniform kernel of bandwidth corresponding to one pixel's worth of time). Although the estimate is cruder than the estimate obtained with the much larger sample generated by the C++ version, the graphs are already representing the same posterior distribution, as can be seen by observing that both graphs place the entire chronology in approximately the same range, and roughly correspond in those regions of little uncertainty.

Figure 2.4 shows the same graph when the Python version runs for 1 million iterations, with a sample point taken every 1000 of them after the first 50,000. While this is still not ideal, it gives a much closer match to the C++ version, but the time taken by the algorithm is already nearly

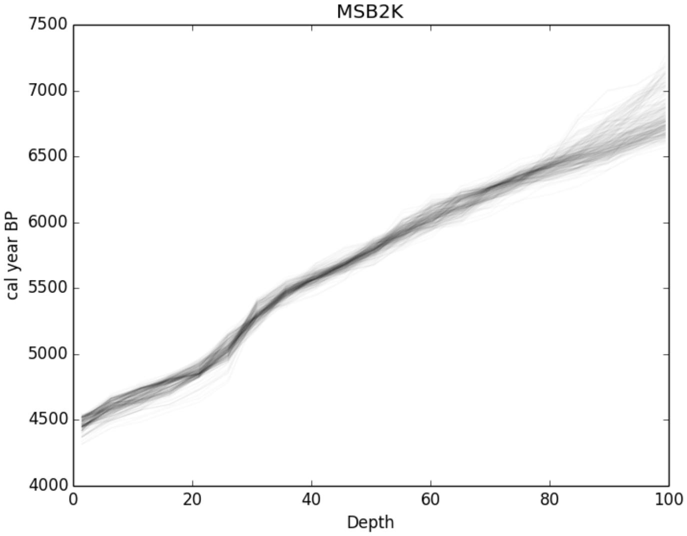

Figure 2.3: A sample from the 50,000 iterations of the python version taking one iteration ever 100. At this many iterations, the posterior estimation is very crude.

20 minutes, in contrast to the C++ version of BACON which takes about 3 minutes to do a full run (about 8 million iterations) on the same hardware.

There is a way in which the inefficiency of the Python programming language might be improved, yielding times which could potentially be comparable to those produced by the original BACON program. A time analysis of the program using the cProfile python library found that the great majority of the time used by the program is spent on a small number of functions. These functions could be programmed in C++ (without having to use the GSL library) and incorporated into a Python library, which would take advantage of the speed from the C++ version.

This proposed solution (which is explained in slightly greater detail in section 6.2.1) would only slightly increase the difficulty of maintaining the program, while potentially increasing its usefulness - hopefully to the point of being on par with the original program.

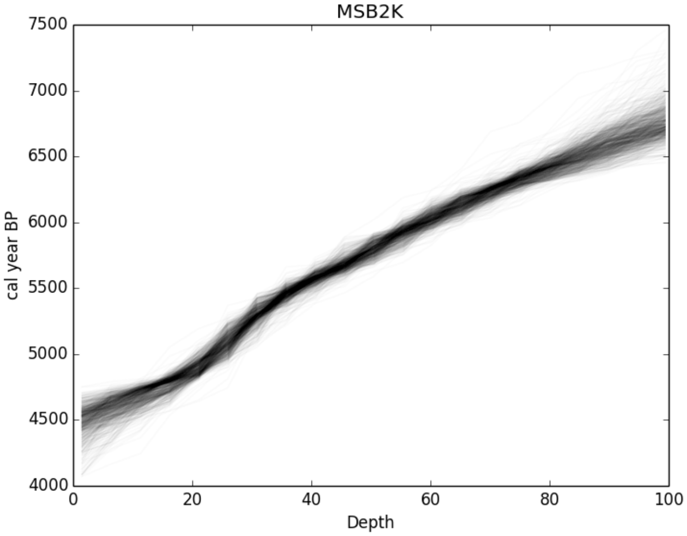

Figure 2.4: At 1,000,000 iterations, taking one every 1000, the graph is now very similar to figure 2.1.

Chapter 3

Ghost Maps

3.1 Properties of the BACON model

Recall from the previous chapter, that the BACON model for the age of a sediment depth is

$$a = G(d, \theta, x) = \theta + \sum_{j=1}^{i} x_j \Delta c + x_{j+1}(d - c_j)$$

Note that θ and x are the only random elements in G. In other words,$G(d, \theta, x)|\theta, x$ is deterministic for each depth d. Also note that since $P_{post}(x_j \leq 0) = 0$, $\forall j$, then $P_{post}(G\text{is strictly increasing}) = 1$ also.

Note now that with probability 1, $G|(\theta, x)$ is injective, and hence invertible. Moreover, for fixed θ and x, $(G|(\theta, x))^{-1}$is deterministic, and $(G|(\theta, x))^{-1}(a)$ can be explicitly computed for any age a in the counterdomain of $G|(\theta, x)$ with relative ease (further still, $(G|(\theta, x))^{-1}$is also strictly increasing and piecewise linear).

3.2 Proxies

For paleoecologists, a mapping of age to depth in a vertical core is not a completed product. The main reason to use BACON is not to assign a date to a depth, but rather to assign a date to the various things found at the corresponding depth.

While it is, for example, of little interest to paleoecology to know that at a certain location, a depth of 3.2 meters corresponds to an age of 7300 years, it is of much greater interest that this is the time of the extinction of a certain species of plant. While carbon dating usually provides information about the age of specific samples directly, BACON does so indirectly by reporting information about the depth at which these samples were found.

The use of depth as a covariable is a very useful technique to improve both the cost-efficiency and accuracy of carbon dating. That said, it does make the information BACON provides more difficult to interpret in the context of paleoecology.

It is frequently the case, therefore, that depth is usually associated with other covariables, known as proxies. Proxies are usually environmental indicators, such as the pollen of a specific kind of flower, the water level of a lake, or the average temperature.

Several proxies of interest may be difficult or imposible to measure directly. That being said, for the purpose of our mathematical model, it is convenient to consider only those proxies for which it may be expected that an expert could obtain a good (point) estimate at any given depth in a core.

The focus of BACON may hence be shifted from merely attempting to reduce the uncertainty of G to one which addresses the needs of paleoecologists more directly. While it is not desireable to completely omit the use of detph to improve our estimation, it is important to realize that the use of depth to ascertain age is merely a secondary concern, and that what really interests us is the age of the proxies found at this depth.

3.3 Ghost maps

To directly address the dating of proxies, the function G which associates age and depth may be used along with the measurement of proxies at different depths to produce a new kind of function which relates age and a proxy. This kind of function was first described by Blaauw et al. (2007) and is called a ghost map.

These new functions inherit the uncertainty of G and are therefore, also stochastic. It is also worth noting that these functions may behave very differently depending on what proxy is being dated. Some proxies are multidimensional, some are categorical, and some are nonzero only in a very narrow range of depths.

We will assume that all proxies of interest may be measured at any depth. Let $p = (p_1 \dots p_l)$ be the vector of proxies of interest, and let Y denote the information used by BACON (the carbon 14 measurements as well as the depths at which the measurements take place). We intend to move from a function $G : D \to A$ from depth to age into a set of functions $F_i : A \to P_i$ from age to proxy i. It is of interest to estimate the distribution of $F_i(a)|Y$. for arbitrary a and i.

It is worth noting that G is assumed to be parametrized by θ, x and piecewise linear, while F_i is not parametric.

For ease of notation, we will now consider a single proxy at a time, so $F_i = F$. Also we will consider all proxies as one-dimensional. If a proxy is multidimensional, each dimension will be treated as a separate proxy.

3.4 The algorithm

BACON generates samples which are distributed approximately iid$\sim \pi(\theta, x|Y)$. We will assume that this distribution is exact.

Let $q(d)$ denote the measurement of a proxy at depth d. Let $A = (A_1 \dots A_m)$ be a vector of finitely many ages of interest at which we wish to estimate p. The technique will be to generate a large sample $\sim \pi(F(A)|Y)$ using BACON.

The algorithm to generate p_k, the kth element of a sample from the target distribution, is the following:

1. Generate $(\theta^{(k)}, x^{(k)}) \leftarrow \pi(\theta, x|Y)$ using BACON
2. Compute the depth$\gamma^{(k)} = G^{-1}(A)|\theta^{(k)}, x^{(k)}$
3. Set $p_k = q(\gamma^{(k)})$

3.5 Why the algorithm works

We intend to prove that for a given proxy ρ, we have $p_k \sim \pi(\rho|Y, Z, A)$. An important assumption is needed, and that is that (θ, x) and proxies are independent when we condition on Y, or equivalently that $G(d)|Y, Z \overset{d}{=} G(d)|Y$ where Z is all the information relevant to the depth distribution of proxies. While this assumption is often not true, it is in keeping with the methodology of BACON since proxy information is ignored when estimating $\pi_{post}(\theta, x)$.

Consider the i^{th}entry of p_k:

$$P(p_{k_i} \leq \beta|\theta^{(k)}, x^{(k)}, Y) = \int_{-\infty}^{\beta} \pi_{p|\theta,x}(t|\theta^k, x^{(k)}, Y)dt$$

(In this case this integral is of a degenerate distribution: Conditioning on $(\theta^{(k)}, x^{(k)})$ is equivalent to conditioning on $\gamma^{(k)}$)

Then:

$$P(p_k \leq \beta) = \int_{-\infty}^{x} \int \pi_{p|\theta,x}(t|u, v, Y)\pi_{\theta,x}(u, v, Y)\, d(u, v)\, dt$$

by the law of total probability. This quantity is equal to

$$P_{\rho|Y,Z,A_i}(\rho < \beta)$$

because of our assumption that the proxy does not affect $\pi_{post}(\theta, x)$.

3.6 Implementation

A program was written in in R to execute the algorithm using the standard output from BACON (either version). There are three points worth noting about the way the program works:

1. Although the stated assumption to develop the algorithm is that proxies may be measured at any depth, what is actually available to the program is only a finite set of depths and their respective proxy measurements. The algorithm requires $q(\gamma^k)$ which is the value of proxies measured at an unpredictable depth. Since this measurement is (almost surely) unavailable, the value that is used is calculated as if proxies vary linearly between the depths at which they are observed. This will yield a reasonable approximation in most cases, but it is unreliable when the differences between the depths of measurements are large. On some occasions, problems with discrete proxies are ameliorated by the histograms separating into bins assigned by integer values (which is the default), but sometimes it is best if a different estimation algorithm is used for the value of proxies.

2. Although the program generates the full sample, the graphical presentation of the data only uses a small amount of the information

available. It calculates a histogram for each age of interest, but since each p_k is a vector which corresponds to the vector of ages of interest, and comes from a single simulation of (θ, x) it is actually possible to use the information from the sample to estimate probabilities such as $P((F(A_m), F(A_n)) \in B|Y, Z)$ for any measureable $B \subseteq R^2$. This can often be very useful since it allows us to understand the evolution of proxies over time, for instance estimating the probability of a proxy increasing between two time periods.

3. The original BACON program has a ghost mapping function, which produces graphs which relate proxies to time. These graphs are not the product of any mathematically sound reasoning, and are unable to produce any estimations relating to the probability of proxy values at any time. Nonetheless, the graphs are intuitively very easy to intepret, and roughly match the results of this algorithm. The graphs produced by the ghost mapping function developed here, while mathematically correct, are less clear. For this reason, the implementation developed here is not a complete replacement for the original function.

3.7 Results and observations about graphical parameters

BACON was used to estimate the age-depth function of the MSB2K peat core. Ages of interest were designated every 100 years from 4800 years ago to 6600 years BP. The function computes a histogram at each age of interest and renders the histograms vertically in greyscale above the year which corresponds to each. Slightly less than 1 minute of computation (on a relatively low powered netbook using the Rstudio software package) yields figure 3.1.

We can see that each year, the probability of the proxy being zero is reasonable. We may also observe that the likely nonzero region for the proxy is concentrated between the years 5800 and 6500 BP. We can now easily refine our range of interest to pay particular interest to where the proxy is likely to be nonzero, making a histogram every 5 years in the range 6000 to 6400 BP. The result can be seen in figure 3.2.

This image alone actually gives us quite a lot of information regarding the distribution of the proxy over time.

Of course, for actual estimation of probabilities, the sample itself must be

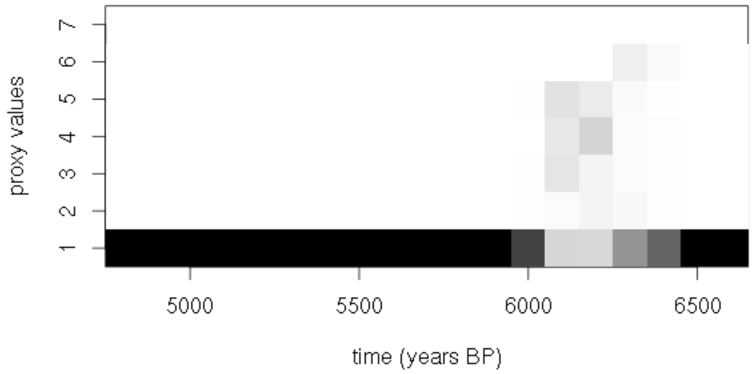

Figure 3.1: A ghostmap for proxy 1 on the MSB2K peat core: Each vertical cross section is a greyscale histogram for an age of interest.

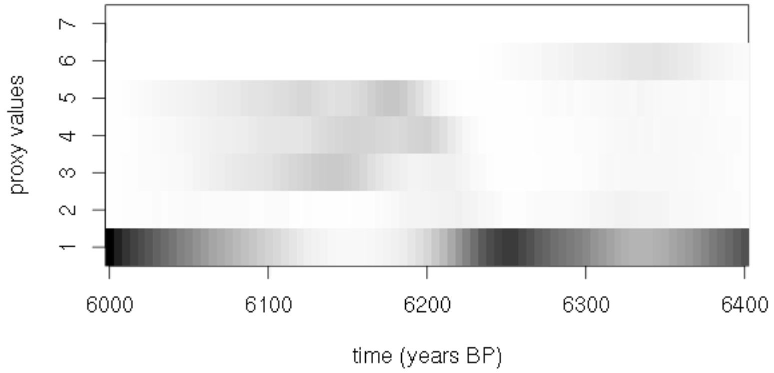

Figure 3.2: Proxy 1 examined with more ages of interest, spread over a smaller range

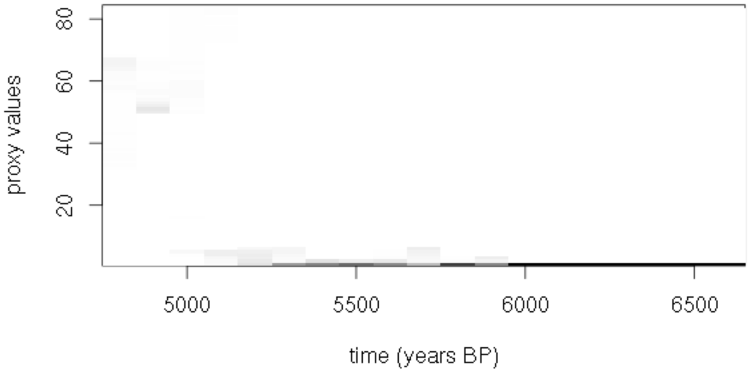

Figure 3.3: A ghostmap from proxy 4 over the same core. In this case proxy dispersion makes the greyscale nearly invisible.

used. One simple example is the calculation of the probability that 6100 years before present the proxy had value zero, which is estimated at 1103/3376 or about 0.327.

Ghost maps have proven to be very useful for estimating the behaviour of proxies over time. Nonetheless, the usefulness of the graph might be sensitive to the exact values of the graphical parameters used:

A test of this program on proxy number 4 with the same settings that it was originally run on for proxy 1 generates figure 3.3

As we can see, this proxy behaves very differently. The graph automatically adjusts its height and the number of bins to compensate for a proxy which has much greater dispersion than the previous. The program will only have problems creating the graph if all observed values of the proxy are within a unit of each other, in which case either the number of bins in the histograms or the scale in which the proxy is measured can be adjusted. This may also be desireable for proxies where the dispersion is too great and the graph becomes difficult to see. An adjustment was made to the scale of figure 3.3. This produces figure 3.4, which is much easier to intepret.

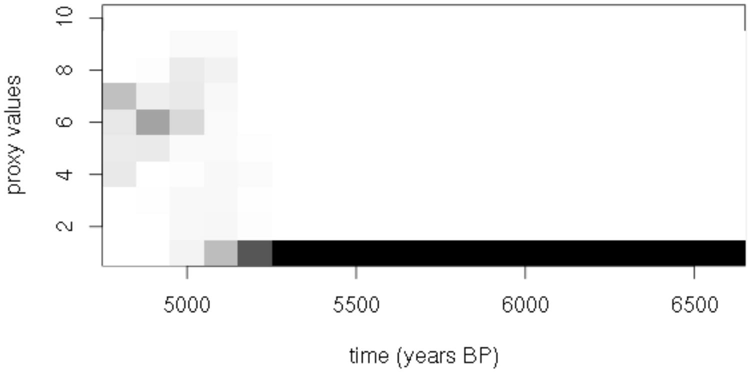

Figure 3.4: The same graph as figure 3.3, but the proxy has been re-scaled. Now the graph easier to understand

Chapter 4

BLT0: A restricted form of BACON

4.1 The objective of BLT

4.1.1 Limitations of BACON

While BACON is very useful for dating cores which meet certain conditions, these conditions are not trivial. For starters, peat cores with sufficient datable material are only found in high lattitudes, and the quality of the information produced by BACON suffers greatly if there are not sufficiently many measurements.

The cost of ^{14}C measurements can itself be prohibitive: Even in situations where peat cores are plentiful and material from which to draw ^{14}C samples is readily available, not all cores will have the necessary information about each proxy of interest. In this situation, it may be necessary to perform BACON on several different cores in order to obtain all of the information of interest. The resulting multiplication of the required^{14}C measurements can dramatically increase the cost of paleoecological studies, making the entire process prohibitively expensive.

Occasionally, it may also happen that BACON has the opposite problem: There is plentiful information available about the chronology of a certain core, but much of this information is not encoded into the BACON model. While BACON still performs well in these situations, it is clearly not the optimum solution.

4.1.2 Tephras and proxy information

One way to improve BACON is to incorporate additional information into the model. When such information is directly available, this is an obvious way to improve inference (see section 4.4), but often the information that can be used is not so direct.

The main source of external information for BACON comes from tephras. Tephras are layers of ash created by volcanic eruptions, and which experts are able to identify in the strata of peat cores. The reason tephras are useful is because they can be identified. The ash from each volcanic eruption is different, and studying the ash from each tephra will in certain situations allow paleoecologists to identify which eruption generated it. In this situation, any information known about the time of the eruption can be used for that layer of sediment.

The second source of external information can be found in the proxies themselves. While BACON is used to obtain information about the evolution of proxies over time, this information does not exist within a void. Sometimes outside information exists about the evolution of proxies. The use of proxies for dating is very prevalent in paleoecology and has been named "tuning". The direct use of tuning to build chronologies has yet to receive adequate statistical treatment and is not recommended, but the information obtained from proxies is nonetheless useful, and some proxy information can be encoded into BACON.

4.1.3 Information from correlation

The information available from tephras and proxies is seldom direct. Identifying the volcanic eruption which produced a tephra is only immediately informative if a historical date for the eruption is already known. A more common situation is that tephras from different cores can be identified as being produced by the same eruption. What can be inferred from this situationis is that the strata in these cores are the same age. Thus, any information which BACON can give us about the age of the strata in one core is also useful for the others.

A similar situation is found in proxies. In geographically proximal cores, certain cataclysmic events can occur which drastically alter the evolution of proxies. While gradual trends in proxies may not be reliable as a source of information for chronology building, sudden cataclysmic change can often be attributed to ecologically significant events. It is sometimes possible for experts to identify these events by direct study of the proxies at different depths.

Both of these forms of information are indirect, in the sense that they do not in and of themselves produce any information about the age of strata, but they are useful in that they indicate correlations between the ages of depths in different cores. The main improvement which is possible to the BACON model is to use the correlation between the chronologies of different cores to improve our estimations of all of them. This extension of the BACON model to link the timescales of different cores, is called BLT (Bacon Linking Timescales).

4.2 Changes to the BACON model

4.2.1 The simplest idea

The primary characteristic of the information used by BLT is that BACON is improved by using other instances of BACON. One naive way to attempt to incorporate this information is to use the posterior estimation from BACON at a given depth in one core as a prior for the corresponding depth in others. The issue with this approach is that it treats the information as if it only improved estimations in one direction. When we identify strata in different cores, the information is useful to improve estimation in all of the cores involved.

4.2.2 Revising the model

A better approach is to revise the BACON model so that it does not treat different cores as independent. This way, rather than estimating the posterior one core at a time, the posterior distribution can be estimated for all cores of interest simultaneously. Now instead of having only (θ, x, w) we have instead $\{(\theta, x, w)_i\}, P$: one set of parameters for each core, and P : a vector of parameters which contains all information relating to the correlation between cores, supposing that $(\theta, x, w)_i|P$ and $(\theta, x, w)_j|P$ are independent.

The exact nature of P will be different depending on the type of correlation we wish to consider, but in general all forms of BLT will have a parameter space defined in this way.

4.2.3 Handling the larger parameter space

When we change our parameter space from (θ, x, w) to $\{(\theta, x, w)_i\}, P$ the number of parameters increases dramatically, but for the ith core it is still possible to work with $(\theta, x, w)_i|P, Y$ using a very similar structure to the original BACON.

This is the key characteristic of BLT models which allows us to simulate from the posterior, and may also permit the use of parallel computing. Modern computers often use more than one processor, and this allows different processors to work on separate tasks. While the primary purpose of this sort of structure is to allow the computer to function normally even when a process requires a very heavy load, we can take advantage of this structure to divide BLT among separate processors.

Essentially, when working with k cores we will require k simulations of $(\theta, x, w)_i|P, Y$. Since the various cores are independent, we can then consider the resulting simulation to come from $\{(\theta, x, w)_i\}|P, Y$ for any given P. This is not exactly the posterior distribution we are looking for, since P is unknown, but when coupled with a procedure which changes the value of P, it is possible to simulate from the full posterior distribution.

4.3 BLT0: How to restrict BACON

4.3.1 BLT0

The objective of this chapter is to use MCMC techniques to simulate a large sample from the conditional posterior distribution $(\theta, x, w)_i|P, Y$. This is not actually a proper form of BLT since it does not link any timescales and still works one core at a time. It is, however, an essential ingredient for any form of full scale BLT.

The process of chronology building when conditioning on outside information P is actually just a specialized form of BACON, but its relationship to BLT is very strong. The process will be called BLT0.

4.3.2 The condition P

In the context of BLT0, conditioning on P is not based on correlation, since there is no other core. While in some circumstances P can contain distributional information relating to (θ, x, w), for the forms of BLT treated in this work, P will take the form of a restriction. We will fix $\pi(\theta, x, w)|P, Y = 0$ if the restriction P is not met. There are two kinds of restrictions that we will use for BLT:

1. Exact restrictions: $G(d, \theta, x) = a$ for a specific value of d and a.
2. "Fuzzy" restrictions: $G(d_1, \theta, x) \leq a \leq G(d_2, \theta, x)$ for specific values of d_1,d_2, and a.

Fuzzy restrictions are only used for BLT3, and are actually much easier to handle, so they will be treated in the corresponding section. This chapter focuses on exact restrictions.

4.4 Why BLT0 is useful on its own

There are occasions when information is known about a chronology which is completely separate from BACON. Certain tephras are known to correspond with volcanic eruptions with precisely known dates. While most dates for eruptions are estimated using ^{14}C and other radiometric techniques, some eruptions (particularly more recent ones) are historically documented, and the corresponding tephra has a precise known age.

This sort of information does not require the linking of timescales to be incorporated into the BACON model. If a tephra is found at a depth d_0 with a known age a_0, then the extra information can be encoded as simply $G(d_0, \theta, x) = a_0$, which is nothing other than an exact restriction on BACON of the form studied by BLT0.

BLT0 is useful not only as a stepping stone to develop other forms of BLT, but is actually on its own a useful technique for chronology building in some special circumstances.

4.5 Sampling from the posterior distribution

4.5.1 The model

The model for BLT0 will be exactly the same as the model for BACON, with the added restriction from P. The posterior distribution will be normalized over a (significantly) smaller support, but the likelihood and priors are all proportional to the likelihood and priors used in BACON.

4.5.2 Why using t-walk directly does not work

Given that the model is mostly unchanged, one might think that all that is required is to adjust the support function given to the t-walk algorithm used to sample from the posterior distribution. For the case of fuzzy restrictions, this is indeed the solution, but for exact restrictions the technique is ineffective.

Note that the set of parameters (θ, x, w) such that $G(d_0, \theta, x, w) = a_0$ has Lebesgue measure 0. The $t-walk$ algorithm simulates from the posterior by using a Markov chain, which means that the next element of the simulation is proposed at random and then accepted or rejected using the posterior (See next chapter for a full description of the Metropolis-Hastings algorithm), but the probability of randomly choosing an element from the restricted support is 0, and hence every proposed change will be rejected.

4.5.3 Changing the parameter space

Let l be the number of piecewise linear segments and let $i < l$ be such that $c_i < d_0 < c_{i+1}$, the index of the segment that includes d_0. We note that, conditioning on P and on $\theta, (x_0 \dots x_{i-1})$ there is a unique value for x_i which satisfies $G(d_0, \theta, x, w) = a_0$, and this unique value can actually be calculated as

$$x_i = \frac{a_0 - G(c_i)}{d_0 - c_i}$$

while $x_{i+1} \dots x_l$ remain free.

We observe that x_i is no longer a free parameter of the model, and hence should not be considered in the parameter space. While its prior distribution still exists, it can now be implicitly viewed as part of the prior distribution of $(\theta, x_0 \dots x_{i-1}, x_{i+1}, \dots x_l, w)$.

We can now view the parameter space as reduced by one dimension for each exact restriction we wish to add (It is also worth noting that it is not possible to add more than one exact restriction in the same piecewise linear segment).

4.5.4 The algorithm

As mentioned before, it is not possible to use the t-walk to directly sample from the posterior distribution, the primary issue being that the parameter space is now $(\theta, x_0 \ldots x_{i-1}, x_{i+1}, \ldots x_l, w)$, with x_i determined by the other parameters. In this case, we will call f_1 the density of the posterior distribution of $(\theta, x_0 \ldots x_{i-1}, x_{i+1}, \ldots x_l, w)$ from which we intend to sample. Rather than using the t-walk to sample from f_1, we will instead sample from a second distribution with density f_2, which has parameter space (θ, x, w) and for which f_1 is the correct marginal distribution.

In order to do this, the support and energy functions are modified by taking (θ, x, w) and ignoring the value of x_i. Instead a new set of parameters is created:

$$\left(\theta, x_0 \ldots x_{i-1}, \frac{a_0 - G(c_i)}{d_0 - c_i}, x_{i+1} \ldots x_l, w\right)$$

and the usual BACON support and energy functions are calculated for this vector of parameters.

The standard accept and reject technique is applied with the proposed x_i being accepted or rejected as well.

After BLT0 terminates and a sample is produced, some post-processing is requried since each x_i in the sample must be changed for the correct x_i before any analysis is done.

4.6 Why the algorithm works

We intend to prove that when following this algorithm, the marginal density of the sample does is in fact converge to f_1.

Let $f_2^{(j)}$be the density of the jth sample accepted by BLT. Note that this sample was accepted using a marginal distribution $f_1^{(j)}$, so in other words

$$\int f_2^{(j)}(\theta, x, w) f_{x_i}^{(j)}(x_i) dx_i = f_1^{(j)}(\theta, x_0 \ldots x_{i-1}, x_{i+1}, \ldots x_l, w)$$

Now observe that $f_1$has a compact support and is continuous over this support. Then f_1 is bounded, and so is each $f_2^{(i)}$ and each $f_1^{(i)}$. Now all of these functions can be bounded above by a constant function g over their support. Since the support has finite measure (it is bounded in every direction) then g is Lebesgue-integrable.

Now note that $f_2^{(j)}(\theta, x, w) \overset{j \to \infty}{\to} f_2(\theta, x, w)$.

By the dominated convergence theorem, we can now see that

$$\int f_2^{(j)}(\theta, x, w) f_{x_i}^{(j)}(x_i) dx_i \overset{j \to \infty}{\to} \int f_2(\theta, x, w) f_{x_i}(x_i) dx_i$$

and

$$\int f_2^{(j)}(\theta, x, w) f_{x_i}^{(j)}(x_i) dx_i = f_1^{(j)}(\theta, x_0 \ldots x_{i-1}, x_{i+1}, \ldots x_l, w)$$

(as stated above) and hence

$$f_1^{(j)}(\theta, x_0 \ldots x_{i-1}, x_{i+1}, \ldots x_l, w) \overset{j \to \infty}{\to} \int f_2(\theta, x, w) f_{x_i}(x_i) dx_i = f_1(\theta, x_0 \ldots x_{i-1}, x_{i+1}, \ldots x_l, w)$$

which is what we intended to prove.

4.7 Results and discussion

A program called BLT0.py was written in Python using the above technique. The results are as expected, and the efficiency is similar to what we obtained from BACON.py.

For purposes of comparison, BLT0.py was run on the MSB2K peat core using $d_0 = 20cm$ and $a_0 = 4850$ years BP. This is not known to be the case, but seems like a perfectly plausible age for this depth, as suggested by our previous runs with BACON.

As one might expect, estimations using only 50,000 iterations are somewhat crude and untrustworthy, as can be seen in figure 4.1

With 1000,000 iterations, the posterior distribution becomes clear, as seen in figure 4.2. It is now possible to compare the distribution using BLT0 to the distribution using BACON (figure 2.4 initially, presented again as figure 4.3).

The good news is that knowing the age at a single depth is actually very informative. While it might be expected that only depths very close to 20cm would be affected, the uncertainty becomes visibly reduced from depths of 0 through 50 cm. And estimations of the standard deviation are lower throughout the entire depth of the core.

Overall, the effects of outside information on the chronology are very beneficial. Including a single tephra of known age is enough to dramatically reduce uncertainty and improve the accuracy of estimations.

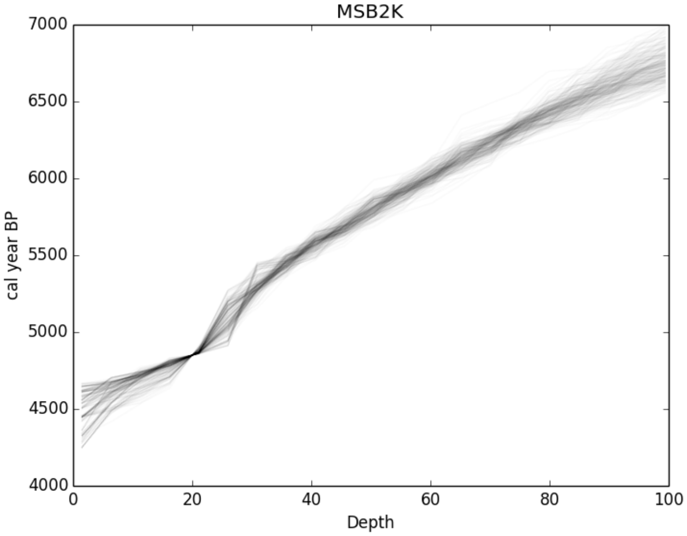

Figure 4.1: BLT0 tracing one iteration every 100 from 20,000 to 50,000 iterations on the MSB2K peat core. At a depth of 20 cm. we have placed a 4850 year old tephra. This tephra does not actually exist, and is used only to test the program.

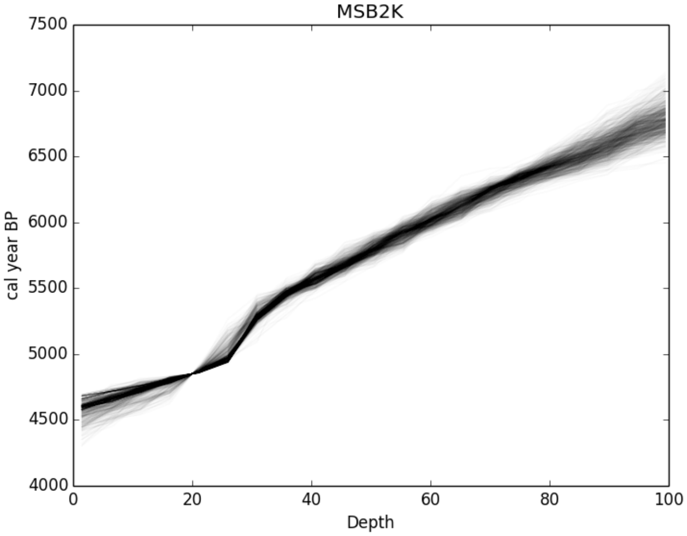

Figure 4.2: BLT0 tracing one iteration every 1000 from 50,000 to 1,000,000. This is the equivalent of the Python version of BACON (see next figure).

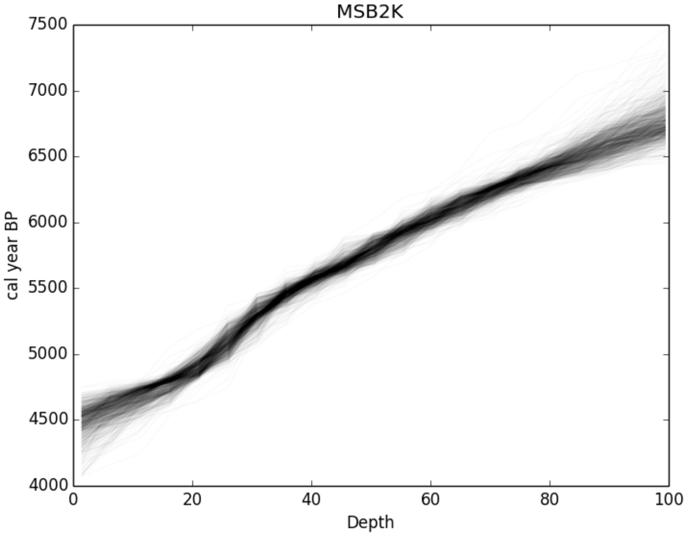

Figure 4.3: The same plot using BACON, same as figure 2.4. From the comparison, we can see that BLT0 provides less dispersion not only at the tephra, but throughout the graph.

Chapter 5

BLT

5.1 Outline

5.1.1 Overview

One of our primary interests is to extend the scope of BACON by incorporating information not included in the original model. For any phenomenon, there is no model which will adequately incorporate all of the information that is available, nor is it necessarily desirable to incorporate all such information, even if it were possible.

BACON is a reasonably popular program that has seen use by paleoecologists for a number of different purposes. It is therefore worth investigating whether incorporating further information into the model would improve the results of the program. The types of information which it is possible to include in BACON are varied, and the choice of what information to try is by no means trivial.

The main source of what information to add to the BACON model is to directly consult expert opinion. Specifically, that of paleoecologist Dr. Maarten Blaauw, who participated in the initial development of BACON and who has spoken to numerous paleoecologists on the subject. Dr. Blaauw's suggestions primarily centered around information from tephras and proxies, as discussed in section 4.1.

Of the two classes of information, the information available from proxies is much more difficult to manage. While tuning (using proxies directly to build chronologies) is a common practice in paleoecology, it is not – at present – supported by sound mathematical theory, and it is difficult to quantify the uncertainty in estimations which rely on it. Because this information is mathematically intractable, only a very specific case of proxy information has been attempted (see BLT3, section 5.6).

The information gathered from tephras is far more reliable, and incorporating data from tephras has been explored in three different ways. The first of these ways is BLT0, or restricted BACON, which is discussed in chapter 4, and which can be seen to significantly improve on the results from BACON in a situation in which a specific kind of information is available. In this chapter we will explore the use of information from tephras in the much more common situation in which the information required for BLT0 is unavailable.

While in practice it may well be the case that various forms of information are available at different depths of the same core, for the -primarily exploratory- purposes of the present work, we will consider each form of information separately. This will allow us to appraise the usefulness of adding each type of information into the model.

5.1.2 The general BLT model

The information that will be incorporated into BLT is obtained from correlations between different cores. We will suppose that there are l cores for which we intent to construct a piecewise linear chronology using k segments. For the jth core, the ith segment will have a slope $x_i^{(j)}$, the age of the uppermost stratum will be $\theta^{(j)}$, and the core will have a memory parameter $w^{(j)}$. The entire set of parameters corresponding to the jth core will be written as $\Theta^{(j)}$, and the set of l such sets of parameters will be written simply as Θ.

Along with Θ, there is a second form of paramter which contains all of the information pertaining to the correlation between cores. This information will vary between the forms of BLT, but for general purposes, it will be written as P. Our model will suppose that $\Theta^{(j)}|P = p$ and $\Theta^{(m)}|P = p$ are independent for any $j \neq m$. The full parameter space is (Θ, P).

The model for $F_{\Theta^{(j)}|P}(\Theta^{(j)}|P)$ (prior and posterior) is exactly BLT0 for the case of BLT1 and BLT2, but for BLT3 a different model will be used. The model for $F_P(P)$ will be different for each form of BLT. The four forms of BLT are summarized in table 5.1.

Name	Type of correlation	P	Restriction
BLT0	None	d_0, a_0fixed	$G(d_0) = a_0$
BLT1	Exact	$d_0^{(i)}$fixed, $\theta_0$unknown	$G^{(i)}(d_0^{(i)}) = \theta_0$
BLT2	Candidate	$d_{(0,j)}^{(i)}$fixed i_0, θ_0unknown	$G(d_{(0,i_0)}^{(i)}) = \theta_0$
BLT3	Fuzzy	$d_0^{(i)}, d_1^{(i)}$fixed, $a_0$unknown	$G^{(i)}(d_0^{(i)}) \leq a_0 \leq G^{(i)}(d_1^{(i)})$

Table 5.1: The forms of BLT

5.2 The Metropolis-Hastings algorithm

5.2.1 Limitations of the t-walk

The t-walk algorithm, which was used for BACON and BLT0, is a general purpose algorithm which can be used to sample from any form of probability distribution for which the support and energy functions can be calculated. The efficiency of this algorithm for sampling from a posterior distribution has been found to be very high for certain types of distributions (see Christen and Fox, 2010), but its very generality is its primary limitation: The t-walk does not take advantage of any structural information which may be available about the target distribution. In the case of BACON, very little was known about the posterior distribution of the parameters, and hence it was appropriate to use the t-walk directly. The same cannot be said about BLT, wherein a form of conditional independence has been incorporated into the posterior distribution by design. If possible, in this situation much can be gained by using an ad-hoc MCMC algorithm, specifically designed to take advantage of the structure of the posterior distribution.

5.2.2 MCMC in general

The general objective of MCMC is to produce a sample from a target distribution with density $p(x)$. It is possible that p is known explicitly, but it is often the case (especially in Bayesian statistics) that what is known is a function f which is proportional to p but which is not normalized (does not integrate 1). In this case

$$p(x) = \frac{f(x)}{\int f(x)dx}$$

but this can only be calculated if it is possible to calculate $\int f(x)dx$, which is frequently impossible.

The technique used to generate a sample of the target distribution is to propose a Markov chain over the support which has p as its asymptotic distribution. In other words, if $S(i)$ is the state of the Markov chain after i iterations, we have

$$P(S(i) \in A) \overset{i\to\infty}{\to} \int_A p(x)dx$$

and hence, if L is chosen to be sufficiently large, then $S(L)\dot{\sim}p$.

It is particularly desirable that if k is also sufficiently large then $S(L)$ and $S(L+k)$ are approximately iid $\dot{\sim}p$. If this is the case, then an MCMC algorithm will produce an arbitrarily large sample from p in the following way:

1. Begin the algorithm at any point $S(0)$ within $supp(p)$.

2. Simulate the Markov Chain for a sufficiently long time L and take $X_0 = S(L)$.

3. Continue to simulate the Markov chain for as long as is required, taking $X_i = S(L+ik)$ for k sufficiently large, and $i = 1, 2, 3, \ldots$ as large as is desired.

4. Take the sample $\{X_i\}$ as an approximately iid sample $\dot{\sim}p$

This algorithm only makes sense if $k \leq L$ (otherwise just repeat steps 1 and 2), and it is reasonable to expect that this will be the case so long as $a \to b$ in the Markov chain for any a and b in $supp(p)$.

It is worth noting that in most cases, the entire path of the Markov chain $S(a)$ for $a = \{L, L+1, \ldots\}$ is useful for most estimations, even if it cannot be treated as an iid sample from p.

The various forms that MCMC algorithms take are variations on the Markov chain with the desired asymptotic distribution.

5.2.3 The M-H algorithm

The Metropolis-Hastings algorithm is one variation on MCMC which has been found to yield good results in situations where f is known and p is not.

For each point a in the support, a distribution (known as the instrumental distribution at the point a) is required with density q_a from which an efficient simulation algorithm is known. Some further requirements are placed on $q = \{q_a | a \in supp(p)\}$ which will be discussed in the next section. Despite being a set of distributions, q is called the instrumental distribution.

Now let $a = S(i)$ be the state of the Markov chain at time i. The Metropolis-Hastings algorithm will produce $S(i+1)$ in the following way:

1. Simulate $b \leftarrow q_a$ $\qquad$ *"propose a move to b"*

2. Let $S(i{+}1) = \begin{cases} b & with\,probability\,\rho_i = max(1, \left(\frac{q_b(a)}{q_a(b)}\frac{p(b)}{p(a)}\right) & "b\,is\,accepted" \\ a & otherwise & "b\,is\,rejected" \end{cases}$

Now observe that $\rho_i = max(1, A_i)$ where

$$A_i = \frac{q_b(a)}{q_a(b)}\frac{p(b)}{p(a)} = \frac{q_b(a)}{q_a(b)}\frac{\frac{f(b)}{\int f(x)dx}}{\frac{f(a)}{\int f(x)dx}} = \frac{q_b(a)}{q_a(b)}\frac{f(b)}{f(a)}$$

is known as the Metropolis-Hastings quotient and can be calculated without knowing the value of $\int f(x)dx$. Hence $log(\rho_i)$ can be obtained from the ability to calculate $q_a(b)$ for arbitrary a and b in $supp(p)$ as well as the support and energy functions.

5.2.4 Requirements for the instrumental distribution

In order for the M-H algorithm to produce a Markov chain which converges to the target distribution, the instrumental distribution q must satisfy the following requirements:

1. For a and b in $supp(p)$ there must be a finite sequence $\{c_1, c_2, c_3 \ldots, c_k\}$ with $q_a(c_1)$, $q_{c_i}(c_{i+1})$ (for all i) and $q_{c_k}(b)$, as well as $p(c_i)$, all greater than 0.

2. For a and b in $supp(p)$ if $q_a(b) > 0$ then $q_b(a) > 0$.

3. The resulting Markov chain must be aperiodic. For our purposes we will instead ask for a stronger condition, which is that the Markov chain be strongly aperiodic, which means that $\forall a \in supp(p)$, $P(S(i+1) = a|S(i) = a) > 0$, or in other words, that rejection is possible.

For all of the M-H algorithms used, these three properties will be self-evident.

5.2.5 Hybrid kernels

The transition kernels of the resulting Markov Chain are referred to as those induced by the instrumental distribution used. Let $q^{(i)}$, $i \in \{1, 2, \dots k\}$ be a finite set of instrumental distributions. The kernel induced by $q_a = q_a^{(j)}$ *with probability* $\pi^{(j)}$ is called the hybrid kernel induced by $\{q^{(i)}\}$and $\{\pi^{(i)}\}$so long as $\pi^{(i)}$are all greater than zero and $\sum_i \pi^{(i)} = 1$.

Note that one way to execute the M-H algorithm from a hybrid kernel is to simulate a discrete random variable with probabilities $\{\pi^{(i)}\}$ and then perform an M-H step using the appropriate instrumental distribution.

Also note that not all of the $q^{(i)}$ need to satisfy properties 1 and 2 from the previous section. It is enough for one of the kernels to satisfy each property, or enough for one kernel to satisfy property 2 and a set of them together to satisfy property 1 (That all states in the support can communicate).

5.2.6 The Gibbs kernel

One particular kind of kernel which is frequently useful to construct a hybrid kernel is called a Gibbs kernel. Let $\theta = (\alpha, \beta)$ be a point in $sup(p)$ with α and β some partition of the dimensions of θ. The gibbs kernel is the one induced by the instrumental distribution:

$q_{(\alpha,\beta)}(x, y) \propto p_{A|B}(x|B = \beta)I_{y=\beta}$

This can be interpreted as proposal to keep some parameters fixed and then condition on these and change the rest of them using the conditional distribution.

The practice of fixing some parameters and moving others is very common in Metropolis-Hastings algorithms using hybrid kernels, but the Gibbs kernel is particularly interesting because of the value of the Metropolis-Hastings quotient. The probability of accepting a proposal to change state from (α, β) to (γ, β) is the maximum of one and

$$A_i = \frac{q_{(\gamma,\beta)}(\alpha, \beta)}{q_{(\alpha,\beta)}(\gamma, \beta)} \frac{p(\gamma, \beta)}{p(\alpha, \beta)} = \frac{p_{A|B}(\alpha|B = \beta)I(\beta = \beta)}{p_{A|B}(\gamma|B = \beta)I(\beta = \beta)} \frac{p(\gamma, \beta)}{p(\alpha, \beta)} = \frac{\frac{p(\alpha,\beta)}{p_\beta(\beta)}}{\frac{p(\gamma,\beta)}{p_\beta(\beta)}} \frac{p(\gamma, \beta)}{p(\alpha, \beta)} = 1$$

so the proposal is always accepted. The difficulty in using this algorithm is that simulating directly from this instrumental distributionis not always possible.

5.3 The data

5.3.1 The datasets that were used

For all of the examples in this chapter, the datasets used were the Deadisland and Slieveanorra2 peat cores. These are two highly correlated cores, which have several tephras which are known to correspond. There are no instances of correlations of the type used in BLT2 (see section 5.5) or BLT3 (see section 5.6), but it is not difficult to find reasonable settings to test these programs with. It is extremely useful to use the same datasets in each version in order to adequately compare the performance of different versions.

5.3.2 Building chronologies for these cores using simple BACON

In order to get a proper notion of the effect of the correlations between these cores on the inference about their chronologies, it is important to observe what chronologies are obtained without considering the correlations. Figure 5.1 is the result of running the Python version of BACON using the Deadisland dataset for one million iterations. Figure 5.2 is the result using the Slieveanorra2 dataset.

While these graphs on their own are of little significant interest for our purposes, there is a salient feature worth mentioning, which is the area around 230 cm deep in the Deadisland core, and around 150 cm deep in the Slieveanorra2 core. The uncertainty in these points is significantly less than in the rest of the core, and the strata at these depths are around 3000 years old. These correspond to a tephra present in both datasets and which will be used for several of the examples below. Since this tephra has been dated previously and elsewhere, the data relating to its age is more precise.

5.4 BLT1

5.4.1 The model

The first form of BLT that will be approached is known as BLT1. The correlation that we are interested in is the case in which a single tephra of unknown age has been identified accross numerous cores. This situation is

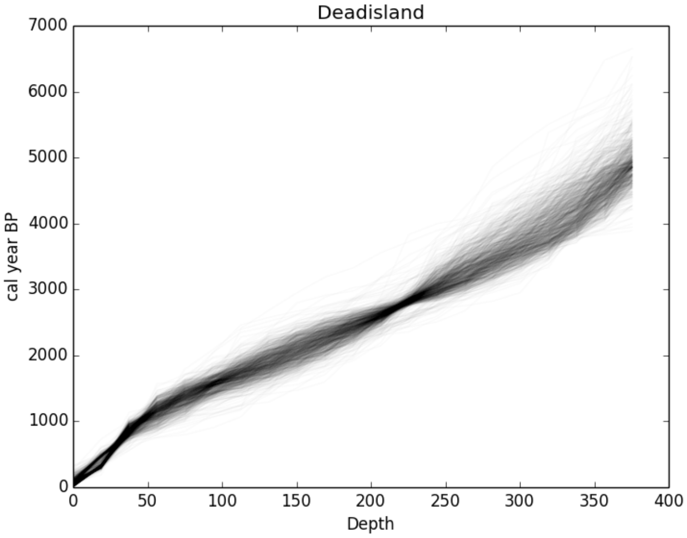

Figure 5.1: Deadisland chronolgy tracing one iteration every 1000 from 50,000 to 1,000,000 using BACON.

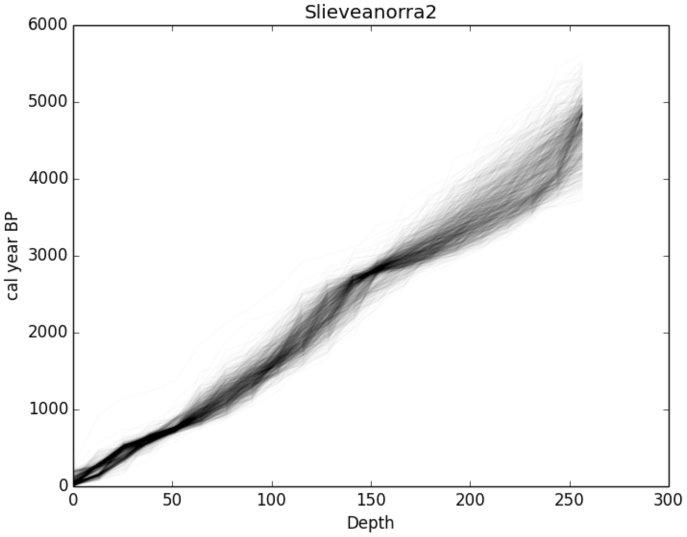

Figure 5.2: The chronology of the Slieveanorra2 core, built independently. Along with figure 5.1, these will serve to compare the results of BLT.

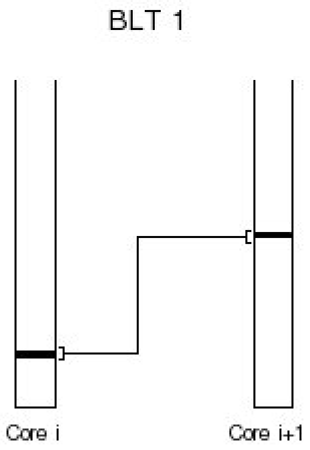

Figure 5.3: The situation modeled in BLT1: Strata in two cores share a corresponding tephra of unknown age.

depicted by figure 5.3: The strata at the corresponding heights for this tephra can be considered to be the same age as one another. The strength of this model relative to BLT0 is that we do not need an exact historical date for the tephra.

In this case, we build on the general BLT model (section 5.1.2) by representing $P = \theta_0$ as simply the age of the tephra. In this case the conditional posterior distribution $\pi(\Theta^{(j)}|\theta_0 = p)$ is precisely the BLT0 model with $a_0 = p$ and d_0 the height of the tephra in the jth core.

The distribution of θ_0 has an arbitrary prior $\pi_0(\theta_0)$ which may contain prior information relating to the age of the tephra, or may simply be over the support (as is the case in the examples presented). The value of θ_0 also has an effect on the priors for all of the other parameters $((\theta, x, w)^{(j)}$ for each $j)$ and on the likelihood, as it is the value of $G^{(j)}((d_0, x, w)^{(j)})$ but this effect is only on the values, and has no effect on how they are computed.

5.4.2 The algorithm

In order to draw a sample from a posterior distribution of a BLT1 model, a Metropolis-Hastings algorithm was constructed, using a two part hybrid kernel. The weights of these kernels cannot be chosen universally, as the effectiveness depends on the number of cores, and the number of segments in each core: Since this is not a t-walk algorithm, some calibration must be done manually.

The first kernel is simply BLT0 performed on each of the cores with the ith iteration of the algorithm setting $a_0^{(j)(i)} = \theta_0^{(i)}$ for 100 iterations (which may be changed if the structure of the cores demands it, but 100 has proven to be reasonable in most situations when testing). 100 iterations is not nearly enough for the result to be considered a sample from the posterior conditional distribution, so this is not a Gibbs kernel, but BLT0 is an instance of t-walk, and the t-walk is precisely a Markov chain with a Metropolis-Hastings transition kernel.

The second kernel takes $(\Theta, \theta_0)_i$ and proposes $(\Theta, \theta_0)_{i+1}$ as $(\Theta_i, \theta_{0,i+1})$ with $\theta_{0,i+1}$ distributed uniformly in the range of dates which would continue to be in the support of $\pi^{(j)}(\theta, \bar{x}, w)^{(j)}$ for all cores j, where $\bar{x} = (x_1, \ldots x_{d_j-1}, x_{d_j+1}, \ldots x_k)$ and d_j is the index of the segment wherein the tephra is located in the jth core (note that x_{d_j} is uniquely determined by the other parameters, see section 4.5.3). The Metropolis-Hastings quotient is calculated using the same energy function which is given to the t-walk for the first kernel. Since this second kernel is not a gibbs kernel either, rejections can – and do – occur.

5.4.3 Test run

Calibration of the algorithm was done by testing it establishing a correlation between the MSB2K peat core and itself, treating a depth of d=50 cm. as if it was a tephra. Since adding the information that a depth is equal to itself does not add any information at all regarding the MSB2K chronology, a properly callibrated run of BLT1 will produce the same posterior distribution as the distribution produced by BACON (see chapter 2, section 2.7). Figure 5.4 shows the results from BACON, and figure 5.5 shows the results of BLT1 run with a weight of 0.5 for each core, for 10,000 iterations, sampling one of every 10. These graphs are very similar, and we can conclude that these settings should be reasonable for two cores with one matching tephra and 20 sections each.

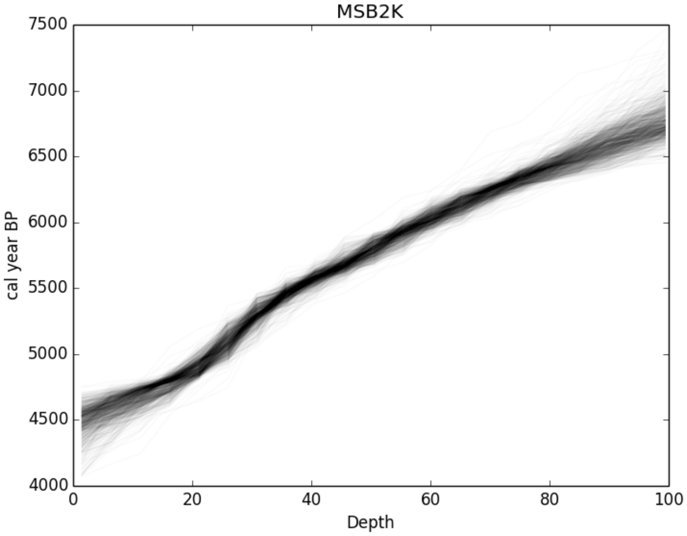

Figure 5.4: The original plot using ordinary BACON with the msb2k core. If BLT1 is properly calibrated then this should be similar to figure 5.5

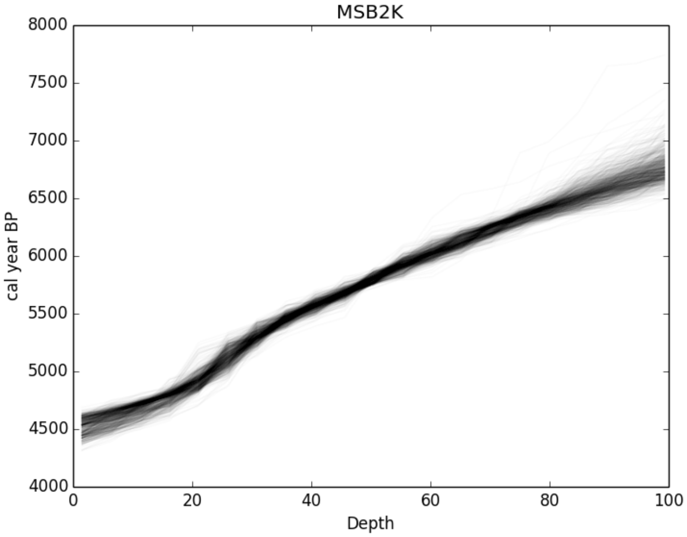

Figure 5.5: The same chronology using BLT1. The graph is very similar and we can conclude that this calibration is reasonable.

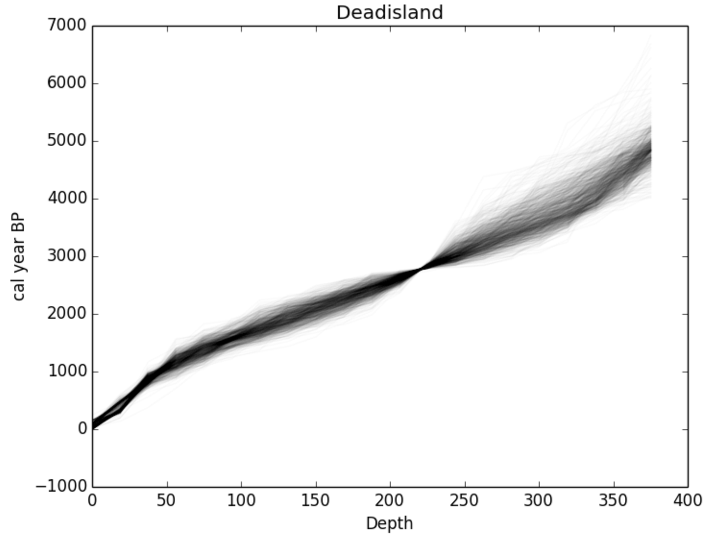

Figure 5.6: Deadisland chronology using BLT1. A tephra at a depth of 221 cm. is equated to one at a depth of 150 cm. in Slieveanorra2. BLT1 reduces uncertainty almost to the point of providing an exact date for the tephra.

5.4.4 Results

One teprha that is strongly associated between Deadisland and Slieveanorra2 is the microlite teprha, found at a depth of 219.5 cm in the Deadisland core and at a depth of 148.5 cm in Slieveanorra2. At these two points there are also ^{14}C measurements, which coincide. Rather than set the program at exactly this depth, it is slightly more interesting from a testing perspective to have the tephra be set where there is no measurement, so instead the program was set to associate a depth of 221 cm in Deadisland with 150 cm in Slieveanorra2.

Using the settings discovered from the callibration, BLT1 produced figures 5.6 and 5.7 for the chronologies of the Deadisland and Slieveanorra2 cores.

The most surprising aspect of these graphs is that the algorithm provides nearly an exact date for the tephra. while the reduction in uncertainty throughout the core is not quite as dramatic as what was obtained from BLT0 in chapter 4, the posterior distribution tightens up dramatically when near the tephra, and shows some improvement throughout. Even though the ^{14}C measurements at the tephra are identical, the information from the sur-

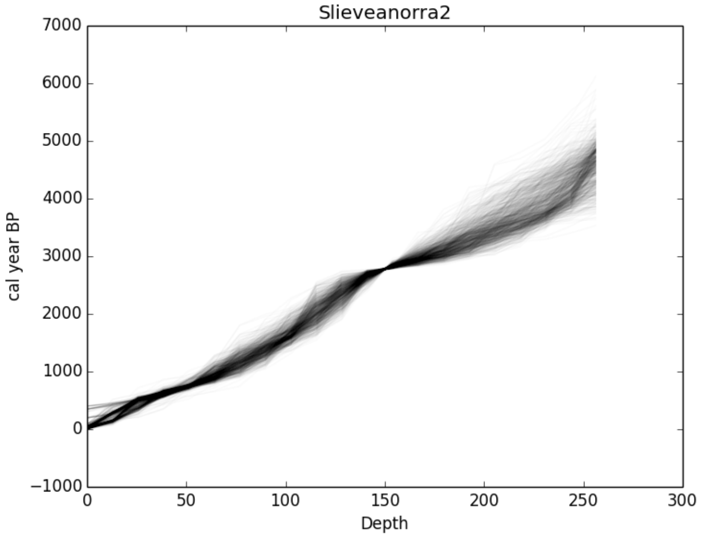

Figure 5.7: Slieveanorra2 chronology using BLT1. The sample at 150 cm. of Slieveanorra2 is identical to the sample at 221 cm. for Deadisland.

rounding cores provides much greater precision.

A second run was done changing the depth of the tephra from the the Slieveanorra2 core to 140 cm. The results are shown in figures 5.8 and 5.9

Although it may look like there is greater dispersion, this is caused by the change of scale; the dispersion is actually less: No elements of the sample predict an age older than 6000 years at any depth. Along with this reduction of the posterior's disepersion, it is also interesting to note that the shape of the Slieveanorra 2 chronology has been slightly altered, creating a very flat section around the depths of 140-150cm.

5.5 BLT2

5.5.1 The model

The situation modeled in BLT1 is significantly more common than the situation of BLT0, but it is still somewhat optimistic. It is not always the case that tephras can be so precisely identified. While it is true that the chemical compositions of tephras resulting from each volcano tend to differ enough to

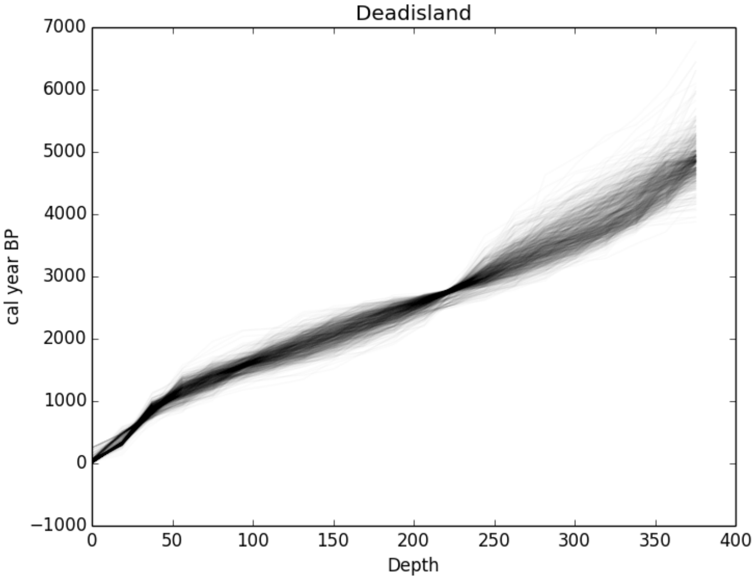

Figure 5.8: Deadisland chronology altering the depth of the Slieveanorra2 tephra to be 140 instead of 150 cm. This time there is slightly more dispersion.

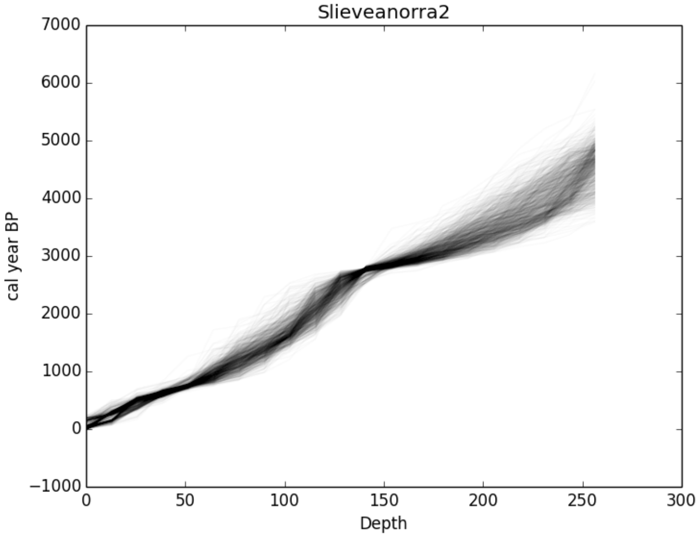

Figure 5.9: Slieveanorra2 chronology changing the depth of the core. In this case the shape of the chronology is slightly altered, favoring low accumulation rates near the tephra.

be distinguished from one another, this will only serve to distinguish tephras created by different volcanos. If a single volcano erupts more than once, it may not be possible to clearly distinguish a tephra created by one eruption from a tephra created by a different one.

It is easy to think that this problem is completely trivial: A volcano erupts twice, and all of the cores we are interested in show one lower and one higher tephra. In this case, assuming the monotonic BACON model is applicable, all of the lower tephras are the same, as are all of the higher ones.

Unfortunately, the picture is not always so clear: Some cores may not express all of the tephras. In this case, rather than associating a tephra in one core to a tephra in another, we instead associate it to one of a set of candidate tephras from the second core. If we allow any number of cores having any combination of different numbers of candidate tephras, the resulting combinations can result in extremely many possibilities and very complex dependence structures. It is not particularly clear that modeling this situation is even particularly useful, since truly complex structures of this kind are rarely found in practice. Rather than solving the general case, the approach

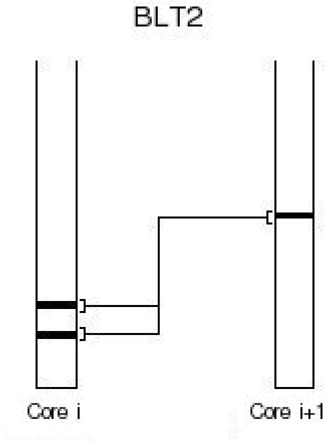

Figure 5.10: The situation modeled by BLT2: It may be unclear which of the tephras in core i corresponds to the teprha in core $i + 1$

has been to choose one particular case which is fairly common: A single core expresses multiple tephras, of which only one is found in the other cores, as illustrated in figure 5.10

In this situation, we will build off of the general BLT model, having $P = (\theta_0, i_0)$ whereθ_0 represents the age of the shared teprha, and i_0 takes on a discrete set of values and indicates which tephra in the multi-tephra core actually corresponds to the tephra in the other cores.

We are free to reorder the cores, so let core 0 be the core in which there are multiple candidate tephras. Let $d_{(0,1)} \dots d_{(0,\delta)}$ be the depths of the candidates. Since there is only one age among the candidates that is actually correlated, we will let θ_0 be the age. Then $(\Theta|\theta_0, i_0)$ will follow the BLT0 model with $a_0 = \theta_0$, $d_0^{(0)} = d_{(0,i_0)}$ and with $d_0^{(i)}$ being the height of the tephra in question for the ith core.

θ_0 has a prior of the same sort that is found in BLT1, and i_0 has prior probabilities $\pi_0 \dots \pi_\delta$ of being $0 \dots \delta$ respectively. Each π_i is positive and $\sum_{i=0}^{\delta} \pi_i = 1$. The log-likelihood is calculated as it was in BACON, and the corresponding log-priors are added to obtain the energy function.

5.5.2 The algorithm

The algorithm used to sample from the posterior distribution is a Metropolis-Hastings algorithm with a three part hybrid kernel, with weights which are chosen ad-hoc, as in BLT1:

The first kernel used is exactly the same BLT0 kernel as was used for BLT1. The number of iterations remains 100, as it has given reasonable results.

The second kernel is the kernel generated by the instrumental distribution which takes $(\Theta, \theta_0, i_0)_i$ and proposes $(\Theta, \theta_0, i_0)_{i+1}$ as $(\Theta_i, \theta_{0,i+1}, i_{0,i})$ with $\theta_{0,i+1}$ distributed uniformly in the range of dates which would continue to be in the support of $\pi^{(j)}(\theta, \bar{x}, w)^{(j)}$ for all cores j. This is essentially the same as the second kernel for BLT1, and we note that since the value of i_0 does not change, its contribution to the energy is the same before and after the transition; hence this prior does not need to be recomputed in order to determine acceptance or rejection.

The third kernel is a kernel generated by the instrumental distribution which takes $(\Theta, \theta_0, i_0)_i$ and proposes $(\Theta, \theta_0, i_0)_{i+1}$ as $(\Theta_i, \theta_{0,i+1}, i_{0,i+1})$ with $i_{0,i+1}$ simulated from the prior distribution for $i_{0,i+1}$ and $\theta_{0,i+1}$ is $G_0(d_{i_{0,i+1}}, \Theta_i)$. We note that this instrumental distribution does not depend on the parameters at all. Since changing i_0 changes the entire function G_0, then the contribution to the energy of this core must be recalculated in order to compute the Metropolis-Hastings quotient.

5.5.3 Calibration

Setting the kernel weights all at 1/3, we can observe that if $d_{(0,1)} = d_{(0,2)}$ then the algorithm will be equivalent to the BLT1 algorithm except with a 1/3 chance of doing nothing for one step. Hence, a number of iterations equal to 3/2 of the iterations used for BLT1 would produce the same results, and hence gives a lower bound for the number of iterations that will be required. This is only a lower bound, since if $d_{(0,1)} \neq d_{(0,2)}$ then the exploration of the posterior given each value for i_0 is desireable, unless one of the values for i_0 is found to be significantly less likely than the other (In which case the more likely value of i_0 will be explored faster anyways). One option is to perform a shorter run and obtain an approximate posterior distribution for i_0. The closer it is to being balanced, the longer the chain should run. This is what has been done for the "results" section.

5.5.4 Results

There are two estimations of interest in BLT2. While it is certainly an interesting tool for chronology building in general, often we may be interested in a much simpler question: Which candidate tephra corresponds to the other tephras?

Using Slieveanorra2 as core 0, $d_{(0,1)} = 120$ and $d_{(0,2)} = 150$ the algorithm was initialized at $i_0 = 1$, deliberately setting it to begin in the wrong place, and with a balanced prior (meaning $P(i_0 = 2) = 0.5$). A short run of 1000 iterations was used to estimate the posterior distribution of i_0, arriving at an estimate of $P(i_0 = 2) \approx 0.996$. BLT2 makes a very clear choice, and designates the correct depth. With such a decisive estimation, the most prudent choice for constructing this chronology would not be to use BLT2, but simply to choose $i_0 = 2$ and use BLT1.

To test chronology building, we need a more balanced posterior. Instead we set $d_{(0,1)} = 140$. The same procedure gives us an estimate of $P(i_0 = 2) \approx 0.825$. A conservative iteration count was chosen to be 30,000 iterations: Three times what was used for BLT1. Samples were taken once every 30 iterations, discarding the first 150. This produces a sample size which is equivalent to what has been used before. The new estimation of the distribution of i_0 is $P(i_0 = 2) \approx 0.961$ which is still quite high (BLT2 is almost too powerful to test properly with these data!) The resuting graphs can be seen in figures 5.11 and 5.12.

Once again the reduction in the uncertainty of the Deadisland core is quite significant. For the Slieveanorra2 core the effect is less dramatic. The biggest change comes in the shape of the core, which favors lower accumulation rates around the depth of the tephra. These rates have a better chance of satisfying the matching of either of the two depths to the Deadisland tephra.

5.6 BLT3

5.6.1 The model

Thus far all forms of BLT have been chosen to incorporate information from tephras only, with no attention given to information which is available from proxies. It is true that information from proxies is often extremely unreliable for dating purposes, but this is not always the case. While gradual trends and changes in proxies may vary greatly even in areas of close geographical

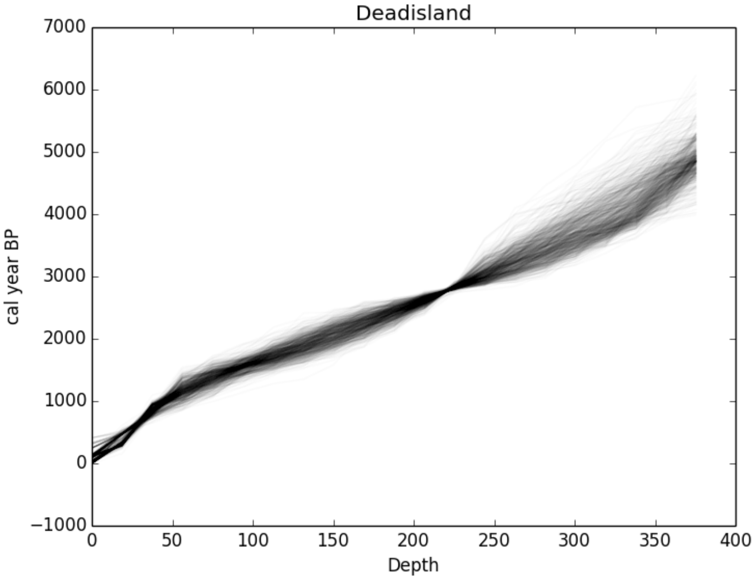

Figure 5.11: Deadisland chronology using BLT2. The tephra in Deadisland is associated to a depth of either 140 or 150 cm in Slieveanorra2. In this situation, the Deadisland chronology is almost identical to the chronology using BLT1.

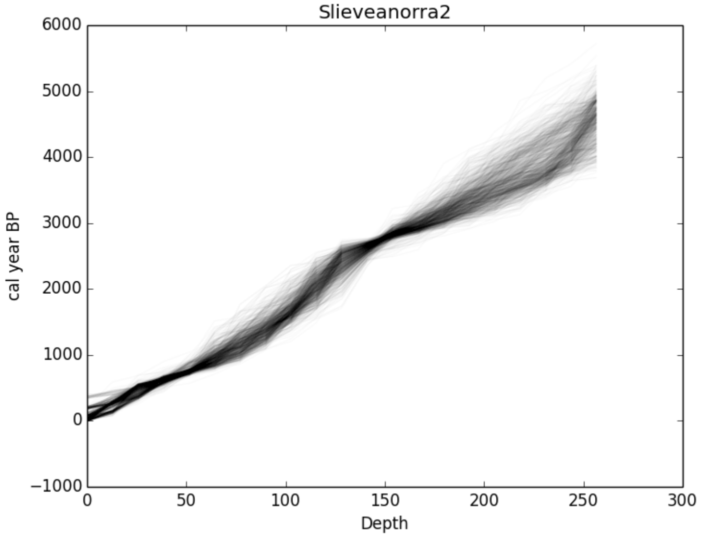

Figure 5.12: Slieveanorra2 chronology using BLT2: The possibility of choosing between a depth of 140 and 150 cm. has made the chronology slightly flatter in this area, favoring slopes which make the age at these depths similar to one another.

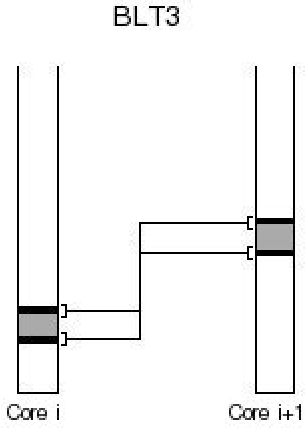

Figure 5.13: The situation modeled in BLT3: There are upper and lower bounds for the depth of an event in two different cores.

proximity, sometimes proxies can be indicators for significant cataclysmic events, which indicate dramatic change and which can be used to establish correlations between chronologies.

An example of an event which can be inferred from proxies is the extinction of a species. It may occur that in more than one core, the pollen of a certain type of flower is found throughout the core up until a certain depth, above which it disappears altogether.

The situation in BLT3 is somewhat similar to BLT1, with one significant difference: The depth of a tephra is known exactly, whereas the depth corresponding to the extinction of the species is not. The highest point at which the pollen is found does not necessarily correspond exactly to the moment of the species's disappearance. It may be that the species was present, but no pollen landed in that particular area, or similarly that the species had already died out, but that some older pollen was still found adrift. What can be done is that an interval can be chosen: The bottom of the interval is a depth at which the species certainly continued to flourish, and the top is a depth at which it was definitely extinct. This situation is illustrated in figure 5.13.

A similar situation can happen sometimes with some tephras, where ashes are occasionally scattered over a range of depths, rather than found at a single depth.

The first approach to model this situation would be simply to put a restriction on the support of Θ such that $\pi(\Theta) = 0$ if the corresponding intervals in different cores are disjoint. While this approach is entirely functional, it has a problem: It does not correspond with the structure of the general BLT model as outlined previously. While this is not a serious issue in itself it does lose any form of conditional independence, and to test the support for any core, there is no simple piece of information that can be checked without having to perform significant computation on all other cores.

A second modeling approach has been chosen instead which corresponds to the general BLT model and which also represents the phenomenon slightly more explicitly: We will use $P = a_0$ as the age of the cataclysmic event which is being modeled. The prior for a_0 is simply uniform over the entire timespan which is being considered (this may seem very broad, but as we will see below, the conditional prior is much more reasonable). For each core j we will have $d_0^{(j)}$ and $d_1^{(j)}$ representing the endpoints of the interval corresponding to the times in which the event is known to have occurred. Then the conditional distribution $\Theta^{(j)}|a_0$ is the BACON model with the support restricted to

$$G^{(j)}(d_0^{(j)}, \Theta^{(j)}) \leq a_0 \leq G^{(j)}(d_1^{(j)}, \Theta^{(j)}).$$

We can now consider the prior conditional distribution of $a_0|\Theta$. Note that the prior for a_0 is uniform, and also that it does not have any bearing on the relative value of the prior of Θ at different points in its support. Now if we consider the sets $A = \{G^{(j)}(d_0^{(j)}, \Theta^{(j)})|0 \leq j \leq l\}$ and $B = \{G^{(j)}(d_1^{(j)}, \Theta^{(j)})|0 \leq j \leq l\}$ then $a_0|\Theta$ is uniform in $[max(A), min(B)]$, which is actually quite reasonable.

Even more interesting, we note that a_0 does not have any relationship to the value of the likelihood either, since the likelihood is simply the likelihood of the BACON model. So long as a_0 changes within $[max(A), min(B)]$, the calculation of the likelyhood does not depend on a_0. This is particularly useful for purposes of calculation.

5.6.2 The algorithm

Simulating from the posterior distribution of (Θ, a_0) was accomplished using an MCMC algorithm with a two part hybrid kernel.

The first kernel is created using a t-walk algorithm which is very similar to BACON. We add the restriction $G^{(j)}(d_0^{(j)}, \Theta^{(j)}) \leq a_0 \leq G^{(j)}(d_1^{(j)}, \Theta^{(j)})$ to the support function, creating an algorithm that - if allowed to run for long enough - would converge to the conditional distribution of $(\Theta|a_0)$. The algorithm is allowed to run for 100 iterations.

The second kernel is generated by using an instrumental distribution which takes $(\Theta, a_0)_i$ and proposes $(\Theta, a_0)_{i+1} = (\Theta_i, a_{o,i+1})$ with $a_{0,i+1}$ generated uniformly over $[max(A), min(B)]$, as defined in the previous section. We have already noted that this is the conditional prior distribution, but we can also note that changing $a_{0,i}$ to $a_{0,i+1}$ does not affect the value of the likelihood, so this is actually the conditional posterior as well. Since the target distribution is exactly the posterior distribution, then simulating from the conditional posterior is exactly a Gibbs kernel, so this second kernel never rejects the proposal from the instrumental distribution.

In order to remain strongly aperiodic, it is required that a rejection be possible, and this happens when the first kernel returns the same parameters as it received originally. For this to happen, the t-walk algorithm must have 100 rejections in a row, which, while extremely unlikely, is not impossible, which is all that is required.

5.6.3 Test run

It makes sense to expect correlation time in BLT3 to be slightly lower than for BLT1. To test this, the algorithm was tested for the MSB2K peat core with 10,000 iterations (the same as was used for BLT1) with $d_0 = 40$ and $d_1 = 50$ on both instances. As with BLT1, the idea is that no new information is added, so we expect the posterior sample to match the sample we obtain with simple BACON. The result can be seen in figure 5.14.

As with BLT1, the result is virtually identical to the result obtained by simply using BACON, so these settings can be considered reasonable.

5.6.4 Results

BLT3 was used on the Deadisland and Slieveanorra2 datasets using $d_0 = 215$, $d_1 = 225$ for Deadisland and $d_0 = 145$, $d_1 = 155$ for Slieveanorra2. The results are in figures 5.15 and 5.16

The improvements are far less dramatic than with any of the previous forms of BLT, the main difference bing in the Slieveanorra2 core near the

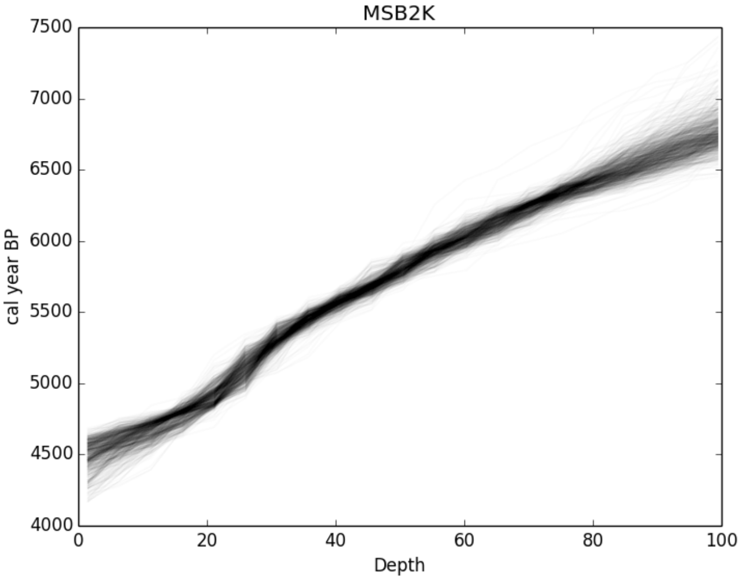

Figure 5.14: MSB2K chronology using BLT3. If the calibration is correct, it should be very similar to the graph which is created by BACON, and it is.

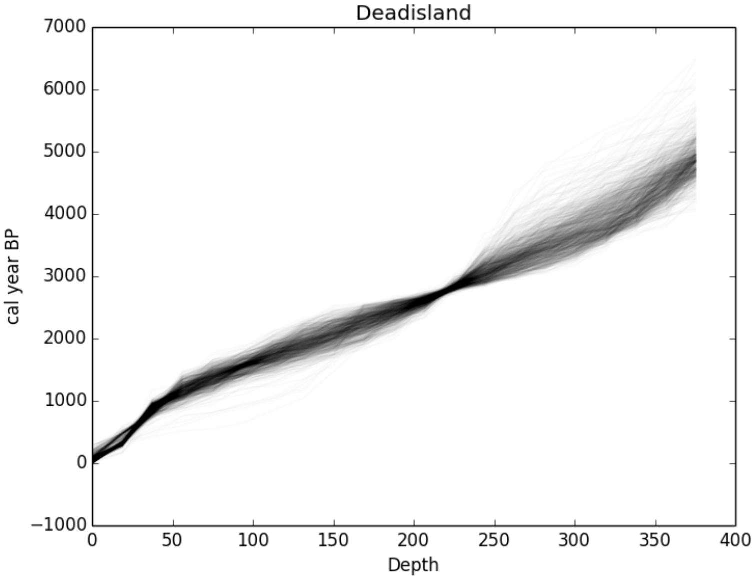

Figure 5.15: Deadisland chronology using BLT3. An event between 215 and 225 cm. is set to correspond to an event between 145 and 155 cm. in Slieveanorra2. The result is not that big of an improvement over the chronology built independently.

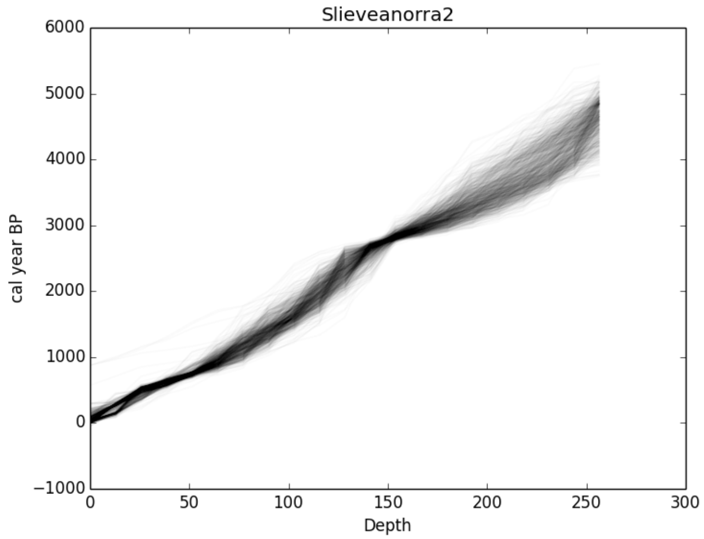

Figure 5.16: Slieveanorra2 chronology using BLT3. There is slightly less uncertainty near the tephra, but the difference is not very big.

tephra, where the uncertainty is slightly reduced.

One possible reason why the results are not so different from the BACON results is that the intervals where the correlation is being supposed is already a region of fairly little uncertainty. As an experiment, BLT3 was used again, using $d_0 = 300$, $d_1 = 310$ for the Deadisland core, and $d_0 = 230$, $d_1 = 240$ for the Slieveanorra core. There is no particular reason to think that these regions are correlated in reality, but a relationship is plausible given the data. Using these settings, BLT3 produced figures 5.17 and 5.18.

Using these settings, a significant difference is found, with lower uncertainties and different shapes in the chronologies. This improvement does not, however, seem to propagate as far beyond the region of interest as they did with previous versions of BLT. This may be because the added information is not as strong, but it also may simply be because of the region near the tephra, where uncertainty is already lower, and is therefore not so affected by the change.

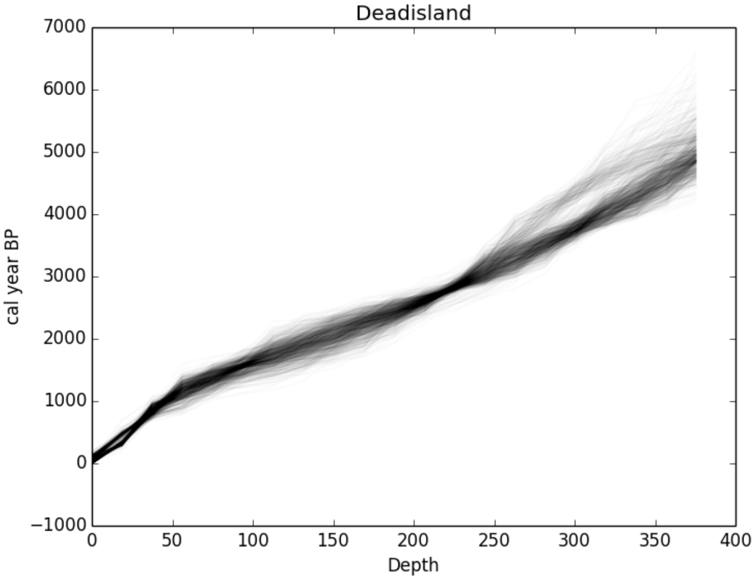

Figure 5.17: Deadisland chronology using BLT3, using for correlation a false event between 300 and 310 cm., and matching it to depths between 230 and 240 cm. in Slieveanorra. This time thre is a big difference, because there was less information available in the specified region.

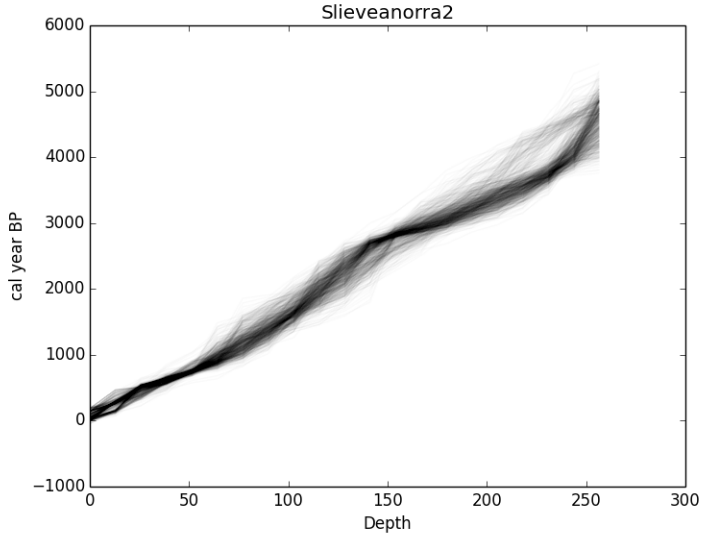

Figure 5.18: Slieveanorra2 chronology using BLT3 and the false event. The entire right half of the chronology has a different shape.

Chapter 6

General discussion and conclusions

6.1 Overview

The techniques investigated in this work have produced favorable results overall. While BACON is a useful tool for chronology building, the techniques presented here have shown that significant improvements can be made over its present structure.

The first change proposed to BACON was a response to the difficulty of compiling it on all targeted operating systems. In order to address this problem, BACON was rewritten from scratch in the Python programming language, and this language was also used for all forms of BLT. Python proved to be able to handle the heavy computational load, but this did come at a significant hit to runtime. As such, it cannot be said that Python is strictly an improvement over the original C++/GSL approach. This being said, a hybrid Python/C++ approach has been suggested which may solve both performance and portability issues.

The second problem which was posed was that of permitting BACON to be used to directly associate proxies with age. This problem was solved using a relatively simple algorithm which makes the results from BACON immediately useful. The algorithm produces a sample from the posterior distribution of proxies for each age of interest. The ages of interest have their samples associated according to which of them came from a certain vector of posterior parameters, thus allowing the estimation of the posterior probability of nearly any event that one might be interested in.

The final challenge was finding ways of using external sources of information, such as the data from other cores, to improve the posterior estimation made by BACON. The scope of this problem is perhaps too broad for there to be any general solution. Four forms of information were chosen, and for each of these a model was proposed which reduced the uncertainty in the posterior estimation. In all cases, besides BLT3, this reduction was significant even in regions where the uncertainty was already relatively low.

6.2 Python implementation

With the exception of the work detailed in chapter 3 (relating to ghost maps and proxy estimation, where ther programming was done in R) the Python programming language has been used throughout this work as an alternative to C++. Despite significant performance hits, Python was able to perform even the most complex of tasks presented.

6.2.1 Performance

The biggest problem with using Python instead of C++ are the performance issues. While C++ is able to perform eight million iterations of BACON in about three minutes, Python takes nearly 20 minutes simply to perform one million iterations (and another three of sample postprocessing, required to produce the graphs). This problem is even worse than it sounds since the number of required iterations scales (nonlinearly) with the dimension of the parameter space. Throughout this work we have consistently used 20 linear sections to create all of our graphs (yielding 22 parameters for simple BACON).

Examining the cost of the program in terms of what functions consume the most time, it appears to be the case that reading specific entries from arrays created by the Python library *numpy* takes up a surprisingly large amount of time. In C++ reading entries from arrays is extremely fast, and does not require any functions from the GSL or any other nonstandard library. Because of this, it has been proposed that C++ could be used to handle array lookups only, letting Python do the rest of the computation. It is possible that this hybrid approach would be enough improve the performance of the program to tolerable levels.

6.2.2 Graphs

In the original form of BACON, C++ was used only as a number cruncher. All forms of graphing was done separately with the R programming language. Unlike C++, Python has a relatively standard library called *matplotlib*, which is able to adequately perform all of the graphing functions required for BACON.

The result of doing all of the calculation and graphing in the same language is that each program can be kept down to a single file.

6.2.3 Libraries and maintenance

The libraries used with Python were *scipy, numpy, matplotlib, cProfile, copy, random,* and *pytwalk*. Of these, *cProfile, copy* and *random* are included by default with a standard Python installation. *Scipy, numpy* and *matplotlib* are all standard libraries, which can easily be installed under nearly any operating system, and which come by default in many Python scientific computation packages, such as Anaconda (http://docs.continuum.io/anaconda/). *Pytwalk* is a library created by A. Christen which contains functions to run the t-walk algorithm using Python. This library, along with installation instructions is found at http://www.cimat.mx/~jac/twalk/.

Since Python is a scripting language, code is never compiled, so maintenance of the program can be done on the code only. In general, Python is a friendly and easy to read language, so no special understanding of the code is needed to make changes. Updates to the libraries used are handled automatically by most operating systems, and are not difficult to do manually.

The main issue with maintenance at present is that many functions in the code are repeated between the different programs (see section 6.4).

6.3 Proxies

From a computational standpoint, the generation of ghost maps to study proxies is a very simple task. The code was written in the R programming language to maintain maximum compatibility with the original BACON format, primarily because this is the section of the code that is closest to being immediately useful as it stands (see section 6.4).

From a practical standpoint, the contribution to ghost maps is probably the most significant. The majority of research in paleoecology (or any science) is not so much focussed on reducing the uncertainty as in producing a greater amount of information. In this sense, ghost maps are able to directly produce information of the sort that is desired, and are overall very useful for studying the trends in proxies over time.

From a statistical standpoint, the most interesting aspect of ghost maps is the change in the regression paradigm for this sort of chronology: In general, the depth is considered to be known and the age of said depth is treated as random, but for ghost maps, this perspective is reversed. The imporant property that was required to do this is the fact that the function G is injective with probability 1 when we condition on the regression parameters. It is hypothesized that for a bayesian regression model with a conditionally injective regression function will allow an inversion of this sort.

6.4 BLT

BLT is a complex process. Even the very basic approach of BLT0 is still a nontrivial extension of BACON from a computational perspective, and further forms of BLT add additional complications to the model. Overall, BLT has proven successful at reducing the uncertainty in BACON. It also may be of interest to study the correlation of the posterior samples for proxy information. It is possible that BLT will reveal information on proxy correlation that is not immediately apparent, and provide useful insights, although this has not been modeled or implemented.

6.4.1 BLT0

BLT0 produces the most dramatic improvements to BACON, and this is unsurprising since the information which has been added is significantly stronger. It also is the simplest of the BLT models and the only one where the difficulty does not scale with the number of cores. The reduction in uncertainty will, of course, be more impressive if the tephra is found in a region where the uncertainty (in the unconstrained chronology) is large.

It is possible to perform BLT0 with either the BLT1 or BLT2 programs, simply by setting the probability of choosing any kernel other than the first to be zero. It is unclear how big the loss of efficiency would be, since this is primarily an issue of how much time is consumed by the initialization of the t-walk algorithm. It is clear, however, that it is better to initialize the algorithm only once, so simply using the BLT0 program as it stands is a better choice.

6.4.2 BLT1

BLT1 is the second simplest of the BLT models, but in terms of implementation BLT3 is simpler. The results from BLT1 on the data that was tested were surprisingly good. The improvement was not as dramatic as in BLT0, but given that the information is much weaker, the difference is not so great.

Since the situation in BLT1 is far more common than the one in BLT0, we can conclude that BLT1 is probably the most useful of the BLT techniques. Like BLT0, the usefulness will relate to the uncertainty in the area, but even when used on an area of low uncertainty, as in the examples seen here, the improvement is very noticeable.

6.4.3 BLT2

The most impressive feature of BLT2 is how sensitive it is when selecting which of the candidate tephras correspond to a tephra in a different core. Even when a cursory glance may indicate that the distribution at the depths of each candidate is not very different, BLT2 still makes a strong choice. When testing this algorithm, most of the time the choice was so clear that it would be reasonable to use BLT2 merely as a preliminary step and then proceed with BLT1.

In those few cases where it makes sense to use BLT2 to infer the chronology directly, the loss when compared to BLT1 is not so great. Overall, if there is any uncertainty at all when identifying tephras, the use of BLT2 is recommended either to identify the tephras clearly, or to build a chronology if the identity of the tephras remains uncertain.

6.4.4 BLT3

BLT3 is the only case in which the improvement over BACON is not always significant. It is strongly influenced by the width of the intervals that are assigned, the level of uncertainty in the regions studied (and the dispairty in these levels between cores), and other factors. Also, it is the only form of BLT in which the correlation is frequently determined not by clear objective evidence, but by human judgment. The judgment applies not only to the widths of correlated intervals, but also frequently to the identification of catastrophic events.

The above being said, if the evidence from proxies is strong enough to provide relatively thin intervals in regions where the number of radiometric measurements is scant, then BLT3 can significantly improve the information about chronology in that region.

6.5 Further work

6.5.1 Computational work

6.5.1.1 Interface

The main problem with the form of the programs as they stand is that there is no interface of any kind. Each program performs one very specific task and if anything even remotely different is desired, then the code itself must be changed. Python is a reasonable language to program interfaces in, and even a very basic interface would be enough to make the programs useful.

The one piece of software that is nearly useable as it stands is the ghost map program, written in R. The interface used for the initial C++ version of BACON uses an R console as a command line interface with R functions as commands. In this sense the ghost map code can be added to the existing R program as a function, allowing it to integrate with the existing interface.

6.5.1.2 Integration

At present, each of the different algorithms is implemented in a different program and exists in a separate file. It would be desireable to allow for mixed forms of correlation to be allowed between cores. Likewise, not all pairs of cores would have all forms of correlations present in the entire set. This would entail bringing all of the different forms of BACON and BLT into a single program, and allowing them to work together. This problem does not involve any new statistical modeling or theory, and is essentially a computational excercise.

6.5.1.3 Performance

The last computational problem with the work as it stands is performance-related. As discussed previously, Python is a less than perfect alternative to C++, and BACON runs fairly slowly. One possible solution, involving a Python/C++ hybrid was detailed in 6.2.1, but it is not completely certain that this will necessarily fix the problem, or that there is no superior alternative which has not been contemplated.

Along with the performance issues that come with the Python programming language, another potential improvement comes in the form of parallel computing. As suggested in 4.2, the structure of the BLT algorithm is a good candidate for parallel computing, and was even designed with parallel computing in mind. In the examples used for this discussion, there were never more than 2 cores used at any time, and parallel computing is unlikely to produce much of an improvement (it may even make the process slower), but with a large number of cores it is reasonable to expect that running the algorithm in parallel would result in significant performance benefits.

6.5.2 Statistical work

Most of the improvements possible arise by extending the functionality of BLT, but ghost maps can also be improved. One suggestion which came up ater completion of this project was to allow for uncertainties in proxy measurements, which, depending on the proxy, may be quite frequent.

The idea of using other sources of information to improve BACON is so broad that most likely it will never be completely covered. In this work four sources of information were used, then models were proposed and implemented for each, but these four sources of information are in no way a complete survey of all of the possible external information which could be used to improve the chronologies.

One area which has sparked some interest is in further restrictions on single-core work in the form of extensions to BLT0. One such extension is to allow for uncertainties in the depth of events of known age (unlike BLT0 which treats the depths of events as exactly known). Another is to allow for ages with known differences (ie: with $G(d_1) - G(d_2)$ known, or having a particular prior distribution).

Paleoecologists frequently attempt to use proxy information for dating in a much more general sense than what is used in BLT3. One direction of interest is a mathematical treatment of the use of proxy trends - rather than only cataclysmic events - to improve chronology building.

Chronology building in general is still an area of great interest, and one very active area of study in paleoecology is chronology building using correlation *only*, without any sort of radiometric data at all. This practice, known as "tuning" has yet to receive any form of adequate mathematical treatment whatsoever, but is nonetheless quite prevalent. One possible further development from the present work is to focus on what information can be modeled based only on correlations with cores where radiometric data is available.

References

- Blaauw,M., Christen, J.A., 2011, “Flexible paleoclimate age-depth models using an autoregressive gamma process.” *Bayesian Analysis* 6, 457-474

- Blaauw, M. and Christen, J.A., 2005, “Radiocarbon peat chronologies and environmental change”, Applied Statistics, 54(4), 805-816

- Christen, J.A. and Fox, C. 2010, "A General Purpose Sampling Algorithm for Continuous Distributions (the t-walk)", Bayesian Analysis, 5(2), 263-282

- Christen, J.A and Pérez E., S. 2009. "A New Robust Statistical Model for Radiocarbon Data", Radiocarbon, 51(3), 1047-1059

- Blaauw et al. 2007, “Testing the timing of radiocarbon-dated events between proxy archives”, The Holocene, 17, 283-288

- Reimer et al 2013, “INTCAL13 and marine13 radiocarbon age calibration curves 0 – 50,000 years cal BP” Radiocarbon , 55, 1869–1887

- Bowman S., 1990, “Radiocarbon Dating (Interpreting the Past)”, University of California Press

- Bernardo, J. and Smith, A., 2000, “Bayesian Theory” Second edition, Wiley

- Robert, C. and Casella, G., 2004, “Monte Carlo Statistical Methods” Second Edition, Springer

- Duque, R.G. , 2008, “Python para todos”, Creative Commons electronic book, it can be found at the URL: http://www.ceibal.edu.uy/contenidos/areas_conocimiento/aportes/python_para_todos.pdf

- Ahreya, K. and Lahiri, S., 2006, “Measure Theory and Probability Theory”, Springer
- Blaauw, M. 2010, “Out of tune: the dangers of alighning proxy archives”, Quaternary Science Reviews, 36, 38-49
- Kalbfleisch, J.G., 1985, “Probability and Statistical Inference volume 2: Statistical Inference” Second Edition, Springer
- Roussas, G., 1997, “A Course in Mathematical Statistics” Second Edition, Academic Press
- Evans, M. Hastings, N. and Peacock, B. “Statistical Distributions”, Third Edition, Wiley
- Norris, J.R., 1997, “Markov Chains”, Cambridge University Press

The code for all of the programs mentioned in this thesis can be found at http://www.cimat.mx/~jac/software/NKK_Thesis_Programs.zip